



REPÚBLICA DEL ECUADOR

# Escuela Politécnica Nacional

"SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

***Respeto hacia sí mismo y hacia los demás.***

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE  
SOFTWARE PARA HELPDESK Y DISTRIBUCIÓN DE CLAVES DE  
ACTIVACIÓN DE LICENCIAS DE SISTEMAS VÍA WEB, PARA LA  
CORPORACIÓN MILENIOSYSTEM S.A.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y REDES DE INFORMACIÓN**

**DAVID HERNÁN GUANO VISCARRA**

**david.guano@gmail.com**

**DIRECTOR: Ing. Xavier Armendáriz, MBA**

**xarmendariz@udla.edu.ec**

**Quito, Septiembre 2011**

## **DECLARACIÓN**

Yo David Hernán Guano Viscarra declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

David Hernán Guano Viscarra

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por David Hernán Guano Viscarra bajo mi supervisión.

---

**Ing. Xavier Armendáriz, MBA**  
**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTO**

A Dios Él sabe como hace las cosas. A mi familia, que siempre me alentó a seguir adelante. A mis hermanos que me sirvieron de ejemplo y aliento para poder enfrentar las adversidades. Y una mención especial al Ing. Xavier Armendáriz que oportunamente supo darme la mano cuando lo necesite.

## **DEDICATORIA**

Dedico este logro a mis padres y en especial a mis hermanos, el logro es de todos nosotros, sin su sacrificio y ejemplo no habría podido continuar y conseguir esta meta.



## ÍNDICE GENERAL

<b>ÍNDICE DE CONTENIDOS.....</b>	<b>I</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>VIII</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>X</b>
<b>RESUMEN.....</b>	<b>XII</b>
<b>PRESENTACIÓN.....</b>	<b>XIII</b>

## ÍNDICE DE CONTENIDOS

<b>CAPÍTULO 1. APLICACIONES EMPRESARIALES WEB. PLATAFORMAS Y METODOLOGÍAS DE DESARROLLO.....</b>	<b>1</b>
1.1 <b>APLICACIONES EMPRESARIALES WEB.....</b>	<b>1</b>
1.1.1 <b>DEFINICIÓN.....</b>	<b>1</b>
1.1.2 <b>CARACTERÍSTICAS DE UNA APLICACIÓN EMPRESARIAL.....</b>	<b>2</b>
1.1.2.1 <b>Arquitectura de las Aplicaciones Empresariales.....</b>	<b>2</b>
1.1.2.1.1 <b>Modelo Lógico.....</b>	<b>2</b>
1.1.2.1.2 <b>Modelo Físico – Capas (Tiers).....</b>	<b>3</b>
1.1.3 <b>HISTORIA DE APLICACIONES EN LA WEB.....</b>	<b>5</b>
1.2 <b>PLATAFORMAS DE DESARROLLO DE APLICACIONES             EMPRESARIALES .....</b>	<b>7</b>

1.2.1	LA COMUNICACIÓN ENTRE APLICACIONES.....	7
1.2.1.1	Protocolos de comunicación en las redes locales.....	7
1.2.1.2	Comunicación de aplicaciones y la web.....	8
1.2.1.3	Plataformas Alternativas.....	9
1.2.2	JEE vs .NET.....	10
1.2.3	JEE.....	11
1.2.3.1	Características JEE.....	12
1.2.3.2	Arquitectura J2EE.....	12
1.2.3.3	Paquetes de aplicaciones en JEE.....	13
1.2.3.4	Evolución de JEE.....	14
1.2.4	.NET.....	14
1.2.4.1	Objetivos de la plataforma .NET.....	16
1.2.4.2	Arquitectura Empresarial .NET.....	17
1.2.4.3	Evolución de .NET.....	18
1.3	METODOLOGÍAS DE DESARROLLO DE APLICACIONES.....	19
1.3.1	CLASIFICACIÓN DE LAS METODOLOGÍAS.....	19
1.3.2	METODOLOGÍAS TRADICIONALES.....	19
1.3.3	METODOLOGÍAS ÁGILES.....	20
<b>CAPÍTULO 2. REQUERIMIENTOS DE LA APLICACIÓN. ESTÁNDARES APLICABLES A HELPDESK EN SISTEMAS WEB.....</b>		<b>22</b>
2.1	ANTECEDENTES DEL PROBLEMA.....	22

2.1.1	SITUACIÓN ACTUAL DEL PROCEDIMIENTO DE ACTIVACIÓN DE LICENCIAS DE SISTEMAS EN LA CORPORACIÓN.....	22
2.1.2	SITUACIÓN ACTUAL DEL PROCEDIMIENTO DE AYUDA A CLIENTES EN LA CORPORACIÓN .....	25
2.2	SOLUCIÓN PROPUESTA.....	26
2.2.1	ELECCIÓN DE LA PLATAFORMA DE DESARROLLO.....	26
2.2.2	REQUERIMIENTOS DE LA APLICACIÓN Y SELECCIÓN DE HERRAMIENTAS DE DESARROLLO.....	27
2.2.2.1	Robustez y Seguridad.....	27
2.2.2.2	Interfaz web flexible.....	28
2.2.2.3	Rendimiento.....	29
2.2.2.4	Gestión de los datos.....	30
2.2.2.4.1	Gestor de bases de datos.....	31
2.2.2.4.2	Proveedor de Persistencia.....	33
2.2.2.5	Portabilidad y escalabilidad.....	34
2.2.2.6	Presentación de la información.....	35
2.3	PRACTICAS PARA GESTIONAR Y OPTIMIZAR LA AYUDA QUE PRESTA UNA APLICACIÓN WEB.....	36
2.3.1	ANTECEDENTES.....	36
2.3.2	ITIL.....	38
2.3.2.1	Definición.....	38
2.3.2.2	Objetivos.....	38
2.3.2.3	Beneficios.....	39

2.3.2.4	Versiones de ITIL.....	40
2.3.2.5	Estructura.....	40
2.3.3	GESTIÓN DEL SERVICIO – ITIL V2.....	42
2.3.3.1	Entrega del servicio.....	42
2.3.3.1.1	Availability Management (Gestión de la disponibilidad).....	42
2.3.3.1.2	Capacity Management (Gestión de la Capacidad).....	42
2.3.3.1.3	Financial Management for IT services (Gestión Financiera para los servicios de TI).....	42
2.3.3.1.4	Service Continuity Management (Gestión para la continuidad del servicio).....	43
2.3.3.1.5	Service Level Management (Gestión de los niveles del servicio).....	43
2.3.3.2	Soporte del servicio.....	43
2.3.3.2.1	Service Desk.....	44
2.3.3.2.2	Configuration Management (Gestión de la configuración)....	45
2.3.3.2.3	Incident Management (Gestión de Incidentes).....	46
2.3.3.2.4	Problem management (Gestión de problemas).....	46
2.3.3.2.5	Changing management (Gestión de los cambios).....	46
2.3.3.2.6	Release management (Gestión de las versiones).....	46
2.3.4	HELP DESK .....	47

## **CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN....48**

3.1	HERRAMIENTAS DE LA METODOLOGÍA XP.....	48
-----	--	----

3.1.1	ITERACIONES.....	49
3.1.2	HISTORIAS DE USUARIO.....	49
3.1.2.1	Elementos de una Historia de usuario.....	49
3.1.3	TAREAS DE UNA HISTORIA DE USUARIO.....	50
3.1.3.1	Elementos de una Tarea.....	50
3.2	PLANIFICACIÓN DEL PROYECTO APLICANDO LA METODOLOGÍA.....	50
3.2.1	REDACCION DE LAS HERRAMIENTAS DE LA PRIMERA ITERACIÓN.....	53
3.2.1.1	Administración de Tablas Base.....	53
3.2.2	REDACCION DE LAS HERRAMIENTAS DE LA SEGUNDA ITERACIÓN.....	68
3.2.2.1	Administración de Contratos.....	68
3.2.2.2	Módulo de Obtención de Claves de activación de Licencias.....	70
3.2.2.3	Módulo de Mesa de Ayuda para resolución de problemas comunes...71	
3.2.3	REDACCION DE LAS HERRAMIENTAS DE LA TERCERA ITERACIÓN.....	72
3.2.3.1	Administración de Licencias obtenidas e Incidencias ocurridas.....	72
3.3	IMPLEMENTACION DEL PROTOTIPO DE SOFTWARE.....	74
3.3.1	DISEÑO DE BASE DE DATOS.....	74
3.3.2	HERRAMIENTAS DE SOFTWARE.....	76
3.3.2.1	Servidor de Base de Datos.....	76

3.3.2.2 IDE y Servidor de Aplicaciones.....	77
3.3.3 INTERFACES DE USUARIO.....	81
3.3.3.1 Interfaz de ingreso a la aplicación.....	82
3.3.3.2 Menús de la aplicación.....	82
3.3.3.3 Formularios de visualización e ingreso de información.....	84
3.3.3.4 Interfaz del módulo de Obtención de claves.....	85
3.3.3.5 Interfaz del módulo de Mesa de Ayuda(Helpdesk).....	86
3.3.3.6 Interfaces para evaluación de información ingresada por el cliente.....	88
3.3.3.7 Seguridad de la aplicación.....	89
3.3.4 CODIGO.....	90
3.4 PRUEBAS DE ACEPTACION DEL PROTOTIPO DE SOFTWARE.....	91
3.4.1 HU1: ADMINISTRACION DE LA EMPRESA.....	91
3.4.2 HU2: ADMINISTRACION DE LOS VENDEDORES.....	92
3.4.3 HU3: ADMINISTRACION DE USUARIOS.....	93
3.4.4 HU4: ADMINISTRACION DE PROBLEMAS DEL SISTEMA.....	94
3.4.5 HU5: ADMINISTRACIÓN DE LOS MÓDULOS DEL SISTEMA.....	95
3.4.6 HU6: ADMINISTRACION DE SOLUCIONES A LOS PROBLEMAS.....	96
3.4.7 HU7: ADMINISTRACION DE SUGERENCIAS DE SOLUCIÓN.....	97
3.4.8 HU8: INGRESO A LA APLICACIÓN.....	98
3.4.9 HU9: VERIFICACIÓN DE INFORMACIÓN DE USUARIO.....	99
3.4.10 HU10: ADMINISTRACION DE CONTRATOS.....	100
3.4.11 HU11: OBTENCIÓN DE CLAVES DE ACTIVACIÓN DE LICENCIAS.....	101
3.4.12 HU12: MESA DE AYUDA, UBICACIÓN DEL PROBLEMA.....	102

3.4.13	HU13: MESA DE AYUDA, SOLUCIÓN AL PROBLEMA.....	102
3.4.14	HU14: ADMINISTRACION DE LICENCIAS OBTENIDAS.....	103
3.4.15	HU15: ADMINISTRACION INCIDENCIAS REGISTRADAS.....	104
<b>3.5</b>	<b>COSTO Y RESULTADOS DE LA APLICACION.....</b>	<b>105</b>
3.5.1	COSTO DEL DESARROLLO DE LA APLICACIÓN.....	105
3.5.1.1	Costo de la aplicación.....	106
3.5.2	RESULTADOS DE LA APLICACIÓN.....	107
<b>CAPÍTULO 4.CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>109</b>
4.1	CONCLUSIONES.....	109
4.2	RECOMENDACIONES.....	111
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>114</b>
<b>ANEXOS.....</b>		<b>A</b>
ANEXO A – Diccionario de Datos.....		B
ANEXO B – Manual de Instalación.....		H
ANEXO C – Manual de Usuario.....		U
ANEXO D – Código.....		AA

## ÍNDICE DE FIGURAS

### CAPÍTULO 1. APLICACIONES EMPRESARIALES WEB. PLATAFORMAS Y METODOLOGÍAS DE DESARROLLO

<b>Figura 1.1</b> – Modelo Lógico en Aplicaciones Empresariales.....	3
<b>Figura 1.2</b> – Modelos Físicos de 2 capas en aplicaciones Empresariales.....	4
<b>Figura 1.3</b> - Modelos Físicos de 3 capas en aplicaciones Empresariales.....	4
<b>Figura 1.4</b> – Arquitectura JEE.....	13
<b>Figura 1.5</b> – Arquitectura .NET.....	18

### CAPÍTULO 2. REQUERIMIENTOS DE LA APLICACIÓN. ESTÁNDARES APLICABLES A HELPDESK EN SISTEMAS WEB

<b>Figura 2.1</b> – Activación de Licencia, paso1.....	23
<b>Figura 2.2</b> – Activación de Licencia, paso2.....	23
<b>Figura 2.3</b> – Activación de Licencia, paso3.....	24
<b>Figura 2.4</b> – Proceso JSF.....	29
<b>Figura 2.5</b> – Elementos del marco de trabajo de ITIL.....	41

### CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN

<b>Figura 3.1</b> – Modelo Físico de la base de datos.....	75
<b>Figura 3.2</b> – Consola de administración Mysql.....	76
<b>Figura 3.3</b> – Interfaz de administración de phpMyadmin.....	76
<b>Figura 3.4</b> – Prestaciones de Netbeans 6.8.....	77
<b>Figura 3.5</b> – Consola de administración de Glassfish v3.0.....	77
<b>Figura 3.6</b> – Configuración de un pool de conexiones.....	79
<b>Figura 3.7</b> – Comprobación de la conexión.....	79
<b>Figura 3.8</b> – Configuración del recurso JDBC.....	79
<b>Figura 3.9</b> – Proyecto JEE Aplicación Web sasm.....	81
<b>Figura 3.10</b> – Interfaz de ingreso a la aplicación.....	82
<b>Figura 3.11</b> – Menú principal de la aplicación (vista del administrador).....	82
<b>Figura 3.12</b> – Menú principal de la aplicación (vista del cliente).....	83
<b>Figura 3.13</b> – Zona de administradores.....	83
<b>Figura 3.14</b> – Visualización de los datos de la empresa.....	84
<b>Figura 3.15</b> – Lista de Usuarios de la aplicación.....	84
<b>Figura 3.16</b> – Interfaz para obtención de claves.....	85
<b>Figura 3.17</b> – Interfaz para obtención de claves, visualización de clave.....	85
<b>Figura 3.18</b> – Interfaz para obtención de claves, claves agotadas.....	86
<b>Figura 3.19</b> – Mesa de ayuda, ubicación del problema.....	86
<b>Figura 3.20</b> – Mesa de ayuda, sugerencia de solución.....	87
<b>Figura 3.21</b> – Confirmación de la incidencia.....	87
<b>Figura 3.22</b> – Incidencias registradas.....	88
<b>Figura 3.23</b> – Licencias obtenidas.....	88

**Figura 3.24** – Interfaz de la aplicación cuando no existe un usuario registrado.....89

**Figura 3.25** – Fallo de la autenticación en la aplicación.....89

## ÍNDICE DE TABLAS

### CAPÍTULO 1. APLICACIONES EMPRESARIALES WEB. PLATAFORMAS Y METODOLOGÍAS DE DESARROLLO

**Tabla 1.1** – Evolución del estándar J2EE.....14

**Tabla 1.2** – Lenguajes que soporta la plataforma .NET.....16

**Tabla 1.3** – Evolución de .NET.....19

### CAPÍTULO 2. REQUERIMIENTOS DE LA APLICACIÓN. ESTÁNDARES APLICABLES A HELPDESK EN SISTEMAS WEB

**Tabla 2.1** – Comparativa .NET vs JEE .....27

**Tabla 2.2** – Frameworks MVC para JEE.....28

**Tabla 2.3** – Servidores de Aplicaciones para JEE.....30

**Tabla 2.4** – Gestores de Bases de datos compatibles con JEE.....32

**Tabla 2.5** – Proveedores de Persistencia tipo ORM.....33

**CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN.**

<b>Tabla 3.1</b> – Herramientas de la metodología XP en nuestro desarrollo.....	51
<b>Tabla 3.2</b> – Perfiles de Usuario.....	52
<b>Tabla 3.3</b> – Perfiles de usuario para cada departamento de la empresa.....	52
<b>Tabla 3.4</b> – Cálculo del costo de la aplicación.....	107
<b>Tabla 3.5.-</b> Evaluación del cuestionario SUS para la aplicación implementada.....	110

## RESUMEN

Este proyecto inicia con un análisis de las características que debe tener una aplicación que se encargará de distribuir información privada y además tendrá información de ayuda disponible y personalizada para cada uno de los clientes de la empresa. Luego se hará un análisis comparativo de las plataformas y metodologías de desarrollo aplicadas a sistemas web, a fin de determinar cuáles son las apropiadas de usar en el prototipo.

Luego se estudiará la situación actual de la empresa y los procedimientos que realiza para intentar solventar estos problemas, a fin de determinar algunos requerimientos para el diseño de la aplicación, principalmente para el manejo de la información privada a almacenar y enviar. Además se realizará el análisis de procesos basados en los conceptos ITIL para estandarizar las operaciones en helpdesk para determinar las mejores prácticas aplicadas a nuestro prototipo.

Luego se diseña y elabora la aplicación sobre una plataforma específica y con la metodología de desarrollo de software seleccionada. Durante este proceso se deberá tomar en cuenta criterios, requerimientos y estándares ya definidos.

Una vez terminado el desarrollo del prototipo realizamos la implementación a nivel local en la intranet de la empresa con fin de probar su funcionamiento. Se efectuarán procedimientos de prueba de software según estándares de prueba o con los definidos en la metodología de desarrollo seleccionada.

Finalmente se determina el costo de la aplicación y se evaluará los resultados obtenidos.

## PRESENTACIÓN

En la actualidad una de las principales preocupaciones de las empresas que comercializan productos que implican desarrollo intelectual como por ejemplo sistemas informáticos, es el impedir que se violen los derechos reservados de autor, reproduciendo ilegalmente el software instalador. Este problema se complica aún más cuando la venta del producto se intensifica y se expande sobrepasando los límites de un par de ciudades. Una solución preliminar es la de programar el software para que su licencia se active con una clave que solo la empresa la puede conocer. Sin embargo este procedimiento está expuesto a abrir brechas de seguridad conforme el volumen de clientes vaya aumentando. Es por eso que una empresa de estas características como la Corporación Mileniosystem S.A. se ve obligada a buscar una solución a este problema. Al mismo tiempo el éxito de una implementación masiva de este tipo de software depende mucho del soporte y la atención al cliente, lo que se conoce como helpdesk, y para una empresa el poder brindar soporte rápido y fácilmente a un cliente , es otro objetivo a alcanzar.

Gracias al internet existen varias aplicaciones que sirven para intercambiar información comercial entre vendedor y cliente, y entre estas sobresalen las aplicaciones empresariales que usan recursos web para brindar soluciones para requerimientos específicos. Basándose en la experiencia en implementación y registro de licencias se puede adelantar que una aplicación que esté disponible en internet y que permita dar una solución a los problemas expuestos, representa un avance en el desarrollo y desempeño de este tipo de empresas.

# **CAPÍTULO 1. APLICACIONES EMPRESARIALES WEB. PLATAFORMAS Y METODOLOGÍAS DE DESARROLLO**

## **1.1 APLICACIONES EMPRESARIALES WEB**

“Las aplicaciones empresariales son la sangre vital que nutre a un negocio, poniendo en contacto a sus empleados para colaborar y compartir información, y proporcionando los vínculos necesarios para establecer una comunicación efectiva con sus clientes, socios y proveedores. Los ejecutivos dependen de estas aplicaciones para obtener un acceso fácil y rápido a la información que necesitan con el fin de tomar decisiones informadas y mantener los niveles más altos de satisfacción del cliente.”<sup>[1]</sup>

La implementación de aplicaciones empresariales es una respuesta a las necesidades de empresarios por establecer en su industria las prácticas generalmente aceptadas. Cuando se habla de una implementación de software empresarial, las compañías tienen que considerar la elección de tecnología y el impacto que tendrá en la gente. La capacitación y soporte de aplicaciones empresariales permite el aprendizaje efectivo y la administración del cambio con una sola plataforma para los diferentes procesos.

Gracias al Internet y a la tecnología Web<sup>1</sup> estas aplicaciones han adquirido un campo donde explotar al máximo las necesidades de los clientes.

### **1.1.1 DEFINICIÓN**

Una aplicación empresarial representa un software que trabaja en beneficio de una empresa satisfaciendo necesidades específicas de una o varias áreas del negocio. Una aplicación web es una aplicación en la cual un usuario por medio de

---

<sup>1</sup> World Wide Web, sistema compuesto de una serie de archivos enlazados accesibles a través de internet.

un navegador realiza peticiones a una aplicación remota accesible a través de Internet o de la red local y que recibe una respuesta que se muestra en su propio navegador. Una aplicación empresarial web es un software que cumple su propósito empresarial ayudándose de la tecnología web.

### **1.1.2 CARACTERÍSTICAS DE UNA APLICACIÓN EMPRESARIAL**

Una aplicación empresarial debería ser:

- Heterogénea: por dejar abierta la posibilidad de su desarrollo con múltiples plataformas, sistemas operativos y lenguajes de desarrollo.
- Fiable: brindando confianza en que va a cumplir sus objetivos.
- Segura: garantizando el control de acceso, la autorización y el transporte de datos.
- Robusta: siendo lo suficientemente tolerante a fallos.
- Escalable: diseñada de tal forma que se facilite la ampliación y modificaciones necesarias.
- Con alta disponibilidad: debe ser fácil su adquisición, ejecución y actualización.
- De fácil mantenimiento: debe ser fácil mantener el sistema mediante la actualización de sus datos y componentes”.

Además de todas estas características, una aplicación empresarial debe poseer total comprensión de una lógica de negocio. Para entender este concepto debemos analizar la arquitectura de una aplicación de este tipo.

#### **1.1.2.1 Arquitectura de las Aplicaciones Empresariales**

La arquitectura de una aplicación representa la forma en la que es diseñada tanto física como lógicamente.

### 1.1.2.1.1 Modelo Lógico

En el modelo lógico se especifica la estructura de la aplicación y sus componentes sin tener en cuenta donde se localizará el software ni el hardware ni la infraestructura. En la figura 1.1 podemos observar el modelo lógico que se representa en las siguientes capas:

- Gestor de Presentación: Muestra la interfaz de usuario.
- Lógica de Presentación: Define la información que se muestra.
- Lógica de Aplicación: Define las funciones de la aplicación.
- Lógica de negocio: Define requerimientos de la empresa fijos para todas las aplicaciones.
- Lógica de Datos: Provee la definición lógica de los datos (tablas, consultas, vistas, tipos de datos, claves)
- Gestor de Datos: Se encarga de escribir y leer en la base de datos.



**Figura 1.1 Modelo Lógico en Aplicaciones Empresariales** <sup>[2]</sup>

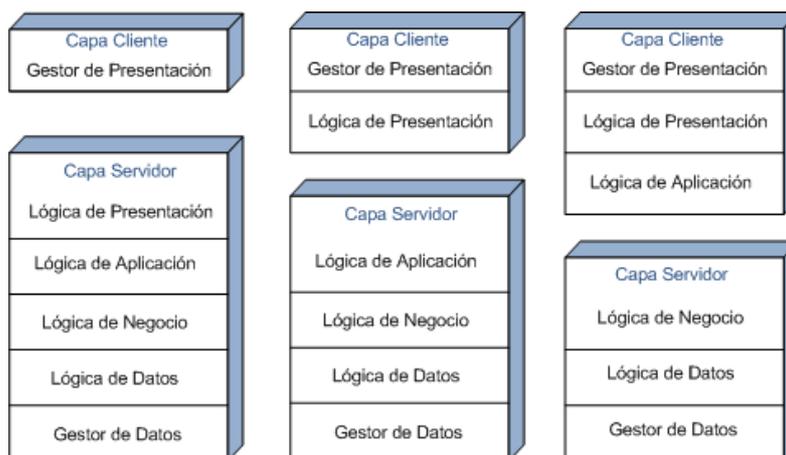
### 1.1.2.1.2 Modelo Físico – Capas (Tiers)

El modelo físico se refiere al lugar donde estarán las piezas de la aplicación. Una aplicación con una sola capa en el modelo físico, corresponde a un sistema monolítico<sup>2</sup> en el cual todas las capas del modelo lógico están embebidas en una

---

<sup>2</sup> Concentra todas las funcionalidades posibles dentro de un solo programa.

so la capa llamada capa de aplicación. A partir de dos capas en el modelo físico, ya podemos hablar de una arquitectura cliente / servidor<sup>3</sup>. Aquí las capas del modelo lógico se distribuirán en una capa cliente y en una capa servidor. Esta distribución es la que determinara si el modelo físico tiene dos o más capas. En la figura 1.2 podemos observar tres variantes de la distribución de capas en el modelo físico de dos capas.



**Figura 1.2 Modelos Físicos de 2 capas en Aplicaciones Empresariales <sup>[2]</sup>**

Cuando se habla de tres capas, el servidor absorbe las operaciones que enlazan al cliente con el manejo de los datos y toma el nombre de capa de negocio. En la figura 1.3 vemos dos variantes de la distribución de capas en el modelo físico de tres capas.



**Figura 1.3 Modelos Físicos de 3 capas en Aplicaciones Empresariales <sup>[2]</sup>**

<sup>3</sup> Consta de una aplicación que funciona como servidor que ofrece un servicio a otra llamada cliente, y que se pueden ejecutar en el mismo o en diferentes lugares.

Sin duda la evolución que tuvieron estas arquitecturas dependió de cómo se fueron desarrollando las tecnologías web. Gracias a estas, la comunicación entre las capas ha alcanzado altos niveles de satisfacción, tanto para el desarrollador como para el usuario.

### 1.1.3 HISTORIA DE APLICACIONES EN LA WEB

En un inicio la web era sencillamente una colección de páginas estáticas, documentos, etc., para su consulta o descarga luego vino la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico, o sea, generar datos a partir de la petición. Este método fue conocido como CGI<sup>4</sup> ("Common Gateway Interface") y definía como se podía pasar información entre el servidor y programas externos. Los CGIs siguen utilizándose ampliamente; la mayoría de los servidores web permiten su uso debido a su sencillez. Además, dan total libertad para elegir el lenguaje de programación que se desea emplear.

Sin embargo el punto débil de los CGIs es que cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl o Python<sup>5</sup>, o en lenguajes que requerían "run-time environment", como Java o VisualBasic<sup>6</sup>, el servidor se veía sometido a una gran carga debido a la concurrencia y esto representaba un gran problema.

Por este motivo se empezaron a desarrollar alternativas a los CGIs que solucionarían el problema del rendimiento.

---

<sup>4</sup> Es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es añadir mayor interacción a los documentos web que por medio del HTML se presentan de forma estática.

<sup>5</sup> Son lenguajes de programación diseñados para ser ejecutados por medio de un intérprete.

<sup>6</sup> Son lenguajes que compilan a bytecode. Compilar consiste en traducir el bytecode a código máquina nativo en tiempo de ejecución.

Primero se diseñaron sistemas de ejecución de módulos que se integran de mejor manera con el servidor, evitando la instanciación y ejecución de varios programas. Al mismo tiempo se intento dotar a los servidores con un intérprete de algún lenguaje de programación que permita incluir el código en las páginas de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Gracias a esto se produjo un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones web. Todas siguen alguna de estas vías, pero lo más útil ha sido fusionar los mecanismos citados anteriormente, dando como resultado un lenguaje integrado que permita al servidor interpretar comandos en las páginas HTML<sup>7</sup> y, además, se busco tener un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento que tienen los CGIs.

De inmediato empresas como Sun Microsystems presentaron su solución basada en sus propios métodos. Y también surgieron lenguajes con mucha acogida como PHP<sup>8</sup> por ejemplo. PHP es un lenguaje interpretado que permite la incrustación de HTML en los programas, con una sintaxis derivada de C y Perl. Existen otros métodos, que basan su solución vinculándose a un servidor web concreto, como mod\_perl para Apache<sup>9</sup> o RXML para Roxen<sup>10</sup>. No tardaría mucho tiempo hasta que algunas de estas tecnologías dieran lugar a la creación de plataformas orientadas al desarrollo de aplicaciones empresariales que trabajen en ambientes tanto locales como externos con la ayuda de la Web.

---

<sup>7</sup>HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web.

<sup>8</sup> PHP del inglés, Hipertext preprocessor, es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor.

<sup>9</sup> Tecnología de desarrollo que trabaja junto con el servidor Apache

<sup>10</sup>Tecnología de desarrollo que trabaja junto con el servidor Roxen.

## **1.2 PLATAFORMAS DE DESARROLLO DE APLICACIONES EMPRESARIALES**

### **1.2.1 LA COMUNICACIÓN ENTRE APLICACIONES**

En un inicio las aplicaciones funcionaban con la ayuda de computadores de propósito especial a las cuales se conectaban terminales para que los usuarios puedan acceder a algunos servicios. Varios años después surgió el computador personal o PC que podía albergar aplicaciones que los mismos usuarios puedan ejecutar.

“En los años 80’s especialmente en el sector de estos computadores personales, los protocolos de comunicación no ocupaban un lugar demasiado importante para los desarrolladores, la dificultad consistía en que varias aplicaciones se comunicaran entre sí. En los 90’s, algunas estructuras de objetos, como COM (Component Object MODEL) de Microsoft y CORBA (Common Object Request Broker Architecture), que se comercializó como una iniciativa entre proveedores de OMG (Grupo de Gestión de Objetos), cobraron popularidad.”<sup>[3]</sup>

El modelo COM y la arquitectura CORBA fueron diseñados para la escritura y encapsulamiento del código binario en componentes que eran llamados por los programadores desde cualquier aplicación. COM y CORBA no inter-operaban fácilmente y ya que la atención la tenían las computadoras personales independientes, la comunicación entre aplicaciones aun no era necesario.

#### **1.2.1.1 Protocolos de comunicación en las redes locales**

Gracias a la popularidad de las redes locales a principios de los 90’s, y la conexión entre máquinas la comunicación entre aplicaciones se volvió una prioridad. Los fabricantes y las organizaciones que ya contaban con sus propias estructuras de modelo-objeto, empezaron a estandarizarlas creando sus protocolos para permitir la comunicación a través de redes.

“OMG estableció el IIOP (Internet Inter-ORB Protocol) como protocolo de comunicación de CORBA. Microsoft introdujo el DCOM (Distributed COM) como su protocolo de comunicación que permitía cruzar las fronteras entre equipos. Sun Microsystems lanzó el RMI (Remote Method Invocation) utilizado por los usuarios de Java.”<sup>[3]</sup>

### **1.2.1.2 Comunicación de aplicaciones y la web**

La comunicación entre aplicaciones mediante el uso de los protocolos mencionados tuvo un aceptable funcionamiento, en especial cuando dichas aplicaciones se encuentran en la misma red local.

