

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. GALO ESTEBAN SOLÍS GALLO como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA

Sangolquí, 25 de noviembre de 2008

ING. TATIANA GUALOTUÑA
DIRECTORA DE TESIS

DEDICATORIA

Esta tesis está dedicada a mi Abuelo Félix Fortunato Solís Rosero, que a pesar de no estar aquí físicamente para poder disfrutar de este gran logro, me guió desde el cielo en cada paso de mi vida universitaria y no dejó que me rindiera después de mis tropiezos. Es una pena que en todos estos años no haya podido celebrar contigo cada etapa superada, pero siempre has estado en mi mente y mi corazón, es por eso que esta tesis de grado te la dedico a ti.

También dedico este trabajo a mis papás, que siempre están a mi lado, tanto para apoyarme como para regañarme cuando me equivoco y felicitarme cuando hago las cosas bien, sin los cuales no hubiera podido llegar a este punto de mi vida.

Galo Esteban Solís Gallo

AGRADECIMIENTOS

Quiero agradecer principalmente a Dios por guiarme durante toda mi vida, por rodearme de gente buena y luchadora, por darme una brisa de frescura en los momentos que más necesitaba de un respiro y por siempre brindarme una mano amiga pero nunca sentirme solo.

A ti madre por tu entrega fiel y abnegada a nuestra familia, por creer en mí, por los años en los que me cuidaste, por ser mi compañera en algunos desvelos cuando necesito conversar y desahogarme, por ser mi amiga y además por ser la auspiciante oficial de esta tesis, te quiero mamá.

A ti padre por todas las conversaciones, los consejos, el apoyo, por las noches de copas y karaoke que tanto disfrutamos juntos, por tus vivencias y tus valores, por haberme enseñado a ser una persona de bien aunque las personas que me rodeen no hagan lo mismo.

A ti ñaña, que has sido una gran amiga, gracias por estar ahí cuando te necesito, por compartir valiosos y numerosos momentos juntos, por tus noches de desvelo haciendo trabajos, en las que me acompañabas mientras hacía mi tesis y sobre todo por esta relación tan buena que tenemos como hermanos.

A ti mi querida novia por estar los últimos años a mi lado, gracias por estar en cada momento, por encontrar siempre la mejor manera para decir las cosas, por abrirme los ojos cuando crees que estoy equivocado, por tu amor y tu entrega incondicional, y por tu ayuda en la parte académica, porque juntos logramos dar cada paso de nuestra vida universitaria.

A mi abuelita Cecilia Acosta, mi querida madrina, gracias por haber sido como una madre para mí, por todo el cariño, los mimos y las golosinas, que me has dado tanto de niño como de grande, por aguantarme el sinnúmero de travesuras que hice en tu casa, por estar siempre tan pendiente de todos, por siempre estar entregando todo de ti.

A mis profesores, por haber tomado el rol de amigos, compañeros, padres y guías, ya que sin ustedes no habría la satisfacción de un logro realizado.

Y a mis amigos, por todos los momentos inolvidables, por las farras, los chupes, las amanecidas, por los momentos de alegría y tristeza que compartimos juntos, por preocuparse por mí, por apoyarme, por escucharme, por hacerme reír y llorar y sobre todo gracias por ser parte de mi vida y dejarme ser parte de la suya.

Galo Esteban Solís Gallo

ÍNDICE DE CONTENIDOS

1. CAPÍTULO I: INTRODUCCIÓN:.....	2
1.1. JUSTIFICACIÓN:.....	3
1.2. OBJETIVOS:	5
1.2.1. GENERAL:	5
1.2.2. ESPECÍFICOS:	5
1.3. ALCANCE:.....	5
1.4. METODOLOGÍA:	6
1.5. HERRAMIENTAS:.....	6
2. CAPÍTULO II: MARCO TEÓRICO	7
2.1. DEFINICIONES	7
2.1.1. Servidor Web.....	7
2.1.2. Página Web	7
2.1.3. Sitio Web.....	8
2.1.4. Aplicación Web	8
2.1.5. Sistema Web.....	9
2.1.6. Definición Help Desk	9
2.1.7. Mantenimiento de Computadoras.....	9
2.1.7.1. Mantenimiento Preventivo de Computadoras	10
2.1.7.2. Mantenimiento Correctivo de Computadoras.....	10
2.2. HELP DESK (Mesa de Ayuda).....	11
2.2.1. Recursos de un Help Desk	11
2.2.2. Procesos principales de un Help Desk.....	12
2.2.2.1. Manejo de Incidentes	13
2.2.2.2. Manejo de Problemas.....	18
2.2.2.3. Manejo de la Configuración	20
2.2.2.4. Manejo de Cambios	22

2.2.2.5.	Manejo de Liberación	23
2.3.	METODOLOGÍA: MIDAS	23
2.3.1.	Introducción	23
2.3.2.	Arquitectura MDA.....	24
2.3.2.1.	Características de MDA	24
2.3.2.2.	Modelo Independiente de Computación (CIM).....	25
2.3.2.3.	Modelo Independiente de la Plataforma (PIM)	25
2.3.2.4.	Modelo Específico de la Plataforma (PSM).....	26
2.4.	TECNOLOGÍA J2EE	26
2.4.1.	Introducción	26
2.4.2.	Características y Servicios	27
2.4.3.	Arquitectura.....	27
2.5.	EJBs	28
2.5.1.	Introducción	28
2.5.2.	Tipos de EJBs.....	29
2.5.2.1.	Session (de sesión).....	29
2.5.2.2.	Entity (de entidad).....	29
2.5.2.3.	Message Driven Bean (MDB):	29
2.5.3.	Funcionamiento	29
2.5.3.1.	Interfaz Local (“home”)	30
2.5.3.2.	Interfaz Remote.....	31
3.	CAPÍTULO III: ANÁLISIS	32
3.1.	Manual de Requerimientos	32
3.1.1.	Introducción	32
3.1.1.1.	Objetivo.....	32
3.1.1.2.	Alcance.....	32
3.1.1.3.	Definiciones, acrónimos y abreviaturas	33

3.1.1.4.	Referencias	35
3.1.1.5.	Visión general del documento	35
3.1.2.	Descripción general	35
3.1.2.1.	Perspectiva del producto	35
3.1.2.2.	Funciones del producto	38
3.1.2.3.	Características de usuario	39
3.1.2.4.	Restricciones.....	40
3.1.3.	Requisitos específicos.....	40
3.1.3.1.	Requerimientos de interfaz externo	40
3.1.3.2.	Requisitos Funcionales.....	41
3.1.3.3.	Requisitos del desarrollo	64
3.1.3.4.	Restricciones del diseño	65
4.	CAPÍTULO IV: DISEÑO	66
4.1.	Diseño de Datos.....	66
4.2.	Diseño Arquitectónico	68
4.3.	MODELAMIENTO.....	71
4.3.1.	MODELO DE DOMINIO	71
4.3.2.	MODELO DE NEGOCIO	73
4.3.3.	CASOS DE USO.....	73
4.3.3.1.	DIAGRAMAS DE CASOS DE USO.....	73
4.3.3.2.	DIAGRAMAS DE CASOS DE USO EXTENDIDO.....	74
4.3.4.	Diagramas de Actividades	80
4.3.4.1.	Administración de Usuarios	80
4.3.4.2.	Gestión de Incidentes	81
4.3.4.3.	Gestión de Configuraciones	82
4.3.4.4.	Solicitud de Servicio	83
4.3.4.5.	Genérico de Administración de Tablas	84

4.3.5.	Diagramas de Fragmentos.....	85
4.3.5.1.	Administrar Usuarios	85
4.3.5.2.	Gestión de Incidentes	86
4.3.5.3.	Gestión de Configuraciones 1 / 2.....	87
4.3.5.4.	Gestión de Configuraciones 2 / 2.....	88
4.3.5.5.	Solicitud de Servicio	89
4.3.6.	Diagramas de Navegación.....	90
4.3.6.1.	Inicio	90
4.3.6.2.	Administrar Usuarios	91
4.3.6.3.	Gestión de Incidentes	92
4.3.6.4.	Gestión de Configuraciones 1 / 2.....	93
4.3.6.5.	Solicitud de Servicio	95
4.3.7.	Modelo Objeto Relacional	96
4.3.8.	Modelo X-LINK.....	105
5.	CAPÍTULO V: IMPLEMENTACIÓN	109
5.1.	PRUEBAS.....	109
5.2.	ARQUITECTURA J2EE	110
5.2.1.	Diagrama.....	110
5.2.2.	Beans de Persistencia.....	110
5.2.3.	Beans de Sesión.....	116
5.2.4.	JSF's.....	119
5.2.4.1.	JSPs	120
5.2.4.2.	Backing Beans	121
5.3.	Despliegue en el Servidor de Aplicaciones	122
5.4.	MANUAL DE USUARIO	136
6.	CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	137
6.1.	CONCLUSIONES:.....	137

6.2. RECOMENDACIONES:.....	139
7. BIBLIOGRAFÍA	141

LISTADO DE TABLAS

Tabla 1.1: Servicios de un Help Desk.....	2
Tabla 3.1: Definiciones	33
Tabla 3.2: Abreviaturas y Acrónimos	34
Tabla 4.1: Entidades y Atributos.	66
Tabla 5.1: Resumen de Pruebas.....	109
Tabla 5.2: Anotaciones de un Bean de Persistencia	110
Tabla 5.3: Anotaciones Beans de Sesión: Archivos de definición.	116
Tabla 5.4: Anotaciones Beans de Sesión: Archivos de definición.	116

LISTADO DE FIGURAS

Figura 2.1: Help Desk	12
Figura 2.2: Gráfico Ilustrativo del proceso de Manejo de Incidentes	14
Figura 2.3: Proceso de desarrollo en MIDAS	25
Figura 2.4: Arquitectura J2EE.....	28
Figura 3.1: Entidades 1/2	69
Figura 3.2: Entidades 2/2	70
Figura 3.3: Diagrama de clases 1/2.....	71
Figura 3.4: Diagrama de clases 2/2.....	72
Figura 3.5: Diagrama de casos de uso.....	73
Figura 3.6: Diagrama de casos de uso extendido 1.....	75
Figura 3.7: Diagrama de casos de uso extendido 2.....	76
Figura 3.8: Diagrama de casos de uso extendido 3.....	77
Figura 3.9: Diagrama de casos de uso extendido 4.....	78
Figura 3.10: Diagrama de casos de uso extendido 5.....	79
Figura 3.11: Diagrama de actividades 1.....	80

Figura 3.12: Diagrama de actividades 2.....	81
Figura 3.13: Diagrama de actividades 3.....	82
Figura 3.14: Diagrama de actividades 4.....	83
Figura 3.15: Diagrama de actividades 5.....	84
Figura 3.16: Diagrama de fragmentos 1.....	85
Figura 3.17: Diagrama de fragmentos 2.....	86
Figura 3.18: Diagrama de fragmentos 3.....	87
Figura 3.19: Diagrama de fragmentos 4.....	88
Figura 3.20: Diagrama de fragmentos 5.....	89
Figura 3.21: Diagrama de navegación 1	90
Figura 3.22: Diagrama de navegación 2	91
Figura 3.23: Diagrama de navegación 3	92
Figura 3.24: Diagrama de navegación 4	93
Figura 3.25: Diagrama de navegación 5	94
Figura 3.26: Diagrama de navegación 6	95
Figura 5.1: Arquitectura del Sistema	110
Figura 5.2: Página de Administración de Usuarios	120

LISTADO DE ANEXOS

Anexo A: Manual de Usuario.....	142
Anexo B: ITIL – Incident Management	162

RESUMEN

El presente proyecto trata del desarrollo de un Sistema Web, el cual consiste en desarrollar un módulo para apoyar al funcionamiento de un Help Desk, mediante la utilización de herramientas de software libre.

La metodología aplicada es MIDAS, la cual se basa en la arquitectura MDA, ésta se divide en tres grupos de modelos: el modelo independiente de computación, el modelo independiente de plataforma y el modelo dependiente de plataforma. Los tres grupos de modelos dan las pautas necesarias para lograr un sistema estable y que cumpla con los requerimientos del negocio.

Los Help Desk, también conocidos como mesas de ayuda, son departamentos que existen dentro de una empresa, encargados de recibir y solventar todos los incidentes que se presentan dentro de la Infraestructura de las Tecnologías de la Información (TI). El objetivo de un Help Desk es dar valor agregado a la empresa, brindando servicios de gestión de incidentes, problemas, cambios, liberación y configuración. Su función es solucionar los fallos en el menor tiempo, para que no afecten al funcionamiento normal de los procesos de las TI.

Los procesos analizados para el funcionamiento del sistema se basaron en las buenas prácticas de ITIL (Information Technologies Infrastructure Library), donde se define paso a paso cómo deben ser los procesos de una mesa de servicios.

La tecnología utilizada en el desarrollo de la aplicación Web es J2EE, que es una plataforma independiente utilizada en la generación de software distribuido. El desarrollo del sistema utiliza dos tipos de EJBs, los de persistencia o entidad y los de sesión. Finalmente para la interfaz del usuario se utilizaron páginas JSF, que se componen de un archivo JSP y un archivo JAVA, llamado Backing Bean.

CAPÍTULO I: INTRODUCCIÓN:

Actualmente las personas utilizan los sistemas de información para manejar todo tipo de datos, tanto corporativos como de carácter personal. La información está distribuida en computadores personales, servidores, redes de área local y redes de área extendida. Las escuelas, colegios y universidades capacitan a los estudiantes para poder manejar los sistemas actuales y sus herramientas, con el fin de facilitar el trabajo, pero hay cierto conocimiento que no se enseña, como es la forma de enfrentar los problemas que van apareciendo al encontrar fallas en el computador, cómo evitar que estos problemas se den, o cómo realizar el mantenimiento preventivo o correctivo del computador.

La falta de comprensión sobre estos temas es un inconveniente para muchas personas, a pesar de que casi todas manejan un computador, muy pocas saben cómo enfrentar un fallo del mismo y cuando ocurre, el inconveniente principal que se presenta es que no se puede trabajar en el computador hasta que un técnico lo repare.

Por esta falta de conocimiento es que han aparecido los Help Desk, (Mesa de Servicios), con el fin de tratar de ayudar a los clientes con sus problemas típicos, optimizar el tiempo de llevar el computador a mantenimiento, solventar las dudas e inquietudes del cliente, para realizar peticiones de servicio, y con esto evitar que el técnico tenga que acercarse al usuario en cada petición de servicio que este realice.

Los Help Desk son un conjunto de servicios, proporcionados por la empresa a sus clientes internos. Entre los servicios que podemos apreciar están:

Tabla 1.1: Servicios de un Help Desk

Soporte técnico	Se da servicio de soporte técnico a los clientes internos.
Registro y seguimiento	Se realiza el registro y seguimiento de los incidentes reportados.

Solución	Se le proporciona al cliente las posibles soluciones a un fallo en los sistemas de información y en las comunicaciones, hasta solventarlo.
Capacitación	Se planifica capacitaciones a los clientes, sobre el uso de los sistemas que se utilizan en la empresa.
Investigación	Si no se conoce la solución a un error, el personal técnico realiza investigaciones acerca de nuevos errores para mantener actualizada la base de soluciones a errores conocidos.

Actualmente está en auge la utilización de Sistemas Web, los cuales son una o varias aplicaciones, que corren sobre un servidor al cual puede accederse desde cualquier parte del mundo por la Internet.

Muchas empresas utilizan los Sistemas de Información Web (SIW) para poder controlar datos como producción, ventas, compras, etc. simplemente ingresando a la página Web de su compañía desde cualquier parte del mundo con su computador y una conexión a Internet. Los sistemas Web se encuentran en funcionamiento todo el tiempo sin importar la hora en la que se lo necesite, permitiendo al usuario acceder al mismo sin la preocupación que el sistema se encuentre fuera de servicio.

En este proyecto se propone realizar un Sistema de Información Web que sirva de apoyo a una Mesa de Servicios en los procesos principales. Este sistema ayudará tanto para los clientes como a los técnicos de la mesa de servicios.

1.1. JUSTIFICACIÓN:

Si la mayoría de personas que usan un computador no saben cómo solventar por si solos los problemas e inquietudes que se presentan, la poca importancia que le dan al

mantenimiento preventivo de su computador; así como también la falta de una herramienta que provea ayuda técnica de forma rápida, eficiente y eficaz a sus problemas. Este proyecto propone como solución al problema descrito el desarrollo de un Sistema Web de Apoyo a un Help Desk.

Se van a utilizar herramientas de software libre para implementar un sistema de bajo costo, que las empresas puedan implementar en su área de help desk. Con esto también, fomentar el uso de este tipo de herramientas en los desarrolladores de sistemas con el fin de disminuir el uso de software pirata.

El objetivo de ser implementado vía Web es facilitar el acceso al sistema, desde cualquier parte, haciendo uso de todas las fortalezas y las ventajas de los sistemas distribuidos. En el presente proyecto lo principal es la realización de una interfaz que le permita a los clientes internos reportar sus incidentes a la mesa de servicios y que el personal de ésta a su vez pueda buscar en una base de datos de errores conocidos la solución al incidente planteado, con el fin de que los usuarios no paralicen su trabajo hasta que un técnico se acerque personalmente a corregir el problema. Esto beneficiaría tanto al cliente interno como a la empresa en la que se implante el sistema.

El proyecto proporcionará todas las herramientas posibles a los técnicos, para que puedan llegar a la solución del fallo y transmitan el conocimiento a los clientes internos, los que utilizarán el conocimiento entregado por el técnico del help desk para tratar de solventar su problema.

Dado que nuevos problemas van a aparecer con el tiempo, así como las nuevas soluciones a otros anteriores, el proyecto también plantea la posibilidad de incrementar la base de errores y soluciones en el sistema, con el fin de tener siempre actualizada la información que se brinda al cliente.

1.2. OBJETIVOS:

1.2.1. GENERAL:

- Realizar el análisis, diseño, desarrollo e implantación de un sistema Web, mediante la utilización de herramientas de software libre, para brindar apoyo a un Help Desk.

1.2.2. ESPECÍFICOS:

- Utilizar una metodología de desarrollo de software que se aplique al proyecto y facilite el desarrollo de sistemas Web.
- Desarrollar el marco teórico de referencia del proyecto.
- Investigar la tecnología J2EE para el desarrollo de aplicaciones empresariales con JAVA.
- Investigar las funciones principales de un Help Desk, basándose en normas ITIL.
- Realizar un manual de requerimientos del sistema utilizando el estándar IEEE 830.
- Utilizar la tecnología J2EE para el desarrollo del sistema Web.
- Generar los diagramas sugeridos en la metodología que aporten al desarrollo del sistema.
- Diseñar e implementar un Sistema Web amigable y fácil navegación para los usuarios.

1.3. ALCANCE:

En este proyecto se realizó el proceso completo de desarrollo del sistema Web mediante la metodología MIDAS, utilizando las actividades propuestas en la misma.

Se entregará la documentación referente al modelamiento de la aplicación Web

que se plantea de acuerdo a la metodología, el script de la base de datos, el código fuente que se genere en la programación, los registros de datos, con problemas y soluciones por cada caso de mantenimiento.

Además se pretende dejar el sistema implantado sobre un servidor en la Web con el fin de brindar un servicio a la comunidad.

1.4. METODOLOGÍA:

La metodología a utilizar en el proyecto es MIDAS ya que es una metodología para el desarrollo de Sistemas de Información Web de forma rápida mediante el uso de iteraciones, avances, adaptaciones y prototipos.

Además MIDAS presenta una arquitectura por capas, donde cada capa presenta una vista del sistema.

1.5. HERRAMIENTAS:

- NetBeans 6.1 - Desarrollo
- MySql - Motor de base de datos
- GlassFish - Servidor de Aplicaciones
- Open Office - Procesador de textos, hoja de cálculo y herramienta para crear presentaciones.
- Mozilla Firefox - Navegador.
- Dia 0.94 y Star UML - Diagramas, modelamiento.

CAPÍTULO II: MARCO TEÓRICO

2.1. DEFINICIONES

2.1.1. Servidor Web

Un *Servidor Web*, es un programa que utiliza el protocolo http (*hypertext transfer protocol – Protocolo de Transferencia de Hipertextos*) para la transmisión de datos, llamados hipertextos, páginas web o páginas HTML. Los cuales son textos con información compleja, como formularios, imágenes, enlaces, animaciones, videos, etc.

Este tipo de programa llamado *Servidor Web*, se ejecuta continuamente en un computador, y se encuentra en estado de espera de solicitudes por parte de los clientes (browsers) y les responde mediante páginas Web; así mismo si se encuentra un error en el proceso se le mostrará al cliente una página web informando que hubo un error al procesar su solicitud.

Existen muchos servidores web, y entre los importantes se encuentran el Apache, Apache Tomcat, Glassfish (Aplicaciones JAVA) e Internet Information Services (IIS – Aplicaciones .net).

2.1.2. Página Web

Una *Página Web*, es un documento que contiene información relacionada a un tema en especial, en este caso la información estará relacionada con el mantenimiento de computadoras.

Este documento se encuentra almacenado en un servidor web, el cual a su vez está conectado a la Internet, para que cualquier persona en el mundo, que posea un dispositivo con conexión a Internet pueda acceder a dicha información.

Una *Página Web* puede poseer documentos, gráficos, sonidos, videos, instaladores de diversos programas, imágenes de CDs o DVDs, y cualquier tipo de

archivo en general, los cuales se presentan como vínculos dentro de la *Página Web* que se direccionan a la información digital guardada dentro del servidor web.

2.1.3. Sitio Web

Un *Sitio Web*, es la localización dentro de la red de Internet de un grupo de Páginas Web (documentos) relacionadas. Los documentos almacenados dentro de un sitio Web se encuentran organizados de manera jerárquica.

Los sitios Web tienen la característica de tener una página de inicio, la cual es el primer documento que se muestra al usuario cuando este ingresa al dominio del sitio web mediante un navegador.

2.1.4. Aplicación Web

Una *Aplicación Web*, es como su nombre lo indica una aplicación informática, a la cual los usuarios para utilizarla se conectan a un servidor Web mediante Internet. Los usuarios se conectan a ella utilizando un cliente, que es un navegador Web, como Internet Explorer, Mozilla Firefox, Opera, etc.

Las *Aplicaciones Web* sirven para que exista interacción entre el usuario y la información, por lo que están construidas de tal forma que respondan a cada una de las solicitudes generadas por el usuario.

Anteriormente las *Aplicaciones Web* después de cada solicitud del usuario tenían que volver a cargar la página con el fin de poder mostrar la respuesta enviada por el servidor web, esto ocasionaba un mayor trabajo por parte del servidor, por lo que hoy en día, se está utilizando lenguajes interpretados del lado del cliente como por ejemplo PHP o AJAX, que mejoran la interacción con el usuario ya que no deben recargar la página después de cada solicitud.

Las *Aplicaciones Web* se componen comúnmente de tres capas. La primera capa es la interfaz que ve el usuario mediante el navegador web, la segunda sería el motor

desarrollado bajo una tecnología web dinámica como JSP, JSF, JavaServlets, ASP, ASPX, PHP y la tercera es la base de datos, donde se encontrará toda la información que el usuario puede solicitar.

2.1.5. Sistema Web

Un *Sistema Web*, no es más que el conjunto de aplicaciones web que se encuentran instaladas dentro de un servidor web, el cual a su vez se encuentra instalado dentro de un computador que posee conexión a Internet.

Es decir que un *Sistema Web* es un conjunto de hardware y software necesarios para que una aplicación pueda ser publicada en Internet y accedida por un usuario desde cualquier parte del mundo mediante un browser.

2.1.6. Definición Help Desk

Help desk o mesa de ayuda, es un conjunto de servicios brindados a los clientes, para poder gestionar y solucionar problemas relacionados a equipos computacionales, dando un valor agregado a la empresa.

Las mesas de ayuda, también llamadas mesas de servicios son el punto de contacto al que acuden los clientes para reportar un incidente, esto quiere decir que la mesa de servicios atiende todas las llamadas y solicitudes de los clientes.

2.1.7. Mantenimiento de Computadoras

El mantenimiento de computadoras es un proceso de limpieza que se realiza para prevenir fallas futuras. Para el mantenimiento del equipo hay que tomar en cuenta la ubicación física del mismo, así como el ambiente en el que se encuentra.

Existen dos tipos de mantenimiento, el preventivo y el correctivo, los cuales se detallan a continuación.

2.1.7.1. Mantenimiento Preventivo de Computadoras

El mantenimiento preventivo, como su nombre lo indica, es el tipo de mantenimiento que sirve para prevenir daños o fallas en el funcionamiento del computador.

Consiste en revisar en forma repetitiva, aspectos tanto de hardware como de software que pueden afectar al equipo. Estos aspectos influyen en el sistema para que este funcione de forma correcta y tenga una vida útil más prolongada. Dado que este tipo de mantenimiento protege la parte física, tiene una incidencia directa sobre la integridad de la información almacenada, la transferencia de archivos y el funcionamiento a una velocidad óptima.

Los principales elementos en el mantenimiento preventivo son:

La limpieza frecuente del ordenador, impide que las acumulaciones de polvo y grasa recalienten los componentes y reduzcan su vida útil, ya que funcionan como aislante térmico.

Limpieza de las unidades ópticas como son el CDROM y el DVDROM, con discos especiales para la limpieza de estos dispositivos.

Revisión de la integridad de datos del disco duro, con herramientas del sistema operativo como el Check Disk de Windows.

2.1.7.2. Mantenimiento Correctivo de Computadoras

El mantenimiento correctivo consiste en la reparación de uno o más componentes del computador, de forma planificada o no planificada.

El mantenimiento correctivo planificado se trata de reparar un equipo cuando se dispone del tiempo, personal, repuestos, y los documentos necesarios para realizarlo.

El mantenimiento correctivo no planificado se da típicamente tras una avería del computador, el problema que ocasiona esto a la persona que va a realizar el mantenimiento, es que no se puede saber con certeza cuál fue el inconveniente que dio lugar al fallo.

Un fallo puede darse, en la mayoría de los casos, por desgaste natural, mal trato, falta de mantenimiento preventivo o desconocimiento del manejo de un equipo.

2.2. HELP DESK (Mesa de Ayuda)

Un Help Desk, como ya se dijo anteriormente, tiene como objetivo dar soporte a los clientes ante cualquier tipo de necesidad o eventualidad que se les presente.

Además esta área debe ser el punto central de contacto con los clientes, donde estos puedan solventar sus dudas, inquietudes, problemas y además se les entregue el conocimiento acerca de sus peticiones de servicio.

El área de soporte genera un valor agregado en cualquier organización brindando soluciones y en el caso de que exista un problema que no puedan resolver, se encarga de enviar el requerimiento a otro departamento que pueda brindar la ayuda requerida, como puede ser el departamento de mantenimiento o de adquisiciones en el caso de cambio de un equipo.

2.2.1. Recursos de un Help Desk

La mesa de servicios provee recursos de gran importancia para la organización entre los que se encuentran:

Información en tiempo real sobre todos los problemas y dudas que tienen los clientes sobre las diversas operaciones que se realicen en la empresa.

Información acerca de acciones que se están tomando para resolver los problemas y dudas de los clientes, además del estado de avance en el que se encuentran.

Información histórica de todos los sucesos, para la generación de reportes que faciliten el análisis de futuras eventualidades.

2.2.2. Procesos principales de un Help Desk

La mesa de ayuda tiene cinco procesos principales de soporte al servicio de tecnologías de la información, los cuales interactúan entre ellos para dar mayor calidad a los servicios que se ofrecen.

Estos procesos son: Manejo de Incidentes, Manejo de Problemas, Gestión de Configuraciones, Administración de Cambios y Administración de Liberación. A continuación se muestra un gráfico donde se puede observar como se relacionan los diferentes procesos.

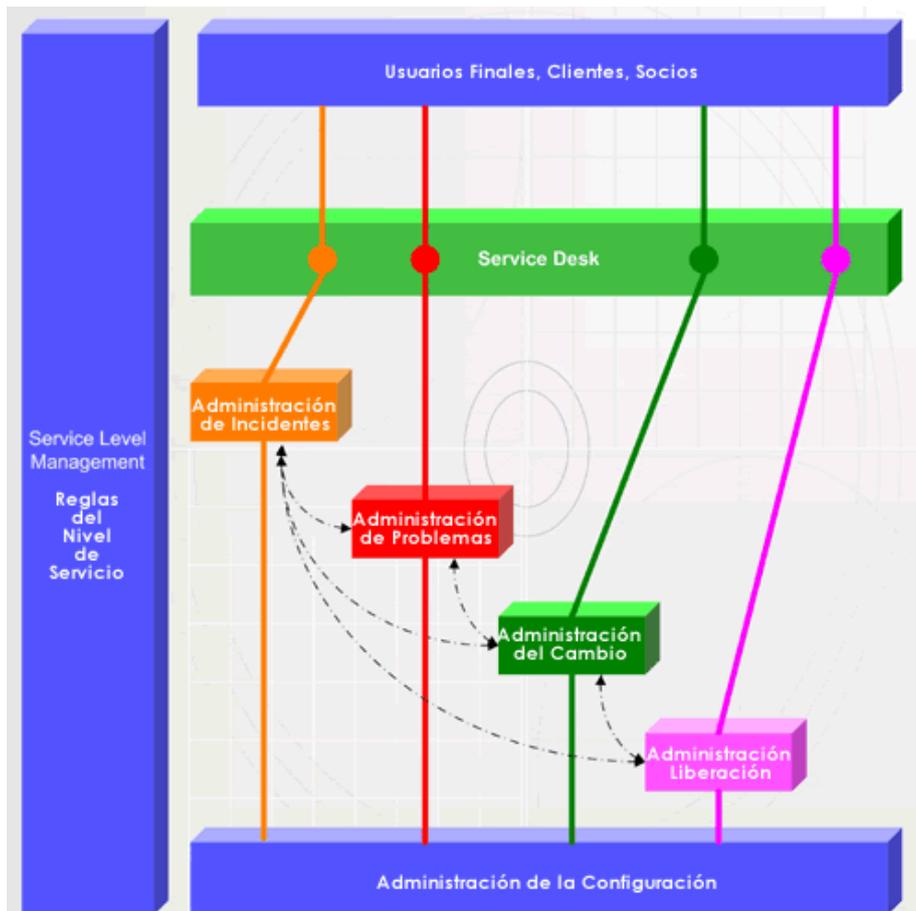


Figura 2.1: Help Desk ¹

¹ Fuente: <http://www.dric.com.mx/help-desk/itil-y-la-mesa-de-ayuda.html>

2.2.2.1. Manejo de Incidentes

Para entender lo que se realiza en esta primera etapa se debe conocer el significado de lo que es un incidente, según la terminología de ITIL un incidente se define como:

*“Any event which is not part of the standard operation of a service and which causes, or may cause, an interruption to, or a reduction in, the quality of that service. The stated ITIL objective is to restore normal operations as quickly as possible with the least possible impact on either the business or the user, at a cost-effective price”.*²

Esto significa que un incidente es:

Cualquier evento que no forma parte del funcionamiento normal de un servicio y que causa, o pueda causar, una interrupción a, o una reducción en, la calidad del servicio. El objetivo planteado por ITIL es restaurar las operaciones normales tan rápido como sea posible con menor impacto posible en el negocio o en el usuario, a un precio más que rentable.

2.2.2.1.1. Procesos incluidos en el Manejo de Incidentes

Los procesos que están incluidos en la administración de incidentes son:

- Levantar el incidente o el evento.
- Registrar el incidente.
- Clasificar el incidente y dar el soporte inicial.
- Investigación y diagnóstico.
- Solución y recuperación.
- Cerrar el incidente.
- Monitorear, rastrear y realizar seguimiento al incidente.

2 Fuente: [http://en.wikipedia.org/wiki/Incident_Management_\(ITSM\)](http://en.wikipedia.org/wiki/Incident_Management_(ITSM))

A continuación se presenta un mapa de estaciones donde cada estación es un proceso de la gestión de incidentes.

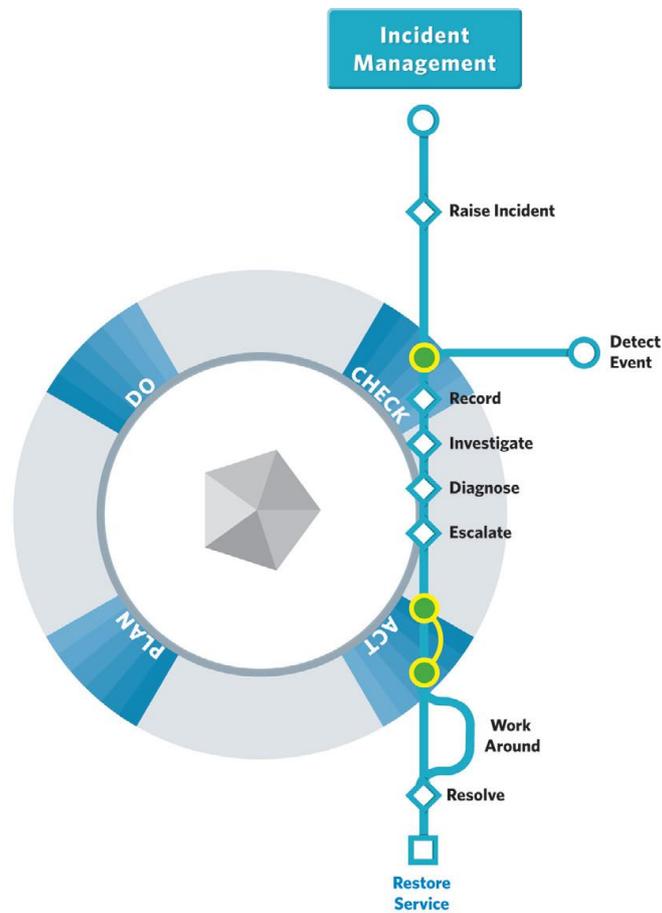


Figura 2.2: Gráfico Ilustrativo del proceso de Manejo de Incidentes ³

2.2.2.1.1.1. Levantar el incidente:

El proceso de levantar el incidente comienza con un incidente, que es reportado por un usuario a la mesa de ayuda.

Un incidente puede incluir errores de hardware, fallas de software, y también puede incluir solicitudes de servicio de usuario, que en ciertas ocasiones puede que no se encuentren relacionadas con la infraestructura de las TI.

2.2.2.1.1.2. Detectar un Evento:

³ Fuente: http://ca.com/files/technologybriefs/incident_mgmt_tb.pdf

Un evento tiene ciertas similitudes a un incidente. Lo que diferencia un evento de un incidente se centra en que un evento es detectado dentro de la infraestructura de las TI, y resulta en desviaciones del funcionamiento normal del servicio, sin embargo no es reportado por un usuario, mientras que un incidente si lo es.

2.2.2.1.1.3. Registrar el incidente:

El incidente en la mayor parte de los casos será registrado por la mesa de ayuda para que se encuentre reportado y registrado correctamente en la base de datos. De acuerdo a la naturaleza del incidente será un cliente o un sector de la infraestructura de TI la que lo reporte.

La mesa de servicios está provista con una base de datos de errores y soluciones que es la que ayuda a que los incidentes sean resueltos en el primer nivel y se realice de la manera más rápida posible. Es por esto que todos los incidentes, sin excepción, deben ser registrados en la base de datos para lograr tener un histórico exacto de los incidentes registrados, a fin de facilitar mejoras en otros procesos de Administración de Servicios de las TI (ITSM- Information Technology Service Management).

Para reducir el número de llamadas a la mesa de servicios algunas empresas proveen a los usuarios de una herramienta Web de auto-servicio para que ellos puedan solventar sus inquietudes sin requerir de personal de soporte.

2.2.2.1.1.4. Clasificar el incidente y dar el soporte inicial.

Para categorizar los incidentes al momento de ser registrados se debe:

Clasificar para poder determinar el tipo de incidente.

Realizar una comparación con otros registros para ver si un evento parecido ya ha ocurrido.

A continuación se presenta un ejemplo de cómo podría hacerse la clasificación: ⁴

⁴ Fuente: [http://en.wikipedia.org/wiki/Incident_Management_\(ITSM\)](http://en.wikipedia.org/wiki/Incident_Management_(ITSM))

Aplicación:

- El servicio no se encuentra disponible.
- Error de la aplicación.
- El uso del umbral del disco ha sido superado.

Hardware

- La computadora se apagó sola.
- Alerta de exceso de temperatura.
- La impresora no imprime

Petición de servicio

- Solicitud de información, consejo o documentación.
- El usuario olvidó su contraseña.

Los incidentes que no puedan ser resueltos de forma rápida por la mesa de ayuda pasarán a grupos especializados de técnicos de soporte. Si el incidente es una falla de un servicio con alta prioridad, se deben tomar resoluciones rápidas para devolver el servicio a su funcionamiento normal.

Luego de la clasificación del incidente se lo debe priorizar, de acuerdo a la urgencia con la que debe ser tratado. Con esto los incidentes que posean una alta prioridad serán tratados y resueltos de forma inmediata. Si un incidente es de prioridad alta y no se puede resolver, el administrador de incidentes debe crear un registro del problema e iniciar las actividades del proceso de Administración de problemas.

El soporte inicial consiste en darle al usuario una serie de posibles soluciones tomada del banco de conocimientos hasta que una de éstas logre solventar el inconveniente que éste posea.

2.2.2.1.1.5. Investigación y diagnóstico

Este proceso llega cuando no existen soluciones inmediatas al incidente y se requiere profundizar en el tema. La función de la mesa de servicios debe ser capaz de enviar el incidente a un Experto en la materia (SME).

En esta fase los analistas de soporte obtendrán los detalles del incidente actualizado. Para poder llegar a los detalles, el personal de apoyo debe tener acceso al histórico de incidentes, y a la base de errores y soluciones, que está vinculada con la mesa de servicios.

Este proceso se encuentra directamente relacionado con la Gestión de Problemas, ya que en éste se realizan las investigaciones de errores no conocidos para luego alimentar la base de datos con el problema ya solucionado.

2.2.2.1.1.6. Solución y recuperación

Este es el proceso para resolver el incidente y restaurar el funcionamiento normal del servicio.

Este proceso es la etapa final. Las principales actividades incluyen resolver el incidente con soluciones o métodos alternativos obtenidos a partir de los procesos anteriores. Para algunas soluciones, se generará una petición para el cambio (RFC), por lo que es importante la precisión y rapidez con la que se haga la transferencia de datos al proceso de gestión del cambio. Una vez que los grupos SME proveen la solución, el incidente se dirige de nuevo a la mesa de servicios. El técnico de la mesa de ayuda, que registró el incidente, se comunica con el usuario, que reportó el incidente, que el error se ha rectificado y que el incidente puede ser cerrado.

2.2.2.1.1.7. Cerrar el incidente

El cierre del incidente es el proceso que sigue a la solución y recuperación del incidente, donde el técnico de la mesa de servicios le pregunta al cliente si es que se

encuentra satisfecho con la solución del mismo, y luego de la confirmación del cliente, el técnico procede a realizar el cierre.

2.2.2.1.1.8. Monitorear, rastrear y realizar seguimiento al incidente

Lo que nos quiere decir este punto es que durante el ciclo de vida del incidente, que son todos los procesos detallados anteriormente, hay que mantener actualizada la base de datos de gestión de configuración (CMDB – Configuration Management Database), con el fin que el personal técnico de la mesa de servicios y otros involucrados dispongan información sobre el estado en el que se encuentra el incidente.

2.2.2.2. Manejo de Problemas

Al igual que en el manejo de incidentes vamos a comenzar con una pequeña definición de lo que es un problema: Un problema se define como una causa desconocida de uno o más errores, identificado como un resultado de dos o más incidentes similares.

El manejo de problemas tiene como objetivo principal resolver la causa raíz de los incidentes y eventos causados por errores dentro de la estructura de TI, con el fin de minimizar el impacto que tengan estos sobre la empresa, y así prevenir la recurrencia de incidentes relacionados a los mismos errores.

Con el concepto de problema y lo dicho del manejo de problemas, entendemos que el fin de este proceso es generar varias soluciones a errores conocidos. Un error conocido es una condición, identificada durante el análisis de la causa raíz de un problema, y el trabajo que se debe llevar a cabo para solucionarlo.

La administración de problemas, se diferencia de la administración de incidentes, en que su propósito se centra en encontrar y remediar la causa raíz de un problema y prevenir con esto futuros incidentes. Se pretende también disminuir la cantidad y la intensidad de los problemas e incidentes en el negocio. Los problemas que

son resueltos se reportan para alimentar la base de errores conocidos y soluciones para así servir a los primeros niveles de la mesa de ayuda.

Un proceso proactivo debería detectar y resolver los problemas antes de que estos sean reportados a la mesa de ayuda como incidentes. Esto se puede lograr mediante un análisis de tendencias, destinando acciones de soporte y facilitando información al resto de la organización.

Existen técnicas para encontrar la causa raíz del problema, la más conocida es realizando El manejo de la configuración es un proceso de administración de servicios, que rastrea todos los ítems individuales de configuración (CI - Configuration Items) en un sistema de TI. En otras palabras el proceso de gestión de la configuración es la implementación de una base de datos definida como CMDB (Configuration Management Database), la cual contiene detalles de los elementos del negocio que se utilizan en el aprovisionamiento y gestión de estos servicios de TI.

La información que se encontrará contenida dentro de la base de datos, estará relacionada con temas como: Mantenimiento, Movimientos, y Problemas experimentados con los ítems de configuración.