Pero cuando apareció el concepto de internet y la Web, la red creció y se volvió extremadamente distribuida y descentralizada. Ni personas ni empresas eran capaces de tomar la decisión sobre qué sistema operativo o entorno de programación/lenguaje, se ejecutaría en los computadores conectados a Internet. Esto hizo dar cuenta de que la manera como se comunicaban las aplicaciones en redes locales no servía dentro de estos nuevos conceptos.

“En cuanto a aplicaciones dinámicas se refiere, primero se implantó la interoperabilidad entre aplicaciones y sitios web por medio de los Frames<sup>11</sup>. Posteriormente se empleo CGI para la publicación de la información (HTTP GET, HTTP POST)<sup>12</sup>.

“Tras la aparición de XML<sup>13</sup> a mediados de los noventa, los desarrolladores se encontraron con la posibilidad de expresar una estructura de información y mensajes de forma auto descriptivo y uniforme, lo que impulso a XML, para aplicar un formato a los mensajes intercambiados entre sistemas.”<sup>[3]</sup>

---

<sup>11</sup> Son la herramienta que el HTML pone a nuestra disposición para presentar varias páginas en una sola.

<sup>12</sup> Dos métodos diferentes definidos en HTTP, usados para recibir y enviar un archivo u otro recurso.

<sup>13</sup> Es un lenguaje cuya función principal es describir datos y no mostrarlos como es el caso de HTML.

El primer intento fue XML-RPC<sup>14</sup>, sin embargo a pesar de ser novedoso y de brindar varias ventajas, la mayoría de proveedores optaron por su sucesor SOAP (Simple Object Access Protocol). SOAP ayudo a que el concepto de web-services o Servicios Web tome forma definitiva, definiéndose como un protocolo de mensajería entre computadores. A pesar de que su nombre pueda parecer que requiere del uso de determinados objetos, SOAP especifica el formato del mensaje que accede e invoca a los objetos, en vez de especificar los objetos en sí.

“En mayo del año 2000, el W3C (World Wide Consortium) reconoció la propuesta SOAP presentada de forma conjunta por un conglomerado de empresas (Arriba Inc., CommerceOne Inc., Compaq Computer Corp., Microsoft Corp., Development Corp., IBM Corp., Hewlett-Packard Corp., etc). Siendo desarrollado en base a un estándar abierto.” [3]

### **1.2.1.3 Plataformas Alternativas**

CORBA, soporta otras plataformas y lenguajes de programación. Dispone de una amplia cantidad de servicios y se debe a una organización seria e independiente como OMG. Existe complejidad en el desarrollo y su adaptación suele ser lenta, además existen pocas implementaciones libres.

J2EE (ahora JEE), soporta múltiples plataformas y sistemas operativos. Esta avalado por empresas como IBM, Oracle y principalmente Sun Microsystems. Es una plataforma que genera competitividad. Es muy común en implementaciones libres. Su lenguaje primario es Java, y a diferencia de las otras no existe un entorno de desarrollo definido.

.NET, soporta varios lenguajes de programación. Su entorno de programación Visual Studio .NET es muy reconocido, así como su reconocimiento a nivel

---

<sup>14</sup> Es un protocolo de llamada a procedimiento remoto(Remote Procedure Call) que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

comercial. Funciona sobre sistemas operativos Windows y es propiedad exclusiva de Microsoft por lo que es fácil deducir, que casi no existe alguna implementación libre.

### 1.2.2 JEE vs .NET

El debate más acalorado en el mundo de desarrollo de aplicaciones empresariales es el que enfrenta a JEE y a .NET. Ambas tecnologías, de empresas rivales hasta la muerte (Sun Microsystems y Microsoft) pugnan por un mercado cada vez más exclusivo, en el que los competidores van quedando cada vez más rezagados ya que lo que se busca son tecnologías de desarrollo web que ofrezcan un modelo de desarrollo de componentes empresariales. Muchas son las alternativas de desarrollo para una aplicación web: PHP, paginas ASP<sup>15</sup>, ColdFusión, etc. y la mayoría de ellas están en un estado tecnológico excelente, que les permiten solventar la mayoría de los proyectos Web actuales dependiendo del alcance de la aplicación.

La diferencia entre estas tecnologías, es su orientación real hacia un modelo de desarrollo de componentes empresariales. Esto quiere decir que nuestra plataforma tecnológica tiene que servir y propiciar un modelo organizado de componentes que puedan actuar dentro de unos servicios básicos del sistema, desde el entorno de ejecución<sup>16</sup> hasta el mismo IDE<sup>17</sup>,.

Para intentar comparar .NET y JEE deben definirse aspectos concretos en especial que tengan que ver con aplicaciones Web por ejemplo, cabe indicar que .NET nace como una tecnología multipropósito y JEE como un estándar.

---

<sup>15</sup> Active Server Pages, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente.

<sup>16</sup> Es un estado de máquina virtual que suministra servicios de software para procesos o programas mientras que una computadora se está ejecutando

<sup>17</sup> Integrated development environment - entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación.

### 1.2.3 JEE

JEE desde el 2006 es el nuevo nombre para el estándar J2EE, Java 2 Enterprise Edition, arquitectura creada por Sun para el desarrollo de todo tipo de aplicaciones para empresas y usuarios en general facilitando las herramientas necesarias para la programación en el lenguaje Java para el desarrollo, implementación y administración de software empresarial. Sun lo define como un estándar para el desarrollo de aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

Con respecto al desarrollo de aplicaciones web Sun presenta por un lado, la tecnología de Servlets<sup>18</sup> y paginas JSP (Java Server Pages) que permiten abordar una gran gama de desarrollos, desde Intranets hasta portales verticales, con unas ciertas garantías de éxito. Pero por otra parte se tiene JEE, que ofrece al desarrollador el entorno empresarial antes citado. JEE se trata en realidad de un estándar que otras empresas se encargan de implementar. Así cuando hablamos de “servidores de aplicaciones JEE”, no nos referimos a un producto exclusivo y realizado por SUN, sino a un servidor que cumple el estándar JEE y que puede estar desarrollado por otras empresas, como BEA, Netscape o Allaire por citar unos cuantos. Este estándar ya lleva muchos años en el mercado, y ha alcanzado la versión JEE 6 en diciembre de 2009.

#### 1.2.3.1 Características JEE

- “Posibilidad de altas productividades en el desarrollo de las distintas tecnologías J2EE para la integración de aplicaciones corporativas e integración de sistemas existentes.

---

<sup>18</sup> Aplicación *JAVA* que permite la ejecución de un propio servidor web que permite la interactividad del usuario, permitiéndole realizar algunas opciones.

- Mayor escalabilidad al describir básicas transacciones y desarrollando distintos tipos de componentes de aplicación J2EE con modelos flexibles de seguridad.
- Libertad de elección de plataformas de desarrollo y producción, lo que define los aspectos necesarios que puedan solucionar una determinada problemática.
- El desarrollo de aplicaciones web utilizando conceptos de la arquitectura J2EE que permiten la construcción de este tipo de aplicaciones.
- La utilización de herramientas libres que agilizan el desarrollo de software con J2EE y que permiten el funcionamiento en los distintos módulos de ejecución.”<sup>[5]</sup>

### 1.2.3.2 Arquitectura J2EE

J2EE contiene un completo conjunto de APIs<sup>19</sup>, algunas de las cuales están insertas dentro de la arquitectura J2SE<sup>20</sup> que se agregan a la arquitectura. Éstas permiten la ejecución entre las distintas capas.

La figura 1.4 muestra el modelo de ejecución en capas de J2EE que presenta distintas áreas de contenedores (Containers) los cuales contienen API's de funcionamiento, estos contenedores son: Contenedores Web los cuales almacenan los Servlet y páginas JSP que corren del lado del servidor y Contenedores EJB (Enterprise Java Beans) que manejan la ejecución de los Beans<sup>21</sup> de aplicaciones J2EE que también corren del lado del servidor. Por su parte el Contenedor de la aplicación cliente maneja la ejecución de todos los

---

<sup>19</sup> Application programming interface - interfaz de programación de aplicaciones.

<sup>20</sup> Java 2 Stándar Edition- Edición Estándar para el desarrollo de aplicaciones Java.

<sup>21</sup> También llamados JavaBeans, son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java, encapsulan varios objetos en un único objeto para hacer uso de un sólo objeto en lugar de varios más simples.

componentes de servidor y a su vez el Contenedor Applet<sup>22</sup> la ejecución de dichos applets en el cliente. Se puede interactuar con los datos desde cualquiera de estos contenedores.

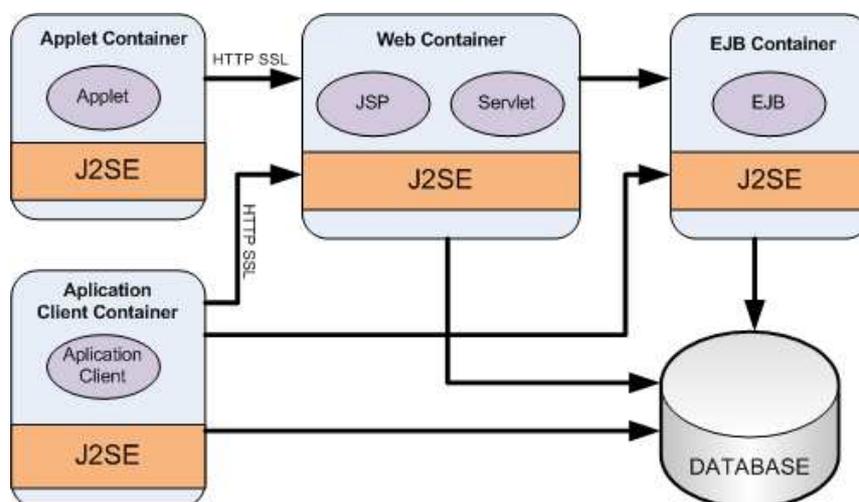


Figura 1.4 Arquitectura JEE <sup>[3]</sup>

### 1.2.3.3 Paquetes de aplicaciones en JEE

Son encapsulaciones de aplicaciones J2EE con distinta orientación, definidas en tipos de extensiones de archivos.

- “JAR (Java archive): es un formato de archivo independiente de la plataforma que permite que varios archivos puedan ser encapsulados dentro de uno, permitiendo que éste pueda ser una aplicación completa de fácil movilidad y ejecución.
- WAR (Web Application archive): este componente es un archivo que contiene un archivo JAR que posee uno o más módulos WEB. Pudiendo ser desde un sistema JSP a un servicio WEB.

<sup>22</sup>Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

- EAR (Enterprise Archive File): un archivo EAR es un archivo JAR que contiene una aplicación J2EE. Utilizando archivos EAR es posible unir un número de diferentes aplicaciones J2EE usando los mismos componentes sin necesidad de códigos extra solo incluyéndolos dentro de este módulo.
- RAR (Resource Adapter Archive): un archivo RAR es un archivo JAR que contiene un adaptador de recursos que puede ser desplegado en un servidor J2EE. Es similar a un archivo EAR e incluso un RAR puede estar contenido dentro de este.”<sup>[5]</sup>

#### 1.2.3.4 Evolución de JEE

En la tabla 1.1 se muestra la forma en cómo ha ido evolucionando el estándar que actualmente se lo conoce como JEE y que en un principio fue JPE Y J2EE.

<b>Estándar</b>	<b>Versión del Estándar</b>	<b>Fecha de Publicación</b>
JPE	Java Professional Edition	Mayo 1998
J2EE	Versión 1.2	Diciembre 1999
J2EE	Versión 1.3	Septiembre 2001
J2EE	Versión 1.4	Noviembre 2003
JEE	Versión 5	Mayo 2006
JEE	Versión 6	Diciembre 2009

**Tabla 1.1 Evolución del estándar J2EE**

#### 1.2.4 .NET

Antes de .NET Microsoft ofrecía su plataforma Windows DNA<sup>23</sup> basada en una mezcla de tecnologías. Con el servidor Web Internet Information Server 4.0 (IIS 4.0) como nexo común, el servidor de base de datos SQL Server, tecnología de

<sup>23</sup> Windows Distributed interNet Applications Architecture – Arquitectura de aplicaciones distribuidas en internet de Windows.

objetos COM, el servidor transaccional MTS<sup>24</sup> y MSMQ<sup>25</sup> y las páginas ASP para el desarrollo de aplicaciones Web empresariales. Pero este modelo ofrecía algunas desventajas. Por un lado, la comunicación y utilización entre componentes se realizaba a través de una interfaz binaria definida por el estándar COM. Así, el desarrollador podía elegir entre diversos lenguajes de desarrollo como Visual Basic, Visual C++, o diversas librerías como ATL o MFC<sup>26</sup>. Esto, que en un principio puede parecer una ventaja en la práctica fue un gran problema. Los desarrollos de componentes en Visual Basic escalaban mal por su pobre soporte a tecnologías multi-hilo, y los desarrolladores de Visual C++ se encontraban con una tecnología que sí escalaba muy bien, pero que como contrapartida resultaba compleja y resultaba muy sencillo cometer errores de pérdida de recursos y memoria. Más tarde había que integrar estos objetos con las páginas activas ASP. El modelo de desarrollo funcional de estas páginas hacia que la mayor parte de los proyectos desembocase en una amalgama de páginas con “spaguetti-code”<sup>27</sup> y con una fuente interminable de errores.

De aquí nace la oferta llamada .NET, que es un proyecto de Microsoft para el desarrollo de software con énfasis en transparencia de redes con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones. Basado en esto, la empresa desarrolla una estrategia horizontal integrando sus productos, desde el sistema operativo hasta las herramientas de mercado. .NET soporta la programación con varios lenguajes, a diferencia de JEE que tiene a Java como único lenguaje.

En la tabla 1.2 nombramos 24 lenguajes que soporta .NET.

---

<sup>24</sup> Microsoft Transaction Server, herramienta para controlar y optimizar el rendimiento de aplicaciones de servidor.

<sup>25</sup> Microsoft Message Queuing, permite que aplicaciones que se ejecutan en momentos distintos puedan comunicarse a través de redes y sistemas heterogéneos.

<sup>26</sup> Active Template Library y Microsoft Foundation Classes, librerías basadas en templates y jerarquías de clases respectivamente.

<sup>27</sup> Es el nombre peyorativo, que se le da a código fuente que se caracteriza por su extrema complejidad y por tener una estructura desordenada.

Lenguajes soportados en .NET			
APL	Eifell	Mercury	Python
C++	Forth	Mondrian	RPG
C#	Fortran	Oberon	Scheme
COBOL	Haskell	Oz	Small Talk
Componnet Pascal	Java	Pascal	Standar ML
Curriculum	Microsoft Jscript	Perl	Microsoft Visual Basic

Tabla 1.2 Lenguajes soportados por la plataforma .NET

#### 1.2.4.1 Objetivos de la plataforma .NET

- Liberar al programador de temas como la infraestructura (aspectos no funcionales) Así, el Framework .NET se encarga de gestionar automáticamente funciones como la gestión de la memoria, de los hilos o de los objetos remotos.
- Proporcionar integración entre diferentes lenguajes. Con el auge de los sistemas distribuidos, la interoperabilidad se ha convertido en una de las principales metas a conseguir de los desarrolladores de sistemas. El problema de la interoperabilidad ha sido considerado durante muchos años, desarrollándose varios estándares y arquitecturas con diferente nivel de éxito.
- Proporcionar una ejecución multiplataforma. .NET ha sido diseñado para ser independiente de la plataforma sobre la cual se ejecutaran las aplicaciones, si bien actualmente solo está disponible el Framework .NET para las distintas plataformas Windows, hay proyectos independientes que buscan la utilización de .net en otras plataformas que no sean Windows.
- Sistema de despliegue simple. Se ha eliminado la necesidad de tratar con el registro, con GUID<sup>28</sup>s, de forma que la instalación de una aplicación es tan sencilla como copiar un directorio.
- Mejora de la escalabilidad. La gestión por parte del sistema de ejecución de .NET de recursos como la memoria, permite mejorar la escalabilidad. Se

<sup>28</sup> Son números pseudoaleatorios empleados en aplicaciones de software, En el COM de Microsoft, tan sólo se usan para distinguir interfaces de componentes software diferentes.

proporciona soporte para arquitecturas fuertemente acopladas y débilmente acopladas. Una arquitectura es fuertemente acoplada si los componentes de cada capa tienen alta dependencia entre sí y es débil si los componentes pueden ser utilizados por separado. Se consigue un buen rendimiento, escalabilidad y confiabilidad en grandes sistemas distribuidos, con operaciones en las cuales los componentes están fuertemente acoplados. Sin embargo, también debe soportarse una comunicación débilmente acoplada orientada a mensajes, de forma que una transacción no quede interrumpida o bloqueada por cualquier dependencia en tiempo de ejecución.

- Proporcionar un mecanismo de errores consistente. En .NET todos los errores son manejados mediante un mecanismo de excepciones que consiste en propagar excepciones (divisiones por cero, lectura de archivos no disponibles, etc.), entre distintos módulos, permitiendo aislar el manejo de errores del resto.

#### **1.2.4.2 Arquitectura Empresarial .NET**

.NET presenta distintos niveles de aplicaciones y componentes ofreciendo distintas capacidades y servicios apoyándose en los sistemas operativos Windows, y sistemas compatibles. De esta forma puede presentar a los usuarios incluidos dentro del nivel superior de la arquitectura los siguientes aspectos divididos en 3 capas:

1.- Capa presentación: formada por los componentes de la interfaz de usuario (IU), y sus componentes de proceso. Los componentes de IU son los cuales interactúan con el usuario (por ejemplo ventanas o páginas Web). Los componentes de proceso de IU son los distintos tipos de clases del tipo controladora en UML, los cuales encapsulan la lógica de navegación y control de eventos de la interface.

2.- Capa de Negocios: es la capa o nivel que encapsula la lógica de negocios. Las entidades empresariales representan objetos que van a ser manejados por la aplicación (modelo de objetos, XML, data sets con tipo, estructura de datos), las cuales permiten representar objetos que han sido identificados durante el modelado. Los componentes empresariales contienen lógica de negocios y en algunos casos pueden ser los objetos raíz que inician las transacciones.

3.- Capa de Acceso a datos: capa que contiene las clases que interactúan con las bases de datos. Estas surgen con la necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y capas. Aquí se encuentra la clase de métodos estáticos que permite uniformizar las operaciones de acceso a datos a través de un único conjunto de métodos.

En la figura 1.5 se muestra la arquitectura empresarial .NET, sus capas y sus componentes.

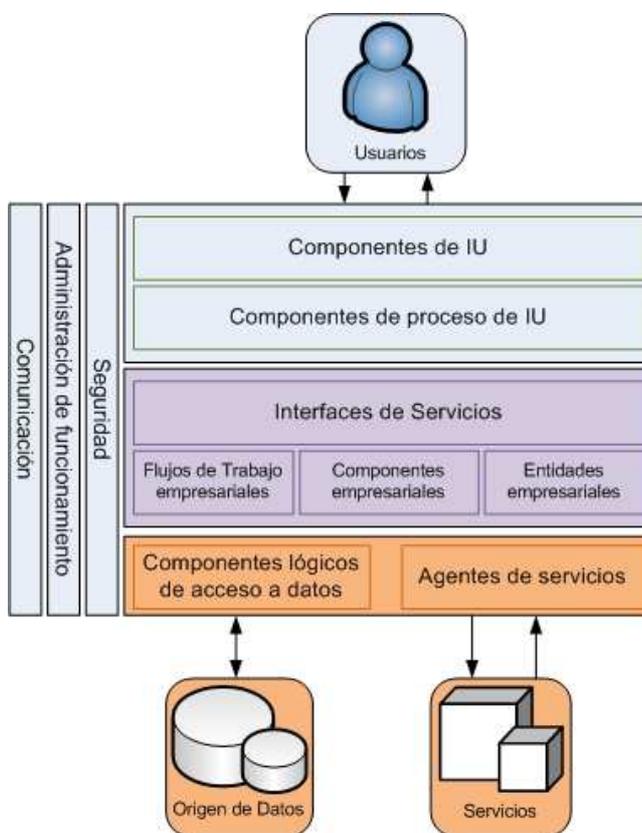


Figura 1.5 Arquitectura .NET <sup>[5]</sup>

### 1.2.4.3 Evolución de .NET

En la tabla 1.3 se muestra la evolución de la plataforma .NET y su respectivo framework de trabajo.

<b>Fecha de liberación</b>	<b>Plataforma</b>	<b>Framework</b>
2002-02-13	Visual Studio .NET	Framework .NET
2003-04-24	Visual Studio .NET 2003	Framework .NET 1.1
2005-11-07	Visual Studio 2005	Framework .NET 2.0
2007-11-19	Visual Studio 2008	Framework .NET 3.0 y 3.5
2010-04-02	Visual Studio 2010	Framework .NET 4.0

**Tabla 1.3 Evolución de .NET**

## 1.3 METODOLOGÍAS DE DESARROLLO DE APLICACIONES

Consisten en un conjunto de pasos y procedimientos que deben seguirse para el desarrollo de software. Éstas tienen un enfoque disciplinado y sistemático. Las metodologías ayudan a optimizar el desarrollo, además estandarizan y mejoran el producto final.

Una metodología debería definir cómo dividir el desarrollo en etapas, qué tareas realizar en cada etapa, qué resultados se producirían y cuándo, que requerimientos y restricciones se aplican, que herramientas se utilizan y por último como gestionar y controlar el software. Todo enfocado a lograr objetivos como, definir requisitos de forma acertada, sistematizar el desarrollo, ahorrar tiempo y dinero identificando los cambios a realizar, obtener aplicaciones documentadas y fáciles de mantener.

### 1.3.1 CLASIFICACIÓN DE LAS METODOLOGÍAS

A lo largo de los años se han propuesto numerosas metodologías. Algunas han sido producto de la investigación de autores del ámbito académico otras por autores del ámbito empresarial de desarrollo del software y otras por administraciones públicas.

Algunas metodologías intentan describirlo todo al detalle. Otras son más genéricas. Algunas hacen hincapié en los datos, otras en los usuarios, otras en las tareas, otras en la documentación o en cualquier aspecto o combinación de aspectos que puedan darse en el desarrollo del software.

### 1.3.2 METODOLOGÍAS TRADICIONALES

“Hay una serie de metodologías que solemos llamar **tradicionales** propuestas casi todas ellas con anterioridad a los años 90 que pretendían ayudar a los profesionales indicando pautas para realizar y documentar cada una de las tareas del desarrollo del software. Sin embargo, tienen casi todas ellas un gran problema: asumen que un proyecto informático es casi una extensión de un proyecto burocrático tradicional. Así pues, los pasos que sugieren para llevar a cabo cada tarea, aunque bien intencionados, están cargados de burocracia, reiteraciones, ambigüedades. No suelen tener en cuenta cosas como la calidad, la satisfacción, la competitividad, los beneficios. Fueron metodologías creadas en los años 70-80 pensando en los negocios de los años 50.”<sup>[5]</sup>

La mayor parte de las metodologías tradicionales ya no funcionan. Están obsoletas desde casi todos los puntos de vista. Sólo algunas metodologías tradicionales han sido revisadas y adaptadas y su funcionalidad suele estar limitada a proyectos no muy innovadores.

Entre las **metodologías tradicionales** podemos citar:

- Desarrollo de sistemas de Jackson (JSD). De los años 80.
- Ingeniería de la información. De los 80 también
- Structured System Analysis and Design Method (SSADM). También de los 80. Muy popular en Europa, ya que tiene su origen el Reino Unido.

Algunas, como las dos primeras (Jackson, Ingeniería de la información), tienen un interés principalmente histórico. Otras, como SSADM, tienen cierta vigencia, en especial en lo que concierne a proyectos públicos.

### 1.3.3 METODOLOGÍAS ÁGILES

Las metodologías surgidas desde los años 90 hasta la actualidad, parten de otra mentalidad, adoptando el concepto de agilidad. Siendo conscientes de lo cambiante y amplio que es el mundo del software, una metodología debe ser lo suficientemente precisa como para que todo el mundo la pueda seguir y sea de utilidad como pauta común, pero también debe ser lo suficientemente adaptable como para poder aplicarse en distintos proyectos. Lo suficientemente sencilla como para que no resulte muy compleja su utilización, pero lo suficientemente completa como para que la utilización por parte del equipo sea provechosa. Por lo tanto aunque el término de agilidad es discutible, es evidente la importancia que tienen las metodologías "modernas" que se basan en esta nueva mentalidad.

Entre las **metodologías modernas** -unas más, otras menos- se puede destacar:

- Rapid Application Development (Desarrollo rápido de aplicaciones - RAD).
- Scrum
- Extreme programming. (Programación extrema - XP)
- Rational Unified Process. (Proceso Racional Unificado - RUP)
- Agile Unified Process. (Proceso Ágil Unificado - AUP)

Hay muchas más, pero quizá éstas sean las más populares en el momento de escribir estas líneas. Personalmente recomendaría un cierto conocimiento general de al menos las cuatro últimas (Scrum, XP, RUP y AUP)

## **CAPÍTULO 2. REQUERIMIENTOS DE LA APLICACIÓN. ESTÁNDARES APLICABLES A HELPDESK EN SISTEMAS WEB**

### **2.2 ANTECEDENTES DEL PROBLEMA**

A continuación detallaremos el propósito del desarrollo de nuestro software, basándonos en los antecedentes que impulsaron a desarrollarlo.

Una empresa que comercializa software siempre deberá preocuparse en el cómo evitar la falsificación y plagio del mismo, ya que esto causaría malestar en el usuario final, malos entendidos dentro de la empresa y consecuentemente pérdida de dinero.

Por otro lado cuando un cliente adquiere un software, lo primero que desearía es conocer el manejo completo en el más corto tiempo posible, lamentablemente cuando hablamos de software de tipo empresarial, la empresa dueña del programa no suele destinar suficiente tiempo y recursos para una completa capacitación que ayude a afrontar todas las funcionalidades y problemas presentes en cualquier aplicación de este tipo.

#### **2.2.1 SITUACIÓN ACTUAL DEL PROCEDIMIENTO DE ACTIVACIÓN DE LICENCIAS DE SISTEMAS EN LA CORPORACION MILENIOSYSTEM S.A.**

En la actualidad el proceso de activación de una licencia de cualquiera de los sistemas desarrollados por la empresa consiste en lo siguiente.

Una vez adquirido el software, es decir firmado el contrato entre las partes, se debe especificar claramente en cuantos y en cuales computadores se procede a instalar la aplicación.

El programa instalador se ejecuta en cada uno de los computadores como si fuera cualquier programa para Windows.

Una vez terminada la instalación se accede por primera vez al sistema y aparece un mensaje que indica que el sistema está por primera vez instalado y para funcionar necesita una licencia.

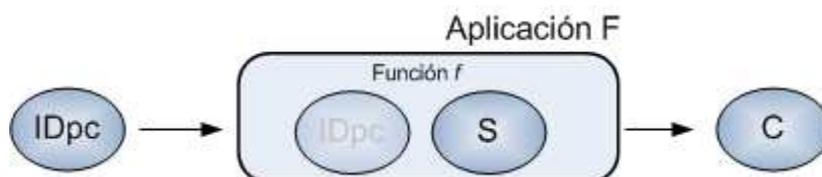
En ese momento empieza el proceso de activación de la licencia basado en este esquema.

Paso1: Se ejecuta el programa instalador en un computador con S.O Windows, cuando el sistema se abre por primera vez, obtiene el IDpc y solicita la clave de activación. Ver Figura 2.1.



**Figura 2.1 Activación de licencia, paso1**

Paso2: El personal autorizado de la empresa tiene acceso a una aplicación que calcula la clave de activación a partir de la IDpc. En la figura 2.2. se muestra una diagrama de flujo de información de la aplicación F.



**Figura 2.2 Activación de licencia, paso2**

Paso3: Ingresar la clave de activación en el sistema finalizando la activación. Ver figura 2.3.



**Figura 2.3 Activación de licencia, paso3**

- IDpc: número que identifica a un computador. Cualquiera que fuese este código es único y nunca se repetirá ya que es el resultado de aplicar una función al código de la placa del procesador del computador.
- S: código secreto utilizado por la empresa, diferente para cada tipo de sistema comercializado.
- $f(S, IDpc)$ : función  $q$  utiliza un parámetro constante (S) y uno variable (IDpc) para generar la clave.
- C: clave de activación de licencia para el computador, por las características de la función  $f(S, IDpc)$ , esta clave puede ser la misma en diferentes computadores. Pero la probabilidad de que una clave sirva para diferentes computadoras dentro de una misma organización es muy pequeña.
- F: aplicación desarrollada por la empresa que arroja la clave C, basándose en la función  $f(S, IDpc)$ .

El problema surge si nos ponemos a pensar ¿que pasaría si el programa F, cae en manos equivocadas? Primero se obligaría a cambiar el código S para todos los sistemas desarrollados, y luego desechar todos los discos instaladores que tengan el instalador creado con el código S antiguo.

Adicionalmente se debe llevar un control riguroso de las claves otorgadas, para poder tener un registro de los clientes y usuarios del sistema.

### **2.2.2 SITUACIÓN ACTUAL DEL PROCEDIMIENTO DE AYUDA A CLIENTES EN LA CORPORACION MILENIOSYSTEM S.A.**

Normalmente los sistemas desarrollados por empresas independientes, tratan en lo posible de garantizar la facilidad de uso. A pesar del esfuerzo de los desarrolladores por brindar comodidad al usuario a la hora de manejar los diferentes módulos, existen procesos que por el diseño y funcionalidad del sistema, resultan complejos de aprender a manejarlos. Como una de las varias consecuencias de este tipo de inconvenientes, es la generación de errores que a la postre representan una pérdida de tiempo tanto como para el usuario como para las personas encargadas de corregir el error.

Por un lado se cuenta con manuales, tanto escritos como electrónicos en donde los usuarios se podrían guiar para no cometer u corregir errores, pero lastimosamente no existe una cultura que acostumbre a las personas a intentar solucionar ellos mismo los problemas por mas difíciles que parezcan.

La ayuda telefónica, aparece como una forma práctica de resolver inquietudes, desde pequeñas y simples hasta grandes y complejas. Tomando en cuenta que debe existir personal capacitado disponible para contestar el teléfono a cualquier hora.

El soporte remoto vía software, es una forma aun mas practica de resolver problemas que necesitan pronta solución, pero este procedimiento de ayuda depende de factores como, la conexión de internet del cliente y la del técnico que brinda el soporte.