Un Ítem de Configuración es un activo de TI o una combinación de activos de TI que pueden depender y tener relaciones con otros procesos de TI. El administrador de configuración estará encargado de realizar las relaciones y asignar la jerarquía de los atributos de los ítems de configuración. Estos deben ser descompuestos hasta un nivel donde puedan ser instalados o cambiados de forma independiente. Entre los tipos de IC tenemos Hardware, Software, Sistemas, Bases de datos, Documentación, Información de Usuarios, Procesos, Incidentes, Problemas, Errores Conocidos, etc.

La base de datos CMDB, aparte de contener los IC, contiene información de otros ítems que son necesarios para una organización de TI, esta información incluye:

Hardware, Software, Documentación y Personal. Tomando como personal a los usuarios de los servicios de TI.

Para la solución de problemas, el objetivo será encontrar la causa y el efecto del problema mediante el diagrama de Ishikawa

2.2.2.2.1. Actividades:

- Las actividades del proceso de control de problemas son:
- Identificación y registro de problemas.
- Clasificación de problemas.
- Investigación y diagnóstico de Problemas.

2.2.2.3. Manejo de la Configuración

2.2.2.3.1. Actividades:⁵

La gestión de configuración, apoyándose en la información contenida en la base de datos CMDB, consiste en cinco actividades básicas:

2.2.2.3.1.1. Planificación:

El plan de la gestión de configuración se debe revisar al menos dos veces al año y debe incluir: una estrategia, políticas, ámbito, objetivos, funciones y responsabilidades, los procesos de gestión de la configuración, actividades y procedimientos, la base de datos CMDB, las relaciones con otros procesos, así como herramientas y otros recursos. El número de IC para realizar un seguimiento de las categorías en la CMDB determina el ámbito de aplicación. El detalle de la industria de la información es la profundidad.

2.2.2.3.1.2. Identificación:

Es la selección, identificación y etiquetado de todos los IC. Esto incluye el registro de información sobre los IC. Los IC deberán registrarse con un nivel de detalle

de acuerdo a las necesidades del negocio, por lo general de acuerdo al nivel de "independencia al cambio". Esto incluye la definición de las relaciones de los CI en el sistema.

2.2.2.3.1.3. Control:

El control nos da la garantía de que sólo los IC autorizados e identificados, son aceptados y registrados a partir de la recepción hasta que se los necesite. También garantiza que no se añada, modifique, reemplace o retire ningún IC, sin el adecuado control de documentación. Por ejemplo, las solicitudes aprobadas para el cambio de un CI, han cumplido con todas las condiciones. Todos los ítems de control estarán bajo el control del proceso de gestión del cambio (ITSM).

2.2.2.3.1.4. Seguimiento:

En esta etapa se lleva la cuenta y los reportes de toda la información nueva y los datos históricos relacionados con cada uno de los IC a lo largo de su ciclo de vida. Este elemento permite cambios a los IC y el seguimiento de sus registros a través de varios estados. Posibles estados podrían ser por ejemplo: Ordenado, Recibido, Bajo Pruebas, En uso, Bajo Reparación, Retirado o Para Eliminación.

2.2.2.3.1.5. Verificación:

Por último en el proceso de verificación se realizan revisiones y auditorías para verificar la existencia física de los ítems de configuración, y que se encuentren correctamente registrados en la base de datos CMDB y en la lista de partes. En la verificación se incluye el proceso de verificación de la gestión de liberación (Release Management) y la documentación del gestor de configuración antes de realizar cambios al ambiente normal de funcionamiento.

⁵ Tomado de: [http://en.wikipedia.org/wiki/Configuration_Management_\(ITSM\)](http://en.wikipedia.org/wiki/Configuration_Management_(ITSM))

2.2.2.4. Manejo de Cambios

Es de vital importancia para la eficiencia en la entrega de los servicios de TI, que los cambios sean manejados y controlados sistemáticamente, para minimizar cualquier perturbación a los servicios de TI que son entregados a los clientes.

El objetivo del proceso de gestión del cambio, es asegurar que se use métodos y procedimientos estandarizados para manejar y controlar la forma correcta en que los cambios son iniciados, evaluados, planificados, programados y ejecutados, a fin de reducir al mínimo el número y el impacto de los posibles incidentes luego de haber prestado servicio.

El proceso de manejo de cambios asimismo puede garantizar que los métodos, procesos y procedimientos, que se utilizan para realizar todos los cambios, son normalizados, y con esto facilitar la realización eficiente y rápida de todos los cambios, manteniendo el equilibrio adecuado entre la necesidad de cambio y el posible impacto negativo que estos puedan tener.

Los cambios surgen como consecuencia de problemas, pero muchos cambios pueden proceder de la búsqueda de beneficios empresariales, como la reducción de costes o la mejora de los servicios. Los detalles del proceso de cambio están documentados en los SLA (Acuerdo de nivel de servicio), para garantizar que los usuarios conozcan el procedimiento para solicitar cambios, la cantidad de veces que pueden hacerlo, y el impacto de la introducción de los mismos.

Todos los detalles de los cambios realizados deben ser comunicados a la mesa de servicios, ya que inclusive en ocasiones, después de varias capacitaciones, como por ejemplo, en el cambio de una aplicación, existen dudas y dificultades que se irán presentando.

2.2.2.5. Manejo de Liberación

El manejo de liberación es un proceso de gestión de servicios, cuyo objetivo fundamental es que asegurar que no exista ningún error en los cambios realizados a la infraestructura de TI, teniendo una visión amplia y clara de los efectos de cada cambio realizado a la infraestructura de servicios de TI.

Debe ser usado principalmente cuando existen cambios críticos o de gran escala tanto en hardware como en software. El manejo de liberación coordina los proveedores de servicios y los proveedores que se encuentran involucrados en una liberación de hardware, software y la documentación asociada.

En la mayoría de casos la gestión de liberación se entiende como el proceso responsable del manejo del desarrollo de software, instalación y soporte de los productos de software de una empresa.

2.3. METODOLOGÍA: MIDAS

2.3.1. Introducción

En la actualidad hemos visto un crecimiento exponencial del uso de Internet, la mayor parte de la población mundial tiene acceso a una cuenta de correo electrónico, los estudiantes ahora basan sus investigaciones en artículos encontrados en la Web, se ha popularizado el uso de los locales denominados “café net” para la gente que no posee un computador personal en su domicilio. Con todo esto se ha logrado que todo el mundo se encuentre conectado por esta telaraña inmensa que llamamos Internet. Todo esto ha difundido el desarrollo de Sistemas de Información Web (SIW), que hoy en día hacen desde reservas de boletos en una aerolínea, compras de artículos, seminarios en tiempo real en línea, hasta complejas transacciones bancarias.

Junto con la evolución de los Sistemas de Información Web han ido creciendo las herramientas para el desarrollo de los mismos. Sin embargo las metodologías para desarrollo de los SIW no evolucionaron con la misma velocidad, lo que ocasionó que se hagan adaptaciones a las metodologías tradicionales, pero esto ocasionaba confusión en los usuarios finales ya que la navegación dentro del SIW tendía a ser compleja y el usuario no tenían la seguridad de cuáles eran los pasos seguir para realizar una determinada operación.

2.3.2. Arquitectura MDA

MDA son las siglas de *Model Driven Architecture*, cuyo significado en español es *Arquitectura dirigida por modelos*. Fue creada en el año 2001 por OMG (*Object Management Group*)⁶.

MDA está orientada al desarrollo de aplicaciones distribuidas, enfocándose principalmente en la arquitectura del aplicativo.

Sirve para modelar software distribuido mediante el uso de diferentes modelos, de los cuales se obtienen beneficios como mejoras en la productividad, portabilidad, interoperabilidad y el mantenimiento.

2.3.2.1. Características de MDA⁷

Se enfoca en la funcionalidad y el comportamiento de un sistema o aplicación distribuida.

No interviene en las cualidades de las tecnologías donde se implementa.

Separa detalles de implementación de las funciones de negocio.

No es necesario remodelar la funcionalidad o funcionamiento del sistema o aplicación dist. Cada vez que aparece una nueva tecnología.

⁶ Página del Grupo OMG: <http://www.omg.org>

⁷ Fuente: Información proporcionada en la materia “Ingeniería de Software I”

Las principales metas de MDA son portabilidad, interoperabilidad y reusabilidad.

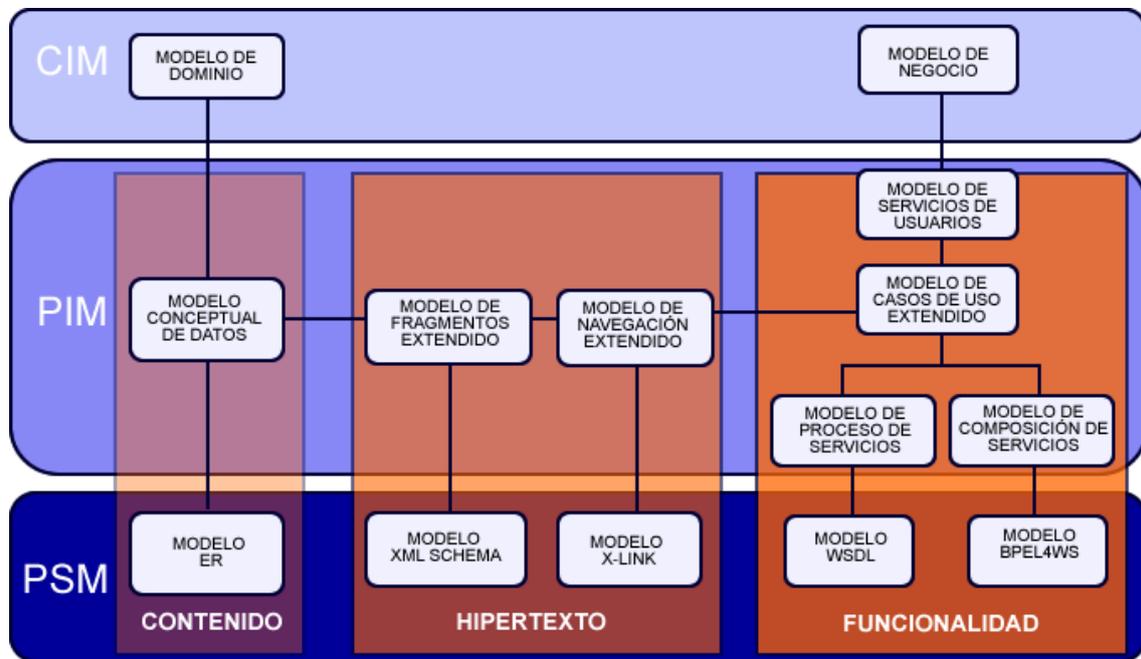


Figura 2.3: Proceso de desarrollo en MIDAS

2.3.2.2. Modelo Independiente de Computación (CIM) ⁸

Un modelo independiente de computación, es una vista global de un sistema independiente del punto computacional. Este modelo no posee detalles sobre la estructura del sistema. Es decir el modelo describe los procesos del negocio que serán resueltos en el sistema sin tener en consideración que será ejecutado por un computador.

En este modelo no se muestra detalles de la estructura del sistema.

2.3.2.3. Modelo Independiente de la Plataforma (PIM)

En el modelo PIM como su nombre lo indica, el punto de vista es independiente de la plataforma. Esto quiere decir que este es un modelo de sistema que permite representar la estructura, la funcionalidad y las restricciones del mismo sin considerar una plataforma establecida.

Este modelo es la base del desarrollo de la arquitectura MDA, ya que de aquí se pueden generar varias implementaciones en diversas plataformas, sin que estas afecten a la funcionalidad básica.

2.3.2.4. Modelo Específico de la Plataforma (PSM)

El modelo PSM combina las especificaciones halladas en el modelo PIM con los detalles que especifican la forma en que el sistema utiliza un determinado tipo de plataforma. El PSM describe la arquitectura final del sistema.

A partir de un modelo PIM se pueden obtener varios modelos PSM, en la que cada modelo específico de plataforma representa una parte específica del sistema, como un módulo Web, un componente o un esquema de base de datos.

Uno de los objetivos del PSM es llegar a los niveles más bajos para que pueda ser transformado directamente en código.

2.4. TECNOLOGÍA J2EE

2.4.1. Introducción⁹

Java 2 Platform, Enterprise Edition (*J2EE*) es una plataforma independiente, para el desarrollo, la construcción y el levantamiento de aplicaciones Web empresariales basado en Java, desarrollada por Sun Microsystems. La plataforma J2EE posee un conjunto de servicios, APIs y protocolos que proporcionan la funcionalidad para el desarrollo de aplicaciones multicapa y aplicaciones Web.

J2EE simplifica el desarrollo de aplicaciones, reduce la necesidad de programación y la capacitación por parte del programador, mediante la creación de componentes estandarizados y reutilizables. Además permite el manejo de creación automática de código, con el objetivo de que el programador se pueda centrar en la lógica del negocio.

⁸ Fuente: <http://www.omg.org/docs/omg/03-06-01.pdf>

2.4.2. Características y Servicios¹⁰

En la capa del cliente, J2EE soporta código HTML puro, aplicaciones Java y applets. Utiliza Java Server Pages (JSPs) y Servlets para crear código HTML u otro formato de datos para el cliente.

Los Enterprise JavaBeans (EJBs) proveen otra capa donde la lógica de la plataforma es almacenada. Un servidor EJB provee funciones como concurrencia, seguridad y manejo de memoria. Los servicios son transparentes para el desarrollador.

Java Database Connectivity (JDBC), que es el equivalente Java de un ODBC, es el interfaz estándar para bases de datos Java.

El API Servlet de Java aumenta la coherencia para los desarrolladores sin necesidad de una interfaz gráfica de usuario.

2.4.3. Arquitectura

La arquitectura J2EE como se dijo anteriormente sirve para crear aplicaciones multicapa, esto quiere decir que J2EE sirve para el desarrollo de sistemas distribuidos.

La arquitectura J2EE muestra cuatro capas:

Cliente: browser – celular.

Aplicaciones Web: JSPs., Servlets, JSFs.

Aplicaciones del Negocio: Ejes (Componentes).

Datos del Negocio: Se encarga del almacenamiento de datos: SGBD(sistema de gestión de base de datos).

⁹ Fuente: <http://www.java.com/en/download/faq/j2ee.xml>

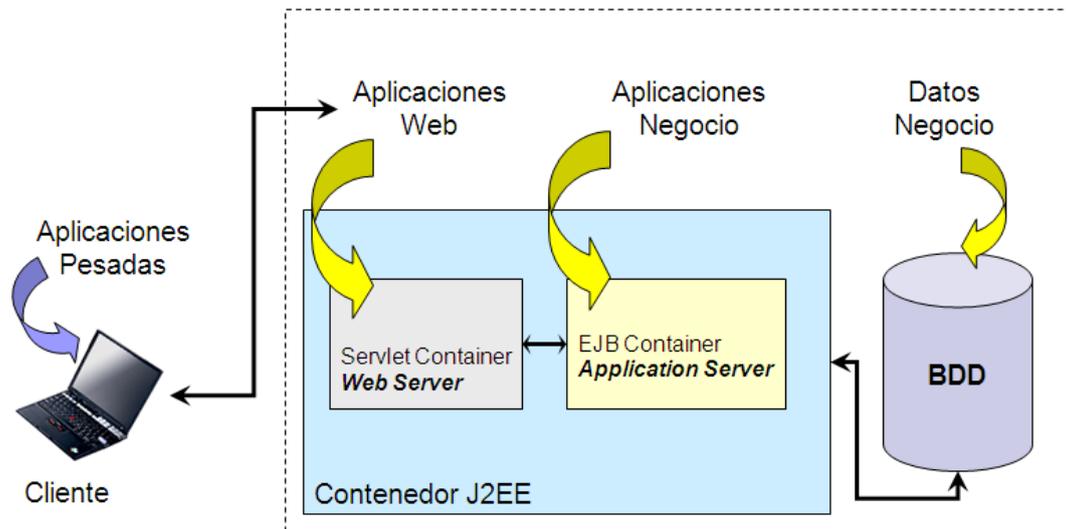


Figura 2.4: Arquitectura J2EE¹¹

2.5. EJBs

2.5.1. Introducción

Los EJB forman parte del API de la plataforma J2EE y se centran en la lógica de negocio. En los sistemas distribuidos existen cuatro capas: Cliente, Contenedor Web, Contenedor EJB y Datos. Los EJBs se encuentran ubicados dentro del contenedor EJB del servidor de aplicaciones (JBOSS, WAS, GlassFish, etc.).

El contenedor EJB es un run-time, que provee los servicios de: Manejo de transacciones, Seguridad, Concurrencia, Networking, Manejo de Recursos, Persistencia, Mensajería y Personalización en tiempo de despliegue.

Los EJBs proporcionan un modelo de componentes distribuido estándar del lado del servidor. Por estar basado en componentes permite éstos sean flexibles y reutilizables.

Los EJBs son portables a nivel de plataforma y también a nivel de contenedor. Esto quiere decir que son multiplataforma (característica de las aplicaciones Java) y que se puede llevar a otro contenedor EJB en otro servidor de aplicaciones.

¹⁰ Fuente: <http://www.webopedia.com/TERM/J/J2EE.html>

2.5.2. Tipos de EJBs

Existen tres tipos de EJBs:

2.5.2.1. Session (de sesión)

Los Beans de Sesión son utilizados para llamar a métodos del negocio y realizan una acción sobre una entidad. Existen dos tipos de Beans de Sesión: con estado y sin estado.

Con estado (stateful): son objetos distribuidos que poseen un estado, el acceso a este bean se limita solamente a un usuario. Se los utiliza para mantener información del cliente a través de varias invocaciones a métodos del negocio.

Sin estado (stateless): son objetos distribuidos que carecen de estado, por lo que permiten el acceso concurrente. No garantizan que los contenidos de las variables de la instancia mantengan su valor cuando se realice invocaciones a otros métodos.

2.5.2.2. Entity (de entidad)

Los beans de entidad sirven para representar objetos en un repositorio persistente. Estos EJBs representan una tabla (entidad) de la base de datos.

2.5.2.3. Message Driven Bean (MDB):

El funcionamiento de los MDB es asíncrono, se utilizan cuando se necesita de un consumidor JMS (*Java Messaging System*), lo que quiere decir que este tipo de bean puede escuchar mensajes JMS y se activan al recibir el mensaje. No requiere que el cliente lo invoque.

2.5.3. Funcionamiento

El funcionamiento de los EJBs comienza su proceso desde la instalación de los mismos en el contenedor EJB en el servidor de aplicaciones.

¹¹ Fuente: Tomado de la documentación del Curso: “Building EJB’s with J2EE” dictado en la EPN.

Un EJB es un conjunto de una clase de implementación y de dos interfaces Java. En la clase de implementación estará desarrollado el código de los métodos existentes en las interfaces. El contenedor crea instancias de la clase de implementación facilitando así el desarrollo de la misma. Las interfaces son llamadas desde el código cliente del EJB.

Las interfaces de un EJB son interfaz “home” e interfaz remota, y en estas se establecen las firmas de los métodos del EJB. Los métodos remotos se dividen en dos grupos de acuerdo a su función.

En la interfaz “home” se van a ubicar los métodos que no están relacionados a una instancia específica.

En la interfaz remota se ubican los métodos que se encuentran ligados a una instancia específica.

“Dado que se trata simplemente de interfaces Java y no de clases concretas, el contenedor EJB genera clases para esas interfaces que actuarán como un proxy en el cliente. El cliente invoca un método en los proxies generados que a su vez sitúa los argumentos método en un mensaje y envía dicho mensaje al servidor EJB. Los proxies usan RMI-IIOP para comunicarse con el servidor EJB.”¹²

2.5.3.1. Interfaz Local (“home”)

Este tipo de interfaz permite al código cliente manipular métodos de la clase EJB, como crear instancias de un EJB de persistencia o de sesión.

Un cliente local tiene las siguientes características:

- Se debe ejecutar en la misma máquina virtual que el EJB al que accede.
- Puede ser un componente Web u otro EJB.
- Para el cliente local, la ubicación del EJB no es transparente.

2.5.3.2. Interfaz Remote

En la interfaz remota se colocan todos los métodos de acceso público encargados de realizar operaciones (reglas del negocio).

Un cliente remoto posee las siguientes características:

Se puede ejecutar en una máquina virtual JVM distinta a la del EJB que se está accediendo.

Puede ser un componente Web, un cliente de una aplicación J2EE u otro EJB.

La localización del EJB es transparente para el cliente, es decir que el cliente desconoce la localización del EJB, este solamente lo consume.

¹² Tomado de: http://es.wikipedia.org/wiki/Enterprise_JavaBeans

CAPÍTULO III: ANÁLISIS

3.1. Manual de Requerimientos

3.1.1. Introducción

Esta sección detalla a continuación los requerimientos del software, basándose en el estándar “IEEE Recommended Practices for Requirements Specification ANSI/IEEE standard. 830, 1998”. Se describen todos los requerimientos del sistema de forma específica.

3.1.1.1. Objetivo

Detallar el funcionamiento y los requerimientos del software mediante el modelo de requerimientos de la IEEE 830, 1998, para dar al lector de este documento un enfoque claro de lo que se trata el sistema.

3.1.1.1.1. Propósito del documento

El documento de requerimientos, tiene como propósito presentar la especificación de los requisitos de software (ESR), de un sistema Web de apoyo de un HELP DESK para búsqueda de soluciones a problemas típicos y avanzados de un usuario que posea un computador.

En el documento se pretende definir los conceptos funcionales del sistema Web a realizarse.

3.1.1.1.2. Audiencia a la que va dirigido

Este documento va dirigido principalmente al usuario final y servirá como guía tanto al desarrollador como a las personas que revisen el progreso del desarrollo del sistema.

3.1.1.2. Alcance

El SIW que se propone desarrollar va a tener como nombre Mundo Help Desk. El objetivo del sistema es servir de apoyo a la mesa de servicios de una organización.

Este sistema contará con una base de errores conocidos para que los técnicos de la mesa de servicios puedan brindar ayuda de forma rápida y eficiente a los clientes.

Este sistema más específicamente ayudará a:

- Gestión de Incidentes en las actividades:
 - Registro de incidente,
 - Clasificación del incidente y
 - Cierre de incidente.
- Gestión de Configuración en las actividades:
 - Administración de la BD de incidentes,
 - Administración de la BD de problemas y
 - Administración de la BD de cambios,
 - No se profundizará en el manejo de toda la CMDB porque contempla temas de personal, proveedores, documentos administrativos, entre otras cosas que salen del alcance de esta tesis.

3.1.1.3. Definiciones, acrónimos y abreviaturas

Definiciones

Tabla 3.1: Definiciones

Entidad	Persona, institución o elemento de una clase.
Documento Anexo	Archivo digital que acompaña a un trámite y que surge como fruto del análisis realizado por el funcionario competente.
Administrador	Usuario con privilegios de establecer las configuraciones necesarias en el sistema y que le permitan registrar un trámite y canalizarlo a un funcionario para su resolución.

Cliente	Entidad que plantea un requerimiento a ser atendido por el organismo estatal. Es el gestor del trámite.
Mantenimiento	Hace referencia a las operaciones de ingreso, modificación y eliminación de datos.
Rol de usuario	Conjunto de permisos y recursos que se le asigna a un determinado usuario.
Usuario	Es una entidad del sistema a la que se le asigna un rol dentro del mismo.
HELP DESK	Mesa de ayuda.

Abreviaturas y Acrónimos

Tabla 3.2: Abreviaturas y Acrónimos

MDA	Model Driven Architecture (Arquitectura dirigida por modelos).
MIDAS	Metodología dirigida por modelos para el desarrollo ágil de Sistemas de Información Web.
ERS	Especificación de Requisitos de Software.
UML	Unified Modeling Language.
SIW	Sistemas de información Web.
BD	Base de Datos.
IC	Ítem de Configuración
TI	Tecnologías de información
CIM	Modelo Independiente de Computación
PIM	Modelo Independiente de Plataforma
PSM	Modelo Específico de Plataforma

POJO	Plain Old Java Object (Objeto Plano Java Antiguo)
SLA	Acuerdo de Nivel de Servicio

3.1.1.4. Referencias

- IEEE Recommended Practices for Requirements Specification ANSI/IEEE standard 830, 1998.
- Information Technology Infrastructure Library, Gestión de Servicios.

3.1.1.5. Visión general del documento

El documento presenta tres secciones, en las cuales se tratará de describir el funcionamiento del sistema Web a desarrollar.

En la primera se mostrará una visión general del sistema. En la segunda se presentarán las funciones específicas del funcionamiento del sistema y en la última se detallará los requisitos que deberán cumplirse en el desarrollo del sistema.

3.1.2. Descripción general

En esta parte del documento se detallarán las principales funciones del sistema a desarrollar, así como los factores que estén involucrados con el desarrollo del mismo.

3.1.2.1. Perspectiva del producto

El producto aquí descrito es un Sistema de Información Web (SIW), el cual servirá para facilitar el trabajo del personal que se ocupa en una mesa de servicios, brindándoles una herramienta que les ayude en el levantamiento, registro y clasificación de incidentes. Así mismo el sistema va a trabajar con una base de datos con información sobre errores conocidos y la solución a los mismos para que los técnicos de la mesa de ayuda den respuestas rápidas y eficientes a los usuarios.

El sistema posee una interfaz para interactuar con la BD, con el fin de poder almacenar los conocimientos sobre errores, problemas, incidentes y que los usuarios

puedan realizar las consultas respectivas sobre el tema que requieran y que tengan acceso.

Es un sistema desarrollado es independiente, lo que quiere decir que no forma parte de otro.

3.1.2.1.1. Interfaces de usuario

El sistema será desarrollado sobre una plataforma Web por lo cual la interfaz para el usuario será un browser que soporte la arquitectura definida.

3.1.2.1.2. Interfaces hardware

Para el efecto del funcionamiento de cualquier sitio Web la interfaz de hardware requerida es una tarjeta de red, puede ser esta alámbrica o inalámbrica, conectada a un modem con acceso a Internet. Los estándares utilizados, más comunes hoy en día son Ethernet, WIFI, WIMAX, entre otros.

3.1.2.1.3. Interfaces software

El sistema será implementado sobre la Internet por lo que se requiere el uso de un servidor de Aplicaciones y servidor Web para la instalación de la aplicación y navegadores Web por parte de los usuarios para la visualización.

3.1.2.1.4. Operaciones

En esta sección, se van a describir los diferentes tipos o roles de usuario, que tendrán acceso al sistema.

3.1.2.1.4.1. Modos de operación de los distintos grupos de usuarios

3.1.2.1.4.1.1. Administrador

Se requiere que la aplicación diferencie al usuario administrador y le brinde todos los privilegios sobre el sistema, entre los que estarán:

- Módulo de Manejo de Incidentes.
- Módulos de Mantenimiento de Tablas.

- Inserción, modificación y eliminación de registros.
- Manipulación de registros.
- Reportes.
 - Generación, utilización e impresión de reportes.
- Creación, modificación y eliminación de usuarios.

3.1.2.1.4.1.2. Técnico de Incidentes

Se requiere que la aplicación diferencie al usuario llamado técnico de soporte y le brinde los siguientes privilegios sobre el sistema:

- Módulo de Gestión de Incidentes.
 - Registrar incidente (Agregar registro del incidente).
 - Clasificar incidente (Modificar registro del incidente).
 - Realizar cierre de incidente (Modificar registro del incidente).
- Reportes.
 - Generación de reportes sobre incidentes pasados.
 - Generación de reportes sobre errores conocidos.

3.1.2.1.4.1.3. Técnico de Configuración

Se requiere que la aplicación diferencie al usuario llamado técnico de soporte y le brinde los siguientes privilegios sobre el sistema:

- Módulo de Gestión de Configuración:
 - Alimentar y modificar la BD con información acerca de problemas y errores conocidos.
 - Alimentar y modificar la BD con datos sobre soluciones encontradas a los diversos problemas y errores conocidos.
- Reportes:
 - Reportes sobre el estado de los incidentes.

- Reportes de datos.
- Reportes de gestión.

3.1.2.1.4.1.4. Usuario Cliente

Se requiere que la aplicación diferencie al usuario llamado cliente y le brinde los siguientes privilegios sobre el sistema:

- Módulo de Petición de Servicio:
 - Registrar la petición de servicio.
- Reportes
 - Observar el estado en el que se encuentra la solicitud o incidente.
 - Mostrar el historial de las peticiones de servicio realizadas.

3.1.2.2. Funciones del producto

Las funciones principales de la aplicación son:

Para Help Desk:

- Dar soporte a los técnicos mediante un módulo Web de manejo de incidentes.
- Llevar un control histórico de todos los incidentes registrados para poder utilizar la información antigua como base para futuras investigaciones.
- El producto recopila la información generada por los técnicos de incidentes, problemas, soluciones y cambios y los almacena en una base de datos.

Para soporte en línea:

- Brindar consejos a los usuarios para realizar mantenimiento preventivo de forma oportuna y así evitar futuros incidentes.
- Permitir a los usuarios que soliciten ayuda mediante un formulario en línea.

Para cualquier módulo:

- Un módulo de mantenimiento para la alimentación de la base de datos.
- Poseer distintos roles de usuario de acuerdo a los permisos que se requiere tener.

- Administrador.
 - Técnico de Gestión de Configuración
 - Técnico de Soporte de Incidentes.
 - Cliente.
- Autenticación y acceso para los diferentes tipos de usuario según su rol.
 - Mantener la sesión de los usuarios activa mientras navegan por el sistema.
 - Limitaciones en el acceso mediante menús para controlar el acceso a las diferentes secciones del sistema de acuerdo a los roles de usuario.

3.1.2.3. Características de usuario

La aplicación Web está dirigida a cuatro tipos de usuarios:

- Usuario de Administración: es el usuario que posee control absoluto sobre el funcionamiento del sistema, por lo que debe ser una persona de conocimientos amplios en las actividades que le corresponden a una mesa de servicios y además un conocimiento avanzado en el manejo de SIW.
- Usuario de Gestión de Incidentes: es un usuario el cual posee un nivel medio de conocimientos de computación, el cual registrará los incidentes y le dará soporte directo al cliente.
- Usuario de Gestión de Configuración: es un usuario que posee un alto nivel de conocimientos en computación, ya que es el encargado de la administración y mantenimiento de las tablas error, solución, soluciones a errores, ítems de configuración, cambio ítem de configuración, departamento y proveedor.
- Usuario Cliente: Este es un usuario cuyo nivel de conocimientos en computación suele ser bajo, pero no en todos los casos. Este usuario es del que se recibirá las peticiones de servicio.

3.1.2.4. Restricciones

Para el funcionamiento del sistema existen algunas restricciones por parte del lado del servidor, las que se detallan a continuación:

Hardware:

- Un computador con 1.6GHz de velocidad.
- Capacidad de almacenamiento de 2GB para la instalación de la aplicación y la base de datos.
- Tarjeta de red para acceso a Internet.
- Memoria RAM de 256Mb.

Software:

- Sistema operativo: LINUX, Windows, etc. con el protocolo TCP/IP instalado.
- Máquina virtual para Java versión 1.5 mínimo.
- MySql Server 5.0.45.
- Glassfish -v2ur2.

Para el lado del cliente, la única restricción es que posea un computador con un navegador de Internet instalado.

3.1.3. Requisitos específicos

En esta parte del documento se detallará de forma específica todos los requerimientos del sistema, se presentarán los requisitos de interfaces, requisitos funcionales, requisitos del funcionamiento de los procesos, entre otros puntos.

3.1.3.1. Requerimientos de interfaz externo

El sistema debe contar con algunas interfaces. Principalmente se encuentran las interfaces de ingreso de datos, búsqueda y presentación de resultados.

Ingreso de datos: El objetivo de esta interfaz es poder ingresar datos dentro de la base desde la aplicación Web, la información a ingresar será de usuarios, tipos de usuario, problemas y soluciones.

Búsquedas: Esta interfaz corresponde a las búsquedas en general. Debe permitirnos el ingreso de la palabra o frase a buscar y con algunas opciones de filtrado de información en el caso que se necesite. Además la búsqueda debe devolvernos como resultado información correcta con relación a lo que se ingresó y de acuerdo a los parámetros de búsqueda.

Presentación de resultados: Luego de ejecutada la búsqueda los resultados deben mostrarse de forma clara y ordenada para el usuario. En el caso de ser consultas de información dentro de la base en la que se requiera un reporte completo, los datos pueden ser presentados mediante tablas donde se presenten todos los registros. Para presentar los resultados de las consultas de los usuarios se generará una lista con el título de los elementos a presentar y una breve descripción del problema o solución.

3.1.3.2. Requisitos Funcionales

3.1.3.2.1. Identificación en el Sistema

- El sistema debe solicitar un nombre de usuario, que será único, y una clave de acceso. Entre estos dos elementos lo identificarán para el ingreso y el rol al que pertenece con el fin de asignar los respectivos permisos respectivos del usuario.

3.1.3.2.2. Módulo de Administración de Usuarios

- A este módulo solo podrá ingresar el usuario administrador del sistema.
- Este módulo debe permitir crear, modificar y eliminar la información de distintos usuarios y tipos de usuario guardados en la base de datos.
- La información del usuario que se necesitará es: nombre, apellido, mail, alias, contraseña y tipo usuario.

- No deben existir dos usuarios con el mismo alias.
- No deben existir dos usuarios con el mismo mail.
- Los tipos de usuario se identificarán por su nombre.
- Los tipos de usuario que existirán en el sistema son:
 - Administrador: Tendrá acceso ilimitado, es decir permisos de acceso a todos los módulos.
 - Gestión incidentes: Tendrá acceso solamente al módulo de manejo de incidentes.
 - Gestión de configuración: Tendrá acceso solamente al módulo de gestión de configuraciones.
 - Cliente: Tendrá acceso solamente al módulo de solicitud de servicio.

3.1.3.2.3. Módulo de Manejo de Incidentes

- Este módulo permitirá el acceso al usuario administrador del sistema y a los usuarios de gestión de incidentes.
- Se debe permitir la creación y modificación de incidentes.
- Los incidentes se clasificarán en los niveles:
 - Nivel 1: se puede solucionar con la base de datos de errores conocidos, no se necesita más que la intervención de los consejos técnico de soporte y el cliente.
 - Nivel 2: se necesita enviar el equipo a soporte técnico para que solucionen el error.
 - Problema: significa que no se pudo solucionar y se necesita enviar a gestión de problemas para que investiguen el error y encuentren una solución.

- Los incidentes deben ser priorizados de acuerdo al impacto y a la urgencia de ser solucionados. La prioridad será clasificada en: baja, media o alta.
- Se debe poder monitorear el estado de un incidente. Por eso se debe tener un estado del incidente que podrá ser: registrado, activo, suspendido, resuelto, cerrado.
- Los incidentes de deben ser registrados con la siguiente información: fecha y hora en la que se reportó, descripción, mensaje de error (si es que hay alguno), clasificación, prioridad, estado en el que se encuentra, solución al error, usuario que reportó la falla, usuario que registró el incidente.

3.1.3.2.4. Módulo de Manejo de Configuraciones

- A este módulo tendrán acceso el administrador del sistema y el usuario de gestión de configuraciones.
- Este módulo permitirá la administración de: Errores, Soluciones, Relaciones entre errores y soluciones, Ítems de Configuración, Equipos, Departamentos y Proveedores.
- La administración significa que se podrá agregar, modificar, eliminar y buscar registros de los elementos antes descritos.
- Los errores serán registrados en la base de datos con: código, nombre, descripción, causa, tipo de error. Un error puede tener varias soluciones.
- Las soluciones serán registradas con: código, nombre, descripción. Una solución puede servir para corregir varios errores.
- Un error puede poseer varias soluciones y una solución puede servir a varios errores. Dicho esto las relaciones entre los errores y soluciones se registrarán con: código de la relación, código del error y código de la solución.

- Un ítem de configuración será registrado con: código, nombre, tipo de ítem, proveedor, fabricante, costo, versión, año y número de serie.
- La información de la que consta un equipo es: Código, usuario, características y departamento.
- Un departamento se registrará con: código, nombre y descripción.
- Los proveedores se registrarán con: código, nombre, ruc, dirección, teléfono, fax, celular, mail y nombre del contacto.

3.1.3.2.5. Módulo de Solicitud de Servicio

- A este módulo tendrán acceso el administrador del sistema y el usuario cliente.
- Este módulo debe permitir al usuario cliente registrar su solicitud de servicio.
- El momento que el usuario registre su solicitud, este módulo debe crear el registro en la base de datos.
- La información de una solicitud de servicio será similar a la de un incidente, sus datos serán: fecha y hora a la que se registra la solicitud, descripción del error, mensaje de error (en caso que haya), usuario que realiza la petición.

3.1.3.2.6. Consultas y reportes

- Todos los módulos poseerán una opción de consultas relacionadas con los datos que se manejan en el módulo.
- Las consultas serán personalizadas, contando con diferentes opciones de búsqueda de acuerdo a la información que se quiere obtener.
- Los reportes serán presentados de forma clara mediante tablas.
- Los reportes podrán ser ordenados de acuerdo a cualquier atributo por el que se desee establecer.

3.1.3.2.7. Descripción de Casos de Uso

Nombre: Ingresar al Sistema

Descripción: Permite al usuario ingresar al sistema validando que este se encuentre registrado.
Actores: Usuario Interno / Administrador / Usuario de Mantenimiento
Precondiciones: El usuario debe constar en la base de datos caso contrario no podrá ingresar.
Flujo Normal:
El usuario ingresa su nombre de usuario (alias) y su contraseña Con esto el sistema revisa si existe en la base de datos. Si el usuario está registrado el sistema autoriza el ingreso.
Flujo Alternativo:
Si los datos ingresados fueron incorrectos no se permitirá el acceso al sistema.

Nombre: Validar Usuario
Descripción: Verifica que el usuario se encuentre registrado en el sistema.
Actores: Usuario Interno/Administrador/Usuario de Mantenimiento
Precondiciones: El usuario debe constar en la base de datos caso contrario no podrá ingresar.
Flujo Normal:
El usuario ingresa su nombre de usuario (alias) y su contraseña El sistema revisa si existe en la base de datos. El sistema informa si el nombre de usuario y contraseña son correctos.
Flujo Alternativo:
El sistema informa que el nombre de usuario y/o contraseña son incorrectos.

Nombre: Olvidó su contraseña
Descripción: Le permite al usuario recuperar su contraseña
Actores: Todos los usuarios
Precondiciones: El usuario debe constar en la base de datos.
Flujo Normal:
<p>El usuario ingresa su nombre de usuario (alias)</p> <p>El sistema revisa si existe en la base de datos.</p> <p>El sistema le hace llenar un formulario para verificar que es el usuario dueño de la cuenta.</p> <p>Se le pide al usuario que ingrese su nueva contraseña.</p>
Flujo Alternativo:
Si el usuario no contesta bien las preguntas el sistema no le dejará reestablecer la contraseña.

Nombre: Administrar Usuario
Descripción: Accede y manipula toda la información de la tabla usuario
Actores: Administrador
Precondiciones: El usuario administrador debe estar logueado en el sistema.
Flujo Normal:
<p>Se despliega el listado con toda la información de los usuarios del sistema.</p> <p>Se muestran todas las opciones para manipulación de los datos.</p> <p>El administrador escoge la tarea a realizar.</p> <p>El sistema lo actualiza en la base de datos.</p>
Flujo Alternativo:

Si la tarea escogida por el administrador da algún error el sistema lo mostrará al administrador.

Nombre: Crear Usuario

Descripción: Registra a un nuevo usuario en el sistema

Actores: Administrador

Precondiciones: El usuario administrador debe estar logueado en el sistema

Flujo Normal:

El administrador ingresa el nombre del usuario que desea agregar junto con el resto de datos del usuario.

El sistema revisa si no existe el nombre de usuario en la base de datos para que no existan duplicados.

Si el nombre de usuario está libre y es correcto el sistema verifica que todos los demás datos estén bien ingresados.

Si los datos son correctos el sistema informa que el usuario fue agregado correctamente.

Flujo Alternativo:

El sistema informa que el nombre de usuario ya existe y que debe escoger otro nombre de usuario.

Si hay campos obligatorios en blanco el sistema informa que debe llenar todos estos campos.

Si hay datos incorrectos ingresados, el sistema le informará al usuario que debe corregirlos antes de proseguir.

Nombre: Modificar Usuario

Descripción: Permite al administrador cambiar la información de cualquier usuario
Actores: Administrador
Precondiciones: El usuario administrador debe estar logueado en el sistema
Flujo Normal:
<p>El administrador busca y selecciona el usuario al que se desea modificar su información.</p> <p>El administrador modifica los datos que crea necesarios.</p> <p>El sistema verifica que no estén vacíos los campos obligatorios y que la información ingresada sea la correcta y si es así modificará los campos del usuario.</p>
Flujo Alternativo:
<p>Si la información obligatoria no fue entregada por el administrador, el sistema le informará que debe llenar todos los campos obligatorios.</p> <p>Si los datos ingresados son incorrectos el sistema informará al administrador que debe corregirlos.</p>

Nombre: Eliminar Usuario
Descripción: Permite al administrador eliminar un usuario
Actores: Administrador
Precondiciones: El usuario administrador debe estar logueado en el sistema
Flujo Normal:
<p>El administrador busca y selecciona el usuario a eliminar.</p> <p>Si es que el usuario no tiene temas ni comentarios en el foro el sistema remueve de la base de datos el usuario.</p>
Flujo Alternativo:

El usuario es eliminado.

Si el usuario había ingresado un tema en el foro se indicará que el usuario ya no es miembro del sistema pero el tema no se eliminará automáticamente.

Si el usuario había ingresado comentarios al foro se indicará que el usuario ya no es miembro del sistema pero el comentario no se eliminará automáticamente.

Nombre: Verificar alias disponible

Descripción: Verifica que el nombre de usuario escogido no se encuentre tomado por otra persona dentro el sistema.

Actores: Administrador

Precondiciones: El usuario administrador debe encontrarse logueado en el sistema

Flujo Normal:

El usuario administrador ingresa el alias del usuario a crear.

El sistema revisa si existe en la base de datos.

El sistema informa si el nombre de usuario ya está tomado.

Si el alias se encuentra disponible el sistema permitirá la creación del usuario.

Flujo Alternativo:

El sistema informa que el nombre de usuario no está disponible y que seleccione otro.

El sistema vuelve a validar si el alias se encuentra disponible.

Nombre: Buscar Usuario

Descripción: Busca uno o varios registros de usuarios de acuerdo a un criterio.

Actores: Administrador

Precondiciones: El usuario administrador debe encontrarse logueado en el sistema.

Flujo Normal:
<p>El usuario administrador escoge el criterio de búsqueda.</p> <p>El usuario administrador ingresa las palabras clave.</p> <p>El sistema devuelve todos los registros que coinciden con las palabras clave.</p>
Flujo Alternativo:
<p>El sistema informa no se encontró ningún registro relacionado de acuerdo al criterio y palabras clave ingresadas.</p> <p>Le pide al usuario que revise el criterio o las palabras clave y los vuelva a intentar.</p>

Nombre: Crear Incidente
Descripción: Le permite al usuario crear un nuevo registro de un incidente.
Actores: Administrador / Técnico de Incidentes
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
<p>El usuario ingresa los datos del incidente.</p> <p>El sistema verifica que la información ingresada sea correcta.</p> <p>El sistema crea el registro en la base de datos y le indica al usuario que se ha creado el registro correctamente.</p>
Flujo Alternativo:
<p>El sistema le indica al usuario que hubo una complicación al momento de crear el registro y que este no ha sido creado.</p>

Nombre: Modificar Incidente

Descripción: Me permite modificar la información de un incidente, principalmente para cambiar su estado o su prioridad.
Actores: Administrador / Técnico de Incidentes
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
<p>El usuario selecciona el registro a modificar.</p> <p>El usuario modifica los campos que sean permitidos por el sistema.</p> <p>El sistema verifica que la información sea correcta.</p> <p>El sistema actualiza el registro en la base de datos y muestra un mensaje que se ha modificado el registro correctamente.</p>
Flujo Alternativo:
Si hay algún error con los datos ingresados el sistema le informará al usuario que no se puede actualizar el registro y que verifique el campo que posee el error.

Nombre: Buscar Incidente
Descripción: Busca uno o varios registros de incidentes de acuerdo a un criterio.
Actores: Administrador / Técnico de incidentes
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
<p>El usuario escoge el criterio de búsqueda e ingresa las palabras clave.</p> <p>El sistema devuelve todos los registros que coinciden con las palabras clave.</p> <p>El usuario selecciona el registro de incidente que le interesa.</p>
Flujo Alternativo:

Si no se encontraron registros el sistema le solicitará al usuario que modifique su búsqueda para obtener más resultados.

Nombre: Buscar Solución

Descripción: Busca uno o varios registros de soluciones de acuerdo a un error.

Actores: Administrador / Técnico de Incidentes / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

El usuario ingresa como criterio de búsqueda el error e ingresa las palabras clave.

El sistema le muestra los errores que coinciden con su búsqueda.

El usuario selecciona el error del que desea recibir ayuda.

El sistema muestra las posibles soluciones al problema.

El usuario selecciona una solución para visualizar los pasos a seguir.

Flujo Alternativo:

Si el error no se encuentra, el sistema le solicita que modifique las palabras clave y que vuelva a intentarlo.

Nombre: Buscar Error

Descripción: Busca uno o varios registros de error de acuerdo a un criterio.

Actores: Administrador / Técnico de Incidentes / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

El usuario ingresa las palabras clave del error como criterio de búsqueda.

El sistema le muestra un listado de todos los registro de errores que coincidieron con las palabras clave.

Flujo Alternativo:

El sistema informa no se encontró ningún registro relacionado de acuerdo al criterio y palabras clave ingresadas.

Le pide al usuario que revise el criterio o las palabras clave y los vuelva a intentar.

Nombre: Crear Error

Descripción: Permite agregar un nuevo error en la sección de gestión de configuración.

Actores: Usuario Técnico de Configuración

Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema

Flujo Normal:

El usuario ingresa los datos del error.

El sistema verifica que los datos del error sean correctos.

Si los datos ingresados son correctos entonces el sistema actualizará la base de datos y le informará que el error fue agregado correctamente.

Flujo Alternativo:

Si el nombre del error ya existe se le informará que ese error ya existe y que si desea ingresar otro problema.

Si hay campos obligatorios vacíos se le informará que debe llenar los campos obligatorios.

Nombre: Modificar Error

Descripción: Permite modificar un error en la sección de manejo de configuración.
Actores: Usuario Técnico de Configuración
Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema
Flujo Normal:
<p>El usuario busca y selecciona el error que desea modificar.</p> <p>El sistema verifica que los datos del error sean correctos.</p> <p>Si los datos ingresados son correctos entonces el sistema actualizará la base de datos y le informará que el error fue agregado correctamente.</p>
Flujo Alternativo:
<p>Si el nombre del error ya existe se le informará que ese error ya existe y que si desea ingresar otro problema.</p> <p>Si hay campos obligatorios vacíos se le informará que debe llenar los campos obligatorios.</p>

Nombre: Eliminar Error
Descripción: Permite eliminar un error en la sección de gestión de configuración.
Actores: Usuario Técnico de Configuración
Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema
Flujo Normal:
<p>El usuario busca y selecciona el error que desea eliminar.</p> <p>El sistema verifica que no existan relaciones del registro de error con la tabla de soluciones.</p> <p>Si el error no tiene relaciones con soluciones dentro de la base de datos se eliminará.</p>
Flujo Alternativo:

Si el error contiene una o varias relaciones dentro de la base de datos, no se permitirá su eliminación y se le informará que primero debe eliminar los registros relacionados con ese error.

Nombre: Crear Solución

Descripción: Permite agregar un registro de solución en la sección de gestión de configuración.

Actores: Usuario Técnico de Configuración

Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema

Flujo Normal:

El usuario ingresa los datos de la solución.

El sistema verifica que los datos de la solución sean correctos.

Si los datos ingresados son correctos entonces el sistema actualizará la base de datos y le informará que la solución fue agregada correctamente.

Flujos Alternativos:

Si hay campos obligatorios vacíos se le informará que debe llenar los campos obligatorios.

Nombre: Modificar Solución

Descripción: Permite modificar una solución a un problema en la sección de gestión de configuración.

Actores: Usuario Técnico de Configuración

Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema

Flujo Normal:

El usuario busca y selecciona la solución.

El usuario modifica los datos del registro de la solución.

El sistema verifica que los datos de la solución sean correctos.

Si los datos ingresados son correctos entonces el sistema actualizará la base de datos y le informará que la solución fue modificada correctamente.

Flujo Alternativo:

Si hay campos obligatorios vacíos se le informará que debe llenar los campos obligatorios.

Nombre: Eliminar Solución

Descripción: Permite eliminar una solución a un error en la sección de gestión de configuración.

Actores: Usuario Técnico de Configuración

Precondiciones: El usuario Técnico de Configuración debe estar logueado en el sistema

Flujo Normal:

El usuario busca y selecciona la solución a eliminar.

El sistema verifica que el registro de la solución no tenga relaciones con la tabla de errores.

Si el registro de la solución seleccionada no tiene relaciones el sistema actualizará la base de datos y le informará que la solución fue eliminada.

Flujo Alternativo:

Si la solución contiene una o varias relaciones dentro de la base de datos, no se permitirá su eliminación y se le informará que primero debe eliminar los registros relacionados con la solución antes de eliminar el registro.

Nombre:	Crear Error - Solución
Descripción:	Permite agregar una solución a un problema en la sección de gestión de configuración.
Actores:	Usuario Técnico de Configuración
Precondiciones:	El usuario Técnico de Configuración debe estar logueado en el sistema
Flujo Normal:	
	El usuario busca y selecciona un error, al que desea agregar una solución. El usuario busca en la tabla de soluciones el registro con el que desea crear la relación. El sistema crea la relación entre las dos tablas en una tercera llamada error solución.
Flujos Alternativos:	
	Si existe algún error al crear la relación el sistema le informará al usuario que se dio un error, y la relación no ha sido creada.

Nombre:	Crear Ítem de Configuración
Descripción:	Le permite al usuario crear un nuevo registro de un Ítem de Configuración.
Actores:	Administrador / Técnico de Configuración
Precondiciones:	El usuario debe encontrarse logueado en el sistema.
Flujo Normal:	
	El usuario ingresa los datos del Ítem de Configuración y selecciona un proveedor. El sistema verifica que la información ingresada sea correcta. El sistema crea el registro en la base de datos y le indica al usuario que se ha creado el registro correctamente.

Flujo Alternativo:

El sistema informa que existen todos los campos obligatorios deben ser llenados.

Nombre: Modificar Ítem de Configuración

Descripción: Le permite al usuario modificar un registro de un Ítem de Configuración.

Actores: Administrador / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

El usuario busca y selecciona el registro del Ítem de Configuración a modificar.

El usuario cambia los datos de la aplicación.

El sistema verifica que la información ingresada sea correcta.

El sistema modifica el registro en la base de datos y le indica al usuario que el registro se ha modificado correctamente.

Flujo Alternativo:

El sistema informa que existen todos los campos obligatorios deben ser llenados.

Nombre: Eliminar Ítem de Configuración

Descripción: Le permite al usuario eliminar un registro de un Ítem de Configuración

Actores: Administrador / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

El usuario busca y selecciona el registro del Ítem de Configuración a eliminar.

El sistema le pregunta si está seguro de eliminar el registro.

El usuario confirma la eliminación.

El sistema elimina el registro y le informa al usuario.
Flujo Alternativo:
Si el usuario no confirma la eliminación, el registro permanece en la base de datos.

Nombre: Buscar Ítem de Configuración
Descripción: Busca uno o varios registros de los Ítems de Configuración de acuerdo a un criterio.
Actores: Administrador / Técnico de Configuración
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
El usuario selecciona el criterio de búsqueda para encontrar el Ítem de Configuración e ingresa las palabras clave. El sistema le muestra un listado de los registros que coinciden con el criterio de búsqueda ingresado.
Flujo Alternativo:
El sistema informa no se encontró ningún registro relacionado de acuerdo al criterio y palabras clave ingresadas. Le pide al usuario que revise el criterio o las palabras clave y los vuelva a intentar.

Nombre: Crear Proveedor
Descripción: Le permite al usuario crear un nuevo registro de un proveedor.
Actores: Administrador / Técnico de Configuración
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:

<p>El usuario ingresa los datos del proveedor.</p> <p>El sistema verifica que no se encuentre el proveedor registrado mediante el Ruc.</p> <p>Si no hay un Ruc igual al ingresado el sistema verifica que la información ingresada sea correcta.</p> <p>El sistema crea el registro en la base de datos y le indica al usuario que se ha creado el registro correctamente.</p>
<p>Flujo Alternativo:</p>
<p>Si el Ruc ya está asignado a un registro, el sistema le informa que ya existe registrado el un proveedor con ese Ruc.</p> <p>Si la información es incorrecta el sistema le informará al usuario que la revise y la corrija.</p> <p>El sistema informa que existen todos los campos obligatorios deben ser llenados.</p>

<p>Nombre: Modificar Proveedor</p>
<p>Descripción: Le permite al usuario modificar un registro de un proveedor.</p>
<p>Actores: Administrador / Técnico de Configuración</p>
<p>Precondiciones: El usuario debe encontrarse logueado en el sistema.</p>
<p>Flujo Normal:</p>
<p>El usuario busca y selecciona el registro del proveedor a modificar</p> <p>El usuario cambia los datos del proveedor.</p> <p>El sistema verifica que la información ingresada sea correcta.</p> <p>El sistema modifica el registro en la base de datos y le indica al usuario que el registro se ha modificado correctamente.</p>
<p>Flujo Alternativo:</p>

Si la información es incorrecta el sistema le informará al usuario que la revise y la corrija.

El sistema informa que existen todos los campos obligatorios deben ser llenados.

Nombre: Eliminar Proveedor

Descripción: Le permite al usuario eliminar un registro de un proveedor.

Actores: Administrador / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

El usuario busca y selecciona el registro del proveedor a eliminar.

El sistema le pregunta si está seguro de eliminar el registro.

El usuario confirma la eliminación.

El sistema elimina el registro y le informa al usuario.

Flujo Alternativo:

Si el proveedor posee una relación con la tabla de Ítems de configuración se le informará al usuario que el registro no puede ser eliminado hasta que no posea relaciones.

Si el usuario no confirma la eliminación, el registro permanece en la base de datos.

Nombre: Buscar Proveedor

Descripción: Busca uno o varios registros de proveedores de acuerdo a un criterio.

Actores: Administrador / Técnico de Configuración

Precondiciones: El usuario debe encontrarse logueado en el sistema.

Flujo Normal:

<p>El usuario selecciona el criterio de búsqueda para encontrar el proveedor e ingresa las palabras clave.</p> <p>El sistema le muestra un listado de los proveedores que coinciden con el criterio de búsqueda ingresado.</p>
<p>Flujo Alternativo:</p>
<p>El sistema informa no se encontró ningún registro relacionado de acuerdo al criterio y palabras clave ingresadas.</p> <p>Le pide al usuario que revise el criterio o las palabras clave y los vuelva a intentar.</p>

<p>Nombre: Crear Departamento</p>
<p>Descripción: Le permite al usuario crear un nuevo registro de un Departamento.</p>
<p>Actores: Administrador / Técnico de Configuración</p>
<p>Precondiciones: El usuario debe encontrarse logueado en el sistema.</p>
<p>Flujo Normal:</p> <p>El usuario ingresa los datos del Departamento.</p> <p>El sistema verifica que no se encuentre el Departamento registrado mediante el Nombre.</p> <p>Si no hay un Nombre del departamento igual al ingresado el sistema crea el registro en la base de datos y le indica al usuario que se ha creado el registro correctamente.</p>
<p>Flujo Alternativo:</p> <p>Si el Nombre ya está asignado a un registro, el sistema le informa que ya existe registrado el un Departamento con ese Nombre.</p> <p>El sistema informa que existen todos los campos obligatorios deben ser llenados.</p>

<p>Nombre: Modificar Departamento</p>
--

Descripción: Le permite al usuario modificar un registro de un Departamento.
Actores: Administrador / Técnico de Configuración
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
<p>El usuario busca y selecciona el registro del Departamento a modificar</p> <p>El usuario cambia los datos del Departamento.</p> <p>El sistema modifica el registro en la base de datos y le indica al usuario que el registro se ha modificado correctamente.</p>
Flujo Alternativo:
El sistema informa que existen todos los campos obligatorios deben ser llenados.

Nombre: Eliminar Departamento
Descripción: Le permite al usuario eliminar un registro de un Departamento.
Actores: Administrador / Técnico de Configuración
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
<p>El usuario busca y selecciona el registro del Departamento a eliminar.</p> <p>El sistema le pregunta si está seguro de eliminar el registro.</p> <p>El usuario confirma la eliminación.</p> <p>El sistema elimina el registro y le informa al usuario.</p>
Flujo Alternativo:
Si el Departamento posee una relación con una de las tablas de Usuario o Equipo se le informará al usuario que el registro no puede ser eliminado hasta que no posea relaciones.

Si el usuario no confirma la eliminación, el registro permanece en la base de datos.

Nombre: Buscar Departamento
Descripción: Busca uno o varios registros de Departamentos de acuerdo a un criterio.
Actores: Administrador / Técnico de Configuración
Precondiciones: El usuario debe encontrarse logueado en el sistema.
Flujo Normal:
El usuario selecciona el criterio de búsqueda para encontrar el Departamento e ingresa las palabras clave. El sistema le muestra un listado de los Departamentos que coinciden con el criterio de búsqueda ingresado.
Flujo Alternativo:
El sistema informa no se encontró ningún registro relacionado de acuerdo al criterio y palabras clave ingresadas. Le pide al usuario que revise el criterio o las palabras clave y los vuelva a intentar.

3.1.3.3. Requisitos del desarrollo

3.1.3.3.1. Tecnología de desarrollo

La tecnología de desarrollo es la plataforma J2EE, creada por la Sun Microsystems. Es una herramienta de software libre para el desarrollo de aplicaciones empresariales del lenguaje JAVA.

3.1.3.3.2. Motor de Base de datos

El motor de base de datos a utilizar es MySQL, con el administrador MySQL Administrator para la creación y mantenimiento de la base de datos y de sus tablas.

3.1.3.3.3. Servidor de Aplicaciones

El servidor de aplicaciones a utilizar es el Glassfish. Es un servidor de aplicaciones que incorpora un servidor Web, el cual es necesario para subir las páginas JSP y los EJB's. Las páginas JSP son las creadas con la tecnología JAVA y serán las que ejecutan todas funciones del sistema. Además funciona como contenedor de EJB.

Los EJB's son un grupo de componentes que funcionan del lado del servidor, para generar un ambiente distribuido. Es una herramienta que minora la carga de trabajo que va a tener el programador ayudándolo con temas como persistencia, transacciones, seguridad, entre otras, lo que le permite centrarse en desarrollar la lógica del negocio.

3.1.3.4. Restricciones del diseño

3.1.3.4.1. Diagramas

Para la realización de diagramas de flujo, de actividades, y demás diagramas que se necesite realizar, se utilizarán los programas DIA, StarUML y NetBeans.

CAPÍTULO IV: DISEÑO

4.1. Diseño de Datos

En esta sección se visualizarán las estructuras de datos necesarias para la implementación del sistema. Las entidades encontradas son:

Tabla 4.1: Entidades y Atributos.

<i>USUARIO</i>	<i>ERROR</i>	<i>TIPO USUARIO</i>
Código	Código	Código
Nombre	Nombre	Nombre
Apellido	Descripción	Descripción
Mail	Causa	
Alias	Tipo de Error	
Contraseña		
Tipo de usuario		
Departamento		

<i>TIPO ERROR</i>	<i>SOLUCIÓN</i>	<i>ERROR – SOLUCIÓN</i>
Código	Código	Código Error Solución
Nombre	Nombre	Código Error
Descripción	Descripción	Código Solución

<i>ÍTEM CONFIGURACIÓN</i>	<i>PROVEEDOR</i>	<i>TIPO ÍTEM</i>
Código	Código	Código
Nombre	Nombre	Nombre
Tipo	Ruc	Descripción
Proveedor	Dirección	
Fecha Adquisición	Teléfono	
Desarrollador	Fax	
Costo	Celular	
Versión	Mail	
Año de liberación	Contacto	
No. Serie		

<i>INCIDENTE</i>	<i>HISTORIAL INCIDENT.</i>	<i>CIERRE INCIDENTE</i>
Código	Código	Identificador
Fecha	Incidente	Código
Descripción	Estado	Usuario
Mensaje de Error	Prioridad	Características
Clasificación	Fecha actualización	
Estado	Usuario	
Prioridad	Descripción	
Solución al error		
Usuario Cliente		
Usuario Técnico		

<i>DEPARTAMENTO</i>	<i>CAMBIO ÍTEM</i>	<i>EQUIPO</i>
Código	Código	Identificador
Nombre	Descripción	Código
Descripción	Motivo	Usuario
	IC Anterior	Características
	IC Nuevo	Departamento
	Equipo	
	Fecha	

4.2. Diseño Arquitectónico

Aquí se muestran las relaciones entre las entidades estructurales existentes obtenidas del diseño de datos.

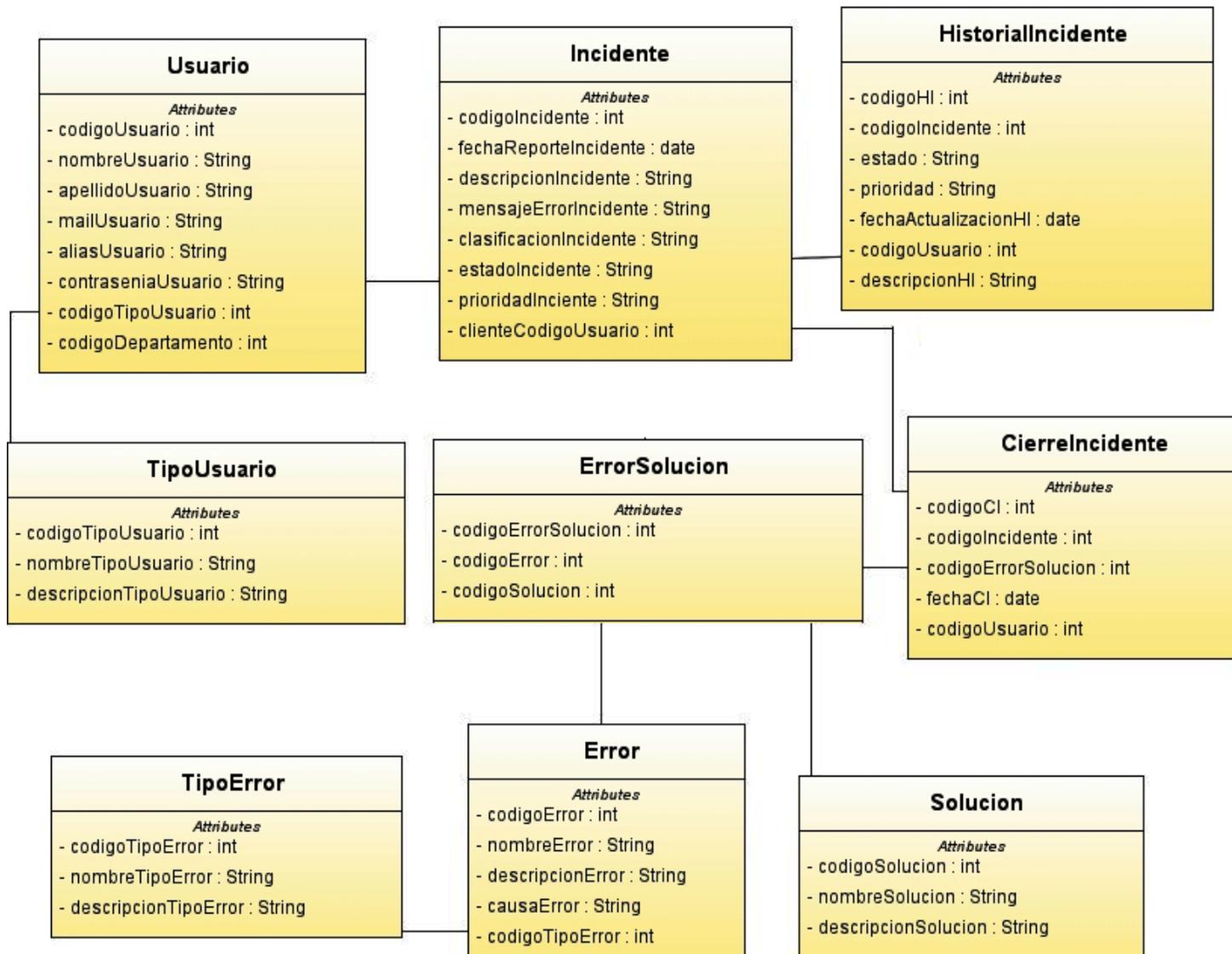


Figura 3.1: Entidades 1/2

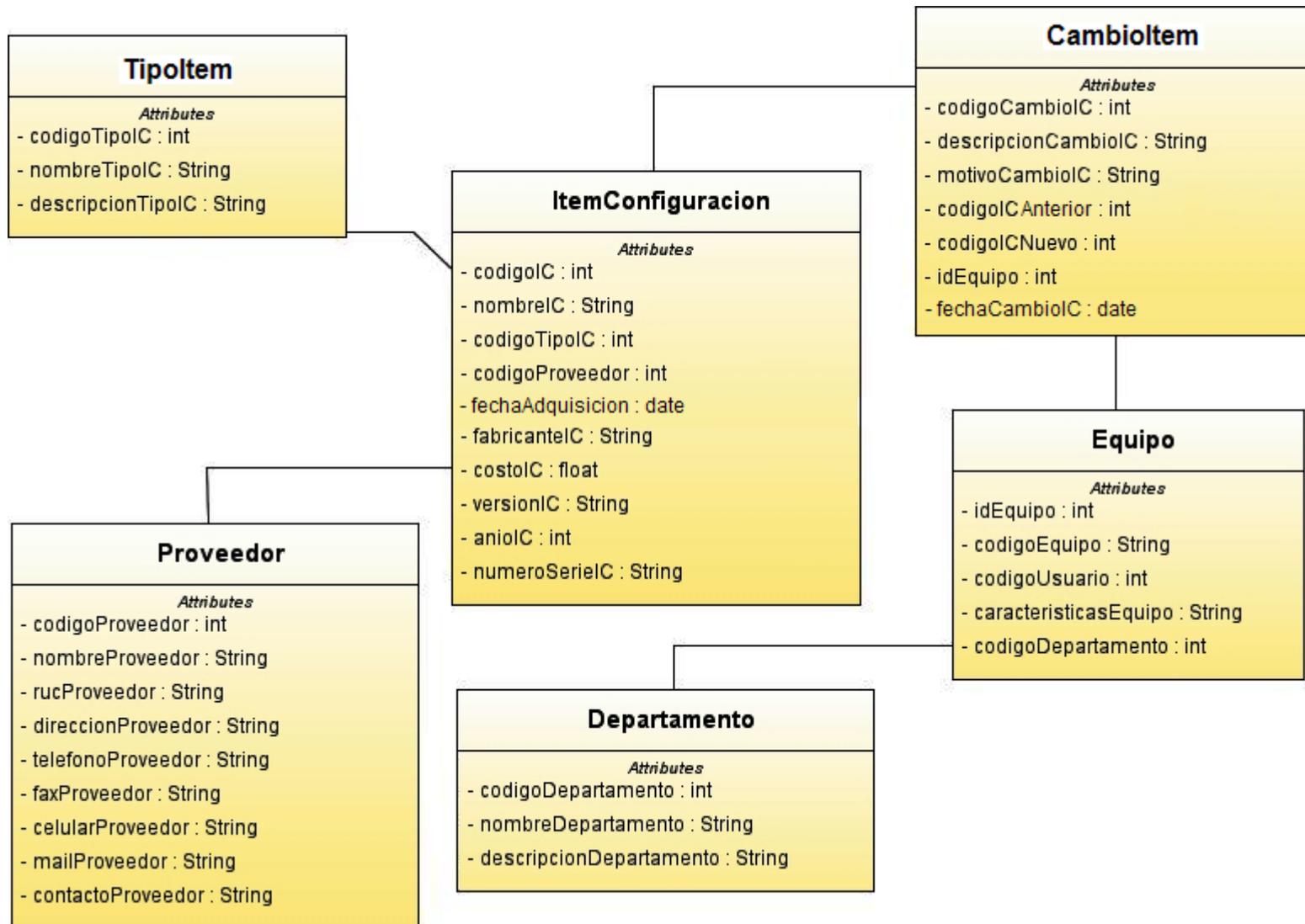


Figura 3.2: Entidades 2/2

4.3. MODELAMIENTO

4.3.1. MODELO DE DOMINIO

El modelo de dominio tiene como objetivo presentar gráficamente las entidades con sus atributos y operaciones.

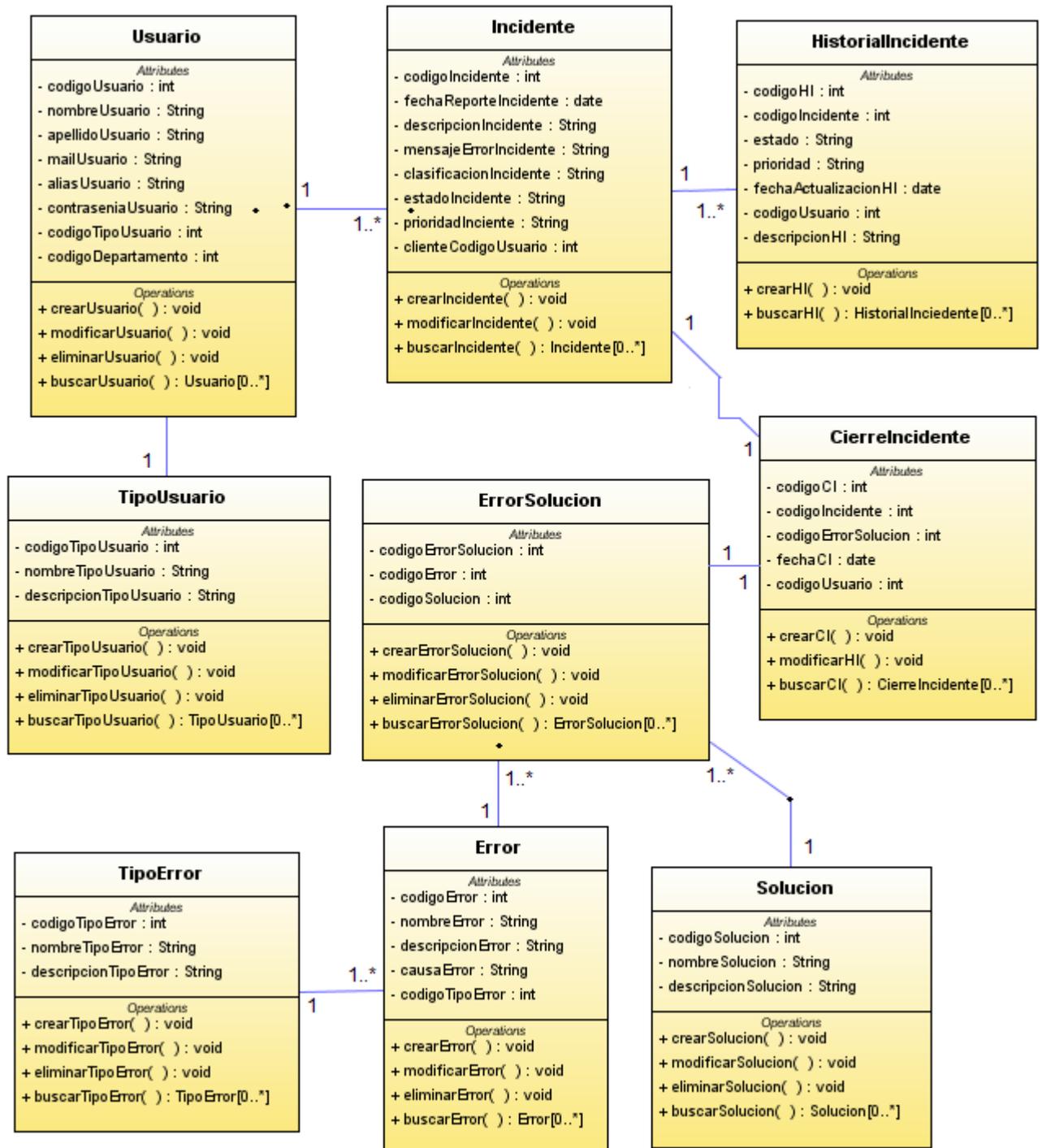


Figura 3.3: Diagrama de clases 1/2

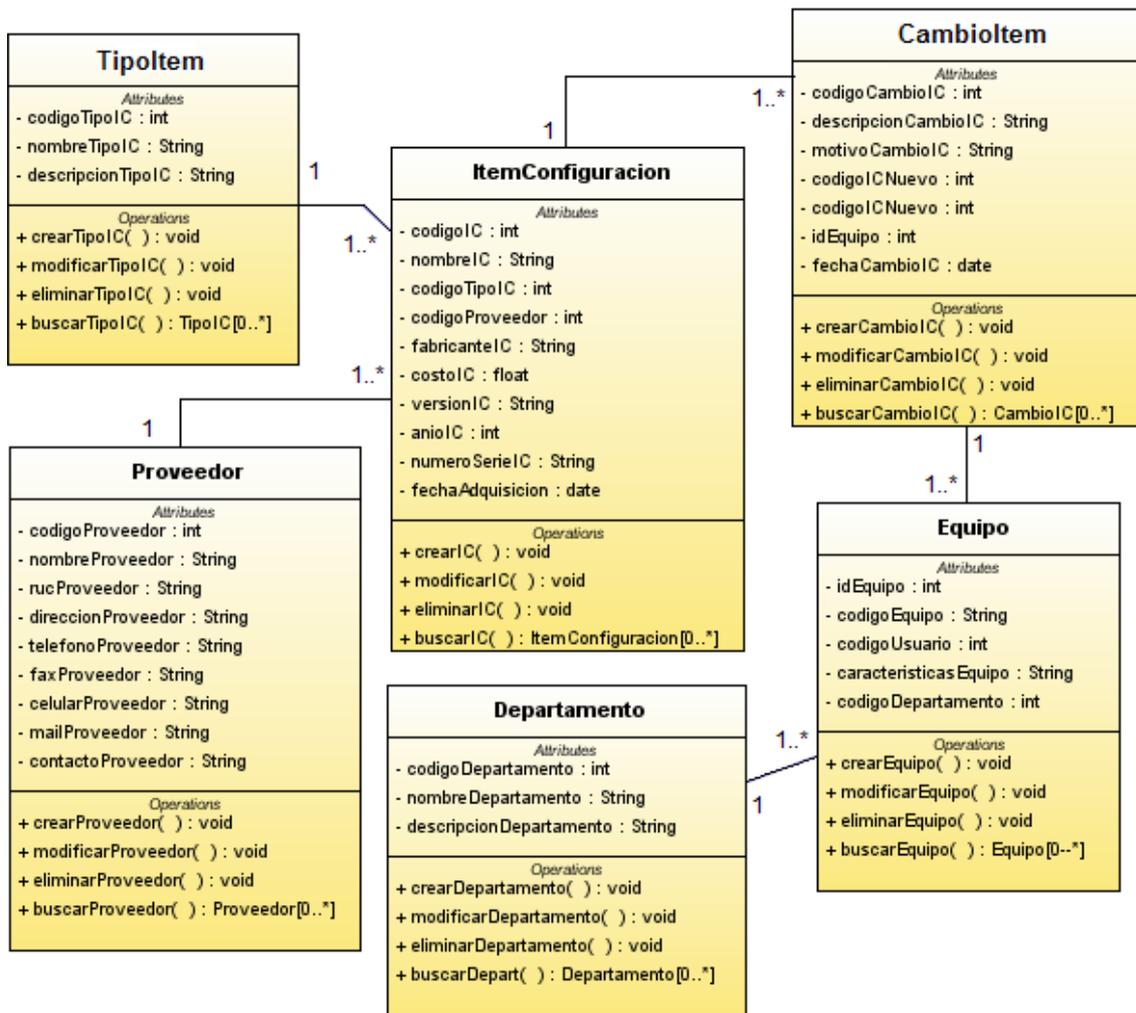


Figura 3.4: Diagrama de clases 2/2

4.3.2. MODELO DE NEGOCIO

En el Modelo de Negocio se presentan los procesos que realizará cada actor del sistema, descritos con sus respectivas precondiciones, flujo normal y flujo alternativo. Después se los presentará en forma de diagramas para visualizar de mejor manera lo descrito en palabras.

4.3.3. CASOS DE USO

4.3.3.1. DIAGRAMAS DE CASOS DE USO

Los diagramas de casos de uso muestran de forma gráfica a los actores del sistema con la funcionalidad de cada uno. La funcionalidad es el caso de uso.

El actor se mostrará con la forma de un muñeco con su nombre en la parte inferior. Los procesos relacionados con el actor se graficarán como elipses con el nombre del proceso en el interior de la forma. Las relaciones el actor y los procesos se representan con una línea sólida y las relaciones entre procesos con líneas entrecortadas. Los procesos necesarios para la ejecución de otro se los graficará como elipses.

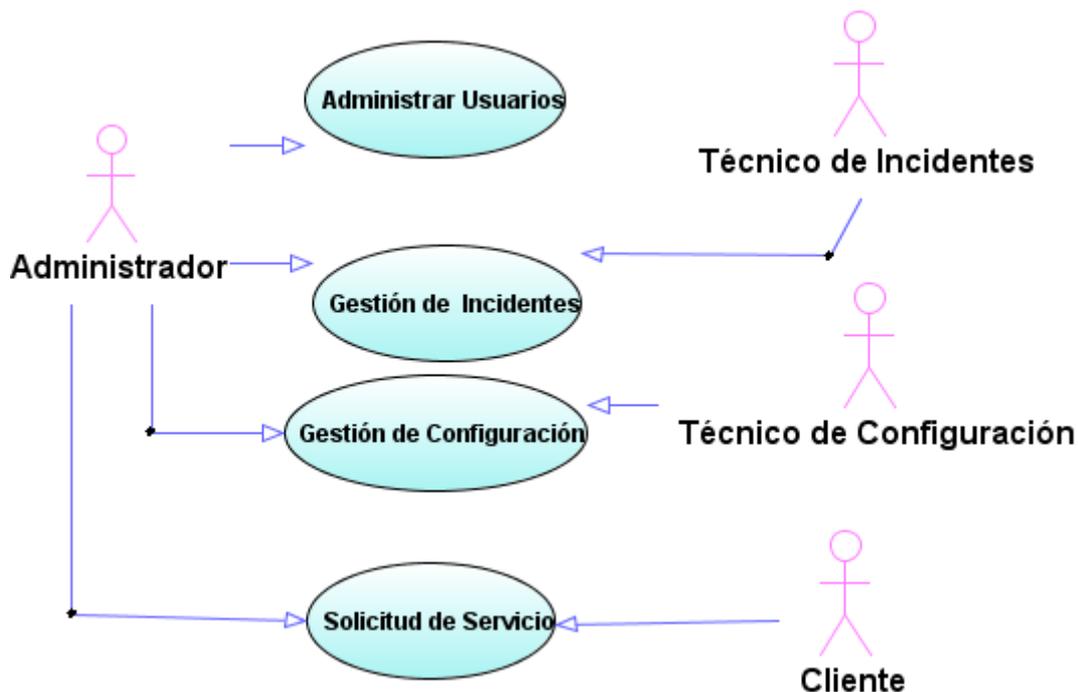


Figura 3.5: Diagrama de casos de uso

4.3.3.2. DIAGRAMAS DE CASOS DE USO EXTENDIDO

Los diagramas de casos de uso extendidos tendrán la misma notación gráfica que los diagramas de casos de uso simples. El objetivo de estos diagramas extendidos es mostrar todos los procesos y actividades que deben existir para el funcionamiento del sistema. Los procesos se encuentran relacionados entre ellos con relaciones Include y Extend.

Las relaciones que estén denotadas con <<include>> son las que usan el caso de uso hijo, en estos casos el caso de uso padre utiliza siempre el comportamiento del caso de uso hijo. Las relaciones de tipo <<extend>> son las que se denota una especialización de caso de uso, o en otras palabras cuando un caso de uso es parecido a otro pero posee una variación en su funcionamiento.