Al final, la solución definitiva y que deja más conforme al cliente es la visita personal de un técnico de la empresa que resuelva el problema, por pequeño que sea. Ya que en la mayoría de los casos, los problemas se solucionan con pocas instrucciones, la visita de un técnico representa un desperdicio de recursos para la empresa.

## 2.3 SOLUCIÓN PROPUESTA

Se propone desarrollar una aplicación web, que permita realizar el registro de nuevos clientes y distribuir de forma rápida y segura las claves de activación de las licencias. Además proporcionara información de ayuda para facilitar la resolución de problemas por parte del mismo usuario.

Como primer paso seleccionamos la plataforma sobre el cual desarrollar la aplicación, para luego definir los requerimientos y relacionarlos con las características y herramientas que nos provee la plataforma seleccionada.

### 2.3.1 ELECCIÓN DE LA PLATAFORMA DE DESARROLLO

En la tabla 2.1 se exponen las diferencias entre las 2 opciones de plataforma de desarrollo.

	.NET	JEE
Sintaxis de lenguaje	VB.NET es lo mejor que tiene .NET	Java es el lenguaje con mejor sintaxis
Portabilidad	Solo entre sistemas Windows	entre diferentes sistemas operativos
Aprendizaje	Bastante Sencillo	Costoso de aprender
Velocidad de Desarrollo	Mas rápido debido a la cantidad de componentes que facilitan el trabajo	Más lento
Plataforma	Solo Windows	Cualquier plataforma
Bases de datos	Optimizada para Microsoft SQL	Buena compatibilidad con MYSQL, Postgres, Oracle
Entornos de Desarrollo	Visual Studio con un coste elevado	Varios IDEs, la mayoría gratuitos
Seguridad	Mala fama debido a fallos de Windows	Más seguro que .NET
Orientación a Objetos	Soportado	Soportado
Rendimiento	Alto	Alto
Servidor Web	IIS	Varios servidores comerciales y opensource
Librerías y Frameworks	Gran cantidad	Gran cantidad, la mayoría gratuitas y opensource

Soporte y Comunidad	Foros y comunidades la mayoría manejados por Microsoft	Foros y comunidades manejado por varios grupos independientes
Coste	Licencias bastante caras	Se puede desarrollar con herramientas gratuitas y de pago

**Tabla 2.1 Comparativa .NET vs JEE**

Por los criterios expuestos y su característica de software abierto, se opta por utilizar la tecnología JEE, que nos permitirá desarrollar una aplicación distribuida capaz de satisfacer los requerimientos de rendimiento, seguridad y escalabilidad de grandes organizaciones, mediante tecnologías diseñadas para trabajar con un modelo lógico de 3 capas, separando la lógica de negocio de los datos y la interfaz de usuario.

## **2.3.2 REQUERIMIENTOS DE LA APLICACIÓN Y SELECCIÓN DE HERRAMIENTAS DE DESARROLLO**

### **2.3.2.1 Robustez y Seguridad**

La plataforma J2EE aprovecha la robustez del lenguaje de programación Java que permite a los desarrolladores escribir el código una sola vez y ejecutar la aplicación en cualquier plataforma. Además brinda seguridad a pesar de la rapidez del desarrollo y despliegue de la aplicación, gracias a la arquitectura en capas sobre la cual trabaja, que ayuda a independizar los procesos que se realizan desde la petición hasta la respuesta.

Para que un servidor de aplicaciones devuelva objetos desde el lado del servidor, JEE utiliza por ejemplo, los componentes EJB, que sirven para separar la lógica de negocio de los datos, El modulo EJB es el responsable de la lógica de negocio y el acceso a la capa de datos. En la arquitectura JEE el contenedor EJB es el único que tiene acceso a la capa de datos, en este contenedor se despliega el modulo EJB.

### 2.3.2.2 Interfaz web flexible

Por otro lado el contenedor web atiende las peticiones http y es donde se despliega el modulo web. Para el desarrollo del modulo web comúnmente se usa un patrón de diseño llamado Model View Controller(MVC)<sup>29</sup>. Bajo este MVC existen diferentes frameworks de interfaces de usuario del lado del servidor para aplicaciones web, que pueden agilizar el desarrollo. En la tabla 2.2 se listan los más importantes frameworks de interfaz de usuario que están trabajando en conjunto con JEE.

<b>Empresa dueña de la Licencia</b>	<b>Nombre</b>
Apache	Grails
LGPL	Framework Dinamica
Apache	Struts
Apache	Beehive
Apache	Spring
Apache	Tapestry
Apache	Aurora
Apache	Java Server Faces

**Tabla 2.2 Frameworks MVC para JEE**

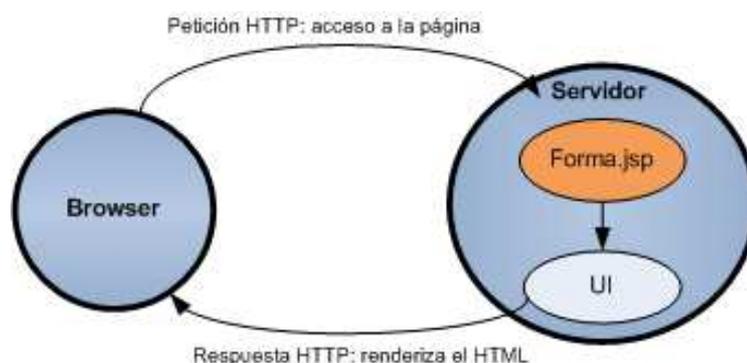
Para nuestro desarrollo seleccionamos Java Server Faces (JSF) un framework que en la actualidad brinda buenos resultados y sobresale de sus competidores, gracias a las siguientes características:

- Estabilidad y decremento del código duplicado.
- Gracias al framework ligero Facelets permite la construcción de interfaces basadas en plantillas, brinda una rápida creación de: componentes por composición, funciones y librerías de componentes.
- Implementa fácilmente librerías como ajax y richfaces.
- Facilita la administración de su respectivo Back-Bean<sup>30</sup>.

<sup>29</sup> Model View Controller (Modelo Vista-Controlador): Es un patrón o estilo de diseño de software que separa los datos, la interfaz de usuario y la lógica de negocio, en tres componentes distintos.

<sup>30</sup> Back-Bean, componente del modelo vista controlador conocido como el gestor o controlador.

Esta tecnología consta de un API (Interfaz de programación de aplicaciones), una implementación de referencia para representar componentes UI (interfaz de usuario) y manejar su estado, y además posee una librería de etiquetas JSP (Java Server pages) personalizadas para dibujar componentes UI dentro de una páginas JSP. El interface de usuario (UI) creado con JSF se ejecuta en el servidor y se renderizan (se generan los componentes) en el cliente. En la figura 2.4 se muestra la interacción entre el cliente y el servidor con JSF.



**Figura 2.4 Proceso JSF**

Java Server Faces ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones Web construidas con tecnología JSP consiguen parcialmente esta separación. Sin embargo, una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos de componentes o manejar elementos UI como objetos con estado en el servidor. La tecnología Java Server Faces nos permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitectura UI del lado del cliente.

### **2.3.2.3 Rendimiento**

El rendimiento de una aplicación empresarial se lo puede evaluar de varias formas pero en general el rendimiento se relaciona con la forma en cómo responden los diversos componentes de la aplicación con el aumento de carga de

información, usuarios simultáneos, número de transacciones, etc. La respuesta al aumento de estos parámetros depende principalmente de cómo se gestione la memoria de trabajo que dispone la aplicación. La herramienta que realiza esta labor en nuestra arquitectura, es el servidor de aplicaciones.

Debido al gran número de usuarios concurrentes que soportan las aplicaciones empresariales, los recursos ilimitados del servidor (bases de datos, listas de mensajes, directorios, etc.) deben ser gestionados óptimamente. Cada uno de esos recursos es accedido por las aplicaciones usando un objeto conexión<sup>31</sup> que es el punto de entrada. El servidor de aplicaciones se encarga de gestionar el acceso a esos recursos compartidos para alcanzar los requerimientos de alto rendimiento de las aplicaciones empresariales. En la tabla 2.3 se lista los más importantes servidores de aplicaciones que son compatibles con JEE.

<b>Desarrollador</b>	<b>Nombre</b>
IBM	Websphere
Oracle	WebLogic
Sybasic Inc.	EAServer
Object web	JOnAS
LGPL(Red Hat)	JBoss
Sun(Oracle)	Glassfish

**Tabla 2.3 Servidores de Aplicaciones para JEE**

Las opciones más reconocidas son Glassfish y JBoss. No hay un motivo fuerte para elegir una de las 2 ya que son las que actualmente compiten el liderazgo entre los servidores de aplicaciones. Para nuestra aplicación seleccionamos Glassfish v3, por ser un poco más fácil de administrar. En el servidor de aplicaciones Glassfish tenemos la opción de crear un pool de conexiones. “Connection Pool” es una técnica para permitir a múltiples clientes compartir un conjunto de objetos conexión almacenados en la cache del servidor, que provee el acceso a una base de datos.

---

<sup>31</sup> Objeto Conexión: se trata de un objeto con estado y comportamiento, que en este caso establece una conexión entre el cliente y el servidor.

Las ventajas de esta técnica de gestión de recursos se ven si nos referimos a la técnica tradicional donde se atacaba a la base de datos directamente sin hacer pooling, creando y destruyendo una conexión por cada petición del cliente. Esto impacta en el rendimiento de la capa EJB debido al sobre esfuerzo de crear y destruir objetos conexión. El proceso del servidor de aplicaciones que establece conexión con la base de datos dedica entre 1 y 3 segundos entre la comunicación con el servidor, la autenticación y otros pasos intermedios. Y esto por cada petición de cliente. En cambio con un pool de conexiones permitimos que múltiples conexiones sean compartidas de forma transparente entre diferentes peticiones de clientes.

Ya no existe sobre esfuerzo en el acceso a la base de datos ya que el gestor del pool crea un conjunto de conexiones (objetos) al arrancar el servidor, son utilizadas indefinidamente hasta que se apaga el servidor. El número de objetos del pool puede crecer dinámicamente según la carga de trabajo en un momento puntual. Los clientes obtienen una conexión disponible del pool cuando lo necesitan y cuando terminan lo devuelven para que pueda ser reusado. Por ejemplo creando un recurso JDBC (Java Database Connectivity)<sup>32</sup> que apunte al pool de conexiones, ponemos a disposición los servicios del pool de conexiones a las aplicaciones con acceso al servidor local o remoto,<sup>[7]</sup>

#### **2.3.2.4 Gestión de los datos**

##### 2.3.2.4.1 Gestor de bases de datos

Los datos manejados por el sistema, en algún momento acabaran siendo almacenados en bases de datos para su posterior recuperación y análisis. Las bases de datos relacionales almacenan los datos en tablas formadas por filas y columnas. Cada registro usa como identificador único, claves primarias. Las relaciones entre tablas se representan con claves foráneas, respetando

---

<sup>32</sup> JDBC (Java DataBase Connectivity): es un API que se presenta como una colección de interfaces java y métodos de gestión de conexiones hacia un modelo específico de bases de datos.

restricciones de integridad. En la tabla 2.4 nombramos a los gestores más utilizados en los desarrollo JEE.

Licencia	Nombre
Oracle	Oracle
Apache	Derby
GPL(Sun)	MySQL
BSD(PGDG)	PostgreSQL

**Tabla 2.4 Gestores de Bases de Datos compatibles con JEE**

Entre los gestores de la lista, sobresalen PostgreSQL y MySQL por su característica de uso libre y buen desempeño. Estos 2 gestores tienen sus ventajas y desventajas según varios criterios:

- MySQL proporciona mejores herramientas de administración y más documentación.
- PostgreSQL garantiza la integridad de los datos mucho más que MySQL.
- PostgreSQL presenta mejor escalabilidad y rendimiento bajo grandes cargas de trabajo.
- MySQL fue diseñado pensando en la velocidad, por ello se suprimieron algunas características de los demás Gestores de Base de Datos, como las transacciones y las sub consultas consumiendo pocos recursos.
- MySQL tiene su tamaño de los registros ilimitado y mejor control de acceso de usuarios.

Para nuestro desarrollo seleccionamos MYSQL, por su rapidez y porque nuestra aplicación no maneja grandes cargas de información.

Toda la terminología de claves y relaciones es propia del modelo relacional, sin embargo es totalmente desconocida en el modelo orientado a objetos. En el modelo orientado a objetos manejamos clases y sus instancias denominados objetos. Los objetos encapsulan su estado y comportamiento mientras corre la

maquina virtual de java (JVM)<sup>33</sup>. Si detenemos la JVM se pierde todos los datos de negocio que manejábamos en aquel momento., para evitarlo necesitamos mecanismos de persistencia.<sup>[9]</sup>

#### 2.3.2.4.2 Proveedor de Persistencia

Persistencia es el proceso que permite almacenar el estado de un objeto de forma permanente en disco duro. Entonces el reto consiste en trasladar datos del modelo orientado a objetos a uno relacional.

En java existen 3 métodos:

- Serialización: convierte los objetos en una serie de bits almacenables en disco, pero no dispone de un lenguaje de consulta y no soporta altas tasas de acceso concurrente.
- API JDBC: para acceder a diferentes tipos de gestores de bases de datos relacionales. Aunque su uso está muy difundido la tendencia es migrar a herramientas más potentes.
- Herramientas de mapeo objeto relacional (ORM): se basan en delegar el acceso a las bases de datos relacionales a un framework. Java define el modelo de programación para la persistencia de entidades pero deja su implementación a proveedores externos.<sup>[6]</sup>

En la tabla 2.5 se muestran los proveedores de persistencia que hacen un mapeo objeto relacional.

<b>Desarrollador</b>	<b>Nombre</b>
Eclipse	Eclipse-Link
LGPL(Red hat)	Hibernate
Oracle	Top-Link
Apache	Open JPA

**Tabla 2.5 Proveedores de Persistencia tipo ORM**

<sup>33</sup> JVM (Maquina Virtual de java): se ejecuta en una plataforma específica y permite interpretar y ejecutar instrucciones expresadas en Javabytecode.

Java define un modelo de programación de persistencia en el cual nosotros solo debemos proporcionarle al motor de persistencia una clase entidad por cada tabla de la base de datos. Una entidad implica objetos que contienen información de mapeo y que viven a corto plazo en memoria y persistentemente en la base de datos. JPA dispone de una interfaz llamada java entity manager con una serie de métodos para que podamos operar con el conjunto de entidades. El conjunto de entidades asociadas con la base de datos se lo llama contexto de persistencia. De tal forma que la aplicación trabajara directamente con este contexto de persistencia, y cualquier cambio afectara a la base de datos subyacente de forma transparente.

Seleccionamos Hibernate que tiene como ventajas: la independencia del sistema gestor de base de datos a utilizar, el lenguaje para realizar Querys permite que no haya que cambiar nada de código para cambiar de Sistema Gestor de Base de Datos. Solo cambiar un parámetro en el fichero de configuración de Hibernate. Además implementa una especie de caché, guardando datos en su sesión. Esto permite liberar la carga a la base de datos y permitir así multitud de consultas simultáneas sin colapsar la base de datos.<sup>[10]</sup>

### **2.3.2.5 Portabilidad y escalabilidad**

Algo de sumo interés para las empresas, es la portabilidad y la escalabilidad de sus aplicaciones empresariales. Portabilidad tiene que ver con la dependencia de una aplicación a correr en una plataforma, mientras menos dependa de la plataforma es más portable. Así que el desarrollo de una aplicación es viable sólo cuando es fácilmente portable a través de plataformas y dispositivos. Escalabilidad es la característica de una aplicación, de ampliar su alcance sin afectar su calidad inicial. Por lo tanto la escalabilidad es un parámetro, que decide la viabilidad a largo plazo de una aplicación. Las aplicaciones empresariales deben ser escalables a fin de que se pueden ampliar para dar cabida a miles de usuarios simultáneamente.

Estas características dependen de la forma en cómo se organice el desarrollo mediante el entorno integrado de desarrollo (IDE). El IDE también brinda las herramientas que podría facilitar la ampliación de los recursos a utilizar en la aplicación.

Entre los IDEs más reconocidos tenemos a Netbeans y Eclipse.

- Netbeans tiene una interfaz muy amigable e intuitiva.
- Netbeans integra varios módulos de JEE sin necesidad de plugins adicionales
- Eclipse ofrece visualización de la vista en tiempo de desarrollo.
- Netbeans tiene mejor manejo de errores.
- Eclipse tiene soporte para JBoss y Glassfish

Elegimos el IDE Netbeans 6.8

### **2.3.2.6 Presentación de la información.**

Este requerimiento a diferencia de los anteriores no tiene que ver con la tecnología ni con la forma en cómo se desarrolla la aplicación. Tiene que ver con la forma en cómo la información que se presenta influirá en la relación Cliente – Tecnología de Información. Es decir, la aplicación debe mostrar los datos de manera que garantice la correcta gestión del servicio que provee.

Para esto, a la presentación de la información, se le dará un trato especial siempre intentando familiarizar poco a poco al usuario con la terminología típica del tratamiento de la información siguiendo estándares. Para esto necesitamos conocer en detalle la terminología definida en los documentos ITIL<sup>34</sup>, todo con el fin de mejorar la gestión de la ayuda que presentará nuestra aplicación web.

---

<sup>34</sup> ITIL La Biblioteca de Infraestructura de Tecnologías de Información.

## **2.4 PRACTICAS PARA GESTIONAR Y OPTIMIZAR LA INFORMACION DE AYUDA QUE MUESTRA UNA APLICACIÓN WEB.**

En la actualidad la dependencia por las TI<sup>35</sup> es un factor crítico en el desarrollo de las organizaciones, ya que cada vez son más las empresas que amplían sus expectativas de negocio y menos las relaciones que se manejan entre las tecnologías, los procesos internos, el recurso humano y las estrategias de negocio; el depender de las TI solamente podrá ser vista como positiva siempre y cuando exista una manera de operar que permita aprovechar las TI y las convierta en una ventaja que aporte funcionalidad y flexibilidad institucional. Para lograr lo anteriormente dicho, es indispensable que existan estándares internacionales que orienten a las organizaciones respecto a la forma cómo es posible organizar y estructurar de la mejor manera y a los mejores costos, todos los servicios de TI que giran en torno a la organización, además de lograr que se comuniquen los principales actores que intervienen en el desarrollo de la estrategia del negocio.

En el caso de la Corporación MilenioSystem S:A. el objetivo es implantar paulatinamente software empresarial que demuestre beneficios para la empresa y que oriente y organice de mejor manera las problemáticas del negocio gracias a la solución que el software brinde.

### **2.4.1 ANTECEDENTES PARA LA CREACION DE ESTANDARES DE ADMINISTRACION DE LAS TI.**

Hace algunas décadas la necesidad de adoptar estándares propios para gestionar la información, en muchos casos causo que se duplicaran esfuerzos al interior de los proyectos y se desperdicien recursos dentro del gobierno británico. Es por eso

---

<sup>35</sup> TI Tecnologías de información.

que en 1980 la CCTA<sup>36</sup> desarrollara las primeras recomendaciones para facilitar la administración y optimización de las TI, luego estas se llamarían ITIL.

Inicialmente CCTA se enfocó a recopilar la información tendiente a verificar como las organizaciones dirigían la administración de los servicios de TI, logrando analizar y filtrar los problemas que se generaban; luego comprobaron la utilidad y los beneficios de sus recomendaciones. En los 90s muchas empresas del gobierno europeo adoptaron este marco de trabajo. Mientras tanto en 1991 se crea en Reino Unido la red mundial de grupos de usuarios de las TI que ofrecen mejores prácticas y guías basadas en estándares para la gestión de servicios de TI, esta red se denomina ITSMF (information technology Service Management Forum) y es el único grupo dedicado exclusivamente a este tipo de gestión.

En el año 2001, la CCTA y todas las actividades que estaban bajo su control, pasaron a formar parte de la OGC (Office Government Commerce) una oficina del ministerio de hacienda británico, convirtiéndose en la nueva entidad propietaria de ITIL. La OGC trabaja en asociación con la ITSMF y otras organizaciones contribuyendo a la industria de las mejores prácticas.

Posteriormente la OGC publicaría nuevas versiones de librerías de buenas prácticas, escritas por expertos internacionales de diversas organizaciones del sector público y privado.

Actualmente existen una serie de estándares o lineamientos con los cuales se pretende ofrecer a las organizaciones un marco de trabajo que les permita adoptar nuevas políticas interinstitucionales para la administración organizada y estructurada de los Servicios de TI. Entre los estándares comúnmente reconocidos a nivel mundial se encuentra ISO 9000, COBIT, BS-15000, ITIL, ISO 20000, etc.

Nosotros nos enfocaremos específicamente en el estudio del marco de trabajo ITIL, que se enfoca en la dirección de las TI de una empresa, especialmente en la gestión del servicio que ofrece.

---

<sup>36</sup> CCTA – Central Computer and telecommunication Agency

## **2.4.2 ITIL**

### **2.4.2.1 Definición**

Se considera una colección de guías especializadas, en temas organizacionales enfocados a la planeación, el suministro y el soporte de servicios de TI. Se reconoce como un estándar global que resume las mejores prácticas para el área de gerencia de servicios de TI. Enfocadas a describir específicamente que funciones o procesos son los que se recomienda desarrollar, mas no en el cómo desarrollarlos, para éste último es responsabilidad de la organización definir las estrategias y métodos necesarios para implementarla, siempre y cuando se adapten al tamaño, a la cultura y a las necesidades internas de la organización.

ITIL ofrece un marco de trabajo para las actividades del área de ti, describiendo los roles, tareas, procedimientos y responsabilidades que pueden ser adaptados en organizaciones de TI cuya finalidad sea mejorar la Gestión de sus Servicios; gracias a la cantidad de temas que cubre, se considera un elemento de referencia útil para las organizaciones, ya que permite fijar nuevos objetivos de mejora para la organización de TI, basándose en la calidad de servicio y en el desarrollo de los procesos de una manera eficaz y eficiente.

Las mejores prácticas son procesos que cubren las actividades más importantes a tener en cuenta al interior de las organizaciones se servicios de TI. ITIL es una colección coherente de mejores prácticas desarrolladas en la industria y no solamente puede ser adaptado por empresas públicas sino en privadas también

### **2.4.2.2 Objetivos**

ITIL se concentra en ofrecer servicios de alta calidad, dando especial importancia a las relaciones establecidas con los clientes, para lo cual el departamento de TI debe proveer y cumplir con todos los acuerdos de servicios previamente definidos. Es por eso los objetivos tratan de que exista una fuerte relación entre departamento de TI y clientes.

- Promover la visión de IT como proveedor de servicios.
- Generar mejoras en la calidad del suministro de los servicios de TI.
- Fomentar la reducción de costos de los servicios de TI.
- Alinear la prestación de los servicios de TI con las necesidades actuales y futuras del negocio de las organizaciones, además de mejorar la relación con los clientes.
- Estandarizar los procesos que forman parte de la gestión de servicios de TI.
- Promover el uso de un lenguaje común por parte de las personas para mejorar la comunicación al interior de las organizaciones.
- Servir de base para la certificación de las personas y de las organizaciones que desean adoptar el estándar.
- Mejorar la eficiencia, aumentando la efectividad.
- Reducir posibles riesgos que se puedan presentar.

#### **2.4.2.3 Beneficios**

ITIL centra sus esfuerzos en la satisfacción de los requerimientos organizacionales con la mejor relación costo beneficio, a través de la descripción de un enfoque sistemático y profesional de la gerencia de servicios de TI.

- Los acuerdos sobre la calidad de servicio mejoran la relación cliente departamento de TI.
- Mejoramiento de niveles de satisfacción de cliente
- Los estándares permiten realizar el control, la administración y operación de los recursos de la organización.
- El servicio ofrecido resulta más cómodo y con mayores detalles para el cliente.
- Se gestionan de mejor manera, la calidad, disponibilidad, fiabilidad y coste de los servicios ofrecidos en la organización.
- Ayuda al departamento de TI a acordar los puntos de contacto.

- El departamento de TI genera una estructura clara y centrada en los objetivos de la organización.
- El departamento de TI cuenta con un mayor control de la infraestructura y los servicios que tiene a cargo, obteniendo una visión clara de la capacidad del departamento.
- Permite administrar los cambios de una manera sencilla y fácil de manejar.
- Definición de funciones, roles y responsabilidades en el área de los servicios.
- Disminución de costos en la definición de procesos, procedimientos e instructivos de trabajo.
- Incremento y mejoras en la productividad y eficiencia organizacional.
- Cambio cultural hacia la provisión de servicios.

#### **2.4.2.4 Versiones de ITIL**

Versión 1, desarrollada en los años 80, consto con 10 libros enfocados a la gestión de servicio (entrega y soporte), y luego con 30 libros complementarios con diversos temas.

Versión 2, reestructurada entre 1999 y 2001, reorganizando de manera más sencilla, abarcando la gestión de servicio con 2 grandes volúmenes, y 10 libros en total que se convirtieron en la base de las mejores prácticas de TI.

Versión 3, liberada en 2007, redujo las publicaciones a 5 volúmenes articulados, las cuales ya se enfocan en el concepto y desarrollo del ciclo de vida de los servicios de TI.

#### **2.4.2.5 Estructura**

El marco de trabajo ITIL se conforma por 5 elementos:

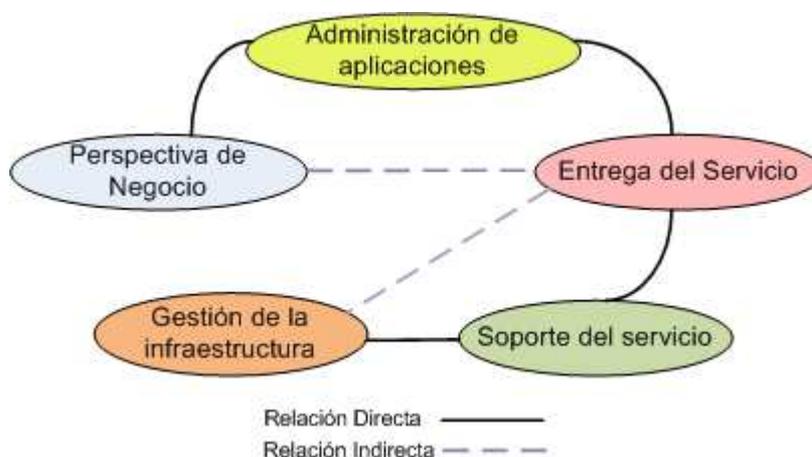
- The business perspective (Perspectiva del negocio)
- Managing applications (Administración de aplicaciones)

- Deliver IT services (Suministro de servicios de TI)
- Support IT services (Soporte de servicios de TI)
- Manage the infraestructre (Gestión de la infraestructura)

Administración de aplicaciones, se encarga de que la perspectiva de negocio este fundamentada en la prestación o entrega de servicios y viceversa. La entrega de servicios, depende de un soporte que esté disponible para eventuales fallos, pero para que esté disponible es necesario contar con una gestión de la infraestructura.

De otra forma se podría decir que la prestación de servicios y la perspectiva de negocio tienen una relación indirecta, así como la gestión de la infraestructura ayuda indirectamente a cumplir el propósito de la prestación del servicio.

En la figura 2.5 se muestran las relaciones entre los elementos del marco de trabajo ITIL.



**Figura 2.5 Elementos del marco de trabajo de ITIL.**

## 2.4.3 GESTIÓN DEL SERVICIO – ITIL V2

### 2.4.3.1 Entrega del servicio

Es uno de los elementos principales de la gestión del servicio describe los aspectos a tener en cuenta para la planificar y mejorar constantemente el servicio de TI a largo plazo de tal manera que el suministro del servicio se mantenga en un alto nivel y permite satisfacer a los clientes actuales y futuros. A continuación se describen algunos de los aspectos relacionados.

#### 2.4.3.1.1 Availability Management (Gestión de la disponibilidad)

Este proceso garantiza que los servicios de TI puedan ser accedidos de una manera confiable y se encuentren disponibles y funcionando correctamente cada vez que los clientes o usuarios así lo requieran, enmarcados en los SLAs<sup>37</sup> que se hayan definido para la prestación de servicios.

#### 2.4.3.1.2 Capacity Management (Gestión de la Capacidad)

Este proceso asegura la existencia de cierta capacidad a nivel de infraestructura de TI, la cual se encuentra disponible constantemente para satisfacer os requerimientos del negocio a nivel de volumen de transacciones, tiempo de proceso, tiempo de respuesta y ante todo tomar en cuenta la viabilidad cuantitativa y económica para no incurrir en costos desproporcionados.

---

<sup>37</sup> SLA-Service Level Agreement, acuerdo de nivel de servicio. Es un contrato escrito entre el proveedor de un servicio y su cliente para limitar la calidad de dicho servicio.

#### 2.4.3.1.3 Financial Management for IT services (Gestión Financiera para los servicios de TI)

Se concentra en realizar un adecuado manejo del recurso financiero (ingresos y egresos) asociado a lo que implica la prestación de los servicios de TI, siempre enfocándose en el cumplimiento de los SLAs definidos.

#### 2.4.3.1.4 Service Continuity Management (Gestión para la continuidad del servicio)

Evita que una grave e imprevista interrupción en el servicio atente contra la continuidad del negocio, por lo que se concentra en la preparación y planificación de las medidas que se deben tomar en caso de que suceda algún error en el suministro del servicio.

#### 2.4.3.1.5 Service level Management (Gestión de los niveles de servicio)

Define los servicios de TI ofrecidos, formalizándose a través de SLA y SLO<sup>38</sup>, además este proceso evalúa el impacto que se genera si ocurre un cambio en la calidad del servicio o en los SLAs luego de que estos cambios hayan sido propuestos e implementados, asegurando que cualquier impacto negativo sobre la calidad del servicio sea relativamente bajo.

### 2.4.3.2 Soporte del servicio

Es otro de los ejes principales de la Gestión del servicio de TI, se encarga de describir la forma en cómo los clientes y usuarios pueden acceder a los servicios que les permitan apoyar el desarrollo de sus actividades diarias y las del negocio, así como, la forma en que dichos servicios deben ser soportados.

---

<sup>38</sup> SLO Service Level Objectives.

#### 2.4.3.2.1 Service Desk

Se lo conoce como el único punto de contacto entre el cliente y el usuario con los proveedores del servicio, para todo lo relacionado con el suministro del servicio. Además es el punto de partida para informar sobre los incidentes y la toma de solicitudes de servicio realizadas por los usuarios.

Su obligación es mantener informados a los usuarios del servicio sobre los eventos, acciones y oportunidades que pueden llegar a afectar de alguna manera la disponibilidad del servicio y por consiguiente la continuidad en el suministro del servicio en el día a día, todo esto es posible a través del registro, monitoreo y resolución de problemas.

Dentro del marco de trabajo de ITIL, Service desk no fue concebido como un proceso sino como una función a desarrollar dentro de la organización que provee los servicios. Esta función deberá encargarse de algunas tareas.

- Ser el punto primario de contacto con los clientes y usuarios
- Recibir y atender todas las solicitudes, consultas e inquietudes de los clientes y usuarios relacionados con el suministro del servicio.
- Documentar, priorizar y realizar un seguimiento adecuado a las solicitudes de modificaciones o cambios realizados por los usuarios; atender todos los procesos de la Gerencia de Servicios definidos por ITIL.
- Mantener informados sobre el estado y progreso de las solicitudes a los usuarios que las realizaron.
- Clasificar las solicitudes recibidas e iniciar su proceso de respuesta según los acuerdos del nivel de servicio SLA y procedimientos establecidos.
- Cuando se necesite un soporte de segundo nivel, realizar la coordinación entre el suministro, el soporte del servicio, los proveedores y participación de terceros.
- Gestionar la restauración del servicio con el mínimo impacto en el negocio.

- Cerrar las solicitudes de servicio realizadas por los usuarios, y aplicar una evaluación de la satisfacción del cliente.
- Realzar seguimiento de los SLAs definidos tomando las acciones necesarias en caso de presentarse incumplimientos.
- Suministrar la información solicitada por la Gerencia de TI para mantener y mejorar la calidad de los servicios prestados.

Según su ubicación un service desk puede ser de 3 tipos:

- Service desk local, a través del cual se busca canalizar localmente las necesidades y solicitudes del negocio. Es práctico en varios sitios que requieren servicio de soporte pero siendo grande los costos que se pueden llegar a incurrir, ya que el servicio es ofrecido en diferentes lugares.
- Service desk central, busca que todos los requerimientos del servicio sean registrados en una ubicación central, minimizando costos operacionales ya que existe solamente una mesa de ayuda a nivel organizacional que atiende todos los requerimientos.
- Service desk virtual, cuya finalidad es ofrecer el servicio en cualquier parte del mundo a través de la red, no importa su ubicación física ya que el servicio se encuentra disponible en todo momento. Y su éxito se da siempre y cuando todos los usuarios de la organización cuenten con infraestructura tecnológica para poder acceder a ella.

#### 2.4.3.2.2 Configuration Management (Gestión de la configuración)

Tiene como objetivo controlar los elementos de configuración que forman parte de la infraestructura de TI, se encarga de todos los procesos, herramientas y técnicas necesarias para lograrlo. Además este proceso proporciona información confiable y actualizada de los elementos específicos de la infraestructura y de las relaciones entre dichos elementos, para garantizar la integración con las demás disciplinas de la gestión del servicio. Además permite el desarrollo de los servicios

informáticos de mejor calidad de forma viable y económica, suministrando información importante para el cálculo de los costos de los servicios.

#### 2.4.3.2.3 Incident Management (Gestión de Incidentes)

Su objetivo es resolver cualquier incidente que genere una interrupción en la prestación del servicio, restaurándolo de la manera más rápida y efectiva posible. Este proceso no se detiene a evaluar, encontrar y analizar lo causó el incidente sino solo solucionarlo de cualquier forma, ya sea definitivamente o temporalmente.

#### 2.4.3.2.4 Problem management (Gestión de problemas)

Este proceso si se dedica a identificar las causas que ocasionan los problemas que se presentan en la infraestructura de TI y su solución definitiva para evitar que vuelva a ocurrir.

#### 2.4.3.2.5 Changing management (Gestión de los cambios)

Este proceso tiene relación con el Configuration management y con el Service level management, ya que se encarga de realizar una valoración de los cambios y garantizar que pueden ejecutarse ocasionando el mínimo impacto en la prestación de los servicios de TI y en la infraestructura actual o nueva, y asegura de manera simultánea la trazabilidad de los cambios.

#### 2.4.3.2.6 Release management (Gestión de las versiones)

Un cambio puede implicar la instalación de nuevo hardware, o la instalación de nuevas versiones de software o solo actualización o aumentó de datos, por eso es necesario que estas acciones sean controladas por un proceso aparte, llamado Gestión de las versiones.

### **2.3.4 HELP DESK**

Helpdesk o Mesa de escritorio, se lo concibió principalmente dentro de las prácticas ITIL, como un service desk local, pero con el tiempo ha evolucionado de tal forma que una Mesa de ayuda puede ser local, ubicándose en el mismo sitio donde ocurre la incidencia, centralizada si la información de ayuda se encuentra en una sola ubicación y virtual si brinda el soporte medio de software en línea o aplicaciones web.

Precisamente nuestra aplicación se ayudará de esto, para poder utilizar conceptos y definir practicas que ayuden a la gestión del servicio a partir de los dos procedimientos que ofreceremos.

## **CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN.**

En este capítulo se detalla el análisis previo y la implementación del prototipo de software, utilizando una metodología de tipo ágil, la metodología XP (eXtreme Programming). Esta metodología nos muestra de una forma clara y concreta los procedimientos necesarios para realizar el diseño, partiendo de una planificación inicial resultado de un levantamiento de información, pasando por el proceso de implementación y terminando con un proceso de pruebas.

Pero antes debemos mencionar algunas características propias de nuestro prototipo de software y que influyen en la manera de seguir los procedimientos de la metodología:

- Los requerimientos de interfaces y de procesos planteados por el usuario del sistema, fueron entregados por el futuro usuario del software, por lo que se omite el proceso del levantamiento de información entre el usuario y el desarrollador.
- La elección de las herramientas y tecnologías a usarse se basó en los criterios y comparaciones expuestas en el capítulo 2.
- La manera en cómo se presenten los datos al usuario y como influya esto en el mejoramiento del negocio, necesariamente debe seguir patrones, estándares. En nuestro caso nos ayudamos de las practicas y terminología recomendada en ITIL V.2

### **3.1 HERRAMIENTAS DE LA METODOLOGÍA XP**

La metodología ágil XP tiene como unidad de desarrollo a la “Historia de usuario”, Una historia de usuario describe un proceso específico completo, la suma de historias de usuario conforman un proceso general o iteración. Una historia describe los procesos que efectuarán los usuarios de la aplicación. La historia se

detalla aun más, gracias a la definición de tareas, que muestran información importante para el desarrollador. Y las pruebas de aceptación del desarrollo, nos permiten evaluar el desarrollo, encontrar fallas y corregirlas.

### **3.1.1 ITERACIONES**

En desarrollo de software, Iteración es la repetición de una serie de instrucciones que permiten obtener un resultado determinado. Comúnmente una iteración corresponde a un proceso completo que se efectúe dentro de una aplicación.

### **3.1.2 HISTORIAS DE USUARIO**

Una historia de usuario representa a procesos específicos que se debe seguir para cumplir una iteración. Se caracteriza por definir los alcances, la prioridad y los usuarios que van a participar en el proceso.

#### **3.1.2.1 Elementos de una Historia de usuario**

- **Identificador:** Identifica a una historia de usuario. Su nomenclatura será el número precedido de HU de Historia de Usuario.
- **Nombre:** Nombre de la Historia de Usuario.
- **Iteración:** Indica a que Iteración pertenece la historia de usuario.
- **Usuario:** Indica el o los usuarios del software, que participan en la historia.
- **Descripción:** Describe el o los procedimientos que componen la historia.
- **Estimación:** Es un valor numérico que indica el nivel de dificultad que tendrá el desarrollo de la historia.
- **Prioridad:** Es un valor que indica la prioridad que tiene el desarrollo de la historia.
- **Dependencias:** Indica la o las historias de usuario que necesitan ser desarrolladas antes que la historia actual.

### **3.1.3 TAREAS DE UNA HISTORIA DE USUARIO**

Las tareas tienen información importante para el personal de desarrollo de la aplicación. Tratan de especificar lo necesario para optimizar el tiempo de desarrollo.

#### **3.1.3.1 Elementos de una Tarea**

- **Identificador:** Identifica a una Tarea de desarrollo. La nomenclatura es un número precedido de la letra T de Tarea.
- **Nombre:** Muestra el nombre de la Tarea.
- **Historia:** Muestra la Historia de Usuario a la que pertenece la Tarea.
- **Tipo:** Describe un tipo al que puede pertenecer la Tarea. El tipo más común es el tipo “desarrollo” e indica que la tarea consta en desarrollar una parte específica del software.
- **Fecha de Inicio y de Fin:** Muestra la fecha de inicio y fin de una tarea. Estas fechas dependen de la estimación de la historia a la que pertenece la tarea.
- **Descripción:** Describe los procedimientos necesarios para realizar la tarea.

## **3.2 PLANIFICACIÓN DEL PROYECTO APLICANDO LA METODOLOGÍA.**

El proceso que seguiremos para empezar el desarrollo es el siguiente:

- Definir las herramientas de la metodología. Iteraciones, historias, tareas.
- Definir los perfiles de usuario de la aplicación, basándonos en la arquitectura organizacional de la empresa.
- Redactar las herramientas de la metodología y empezar el desarrollo.
- Realizar pruebas de aceptación del desarrollo.

A continuación en la tabla 3.1 se definen las herramientas de nuestro desarrollo aplicando la metodología.

<b>Iteración</b>	<b>Historia de Usuario</b>		<b>Tarea de Historia</b>	
<b>I1</b>	<b>HU1</b>	Administración de la Empresa	<b>T1</b>	Mostrar Datos de la empresa
<b>I1</b>	<b>HU1</b>	Administración de la Empresa	<b>T2</b>	Modificación de la empresa
<b>I1</b>	<b>HU2</b>	Administración de Vendedores	<b>T3</b>	Listar Vendedores
<b>I1</b>	<b>HU2</b>	Administración de Vendedores	<b>T4</b>	Crear un vendedor
<b>I1</b>	<b>HU2</b>	Administración de Vendedores	<b>T5</b>	Modificar un Vendedor
<b>I1</b>	<b>HU2</b>	Administración de Vendedores	<b>T6</b>	Eliminar un vendedor
<b>I1</b>	<b>HU3</b>	Administración de Usuarios	<b>T7</b>	Listar los Usuarios
<b>I1</b>	<b>HU3</b>	Administración de Usuarios	<b>T8</b>	Crear nuevo Usuario
<b>I1</b>	<b>HU3</b>	Administración de Usuarios	<b>T9</b>	Modificar un usuario
<b>I1</b>	<b>HU3</b>	Administración de Usuarios	<b>T10</b>	Eliminar un usuario
<b>I1</b>	<b>HU4</b>	Administración de Problemas	<b>T11</b>	Listar los problemas
<b>I1</b>	<b>HU4</b>	Administración de Problemas	<b>T12</b>	Crear un problema
<b>I1</b>	<b>HU4</b>	Administración de Problemas	<b>T13</b>	Modificar un problema
<b>I1</b>	<b>HU4</b>	Administración de Problemas	<b>T14</b>	Eliminar un problema
<b>I1</b>	<b>HU5</b>	Administración de Módulos	<b>T15</b>	Listar los Módulos
<b>I1</b>	<b>HU5</b>	Administración de Módulos	<b>T16</b>	Crear un modulo
<b>I1</b>	<b>HU5</b>	Administración de Módulos	<b>T17</b>	Modificar un modulo
<b>I1</b>	<b>HU5</b>	Administración de Módulos	<b>T18</b>	Eliminar un modulo
<b>I1</b>	<b>HU6</b>	Administración de Soluciones	<b>T19</b>	Listar Soluciones
<b>I1</b>	<b>HU6</b>	Administración de Soluciones	<b>T20</b>	Crear una solución
<b>I1</b>	<b>HU6</b>	Administración de Soluciones	<b>T21</b>	Modificar una solución
<b>I1</b>	<b>HU6</b>	Administración de Soluciones	<b>T22</b>	Eliminar una solución
<b>I1</b>	<b>HU7</b>	Administración de Sugerencias	<b>T23</b>	Listar las sugerencias
<b>I1</b>	<b>HU7</b>	Administración de Sugerencias	<b>T24</b>	Crear una sugerencia
<b>I1</b>	<b>HU7</b>	Administración de Sugerencias	<b>T25</b>	Modificar una sugerencia
<b>I1</b>	<b>HU7</b>	Administración de Sugerencias	<b>T26</b>	Eliminar una sugerencia
<b>I1</b>	<b>HU8</b>	Ingreso a la Aplicación	<b>T27</b>	Procedimiento de Login
<b>I1</b>	<b>HU9</b>	Verificación de datos de usuario	<b>T28</b>	Mostrar datos del usuario actual
<b>I1</b>	<b>HU9</b>	Verificación de datos de usuario	<b>T29</b>	Modificar datos del usuario actu.
<b>I2</b>	<b>HU10</b>	Administración de un Contrato	<b>T30</b>	Listar los contratos
<b>I2</b>	<b>HU10</b>	Administración de un Contrato	<b>T31</b>	Crear un contrato
<b>I2</b>	<b>HU10</b>	Administración de un Contrato	<b>T32</b>	Anular o Dar de baja un contrato
<b>I2</b>	<b>HU11</b>	Obtención de claves	<b>T33</b>	Obtención de clave
<b>I2</b>	<b>HU12</b>	Mesa de ayuda-ubicación del problema	<b>T34</b>	Ubicación del problema
<b>I2</b>	<b>HU13</b>	Mesa de ayuda-Solución al problema	<b>T35</b>	Mostrar sugerencia de solución al problema
<b>I3</b>	<b>HU14</b>	Gestión de Licencias Obtenidas	<b>T36</b>	Mostrar listado de licencias obtenidas
<b>I3</b>	<b>HU15</b>	Gestión de Incidencias Ocurridas	<b>T37</b>	Mostrar listado de Incidencias ocurridas

**Tabla 3.1 Herramientas de la metodología XP en nuestro desarrollo.**

En la tabla 3.2 se muestran los perfiles de usuario que se aplican para nuestro desarrollo, el momento en el que son creados o definidos y a que módulos pueden acceder.

<b>Perfil</b>	<b>Momento de definición</b>	<b>Acceso</b>
<b>SUPERUSUARIO</b>	En la instalación de la aplicación	Todos los módulos de la aplicación
<b>USUARIO ADMINISTRADOR</b>	En la creación de un usuario por parte del superusuario u otro administrador	Todos los módulos excepto la configuración de parámetros de la empresa
<b>USUARIO CLIENTE</b>	En la creación de un usuario por parte de un administrador	Solo módulo de licencias y Mesa de ayuda

**Tabla 3.2 Perfiles de Usuario**

En la tabla 3.3 se muestra los departamentos de la empresa, los perfiles asignados y las labores encargadas a realizar.

<b>Departamento</b>	<b>Perfil</b>	<b>Procesos encargados</b>
<b>PRESIDENCIA</b>	USUARIO ADMINISTRADOR	Gestión de Licencias, e Incidencias
<b>GERENCIA GENERAL</b>	SUPERUSUARIO	Gestión de la empresa, gestión de usuarios administrador, licencias e incidencias
<b>CONTABILIDAD</b>	USUARIO ADMINISTRADOR	Gestión de Licencias
<b>DEPARTAMENTO TECNICO</b>	USUARIO ADMINISTRADOR	Gestión de problemas, soluciones, módulos, sugerencias, incidencias
<b>DEPARTAMENTO COMERCIAL</b>	USUARIO ADMINISTRADOR	Gestión de Usuarios tipo Cliente, gestión de contratos
<b>CLIENTE</b>	USUARIO CLIENTE	Obtención de Licencias y Mesa de Ayuda.

**Tabla 3.3 Perfiles de Usuario para cada departamento de la empresa**

### 3.2.1 REDACCIÓN DE LAS HERRAMIENTAS DE LA PRIMERA ITERACIÓN

Esta iteración corresponde al proceso completo de ingreso de información base para el funcionamiento de la aplicación.

#### 3.2.1.1 Administración de Tablas Base

##### 3.2.1.1.1 Administración de la empresa

La empresa es la Corporación MilenioSystem S.A dueña del software y la responsable del servicio que se quiere brindar.

ID: HU1	USUARIO: SuperUsuario	ITERACION: 1
NOMBRE: Administracion de la empresa		
DESCRIPCIÓN: El superusuario puede modificar la información de la empresa distribuidora del sistema		
ESTIMACIÓN: 1	PRIORIDAD: 1	DEPENDENCIAS:

Tareas de HU1: el usuario con permisos debe poder ver los datos de la empresa y poder modificarlos.

ID: T1	HISTORIA: HU1	
NOMBRE: Mostrar datos de la empresa		
TIPO: Desarrollo	FECHA INICIO: 22/09/2010	FECHA FIN: 22/09/2010
DESCRIPCIÓN: Se mostrara la información importante de la empresa en forma horizontal. -RUC.nombre, direccion, sitio web, email  Existirán enlaces para Volver al menu anterior, ver los datos verticalmente y modificar los datos de la empresa.		

ID:	T2	HISTORIA:	HU1
NOMBRE: Modificación de los datos de la empresa.			
TIPO:	Desarrollo	FECHA INICIO:	22/09/2010
		FECHA FIN:	22/09/2010
DESCRIPCIÓN: Se podrá únicamente modificar los datos de la empresa. Tomando en especial consideración el campo "FRASECLAVE" ya que esta cadena sirve para otorgar claves de activación de licencias. -Debe existir una opción para autogenerar la fraseclave en el caso de que se desee. Existirá enlaces para ver los datos de la empresa o para volver al Menú anterior.			

### 3.2.1.1.2 Administración de vendedores

Los vendedores de la empresa son importantes ya que en el momento de crear un usuario se debe especificar el vendedor el cual lo contacto o le vendió el sistema.

ID:	HU2	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE: Administracion de Vendedores de la empresa					
DESCRIPCIÓN: Se pueden crear, modificar y eliminar vendedores del sistema Milenio					
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	

Tareas de HU2: el usuario con permisos debe poder visualizar los datos en una lista de todos los vendedores. Y además debe poder crear, modificar y eliminar vendedores.

ID:	T3	HISTORIA:	HU2
NOMBRE:	Lista de Vendedores		
TIPO:	Desarrollo	FECHA INICIO:	23/09/2010
		FECHA FIN:	23/09/2010
DESCRIPCIÓN:			
<p>Se muestra una lista de los vendedores registrados, visualizando los campos: ID, cedula y nombre.</p> <p>Cada ítem de la lista tiene su enlaces hacia la vista, modificación o eliminación de un registro.</p> <p>Existirá enlaces para crear un nuevo vendedor y para volver al menu anterior.</p>			

ID:	T4	HISTORIA:	HU2
NOMBRE:	Creación de un Vendedor		
TIPO:	Desarrollo	FECHA INICIO:	24/09/2010
		FECHA FIN:	24/09/2010
DESCRIPCIÓN:			
<p>Se registra un nuevo vendedor en la base de datos. El campo ID, se incrementa automáticamente por lo tanto debe existir un cuadro para ingresarlo.</p> <p>-Se debe validar el campo cedula.</p> <p>Existirá enlaces para ver la lista de vendedores y para volver al menú anterior.</p>			

ID:	T5	HISTORIA:	HU2
NOMBRE:	Modificación de datos de un vendedor		
TIPO:	Desarrollo	FECHA INICIO:	27/09/2010
		FECHA FIN:	27/09/2010
DESCRIPCIÓN:			
<p>Se pueden modificar los datos de un vendedor registrado. El ID es el único dato que no se puede mostrar ni modificar.</p> <p>-Se debe validar el campo cedula</p> <p>Existirá enlaces para ver los datos del vendedor actual, ver la lista de vendedores y para volver al menú anterior.</p>			

ID:	T6	HISTORIA:	HU2
NOMBRE:	Eliminación de un vendedor		
TIPO:	Desarrollo	FECHA INICIO:	28/09/2010
		FECHA FIN:	28/09/2010
DESCRIPCIÓN:	<p>Se puede eliminar un vendedor, si no existen usuarios registrados con este. Ya que su eliminación afectaría a la integridad referencial.</p>		

### 3.2.1.1.3 Administración de usuarios

Los usuarios del software pueden ser empleados de la empresa o clientes de la empresa. En nuestro prototipo el empleado de la empresa se lo define como usuario administrador.

ID:	HU3	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE:	Administracion de Usuarios				
DESCRIPCIÓN:	<p>Se pueden crear, modificar y eliminar usuarios que utilizaran el software.</p>				
ESTIMACIÓN:	3	PRIORIDAD:	1	DEPENDENCIAS:	

Tareas de HU3: el usuario con permisos debe poder visualizar los registros en una lista. Y además debe poder crear, modificar y eliminar usuarios.

ID:	T7	HISTORIA:	HU3
NOMBRE:	Lista de Usuarios		
TIPO:	Desarrollo	FECHA INICIO:	29/09/2010
		FECHA FIN:	29/09/2010
DESCRIPCIÓN:			
Se muestra una lista de usuarios registrados.			
-Id,Nombre, Cedula,Tipo, Direccion,Telefono,Email,Contacto,Actividad,Licencias obtenidas, maximos de licencias, vendedor.			
Cada item de la lista tiene su enlaces hacia la vista, modificación o eliminación de un registro.			
Existirá enlaces para crear un nuevo usuario y para volver al menu anterior.			

ID:	T8	HISTORIA:	HU3
NOMBRE:	Creación de un usuario		
TIPO:	Desarrollo	FECHA INICIO:	30/09/2010
		FECHA FIN:	04/10/2010
DESCRIPCIÓN:			
Se registra un nuevo usuario en la base de datos, llenando los datos necesarios. Tomando en cuenta en especial a los campos "TIPO" y al campo "Licencias Maximas".			
-Se debe validar el campo RUC o cedula, el email.			
Existirá enlaces para ver la lista de usuarios y para volver al menú anterior.			

ID:	T9	HISTORIA:	HU3
NOMBRE:	Modificación de datos de un usuario		
TIPO:	Desarrollo	FECHA INICIO:	05/10/2010
		FECHA FIN:	06/10/2010
DESCRIPCIÓN:			
Se pueden modificar los datos de un usuario registrado. Desde aquí se puede modificar el valor del campo "Licencias Máximas" si previamente el cliente llega a un acuerdo con la empresa.			
El campo contraseña debera volver a ser digitado 2 veces.			
-Se debe validar el campo RUC o cedula, el email.			
Existirá enlaces para ver los datos del usuario actual, ver la lista de usuarios y para volver al menú anterior.			

ID:	T10	HISTORIA:	HU3
NOMBRE:	Eliminación de un usuario		
TIPO:	Desarrollo	FECHA INICIO:	07/10/2010
		FECHA FIN:	07/10/2010
DESCRIPCIÓN:	<p>Se puede eliminar un usuario, si no tiene registros de licencias o incidencias, ya que afectaría a la integridad referencial de la aplicación.</p>		

#### 3.2.1.1.4 Administración de Problemas

Los problemas son la descripción de los posibles inconvenientes que ocurren con frecuencia durante el manejo del sistema. Estos problemas pueden ser de todo tipo, y pueden o no estar relacionados con el sistema.

ID:	HU4	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE:	Administración de Problemas del sistema.				
DESCRIPCIÓN:	<p>Se pueden crear, modificar y eliminar los diferentes problemas que pueden ocurrir durante la utilización del sistema Milenio.</p>				
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	

Tareas de HU4: el usuario con permisos debe poder visualizar los registros en una lista. Y además debe poder crear, modificar y eliminar problemas.

ID:	T11	HISTORIA:	HU4
NOMBRE:	Lista de Problemas		
TIPO:	Desarrollo	FECHA INICIO:	08/10/2010
		FECHA FIN:	08/10/2010
DESCRIPCIÓN:			
<p>Se muestra una lista de la descripción de los problemas que ocurren en el manejo del sistema Milenio.</p> <p>-Se visualiza el ID y la descripción.</p> <p>Cada ítem de la lista tiene sus enlaces hacia la vista, modificación o eliminación de un registro.</p> <p>Existirá enlaces para crear un nuevo problema y para volver al menú anterior.</p>			

ID:	T12	HISTORIA:	HU4
NOMBRE:	Creación de un Problema		
TIPO:	Desarrollo	FECHA INICIO:	11/10/2010
		FECHA FIN:	11/10/2010
DESCRIPCIÓN:			
<p>Se puede registrar la descripción de un nuevo problema.</p> <p>Existirá enlaces para ver la lista de problemas y para volver al menú anterior.</p>			

ID:	T13	HISTORIA:	HU4
NOMBRE:	Modificación de un Problema		
TIPO:	Desarrollo	FECHA INICIO:	12/10/2010
		FECHA FIN:	12/10/2010
DESCRIPCIÓN:			
<p>Se pueden modificar la descripción del problema.</p> <p>Existirá enlaces para ver los datos del problema actual, ver la lista de problemas y para volver al menú anterior.</p>			

ID:	T14	HISTORIA:	HU4
NOMBRE:	Eliminación de un Problema		
TIPO:	Desarrollo	FECHA INICIO:	13/10/2010
		FECHA FIN:	13/10/2010
DESCRIPCIÓN:	<p>Se puede eliminar un problema, siempre y cuando no este registrado en la tabla sugerencias, ya que se afectaría la integridad referencial.</p>		

### 3.2.1.1.5 Administración de Módulos

La lista de módulos nos permitirá identificar en que módulo o interfaz específica del sistema ocurrió un problema.

ID:	HU5	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE:	Administración de Modulos del sistema.				
DESCRIPCIÓN:	<p>Se pueden crear, modificar y eliminar nombres de módulos del sistema Milenio.</p>				
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	1

Tareas de HU5: el usuario con permisos debe poder visualizar los registros en una lista. Y además debe poder crear, modificar y eliminar módulos.

ID:	T15	HISTORIA:	HU5
NOMBRE:	Lista de Módulos		
TIPO:	Desarrollo	FECHA INICIO:	14/10/2010
		FECHA FIN:	14/10/2010
DESCRIPCIÓN:			
<p>Se muestra una lista de los módulos que contiene el sistema Milenio</p> <p>-Se muestra el ID y el nombre del módulo.</p> <p>Cada item de la lista tiene su enlaces hacia la vista, modificación o eliminación de un registro.</p> <p>Existirá enlaces para crear un nuevo módulo y para volver al menú anterior.</p>			

ID:	T16	HISTORIA:	HU5
NOMBRE:	Creación de un módulo		
TIPO:	Desarrollo	FECHA INICIO:	15/10/2010
		FECHA FIN:	15/10/2010
DESCRIPCIÓN:			
<p>Se registra el nombre de un nuevo modulo del sistema Milenio.</p> <p>-Se debe validar el nombre del modulo con un maximo de 20 caracteres</p> <p>Existirá enlaces para ver la lista de módulos y para volver al menú anterior.</p>			

ID:	T17	HISTORIA:	HU5
NOMBRE:	Modificación del nombre de un módulo.		
TIPO:	Desarrollo	FECHA INICIO:	18/10/2010
		FECHA FIN:	18/10/2010
DESCRIPCIÓN:			
<p>Se puede modificar el nombre de un módulo ya registrado.</p> <p>-Se debe validar el nombre del módulo con máximo 20 caracteres</p> <p>Existirá enlaces para ver los datos del módulo actual, ver la lista de módulos y para volver al menú anterior.</p>			

ID:	T18	HISTORIA:	HU5
NOMBRE:	Eliminación de un módulo		
TIPO:	Desarrollo	FECHA INICIO:	19/10/2010
		FECHA FIN:	19/10/2010
DESCRIPCIÓN:	<p>Se puede eliminar un módulo, siempre y cuando no este involucrado en una sugerencia de solución ya que afectaría a la integridad referencial de la aplicación.</p>		

### 3.2.1.1.6 Administración de soluciones

Las soluciones son instrucciones o recomendaciones que se describen de forma clara y con un lenguaje sencillo, con la finalidad de resolver un determinado problema.

ID:	HU6	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE:	Administración de Soluciones a problemas del sistema.				
DESCRIPCIÓN:	<p>Se pueden crear, modificar y eliminar posibles soluciones a los problemas que ocurren al utilizar el sistema Milenio.</p>				
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	1

Tareas de HU6: el usuario con permisos debe poder visualizar los registros en una lista. Y además debe poder crear, modificar y eliminar soluciones.

ID:	T19	HISTORIA:	HU6
NOMBRE:	Lista de Soluciones		
TIPO:	Desarrollo	FECHA INICIO:	20/10/2010
		FECHA FIN:	20/10/2010
DESCRIPCIÓN:	<p>Se muestra una lista de soluciones registradas.</p> <p>-Se muestra el ID y la descripción de la solución.</p> <p>Cada item de la lista tiene su enlaces hacia la vista, modificación o eliminación de un registro.</p> <p>Existirá enlaces para crear una nueva solución y para volver al menú anterior.</p>		

ID:	T20	HISTORIA:	HU6
NOMBRE:	Creación de una Solución		
TIPO:	Desarrollo	FECHA INICIO:	21/10/2010
		FECHA FIN:	21/10/2010
DESCRIPCIÓN:	<p>Se registra una nueva solución con su descripción.</p> <p>Existirá enlaces para ver la lista de soluciones y para volver al menú anterior.</p>		

ID:	T21	HISTORIA:	HU6
NOMBRE:	Modificación de una solución		
TIPO:	Desarrollo	FECHA INICIO:	22/10/2010
		FECHA FIN:	22/10/2010
DESCRIPCIÓN:	<p>Se puede modificar la descripción de la solución.</p> <p>Existirá enlaces para ver los datos de la solución actual, ver la lista de soluciones y para volver al menú anterior.</p>		

ID:	T22	HISTORIA:	HU6
NOMBRE:	Eliminación de una Solución		
TIPO:	Desarrollo	FECHA INICIO:	25/10/2010
		FECHA FIN:	25/10/2010
DESCRIPCIÓN:	<p>Se puede eliminar una solución, siempre y cuando no este involucrado en una sugerencia de solución ya que afectaría a la integridad referencial de la aplicación.</p>		

### 3.2.1.1.7 Administración de sugerencias

Una sugerencia es un registro que enlaza un problema, el módulo donde ocurrió y la solución al problema.

ID:	HU7	USUARIO:	Usuario administrador	ITERACION:	1
NOMBRE:	Administración de Sugerencias de solución				
DESCRIPCIÓN:	<p>Se pueden crear, modificar y eliminar sugerencias de solución. La sugerencia representa el enlazado de una solución, un problema y un modulo.</p>				
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	1

Tareas de HU7: el usuario con permisos debe poder visualizar los registros en una lista. Y además debe poder crear, modificar y eliminar sugerencias.