3.3.3.2.1. Administrar Usuarios

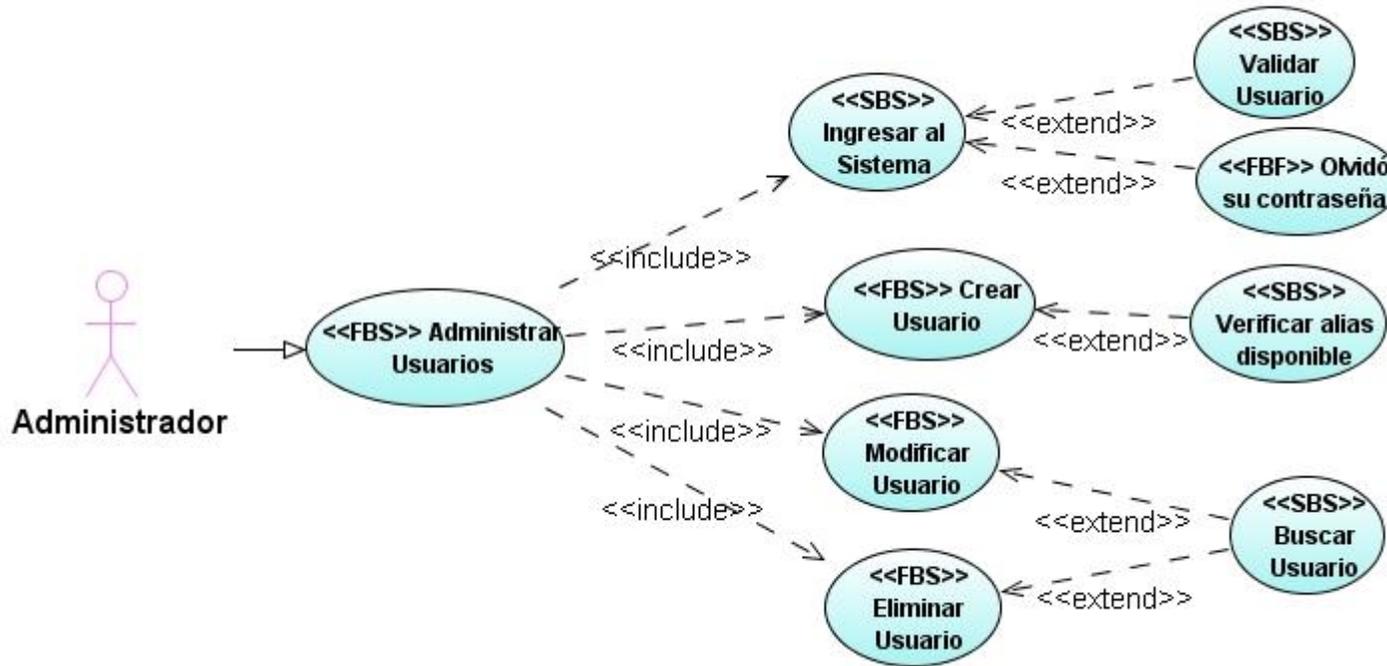


Figura 3.6: Diagrama de casos de uso extendido 1

3.3.3.2.2. Gestión de Incidentes

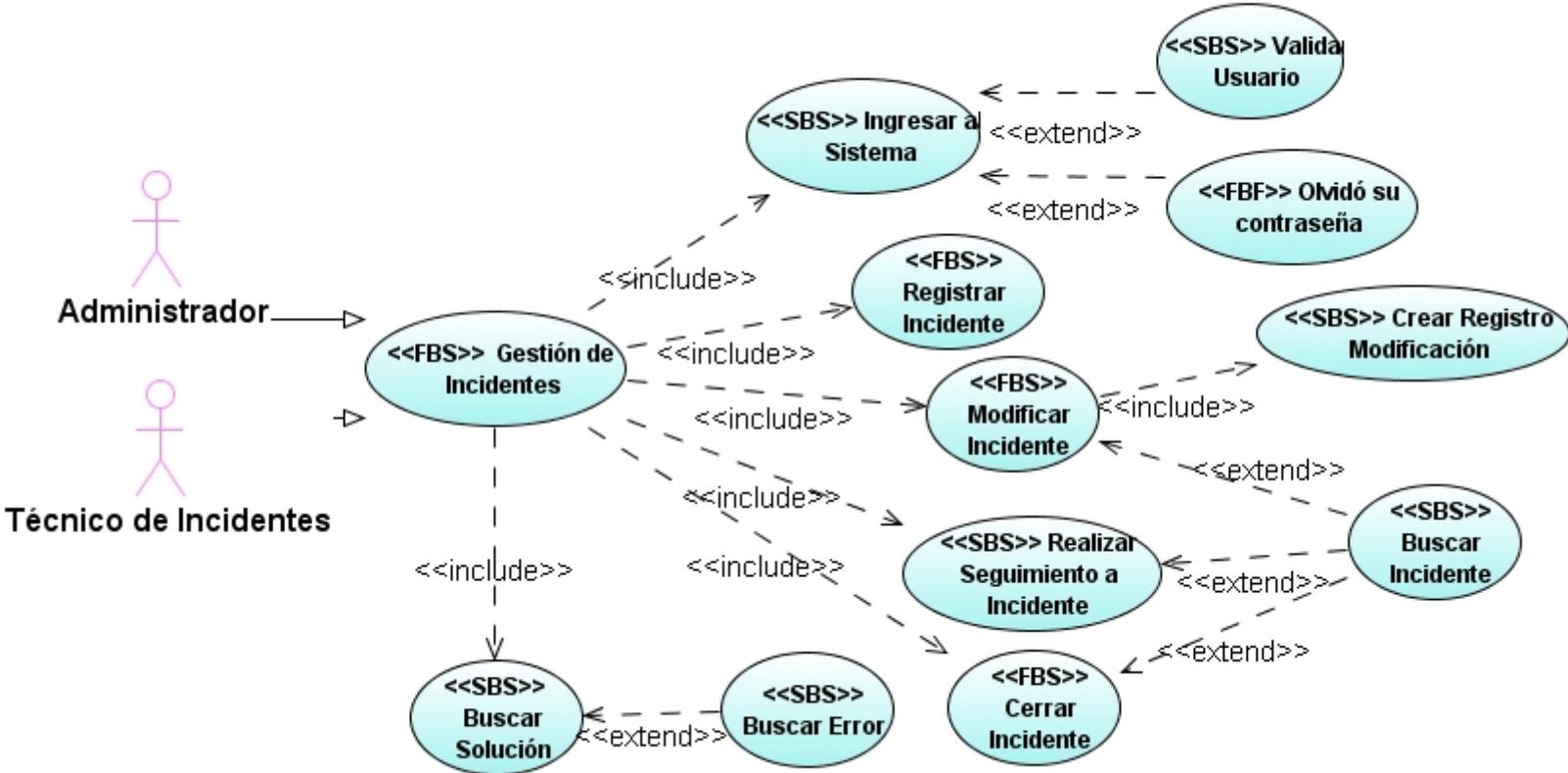


Figura 3.7: Diagrama de casos de uso extendido 2

3.3.3.2.3. Gestión de Configuración 1/2

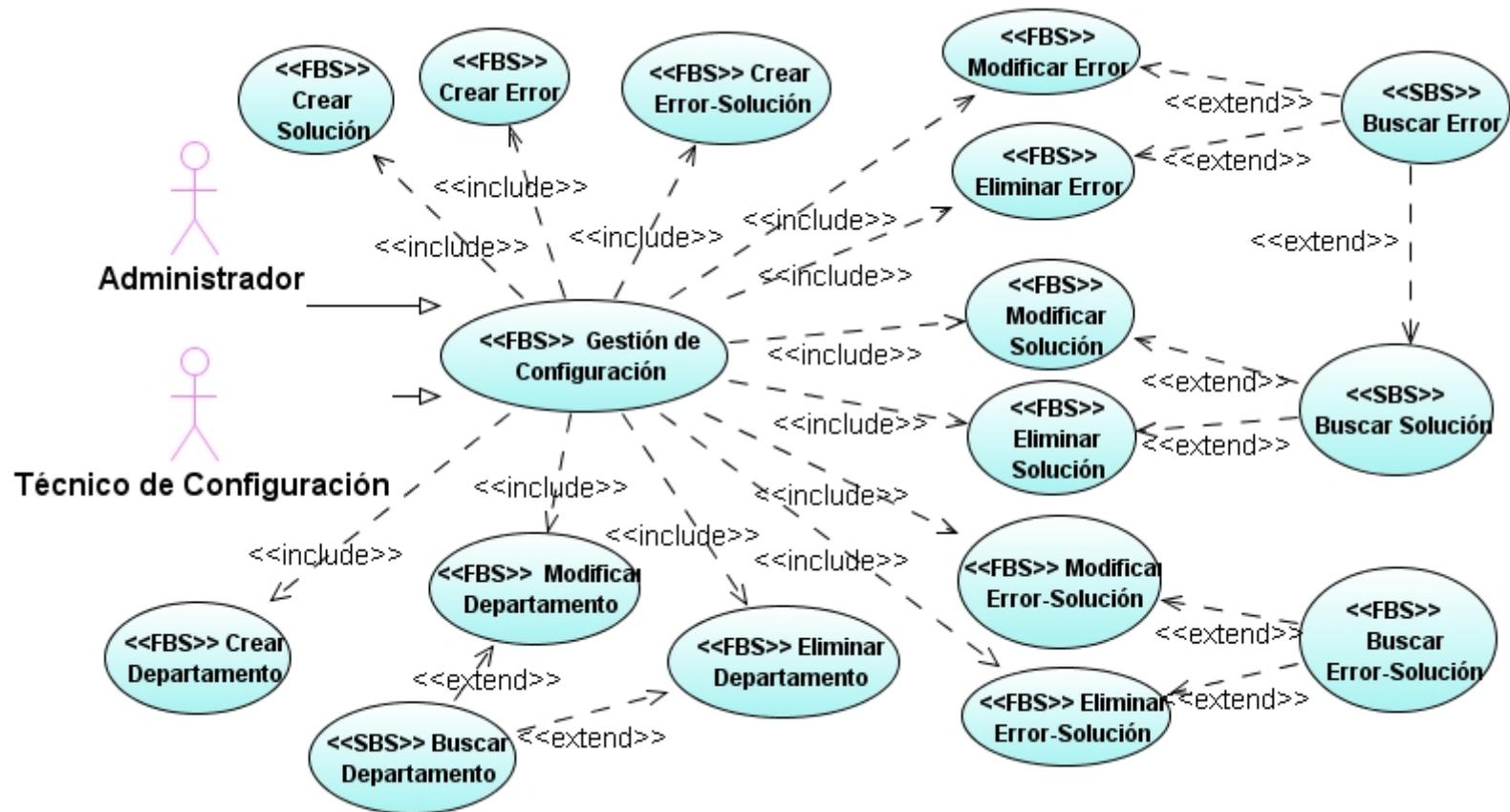


Figura 3.8: Diagrama de casos de uso extendido 3

3.3.3.2.3. Gestión de Configuración 2/2

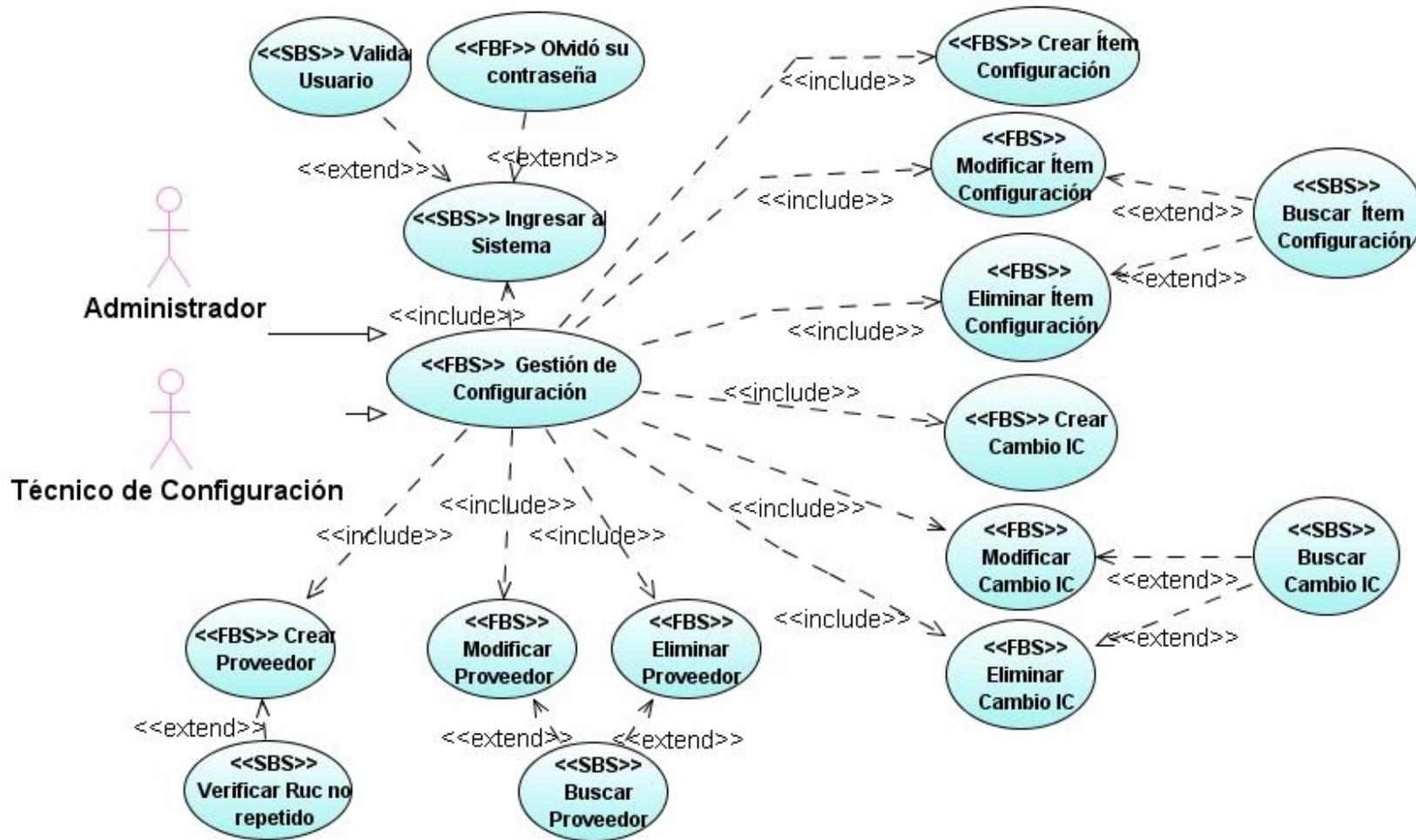


Figura 3.9: Diagrama de casos de uso extendido 4

3.3.3.2.4. Solicitud de Servicio

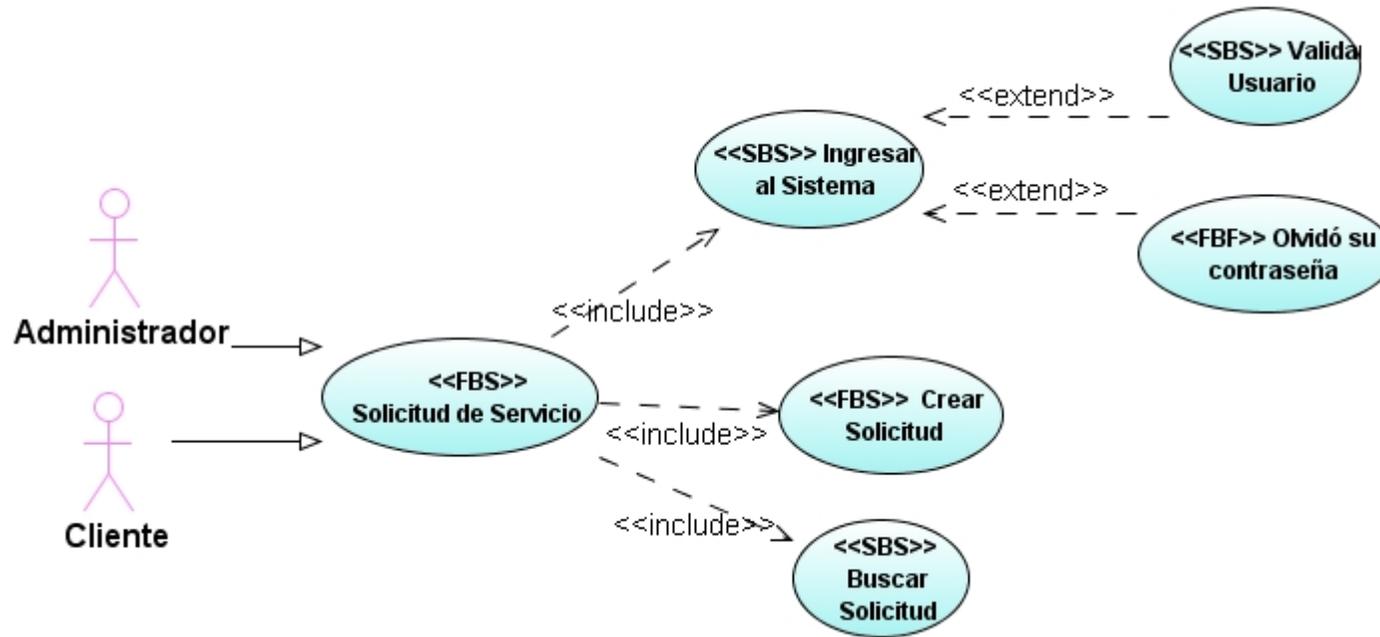


Figura 3.10: Diagrama de casos de uso extendido 5

4.3.4. Diagramas de Actividades

Los diagramas de actividades sirven para mostrar la secuencia de las actividades de un proceso. En el diagrama se muestra el camino desde un punto de inicio hasta el punto final detallando los diferentes caminos que se pueden tomar.

4.3.4.1. Administración de Usuarios

Actores: Administrador

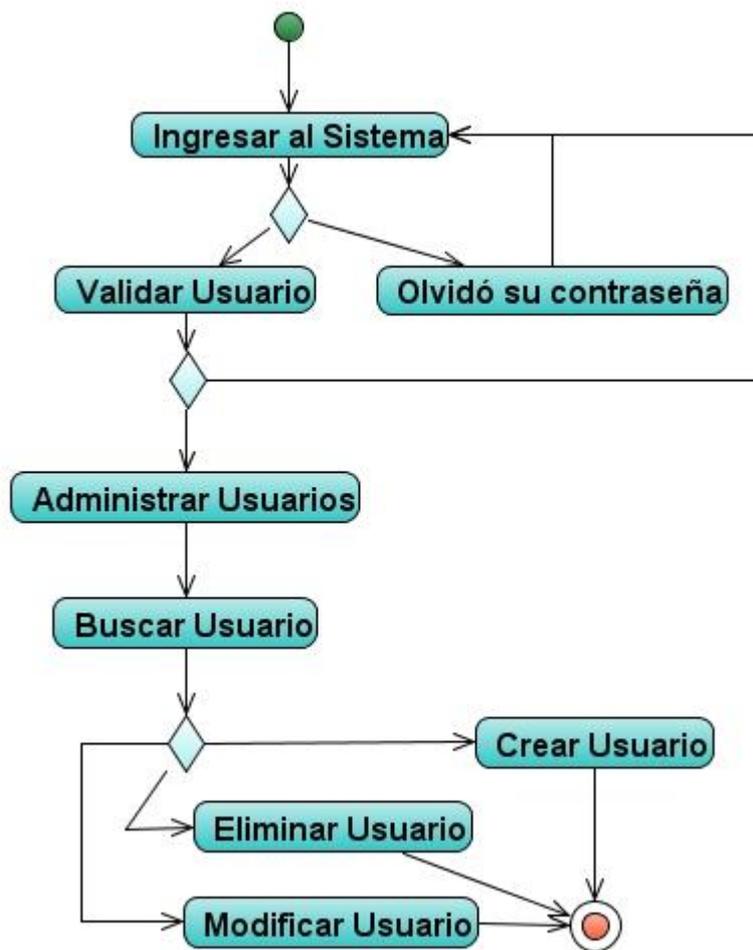


Figura 3.11: Diagrama de actividades 1

4.3.4.2. Gestión de Incidentes

Actores: Administrador y Técnico de configuraciones

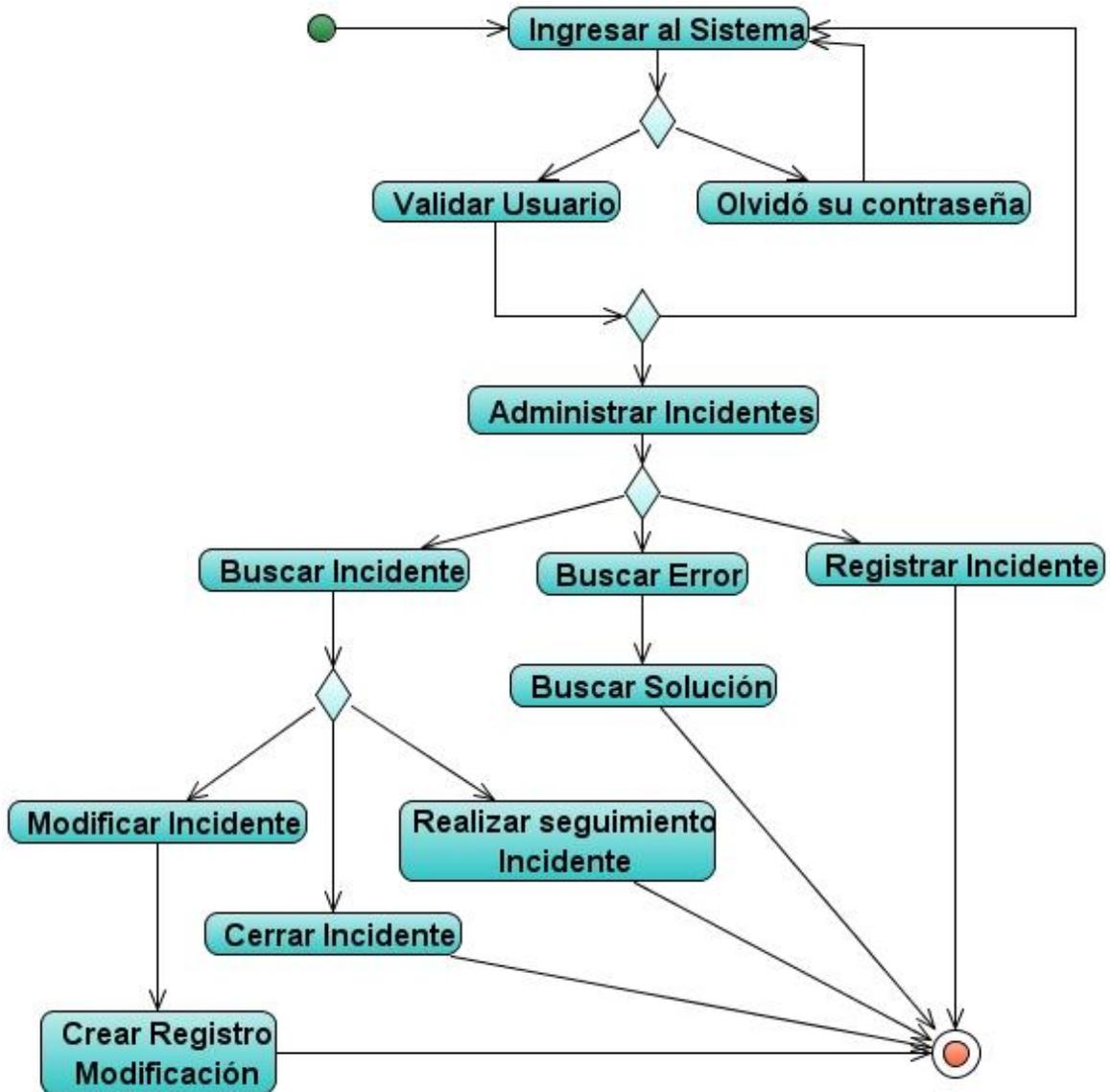


Figura 3.12: Diagrama de actividades 2

4.3.4.3. Gestión de Configuraciones

Actores: Administrador, Técnico de Configuraciones

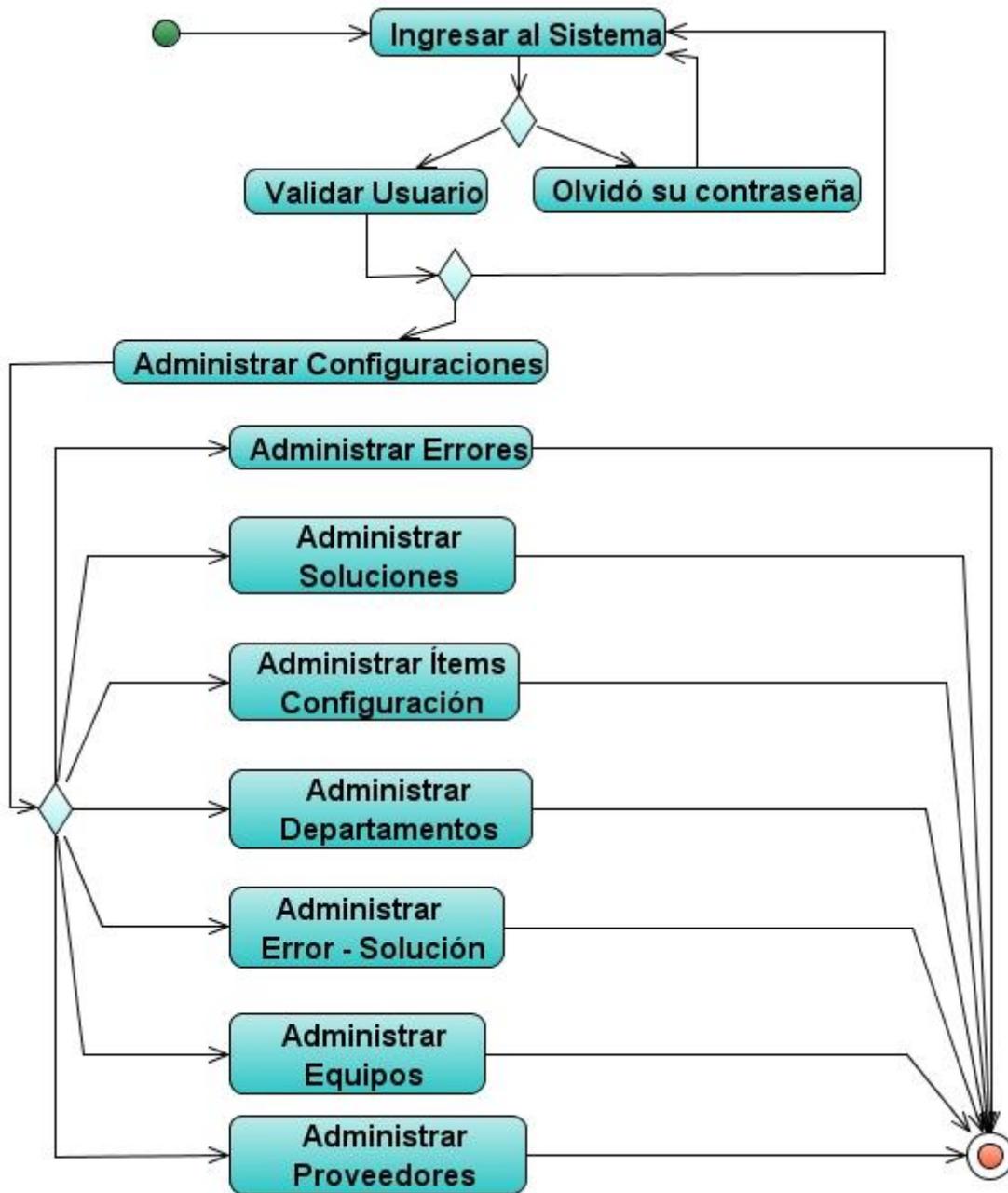


Figura 3.13: Diagrama de actividades 3

4.3.4.4. Solicitud de Servicio

Actores: Administrador, Cliente

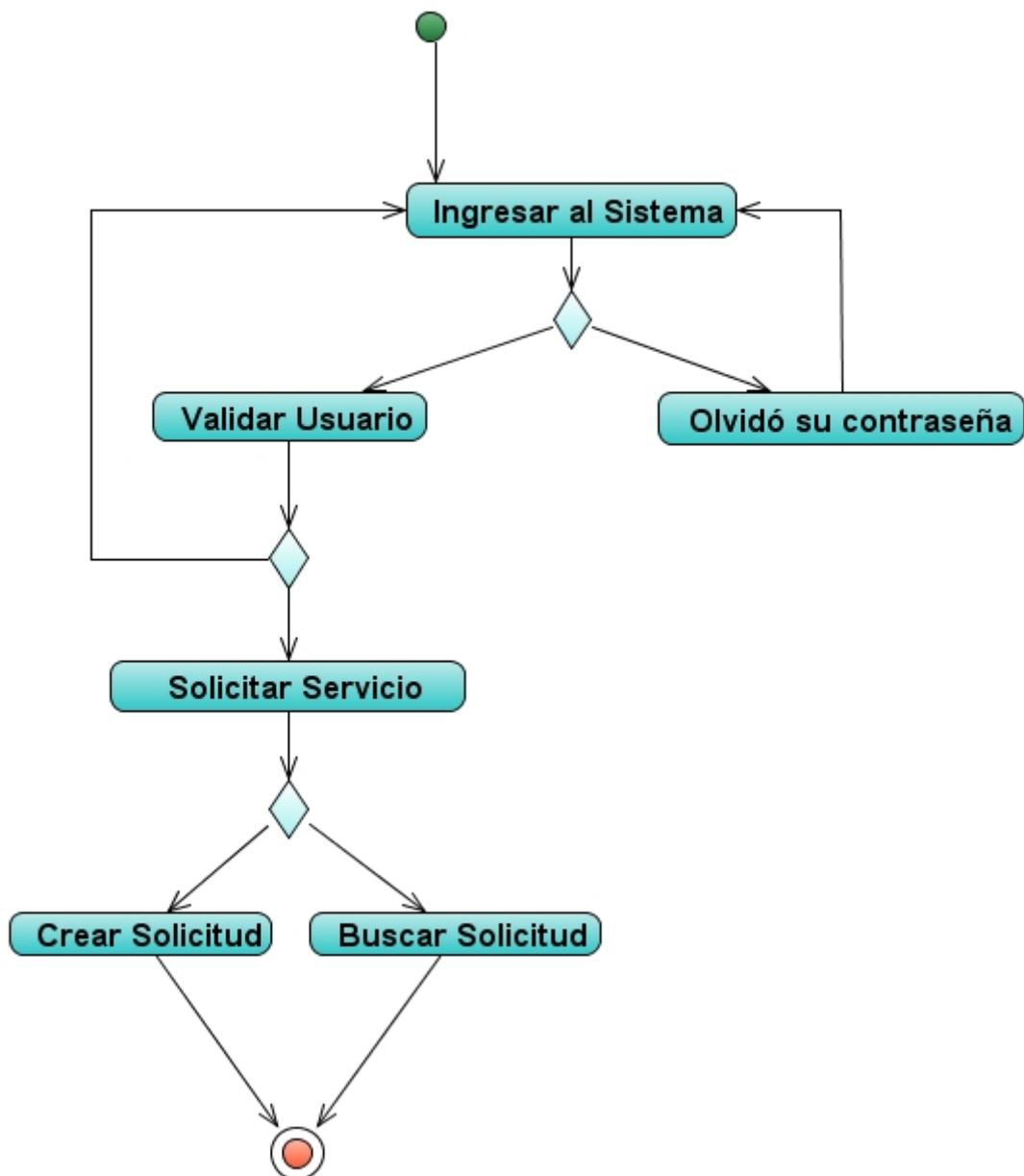


Figura 3.14: Diagrama de actividades 4

4.3.4.5. Genérico de Administración de Tablas

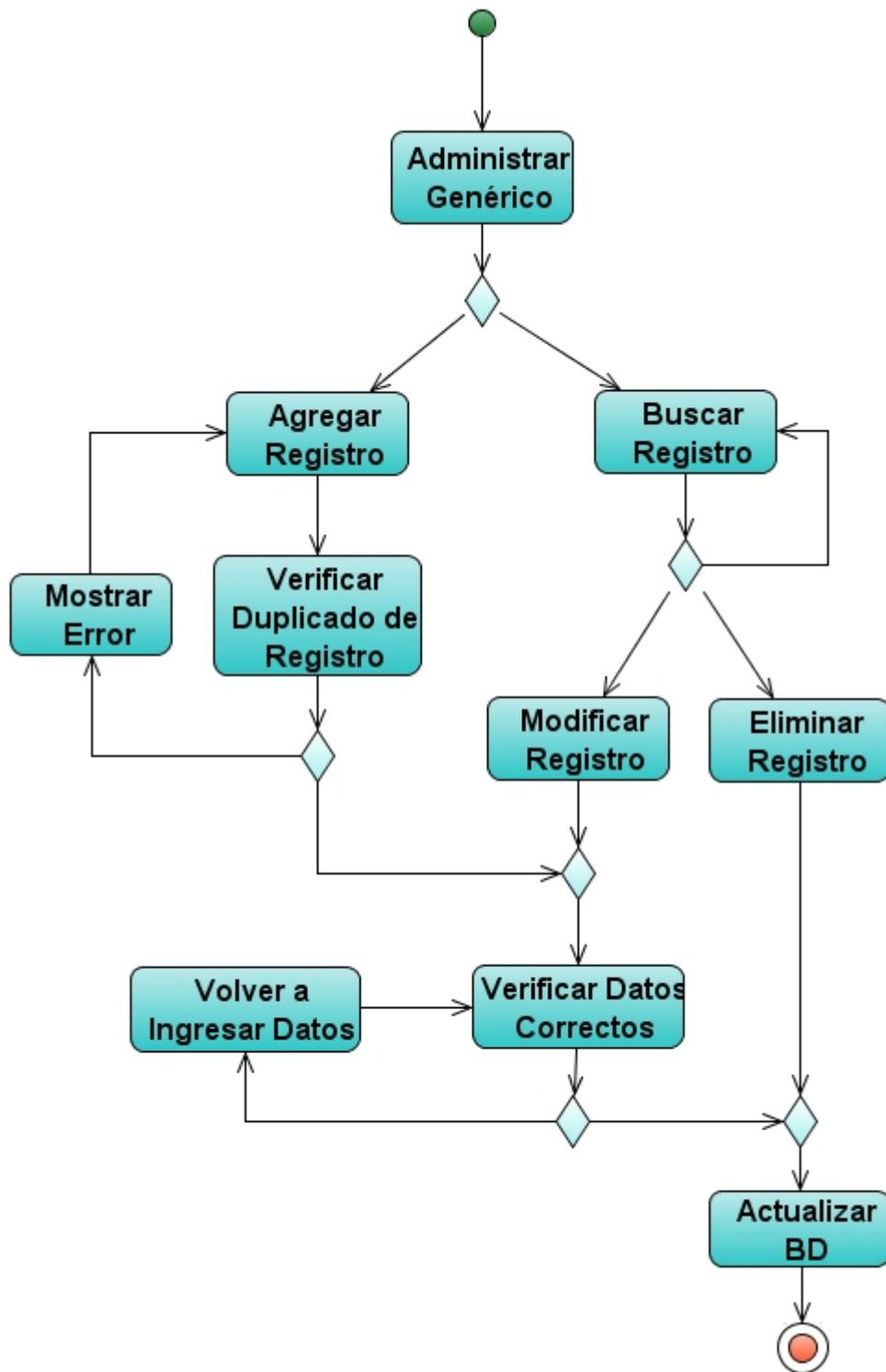


Figura 3.15: Diagrama de actividades 5

4.3.5. Diagramas de Fragmentos

Los diagramas de fragmentos muestran los posibles caminos que existen dentro de la aplicación. El contenido de estos diagramas es:

- Nombre del Proceso.
- Atributos necesarios para el Proceso.
- Rutas entre procesos.

Las rutas se definen de la siguiente forma: <<route>> {nombre de la ruta}.