ID:	T23	HISTORIA:	HU7
NOMBRE:	Lista de Sugerencias		
TIPO:	Desarrollo	FECHA INICIO:	26/10/2010
		FECHA FIN:	26/10/2010
DESCRIPCIÓN:			
<p>Se muestra una lista de las sugerencias registradas.</p> <p>-se muestra el ID, la descripción del problema, el nombre del módulo y el ID de la solución.</p> <p>Cada ítem de la lista tiene sus enlaces hacia la vista, modificación o eliminación de un registro.</p> <p>Existirán enlaces para crear una nueva sugerencia y para volver al menú anterior.</p>			

ID:	T24	HISTORIA:	HU7
NOMBRE:	Creación de una nueva Sugerencia		
TIPO:	Desarrollo	FECHA INICIO:	27/10/2010
		FECHA FIN:	27/10/2010
DESCRIPCIÓN:			
<p>Se registra una nueva sugerencia, enlazando el problema que ocurrió, el módulo donde ocurrió y la solución al problema.</p> <p>-Dar la opción de partir de una sugerencia anterior, de esta manera una sugerencia puede pertenecer a un nivel específico.</p> <p>Existirán enlaces para ver la lista de sugerencias y para volver al menú anterior.</p>			

ID:	T25	HISTORIA:	HU7
NOMBRE:	Modificación de datos de una Sugerencia		
TIPO:	Desarrollo	FECHA INICIO:	28/10/2010
		FECHA FIN:	28/10/2010
DESCRIPCIÓN:			
<p>Se pueden modificar los datos de una sugerencia.</p> <p>Existirán enlaces para ver los datos del usuario actual, ver la lista de usuarios y para volver al menú anterior.</p>			

ID:	T26	HISTORIA:	HU7
NOMBRE:	Eliminación de una Sugerencia		
TIPO:	Desarrollo	FECHA INICIO:	29/10/2010
		FECHA FIN:	29/10/2010
DESCRIPCIÓN:	<p>Se puede eliminar una sugerencia, siempre y cuando no este involucrado en una incidencia ya que afectaría a la integridad referencial de la aplicación.</p>		

### 3.2.1.1.8 Ingreso a la Aplicación

Esta historia describe el ingreso a la aplicación por parte de un usuario cualquiera.

ID:	HU8	USUARIO:	Cliente, Usuario administrador, Superusuario	ITERACION:	1,2,3
NOMBRE:	Ingreso a la aplicación				
DESCRIPCIÓN:	<p>El usuario ingresa a la aplicación a digitando su numero de cedula y su clave.</p>				
ESTIMACIÓN:	2	PRIORIDAD:	1	DEPENDENCIAS:	1

Tarea de HU8: esta tarea garantiza que un usuario registrado ingrese a la aplicación y obtenga el perfil que le corresponda.

ID:	T27	HISTORIA:	HU8
NOMBRE:	Ingreso a la aplicación mediante un procedimiento de Login		
TIPO:	Desarrollo	FECHA INICIO:	03/11/2010
		FECHA FIN:	05/11/2010
DESCRIPCIÓN:	<p>Un usuario accede a la aplicación digitando su numero de cedula y su contraseña personal. Una vez logeado, la aplicación debe saber de que tipo es el usuario actual. Además se puede salir fácilmente del sistema(logout).</p>		

### 3.2.1.1.9 Verificación de datos de usuario

Esta historia describe el procedimiento que realiza un usuario cualquiera, una vez ingresada a la aplicación puede ver y modificar algunos de sus propios datos.

ID: HU9	USUARIO: Cliente, Usuario administrador, Superusuario	ITERACION: 1,2
NOMBRE: Verificación de datos de usuario		
DESCRIPCIÓN:  El usuario puede ver y modificar algunos de sus propios datos.		
ESTIMACIÓN: 1,5	PRIORIDAD: 1	DEPENDENCIAS: 1

Tareas de HU9: el usuario podrá visualizar sus datos y modificar algunos de ellos, como su contraseña de usuario por ejemplo.

ID: T28	HISTORIA: HU9	
NOMBRE: Mostrar Datos del Usuario Logeado		
TIPO: Desarrollo	FECHA INICIO: 08/11/2010	FECHA FIN: 08/11/2010
DESCRIPCIÓN: Se pueden ver los datos importantes del usuario actual: -Nombres, Cedula o RUC, telefono, direccion, email, actividad Existirán enlaces para ir a la modificación de datos o para ir al menú anterior.		

ID:	T29	HISTORIA:	HU9
NOMBRE: Modificación de datos de usuario logeado.			
TIPO:	Desarrollo	FECHA INICIO:	09/11/2010
		FECHA FIN:	10/11/2010
DESCRIPCIÓN: Se puede modificar todos los datos del usuario actual mostrados y adicional la contraseña. Los datos que no se pueden modificar son los de la licencias obtenidas y licencias maximas. Existirá enlaces para ver los datos del usuario actual y para volver al menú anterior.			

### 3.2.2 REDACCIÓN DE LAS HERRAMIENTAS DE LA SEGUNDA ITERACIÓN

Esta iteración corresponde a los procesos que realizará el usuario cliente, para aprovechar las funcionalidades que ofrece la aplicación. Previamente un usuario administrador creara un contrato que formalice el inicio del servicio.

#### 3.2.2.1 Módulo de Administración de Contratos del Cliente

##### 3.2.2.1.1 Administración de Contratos

Esta historia describe el procedimiento que un usuario administrador, realiza para crear un contrato de un cliente, y posteriormente anularlo o darlo de baja si es necesario.

ID:	HU10	USUARIO:	Usuario Administrador	ITERACION:	2
NOMBRE: Administración de un Contrato					
DESCRIPCIÓN: El usuario puede crear un contrato y posteriormente anular o dar de baja al mismo.					
ESTIMACIÓN:	3	PRIORIDAD:	1	DEPENDENCIAS:	1

Tareas de HU10: se debe mostrar un listado de contratos, permitir crear un nuevo contrato y permitir modificar el estado del contrato.

ID:	T30	HISTORIA:	HU11
NOMBRE:	Listado de contratos existentes		
TIPO:	Desarrollo	FECHA INICIO:	15/07/2011
		FECHA FIN:	11/08/2011
DESCRIPCIÓN:			
<p>Se debe mostrar la lista de contratos ya ingresados con informacion:</p> <ul style="list-style-type: none"> <li>-usuario/cliente</li> <li>-fecha</li> <li>-estado</li> <li>-licencias acordadas</li> <li>-licencias obtenidas</li> </ul>			

ID:	T31	HISTORIA:	HU11
NOMBRE:	Creación de un Contrato		
TIPO:	Desarrollo	FECHA INICIO:	15/07/2011
		FECHA FIN:	11/08/2011
DESCRIPCIÓN:			
<p>Se crea un contrato con los siguientes datos:</p> <ul style="list-style-type: none"> <li>-usuario/cliente</li> <li>-fecha(escoger de un calendario)</li> <li>-estado-ACTIVO por defecto</li> <li>-licencias acordadas</li> <li>-licencias obtenidas-0 por defecto</li> </ul>			

ID:	T32	HISTORIA:	HU11
NOMBRE:	Anulación o dada de baja de un contrato		
TIPO:	Desarrollo	FECHA INICIO:	15/07/2011
		FECHA FIN:	11/08/2011
DESCRIPCIÓN:			
<p>Luego de seleccionar un contrato se lo puede cambiar de estado:</p> <ul style="list-style-type: none"> <li>-estado 1 = ANULADO</li> <li>-estado 2 = DADO DE BAJA</li> </ul>			

### 3.2.2.2 Módulo de Obtención de Claves de activación de Licencias

#### 3.2.2.2.1 Obtención de claves de activación de licencias del sistema

Esta historia describe el procedimiento que un usuario cualquiera debe realizar para poder activar el sistema en un computador.

ID: HU10	USUARIO: Cliente	ITERACION: 2
NOMBRE: Obtención de claves de activación de licencias		
DESCRIPCIÓN: El usuario puede obtener claves de activación de licencia del sistema.		
ESTIMACIÓN: 3	PRIORIDAD: 1	DEPENDENCIAS: 1

Tareas de HU11: la tarea consiste en garantizar que se pueda obtener la clave que activa la licencia del sistema de una forma segura tanto para la empresa como para el cliente.

ID: T30	HISTORIA: HU10	
NOMBRE: Obtención de la clave para activación de licencias.		
TIPO: Desarrollo	FECHA INICIO: 11/11/2010	FECHA FIN: 19/11/2010
DESCRIPCIÓN: El usuario puede obtener una clave, digitando la ID del pc. Además se debe ingresar una referencia del computador donde se activará la licencia. Cuando se obtiene la clave de inmediato se crea un registro en la tabla licencias.		

### 3.2.2.3 Módulo de Mesa de Ayuda para resolución de problemas comunes

#### 3.2.2.3.1 Mesa de ayuda(ubicación de problema)

Esta historia describe la manera en como el usuario debe proceder luego de que haya ocurrido un problema con el sistema. El primer paso es ubicar el problema.

ID: HU11	USUARIO: Cliente	ITERACION: 2
NOMBRE: Mesa de ayuda - Ubicación del problema		
DESCRIPCIÓN: El usuario debe ubicar el problema que desea solucionar a partir de una lista que se le presenta.		
ESTIMACIÓN: 2,5	PRIORIDAD: 1	DEPENDENCIAS: 1

Tareas de HU12: esta tarea describe una forma sencilla y optima de ubicar un problema, facilitando la labor del cliente.

ID: T31	HISTORIA: HU11	
NOMBRE: Ubicación del problema mediante mecanismos de búsqueda de la descripción del problema		
TIPO: Desarrollo	FECHA INICIO: 22/11/2010	FECHA FIN: 25/11/2010
DESCRIPCIÓN: Se muestra una lista dinámica mostrando los problemas y el módulo donde ocurre. Además existe la opción de ubicar un item de la lista, digitando una o mas palabras que coincidan con parte de la descripción del problema o con el nombre del módulo.		

#### 3.2.2.3.2 Mesa de ayuda(visualización de la solución)

Esta historia describe el procedimiento de visualización de la solución a un problema determinado y el registro de dicha incidencia.

ID: HU12	USUARIO: Cliente	ITERACION: 2
NOMBRE: Mesa de ayuda - Solución al problema		
DESCRIPCIÓN: El usuario visualizara la(s) sugerencia(s) de solución al problema previamente seleccionada.		
ESTIMACIÓN: 1,5	PRIORIDAD: 1	DEPENDENCIAS: 1

Tareas de HU13: esta tarea indica la visualización de la solución al problema en el modulo especificado. Además se encarga de grabar la incidencia causada.

ID: T35	HISTORIA: HU13	
NOMBRE: Visualización de la recomendación o sugerencia para solucionar el problema detectado.		
TIPO: Desarrollo	FECHA INICIO: 26/11/2010	FECHA FIN: 29/11/2010
DESCRIPCIÓN: Una vez ubicado el problema, existe un enlace para poder visualizar la descripción completa de la solución a este problema. Existe la opción de aceptar la solución o buscar otra. Si se elige la opción "aceptar" se crea un registro en la tabla incidencias. -Si la sugerencia tiene otro nivel se mostrara el enlace para verificar otra posible solucion.		

### 3.2.3 REDACCION DE LAS HERRAMIENTAS DE LA TERCERA ITERACIÓN

Esta iteración corresponde a la administración de la información registrada por la aplicación. El usuario administrador deberá evaluar la información y emitir informes y recomendaciones para mejorar el servicio.

#### 3.2.3.1 Administración de Licencias obtenidas e Incidencias ocurridas.

##### 3.2.3.1.1 Licencias obtenidas:

Esta historia describe la manera en como el usuario administrador debe proceder para monitorear la información registrada en la aplicación.

ID: HU14	USUARIO: Usuario administrador	ITERACION: 3
NOMBRE: Administración de Licencias obtenidas con la aplicación		
DESCRIPCIÓN: El usuario puede visualizar el registro de las licencias obtenidas, para efectuar el monitoreo del negocio.		
ESTIMACIÓN: 1	PRIORIDAD: 1	DEPENDENCIAS:

Tarea de HU14: esta tarea indica la visualización de las licencias registradas.

ID: T36	HISTORIA: HU14	
NOMBRE: Lista de los registros de las claves devueltas por la aplicación.		
TIPO: Desarrollo	FECHA INICIO: 30/11/2010	FECHA FIN: 01/12/2010
DESCRIPCIÓN: Se muestran los datos de las licencias registradas. -Id de la licencia -Id del computador -clave -Cliente -Referencia -Fecha y hora de adquisicion		

### 3.2.3.1.3 Incidencias ocurridas

Esta historia describe la manera en como el usuario administrador debe proceder para monitorear la información registrada en la aplicación.

ID: HU15	USUARIO: Usuario administrador	ITERACION: 3
NOMBRE: Administración de Incidencias registradas por la aplicación		
DESCRIPCIÓN: El usuario puede visualizar el registro de las incidencias para efectuar el monitoreo del negocio.		
ESTIMACIÓN: 1	PRIORIDAD: 1	DEPENDENCIAS:

Tarea de HU15: esta tarea indica la visualización de las incidencias registradas.

ID: T37	HISTORIA: HU15	
NOMBRE: Lista de los registros de las incidencias(problemas-soluciones).		
TIPO: Desarrollo	FECHA INICIO: 02/12/2010	FECHA FIN: 03/12/2010
DESCRIPCIÓN: Se muestra una lista de las incidencias registradas. -ID de la incidencia -Cliente -Modulo donde ocurrio -Problema -Solución dada -Estado de revision de la incidencia		

### 3.3IMPLEMENTACION DEL PROTOTIPO DE SOFTWARE

#### 3.3.1 DISEÑO DE BASE DE DATOS

Existen herramientas como Sybase Power Designer, para realizar el diseño grafico de la base de datos. En la figura 3.1 se muestra el modelo físico de la base de datos relacional.

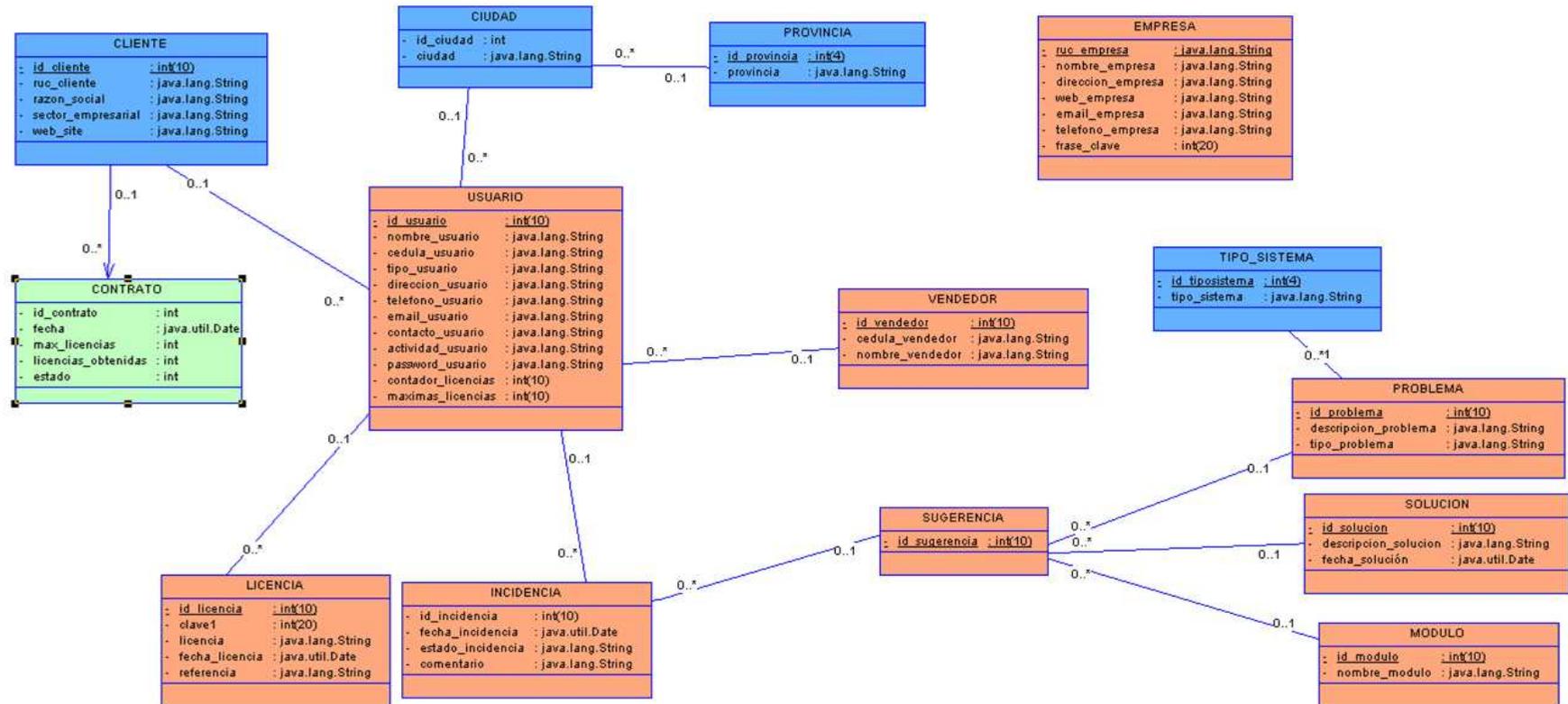
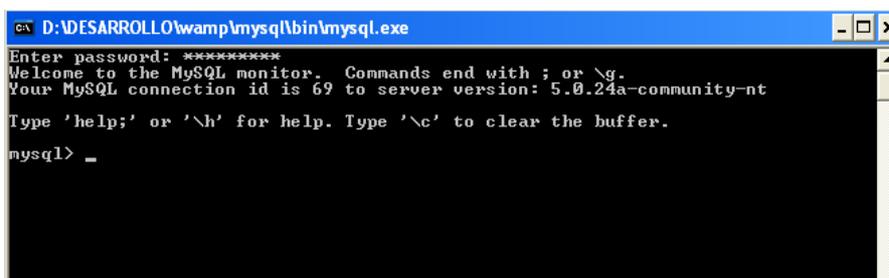


Figura 3.1 Modelo Físico de la Base de datos

### 3.3.2 HERRAMIENTAS DE SOFTWARE

#### 3.3.2.1 Servidor de Base de Datos

El servidor mysql nos provee de una consola para poder ejecutar las instrucciones sql que me permitan crear la base de datos diseñada. Por otro lado existen herramientas como PHPmyadmin, que ofrecen una interfaz grafica web amigable para trabajar con mysql. En la figura 3.2 se muestra la consola del servidor mysql donde se puede realizar configuraciones con comandos. En la figura 3.3 se muestra la interfaz grafica web de phpmyadmin donde se puede configurar mysql gráficamente.



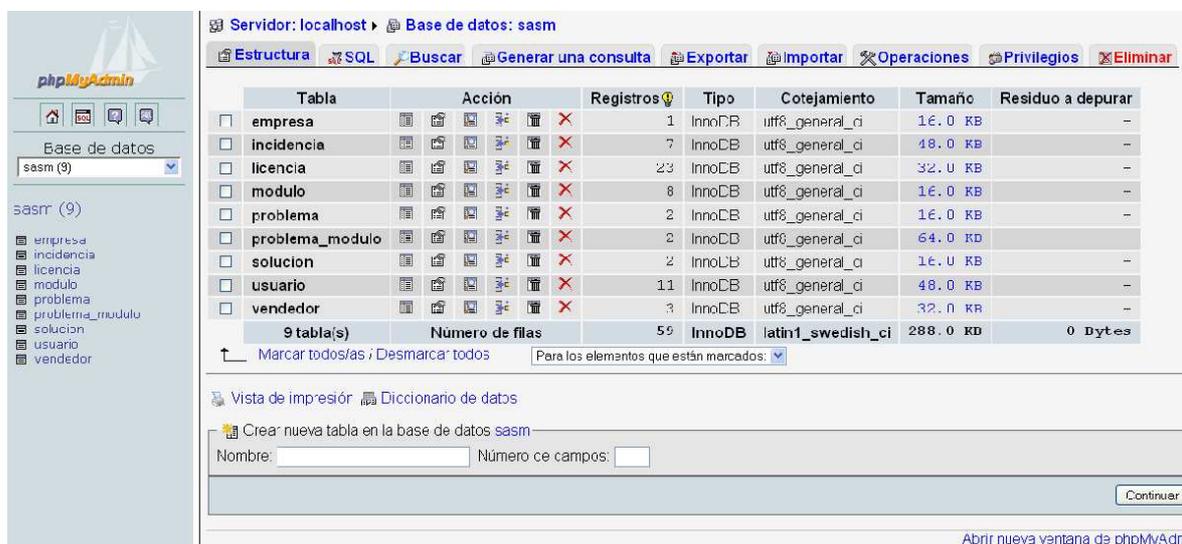
```

D:\DESARROLLO\wamp\mysql\bin\mysql.exe
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 69 to server version: 5.0.24a-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
  
```

Figura 3.2 Consola de administración Mysql.



Server: localhost > Base de datos: sasm

[Estructura](#)
[SQL](#)
[Buscar](#)
[Generar una consulta](#)
[Exportar](#)
[Importar](#)
[Operaciones](#)
[Privilegios](#)
[Eliminar](#)

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> empresa		1	InnoDB	utf8_general_ci	16.0 KB	-
<input type="checkbox"/> incidencia		7	InnoDB	utf8_general_ci	48.0 KB	-
<input type="checkbox"/> licencia		23	InnoDB	utf8_general_ci	32.0 KB	-
<input type="checkbox"/> modulo		8	InnoDB	utf8_general_ci	16.0 KB	-
<input type="checkbox"/> problema		2	InnoDB	utf8_general_ci	16.0 KB	-
<input type="checkbox"/> problema_modulo		2	InnoDB	utf8_general_ci	64.0 KB	-
<input type="checkbox"/> solucion		2	InnoDB	utf8_general_ci	16.0 KB	-
<input type="checkbox"/> usuario		11	InnoDB	utf8_general_ci	48.0 KB	-
<input type="checkbox"/> vendedor		3	InnoDB	utf8_general_ci	32.0 KB	-
<b>9 tabla(s)</b>	<b>Número de filas</b>	<b>59</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>288.0 KB</b>	<b>0 Bytes</b>

Marcar todos/as ;  Desmarca todos
 Para los elementos que están marcados:

Vista de impresión [Diccionario de datos](#)

Crear nueva tabla en la base de datos sasm

Nombre: 
 Número de campos:

Abrir nueva ventana de phpMyAdmin

Figura 3.3 Interfaz de administración phpMyAdmin

Al final de la creación de la base, debemos asegurarnos que las tablas contengan todo lo necesario para almacenar la información.

### 3.3.2.2 IDE y Servidor de Aplicaciones

Una vez estructurada la base de datos, debemos configurar la conexión de la base con la aplicación en el servidor de aplicaciones “Glassfish v3”. Para esto necesitamos saber el nombre de la base, el puerto que utiliza, el usuario y la contraseña.

Primero abrimos Netbeans 6.8 y ubicamos nuestro servidor de aplicaciones para acceder a la consola. En la figura 3.4 se muestra la interfaz principal de Netbeans, seleccionando entre las prestaciones, nuestro servidor de aplicaciones. En la figura 3.5 se muestra la consola web del servidor de aplicaciones Glassfish.

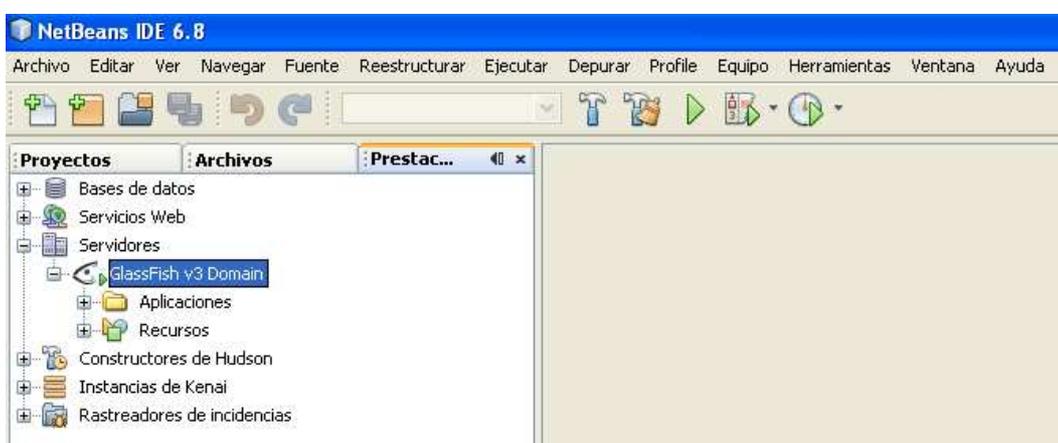


Figura 3.4 Prestaciones de Netbeans 6.8



Figura 3.5 Consola de administración de Glassfish v3

Para configurar Glassfish se siguen los siguientes pasos:

- Iniciar Glassfish
- Entrar en la consola de administración: <http://localhost:4848>
- Usuario: admin
- Contraseña: adminadmin
- Ubicarnos en JDBC-Conjunto de Conexiones: Nuevo
- Nombre: "nombredelpool"
- Tipo de recurso: "javax.sql.datasource"
- Proveedor de la base de datos: "mysql"
- Propiedades adicionales:
- Servername: localhost
- Portnumber: 3306
- Databasename: "nombre"
- User: el nombre de usuario de la base de datos
- Password: la contraseña de usuario.
- Guardar(En algunos casos también puede ser necesario especificar la ruta del driver de mysql)
- Verificar si pasa la prueba de sondeo(ping)
- Ubicarnos en JDBC-Recursos JDBC : Nuevo
- Nombre JNDI: "jdbc/\_\_nombre"
- Nombre del conjunto: "nombredelpool"
- Descripción: (opcional)
- Guardar

En la figura 3.6 se observa la configuración del pool de conexiones para nuestra fuente de datos.

Propiedades adicionales (7)	
Nombre	Valor
User	admin
portNumber	3306
databaseName	almacen
Password	adminadmin
driverClass	com.mysql.jdbc.Driver
URL	jdbc:mysql://localhost:3306/almacen
serverName	localhost

Figura 3.6 Configuración de un Pool de Conexiones

En la figura 3.7 se observa la notificación de que nuestra conexión está bien creada, mediante el sondeo o ping.

**Editar conjunto de conexiones**

Modifique un conjunto de conexión JDBC existente. Un conjunto de conexiones JDBC es un grupo de conexiones reutilizables.

**Configuración general**

Nombre JNDI: mysql\_sasm\_adminPool

Tipo de recurso: javax.sql.DataSource

Nombre de clase de la fuente de datos: com.mysql.jdbc.jdbc2.optional.MysqlDataSource

Figura 3.7 Comprobación de la Conexión

En la figura 3.8 se muestra el último paso que consiste en configurar nuestra fuente de datos a partir del pool de conexiones creado.

**Editar recurso JDBC**

Edite una fuente de datos JDBC existente.

Nombre JNDI: jdbc/sasm

Nombre de conjunto: mysql\_sasm\_adminPool

Descripción:

Estado:  Activado

**Propiedades adicionales (0)**

Nombre	Valor	Descripción
No se han encontrado elementos.		

Figura 3.8 Configuración del recurso JDBC

Ahora en el entorno de desarrollo integrado Netbeans 6.8 debemos crear la aplicación web, mapear las entidades con la ayuda de una tecnología proveedora de persistencia, hibernate en nuestro caso, generar la lógica de negocio utilizando la tecnología EJB 3.0 y las interfaces de usuario usando JSF 2.0.

Las instrucciones a seguir son las siguientes:

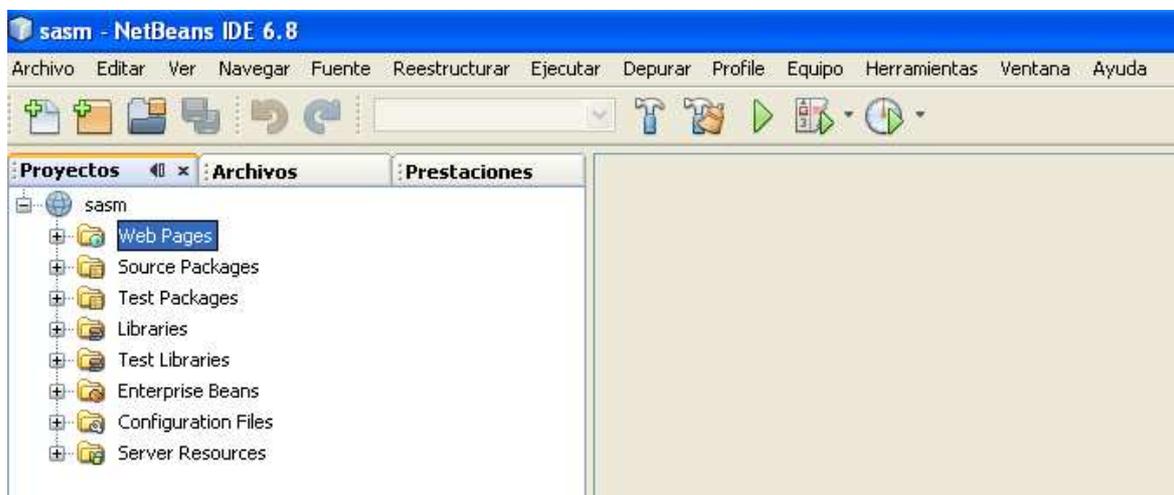
- Nuevo proyecto-JEE-Enterprise Application
- Nombrar el proyecto
- Elegir: Servidor de aplicaciones, versión de JEE, módulos a utilizar.
- Sobre el directorio "Configuration Files" del modulo ejb de nuestro proyecto vamos a crear un nuevo archivo.
- Tipo Persistencia
- Archivo- Unidad de persistencia
- Nombramos la unidad
- Elegimos el proveedor de persistencia
- Escogemos la fuente de datos(correspondiente al DataSource creado en glassfish).
- Escoger todas las tablas.
- Opciones de mapeo: usar las predeterminadas.

Y se crea el esquema de nuestra base de datos en mysql y ahora crearemos la lógica EJB para las entidades mapeadas.

- Sobre el directorio "Source packages" del modulo ejb de nuestro proyecto vamos a crear un nuevo paquete de clases .
- Tipo: Otro: Session beans for entity classes
- Elegir las tablas a partir del esquema o de la fuente de datos.
- Nombrar el paquete: (Session)
- Opciones de mapeo: usar las predeterminadas.

Por último debemos crear las paginas xhtml usando la tecnología JSF que hace uso de beans llamados de tipo controllers, managers o backingbeans.

En la figura 3.9 podemos ver como esta creada nuestra aplicación web y lista para empezar con el desarrollo. La aplicación se ha denominado SASM como abreviación de Software de Administración de Sistemas Milenio.



**Figura 3.9 Proyecto JEE Aplicación Web sasm (software para administración de sistemas Milenio3)**

### 3.3.3 INTERFACES DE USUARIO

Las interfaces web son desarrolladas sobre páginas xhtml compatibles con el framework JSF 2.0. En estas páginas podemos añadir librerías gráficas y de desarrollo importantes como richfaces<sup>39</sup> y a4j<sup>40</sup>.

En nuestra aplicación utilizaremos una plantilla para todas las páginas, esta plantilla tiene definidos los componentes que contendrán los diferentes controles de las páginas. En la plantilla se definirá un cabecera con el logo de la empresa y el nombre de la aplicación. Además luego de que un usuario se autentique se podrá visualizar su nombre. En el contenido de la página se verán algunos menús, listas y los controles de ingreso y modificación de registros.

<sup>39</sup> RichFaces es una librería de componentes visuales para JSF, posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales.

<sup>40</sup> Ajax4Jsf es una librería open source que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript.

### 3.3.3.1 Interfaz de ingreso a la aplicación

La figura 3.10 nos muestra la interfaz que permitirá autenticar al usuario y permitir su ingreso a la aplicación.

**Figura 3.10** Interfaz de ingreso a la aplicación

### 3.3.3.2 Menús de la aplicación

En la figura 3.11 podemos ver el panel principal con el acceso a los módulos que usa el cliente. En el menú horizontal se visualizarán los enlaces dependiendo del perfil de usuario. En la figura 3.11 se muestra el menú “Administración” y en la figura 3.12 esta opción esta oculta.



**Figura 3.11** Menú principal de la aplicación (perfil administrador)



**Figura 3.12 Menú principal de la aplicación (perfil cliente)**

En la figura 3.13 veremos la “zona de administradores” con los enlaces para ingresar la información base que necesita la aplicación para funcionar.



**Figura 3.13 Zona de administradores**

### 3.3.3.3 Formularios de visualización e ingreso de información

En la figura 3.14 vemos los datos de la empresa que tiene activa da la acción de ver y modificar.



The screenshot shows the Milenio3 software interface. At the top left is the logo, a globe with a stylized 'm'. The title is 'SOFTWARE PARA ADMINISTRACIÓN DE SISTEMAS 'MILENIO3''. The user is identified as 'Usuario: JUAN GUANO'. Below this is a navigation bar with links for 'Administración', 'Usuario', 'Acerca de MilenioSystem', and 'Cerrar sesión'. The main section is titled 'Datos de la Empresa' and contains a table with the following data:

Ruc	Hombre	Dirección	Web	Email	Telefono	Acción
0987873812	MILENIOCORP DG.	JORGE Y DROMP Y GASPAR DE VILLAROEEL	WWW.MILENIOSISTEMAS.COM	INFO@MILENIOSISTEMAS.COM	(02)-26042344	<a href="#">Ver</a> <a href="#">Modificar</a>

Below the table is a link 'Volver al Menú' and a footer 'CORPORACION MILENIOSYSTEM S.A. 2011'.

Figura 3.14 Visualización de datos de la empresa

En la figura 3.15 por ejemplo podemos ver la lista de usuarios mostrados en una lista con paginación. Cada registro tiene activadas las acciones ver, modificar y eliminar. Además se permite crear nuevos usuarios.



The screenshot shows the Milenio3 software interface displaying a list of users. The title is 'SOFTWARE PARA ADMINISTRACIÓN DE SISTEMAS 'MILENIO3''. The user is identified as 'Usuario:'. Below this is a navigation bar. The main section is titled 'Lista de Usuarios' and contains a table with the following data:

Id	Nombre	Cédula	Tipo	Dirección	Teléfono	Email	Contacto	Actividad	Contraseña	#Licencias obtenidas	#Max de licencias	Vendedor	Acción
1	DAVID GUANO VISCARRA	1714576574	CLIENTE	CARCELEN JUAN DE GUADALUPE N79 -13	2473275	david.guano@hotmail.com	EPN-redess	Ingeniero en Redes	david	7	7	JUAN FRANCISCO GUANO	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
2	ACEVEDO EDSON ALEXANDER	1720156411	CLIENTE	ISLA ISABELA 653 Y TOMAS DE BERLANGA	2263-470, dw	edsontrias@hotmail.com	Fanny Verdezoto	Imprenta	edson2011	4	5	MILENIOCORP OFICINA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
3	JUAN GUANO	1724472517	ADMIN	CARCELEN Juan de guadalupe n79-13	084775120	juanelfrancisco@hotmail.com	CCC	estudiante	david	12	12	MILENIOCORP OFICINA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>

Below the table is a link 'Crear Nuevo Usuario', a link 'Volver al Menú', and a footer 'CORPORACION MILENIOSYSTEM S.A. 2011'.

Figura 3.15 Lista de Usuarios de la aplicación

### 3.3.3.4 Interfaz del módulo de Obtención de claves.

En la figura 3.16 se muestra la interfaz que me permite obtener una clave de activación de licencia a partir de un id generado en el computador donde se instala el sistema Milenio3.



Figura 3.16 Interfaz para obtención de claves

La figura 3.17 muestra la interfaz inmediatamente después de aceptar la obtención de clave.



Figura 3.17 Interfaz para obtención de claves, visualización de clave

Además este modulo tiene la característica de limitar el número de claves obtenidas dependiendo del usuario. la figura 3.18 muestra la interfaz luego de haber obtenido el máximo de claves permitidas.

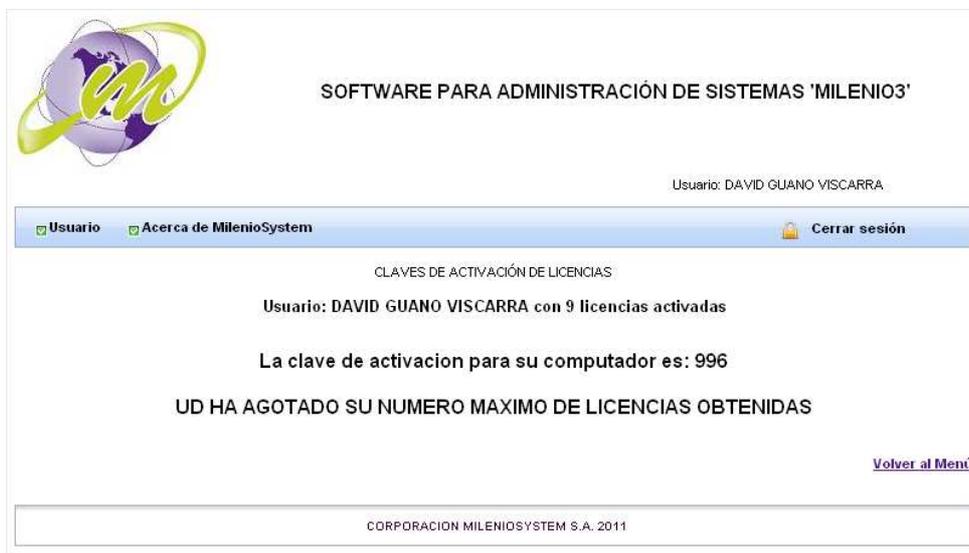


Figura 3.18 Interfaz para obtención de claves, claves agotadas

### 3.3.3.5 Interfaz del módulo de Mesa de Ayuda (Helpdesk)

La figura 3.19 muestra la interfaz de la mesa de ayuda que ofrece la aplicación, el primer paso es ubicar el problema.



Figura 3.19 Mesa de ayuda, ubicación del problema

La figura 3.20 muestra la aplicación luego de haber seleccionado un problema de la lista.



**Figura 3.20** Mesa de ayuda, sugerencia de solución

Para garantizar el registro correcto de información, antes de generar la incidencia le pedimos confirmación al usuario para no guardar datos erróneos dificultarían la labor del administrador de incidencias, tal como muestra la figura 3.21.



**Figura 3.21** Confirmación de la incidencia

### 3.3.3.6 Interfaces para evaluación de información ingresada por el cliente

En la figura 3.22 y 3.23 se muestra la interfaz para monitoreo de las incidencias registradas por los clientes y las claves obtenidas.



SOFTWARE PARA ADMINISTRACIÓN DE SISTEMAS 'MILENIO3'

Usuario: JUAN GUANO

Administración Usuario Acerca de MilenioSystem Cerrar sesión

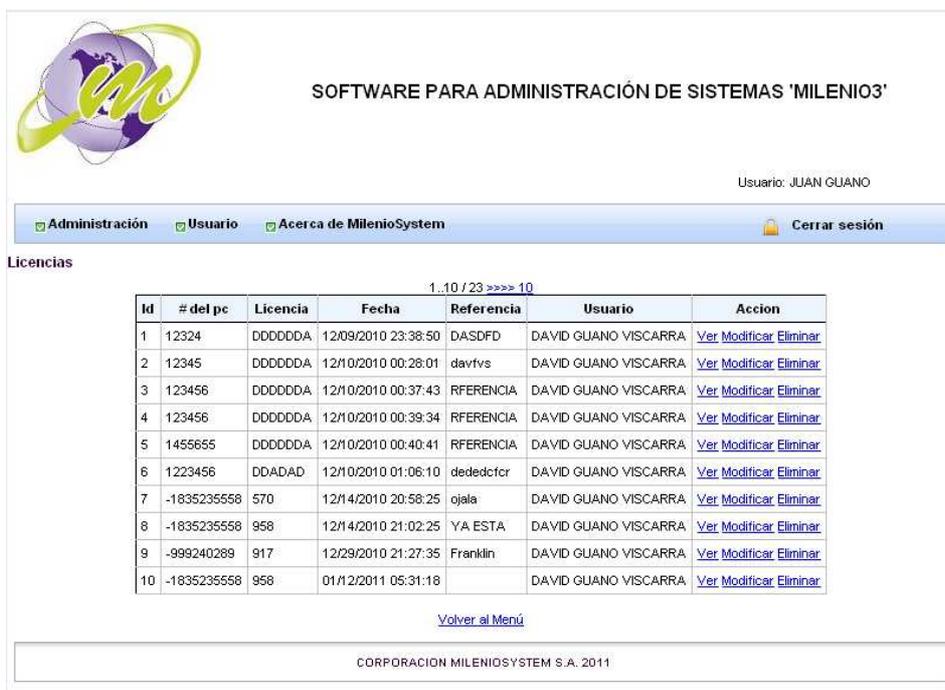
Incidencias 1..7 / 7

Id	Fecha	Estado	Comentario	Problema	Usuario	Accion
1	12/29/2010 05:31:02	ACTIVA	ojalaf	Mensaje: la base de datos esta dañada se debe reparar para volver utilizarla.	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
2	12/29/2010 05:31:53	ACTIVA	ojalaf	Mensaje: esta es una copia no valida del sistema. contactese con la empresa para obtener su licencia.	ACEVEDO EDSON ALEXANDER	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
3	12/29/2010 21:30:57	ACTIVA	ojalaf	Mensaje: la base de datos esta dañada se debe reparar para volver utilizarla.	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
4	12/31/2010 04:35:39	ACTIVA	ojalaf	Mensaje: la base de datos esta dañada se debe reparar para volver utilizarla.	JUAN GUANO	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
5	01/11/2011 04:20:35	ACTIVA	ojalaf	Mensaje: la base de datos esta dañada se debe reparar para volver utilizarla.	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
6	01/15/2011 02:53:05	ACTIVA	ojalaf	Mensaje: la base de datos esta dañada se debe reparar para volver utilizarla.	JUAN GUANO	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>

[Volver al Menú](#)

CORPORACION MILENIOSYSTEM S.A. 2011

Figura 3.22 Incidencias registradas



SOFTWARE PARA ADMINISTRACIÓN DE SISTEMAS 'MILENIO3'

Usuario: JUAN GUANO

Administración Usuario Acerca de MilenioSystem Cerrar sesión

Licencias 1..10 / 23 >>>> 10

Id	# del pc	Licencia	Fecha	Referencia	Usuario	Accion
1	12324	DDDDDDA	12/09/2010 23:38:50	DASDFD	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
2	12345	DDDDDDA	12/10/2010 00:28:01	davfvs	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
3	123456	DDDDDDA	12/10/2010 00:37:43	RREFERENCIA	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
4	123456	DDDDDDA	12/10/2010 00:39:34	RREFERENCIA	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
5	1455655	DDDDDDA	12/10/2010 00:40:41	RREFERENCIA	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
6	1223456	DDADAD	12/10/2010 01:06:10	dededctcr	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
7	-1835235558	570	12/14/2010 20:58:25	ojala	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
8	-1835235558	958	12/14/2010 21:02:25	YA ESTA	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
9	-999240289	917	12/29/2010 21:27:35	Franklin	DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>
10	-1835235558	958	01/12/2011 05:31:18		DAVID GUANO VISCARRA	<a href="#">Ver</a> <a href="#">Modificar</a> <a href="#">Eliminar</a>

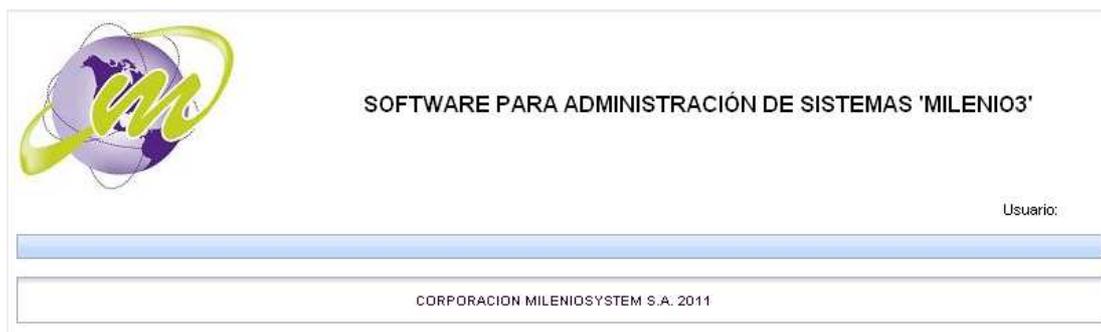
[Volver al Menú](#)

CORPORACION MILENIOSYSTEM S.A. 2011

Figura 3.23 Licencias activadas

### 3.3.3.7 Seguridad de la aplicación

Gracias a la propiedad “rendered” se puede controlar la visualización de los componentes de las páginas. En nuestro caso controlaremos que se visualicen los componentes si existe un usuario autenticado, caso contrario se vería como en la figura 3.24.



**Figura 3.24** Interfaz de la aplicación cuando no existe un usuario registrado

Otro control importante es la autenticación de usuario, que para ingresar a la aplicación debe ingresar su número de cédula y su contraseña. Si este proceso fallá se muestra la interfaz como en la figura 3.25.



**Figura 3.25** Fallo de la autenticación en la aplicación

### 3.3.4 CODIGO

La aplicación está formada de una serie de archivos que almacenan valiosa información de configuración y código fuente.

#### Web

- faces-config.xml: indica la ubicación de los archivos de propiedades y permite configurar la navegación web.
- sun-web.xml: indica el contexto desde donde parte la aplicación
- web.xml: proporciona la configuración y el despliegue de información para los componentes web que conforman una aplicación.
- Archivos xhtml: contienen la presentación web de la aplicación.
- Archivos css: almacenan estilos web para ser utilizados en las páginas xhtml.

#### Paquetes de fuente

- Clases Entidad: representan a una tabla de base de datos, mapeada y transformada a una entidad java.
- Beans de Sesión sin estado: son objetos distribuidos que carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. No se garantiza que los contenidos de las variables de instancia se conserven entre llamadas al método.
- Beans Controladores: (Managed o Backing Beans). Son clases java que se asocian a los formularios JSF. Son beans gestionados por el controlador JSF. este se encarga de su construcción y destrucción automáticas cuando es necesario.
- Archivo de Propiedades: almacenan cadenas de texto que se muestran cada que se especifique su uso desde un bean controlador.

El anexo D muestra lo más importante de la implementación de estos códigos.

### **3.4 PRUEBAS DEL DESARROLLO DEL PROTOTIPO DE SOFTWARE**

Cada prueba está asociada a una historia de usuario diferente, y en ellos se describen los posibles modos de utilización de la aplicación en cada una de estas historias de usuario. Aquí se indica la respuesta que tiene la aplicación en la utilización de cada una de estas funcionalidades, así como los posibles mensajes de error, información o de aceptación que emite la aplicación cuando se utiliza dicha funcionalidad.

#### **3.4.1 HU1: ADMINISTRACION DE LA EMPRESA**

##### **3.4.1.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Datos de la Empresa”. Ahí visualizara los datos importantes de la empresa y podrá modificarlos si desea.

##### **3.4.1.2 Condiciones de Ejecución**

Que el usuario sea administrador y que exista en base de datos el registro inicial de la empresa.

##### **3.4.1.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Datos de la empresa”.
- Se muestra la información de la empresa.
- Se presiona la opción de “Modificar”
- Aparecen los controles con la información a modificar.
- Luego de modificar la información se presiona la opción “Guardar”.

- Si las modificaciones no infringen una regla de validación de la información la empresa se habrá modificado.

#### **3.4.1.4 Resultado Esperado**

Los datos de la empresa serán modificados.

### **3.4.2 HU2: ADMINISTRACION DE LOS VENDEDORES**

#### **3.4.2.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar vendedores”. Ahí se visualizara una lista de vendedores, con la opción de ver, modificar o eliminar un vendedor, o crear uno nuevo.

#### **3.4.2.2 Condiciones de Ejecución**

Que el usuario sea administrador y para ver, modificar o eliminar debe existir registros de vendedores.

#### **3.4.2.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar Vendedores”.
- Se muestra una lista de Vendedores registrados.
- Se puede optar por la opción de ver, modificar o eliminar un vendedor de la lista o crear un nuevo vendedor.
- Luego de modificar un vendedor o ingresar uno nuevo se presiona la opción “Guardar”.

- Si las modificaciones o la creación, no infringen una regla de validación de la información, el vendedor se habrá modificado o creado.

#### **3.4.2.4 Resultado Esperado**

Se creará un vendedor nuevo o se modificará o eliminará uno existente.

### **3.4.3 HU3: ADMINISTRACION DE USUARIOS**

#### **3.4.3.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar usuarios”. Ahí se visualizara una lista de usuarios, con la opción de ver, modificar o eliminar un usuario, o crear uno nuevo.

#### **3.4.3.2 Condiciones de Ejecución**

Que el usuario sea administrador y que para ver, modificar o eliminar un usuario deben existir registros. Además debe existir al menos un vendedor registrado.

#### **3.4.3.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar usuarios”.
- Se muestra una lista de usuarios de la aplicación.
- Se puede optar por la opción de ver, modificar o eliminar un usuario de la lista o crear un nuevo usuario
- Se toma mucha atención en el tipo de vendedor (“ADMIN” o “CLIENTE”).
- A los usuarios tipo cliente se le asigna el vendedor que hizo la negociación.

- Luego de modificar un usuario o ingresar uno nuevo se presiona la opción “Guardar”.
- Si las modificaciones o la creación, no infringen una regla de validación de la información, el usuario se habrá modificado o creado.

#### **3.4.3.4 Resultado Esperado**

Un usuario existente se habrá modificado o eliminado, o uno nuevo se habrá creado.

### **3.4.4 HU4: ADMINISTRACION DE PROBLEMAS DEL SISTEMA**

#### **3.4.4.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar Problemas”. Ahí se visualizara un catálogo de problemas comunes en los sistemas Milenio3, con la opción de ver, modificar o eliminar un problema, o crear uno nuevo.

#### **3.4.4.2 Condiciones de Ejecución**

Que el usuario sea administrador y para ver, modificar o eliminar un problema, exista registros.

#### **3.4.4.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar problemas”.
- Se muestra la lista de problemas registrados.

- Se puede optar por la opción de ver, modificar o eliminar un problema de la lista o crear un nuevo problema.
- Luego de modificar un problema o ingresar uno nuevo se presiona la opción “Guardar”.
- Si las modificaciones o la creación, no infringen una regla de validación de la información, el problema se habrá modificado o creado.

#### **3.4.4.4 Resultado Esperado**

Un problema existente se habrá modificado o eliminado, o un problema nuevo se habrá creado.

### **3.4.5 HU5: ADMINISTRACIÓN DE LOS MÓDULOS DEL SISTEMA**

#### **3.4.5.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar módulos”. Ahí se visualizara una lista de módulos que componen el sistema Milenio3, con la opción de ver, modificar o eliminar un módulo, o crear uno nuevo.

#### **3.4.5.2 Condiciones de Ejecución**

Que el usuario sea administrador y para ver, modificar o eliminar un módulo, exista registros.

#### **3.4.5.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar Módulos”.

- Se muestra la lista de módulos registrados.
- Se puede optar por la opción de ver, modificar o eliminar un módulo de la lista o crear un nuevo módulo.
- Luego de modificar un módulo o ingresar uno nuevo se presiona la opción “Guardar”.
- Si las modificaciones o la creación, no infringen una regla de validación de la información, el módulo se habrá modificado o creado.

#### **3.4.5.4 Resultado Esperado**

Un módulo existente se habrá modificado o eliminado, o un módulo nuevo se habrá creado.

### **3.4.6 HU6: ADMINISTRACION DE SOLUCIONES A LOS PROBLEMAS**

#### **3.4.6.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar soluciones”. Ahí se visualizara una lista de soluciones con la opción de ver, modificar o eliminar una solución, o crear una solución.

#### **3.4.6.2 Condiciones de Ejecución**

Que el usuario sea administrador y para ver, modificar o eliminar una solución, exista registros.

#### **3.4.6.3 Entrada**

- El usuario ingresa su cédula y su password.

- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar Soluciones”.
- Se muestra la lista de soluciones registradas.
- Se puede optar por la opción de ver, modificar o eliminar una solución de la lista o crear una nueva solución.
- Luego de modificar una solución o ingresar una nueva se presiona la opción “Guardar”.
- Si las modificaciones o la creación, no infringen una regla de validación de la información, la solución se habrá modificado o creado.

#### **3.4.6.4 Resultado Esperado**

Una solución existente se habrá modificado o eliminado, o una solución nueva se habrá creado.

### **3.4.7 HU7: ADMINISTRACION DE SUGERENCIAS DE SOLUCIÓN**

#### **3.4.7.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar Sugerencias de solución”. Ahí se visualizara una catalogo de registros los cuales enlazan un problema en un modulo con una solución, con la opción de ver, modificar o eliminar una sugerencia, o crear una nueva.

#### **3.4.7.2 Condiciones de Ejecución**

Que el usuario sea administrador y para ver, modificar o eliminar una sugerencia, exista registros. Además deben existir registros de Módulos, problemas, y soluciones.

### **3.4.7.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar sugerencias”.
- Se muestra la lista de sugerencias registradas.
- Se puede optar por la opción de ver, modificar o eliminar una sugerencia de la lista o crear una nueva sugerencia
- Luego de modificar una sugerencia solución o formar una nueva, se presiona la opción “Guardar”.
- Si las modificaciones o la creación, no infringen una regla de validación de la información, la sugerencia se habrá modificado o creado.

### **3.4.7.4 Resultado Esperado**

Una sugerencia existente se habrá modificado o eliminado, o una sugerencia nueva se habrá creado.

## **3.4.8 HU8: INGRESO A LA APLICACIÓN**

### **3.4.8.1 Descripción**

Un usuario cualquiera debe ingresar a la aplicación, digitando su cédula y su contraseña. La aplicación debe detectar el tipo de usuario que ha ingresado.

### **3.4.8.2 Condiciones de Ejecución**

Que el usuario haya sido creado previamente por un administrador.

### **3.4.8.3 Entrada**

- El usuario ingresa su cédula y su password.

- Se presiona el botón aceptar para ingresar a la aplicación.
- Si la autenticación falla, el usuario no entrara a la aplicación y deberá volver a digitar los datos correctos.

#### **3.4.8.4 Resultado Esperado**

El usuario ingresará a la aplicación. Solo si es de tipo Administrador tendrá acceso al menú de “Administración”.

### **3.4.9 HU9: VERIFICACIÓN DE INFORMACIÓN DE USUARIO**

#### **3.4.9.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Usuario” y puede verificar su información de usuario y si lo desea puede modificar algunos de ellos.

#### **3.4.9.2 Condiciones de Ejecución**

Que el usuario haya ingresado correctamente a la aplicación.

#### **3.4.9.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Usuarios selecciona “Ver datos de usuario” .
- Se muestra la información de la empresa.
- Se muestra la información importante del usuario actual.
- Si desea modificar la información se presiona la opción “modificar”.
- Luego de modificar la información se presiona la opción “Guardar”.
- Si las modificaciones no infringen una regla de validación de la información el usuario actual se habrá modificado.

#### **3.4.9.4 Resultado Esperado**

Los datos del usuario actual serán verificados y modificados si lo desea.

### **3.4.10 HU10: ADMINISTRACIÓN DE CONTRATOS**

#### **3.4.10.1 Descripción**

El usuario administrador luego de haber creado el registro de cliente, procede a crear el contrato que especifica el número máximo de licencias adquiridas a la empresa.

#### **3.4.10.2 Condiciones de Ejecución**

Que el usuario haya ingresado correctamente en la aplicación y que exista el registro de usuario cliente.

#### **3.4.10.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú usuario Milenio presiona la opción “nuevo contrato”.
- Seleccione el cliente, escoja la fecha, digite el número máximo de licencias.
- Se puede ver la lista de contratos y elegir la opción anular o dar de baja un contrato.

#### **3.4.10.4 Resultado Esperado**

Un contrato nuevo se habrá creado, o un contrato existente habrá cambiado de estado

### **3.4.11 HU11: OBTENCIÓN DE CLAVES DE ACTIVACIÓN DE LICENCIAS**

#### **3.4.11.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará la opción “Licencias” y ahí se visualizara los controles necesarios para obtener una clave de activación del sistema en un computador específico.

#### **3.4.11.2 Condiciones de Ejecución**

Que el usuario haya ingresado correctamente en la aplicación y que no haya superado su límite de claves a obtener.

#### **3.4.11.3 Entrada**

- El usuario ingresa su cédula y su password.
- Se presiona la opción “Licencias”.
- Se digita el identificador del computador en donde se acaba de instalar el sistema.
- Se ingresa una referencia del computador donde se va a activar la licencia.
- Se presiona el botón “obtener clave”.
- Si el usuario acepta el mensaje de confirmación se muestra la clave de activación.

#### **3.4.11.4 Resultado Esperado**

Se obtendrá un clave de activación de licencias.

### **3.4.12 HU12: MESA DE AYUDA, UBICACIÓN DEL PROBLEMA**

#### **3.4.12.1 Descripción**

El usuario luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará la opción “Mesa de ayuda”, y se podrá visualizar una lista de problemas que suceden con frecuencia en el sistema Milenio3 con la finalidad que el usuario ubique el problema que le ha sucedido.

#### **3.4.12.2 Condiciones de Ejecución**

Que existan registros de problemas en la aplicación.

#### **3.4.12.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú principal se presiona el botón “Mesa de ayuda”
- Se muestra la lista de problemas que pueden ocurrir con su descripción.
- Se debe ubicar el problema, viendo en la lista o utilizando la búsqueda por módulo o problema.

#### **3.4.12.4 Resultado Esperado**

El problema será ubicado en la lista.

### **3.4.13 HU13: MESA DE AYUDA, SOLUCIÓN AL PROBLEMA**

#### **3.4.13.1 Descripción**

El usuario luego de ubicar el problema puede visualizar una instrucción que le ayude a solucionar el problema lo más pronto posible.

### **3.4.13.2 Condiciones de Ejecución**

Que exista registros de soluciones y sugerencias en la aplicación.

### **3.4.13.3 Entrada**

- Luego de ubicar el problema, se presiona la opción “Ver sugerencia”.
- Se muestra la instrucción que se recomienda seguir para resolver el problema.
- Si el usuario lo decide puede regresar a buscar otro problema o puede aceptar la sugerencia presionando el botón “Aceptar sugerencia”.

### **3.4.13.4 Resultado Esperado**

La solución efectivamente ayudo a resolver el problema.

## **3.4.14 HU14: ADMINISTRACION DE LICENCIAS OBTENIDAS**

### **3.4.14.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar Licencias”. Y ahí puede evaluar periódicamente la información que se ha ingresado por los usuarios.

### **3.4.14.2 Condiciones de Ejecución**

Que el usuario sea administrador y que existan registros de licencias obtenidas.

### **3.4.14.3 Entrada**

- El usuario ingresa su cédula y su password.

- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar Licencias”.
- Se muestra la lista de claves de licencias obtenidas.
- Se puede ver los datos de cada uno de los registros para controlar que los usuarios estén usando el sistema adecuadamente.

#### **3.4.14.4 Resultado Esperado**

La información de claves obtenidas servirá para control de los clientes.

### **3.4.15 HU15: ADMINISTRACION INCIDENCIAS REGISTRADAS**

#### **3.4.15.1 Descripción**

El usuario administrador luego de haber ingresado a la aplicación pasando por el proceso de login, seleccionará en el menú: “Zona de administradores” y luego ingresará a “Administrar Incidencias”. Y ahí puede evaluar periódicamente la información que se ha ingresado por los usuarios.

#### **3.4.15.2 Condiciones de Ejecución**

Que el usuario sea administrador y que existan incidencias registradas.

#### **3.4.15.3 Entrada**

- El usuario ingresa su cédula y su password.
- En el menú de Administración, selecciona “Zona Administradores” y luego en la opción “Administrar Incidencias”.
- Se muestra la lista de incidencias.
- Se puede “ver” cada una para evaluar los resultados y proponer mejoras.

#### **3.4.15.4 Resultado Esperado**

La información de incidencias será evaluada.

## 3.5 COSTO Y RESULTADOS DE LA APLICACION

### 3.5.1 COSTO DEL DESARROLLO LA APLICACIÓN

El costo de un software depende de la suma y cálculo de varios parámetros que se deben tomar en cuenta.