4.3.5.1. Administrar Usuarios

Actores: Administrador

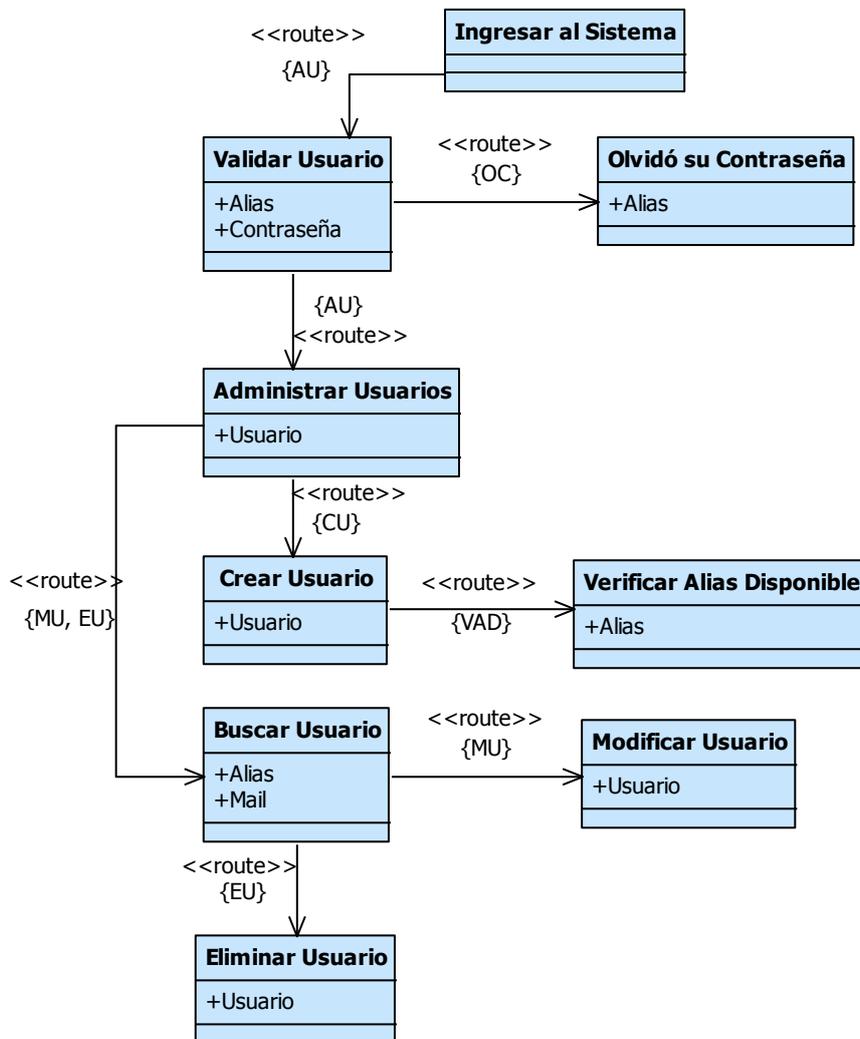


Figura 3.16: Diagrama de fragmentos 1

4.3.5.2. Gestión de Incidentes

Actores: Administrador, Técnico de Incidentes

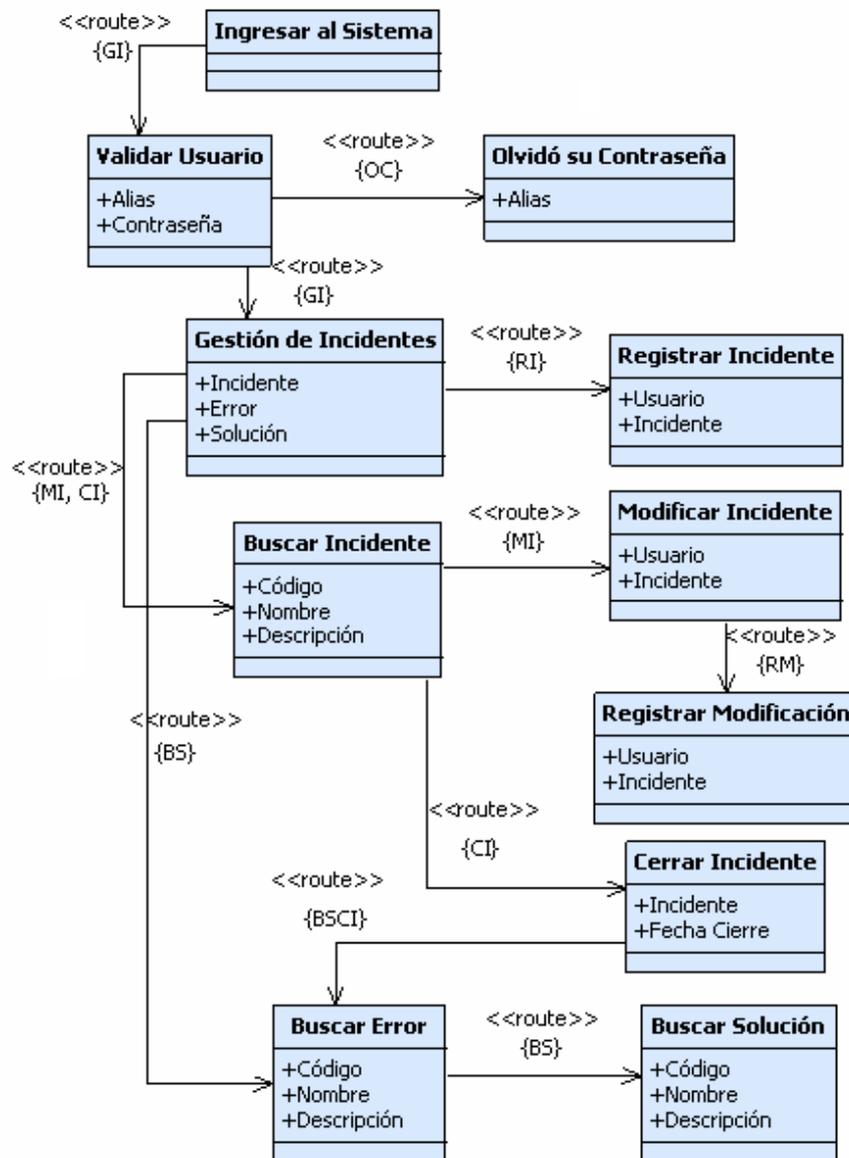


Figura 3.17: Diagrama de fragmentos 2

4.3.5.3. Gestión de Configuraciones 1 / 2

Actores: Administrador, Técnico de Configuraciones

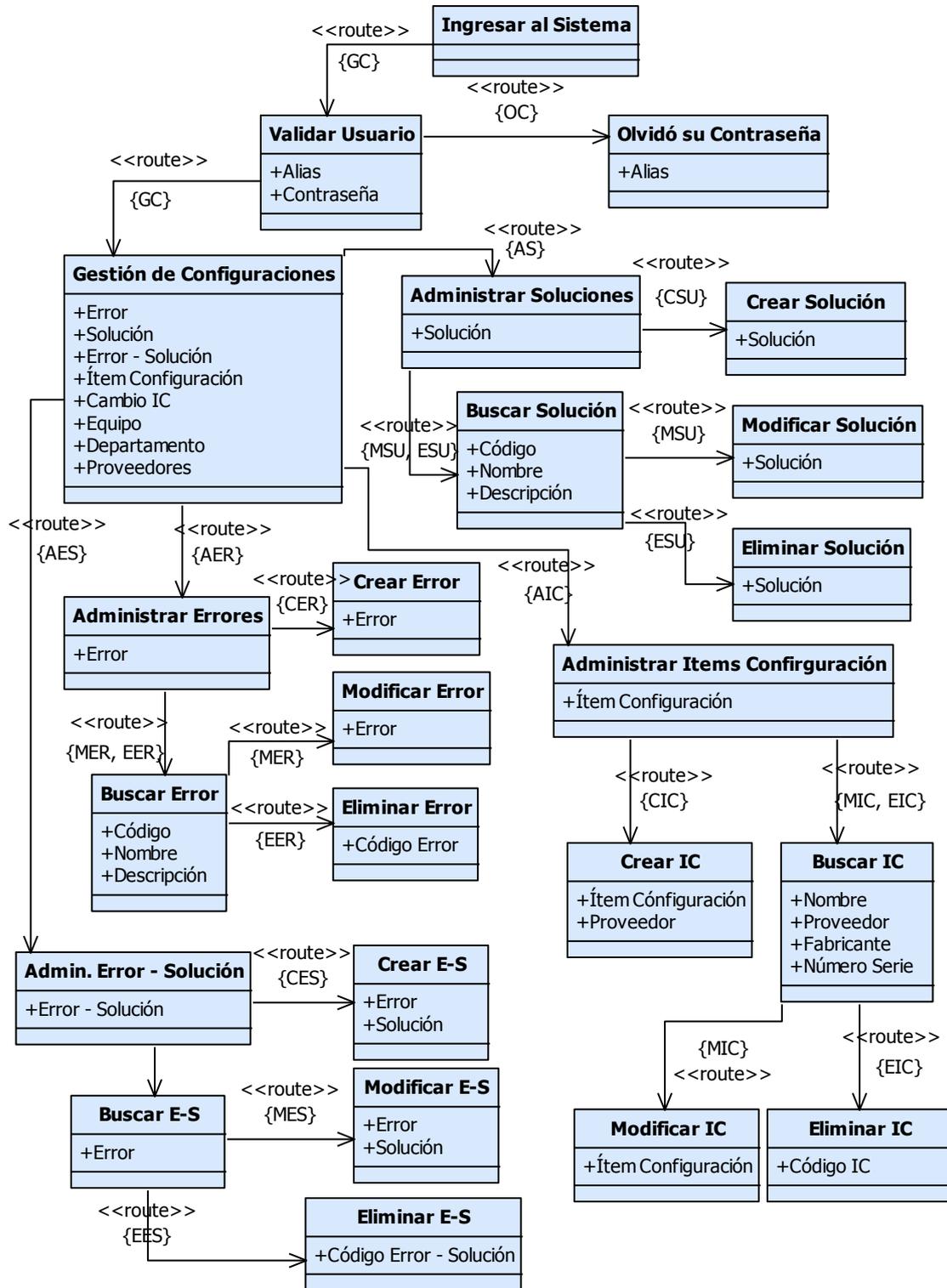


Figura 3.18: Diagrama de fragmentos 3

4.3.5.4. Gestión de Configuraciones 2 / 2

Actores: Administrador, Técnico de Configuraciones

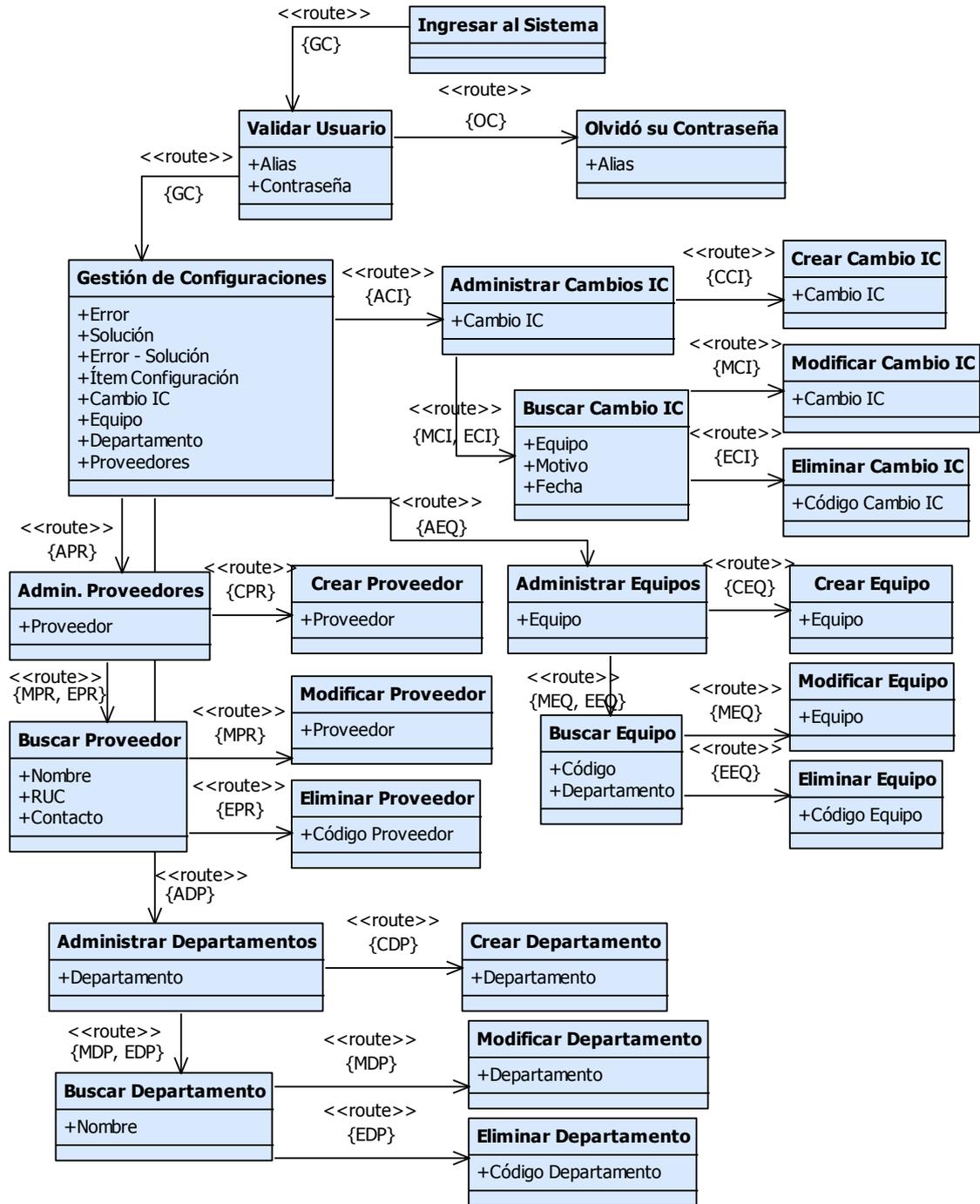


Figura 3.19: Diagrama de fragmentos 4

4.3.5.5. Solicitud de Servicio

Actores: Administrador, Cliente

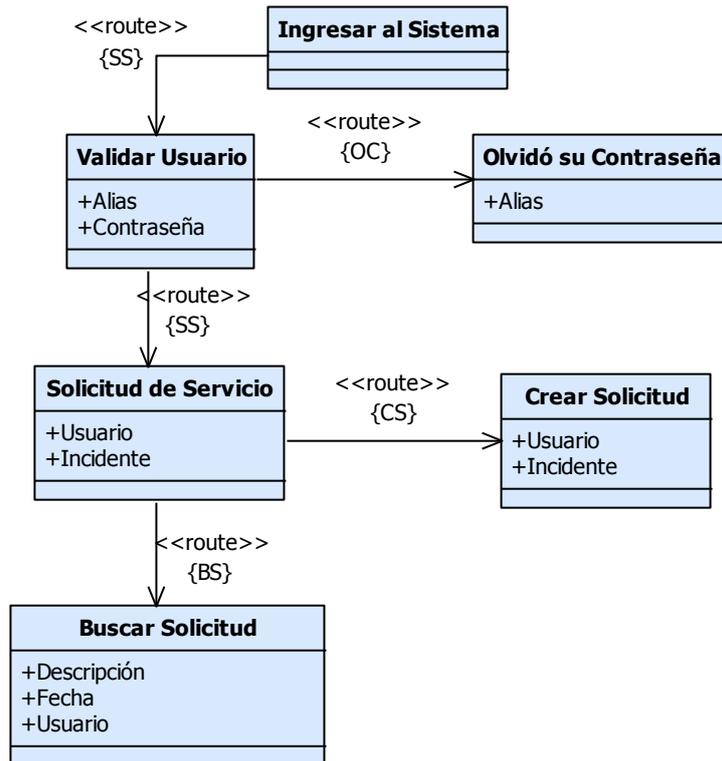


Figura 3.20: Diagrama de fragmentos 5

4.3.6. Diagramas de Navegación

Estos diagramas muestran la forma de navegar dentro del sistema, utilizan la información de rutas obtenida en los diagramas de fragmentos, la información que se muestra en estos diagramas es:

- Nombre del Proceso.
- Atributos necesarios para el Proceso.
- Rutas entre procesos.
- Menús.

4.3.6.1. Inicio

Actores: Administrador

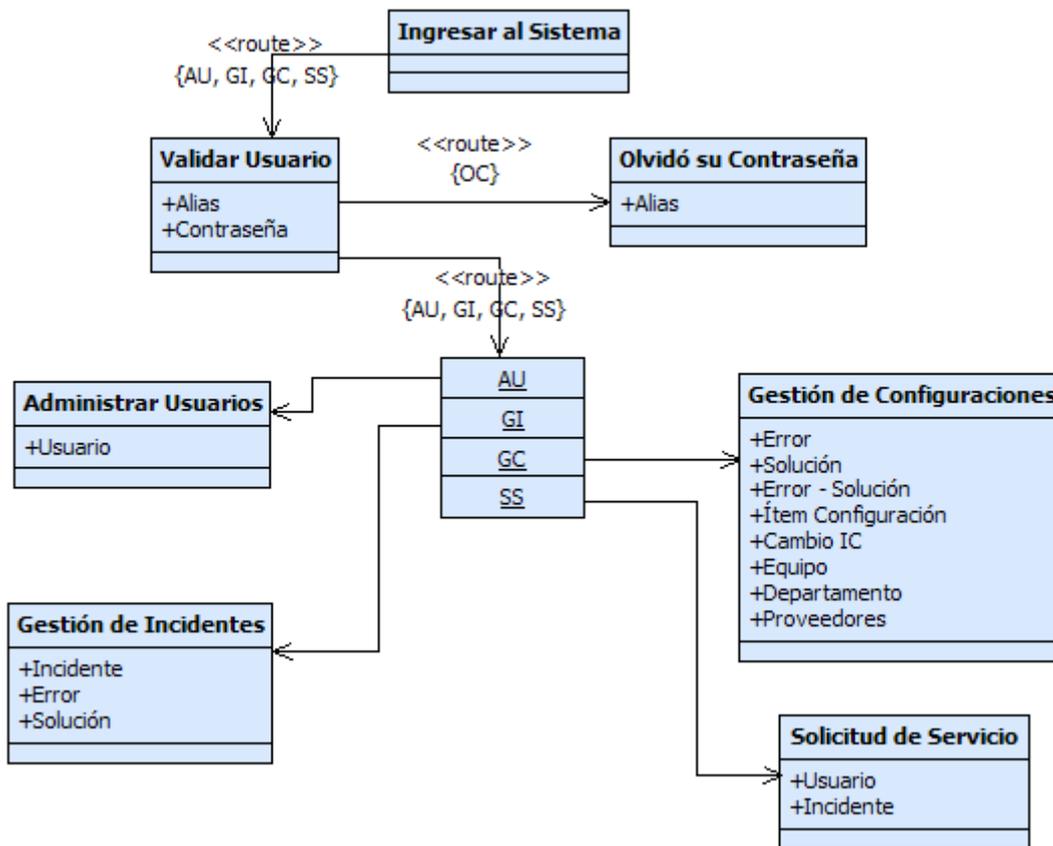


Figura 3.21: Diagrama de navegación 1

4.3.6.2. Administrar Usuarios

Actores: Administrador

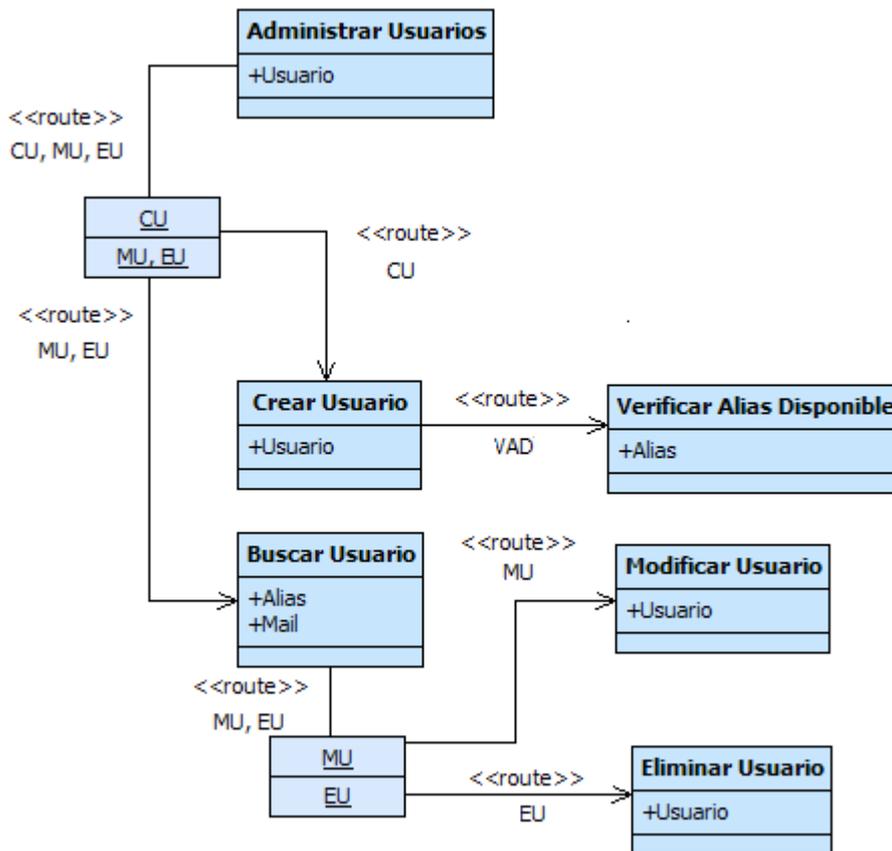


Figura 3.22: Diagrama de navegación 2

4.3.6.3. Gestión de Incidentes

Actores: Administrador, Técnico de Incidentes

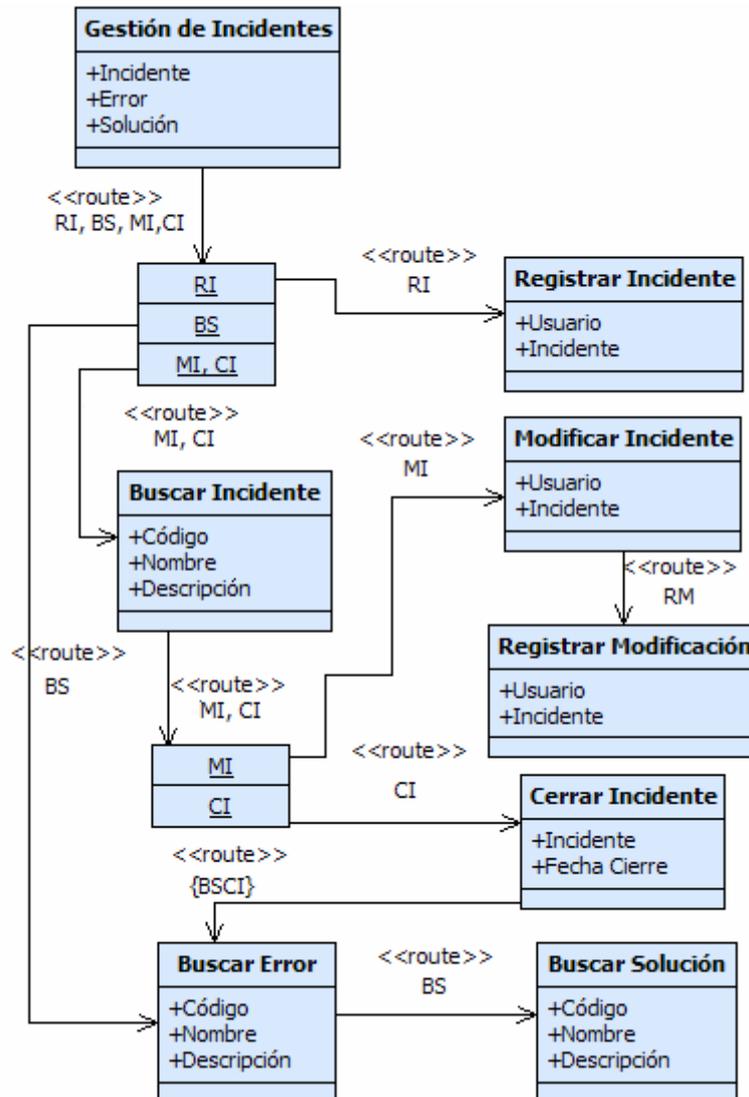


Figura 3.23: Diagrama de navegación 3

4.3.6.4. Gestión de Configuraciones 1 / 2

Actores: Administrador, Técnico de Configuraciones

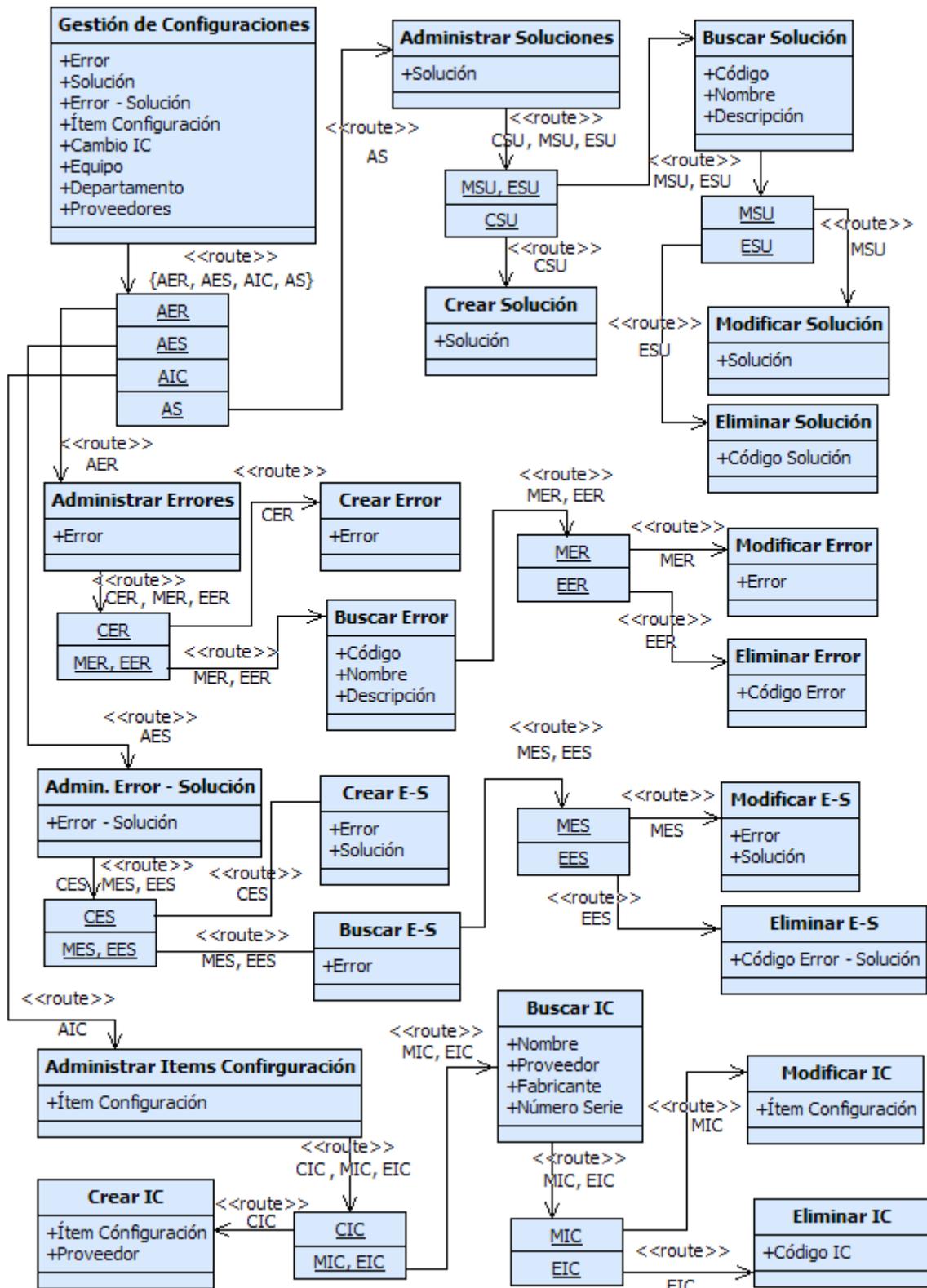


Figura 3.24: Diagrama de navegación 4

3.3.6.4. Gestión de Configuraciones 2 / 2

Actores: Administrador, Técnico de Configuraciones

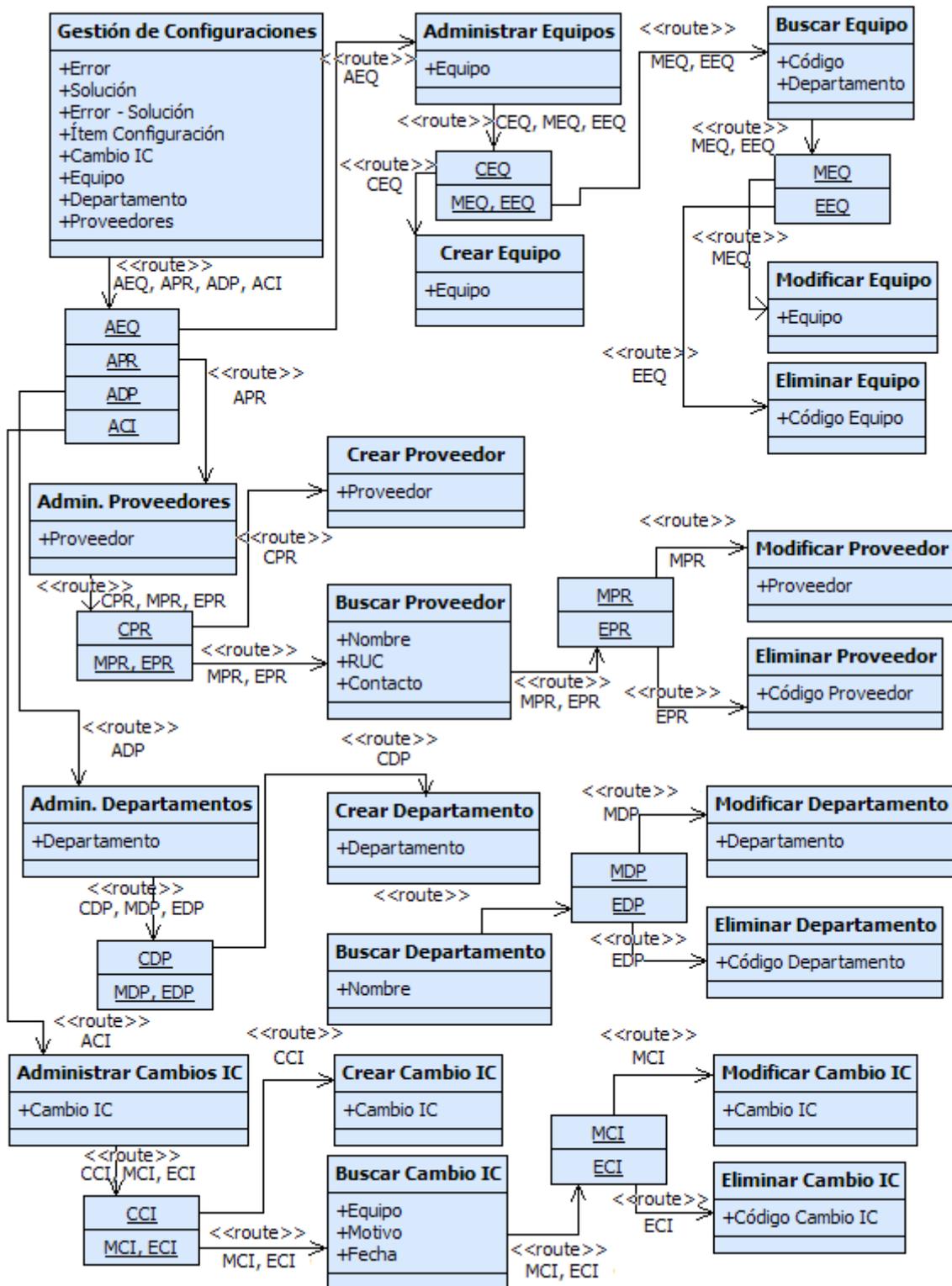


Figura 3.25: Diagrama de navegación 5

4.3.6.5. Solicitud de Servicio

Actores: Administrador, Cliente

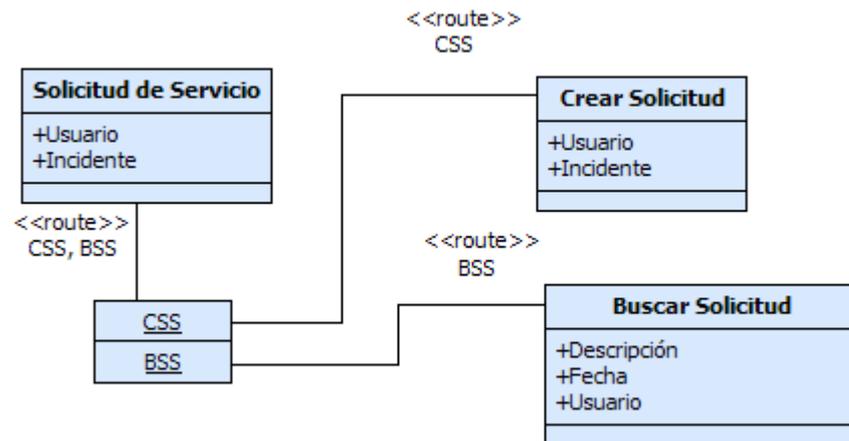


Figura 3.26: Diagrama de navegación 6

4.3.7. Modelo Objeto Relacional

El modelo objeto relacional es parte de la capa PSM de la metodología MIDAS. Muestra la definición de las tablas, vistas, objetos y la relación de estos dentro de la base de datos.

Tabla: Cambio Ítem

```
CREATE TABLE `cambio_item` (  
  `cam_item_codigo` int(10) unsigned NOT NULL auto_increment,  
  `cam_item_descripcion` varchar(500) NOT NULL,  
  `cam_item_motivo` varchar(500) default NULL,  
  `item_anterior` int(10) unsigned NOT NULL,  
  `item_nuevo` int(10) unsigned NOT NULL,  
  `equi_id` int(10) unsigned NOT NULL,  
  `cam_fecha` datetime default NULL,  
  PRIMARY KEY USING BTREE (`cam_item_codigo`),  
  KEY `FK_cambio_item_1` (`item_anterior`),  
  KEY `FK_cambio_item_2` (`item_nuevo`),  
  KEY `FK_cambio_item_equi_id` (`equi_id`),  
  CONSTRAINT `FK_cambio_item_1` FOREIGN KEY (`item_anterior`)  
REFERENCES `item_configuracion` (`item_codigo`),  
  CONSTRAINT `FK_cambio_item_2` FOREIGN KEY (`item_nuevo`)  
REFERENCES `item_configuracion` (`item_codigo`),  
  CONSTRAINT `FK_cambio_item_equipo` FOREIGN KEY (`equi_id`)  
REFERENCES `equipo` (`equi_id`),  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

Tabla: Cierre Incidente

```
CREATE TABLE `cierre_incidente` (  
  `ci_codigo` int(10) unsigned NOT NULL auto_increment,  
  `inc_codigo` int(10) unsigned NOT NULL,  
  `errsolu_codigo` int(10) unsigned NOT NULL,  
  `ci_fecha` datetime NOT NULL,  
  `usua_codigo` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`ci_codigo`),  
  KEY `FK_cierre_incidente_1` (`inc_codigo`),  
  KEY `FK_cierre_incidente_2` (`errsolu_codigo`),  
  KEY `FK_cierre_incidente_3` (`usua_codigo`),  
  CONSTRAINT `FK_cierre_incidente_1` FOREIGN KEY (`inc_codigo`)  
REFERENCES `incidente` (`inc_codigo`),  
  CONSTRAINT `FK_cierre_incidente_2` FOREIGN KEY (`errsolu_codigo`)  
REFERENCES `error_solucion` (`errsolu_codigo`),  
  CONSTRAINT `FK_cierre_incidente_3` FOREIGN KEY (`usua_codigo`)  
REFERENCES `usuario` (`usua_codigo`),  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

Tabla: Departamento

```
CREATE TABLE `departamento` (  
  `dep_codigo` int(10) unsigned NOT NULL auto_increment,  
  `dep_nombre` varchar(50) NOT NULL,  
  `dep_descripcion` varchar(100) NOT NULL,  
  PRIMARY KEY (`dep_codigo`),  
  UNIQUE KEY `UNIQUE` USING BTREE (`dep_nombre`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

Tabla: Equipo

```
CREATE TABLE `equipo` (  
  `equi_id` int(10) unsigned NOT NULL auto_increment,  
  `equi_codigo` varchar(100) NOT NULL,  
  `usua_codigo` int(10) unsigned default NULL,  
  `equi_caracteristicas` varchar(2000) NOT NULL,  
  `dep_codigo` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`equi_id`),  
  UNIQUE KEY `codigo_equipo` (`equi_codigo`),  
  KEY `FK_equipo_1` (`dep_codigo`),  
  KEY `FK_equipo_2` (`usua_codigo`),  
  CONSTRAINT `FK_equipo_1` FOREIGN KEY (`dep_codigo`) REFERENCES  
  `departamento` (`dep_codigo`),  
  CONSTRAINT `FK_equipo_2` FOREIGN KEY (`usua_codigo`) REFERENCES  
  `usuario` (`usua_codigo`),  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

Tabla: Error

```
CREATE TABLE `error` (  
  `err_codigo` int(10) unsigned NOT NULL auto_increment,  
  `err_nombre` varchar(50) default NULL,  
  `err_descripcion` varchar(500) NOT NULL,  
  `err_causa` varchar(500) NOT NULL,  
  `terr_codigo` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`err_codigo`),
```

```

KEY `FK_error_1` (`terr_codigo`),
CONSTRAINT `FK_error_1` FOREIGN KEY (`terr_codigo`) REFERENCES
`tipo_error` (`terr_codigo`),
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

```

Tabla: Error Solución

```

CREATE TABLE `error_solucion` (
`errsolu_codigo` int(10) unsigned NOT NULL auto_increment,
`err_codigo` int(10) unsigned NOT NULL,
`solu_codigo` int(10) unsigned NOT NULL,
PRIMARY KEY (`errsolu_codigo`),
KEY `FK_error_solucion_1` (`err_codigo`),
KEY `FK_error_solucion_2` (`solu_codigo`),
CONSTRAINT `FK_error_solucion_1` FOREIGN KEY (`err_codigo`)
REFERENCES `error` (`err_codigo`),
CONSTRAINT `FK_error_solucion_2` FOREIGN KEY (`solu_codigo`)
REFERENCES `solucion` (`solu_codigo`),
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

```

Tabla: Historial Incidente

```

CREATE TABLE `historial_incidente` (
`hi_codigo` int(10) unsigned NOT NULL auto_increment,
`inc_codigo` int(10) unsigned NOT NULL,
`hi_clasificacion` varchar(45) default NULL,
`hi_estado` varchar(45) default NULL,
`hi_prioridad` varchar(45) default NULL,
`hi_fecha_actualizacion` datetime NOT NULL,

```

```

`usua_codigo` int(10) unsigned default NULL,
`hi_descripcion` varchar(200) default NULL,
`hi_mensaje` varchar(200) default NULL,
PRIMARY KEY (`hi_codigo`),
KEY `FK_historial_incidente_1` (`inc_codigo`),
KEY `FK_historial_incidente_2` (`usua_codigo`),
CONSTRAINT `FK_historial_incidente_1` FOREIGN KEY (`inc_codigo`)
REFERENCES `incidente` (`inc_codigo`),
CONSTRAINT `FK_historial_incidente_2` FOREIGN KEY (`usua_codigo`)
REFERENCES `usuario` (`usua_codigo`),
) ENGINE=InnoDB AUTO_INCREMENT=30 DEFAULT CHARSET=latin1;

```

Tabla: Incidente

```

CREATE TABLE `incidente` (
`inc_codigo` int(10) unsigned NOT NULL auto_increment,
`inc_fecha` datetime NOT NULL,
`inc_descripcion` varchar(200) NOT NULL,
`inc_mensaje` varchar(200) default NULL,
`inc_clasificacion` varchar(45) default NULL,
`inc_estado` varchar(45) NOT NULL,
`inc_prioridad` varchar(45) NOT NULL,
`usua_cliente` int(10) unsigned NOT NULL,
`usua_tecnico` int(10) unsigned default NULL,
PRIMARY KEY (`inc_codigo`),
KEY `FK_incidente_1` (`usua_cliente`),
KEY `FK_incidente_usua_tecnico` (`usua_tecnico`),

```

```

CONSTRAINT `FK_incidente_1` FOREIGN KEY (`usua_cliente`) REFERENCES
`usuario` (`usua_codigo`),
CONSTRAINT `FK_incidente_2` FOREIGN KEY (`usua_tecnico`) REFERENCES
`usuario` (`usua_codigo`),
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;

```

Tabla: Ítem de Configuración

```

CREATE TABLE `item_configuracion` (
`item_codigo` int(10) unsigned NOT NULL auto_increment,
`item_nombre` varchar(100) NOT NULL,
`titem_codigo` int(10) unsigned NOT NULL,
`prov_codigo` int(10) unsigned NOT NULL,
`item_fecha` datetime default NULL,
`item_fabricante` varchar(100) default NULL,
`item_costo` float default NULL,
`item_version` varchar(20) default NULL,
`item_anio` int(10) unsigned default NULL,
`item_num_serie` varchar(100) NOT NULL,
PRIMARY KEY USING BTREE (`item_codigo`),
UNIQUE KEY `UNIQUE` USING BTREE (`item_num_serie`),
KEY `FK_item_configuracion_1` (`prov_codigo`),
KEY `FK_item_configuracion_2` (`titem_codigo`),
CONSTRAINT `FK_item_configuracion_1` FOREIGN KEY (`prov_codigo`)
REFERENCES `proveedor` (`prov_codigo`),
CONSTRAINT `FK_item_configuracion_2` FOREIGN KEY (`titem_codigo`)
REFERENCES `tipo_item` (`titem_codigo`),

```

```
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;
```

Tabla: Proveedor

```
CREATE TABLE `proveedor` (  
  `prov_codigo` int(10) unsigned NOT NULL auto_increment,  
  `prov_nombre` varchar(100) NOT NULL,  
  `prov_ruc` varchar(20) NOT NULL,  
  `prov_direccion` varchar(200) NOT NULL,  
  `prov_telefono` varchar(25) NOT NULL,  
  `prov_fax` varchar(25) default NULL,  
  `prov_celular` varchar(25) default NULL,  
  `prov_mail` varchar(100) NOT NULL,  
  `prov_persona_contacto` varchar(100) NOT NULL,  
  PRIMARY KEY (`prov_codigo`),  
  UNIQUE KEY `UNIQUE` (`prov_ruc`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

Tabla: Solución

```
CREATE TABLE `solucion` (  
  `solu_codigo` int(10) unsigned NOT NULL auto_increment,  
  `solu_nombre` varchar(50) NOT NULL,  
  `solu_descripcion` varchar(1500) NOT NULL,  
  PRIMARY KEY (`solu_codigo`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

Tabla: Tipo Error

```
CREATE TABLE `tipo_error` (  
  `terr_codigo` int(10) unsigned NOT NULL auto_increment,
```

```
`terr_nombre` varchar(50) NOT NULL,  
`terr_descripcion` varchar(100) NOT NULL,  
PRIMARY KEY (`terr_codigo`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

Tabla: Tipo Ítem

```
CREATE TABLE `tipo_item` (  
`titem_codigo` int(10) unsigned NOT NULL auto_increment,  
`titem_nombre` varchar(50) NOT NULL,  
`titem_descripcion` varchar(100) NOT NULL,  
PRIMARY KEY USING BTREE (`titem_codigo`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

Tabla: Tipo Usuario

```
CREATE TABLE `tipo_usuario` (  
`tusua_codigo` int(10) unsigned NOT NULL auto_increment,  
`tusua_nombre` varchar(50) NOT NULL,  
`tusua_descripcion` varchar(100) NOT NULL,  
PRIMARY KEY (`tusua_codigo`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

Tabla: Usuario

```
CREATE TABLE `usuario` (  
`usua_codigo` int(10) unsigned NOT NULL auto_increment,  
`usua_nombre` varchar(50) NOT NULL,  
`usua_apellido` varchar(50) NOT NULL,  
`usua_mail` varchar(100) NOT NULL,  
`usua_alias` varchar(50) NOT NULL,
```

```

`usua_password` varchar(50) NOT NULL,
`tusua_codigo` int(10) unsigned NOT NULL,
`dep_codigo` int(10) unsigned NOT NULL,
PRIMARY KEY (`usua_codigo`),
UNIQUE KEY `Alias` (`usua_alias`),
KEY `FK_usuario_1` (`tusua_codigo`),
KEY `FK_usuario_2` (`dep_codigo`),
CONSTRAINT `FK_usuario_1` FOREIGN KEY (`tusua_codigo`) REFERENCES
`tipo_usuario` (`tusua_codigo`),
CONSTRAINT `FK_usuario_2` FOREIGN KEY (`dep_codigo`) REFERENCES
`departamento` (`dep_codigo`),
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;

```

Tabla: Estado Incidente

```

CREATE TABLE `clasificacion_incidente` (
`cinc_codigo` int(10) unsigned NOT NULL auto_increment,
`cinc_nombre` varchar(50) NOT NULL,
`cinc_valor` varchar(50) NOT NULL,
PRIMARY KEY (`cinc_codigo`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

```

Vista: Registro Estado Incidente

```

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`
SQL SECURITY DEFINER VIEW `estado_incidente_registro` AS
SELECT `e`.`einc_codigo` AS `einc_codigo`,`e`.`einc_nombre` AS
`einc_nombre`,`e`.`einc_valor` AS `einc_valor`

```

```
FROM `estado_incidente` `e` WHERE ((`e`.`einc_nombre` <> _latin1'Cerrado') AND
(`e`.`einc_nombre` <> _latin1'Resuelto') AND (`e`.`einc_nombre` <>
_latin1'Suspendido'))
```

Vista: Modificación Estado Incidente

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `estado_incidente_sincierre` AS
SELECT `e`.`einc_codigo` AS `einc_codigo`,`e`.`einc_nombre` AS
`einc_nombre`,`e`.`einc_valor` AS `einc_valor` FROM `estado_incidente` `e` WHERE
(`e`.`einc_nombre` <> _latin1'Cerrado');
```

4.3.8. Modelo X-LINK

El modelo X-LINK presenta los enlaces que van a existir entre los elementos del sistema y sirve para facilitar la navegación dentro del mismo. Por la cantidad reglas de navegación existentes se define a continuación un ejemplo de las reglas necesarias en el módulo de gestión de incidentes.

Gestión de Incidentes

```
<navigation-rule>
  <from-view-id>/buscarincidente.jsp</from-view-id>
  <navigation-case>
    <from-outcome>modificarincidente</from-outcome>
    <to-view-id>/modificarincidente_.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>seleccionarcliente</from-outcome>
    <to-view-id>/seleccionarcliente.jsp</to-view-id>
```

```
</navigation-case>
<navigation-case>
  <from-outcome>registrarincidente</from-outcome>
  <to-view-id>/registrarincidente.jsp</to-view-id>
</navigation-case>
<navigation-case>
  <from-outcome>cerrarincidente</from-outcome>
  <to-view-id>/cerrarincidente.jsp</to-view-id>
</navigation-case>
<navigation-case>
  <from-outcome>indexincidentes</from-outcome>
  <to-view-id>/indexincidentes.jsp</to-view-id>
</navigation-case>
<navigation-case>
  <from-outcome>vercierreinincidente</from-outcome>
  <to-view-id>/vercierreinincidente.jsp</to-view-id>
</navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/modificarincidente_.jsp</from-view-id>
  <navigation-case>
    <from-outcome>buscarincidente</from-outcome>
    <to-view-id>/buscarincidente.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

```
<navigation-rule>
  <from-view-id>/vercierreincidente.jsp</from-view-id>
  <navigation-case>
    <from-outcome>buscarincidente</from-outcome>
    <to-view-id>/buscarincidente.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>versolicitudes</from-outcome>
    <to-view-id>/versolicitudes.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/cerrarincidente.jsp</from-view-id>
  <navigation-case>
    <from-outcome>buscarincidente</from-outcome>
    <to-view-id>/buscarincidente.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>buscarrerrorsolucion</from-outcome>
    <to-view-id>/buscarrerrorsolucion.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/buscarrerrorsolucion.jsp</from-view-id>
  <navigation-case>
```

<from-outcome>cerrarincidente</from-outcome>

<to-view-id>/cerrarincidente.jsp</to-view-id>

</navigation-case>

</navigation-rule>

CAPÍTULO V: IMPLEMENTACIÓN

5.1. PRUEBAS

Tabla 5.1: Resumen de Pruebas

Navegador / Módulo o Función	Mozilla Firefox	Opera	Google Chrome	IE Explorer 7
Ingreso al Sistema	Funciona	Funciona	Funciona	Funciona
Páginas Índice	Funciona	Funciona	Funciona	Funciona
Menú Principal	Demora al cargar	Funciona	Funciona	Funciona
Vínculos de comando	Funciona	Problema con vínculo	Funciona	Funciona
Módulos administración	Funciona	Funciona	Funciona	Funciona
Módulos de búsqueda	Funciona	Funciona	Funciona	Funciona
Tablas de Datos	Funciona	Funciona	Funciona	Funciona
Componentes estándar	Funciona	Funciona	Funciona	Funciona
Componentes AJAX (webuijsf)	Funciona	Funciona	Funciona	Funciona
Manejo de sesión	Funciona	Funciona	Funciona	Funciona
Páginas de Fragmentos	Funciona	Funciona	Funciona	Funciona
Listas Desplegables	Funciona	Funciona	Funciona	Funciona

El sistema funciona sin problemas en los navegadores Google Chrome e Internet Explorer, el navegador Mozilla Firefox presenta problemas menores al cargar el menú principal y el navegador Opera tiene un pequeño problema con los vínculos del menú. En resumen el sistema puede correr en cualquier navegador, pero su funcionamiento óptimo es en Google Chrome por su rapidez al cargar los componentes AJAX.

5.2. ARQUITECTURA J2EE

5.2.1. Diagrama

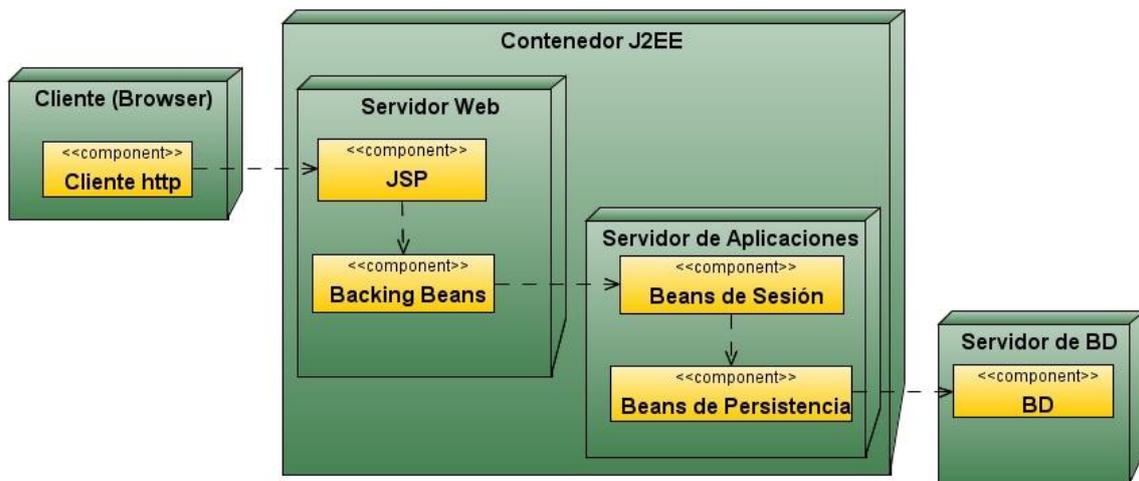


Figura 5.1: Arquitectura del Sistema

5.2.2. Beans de Persistencia

Los Beans de Persistencia son clases JAVA que contienen información de entidades de la base de datos, se los llama también Entity Beans. Su función es almacenar la información de un elemento de la tabla con la que están relacionados.

Su código se escribe en POJOS (Plain Old Java Object), que son simplemente archivos planos. Para hacer que una clase común de JAVA se convierta en un Bean de Persistencia se añaden anotaciones las cuales indican el tipo de Bean que es la clase creada.

Tabla 5.2: Anotaciones de un Bean de Persistencia

Anotación	Descripción
@Entity:	Indica que el archivo es un Bean de Persistencia
@Table(name= "usuario")	El nombre de la tabla en la BD con la que se relaciona es usuario.

@NamedQueries	Agrupación de NamesQuery's
@NamedQuery	Define sentencias de búsqueda utilizando EJBQL, las cuales son sentencias que se realizan sobre beans de persistencia, no sobre las tablas de la BD.
@Id	El atributo es el identificador de la tabla.
@Column(name= "usua_nombre")	El nombre de la columna en la BD con la que se relaciona es usua_nombre.
@OneToMany	El tipo de relación es de uno a varios.
@JoinColumn	El atributo que se encuentra bajo ésta anotación es una relación con otra tabla.
@ManyToOne	El tipo de relación es de muchos a uno.
@Transient	Declara un atributo que no está en la tabla de la BD.
@Override	La función va a ser rescrita.

A continuación se muestra como ejemplo la clase completa del Bean de Persistencia Usuario, con las anotaciones descritas.

```

package persistencia; //Nombre del Paquete

//Librerías importadas para el proyecto

import java.io.Serializable;

import java.util.Collection;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

```

```

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import javax.persistence.Transient;

@Entity

@Table(name = "usuario")

@NamedQueries({

    @NamedQuery(name = "Usuario.findByUsuaCodigo", query = "SELECT u FROM
Usuario u WHERE u.usuaCodigo = :usuaCodigo"),

    @NamedQuery(name = "Usuario.findByUsuaNombre", query = "SELECT u FROM
Usuario u WHERE u.usuaNombre = :usuaNombre"),

    @NamedQuery(name = "Usuario.findByUsuaApellido", query = "SELECT u
FROM Usuario u WHERE u.usuaApellido = :usuaApellido"),

    @NamedQuery(name = "Usuario.findByUsuaMail", query = "SELECT u FROM
Usuario u WHERE u.usuaMail = :usuaMail"),

    @NamedQuery(name = "Usuario.findByUsuaAlias", query = "SELECT u FROM
Usuario u WHERE u.usuaAlias = :usuaAlias"),

    @NamedQuery(name = "Usuario.findByUsuaPassword", query = "SELECT u
FROM Usuario u WHERE u.usuaPassword = :usuaPassword"),

    @NamedQuery(name = "Usuario.findByLoginClave", query="select u from Usuario
u where u.usuaAlias= :usuaAlias and u.usuaPassword = :usuaPassword"))

```

```

public class Usuario implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    @Column(name = "usua_codigo", nullable = false)

    private Integer usuaCodigo;

    @Column(name = "usua_nombre", nullable = false)

    private String usuaNombre;

    @Column(name = "usua_apellido", nullable = false)

    private String usuaApellido;

    @Column(name = "usua_mail", nullable = false)

    private String usuaMail;

    @Column(name = "usua_alias", nullable = false)

    private String usuaAlias;

    @Column(name = "usua_password", nullable = false)

    private String usuaPassword;

    @OneToMany(mappedBy = "usuaCodigo")

    private Collection<Equipo> equipoCollection;

    @JoinColumn(name = "dep_codigo", referencedColumnName = "dep_codigo")

    @ManyToOne

    private Departamento depCodigo;

    @JoinColumn(name = "tusua_codigo", referencedColumnName = "tusua_codigo")

    @ManyToOne

    private TipoUsuario tusuaCodigo;

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "usuaCodigo")

    private Collection<CierreIncidente> cierreIncidenteCollection;

```

```

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "usuaCodigo")
    private Collection<HistorialIncidente> historialIncidenteCollection;

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "usuaCliente")
    private Collection<Incidente> incidenteCollection;

    @Transient
    private String tusuaNombre;

    @Transient
    private String depNombre;

    public Usuario() {
    }

    public Usuario(Integer usuaCodigo) {
        this.usuaCodigo = usuaCodigo;
    }

    public Usuario(Integer usuaCodigo, String usuaNombre, String usuaApellido, String
usuaMail, String usuaAlias, String usuaPassword) {
        this.usuaCodigo = usuaCodigo;
        this.usuaNombre = usuaNombre;
        this.usuaApellido = usuaApellido;
        this.usuaMail = usuaMail;
        this.usuaAlias = usuaAlias;
        this.usuaPassword = usuaPassword;
    }

    //Funciones Get y Set

    public String getDepNombre() {
        return depCodigo.getDepNombre();
    }

```

```

}

public void setDepNombre(String depNombre) {

    this.depNombre = depNombre;

}

...

@Override

public int hashCode() {

    int hash = 0;

    hash += (usuaCodigo != null ? usuaCodigo.hashCode() : 0);

    return hash;

}

@Override

public boolean equals(Object object) {

    // TODO: Warning - this method won't work in the case the id fields are not set

    if (!(object instanceof Usuario)) {

        return false;

    }

    Usuario other = (Usuario) object;

    if ((this.usuaCodigo == null && other.usuaCodigo != null) || (this.usuaCodigo !=

null && !this.usuaCodigo.equals(other.usuaCodigo))) {

        return false;

    }

    return true;

}

@Override

```

```

public String toString() {
    return "persistencia.Usuario[usuaCodigo=" + usuaCodigo + "];
}

```

5.2.3. Beans de Sesión

Los Beans de Sesión son clases JAVA que realizan acciones sobre los Beans de Persistencia.

Están compuestos por dos tipos archivos. Los archivos de definición, que son los cuales en que se definen las funciones y el archivo de implementación, donde se desarrollan todas las funciones definidas.

Al igual que los Beans de Persistencia poseen anotaciones para diferenciarlos. Las anotaciones son:

Tabla 5.3: Anotaciones Beans de Sesión: Archivos de definición.

Anotación	Descripción
@Local	El bean de sesión es local. No puede llamarse remotamente.
@Remote	El bean de sesión es remoto. Se lo puede llamar de forma local y remota.

Tabla 5.4: Anotaciones Beans de Sesión: Archivos de definición.

Anotación	Descripción
@Stateless	El bean no mantiene el estado
@Stateful	El bean mantiene el estado
@TransactionManagement	Indica que el bean va a soportar manejo de transacciones.
@Resource	Define la transacción.

@PersistenceUnit (unitName = "MundoHelpDesk-ejbPU")	= El nombre de la unidad de persistencia es MundoHelpDesk-ejbPU.
@PersistenceContext	Declara el administrador de entidades.

Archivo de Definición de la interfaz remota de Usuario

```

package sesion;

import java.util.List;

import javax.ejb.Remote;

import persistencia.Usuario;

@Remote

public interface UsuarioRemote {

    String getMessage(String nombre);

    List<Usuario> consultarUsuarios();

    List<Usuario> queryUsuarioByLoginClave(String login, String clave);

    void agregarUsuario(String nombre, String apellido, String mail, String alias, String password, Integer tipoUsuario, Integer departamento) throws Exception ;

    void modificarUsuario(Integer codigo, String nombre, String apellido, String mail, String alias, String password, Integer tipoUsuario, Integer departamento) throws Exception;

    void modificarDatosPersonales(Integer codigo, String nombre, String apellido, String mail) throws Exception;

    void eliminarUsuario(Integer codigo, String alias) throws Exception ;

    List<Usuario> consultarPorAlias(String nombre);

    List<Usuario> verificarAliasDisponible(String nombre);

    List<Usuario> consultarPorCodigo(Integer codigo);

```

```

List<Usuario> consultarPorMail(String mail);

List<Usuario> consultarPorTipo(Integer tusua);

List<Usuario> consultarPorTodo(String nombre, String apellido, String alias, Integer
tipoUsuario, Integer departamento);

Usuario find(Object id);
}

```

Archivo de implementación

@Stateless

@TransactionManagement(TransactionManagementType.BEAN)

```
public class UsuarioBean implements UsuarioRemote, Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

@Resource

```
    private UserTransaction utx;
```

@PersistenceUnit(unitName = "MundoHelpDesk-ejbPU")//Nombre del persistence-unit en el persistence.xml

```
    private EntityManagerFactory emf;
```

```
    private EntityManager getEntityManager(){
```

```
        return emf.createEntityManager();
```

```
    }
```

```
    public String getMessage(String nombre) {
```

```
        return "Hello EJB World: Bienvenido "+nombre;
```

```
    }
```

```
    public UsuarioBean(){}
```

```
    public List<Usuario> consultarUsuarios() {
```

```
        EntityManager em = getEntityManager();
```

```

    Query consulta= em.createQuery("SELECT u FROM Usuario u");
    return consulta.getResultList();
}

public List<Usuario> consultarUsuarioPorLoginPassword(String login, String clave)
{
    EntityManager em = getEntityManager();

    Query consulta = em.createQuery("SELECT u FROM Usuario u WHERE
u.usuaContrasenia =?1 and u.usuaAlias =?2 ");

    consulta.setParameter(1, clave);

    consulta.setParameter(2, login);

    return consulta.getResultList();
}

public List<Usuario> queryUsuarioByLoginClave(String login, String clave) {
    EntityManager em = getEntityManager();

    Query consulta = em.createNamedQuery("Usuario.findByLoginClave");

    consulta.setParameter("usuaAlias", login);

    consulta.setParameter("usuaPassword", clave);

    return consulta.getResultList();
}
...
}

```

5.2.4. JSF's

Los JSF's son páginas Web que se ejecutan en el cliente. Se componen de dos archivos. Un archivo JSP y un archivo JAVA llamado backing bean.

5.2.4.1. JSPs

Los archivos JSP sirven para programar la interfaz visual, que es la que verá el usuario final. El lenguaje que se utiliza es XML, lo que significa que se basa en etiquetas, como HTML.

Se inicia el archivo definiendo la versión y la codificación del archivo XML, de la siguiente manera: `<?xml version="1.0" encoding="UTF-8"?>`

Luego se define la raíz del archivo JSP:

```
<jsp:root          version="2.1"          xmlns:f=http://java.sun.com/jsf/core
xmlns:h="http://java.sun.com/jsf/html"
xmlns:jsp="http://java.sun.com/JSP/Page"
xmlns:webuijsf="http://www.sun.com/webui/webuijsf">
```

Y se continúa con la programación de la interfaz visual con una serie de etiquetas que indican cada elemento que se encuentra dentro de la página Web.

Dando como resultado:

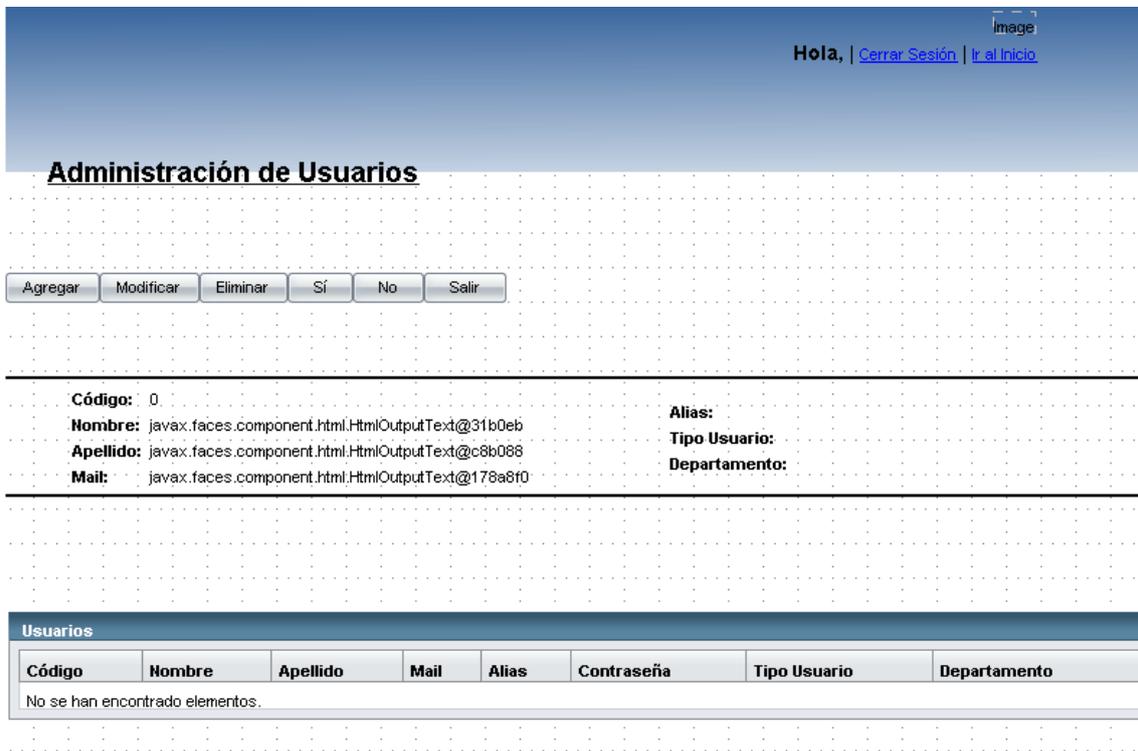


Figura 5.2: Página de Administración de Usuarios

5.2.4.2. Backing Beans

El Backing Bean es un tipo de Bean que realizan toda la lógica del JSF. Es un archivo JAVA que actúa por detrás del JSP, realizando las acciones, consultas, conexiones con los Beans de Sesión, etc.

Estos beans al igual que los anteriores deben definirse dentro de un archivo llamado faces-config.xml, en el cual se definen todos los backing bean que se utilizarán en la aplicación, para que estos puedan comunicarse con los Beans de Sesión que se encuentran en el servidor de aplicaciones.

Los backing bean no utilizan una anotación para definir su funcionalidad, por lo que se deben definir correctamente en el archivo faces-config.xml. La definición correcta es:

```
<managed-bean>  
    <managed-bean-name>controlusuario</managed-bean-name>  
    <managed-bean-class>mundohelpdeskwar.controlusuario</managed-bean-class>  
    <managed-bean-scope>request</managed-bean-scope>  
</managed-bean>
```

Para llamar a un Bean de Sesión desde un Backing Bean se utiliza la anotación @EJB, la cual indica que el objeto que se está definiendo es un EJB de sesión.

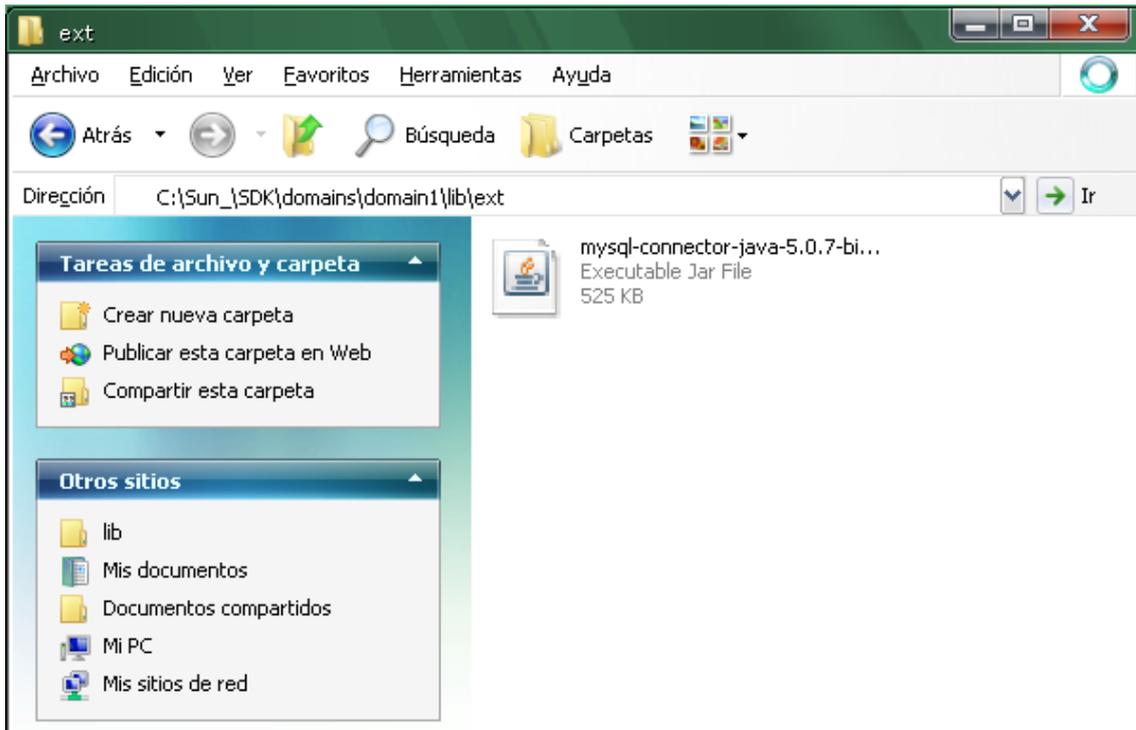
5.3. Despliegue en el Servidor de Aplicaciones

5.3.1. Instalar el SDK.

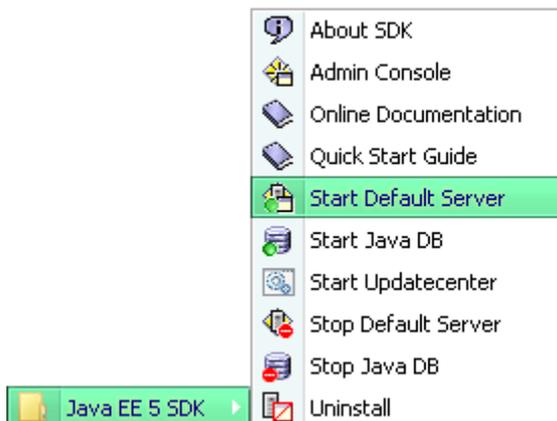
5.3.2. El primer paso antes de iniciar el servidor de aplicaciones, es copiar el driver

JDBC de MySQL para JAVA en la carpeta de librerías que se encuentra en:

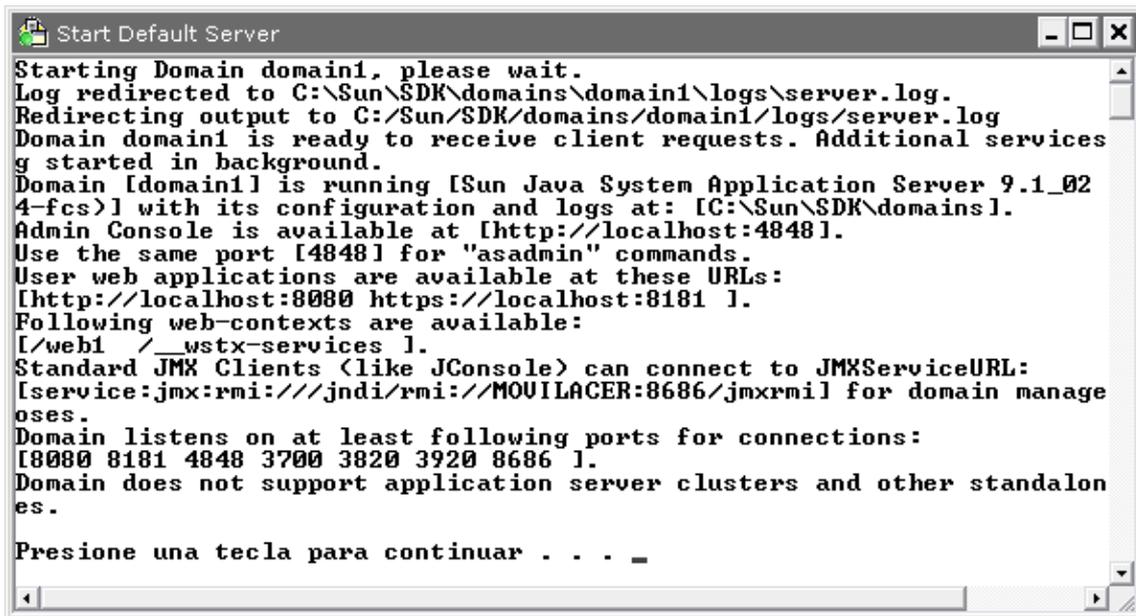
C:\Sun_\SDK\domains\domain1\lib\ext



5.3.3. Se inicia el servidor de aplicaciones.



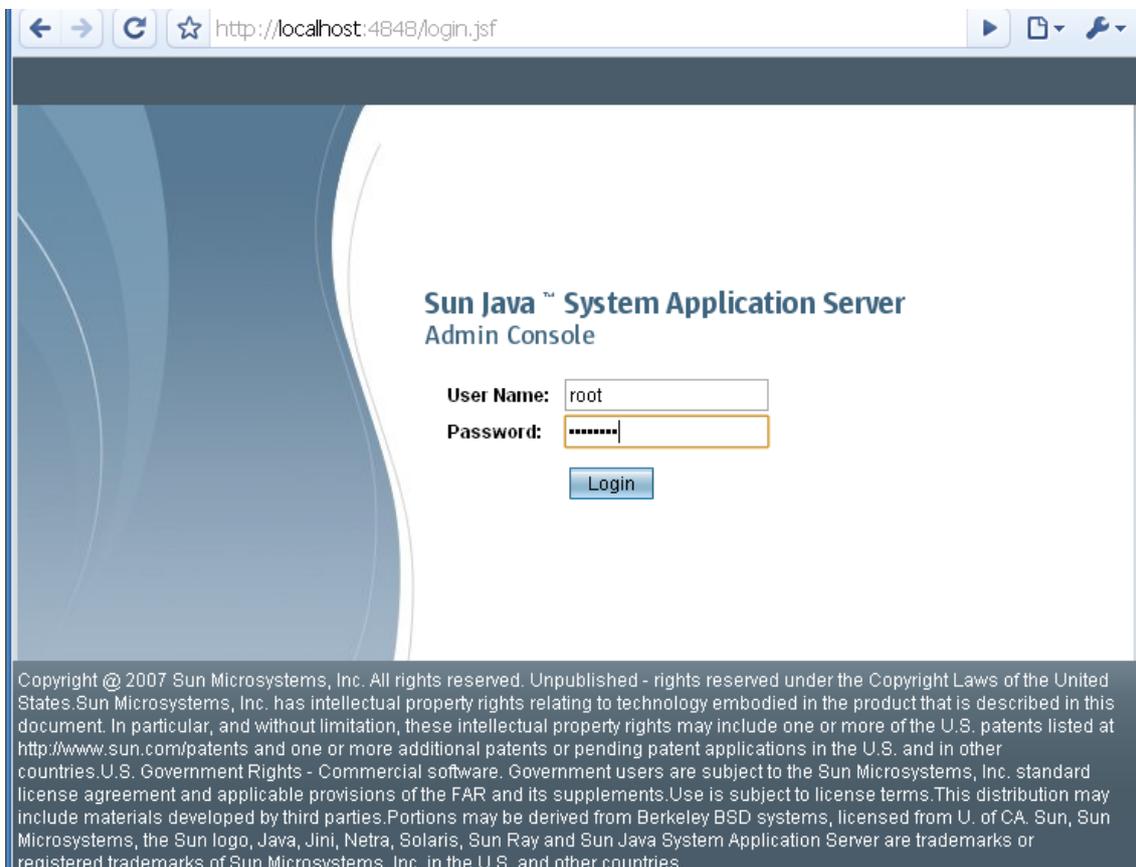
5.3.4. Se despliega una ventana de DOS, que indica que el servidor de aplicaciones está levantado y atendiendo por los puertos 8080, 4848, 3700, 3920 y 8686.



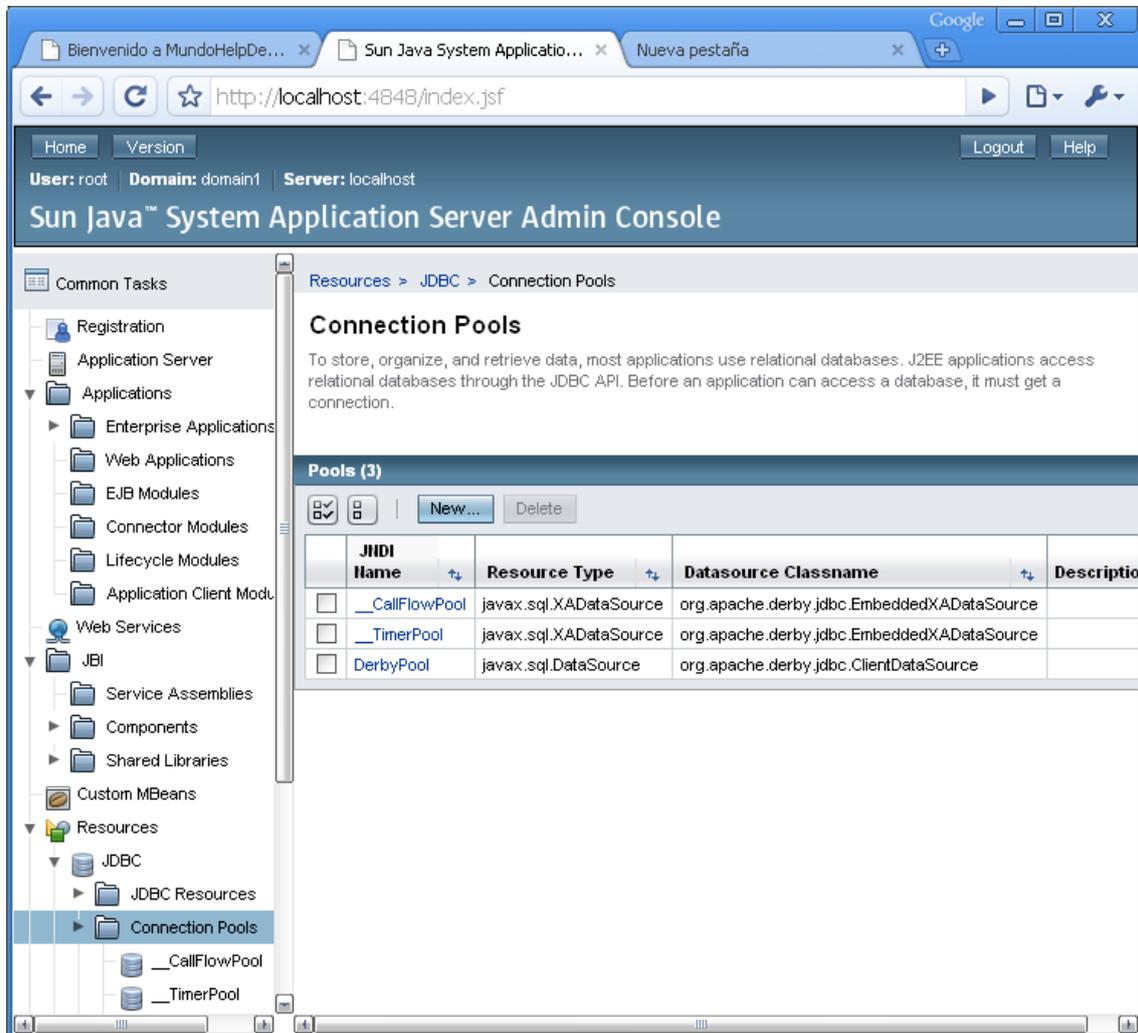
```
Start Default Server
Starting Domain domain1, please wait.
Log redirected to C:\Sun\SDK\domains\domain1\logs\server.log.
Redirecting output to C:\Sun\SDK\domains\domain1\logs\server.log
Domain domain1 is ready to receive client requests. Additional services
g started in background.
Domain [domain1] is running [Sun Java System Application Server 9.1_02
4-fcs] with its configuration and logs at: [C:\Sun\SDK\domains].
Admin Console is available at [http://localhost:4848].
Use the same port [4848] for "asadmin" commands.
User web applications are available at these URLs:
[http://localhost:8080 https://localhost:8181 ].
Following web-contexts are available:
[/web1 /__wstx-services ].
Standard JMX Clients (like JConsole) can connect to JMXServiceURL:
[service:jmx:rmi:///jndi/rmi://MOUILACER:8686/jmxrmi] for domain manage
oses.
Domain listens on at least following ports for connections:
[8080 8181 4848 3700 3820 3920 8686 ].
Domain does not support application server clusters and other standalon
es.

Presione una tecla para continuar . . . _
```

5.3.5. Para configurar el servidor de aplicaciones se debe abrir un navegador. En la barra de direcciones se digita <http://localhost:4848/> y se abre la ventana de inicio.

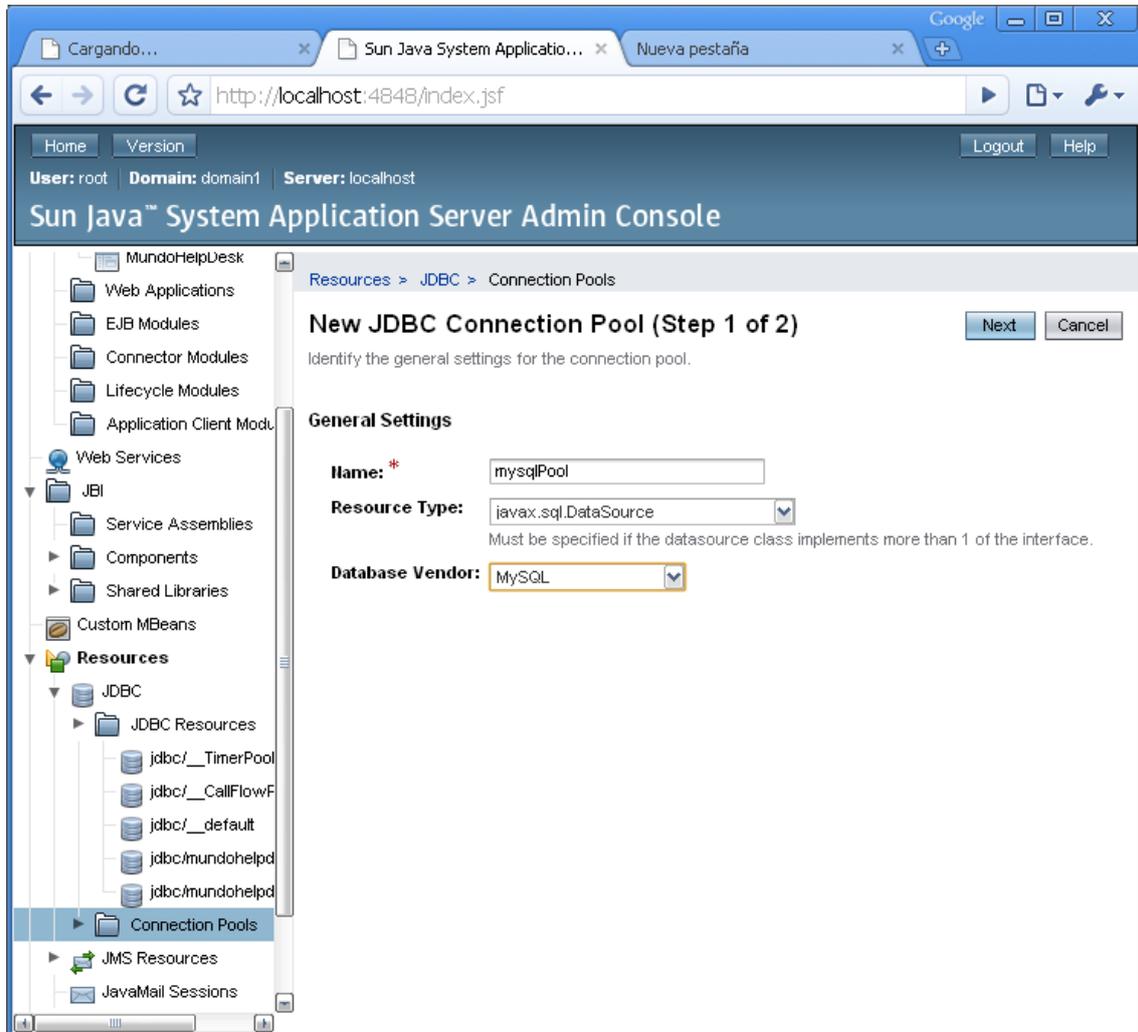


5.3.6. Luego de iniciar sesión, lo primero que se hace es crear el pool de conexión para MySQL, para esto hay que dirigirse a Resources, JDBC, Connection Pools.

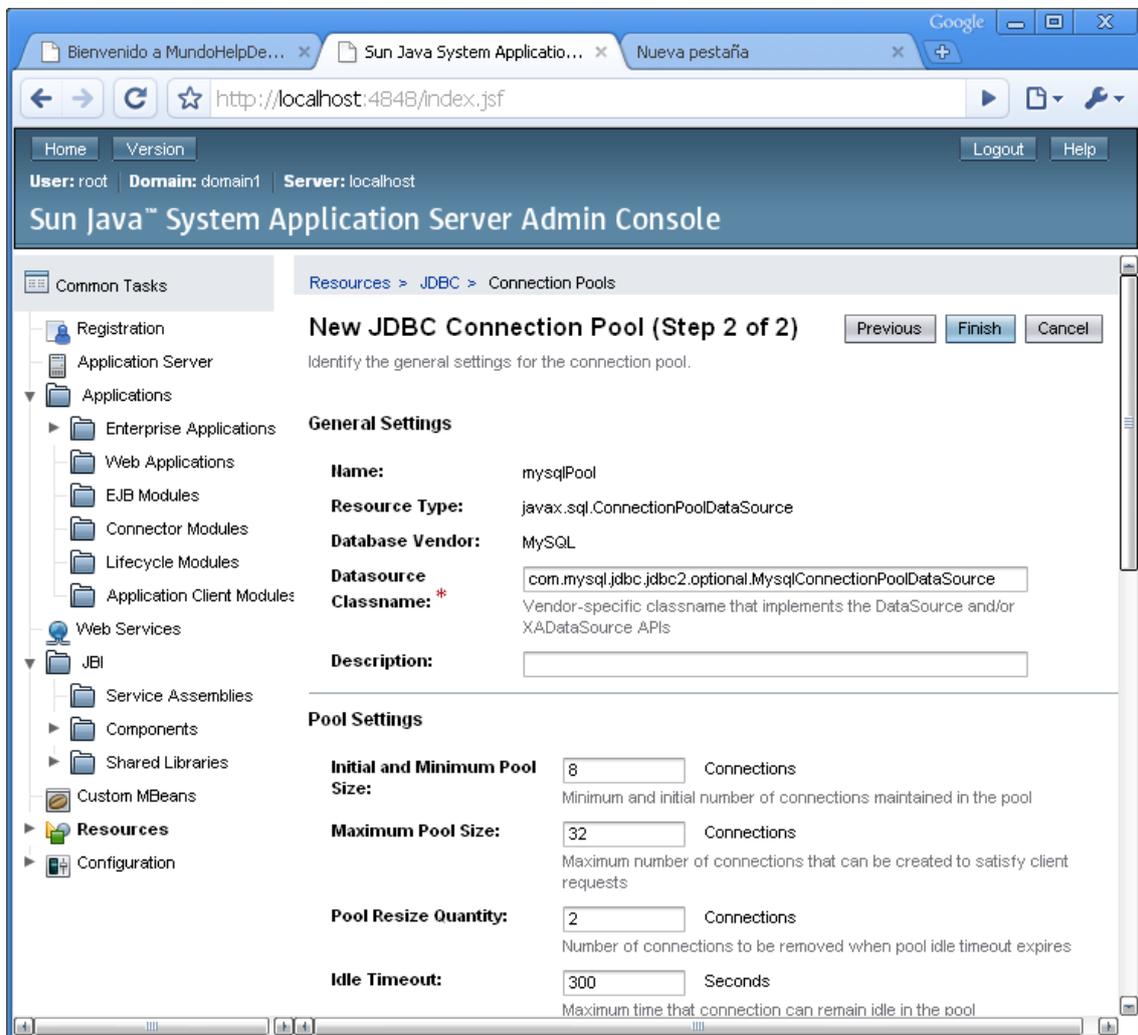


5.3.7. Abierta la ventana Conection Pools dar clic en New... para crear el pool de conexión.

5.3.8. Se debe llenar la información. En este caso Name es mysqlPool, Resource Type es javax.sql.DataSource y Database Vendor es MySQL. Llenada la información clic en Next.

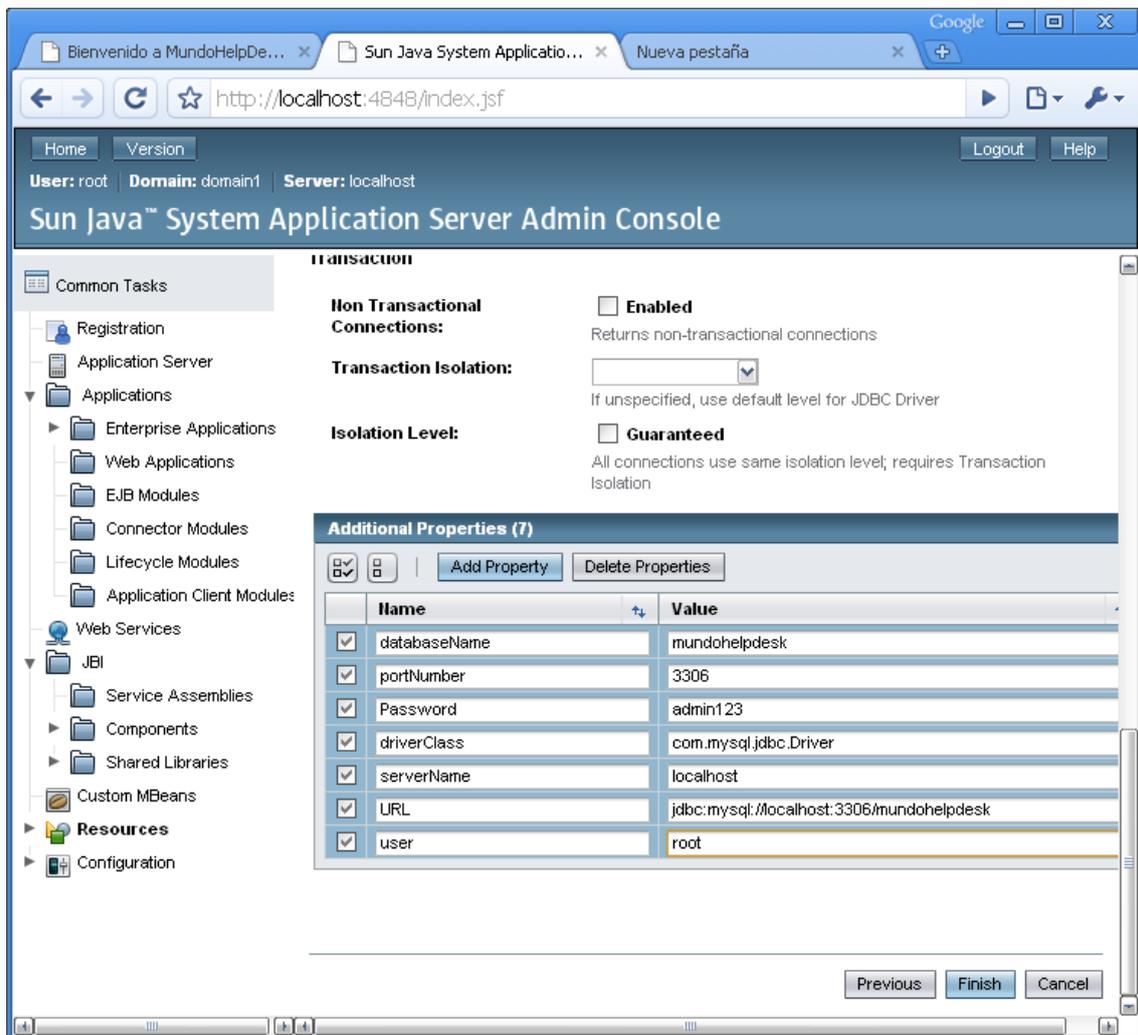


5.3.9. En este paso se configura todo lo referente a la conexión con MySQL. En la primera sección se dejó toda la información por defecto.