- Costo de aprendizaje: es igual al tiempo de aprendizaje o investigación por el valor de la hora del aprendizaje. Se incluye el análisis personalizado de los requerimientos o el levantamiento de información.  
El valor de la hora del aprendizaje lo define el desarrollador, siendo más alto que el valor de la hora de desarrollo.
- Costo de desarrollo: es igual al tiempo de desarrollo por el valor de la hora de desarrollo.  
El valor de la hora de desarrollo lo define el desarrollador. Un criterio válido es que el valor de la hora no puede ser mayor a 1/30 de un sueldo básico. Y no puede ser menor a 1/60 de un sueldo básico.
- Costos de Licencia para desarrollo: si se trabaja sobre una tecnología licenciada, se suma el valor de esta licencia.
- Costos de software para desarrollo: si se trabaja con herramientas de software licenciado, se suman estos valores.
- Costo de software de instalación: si se necesita de un software especial para la instalación.
- Costo de presentación de la aplicación: el costo de los materiales para comercializar la aplicación. CD, manuales, cajas.
- Costo de la instalación por equipo: si la instalación es compleja y requiere de un costo adicional.
- Costo de arrendamiento de sitio web: si se ofrece el servicio de hosting de la aplicación.

Una vez sumados estos costos, se necesita definir la forma en cómo se quiere recuperar la inversión, para poder obtener el costo individual de la aplicación.

### 3.5.1.1 Costo de la Aplicación

Definimos los valores por hora, para este desarrollo y en la tabla 3.4 observamos el cálculo de la aplicación.

Valor de la hora de aprendizaje: \$8

Valor de la hora de desarrollo: \$5

Parámetro	Valor x hora	Cantidad	Total
Aprendizaje	8	160 horas	1280
Desarrollo	5	320 horas	1600
Licencia desarrollo		0	0
Software desarrollo		0	0
Software instalación		0	0
Presentación		20	20
Instalación		20	20
Arrendamiento		0	0
<b>TOTAL</b>			2920

Tabla 3.4 Cálculo del costo de la aplicación.

El valor total se acerca a los \$3000 dólares. Pero hay que tomar en cuenta que casi el 44% del valor total corresponde a la etapa de investigación e aprendizaje de nuevas tecnologías y al levantamiento personalizado de los requerimientos de la empresa. Por lo tanto un próximo desarrollo sobre la misma tecnología y con las mismas herramientas usadas, presentará una disminución en costo.

### 3.5.2 RESULTADOS DE LA APLICACION

Para obtener una visión clara de los resultados obtenidos con la implementación del software en la empresa, vamos a utilizar un cuestionario dirigido a algunos usuarios. Este cuestionario debe ser fácil de completar, fácil de puntuar y permitir establecer comparaciones entre opiniones.

Utilizaremos un cuestionario de tipo SUS<sup>41</sup>. La escala SUS se utiliza generalmente después de que un usuario ha tenido la oportunidad de utilizar un sistema pero antes de que cualquier informe o discusión tenga lugar. La escala SUS es una escala de estilo Likert<sup>42</sup> que genera un único número, representando una medida compuesta de la usabilidad del sistema sometido a estudio.

“Para calcular la puntuación del SUS, hay que sumar primero las contribuciones de cada punto. Cada pregunta representa un punto y este cuestionario se basa generalmente en 10 preguntas. La contribución de cada punto valdrá entre 0 y 4. Para los puntos 1, 3, 5, 7 y 9, la contribución será la posición de la escala menos 1. Para los puntos 2, 4, 6, 8 y 10, la contribución será 5 menos la posición en la escala. Se multiplica la suma de los resultados por 2.5 para obtener el valor global del SUS. El resultado estará entre 0 y 100”.<sup>[13]</sup>

En completo desacuerdo -----1

Completamente de acuerdo-----5

El cuestionario contiene las siguientes preguntas:

1. Me gusta usar con frecuencia esta aplicación web.
2. Encontré la aplicación innecesariamente complejo.
3. Pienso que es fácil utilizar esta aplicación.
4. Creo que necesitaría del apoyo de un experto para usar esta aplicación.
5. Encontré los 2 módulos de la aplicación (distribución de licencias y helpdesk) bastante bien integradas.
6. Pensé que había demasiada inconsistencia en la información de provee la aplicación.

---

<sup>41</sup> System Usability Scale, fue desarrollado en 1986 como parte de la introducción de la ingeniería de usabilidad a los sistemas de oficina de Digital Equipment Co. Ltd.

<sup>42</sup> La escala de tipo Likert es comúnmente utilizada en cuestionarios, cuando respondemos a un elemento de un cuestionario elaborado con la técnica de Likert, lo hacemos especificando el nivel de acuerdo o desacuerdo con una declaración (elemento, ítem o reactivo)

7. Imagino que la mayoría de las personas aprenderían muy rápidamente a utilizar esta aplicación.

8. Encontré la aplicación muy grande al recorrerlo

9. Me sentí muy confiado en el manejo de la aplicación.

10. Necesito aprender muchas cosas antes de manejar la aplicación.

La tabla 3.5 muestra los resultados de la encuesta realizada a 4 diferentes usuarios de la aplicación.

	Usuario1	SUS	Usuario2	SUS	Usuario3	SUS	Usuario4	SUS
<b>Pregunta 1</b>	5	4	4	3	5	4	3	2
<b>Pregunta 2</b>	1	4	3	2	1	4	1	4
<b>Pregunta 3</b>	4	3	3	2	5	4	5	4
<b>Pregunta 4</b>	3	2	3	2	1	4	1	4
<b>Pregunta 5</b>	3	2	5	4	5	4	5	4
<b>Pregunta 6</b>	1	4	1	4	1	4	2	3
<b>Pregunta 7</b>	5	4	5	4	5	4	3	2
<b>Pregunta 8</b>	4	1	3	2	1	4	4	1
<b>Pregunta 9</b>	5	4	4	3	5	4	4	3
<b>Pregunta 10</b>	4	1	3	2	1	4	3	2
<b>Resultado SUS</b>		29		28		40		29
<b>Porcentaje Satisfactorio</b>		<b>72,5</b>		<b>70</b>		<b>100</b>		<b>72,5</b>

Tabla 3.5 Evaluación del cuestionario SUS para la aplicación implementada.

Obteniendo el promedio de esta evaluación se obtiene un 78,75% de aprobación del software implementado por parte de los usuarios encuestados.

## CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 CONCLUSIONES

- El software desarrollado e implementado tiene casi un 80% de aprobación por parte de los usuarios. La cantidad de denuncias de plagio o activación de licencias indebidamente, se han reducido a 0. Existe un ahorro considerable en tiempo y recursos a la hora de brindar soporte a problemas con los sistemas, todo gracias al uso de la aplicación.
- A pesar de que exista una buena capacitación a los usuarios para el manejo de un sistema, siempre queda abierto la posibilidad de que se comentan errores y que se convierten en un problema tanto para el cliente como para la empresa encargada de brindar soporte. Un error causado por mal manejo del sistema implica pérdida de tiempo para el cliente que necesita usar el sistema e implica asignación de tiempo y recursos por parte del proveedor del servicio para resolver el problema.
- En la actualidad las aplicaciones web brindan una solución fiable para interactuar entre el cliente y el proveedor de un servicio determinado. La información está expuesta a menos riesgos de perderse o retrasarse en llegar cuando se envíe de un lado al otro. Y en nuestro caso informa al proveedor de manera más eficiente que otros medios.
- La aplicación desarrollada propone acabar con los inconvenientes que pueden presentarse al momento de seleccionar la mejor forma de otorgar claves de activación de licencias, y cuando se ofrece un servicio de soporte y atención inmediata al cliente.

- Mediante nuestra aplicación web la información generada por los usuarios involucrados con esta línea del negocio estará centralizada y permitirá un fácil control y monitoreo.
- A pesar de que se permite que el mismo usuario cliente pueda obtener una clave de activación de licencia es importante llevar un control, limitando por parte del administrador, el número máximo de licencias que se podrá obtener. En nuestro caso este número es controlado directamente por un contrato que define el cupo luego de que se haya llegado a un acuerdo económico de ambas partes.
- Los términos ServiceDesk o Helpdesk implican un avance de la forma en cómo se brinda soporte técnico y asistencia a un cliente. En la actualidad muchas empresas desconocen estos términos y otras en cambio tienen un concepto equivocado. Por eso es importante empezar a trabajar con estos términos y su verdadero rol en los procesos tecnológicos.
- Para un Helpdesk o Mesa de ayuda es indispensable definir la información de una forma clara y sencilla. Identificar cual es el problema, donde ocurrió el problema y cuál es la sugerencia que tenga más probabilidad de llegar a la solución.
- Utilizar la tecnología JEE6 con EJB 3.0 y JSF 2.0 brinda flexibilidad en el desarrollo gracias a una lógica sencilla y fácil de implementar. Además da la posibilidad de trabajar con buenas tecnologías de visualización de páginas web.
- Glassfish v.3 es un servidor de aplicaciones robusto que es perfectamente compatible con entornos de desarrollo como Netbeans y trabaja sin problemas en conjunto con gestores de bases de datos como Mysql y postgres, permitiendo desarrollar aplicaciones de todo tipo.

- Trabajar con páginas XHTML brinda flexibilidad a la hora de diseñar el formato web que se verá en el navegador, ya que permite insertar hojas de estilo css, librerías como richfaces que son una evolución en la forma de presentar componentes visuales que dinamizan una página web.
- La metodología XP brinda flexibilidad al desarrollo, propone requerimientos reales mediante las historias de usuario pero a pesar de que permite la modificación sobre la marcha de algunas directivas para el desarrollo, se debe tener cuidado en no desperdiciar tiempo y esfuerzo en cumplir requerimientos que no estén claros por parte del usuario final.
- Las tareas que cubren una historia de usuario son parte importante en el desarrollo ya que son las que definen instrucciones específicas que deben seguir los programadores y que además permiten evaluar el avance de un proyecto.
- Las pruebas de aceptación de desarrollo, nos ayuda a verificar el funcionamiento de la aplicación y que la programación no tenga errores pero además puede retroalimentar el desarrollo dando pequeños pero nuevos requerimientos a implementar con la finalidad de satisfacer al usuario final.

## **4.2 RECOMENDACIONES**

- Antes de elegir una plataforma o tecnología de desarrollo, se recomienda analizar detenidamente requerimientos como el equipo donde se va a implementar, base de datos con las que va a trabajar, número de usuarios

que va a soportar, etc. con la finalidad de tener más criterios de elección y no usar una tecnología muy simple o muy compleja para nuestro caso.

- Durante el desarrollo de una aplicación y específicamente en el proceso de levantamiento de requerimientos se recomienda interactuar con el usuario de tal forma que se le induzca a pequeños conocimientos básicos de tecnología para que los requerimientos a cumplir se ajusten más a la realidad del desarrollo.
- Durante el diseño de una base de datos se recomienda tomar muy en cuenta aspectos como el nombre de los campos, el tipo de dato y el tamaño que se le asigna, para que luego no se tengan que realizar cambios en la parte que enlazan los datos con la programación.
- En JEE 6.0 se recomienda usar una serie de estándares en el desarrollo como la manera de nombrar los packages, nombrar las clases y sobre todo los métodos y constructores ya que influye en el funcionamiento y adaptación con las paginas xhtml.
- Durante el desarrollo se recomienda utilizar todas las herramientas que el entorno de desarrollo ofrece para solucionar los problemas que se presentan. Se puede visualizar los errores que muestra el servidor de aplicaciones o se puede utilizar el depurador que permite encontrar errores evaluando las variables que intervienen en el proceso.
- A pesar de que cada tecnología se esfuerza en mejorar la presentación web se recomienda buscar las mejores alternativas para mostrar una interfaz amigable y lo mas intuitiva posible. JSF cuenta con librerías como richfaces que implementan componentes que sirven de mucha ayuda para cumplir con el objetivo antes mencionado.
- Cuando se presenta la información en las interfaces se recomienda evitar el mostrar términos o campos no necesarios y que pudiesen causar

confusión en el usuario. Mensajes de error deben ir resaltados, eventos que impliquen modificación de información sensible deben estar controlados con Mensajes de confirmación.

- En aplicaciones de este tipo se recomienda hacer un buen análisis de los niveles de usuario que se van a necesitar implementar. Es importante controlar la visualización de los menús o accesos directos según el usuario que corresponda.
- Al momento de introducir al usuario conceptos como el de Helpdesk se recomienda ser lo más simple y concreto posible ya que la idea no es ahuyentar al usuario de seguir ciertas prácticas, sino de un futuro acostumbrar a cualquier usuario de TI a trabajar con estos conceptos.

## REFERENCIAS BIBLIOGRÁFICAS

### DIRECCIONES ELECTRÓNICAS

[1] Hewlett Packard Latinoamérica – Servicios – Aplicaciones Empresariales.  
[http://www.hp.com/latam/servicios/aplicaciones\\_empresariales/index.html](http://www.hp.com/latam/servicios/aplicaciones_empresariales/index.html)

[2] Universidad de Huelva – Departamento de ingeniería electrónica y sistemas informáticos. Nuevas tecnologías de la programación. 2006.  
[http://www.uhu.es/josel\\_alvarez/NvasTecnProg/recursos/tTema1.pdf](http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf)

[3] John Jairo Méndez Medero. Análisis comparativo de las plataformas J2EE y .NET aplicado al desarrollo de servicios web. 2008.  
<http://www.scribd.com/doc/19227578/PLATAFORMAS-J2EE-Y-NET-EN-EL-DESARROLLO-DE-SERVICIOS-WEB-J2EE-and-Net-Platforms-in-the-Development-of-Web-Services>

[4] Oscar Gonzales Moreno - Revista Dr.Dobbs. - J2EE vs .NET.  
<http://www.consultec.es/DocInformes/j2ee%20VS%20NET.pdf>

[5] La tecla de escape – Artículos – Metodologías Software.  
<http://latecladeescape.com/w0/ingenieria-del-software/metodologias-de-desarrollo-del-software.html>

[6] Jorge Fiestas Martín – Aplicaciones Empresariales con JEE6 – Capa persistencia con JPA 2.0 <http://www.youtube.com/watch?v=nzFzlQyoCA>

[7] Jorge Fiestas Martín – Aplicaciones empresariales J2EE – Pool de Conexiones y Data source en glassfish <http://www.youtube.com/watch?v=UM6-F-0VG3Q>

[8] Diario Electrónico Terraza de Aravaca – Comparativa de implementaciones de JPA. <http://terrazadearavaca.blogspot.com/2008/12/comparativa-de-implementaciones-de-jpa.html>

[9] Alvaro Marín Illera – MySQL – Instalación, Configuración y manejo de MySql. <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html>

[10] Mundo Geek – Hibernate <http://mundogeek.net/archivos/2007/01/27/hibernate/>

[11] Programación en castellano – Introducción a la tecnología Java Server faces [http://www.programacion.com/articulo/introduccion\\_a\\_la\\_tecnologia\\_javascript\\_faces\\_233](http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233)

[12] Natalia Rodriguez – ITIL, Mejores Prácticas – Centro de Investigación de Telecomunicaciones Bogotá, 2008

[13] Alejandro Floría Cortés –Usabilidad y diseño centrado en el usuario - Departamento de Ingeniería de Diseño y Fabricación.- Universidad de Zaragoza. <http://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/CuestCon.htm>

# ANEXOS

# **ANEXO A**

## DICCIONARIO DE DATOS

**Empresa:** Almacena los datos de la empresa proveedora del servicio.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
RUC	varchar(13)	X					Ruc de la empresa
NOMBRE	varchar(20)						Nombre Comercial
DIRECCION	varchar(50)						Dirección de la empresa
EMAIL	varchar(30)						Email de la empresa
TELEFONO	varchar(50)						Teléfono de la empresa
FRASECLAVE	varchar(20)						Frase Clave para el cifrado de la licencia

**Vendedor:** Almacena los vendedores de la empresa.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador del Vendedor
CEDULA	varchar(10)			X			Cédula del vendedor
NOMBRE	varchar(20)						Nombre del Vendedor

**Usuario:** Almacena los usuarios de la aplicación.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador del usuario
ID_VEND	int(10)		X				Identificador del Vendedor
NOMBRE	varchar(50)						Nombre de Usuario
CEDULA	varchar(13)			X			Cedula o Ruc del Usuario
TIPO	varchar(10)						Tipo de Usuario
DIRECCION	varchar(50)						Dirección del usuario
TELEFONO	varchar(30)						Teléfono del Usuario
EMAIL	varchar(30)						Email del Usuario
CONTACTO	varchar(20)						Contacto del Usuario
ACTIVIDAD	varchar(20)						Actividad del Usuario
PASSWORD	varchar(50)						Contraseña de Usuario
CONTADOR	int(10)						Número de licencias obtenidas
LIC_MAX	int(10)						Máximo de licencias permitidas

**Contrato:** Almacena los contratos entre la empresa y el cliente

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID_CONTRATO	int(11)	pk			X		Identificador de Contrato
FECHA	date						Fecha
MAX_LICENCIAS	int(11)						Máximo de licencias

LICENCIAS_OBTENIDAS	int(11)						Licencias obtenidas
ID_USUARIO	Int(11)		X				Clave del cliente
ESTADO	int(11)						Estado del contrato

**Módulo:** Almacena los diferentes módulos del sistema.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador del Módulo
NOMBRE	varchar(20)						Nombre del Módulo

**Problema:** Almacena los problemas comunes que se presentan en el sistema.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador del problema
DESCRIPCION	text			X			Descripción del problema
TIPO	varchar(15)						Tipo de problema

**Solución:** Almacena las soluciones a problemas del sistema.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador de la Solución
DESCRIPCION	text			X			Descripción de la Solución
FECHA	date						Fecha de ingreso de la solución

**ProblemaMódulo:** Forma una sugerencia de solución, enlazando el problema, el módulo donde ocurre y la solución.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador del problemaMódulo
ID_PROBLEMA	int(10)		X				Identificador del problema
ID_MODULO	int(10)		X				Identificador del Módulo
ID_SOLUCION	int(10)		X				Identificador de la Solucion

**Licencia:** Almacena las claves de activación de licencias obtenidas por el usuario.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	pk			X		Identificador de Licencia
ID_USUARIO	int(10)		X				Identificador de Usuario
CLAVE1	int(20)						Numero que arroja el computador
LICENCIA	varchar(20)						Clave que Activa el sistema
FECHA	datetime						Fecha de Obtención
REFERENCIA	varchar(20)					X	Referencia del computador a activar

**Incidencia:** Almacena las incidencias registradas por el usuario.

CAMPO	TIPO DE DATO	Clave Prim.	Clave For.	Clave Única	Auto Incre.	Valor nulo	Descripción
ID	int(10)	X			X		Identificador de Incidencia
ID_USUARIO	int(10)		X				Identificador del Usuario
ID_PROBLEMA	int(10)		X				Identificador del ProblemaMódulo
FECHA	datetime						Fecha de ocurrencia
ESTADO	varchar(10)						Estado de la Incidencia
COMENTARIO	varchar(30)					X	Comentario de la Incidencia

# **ANEXO B**

## MANUAL DE INSTALACIÓN DE LA APLICACIÓN

### A.- INSTALACIÓN DE MYSQL UTILIZANDO LA HERRAMIENTA WAMP

- Ejecutamos el instalador de wamp e instalamos con las configuraciones por defecto.
- Vamos al directorio de instalación de wamp, phpmyadmin y nos ubicamos en el archivo config.inc.php para editarlo.
- Dentro del archivo cambiamos el password, de vacío al password milenio2011 y guardamos.
- Ejecutamos wamp y vamos a phpMyadmin, en la pantalla principal creamos una base de datos de nombre sasm.
- Luego en el menú principal vamos a Privilegios y editamos las opciones del usuario root y le ponemos el password.
- Luego vamos a Importar y seleccionamos importar desde un archivo sql y buscamos el archivo sasm.sql.

### B.- INSTALACIÓN DE NETBEANS 6.8 INCLUIDO GLASSFISH 3.0 (Windows)

#### *Requerimientos de Hardware:*

- Sistema Operativo: Windows XP Professional SP3, Windows Vista SP1, Windows 7, algunas versiones de Macintosh, Solaris, Ubuntu 9.04 y varias distribuciones de Linux.
- Procesador: Intel Pentium IV o equivalente a 2,6 GHz
- Memoria: 2 GB
- Espacio de disco: 1 GB de espacio libre en el disco

Aunque los requisitos del sistema anteriores se aplican a la mayoría de configuraciones del IDE, algunos aspectos de la compatibilidad con NetBeans Web y Java EE resultan muy exigentes en relación con los recursos del sistema,

sobre todo si se utilizan junto con varios servidores de aplicaciones. Si desea obtener mejores resultados al desarrollar aplicaciones web y Java EE, asegúrese de que el sistema cumpla al menos los requisitos del sistema *recomendados*.

#### *Requisitos de software:*

NetBeans IDE funciona en el kit de desarrollo de Java SE (JDK), que consta del entorno de ejecución de Java y de herramientas para desarrolladores para la compilación, depuración y ejecución de aplicaciones escritas en lenguaje Java.

Los JDK probados con esta versión son los siguientes:

- JDK 6 Actualización 14
- JDK 5 Actualización 17

#### *Instalación:*

Asegurarse de poseer los requerimientos antes mencionados.

En Windows, ejecutar el instalador, escoger las configuraciones por defecto, si se desea se puede personalizar el directorio de instalación y desactivar algunas características del IDE.

En Linux, Otorgamos permisos de ejecución( `chmod +x netbeans-6.8-ml-linux.sh`) y ejecutamos el instalador (`sudo bash netbeans-6.8-ml-linux.sh`).

Esta versión de Netbeans instala automáticamente los controladores que trabajan con gestores de bases de datos como mysql y postgresql.

### **C.-INSTALACION DE GLASSFISH EN LINUX.**

Para instalar el ambiente de debe instalar los siguientes componentes:

- Red Hat Enterprise Linux 5.5 i386

- Apache Web Server 2.2
- JDK 6.0
- Servidor de Aplicaciones Glassfish 3.0.1

En el equipo correspondiente debe existir la instalación del Sistema Operativo RHEL 5.4 o RHEL 5.5 y debe estar activo el servidor Apache Web Server 2.2.

El procedimiento se encuentra realizado para 32 bits, pero se seguirá el mismo para 64 bits

- **Configuración YUM.**

Primeramente vamos a configurar el gestor de paquetes de Linux a partir de los CDs de instalación de Linux. Se asume que se encuentra en el directorio en el que están las imágenes de los CDs.

```
# mkdir /opt/RHEL5.5
# mkdir /mnt/iso
# mount -t iso9660 -o loop rhel-server-5.5-i386-dvd.iso /mnt/iso
# cp -a /mnt/iso/Server /opt/RHEL5.5/
# cp -a /mnt/iso/VT /opt/RHEL5.5/
# umount /mnt/iso
# mount -t iso9660 -o loop rhel-server-supplementary-5.5-i386-disc1.iso
/mnt/iso
# cp -a /mnt/iso /opt/RHEL5.5/ Supplementary
# umount /mnt/iso
```

Crear el archivo /etc/yum.repos.d/dvd-server.repo con el siguiente contenido:

```
[dvd-server]
name=DVD Server
baseurl=file:///opt/RHEL5.5/Server
enabled=1
gpgcheck=0
```

```
[dvd-server-VT]
name=DVD Server VT
baseurl=file:///opt/RHEL5.5/VT
enabled=1
gpgcheck=0
```

Crear el archivo `/etc/yum.repos.d/cd-supplementary.repo` con el siguiente contenido:

Instalar el repositorio EPEL (Extra Packages for Enterprise Linux)

```
# rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm
```

- **Instalación de JDK.**

Para instalar el JDK se debe instalar primeramente el complemento libXP

```
# yum install libXp
```

Luego instalamos el JDK desde el CD suplementario provisto por Red Hat.

```
# yum install java-1.6.0-sun*
```

- **Instalación de glassfish 3.0.1.**

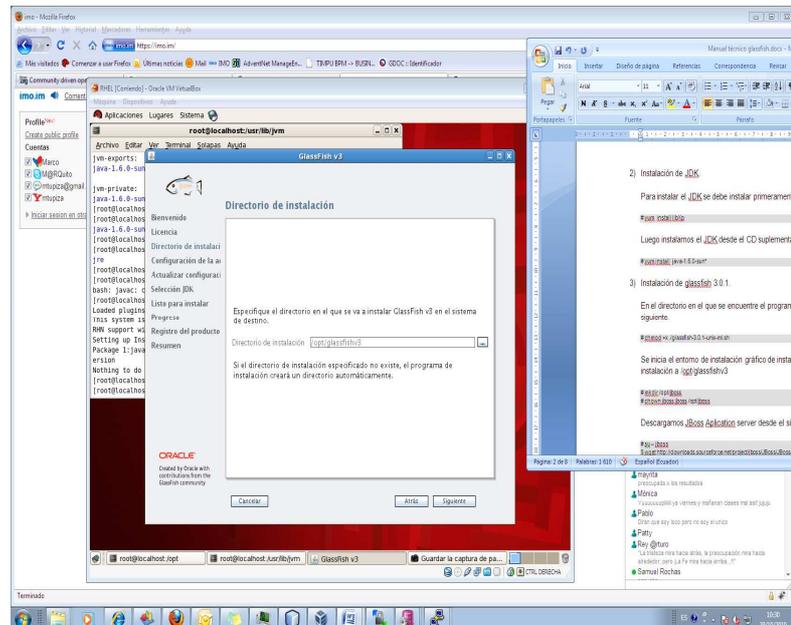
Descargar el software glassfish desde el sitio oficial

<http://download.java.net/glassfish/3.0.1/release/glassfish-3.0.1-unix.sh>

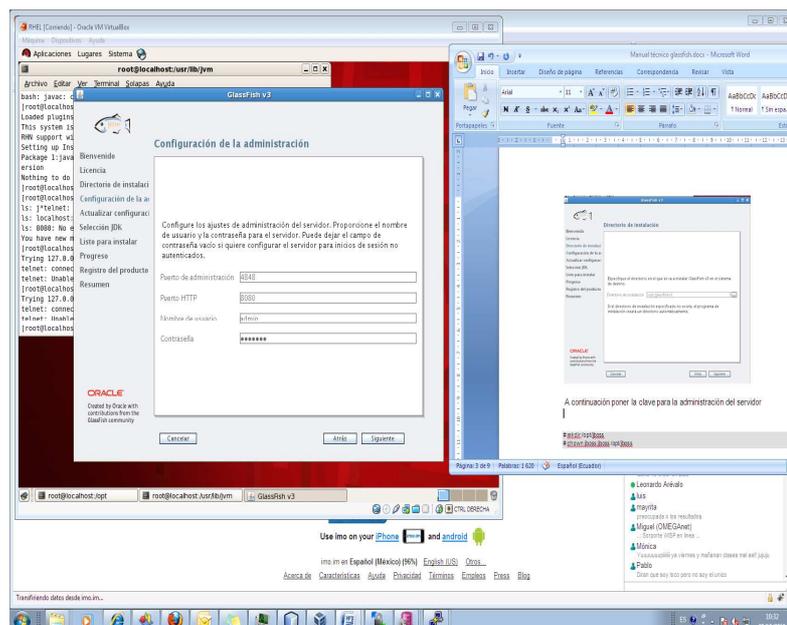
En el directorio en el que se encuentre el programa de instalación "*glassfish-3.0.1-unix-ml.sh*" ejecutamos lo siguiente.

```
# chmod +x ./glassfish-3.0.1-unix-ml.sh
```

Se inicia el entorno de instalación gráfico de instalación, e indicar el directorio de instalación a /opt/glassfishv3



A continuación poner la clave para la administración del servidor



En las siguientes pantallas continuar con la instalación por defecto

Iniciar y detener el servidor

```
# /opt/glassfishv3/glassfish/bin/asadmin start-domain
```

```
# opt/glassfishv3/glassfish/bin/asadmin stop-domain
```

Para instalar el servidor GlassFish como servicio seguimos los siguientes pasos:

Crear el usuario glassfish bajo el cual debe ejecutarse el servidor y poner una contraseña:

```
# useradd glassfish
```

```
# passwd glassfish
```

Asignar permisos al usuario glassfish

```
# chown -R glassfish:glassfish /opt/glassfishv3
```

Crear el archivo de inicio /etc/init.d/glassfish para el servidor glassfish con el siguiente contenido:

```
#!/bin/bash
```

```
#
```

```
# glassfish:      Startup script for Glassfish Application Server.
```

```
#
```

```
# chkconfig: 3 80 05
```

```
# description:    Startup script for domain1 of Glassfish Application Server.
```

```
GLASSFISH_HOME=/opt/glassfishv3/glassfish;
```

```
export GLASSFISH_HOME
```

```
GLASSFISH_OWNER=glassfish;
```

```
export GLASSFISH_OWNER
```

```
start() {  
    echo -n "Starting Glassfish: "  
    echo "Starting Glassfish at `date`" >>  
    $GLASSFISH_HOME/domains/domain1/logs/startup.log  
    su $GLASSFISH_OWNER -c "$GLASSFISH_HOME/bin/asadmin start-  
domain domain1" >> $GLASSFISH_HOME/domains/domain1/logs/startup.log  
    sleep 2  
    echo "done"  
}
```

```
stop() {  
    echo -n "Stopping Glassfish: "  
    echo "Stopping Glassfish at `date`" >>  
    $GLASSFISH_HOME/domains/domain1/logs/startup.log  
    su $GLASSFISH_OWNER -c "$GLASSFISH_HOME/bin/asadmin stop-  
domain domain1" >> $GLASSFISH_HOME/domains/domain1/logs/startup.log  
    echo "done"  
}
```

```
# See how we were called.
```

```
case "$1" in  
    start)  
        start  
        ;;  
    stop)  
        stop  
        ;;  
    restart)  
        stop  
        start  
        ;;  
    *)
```

```
    echo $"Usage: glassfish {start|stop|restart}"  
    exit  
esac
```

- **Instalar el servicio**

```
# chmod +x /etc/init.d/glassfish  
# chkconfig glassfish on
```

Iniciar y detener el servidor.

```
# service glassfish start  
# service glassfish stop
```

Copiar archivo

sql: sqljdbc4.jar

postgres: postgresql-9.0-801.jdbc4.jar

Oracle: ojdbc6.jar

necesario para conexión a la BD en el directorio

/opt/glassfishv3/glassfish/domains/domain1/lib/ext utilizando el usuario  
glassfish.

```
$ cp postgresql.jdbc4.jar /opt/glassfishv3/glassfish/domains/domain1/lib/ext
```

- Instalación de Apache Web Server.