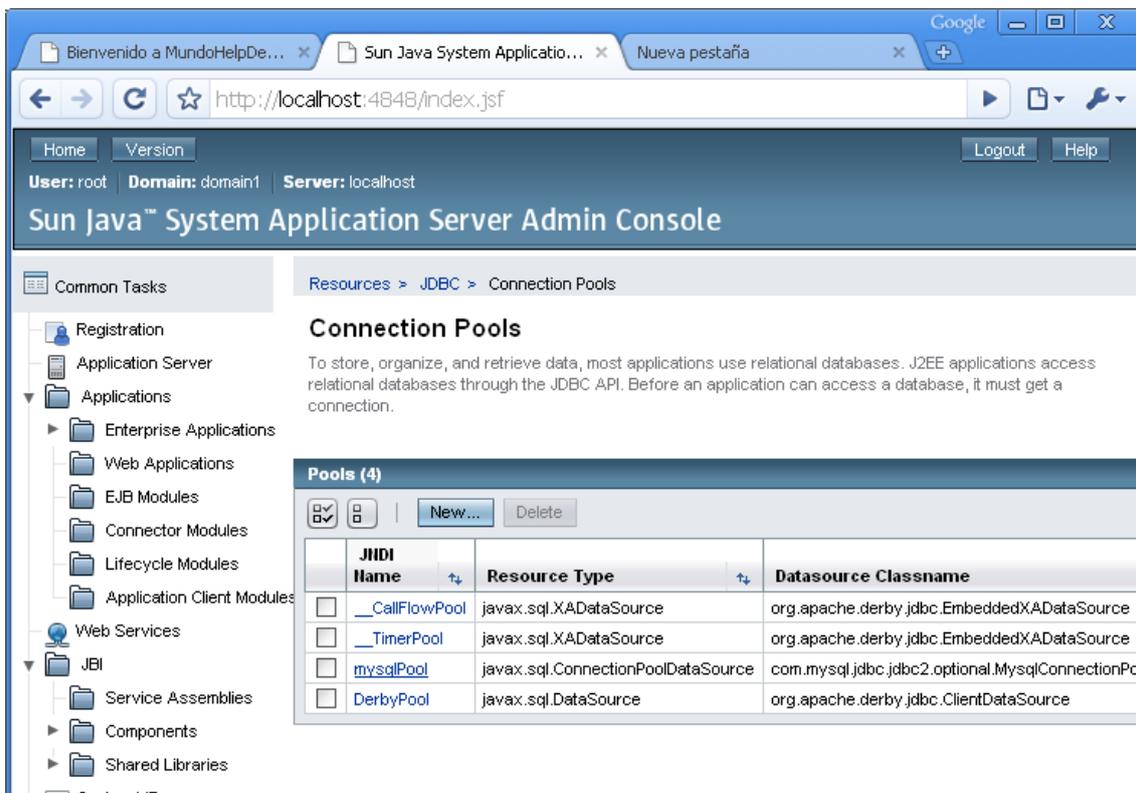
5.3.10. Terminada la configuración básica, hay que ir hasta el final de la página y creamos las variables que nos van a servir para interactuar con la base de datos, las cuales son:

- **databaseName:** mundohelpdesk
- **portNumber:** 3306
- **user:** root
- **Password:** clave
- **driverClass:** com.mysql.jdbc.Driver
- **serverName:** localhost

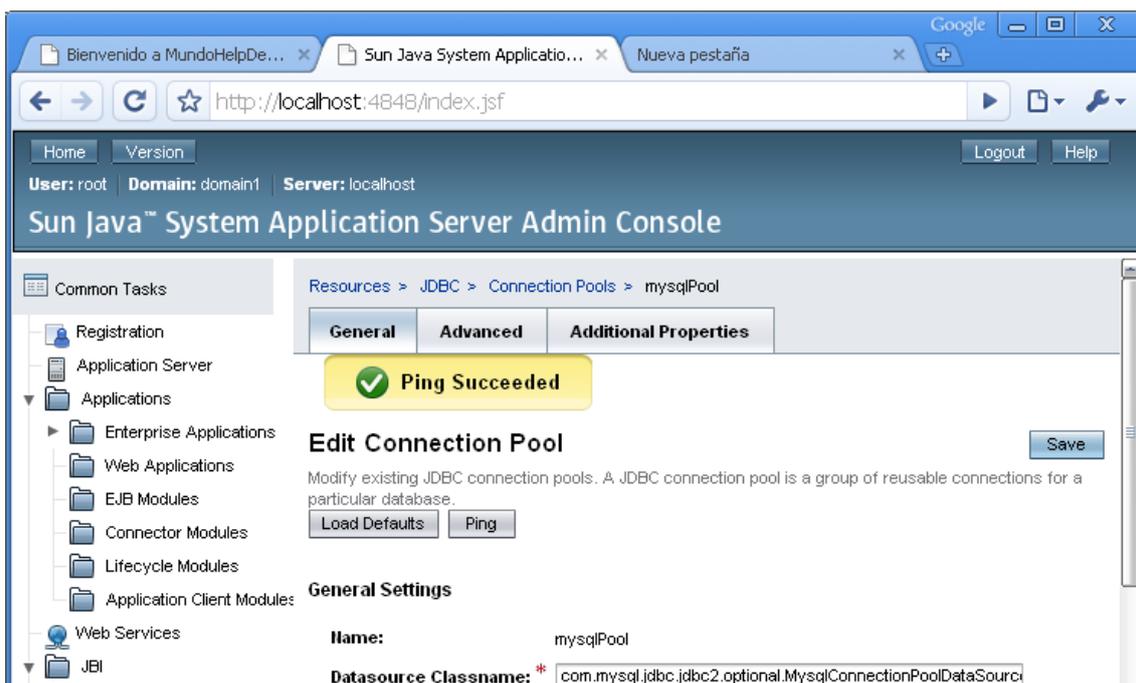


5.3.11. Luego de configurar las variables clic en Finish.

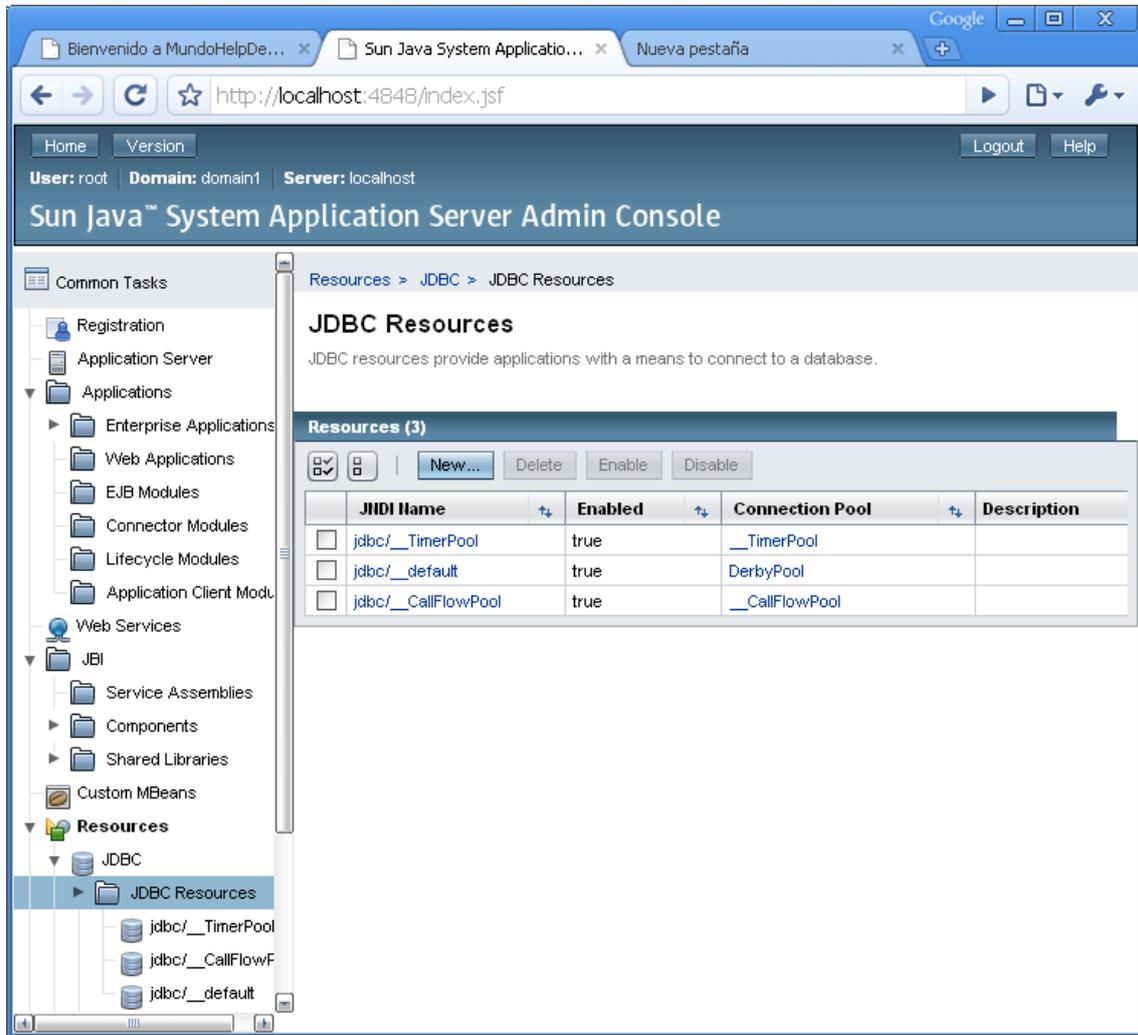
5.3.12. Para comprobar que la configuración se ha realizado correctamente seleccionamos la conexión que acabamos de crear.



5.3.13. En la página que se abrió hay que dar un clic en el botón que dice Ping. Si es correcto aparece un mensaje que dice Ping Suceded.

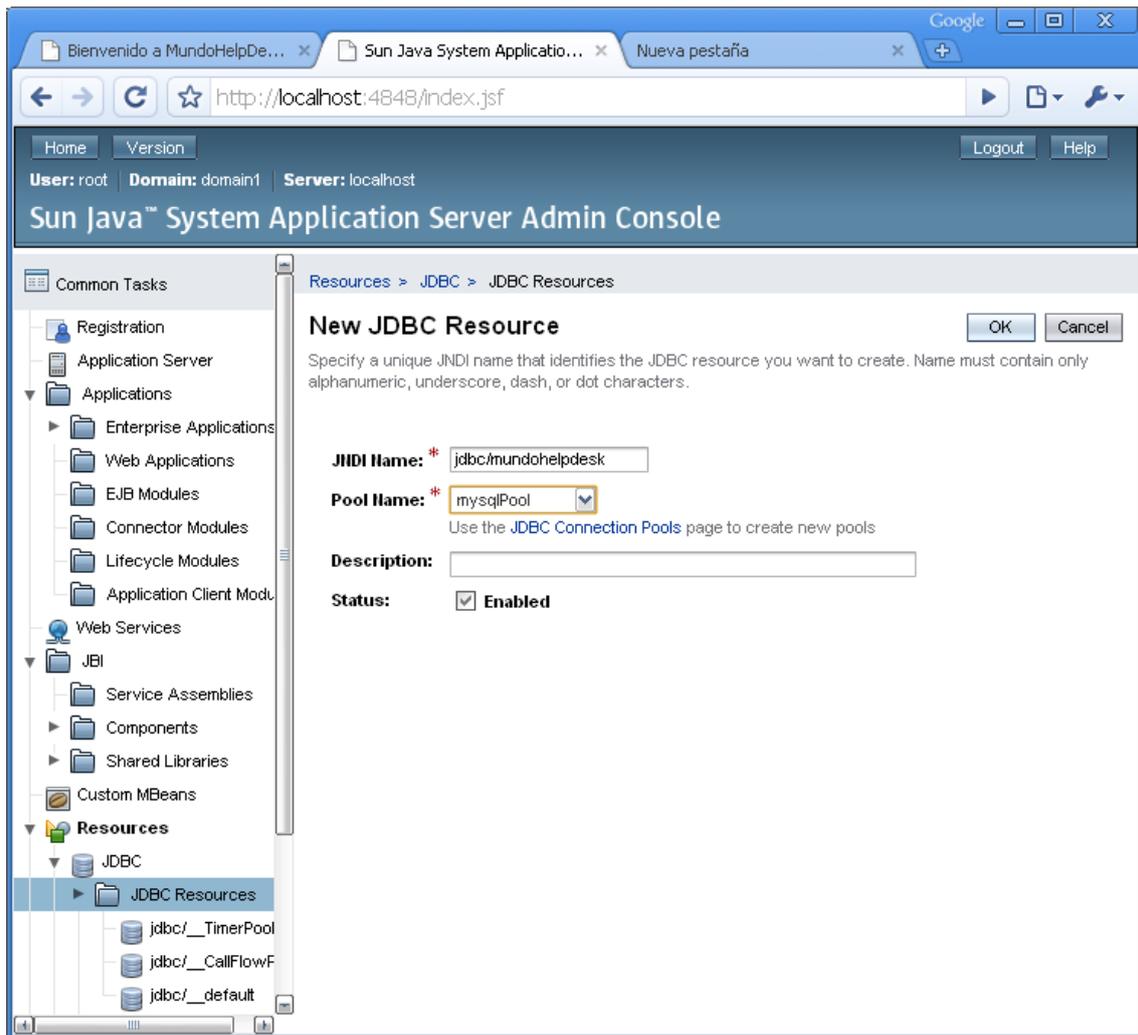


5.3.14. Luego de verificar que la conexión creada es correcta hay que crear los JDBC's necesarios para el funcionamiento del sistema. Para esto hay que ir a la página Resources – JDBC – JDBC Resources del menú de opciones.

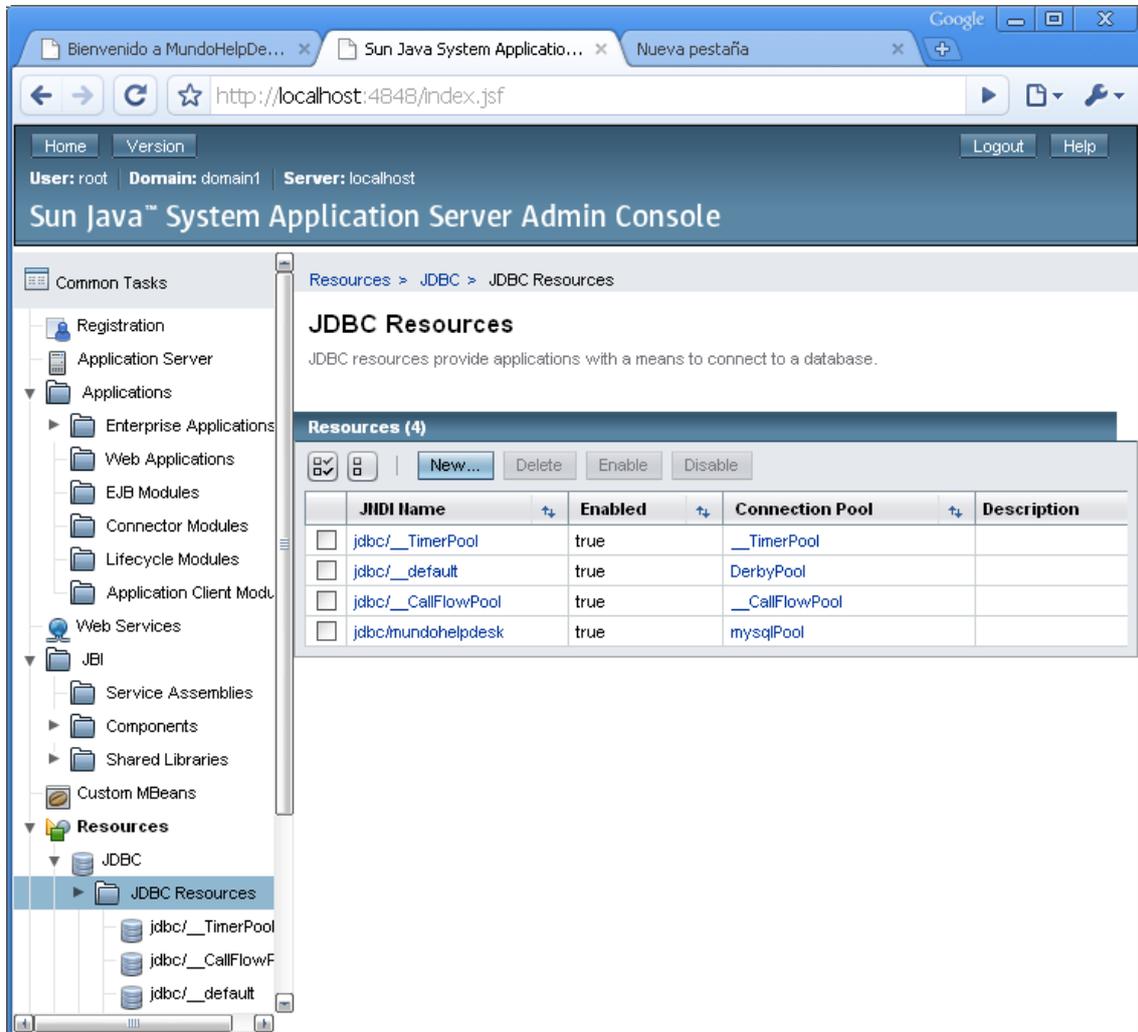


5.3.15. En la página de creación del JDBC se llena la información requerida. En este caso se llenó con la siguiente información.

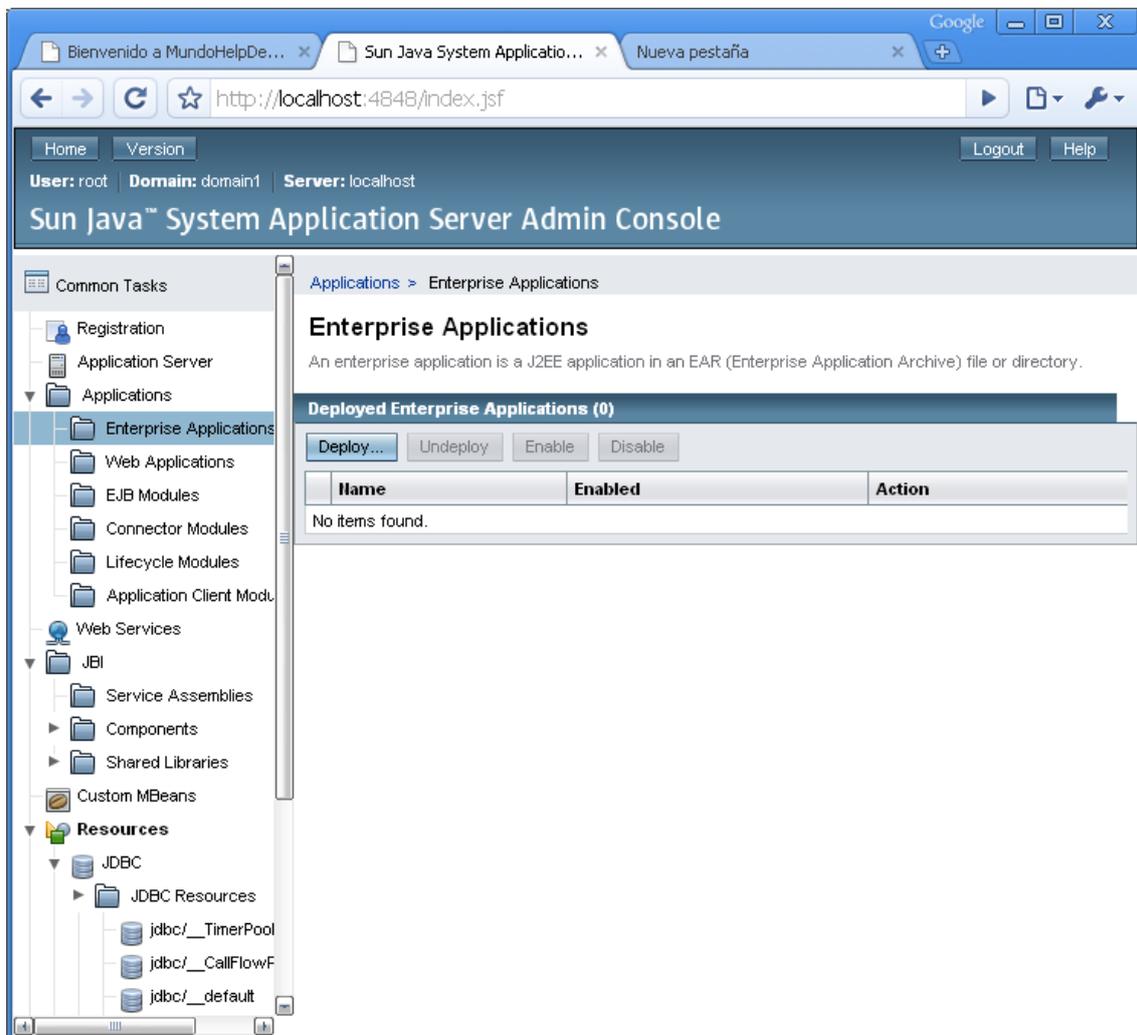
- **JNDI Name:** jdbc/mundohelpdesk
- **Pool Name:** mysqlPool (Pool creado anteriormente)



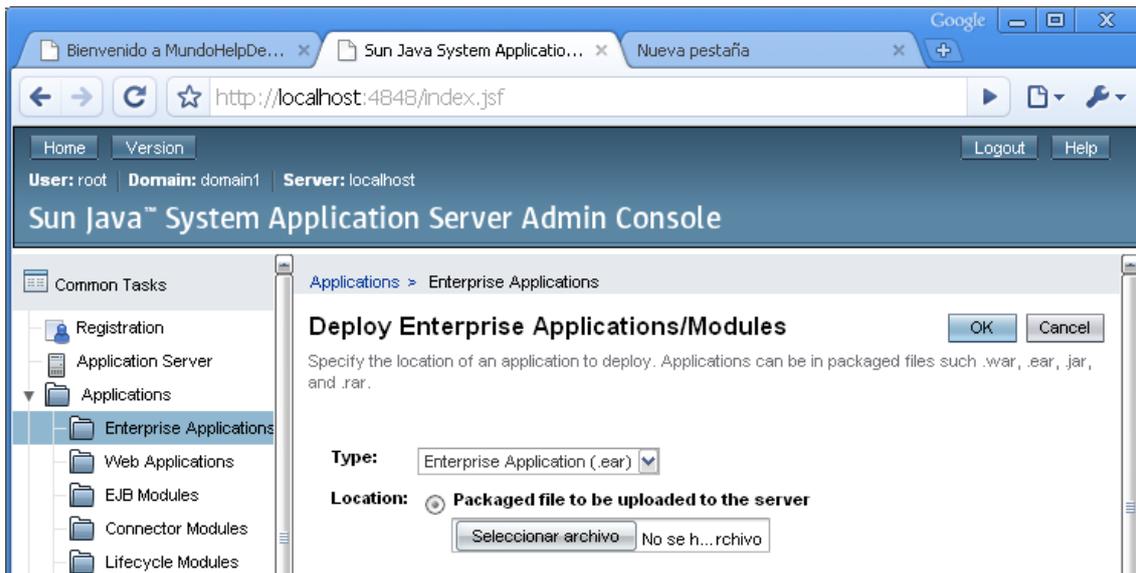
5.3.16. Para confirmar que se haya creado, se busca que el JDBC haya sido creado en la lista Resources.



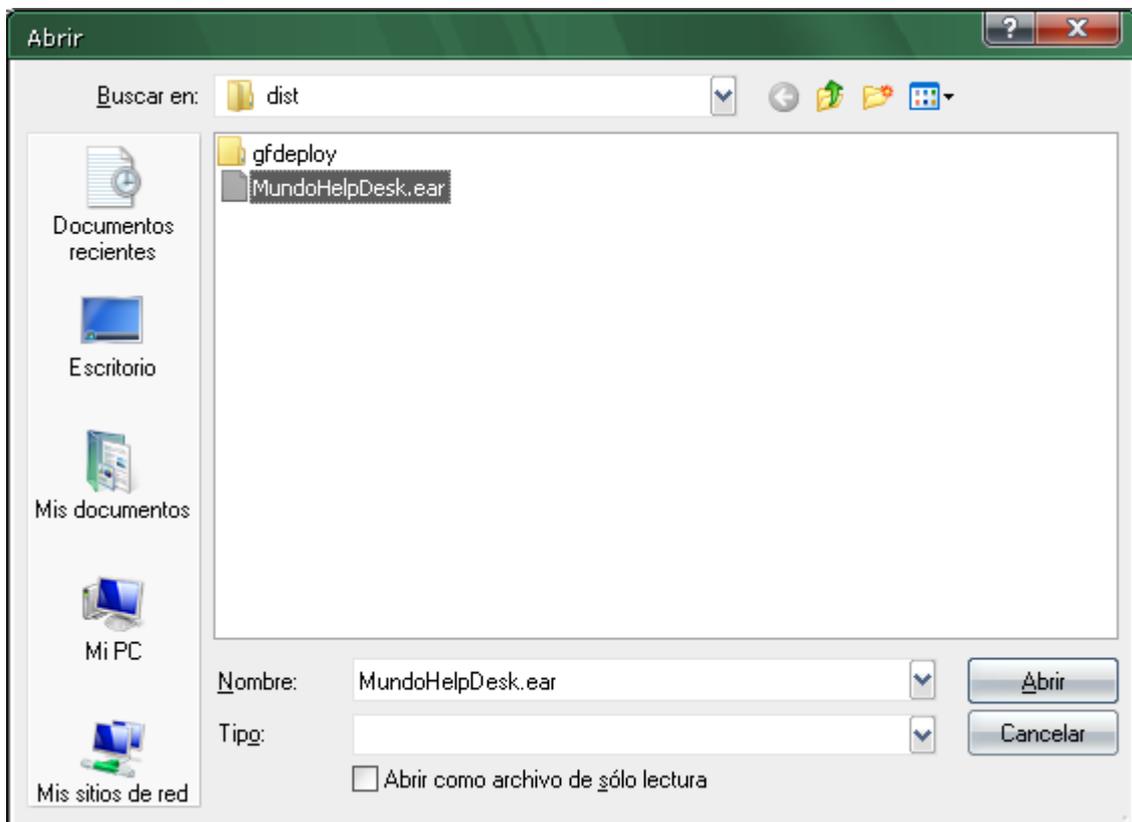
5.3.17. En este momento ya se puede realizar el levantamiento del sistema en el servidor de aplicaciones. Para esto hay que ir a Applications – Enterprise Applications y en la página que se muestra dar clic en Deploy...



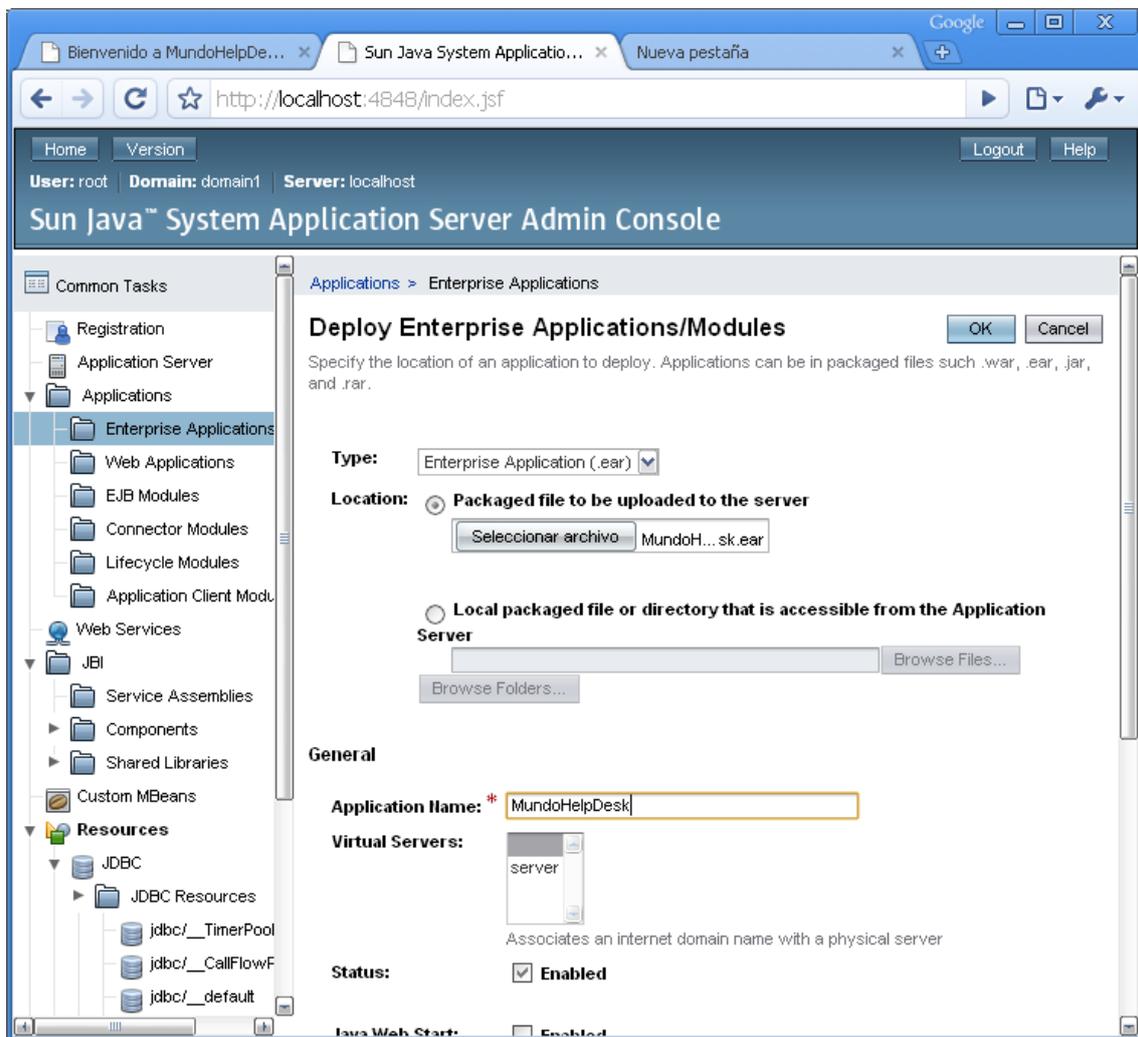
5.3.18. En la página para levantar la aplicación hay que clic en el botón Seleccionar archivo.



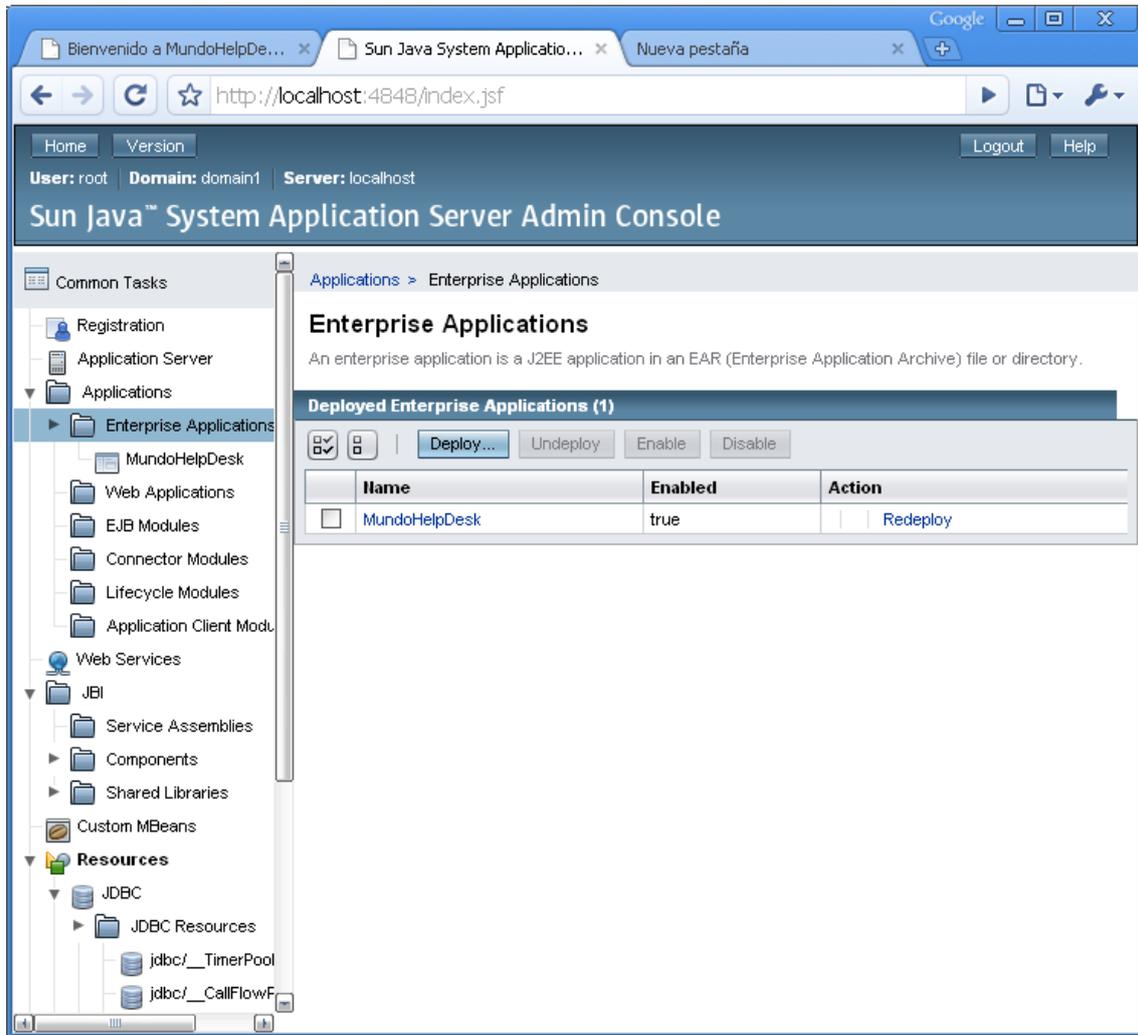
5.3.19. Se localiza el archiva EAR creado al momento de correr el sistema desde el NetBeans. El archivo se localiza dentro de la carpeta del proyecto, en la carpeta llamada dist.



5.3.20. Luego de seleccionar el archivo clic en OK.



5.3.21. Se puede confirmar que la aplicación se levantó sin inconvenientes si al momento de dar OK no se mostró ningún mensaje en la página y porque el nombre del sistema se encuentra en la tabla Deployed Enterprise Applications.



5.3.22. Finalmente en la barra de direcciones se digita el URL del sistema. En este caso <http://localhost:8080/MundoHekoDesk-war/> y se despliega la página de inicio del sistema.



5.4. MANUAL DE USUARIO

Vea el anexo A.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES:

- La investigación bibliográfica dio como resultado el desarrollo del marco teórico de referencia, el cual permitió entender el funcionamiento de los temas tratados y ayudó al diseño y desarrollo de la aplicación.
- El documento de requerimientos debe ser generado antes de comenzar a diseñar y modelar la aplicación, porque es fundamental conocer los requisitos funcionales y no funcionales del aplicativo.
- La metodología de desarrollo MIDAS, permite separar el trabajo de los analistas y de los programadores, gracias a sus tres etapas de modelamiento, donde los analistas realizan los diagramas correspondientes a la lógica del negocio y los programadores se incorporan en la etapa final del modelamiento.
- La investigación y el uso de la arquitectura J2EE en el desarrollo de la aplicación dio como resultado un sistema distribuido estable, multiplataforma y que no exige mayores requerimientos en las máquinas de los clientes.
- Los diagramas obtenidos del modelamiento de datos son una ayuda para el programador porque junto con las herramientas de modelado se puede generar código fuente base de algunos lenguajes de programación.
- El IDE NetBeans, utilizado para el desarrollo de la aplicación aportó mucho en el desarrollo del Sistema, porque incorpora módulos para el desarrollo de aplicaciones empresariales y para desarrollar de forma gráfica las páginas JSF. También porque genera automáticamente el archivo empaquetado necesario para levantar el sistema en cualquier servidor de aplicaciones que soporte la tecnología J2EE.

- El IDE aportó también en la generación automática de código, lo que facilitó y agilitó el desarrollo de los diversos módulos del sistema, en especial en la generación de los EJBs de persistencia y los de sesión.
- El servidor de aplicaciones GlassFish posee un módulo de administración el cual permite subir el sistema de forma fácil y rápida, realizando solamente la configuración de la conexión con el servidor base de datos.
- Las herramientas de software libre utilizadas se encuentran publicadas en los sitios Web de las empresas desarrolladoras como lo es la Sun Microsystems o en páginas Web dedicadas al alojamiento de archivos como Softonic.
- Las herramientas de software libre permiten el libre acceso, descarga, utilización y distribución del software, lo que lo hace muy atractivo para desarrolladores y empresas privadas o públicas que no desean gastar dinero en licencias.
- Las mesas de servicios cumplen con una gran responsabilidad dentro de la empresa, ya que son las que reciben todas las peticiones de servicio de los clientes internos de una organización, por lo que el personal encargado de atender las solicitudes debe estar capacitado en atención al cliente.
- La utilización de ITIL fue fundamental para entender el funcionamiento y las responsabilidades de una mesa de servicios junto con los procesos involucrados, porque define el ciclo de funcionamiento completo de ésta.
- Para la investigación de los temas requeridos en esta tesis, fue fundamental el acceso a Internet, por la cantidad de información gratuita publicada en los diferentes sitios Web.
- La Internet es una de las herramientas más poderosas que existen en el proceso de aprendizaje por estar al alcance de todos, entregando conocimiento gratuito a los que lo necesiten.

- Los navegadores funcionan de forma similar, sin embargo, esto no quiere decir que todos soportan los mismos componentes que existen en las aplicaciones Web, ya que cada navegador está orientado a cierta tecnología.

6.2. RECOMENDACIONES:

- Se recomienda mantener actualizada la base de datos de errores y soluciones, ya que es la piedra angular de un funcionamiento óptimo de la gestión de incidentes.
- Al desarrollar una aplicación Web se debe tener en cuenta que los componentes utilizados funcionen en la mayoría de navegadores. Si se da el caso de no funcionar se deben utilizar componentes estándar para solucionar el problema.
- Los técnicos encargados de la gestión de configuraciones deben tener mucho cuidado con el manejo de la información, ya que poseen permisos de acceso y actualización de gran cantidad de información privada de la empresa.
- Es recomendable la utilización de sistemas Web cuando los requerimientos de los usuarios exijan el acceso continuo e ininterrumpido al sistema, incluyendo el acceso desde fuera la empresa o fuera del país.
- Se debe tener mucho cuidado con la seguridad al desarrollar sistemas Web, porque a diferencia de una aplicación de escritorio al existir una falla en la seguridad de la aplicación cualquier persona podría tener acceso a información privada de la empresa.
- Se recomienda el desarrollo de aplicaciones en JAVA por ser herramientas gratuitas y porque el producto terminado puede ser implementado en diferentes plataformas.

- Es recomendable utilizar la tecnología J2EE para el desarrollo de aplicaciones distribuidas por la estandarización y la facilidad de reutilizar componentes, por los servicios de seguridad prestados y por la facilidad de interactuar con bases de datos.

BIBLIOGRAFÍA

- Información proporcionada en la materia “Ingeniería de Software I”
- Curso: “Building EJB’s with J2EE” dictado en la EPN
- <http://www.dric.com.mx/help-desk/itil-y-la-mesa-de-ayuda.html>
- [http://en.wikipedia.org/wiki/Incident_Management_\(ITSM\)](http://en.wikipedia.org/wiki/Incident_Management_(ITSM))
- http://ca.com/files/technologybriefs/incident_mgmt_tb.pdf
- [http://en.wikipedia.org/wiki/Configuration_Management_\(ITSM\)](http://en.wikipedia.org/wiki/Configuration_Management_(ITSM))
- <http://www.itil-itsm-world.com/itil-1.htm>
- <http://www.itil-survival.com/ITILChangeManagement.html>
- [http://en.wikipedia.org/wiki/Change_Management_\(ITIL\)](http://en.wikipedia.org/wiki/Change_Management_(ITIL))
- <http://www.infra.com.au/ITIL/ITIL-change-management.asp>
- <http://www.itil-itsm-world.com/itil-5.htm>
- http://www.itilpeople.com/What%20is%20ITIL.htm#Release_Management
- <http://www.omg.org>
- <http://www.dsic.upv.es/workshops/dsdm06/files/dsdm06-16-Vara.pdf>
- <http://www.omg.org/docs/omg/03-06-01.pdf>
- <http://www.java.com/en/download/faq/j2ee.xml>
- <http://www.webopedia.com/TERM/J/J2EE.html>
- http://es.wikipedia.org/wiki/Enterprise_JavaBeans

MANUAL DE USUARIO

Ingreso al Sistema

Se Ingresar el nombre de usuario y la contraseña, y se da clic en ingresar.