Ejecutar en caso de no tener instalado Apache Web Server.

```
# yum install httpd  
# service httpd start  
# chkconfig httpd on
```

- **Configurar MOD\_JK**

La configuración del módulo mod\_jk sirve para integrar el servidor GlassFish al Apache web server, para esto se deben seguir los siguientes pasos.

Bajar el módulo mod\_jk del sitio oficial de Jakarta tomcat para la plataforma respectiva, en este caso es para 32 bits, renombramos y damos los permisos necesarios.

```
# cd
# wget wget http://mirrors.kahuki.com/apache/tomcat/tomcat-connectors/jk/binaries/linux/jk-1.2.30/i586/mod\_jk-1.2.30-httpd-2.2.X.so
# mv mod_jk-1.2.30-httpd-2.2.X.so /etc/httpd/modules/mod_jk.so
# chmod +x /etc/httpd/modules/mod_jk.so
# chown root:root /etc/httpd/modules/mod_jk.so
```

Crear el archivo /etc/httpd/conf.d/mod\_jk.conf con el siguiente contenido:

```
# Include mod_jk configuration file
Include conf/mod-jk.conf
```

Crear el archivo /etc/httpd/conf/mod-jk.conf con el siguiente contenido:

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/worker.properties
# Where to put jk logs
JkLogFile logs/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel debug
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
```

```
# JkRequestLogFormat set the request format
JkRequestLogFormat "%w %V %T"
# Send all jsp requests to GlassFish
JkMount /*.jsp worker1
# Send all glassfish-test requests to GlassFish
JkMount /glassfish-test/* worker1
```

Crear el archivo `/etc/httpd/conf/workers.properties` con el siguiente contenido:

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=localhost.localdomain
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.cachesize=10
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.socket_timeout=300
```

Reiniciar el servidor Apache

```
# service httpd restart
```

- **Configurar el servidor GlassFish**

Habilitar `mod_jk`, para ejecutar los comandos debe estar ubicado en el directorio `/opt/glassfishv3/glassfish/bin`

```
# ./asadmin create-http-listener --listenerport 8009 --listeneraddress 0.0.0.0 --
defaultvs server jk-connector
```

```
# ./ asadmin set configs.config.server-config.network-config.network-  
listeners.network-listener.jk-connector.jk-enabled=true
```

Verificar si el servidor está levantado digitando en el navegador web:

[http://IP\\_SERVIOR:8080/](http://IP_SERVIOR:8080/)

#### **D.-CONFIGURACIÓN DEL DATA SOURCE EN GLASSFISH 3.0**

- Una vez iniciado netbeans, vamos a Prestaciones, Servicios, Glassfish 3.0, click derecho e iniciamos el servidor de aplicaciones.
- Si ya está corriendo podemos dar click derecho y elegir Ver Consola de administrador.
- La consola se muestra en el browser, el user y password suelen ser admin y admin.
- Vamos a JDBC, Conjunto de conexiones, y Nuevo conjunto de conexiones agregamos un nombre, el tipo es javax.sql.DataSource, y elegimos el gestor de base de datos: mysql.
- En Propiedades Adicionales debemos tener ingresados los siguientes parámetros:  
User: root  
Password: milenio2011  
ServerName: localhost  
PortNumber: 3306  
DriverClass: mysql.jdbc.driver  
DatabaseName: sasm

- Verificamos que funcione haciendo un Sondeo, y que el resultado tenga éxito
- Luego en el menú principal vamos a JDBC, Datasource, Nuevo Datasource, agregamos un nombre: jdbc/sasm y que trabaje con el conjunto de conexiones antes creado.

#### **D.- IMPLEMENTACIÓN DE LA APLICACIÓN EN NETBEANS 6.8**

- Una vez iniciado el IDE, iniciamos también el servidor de aplicaciones.
- Vamos a abrir proyecto y buscamos el contenido descomprimido del archivo sasm.zip.
- Seleccionamos el directorio y automáticamente se habrá importado la aplicación web.
- Luego le damos click derecho en la aplicación y Ejecutar.

# ANEXO C

## MANUAL DE USUARIO DE LA APLICACIÓN

Para ejecutar la aplicación tenemos que ingresar en nuestro browser la siguiente

URL: <http://direcciondelServidor/sasm>

Y se despliega la pantalla de login de la aplicación.

### A.- PERFILES DE USUARIO

El prototipo de la aplicación establece 2 tipos de Usuario.

Usuario Administrador: empleado de la Corporación MilenioSystem que tiene acceso a todos los módulos de la aplicación.

Usuario Cliente: empleado de la empresa cliente y que utiliza el prototipo para obtener claves de activación del sistema o para obtener información de ayuda para resolución de problemas.

### B.-MANUAL DEL CLIENTE

- Ingreso a la aplicación:

En la pantalla de login, ingrese su número de cédula o RUC acompañado de su clave personal. Si no recuerda la clave debe solicitar el cambio a un administrador.

- Modificación de los datos de usuario:

Una vez dentro de la aplicación, puede actualizar sus datos. Desde *Menu-Usuario-Ver Datos de Usuario*. Y Seguir las links para llamar al evento actualizar.

- Obtención de claves de activación de licencias:

Luego de que haya instalado el sistema Milenio3 en un computador. Ejecute el programa desde el acceso directo del escritorio. Como se trata de la primera vez el sistema le solicita una clave de activación y le muestra el identificador del computador.

Ahora en la aplicación web en el *Menú principal* presione el botón *Licencias*.

Digite con cuidado el número identificador que muestra su computador.

Escriba alguna referencia que permita ubicar el computador donde se instalo el programa.

Revise que los datos sean correctos y presione el botón *Obtener Clave*.

El numero que muestra la aplicación ingréselo en el sistema Milenio3 para dar por finalizado la activación de la licencia.

El número máximo de veces que usted puede obtener claves, la define el administrador cuando crea un usuario. Si usted ha agotado su máximo de claves y desea obtener más, debe ponerse en contacto con el administrador para que aumente su cupo previo a un acuerdo económico de ambas partes.

- Mesa de ayuda de la aplicación:

Si mientras maneja el sistema Milenio 3 ocurre un error inesperado y que influye en su correcto desempeño, diríjase a la aplicación, al *Menú principal* y presione el botón *Mesa de Ayuda*.

El primer paso es ubicar el problema que le acaba de ocurrir. Usted puede digitar parte del mensaje de error que el sistema le muestra, para buscar en la lista de problemas. Otra opción es digitar solo el módulo del sistema en donde acaba de

ocurrir el problema y de esa manera la lista se muestra filtrada y facilita la ubicación del problema.

Una vez ubicado el problema presione el enlace: ver *sugerencia de solución*, lo cual llama a la pantalla que muestra las instrucciones que tiene más probabilidad de solucionar el problema.

Es muy importante que al final indique si el problema fue resuelto o es necesario buscar otro problema. De esta manera ayuda a registrar las incidencias y ayuda a monitorear el desempeño de los sistemas.

- Salir de la Aplicación:

Para salir de la aplicación presione en el *Menú – Cerrar Sesión*.

## **C.- MANUAL DEL ADMINISTRADOR**

- Ingreso a la aplicación:

En la pantalla de login, ingrese su número de cédula o RUC acompañado de su clave personal. Si no recuerda la clave debe solicitar el cambio a otro administrador.

- Administración de la Empresa:

Como administrador de la aplicación usted puede modificar los datos importantes de la empresa. Desde el *menú-Administración-Zona de Administradores* y ingresando al Modulo: *Datos de la empresa*.

Aquí puede seguir los enlaces que llaman a los eventos de modificación.

- Administración de los Módulos:

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Módulos*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de módulos.

- Administración de los Vendedores:

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Vendedores*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de vendedores.

- Administración de los Usuarios:

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Usuarios*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de usuarios.

Es importante tener cuidado al momento de definir el tipo de usuario. Si no existe ninguna disposición adicional, el máximo de licencias permitidas a activar es de 1.

- Administración de Problemas:

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Problemas*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de problemas.

El texto debe contener la descripción exacta del problema, tal cual como lo muestra el sistema o como se lo distinga fácilmente.

- Administración de las soluciones:

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Soluciones*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de soluciones.

El texto de la solución debe ser sencillo y conciso con la finalidad de garantizar el entendimiento del usuario cliente.

- Administración de Sugerencias de solución

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Sugerencias*, se puede seguir los enlaces que me permiten la creación, modificación o eliminación de sugerencias. Una sugerencia consiste en el enlace de un problema, un módulo y una solución.

- Administración de Licencias

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Licencias*, usted puede visualizar todos los registros de claves obtenidas por el usuario. No está permitida la eliminación de algún registro ya que es importante tener la información completa para monitorear el desempeño del proceso del negocio.

- Administración de Incidencias

Desde el *menú-Administración-Zona de Administradores* e ingresando a la sección: *Administrar Incidencias*, usted puede visualizar todos los registros de incidencias registradas por el usuario. No está permitida la eliminación de algún registro ya que es importante tener la información completa para monitorear el desempeño del proceso del negocio.

# ANEXO D

## CONFIGURACIÓN WEB

Archivo faces-config.xml:

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- ===== FULL CONFIGURATION FILE ===== -->

<faces-config version="2.0"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xi
  <application>
    <resource-bundle>
      <base-name>/resources/Bundle</base-name>
      <var>bundle</var>
    </resource-bundle>
  </application>
</faces-config>
```

Archivo sun-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server
<sun-web-app error-url="">
  <context-root>/sasm</context-root>
  <class-loader delegate="true"/>
  <jsp-config>
    <property name="keepgenerated" value="true">
      <description>Keep a copy of the generated servlet class' java code.</desc
    </property>
  </jsp-config>
</sun-web-app>
```

## Archivo web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```

## Ejemplo página login.xhtml:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich">
  <ui:composition>
    <ui:define name="title">
      Iniciar sesión
    </ui:define>
    <ui:define name="datos_titulo">
      Iniciar sesión
    </ui:define>
    <h:form>
      <p><h:outputLabel value="CÉDULA:" for="cedula" /></p>
      <h:inputText id="cedula" value="#{usuarioController.cedula1}" />
      <p><h:outputLabel value="CONTRASEÑA:" for="password"/></p>
      <h:inputSecret id="password" value="#{usuarioController.password}"/>
      <br />
      <center>
        <h:panelGrid columns="2">
          <rich:toolBar itemSeparator="line">
            <rich:toolBarGroup>
              <a4j:commandButton value="Acceder"
action="#{usuarioController.doLogin}" image="/resources/images/icons/singer.gif"/>
              <h:outputLabel value="Aceptar" for="aceptar" />
            </rich:toolBarGroup>
          </rich:toolBar>
          <rich:toolBar itemSeparator="line">
            <rich:toolBarGroup>
              <a4j:commandButton value="Cancelar" action="index"
image="/resources/images/icons/disc.gif"/>
              <h:outputLabel value="Cancelar" for="cancelar" />
            </rich:toolBarGroup>
          </rich:toolBar>
        </h:panelGrid>
      </center>
    </h:form>
  </ui:composition>
```

```

        </h:panelGrid>
    </center>
    <br/>
    <br/>

    <f:subview id="vista">
<rich:modalPanel id="panel" width="350" height="120">
    <f:facet name="header">
        <h:panelGroup>
            <h:outputText value="Corporación Mileniosystem"></h:outputText>
        </h:panelGroup>
    </f:facet>
    <f:facet name="controls">
        <h:panelGroup>
            <h:graphicImage value="/images/close.png" styleClass="hidelink" id="hidelink"/>
            <rich:componentControl for="panel" attachTo="hidelink" operation="hide"
event="onclick"/>
        </h:panelGroup>
    </f:facet>
    <h:outputText value="Ingrese su numero de cedula y su clave personal. Si usted todavia
no es un usuario de este portal,pida la creacion sin costo, al 6042344."></h:outputText>
    <br/>
    <h:outputText value=""></h:outputText>
</rich:modalPanel>

        <div align="righth">
            <h:outputLink value="#" id="link" styleClass="alineacionizq">
                ?
            <rich:componentControl for="panel" attachTo="link" operation="show" event="onclick"/>
        </div>
    </h:outputLink>

</f:subview>
    </h:form>

</ui:composition>
</html>

```

## Archivo sasm.css:

```

.textoImportante{
    height: 36px;
    color: #000000;
    font-weight : bold;
    font-size: 16px;
}
.piel{
    text-align: center;
    height: 36px;
    color: #330033;
    font-weight : bold;
    font-size: 10px;
}
.titulo2{
    text-align: center;
    height: 36px;
    color: #330033;
    font-weight : bold;
    font-size: 12px;
}

```

## CÓDIGO FUENTE:

### Ejemplo Clase entidad: Modulo

```

/**
 * @author David Guano david.guano@gmail.com
 * @version 1.0 01/01/2011
 */
@Entity
@Table(name = "modulo", catalog = "sasm", schema = "")
@NamedQueries({
    @NamedQuery(name = "Modulo.findAll", query = "SELECT m FROM Modulo m"),
    @NamedQuery(name = "Modulo.findById", query = "SELECT m FROM Modulo m WHERE m.id = :id"),
    @NamedQuery(name = "Modulo.findByName", query = "SELECT m FROM Modulo m WHERE m.nombre = :nombre")
})
public class Modulo implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Column(name = "ID")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer id;
    @Basic(optional = false)
    @Column(name = "nombre", nullable = false, length = 20)
    private String nombre;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "idModulo")
    private Collection<ProblemaModulo> problemaModuloCollection;

    public Modulo() {
    }

    public Modulo(Integer id) {
        this.id = id;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public Collection<ProblemaModulo> getProblemaModuloCollection() {
        return problemaModuloCollection;
    }

    public void setProblemaModuloCollection(Collection<ProblemaModulo> problemaModuloCollection) {
        this.problemaModuloCollection = problemaModuloCollection;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Modulo)) {
            return false;
        }
        Modulo other = (Modulo) object;
        if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return this.getNombre() + " - " + id ;
    }
}

```

## Ejemplo bean de sesión sin estado: UsuarioFacade

```

/**
 * Permite obtener una referencia EntityManager, el nombre de la unidad de persistencia debe ser <b>sas
 * <P>
 * <H6>Soporte:EPN <I>david.guano@gmail.com</I></H6>
 * @author David Guano david.guano@gmail.com
 * @version 1.0 01/01/2011
 */
@Stateless
public class UsuarioFacade {
    @PersistenceContext(unitName = "sasmpu")
    private EntityManager em;

    /**
     * Crea un registro en la base de datos en la tabla Usuario
     * @param usuario objeto que se inserta
     */
    public void create(Usuario usuario) {
        em.persist(usuario);
    }

    /**
     * Modifica un registro en la base de datos en la tabla Usuario
     * @param usuario objeto que se modifica
     */
    public void edit(Usuario usuario) {
        em.merge(usuario);
    }

    /**
     * Borra un registro en la base de datos en la tabla Usuario
     * @param usuario objeto que se elimina
     */
    public void remove(Usuario usuario) {
        em.remove(em.merge(usuario));
    }

    /**
     * Devuelve un objeto de tipo Usuario
     * @param id identificador del registro
     * @return Usuario con el id
     */
    public Usuario find(Object id) {
        return em.find(Usuario.class, id);
    }

    /**
     * Devuelve una lista con todos los objetos de tipo Usuario
     * @return Lista de Usuarios
     */
    public List<Usuario> findAll() {
        return em.createQuery("select object(o) from Usuario as o").getResultList();
    }

    /**
     * Devuelve una lista de objetos tipo Usuario
     * @param range rango de donde mostrar la lista
     * @return Lista de Usuarios
     */
    public List<Usuario> findRange(int[] range) {
        Query q = em.createQuery("select object(o) from Usuario as o");
        q.setMaxResults(range[1] - range[0]);
        q.setFirstResult(range[0]);
        return q.getResultList();
    }

    /**
     * Devuelve el numero d registros de la tabla Usuario
     * @return numero de registros
     */
    public int count() {
        return ((Long) em.createQuery("select count(o) from Usuario as o").getSingleResult()).intValue()
    }
}

```

```
//METODOS CREADOS PARA AYUDAR A LAS CLASES DE NEGOCIO

/**
 * Devuelve una Usuario a partir de su cedula
 * @param cedula cedula del usuario a encontrar
 * @return Usuario
 */
public Usuario buscarPorCedula(String cedula) {
    Query q = em.createQuery("SELECT object(u) FROM Usuario AS u " +
        " WHERE u.cedula = :cedula");
    q.setParameter("cedula", cedula);

    List<Usuario> resultados = q.getResultList();

    if (resultados == null) {
        return null; // No encontrado
    }
    else if (resultados.size() != 1) {
        return null; // No encontrado
    }
    else {
        return resultados.get(0); // Devuelve el encontrado
    }
}

/**
 * Devuelve una Usuario a partir de su Id
 * @param ID identificador de un Usuario
 * @return Usuario
 */
public Usuario buscarPorId(int ID) {
    Query q = em.createQuery("SELECT object(u) FROM Usuario AS u " +
        " WHERE u.ID = :ID");
    q.setParameter("ID", ID);

    List<Usuario> resultados = q.getResultList();

    if (resultados == null) {
        return null; // No encontrado
    }
    else if (resultados.size() != 1){
        return null; // No encontrado
    }
    else {
        return resultados.get(0); // Devuelve el encontrado
    }
}
}
```

## Ejemplo backing bean : UsuarioController

```

/**
 * Clase backing Bean que administra la entidad Usuario.
 * <p>
 * <H6>Soporte:EPN <I>david.guano@gmail.com</I></H6>
 * @author David Guano david.guano@gmail.com
 * @version 1.0 01/01/2011
 */
@ManagedBean (name="usuarioController")
@SessionScoped
public class UsuarioController {

    private Usuario current;
    private DataModel items = null;
    private PaginationHelper pagination;
    private int selectedItemIndex;
    private List<String> listaTipos =new ArrayList<String>();
    private int permisos;
    private String cedula1;
    private String password;
    private Usuario usuarioactual;
    private String passConfirm;
    private Boolean variableEstado=false;
    @EJB private UsuarioFacade ejbFacade;
    @EJB private GestorUsuariosFacadeLocal gestorUsuariosFacade;

    /**
     * metodo que me permite logear a un Usuario autenticandolo y lo almacena
     * @return menu principal si la autenticación funciona, sino fallo de login
     */
    public String doLogin() {
        listaTipos.add("ADMIN");
        listaTipos.add("CLIENTE");
        try {
            String destino = null;
            String passwordCifrado;
            passwordCifrado=gestorUsuariosFacade.encriptarMDS(password);

            if (gestorUsuariosFacade.autenticarUsuario(cedula1, passwordCifrado)) {
                usuarioactual=gestorUsuariosFacade.almacenarUsuario(cedula1);
                variableEstado=true;
                this.asignarPermisos();
                destino = "menu";
            } else {
                destino = "/errores/fallologin";
            }

            return destino;
        }catch (Exception e){
            JsfUtil.addErrorMessage("INGRESE SU CEDULA Y SU CONTRASEÑA");
            return null;
        }
    }
}

```

## Ejemplo backing bean: LicenciaController

```

/**
 * Clase backing Bean que administra la entidad Licencia.
 * <p>
 * <H6>Soporte:EPN <I>david.guano@gmail.com</I></H6>
 * @author David Guano david.guano@gmail.com
 * @version 1.0 01/01/2011
 */
@ManagedBean (name="licenciaController")
@SessionScoped
public class LicenciaController {

    private Licencia current;
    private DataModel items = null;
    @EJB private org.jp.a.sesion.LicenciaFacade ejbFacade;
    private PaginationHelper pagination;
    private int selectedItemIndex;
    private String idpc;
    private String clave;
    private Usuario usuarioActual= null;
    private String referencia;
    // Acceso al Managed Bean usuarioController (necesario para recuperar datos del cliente actual)
    @ManagedProperty(value = "#{usuarioController}")
    UsuarioController usuarioController;
    public UsuarioController getUsuarioController() {
        return usuarioController;
    }
    public void setUsuarioController(UsuarioController usuarioController) {
        this.usuarioController = usuarioController;
    }

}

/**
 * Obtiene una clave a partir del identificador
 * @return null
 */
public void obtenerClave() {
    usuarioController.aumentarContador();
    usuarioActual=usuarioController.getUsuarioActual();
    selectedItemIndex = -1;
    String cadena;
    cadena = idpc + "MAURO";
    int num = 0;
    for (int i=0; i < cadena.length(); i++){
        num = num + (int)cadena.charAt(i);
    }
    clave= String.valueOf(num);
    Date fecha1= java.util.Calendar.getInstance().getTime();
    try {
        getFacade().crearLicencia(Integer.parseInt(idpc),clave,referencia,fecha1,usuarioActual);
        JsflUtil.addSuccessMessage(ResourceBundle.getBundle("/resources/Bundle").getString("Licenc
    ) catch (Exception e) {
        JsflUtil.addErrorMessage(e, ResourceBundle.getBundle("/resources/Bundle").getString("Persi
    )
}

```

## Ejemplo backing bean: ListaController

```

/**
 * Clase backing Bean que administra el proceso de Grabar incidencias.
 * <p>
 * <H6>Soporte:EPN <I>david.guano@gmail.com</I></H6>
 * @author David Guano david.guano@gmail.com
 * @version 1.0 01/01/2011
 */
@ManagedBean(name="listaController")
@SessionScoped
public class ListaController extends BaseController{

    private ProblemaModulo problemaActual = null;
    private Usuario usuarioActual = null;
    private List<ProblemaModulo> problemas = null;
    private String consultaProblema;
    private String consultaModulo;
    @EJB private ListaProblemasFacadeLocal listaProblemas;
    @EJB private ProblemaModuloFacade sugerenciaFacade;
    // Acceso al Managed Bean usuarioController (necesario para recuperar datos del cliente actual)
    @ManagedProperty(value = "#{usuarioController}")
    UsuarioController usuarioController;
    public UsuarioController getUsuarioController() {
        return usuarioController;
    }
    public void setUsuarioController(UsuarioController usuarioController) {
        this.usuarioController = usuarioController;
    }
}

```

```

/**
 * Devuelve un objeto seleccionado
 * @return la misma pagina con los componentes actualizados
 */
public String doBusquedaProblema() {
    String destino = null;

    if (consultaProblema != null) {
        problemas = listaProblemas.problemaPorDescripcionProblema(consultaProblema);
        if (problemas != null) {
            if (problemas.size() == 0) {
                destino = "/negocio/helpdesk1";
            } else {
                destino = "/negocio/helpdesk1";
            }
        }
    }
    return destino;
}
/**
 * Devuelve un objeto seleccionado
 * @return la misma página con los componentes actualizados
 */
public String doBusquedaModulo() {
    String destino = null;

    if (consultaModulo != null) {
        problemas = listaProblemas.problemaPorDescripcionModulo(consultaModulo);
        if (problemas != null) {
            if (problemas.size() == 0) {
                destino = "/negocio/helpdesk1";
            } else {
                destino = "/negocio/helpdesk1";
            }
        }
    }
    return destino;
}
}

```

## MÉTODOS IMPORTANTES

### Validar Email - UsuarioFacade

```

/**
 * Valida el correo electrónico
 * @param correo mail que se ingresa
 * @return valor verdadero si el mail es correcto, caso contrario falso
 */
public boolean validarEmail(String correo) {
    Pattern pat = null;
    Matcher mat = null;
    pat = Pattern.compile("^[0-9a-zA-Z]([_\\.w]*[0-9a-zA-Z])*@[0-9a-zA-Z]([-w]*[0-9a-zA-Z])?\\.([a-zA-Z]
mat = pat.matcher(correo);
    if (mat.find()) {
        System.out.println "[" + mat.group() + ""];
        return true;
    } else {
        return false;
    }
}
}

```

### Encriptar Contraseña - UsuarioFacade

```

/**
 * Encripta un String con el algoritmo MD5.
 * @param palabra texto a encriptar
 * @return el texto encriptado
 */
public String encriptarMD5(String palabra) {
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] b = md.digest(palabra.getBytes());
        int size = b.length;
        StringBuilder h = new StringBuilder(size);
        for (int i = 0; i < size; i++) {
            int u = b[i] & 255; // unsigned conversion
            if (u < 16) {
                h.append("0");
                h.append(Integer.toHexString(u));
                //h.append("0"+Integer.toOctalString(u));
            } else {
                h.append(Integer.toHexString(u));
                // h.append(Integer.toOctalString(u));
            }
        }
        return h.toString();
    } catch (NoSuchAlgorithmException ex) {
    }
    return null;
}

```

Validar RUC o Cédula- UsuarioFacade

```

/**
 * Valida el número de RUC o cédula de ciudadanía.
 * @param ruc El número de RUC (13 dígitos) o cédula (10 dígitos), esta no debe contener el guión
 * @return -1 si el RUC o cédula es incorrecto,
 * 1 si es RUC para Sociedades Privadas y Extranjeros sin cédula,
 * 2 si es RUC para Sociedades Públicas,
 * 3 si es RUC para Personas Naturales,
 * 4 si es cédula,
 */
public boolean validarCedulaUsuario(String ruc) {
    if (ruc.length() == 13 || ruc.length() == 10) {
        try { //comprobamos si ruc es numérico
            long l = Long.parseLong(ruc);
        } catch (Exception e) {
            return false;
        }
        try {
            int provincia = Integer.parseInt(ruc.substring(0, 2));
            if (provincia < 1 || provincia > 24) // recomendable utilizar archivo de propiedades
            {
                return false;
            }
            int tercerDigito = Integer.parseInt(ruc.substring(2, 3));
            if (tercerDigito == 9) { // es RUC para Sociedades Privadas y Extranjeros sin cédula

                int[] modulo11 = {4, 3, 2, 7, 6, 5, 4, 3, 2};
                int numeros[] = new int[9];
                for (int i = 0; i < 9; i++) {
                    numeros[i] = Integer.parseInt(ruc.substring(i, i + 1));
                    modulo11[i] *= numeros[i];
                }
                int suma = 0;
                for (int i = 0; i < 9; i++) {
                    suma += modulo11[i];
                }
                int digitVerificador = 11 - (suma % 11);
                if (suma % 11 == 0) {
                    digitVerificador = 0;
                } // si el residuo es 0 digitVerificador es 0

                int decimoDigito = Integer.parseInt(ruc.substring(9, 10));
                if (decimoDigito != digitVerificador) {
                    return false;
                }
                int numeroEstablecimientos = Integer.parseInt(ruc.substring(10, 13));
                if (numeroEstablecimientos > 9) {
                    return false;
                }
                return true;
            } else if (tercerDigito == 6) { // del RUC para Sociedades Públicas

```

...ocntinua...

```

int[] modulo11 = {3, 2, 7, 6, 5, 4, 3, 2};
int numeros[] = new int[8];
for (int i = 0; i < 8; i++) {
    numeros[i] = Integer.parseInt(ruc.substring(i, i + 1));
    modulo11[i] *= numeros[i];
}
int suma = 0;
for (int i = 0; i < 8; i++) {
    suma += modulo11[i];
}
int digitVerificador = 11 - (suma % 11);
if (suma % 11 == 0) {
    digitVerificador = 0;
} // si el residuo es 0 digitoVerificador es 0

int decimoDigito = Integer.parseInt(ruc.substring(8, 9));
if (decimoDigito != digitVerificador) {
    return false;
}
int numeroEstablecimientos = Integer.parseInt(ruc.substring(9, 13));
if (numeroEstablecimientos > 9) {
    return false;
}
return true;
} else if (tercerDigito < 6) { // RUC para Personas Naturales o cédula

int[] modulo11 = {2, 1, 2, 1, 2, 1, 2, 1, 2};
int numeros[] = new int[9];
for (int i = 0; i < 9; i++) {
    numeros[i] = Integer.parseInt(ruc.substring(i, i + 1));
    modulo11[i] *= numeros[i];
    if (modulo11[i] >= 10) {
        modulo11[i] -= 9;
    }
}
int suma = 0;
for (int i = 0; i < 9; i++) {
    suma += modulo11[i];
}
int digitVerificador = 10 - (suma % 10);
if (suma % 10 == 0) {
    digitVerificador = 0;
} // si el residuo es 0 digitoVerificador es 0

int decimoDigito = Integer.parseInt(ruc.substring(9, 10));
if (decimoDigito != digitVerificador) {
    return false;
}
if (ruc.length() == 13) {
    int numeroEstablecimientos = Integer.parseInt(ruc.substring(10, 13));
    if (numeroEstablecimientos > 1) {
        return false;
    }
    return true;
}
return true; // es una cedula
}
} catch (Exception e) {

return false;
}
}
return false; // ruc o cédula incorrecta
}
}

```

## Envío de Email - ListaController

```

/**
 * Permite enviar un mensaje simple a un correo electrónico.
 * @param smtpHost nombre del host SMTP ej:correo.quito.gov.ec
 * @param username nombre del usuario que envía el mail
 * @param password contraseña del usuario que envía el mail
 * @param recipientsTo correo electrónico del destinatario
 * @param subject asunto del mail
 * @param message el mensaje
 * @param from correo electrónico del quien envía el mensaje
 * @throws Exception si existe un error
 */
public static void sendMail(String smtpHost, String username, String password, String recipientsTo[], Str
    try {
        Properties properties = new Properties();
        properties.put("mail.transport.protocol", "smtp");
        properties.put("mail.smtp.host", smtpHost);
        properties.put("mail.smtp.auth", "true");

        Authenticator authenticator = new SMTPAuthenticator(username, password);

        Transport transport = null;
        try {
            //Session session = Session.getDefaultInstance(properties, authenticator);
            Session session = Session.getInstance(properties, authenticator);
            MimeMessage mimeMessage = new MimeMessage(session);
            mimeMessage.setFrom(new InternetAddress(from));
            InternetAddress[] addressTo = new InternetAddress[recipientsTo.length];
            for (int i = 0; i < addressTo.length; i++) {
                addressTo[i] = new InternetAddress(recipientsTo[i]);
            }
            mimeMessage.setRecipients(Message.RecipientType.TO, addressTo);

            mimeMessage.setSubject(subject);
            mimeMessage.setText(message);
            transport = session.getTransport();
            transport.connect(username, password);
            transport.sendMessage(mimeMessage, mimeMessage.getAllRecipients());
        } finally {
            if (transport != null) {
                try {
                    transport.close();
                } catch (Exception e) {
                    throw e;
                }
            }
        }
    } catch (Exception ex) {
        throw ex;
    }
}

```