Menú de acuerdo al tipo de usuario

El menú que se carga al lado izquierdo de la página, muestra distintas opciones de acuerdo al tipo de usuario que se logueó. Al existir cuatro roles distintos de usuario, existen cuatro prototipos de menú:

Administrador	Técnico de Incidentes	Técnico de Configuraciones	Cliente
---------------	-----------------------	----------------------------	---------

ADMINISTRACIÓN Administrar Usuarios Buscar Usuarios GESTIÓN DE INCIDENTES Registrar Incidente Buscar Incidente Buscar Soluciones GESTIÓN DE CONFIGURACIONES Administrar Errores Administrar Soluciones Administrar Error - Solución Administrar Ítems de Configuración Administrar Cambios Administrar Proveedores Administrar Equipos Administrar Departamentos PERSONAL Solicitud de Servicio Cambiar Datos Personales Cambiar Contraseña	GESTIÓN DE INCIDENTES Registrar Incidente Buscar Incidente Buscar Soluciones PERSONAL Cambiar Datos Personales Cambiar Contraseña	GESTIÓN DE CONFIGURACIONES Administrar Errores Administrar Soluciones Administrar Error - Solución Administrar Ítems de Configuración Administrar Cambios Administrar Proveedores Administrar Equipos Administrar Departamentos PERSONAL Cambiar Datos Personales Cambiar Contraseña	PERSONAL Solicitud de Servicio Cambiar Datos Personales Cambiar Contraseña
--	---	---	--

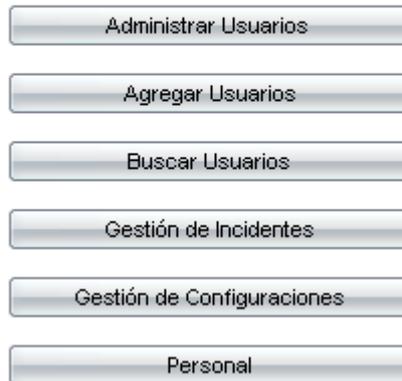
En la parte superior se muestra la información de la página en la que se está navegando, la identificación del usuario que esta logueado y opciones para cerrar sesión e ir al inicio.



Administración

En la página principal de cada grupo de opciones se muestra un menú de opciones del grupo escogido.

Menú de Opciones



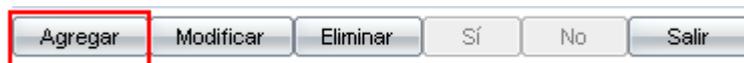
Administrar Usuarios

En la página de administración de usuarios se ubican los botones agregar, modificar, eliminar, sí, no y salir.



Agregar Usuario

Al dar clic en el botón agregar, se abre una página nueva.



La página para agregar un usuario posee en la parte superior dos botones: agregar y salir. El botón agregar crea el registro en la base de datos y el botón salir regresa a la página de administración de usuarios.

Agregar Usuario



Hola, Galo Solís | [Cerrar Sesión](#) | [Ir al Inicio](#)

Hombre:	<input type="text"/>
Apellido:	<input type="text"/>
Mail:	<input type="text"/>
Alias:	<input type="text"/>
Contraseña:	<input type="text"/>
Tipo Usuario:	Administrador <input type="button" value="v"/>
Departamento:	General <input type="button" value="v"/>

En el caso de no haber ingresado datos en los campos requeridos, el sistema muestra una lista de errores donde se especifica qué campos no pueden estar en blanco.

- ◆ **Hombre: Error de Validación: Valor es necesario.**
- ◆ **Apellido: Error de Validación: Valor es necesario.**
- ◆ **Mail: Error de Validación: Valor es necesario.**
- ◆ **Alias: Error de Validación: Valor es necesario.**
- ◆ **Contraseña: Error de Validación: Valor es necesario.**

El sistema también valida que no existan alias repetidos al momento de crear al registro. Igualmente el mensaje de error aparece bajo los botones.

El alias se encuentra en uso

Hombre:	María
Apellido:	Fernandez
Mail:	mhernandez@gmail.com
Alias:	admin
Contraseña:	*****
Tipo Usuario:	Administrador <input type="button" value="v"/>
Departamento:	General <input type="button" value="v"/>

Si la información ingresada es correcta el registro es creado en la base de datos y se muestra un mensaje bajo los botones que dice que el usuario se ha agregado correctamente.

Usuario agregado satisfactoriamente

Hombre:

Apellido:

Mail:

Contraseña:

Tipo Usuario:

Departamento:

Buscar Usuarios

La página de búsqueda de usuario permite buscar y seleccionar el usuario que se desea modificar o eliminar.

Hombre

Apellido

Alias

Tipo Usuario

Departamento

Código: 5
Hombre: usuario1
Apellido: usuario1
Mail: usuario@mundohelpdesk.com
Alias: usuario1
Tipo Usuario: Cliente
Departamento: General

Usuarios								
Código	Hombre	Apellido	Mail	Alias	Contraseña	Tipo Usuario	Departamento	
1	Galo	Solís	admin@mhd.com	admin	*****	Administrador	Sistemas	
2	Lorena	Carrillo H	loreca@mhd.com	loreca	*****	Técnico de Incidentes	Sistemas	
3	Técnico	Incidentes	incidentes@mundohelpdesk.com	incidente	*****	Técnico de Incidentes	Sistemas	
4	Técnico	Configuraciones	configuraciones@mundohelpdesk.com	configuracion	*****	Técnico de Configuraciones	General	
5	usuario1	usuario1	usuario@mundohelpdesk.com	usuario1	*****	Cliente	General	
6	usuario2	usuario	usuario2@mundohelpdesk.com	usuario2	*****	Cliente	Sistemas	
7	usuario3	usuario3	usuario3	usuario3	*****	Cliente	Sistemas	
8	usuario	usuario	usuario@mundohelpdesk.com	usuario4	*****	Cliente	Sistemas	
9	Cris	Cartes	usuario@mundohelpdesk.com	usuario5	*****	Cliente	Administración	
11	Usuario	Apellido	usuario7@mundohelpdesk.com	usuario789	*****	Cliente	Sistemas	

Página: 1 de 2

Si no se selecciona un registro antes de dar clic en el botón modificar o eliminar, se mostrará un mensaje indicando que se debe seleccionar un elemento de la lista.



Primero debe SELECCIONAR un elemento de la lista.

Modificar Usuario

La página para modificar un usuario es similar a la de agregar. La opción aceptar actualiza el registro en la base de datos y la opción salir vuelve a la página de administración de usuarios. Si se deja el campo contraseña en blanco ésta se queda igual y si se lo llena la contraseña se actualiza.

Una captura de pantalla de la interfaz de usuario. En la parte superior izquierda, el título 'Modificar Usuario' está en un recuadro azul. A la derecha, el logo 'mhd' y 'MUNDO HELP DESK' están en un recuadro azul. Debajo del logo, el texto 'Hola, Galo Solis' está acompañado de los enlaces 'Cerrar Sesión' y 'Ir al Inicio'. En la parte inferior izquierda, hay dos botones: 'Aceptar' y 'Salir'.

Código:	3
Nombre:	Técnico
Apellido:	Incidentes
Mail:	incidentes@mundohelpdesk.c
Alias:	incidente
Contraseña:	
Tipo Usuario:	Técnico de Incidentes ▼
Departamento:	Sistemas ▼

Eliminar Usuario

Para eliminar un usuario se debe seleccionar un registro y dar clic en el botón eliminar.



Al dar clic en el botón eliminar se deshabilitan los botones modificar y eliminar, y se habilitan los botones Sí y No.

Al dar clic en Sí el registro se elimina.



Al eliminarse un registro se muestra un mensaje informando que el usuario ha sido eliminado.



Gestión de Incidentes

Menú de opciones para la gestión de incidentes.

Menú de Opciones



Registrar Incidente

La página de registro de un incidente se llena la información del incidente, se selecciona el cliente que lo reportó y finalmente se crea el registro del incidente.

Registrar Incidente



Hola, Técnico Incidentes | [Cerrar Sesión](#) | [Ir al Inicio](#)

Fecha: ^ 2008-11-19

Descripción: ^

Mensaje:

Clasificación: ^ Nivel 1

Estado: ^ Registrado

Prioridad: ^ Alta

Cliente: ^

Luego de llenar la información del incidente se debe seleccionar el cliente dando clic en el botón ... que abre una nueva página en la que se selecciona el cliente.

Fecha: ^ 2008-11-19

Descripción: ^ Problema de Red

Mensaje:

Clasificación: ^ Nivel 1

Estado: ^ Registrado

Prioridad: ^ Media

Cliente: ^

Luego de seleccionar el cliente se da clic en el botón Seleccionar que regresa a la página de registro de incidente con el usuario seleccionado.

Seleccionar Usuario



Hola, Técnico Incidentes | [Cerrar Sesión](#) | [Ir al Inicio](#)

Seleccionar

Nombre
 Apellido
 Alias

Departamento

Código: 5
Nombre: usuario1
Apellido: usuario1

Mail: usuario@mundohelpdesk.com
Alias: usuario1
Departamento: General

Usuarios

Código	Nombre	Apellido	Mail	Alias	Departamento
1	Galo	Solís	admin@mhd.com	admin	Sistemas
2	Lorena	Carrillo H	loreca@mhd.com	loreca	Sistemas
3	Técnico	Incidentes	incidentes@mundohelpdesk.com	incidente	Sistemas
4	Técnico	Configuraciones	configuraciones@mundohelpdesk.com	configuracion	General

Fecha:

Descripción:

Mensaje:

Clasificación:

Estado:

Prioridad:

Cliente:

Finalmente se le da clic en el botón agregar y si los datos ingresados fueron correctos se muestra un mensaje que el incidente se registró satisfactoriamente.

El incidente se ha registrado satisfactoriamente

Buscar Incidente

La página de búsqueda de incidentes permite buscar un incidente para poder modificar el incidente o cerrarlo.

Buscar Incidente



Hola, Técnico Incidentes | [Cerrar Sesión](#) | [Ir al Inicio](#)

[Registrar](#) [Modificar](#) [Cerrar Incidente](#) [Salir](#)

Seleccione el tipo de búsqueda

Fecha

Descripción

Cliente

Estado

Prioridad

Clasificación

No atendidos

Técnico

Presentar Todos

Fecha:

Clasificación:

Estado:

Prioridad:

Cliente:

Descripción:

Mensaje:

Código	Fecha	Descripción	Clasificación	Estado	Prioridad	Cliente
1	2008-02-05	Falla de conexión con la BD	Nivel 1	Registrado	Media	usuario789
2	2008-09-08	No funciona el internet	Nivel 2	Cerrado	Baja	usuario11
3	2008-11-08	No funcionan los parlantes	Nivel 1	Cerrado	Baja	usuario8
4	2008-06-08	Capacitación de Microsoft Excel 2007	Nivel 1	Resuelto	Media	usuario8
5	2007-05-09	No se enciende el monitor	Problema	Cerrado	Alta	felipe

Página: 1 de 3

Modificar Incidente

Al seleccionar un elemento y dar clic en Modificar se abre una página con la información histórica del incidente. Cada vez que se modifica un incidente se crea un registro histórico para saber que no más ha sucedido con ese incidente.

[Modificar](#) [Salir](#)

Fecha: 2008-02-05

Descripción:

Mensaje:

Clasificación:

Estado:

Prioridad:

Cliente: usuario789

Historial Incidente

ID	Fecha	Descripción	Mensaje	Clasificación	Estado	Prioridad	Técnico
1	2008-02-05	Falla de conexión con la BD	n/a	Nivel 1	Registrado	Media	lorece

Cerrar Incidente

Para cerrar un incidente se debe seleccionar la solución con la que se solventó el problema. Para esto se da clic en el botón seleccionar solución.

Cerrar Incidente Salir

Fecha: 2008-02-05
Descripción: Falla de conexión con la BD
Mensaje: n/a

Clasificación: Nivel 1
Estado: Cerrado
Prioridad: Media
Cliente: usuario789

Seleccionar Solución

Código:
Nombre:
Descripción:
Causa:
Tipo Error:

Código:
Nombre:
Descripción:

Se abre una nueva página en la cual se selecciona el error que dio lugar al incidente y la solución. Con el error y la solución cargados se da clic en el botón seleccionar para volver a la página de cierre de incidente con la solución cargada.

Seleccionar

Ingrese criterio de búsqueda

Buscar Error

Código: 6 **Código:** 7
Nombre: **Nombre:** Reiniciar Servicio
Descripción: Error de BD **Descripción:** Si un servicio está colgado, se debe reiniciar el servicio.
Causa: Servidor fuera de servicio
Tipo Error: hardware

Errores Conocidos					
Código	Nombre	Descripcion	Causa	Causa	
6		Error de BD	Servidor fuera de servicio	hardware	

Página: 2 de 2

Soluciones		
Código	Nombre	Descripcion
7	Reiniciar Servicio	Si un servicio está colgado, se debe reiniciar el servicio.

Para finalizar el cierre se da clic en el botón cerrar incidente.

Fecha: 2008-02-05
Descripción:
Mensaje:
Clasificación: Nivel 1
Estado:
Prioridad: Media
Cliente: usuario789

Código: 6
Nombre:
Descripción: Error de BD
Causa: Servidor fuera de servicio
Tipo Error: hardware
Código: 7
Nombre: Reiniciar Servicio
Descripción:

En la página de búsqueda de incidentes al seleccionar un registro que ya se encuentre cerrado, aparecerá una opción para ver el cierre del incidente.

Fecha: 2008-02-05
Clasificación: Nivel 1
Estado: Cerrado
Prioridad: Media
Cliente: usuario789
Cód. Cierre: 7
Descripción:
Mensaje:

Ver Cierre

Para ver el cierre hay que dar clic en el botón Ver Cierre, el cual abre una nueva página con la información del cierre.

Seleccione el tipo de búsqueda

Fecha
 Descripción
 Cliente
 Estado
 Prioridad
 Clasificación
 No atendidos
 Técnico
 Presentar Todos

Fecha: 2008-09-08
Clasificación: Nivel 2
Estado: Cerrado
Prioridad: Baja
Cliente: usuario11
Cód. Cierre: 2
Descripción:
Mensaje:

En la página de incidente cerrado solo se puede visualizar la información. El botón salir vuelve a la página de búsqueda de incidentes.

Incidente Cerrado



Hola, Galo Solis | [Cerrar Sesión](#) | [Ir al Inicio](#)

Fecha: 2008-11-08
Descripción: No funcionan los parlantes
Mensaje: n/a

Clasificación: Nivel 1
Estado: Cerrado
Prioridad: Baja
Cliente: usuario8

ERROR:

Código: 2
Nombre: error 2
Descripción: descripcion error 2
Causa: falla 233
Tipo Error: software

SOLUCIÓN:

Código: 2
Nombre: Conectar Cable

Descripción:

Verificar que el cable se encuentre conectado, si está desconectado reconectelo

Buscar Soluciones

La opción de buscar soluciones es una herramienta de ayuda para los técnicos de incidentes. En esta página se pueden buscar los errores conocidos y seleccionando el error del que se desea conocer las soluciones se cargan todas las soluciones relacionadas al error en la segunda tabla.

Ingrese criterio de búsqueda

Código:
Nombre:
Descripción:
Causa:
Tipo Error:

Código:
Nombre:

Descripción:

Errores Conocidos

Código	Nombre	Descripción	Causa	Causa
1	Falla de scanner	descripcion error 1	falla de corriente	hardware
2	error 2	descripcion error 2	falla 233	software
3	Falla Fuente	La fuente de poder no funciona	la fuente está quemada	hardware
5	Cambiar Cartucho	Canbciar cartucho	Imprime lineas en blanco	hardware
6		Error de BD	Servidor fuera de servicio	hardware

Soluciones

Código	Nombre	Descripción
No se han encontrado elementos.		

Para buscar el error se ingresan las palabras clave en la caja de texto y se da clic en el botón Buscar Error.

Ingrese criterio de búsqueda

Se da clic en el código del registro y se visualiza las posibles soluciones en la segunda tabla.

Errores Conocidos				
Código	Nombre	Descripcion	Causa	Causa
1	Falla de scanner	descripcion error 1	falla de corriente	hardware
2	error 2	descripcion error 2	falla 233	software
6		Error de BD	Servidor fuera de servicio	hardware

Soluciones		
Código	Nombre	Descripcion
7	Reiniciar Servicio	Si un servicio está colgado, se debe reiniciar el servicio.

Gestión de Configuraciones

Menú para el técnico de configuraciones.

Menú de Opciones

- Administrar Errores
- Administrar Soluciones
- Administrar Soluciones a Errores
- Administrar Ítems de Configuración
- Administrar Cambios
- Administrar Proveedores
- Administrar Equipos
- Administrar Departamentos

Administrar Equipos

Pantalla de administración

Las pantallas de administración son similares para todos los casos y tienen el siguiente formato. En la parte superior están los botones de agregar, modificar, eliminar, sí, no y salir, después se localiza la parte de búsqueda seguida de los campos para cargar la información del registro que se seleccione y finalmente la tabla con la información de la opción seleccionada.

Administración de Equipos

MUNDO HELP DESK

Hola, Galo Solis | [Cerrar Sesión](#) | [Ir al Inicio](#)

Código
 Departamento

Características
 Presentar Todos

Id:
Código:
Usuario Propietario:
Departamento:

Características:

ID	Código	Propietario	Características	Departamento
1	EQUI001	admin	Servidor	General
2	EQUI002	lorece	Pc Dual Core 2.0	Sistemas
3	EQUI004	configuracion	Laptop HP	Administración
4	EQUI003	pepe	Mac G5 ram 3gb video 256mb	Tesorería
5	EQUI005	dsalvador	PC Intel Piv 3.0 ram 1gb	Administración

Agregar

En las páginas de agregar un registro existen dos botones, agregar y salir. El botón agregar crea el registro en la base de datos y el salir vuelve a la página de administración.

Código:

Usuario:

Características:

Departamento:

Registro agregado satisfactoriamente

Modificar

En las páginas de modificación se carga la información del registro seleccionado para que se pueda editar y al dar clic en el botón aceptar se actualice el registro en la base de datos.

Id: 3
Código: *
Usuario: 4 ...
Características: *
Departamento: *

Eliminar

Para eliminar un registro se lo debe seleccionar y luego dar clic en eliminar.

Código
 Departamento
 Características
 Presentar Todos

Id: 4
Código: EQUI003
Usuario Propietario: pepe
Departamento: Tesorería

Características:

Mac G5
ram 3gb
video 256mb

Equipos						
ID	Código	Propietario	Características	Departamento		
1	EQUI001	admin	Servidor	General		
2	EQUI002	lorece	Pc Dual Core 2.0	Sistemas		
3	EQUI004	configuracion	Laptop HP	Administración		
4	EQUI003	pepe	Mac G5 ram 3gb video 256mb	Tesorería		
5	EQUI005	dsalvador	PC Intel Piv 3.0 ram 1gb	Administración		

Si se da clic en el botón No, el registro no es eliminado.

Código
 Características

Departamento
 Presentar Todos

Id: 4
Código: EQUI003
Usuario Propietario: pepe
Departamento:

Características:

Mac G5
 ram 3gb
 video 256mb

Si se da clic en Sí, el registro es eliminado, se muestra un mensaje que le informa al usuario que el registro se eliminó satisfactoriamente y se actualiza la tabla sin el elemento eliminado.

Código
 Características

Departamento
 Presentar Todos

Id: 4
Código: EQUI003
Usuario Propietario: pepe
Departamento:

Características:

Mac G5
 ram 3gb
 video 256mb

Equipos						
ID	Código	Propietario	Características	Departamento		
1	EQUI001	admin	Servidor	General		
2	EQUI002	lorece	Pc Dual Core 2.0	Sistemas		
3	EQUI004	configuracion	Laptop HP	Administración		
4	EQUI003	pepe	Mac G5 ram 3gb video 256mb	Tesorería		
5	EQUI005	dsalvador	PC Intel Piv 3.0 ram 1gb	Administración		

Búsquedas

Los módulos de administración tienen una sección de búsqueda, en la cual para buscar se debe seleccionar el criterio de búsqueda, ingresar las palabras clave y dar clic en el botón buscar.

Código
 Características

Departamento
 Presentar Todos

La información resultante de la búsqueda se presentará en la tabla que se encuentra en la parte inferior de la página.

Agregar Modificar Eliminar Sí No Salir

Búsqueda

Código
 Departamento Administración
 Características
 Presentar Todos

Id:
Código:
Usuario Propietario:
Departamento:

Características:

Equipos									
ID	↕	Código	↕	Propietario	↕	Características	↕	Departamento	↕
3		EQUI004		configuracion		Laptop HP		Administración	
5		EQUI005		dsalvador		PC Intel Piv 3.0 ram 1gb		Administración	

NOTA: En las páginas de administración de Errores y administración de Soluciones solamente hay un criterio de búsqueda. La búsqueda se realiza por el campo descripción.

Personal

En la opción de personal, el menú se puede cargar de dos formas. Si el usuario es el administrador o un usuario cliente aparecerá el menú completo, y si el usuario es un técnico de incidentes o un técnico de configuraciones aparecerán deshabilitadas las dos primeras opciones.

Menú de Opciones

Solicitar Servicio

Ver mis Solicitudes

Cambiar Contraseña

Cambiar Datos Personales

Menú de Opciones

Solicitar Servicio

Ver mis Solicitudes

Cambiar Contraseña

Cambiar Datos Personales

Solicitud de Servicio

Para solicitar un servicio se debe llenar la información solicitada en el formulario y dar clic en registrar solicitud. El campo Mensaje es opcional para los casos en los que se necesite especificar un mensaje de error.

Solicitar Servicio

MUNDO HELP DESK

Bienvenido, Cris Cartes | [Cerrar Sesión](#) | [Ir al Inicio](#)

Fecha: 2008-11-23

Descripción: Solicito capacitación en el uso de JBuilder

Mensaje:

Ciente: 1 admin

Solicitar Servicio

La solicitud incidente se ha enviado exitosamente

Ver Mis Solicitudes

La opción Ver Mis Solicitudes muestra una tabla con la información de las solicitudes o incidentes registrados a nombre del usuario que está logueado.

Ver Mis Solicitudes

MUNDO HELP DESK

Bienvenido, Cris Cartes | [Cerrar Sesión](#) | [Ir al Inicio](#)

Fecha: 2008-11-21
Clasificación: Nivel 1
Estado: Activo
Prioridad: Alta
Técnico: admin

Descripción: La impresora no responde a la orden de impresión

Mensaje:

Incidentes

Código	Fecha	Descripción	Clasificación	Estado	Prioridad	Técnico
8	2008-11-06	Peticion de Capacitaciones	Nivel 1	Registrado	Baja	
9	2008-11-08	Capacitación sobre el uso del sistema	Nivel 1	Activo	Media	lorece
10	2008-11-07	Solicito Capacitación sobre el nuevo sistema de contabilidad	Nivel 1	Registrado	Baja	
19	2008-11-21	La impresora no responde a la orden de impresión	Nivel 1	Activo	Alta	admin
20	2008-11-23	Solicito capacitación en el uso de JBuilder	Nivel 1	Registrado	Baja	

Cambiar Datos Personales

Para cambiar los datos personales se debe modificar los campos que se muestran en la página y luego dar clic en el botón aceptar.

Cambiar Datos Personales		MUNDO HELP DESK	
		Bienvenido, Cris Cartes Cerrar Sesión Ir al Inicio	
<input type="button" value="Aceptar"/>	<input type="button" value="Salir"/>		
Código:	9		
Nombre:	<input type="text" value="Cris"/>		
Apellido:	<input type="text" value="Cartes"/>		
Mail:	<input type="text" value="usuario@mundohelpdesk."/>		

Cambiar Contraseña

La opción de cambio de contraseña sirve para que el usuario logueado pueda cambiar la contraseña con la que ingresa al sistema.

Cambiar Contraseña		MUNDO HELP DESK	
		Bienvenido, Cris Cartes Cerrar Sesión Ir al Inicio	
<input type="button" value="Aceptar"/>	<input type="button" value="Cancelar"/>		
Contraseña Anterior:	<input type="text"/>		
Contraseña Nueva:	<input type="text"/>		
Reescribir Contraseña Nueva:	<input type="text"/>		

TECHNOLOGY BRIEF: INCIDENT MANAGEMENT

Incident Management: A CA Service

Management Process Map

Copyright © 2007 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. This document is for your informational purposes only. To the extent permitted by applicable law, CA provides this document "As Is" without warranty of any kind, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised of such damages. ITIL® is a Registered Trademark and a Registered Community Trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Table of Contents

Executive Summary

SECTION 1: CHALLENGE 2

Simplifying ITIL

How to Use the CA Service Management ProcessMaps

SECTION 2: OPPORTUNITY 4

Incident Management

Raise Incident

Detect Event

Record

The Check Junction

Investigate and Diagnose

Escalate

The Act Junction

Resolve

Optimizing the Incident Management Journey

Avoiding Potential Incident Management Issues

SECTION 3: BENEFITS 11

Benefits of Incident Management Best Practices

SECTION 4: CONCLUSIONS 11

ABOUT CA Back Cover

Executive Summary

Challenge

The Information Technology Infrastructure Library (ITIL®) process framework embodies critical guidance for IT organizations seeking to improve service quality and align more closely with business goals. But ITIL can be challenging to interpret. Many organizations are challenged to apply ITIL effectively to achieve business goals.

CA has developed a unique approach to representing the ITIL framework and its interdependent IT Service Management (ITSM) processes at a high level in the form of an easy-to-use subway map. This map is an ideal starting point for understanding and communicating about ITIL in support of successful program planning and implementation.

Opportunity

ITIL places great emphasis on the timely recording, classification, diagnosis, escalation and resolution of incidents. The objective of the Incident Management process is to return IT services to a normal service level, as defined in a Service Level Agreement (SLA), as quickly as possible and with minimum disruption to the business. Incident Management should also keep a record of incidents for reporting purposes, and integrate with other processes to drive continuous improvement.

Within Incident Management the Service Desk plays a key function, acting as the first line of support and actively routing incidents to specialists and subject matter experts (SMEs). Incident Management is also strongly linked to Problem Management, which is charged with determining the root causes of incidents.

Benefits

- Timely resolution of incidents resulting in reduced business impact
- Improved user satisfaction
- More efficient utilization of Service Desk and other staff
- Enhanced ability to measure and monitor IT performance relative to SLAs
- Better data to support executive decisions regarding service quality
- Improved ability to track incidents and service requests efficiently
- Proactive identification of process enhancements

Simplifying ITIL

The ITIL process framework embodies critical guidance for IT organizations seeking to improve service quality and align more closely with business goals.

But the ITIL best practice guidelines are complex and challenging to understand and interpret. Moreover, they are not designed to provide definitive advice about implementing ITSM processes. Many IT organizations consequently undertake an ITIL journey without a firm idea of their goals and the path to achieve those goals.

CA has developed a unique approach to charting the ITIL journey through a visual representation of the ITIL framework and its interdependent ITSM processes in the form of a subway map. This two-part map presents an easy-to-navigate, high-level view of the ITIL terrain. IT executives, strategists and implementers can use these ITSM process maps—along with the family of CA ITSM Process Map Technology Briefs that expand on them—as a common reference point for understanding and communicating about ITIL; as well as to support successful program planning and implementation.

How to Use the CA Service Management Process Maps

CA's Service Management Process Maps apply the analogy of subway system maps to illustrate how best to navigate a journey of continuous IT service improvement. Each map describes the relevant ITIL processes (tracks), the ITIL process activities (stations) that you'll need to navigate to achieve ITIL process goals (your destination), and the integration points (junctions) that you need to consider for process optimization.

CA has developed two maps, Service Support and Service Delivery, since most ITSM discussions focus on these two critical ITIL disciplines.

CA's ITIL Process Maps illustrate at a high level how best to navigate a journey of continuous IT service improvement. Each map describes the relevant ITIL processes and activities you'll need to work with to reach your goals.

The Service Support map (Figure A) represents a journey of improving day-to-day IT service support processes that lay the operational foundation needed upon which to build business value.

THE CA SERVICE SUPPORT SUBWAY MAP



FIGURE A Service Support Subway Map.

The Service Delivery journey (Figure B) is more transformational in nature and shows the processes that are needed to deliver quality IT services.

THE CA SERVICE DELIVERY SUBWAY MAP



FIGURE B Service Delivery Subway Map.

Both maps represent the major ITIL processes as the “stations” en route to an organizational process “destination” or goal. The ITIL process stations are served by “tracks,” which are positioned relative to one another to illustrate how they support the goal of continuous improvement. The ITIL continuous improvement cycle takes the form of a “circle” or “central line,” with each Plan-Do-

Check-Act (P-D-C-A) step as a process integration point or “junction” on the line. Junctions serve both as reference points when assessing process maturity, and as a means to consider the implications of implementing a process in isolation.

This paper is part of a series of Service Management Process Map white papers. Each paper explains how to navigate a particular ITIL process journey, reviewing each process activity that must be addressed in order to achieve process objectives. Along each journey careful attention is paid to how technology plays a critical role in both integrating ITIL processes and automating ITIL process activities.

Incident Management

ITIL places great emphasis on the timely recording, classification, diagnosis, escalation and resolution of incidents. The objective of the Incident Management process is to return IT services to a normal service level, as defined in an SLA, as quickly as possible and with minimum disruption to the business. Incident Management should also keep a record of incidents for reporting purposes, and integrate with other processes to drive continuous improvement. Within Incident Management the Service Desk plays a key function, acting as the first line of support and actively routing incidents to specialists and subject matter experts (SMEs). To be fully effective, the Service Desk has to work in unison with other supporting processes.

For example, if a number of incidents are recorded at the same time, the Service Desk analyst needs sufficient information to prioritize each incident. Technology can be a key contributing factor by ranking incidents according to business impact and urgency. Today, many tools enable the automatic recording of incidents within the Service Desk function, but most IT organizations nevertheless lack the ability to correlate incidents and associate them with business service levels.

Let’s review the Incident Management process journey (illustrated in Figure C), assessing each critical process activity (or station), and examining how technology can be applied to optimize each stage of the journey, ensuring arrival at the process terminus—the efficient restoration of IT services.

While many tools are available to automatically record incidents, most IT organizations nevertheless lack the ability to correlate incidents and associate them with business service levels.

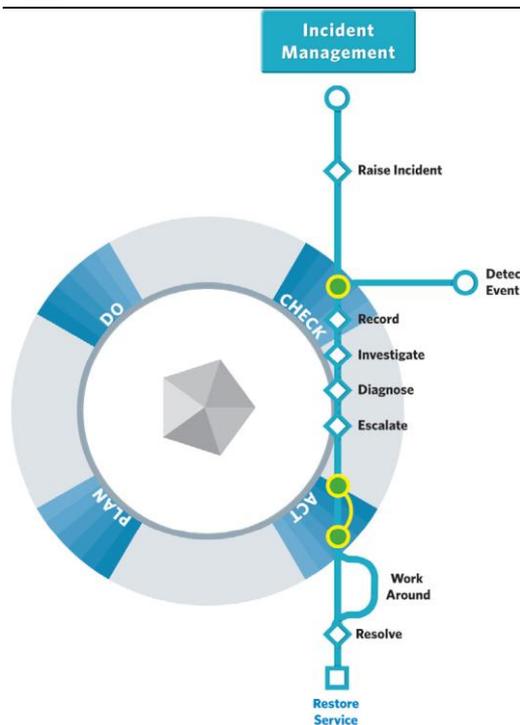


FIGURE C
THE INCIDENT MANAGEMENT PROCESS JOURNEY

Raise Incident

Incident Management starts with an Incident, which ITIL defines as an occurrence that is not part of the standard operation of a service and which causes, or may cause, an interruption or reduction in service quality. Incidents can include hardware and software errors; as well as user service requests, which are typically not associated with IT infrastructure failures. Examples of service requests include functional questions or requests for information, or a request to have a password reset.

Detect Event

An event is similar to an incident but is detected within the IT infrastructure rather than being reported by a user. The first activity along the Incident Management process journey is the mechanism to detect events as they occur within the IT infrastructure and result in deviations from normal service. Users of IT services are often the first to detect service deviations. However, with appropriate automation, IT can rapidly detect incidents before they adversely affect IT services as users experience them (that is, from the viewpoint of business processes). The goal is to detect events and solve the underlying problem(s) quickly, before they affect IT service levels or impact the business.

Track illustrates the process to restore services with minimal business disruption.

Record

In most cases incidents will be recorded by a Service Desk function, which should record all incidents to ensure that compliance with service level agreements can be reported correctly.

The nature of an incident will determine who or what reports it. Naturally, users should have a facility to incidents quickly and easily, supplying all information to the front line analyst. A truly effective reporting function, however, also should enable the system itself to automatically record incidents as they occur.

Many Service Desk solutions provide self-help and knowledge-based capabilities, but even if users resolve the issue themselves, they should record the incident. This is important, since the IT function can proactively leverage an accurate base of recorded incidents to facilitate improvements in other ITSM processes. Also, giving users the ability to log incidents that are not time-critical through a web-enabled interface, combined with a knowledge management tool to provide self-service answers to questions, greatly reduces the number of calls made to the Service Desk.

- Categorizing the incident should also be part of the Incident Management recording function.

Effective categorization of incidents has two aspects:

- Classification to determine incident type (e.g., IT Service=degraded)
- Matching against the IT Service or Configuration Items (CIs) affected (e.g. Order Entry Service), to determine if a similar incident has occurred previously.

During this phase, Service Desk analysts review previous incident activity to understand the reason for the incident. Technology can help by providing front line support with information pertaining to the CIs supporting the user who recorded the incident. The analyst should also have the means to correctly classify the incident using appropriate, standardized coding criteria. Many organizations mistakenly combine the IT Service/CI into the incident type. As a result, their incident classification methodology becomes far too complicated and a high percentage of incidents are incorrectly classified.

After classification, it is important to properly prioritize the incident. Service Desk solutions can help by automatically determining the priority based on the type of the incident (e.g. IT Service=Outage), and the business services that are affected. You can also use current SLAs to help determine priority. After classification, the analyst should use incident matching to see whether a similar incident has occurred previously; and whether there is a solution, workaround or known error. If there is, then the Service Desk function can bypass the investigation and diagnosis stages, and initiate resolution and recovery procedures.

If the incident has high priority and cannot be resolved immediately, the incident manager should create a linked problem record and initiate Problem Management process activities. It is interesting to note that Problem Management has a different focus from Incident Management, which can potentially result in conflicts. That is, Incident Management is charged with restoring the IT service, while Problem Management is charged with determining a root cause and updating the status of the incident to a known error. In the majority of cases where there is a conflict, the Incident

Management should take priority, since it is often more critical to restore normal service levels, even with workarounds, than to ferret out root causes.

The Check Junction

Before continuing along our Incident Management process journey, it is worth considering how the effective detection, recording and classification of incidents (achieved thus far) can facilitate an optimum journey along other ITIL process lines. Figure D shows that, after the detection and recording activities, the Incident Management process arrives at a critical point—the Check junction.

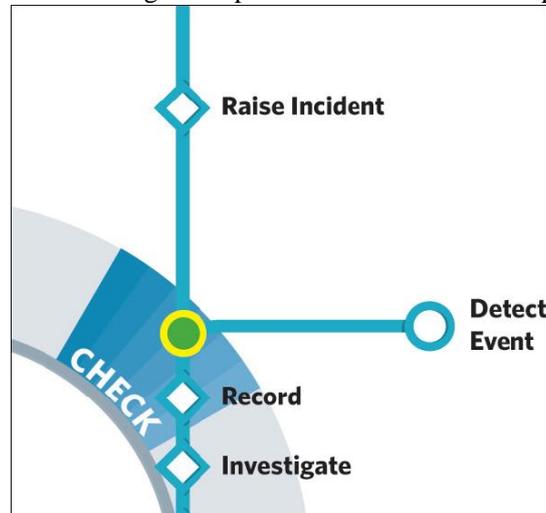


FIGURE D
Check Junction

The Check junction illustrates how Incident Management outputs derived from the timely detection and accurate reporting of incidents provide the means to be more proactive and optimize the Problem Management process. For example, the accurate recording of all incidents will assist Problem Management with the rapid identification of underlying errors. Where justified, Problem Management will strive to permanently correct these errors, and reduce the amount of repeat incidents. Alternatively, the Check junction enables Incident Management to take inputs from Problem Management to further streamline the overall process. For example, by delivering information about known errors (from an integrated known error database) the “journey time” to the ultimate destination—service restoration—will be reduced dramatically. Naturally, technology can play a key role, integrating both Incident and Problem Management within a single solution.

Investigate and Diagnose

If no immediate solutions are available, then the Service Desk function needs to be able to route incidents to SMEs. During the investigation and diagnosis phase, support analysts will collect updated incident details and analyze all related information (especially configuration details from a Configuration Management Database (CMDB) linked to the Service Desk). During this phase, the support staff must have access to comprehensive historical incident, problem and knowledge data, centralized and maintained within the Service Desk. Also critical is the capability to augment incident management records with diagnostic data supplied by SMEs or via integrated management technologies. Management technologies can play a key role here in correctly identifying and routing incidents to the appropriate SMEs.

By its very nature, investigation and diagnosis of incidents is an iterative process, and may involve multiple Level 1, 2 and 3 SME groups as well as external vendors. This demands discipline and a rigorous approach to maintaining records, actions, workarounds and corresponding results. Integrated Service Desk technology can help in this process by providing:

- Flexible routing of Incident Management data according to geographic region, time, etc.
- Automatic linkage and extraction of CMDB data for the examination of failed items
- A strong knowledge base and tools to expedite the diagnostic function
- Management dashboards and reports to provide an overall status of Incident Management

- Controls to ensure process conformance and provide comprehensive audit logs

Escalate

Having conducted investigation and diagnosis, the Incident Management journey arrives at the Escalation station. Critical here is the ability to rapidly escalate incidents according to agreed service levels, and to allocate more support services if necessary. Escalation can follow two paths; horizontal (functional) or vertical. Horizontal escalation is needed when the incident needs to be escalated to different SME groups better able to perform the Incident Management function. If not closely monitored, horizontal escalation can lead to incidents bouncing around the system without anyone taking ownership; which, in turn, increases the likelihood of breaching service level agreements. This is why it is so important to have a proactive approach and use process automation to correctly route incidents to the appropriate SME groups. Vertical escalation is where the incident needs to gain higher levels of priority. As part of the activity, it is essential that rules are clearly in place to ensure timely escalation, and avoid the need for support analysts to work out when to escalate—a recipe for disaster. For every resolution attempt, accurate data must be attached to the incident detail to save repeating recovery procedures, which lengthens overall resolution times. Here technology can play yet another key role, by automating the escalation process itself and pinpointing the exact source of errors. This latter capability is important because it ensures the correct incident hand-off to appropriate SME groups early in the support cycle.

The Act Junction

At this stage of the journey, the Incident Management process line has arrived at the Act junction (see Figure E). Here, iterative investigation and diagnosis will have determined the nature of the incident, and what actions need to be initiated to resolve the problem. Customer service must be restored as quickly as possible (through workarounds if necessary), and incidents should be escalated to Problem Management to detect the underlying cause of the problem, provide resolutions and prevent incidents from reoccurring.

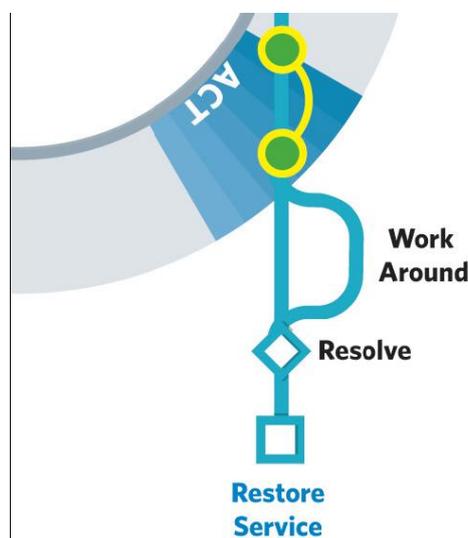


FIGURE E Act Junction

Resolve

The final stage along the Incident Management journey is Resolution and Recovery. Here the main activities include resolving the incident with solutions or workarounds obtained from previous activities. For some solutions, a Request for Change (RFC) will need to be submitted, so it is vital that automation supports the timely and accurate transference of incident details to a Change Management process. Once the SME groups resolve the solution, the incident is routed back to the Service Desk function, which confirms with the initiator of the incident that the error has been rectified and that the incident can be closed. During this phase, integrated processes must support a

number of service improvement functions, such as providing restricted access to the incident closing function, and ensuring that incidents are matched to known errors or problem records.

Optimizing the Incident Management Journey

Since a primary role of the Incident Management process is to ensure that users can get back to work as quickly as possible, activities should incorporate technologies that support the functions of recording, classification, routing to specialists, monitoring and resolution. Tools that help enhance the Incident Management process should at a minimum provide:

- Features to automate the detection, recording, tracking and monitoring of incidents
- Capabilities to ensure the integration of an accurate CMDB that will help estimate the impact of incidents according to business priority. Integrated CMDB information also ensures the support analyst has access to accurate information during critical diagnosis and investigation phases of the Incident Management process
- A comprehensive Knowledge Base (available to both users and support analysts) detailing how to recognize incidents, together with what solutions and workarounds are available
- Strong workflow capability to streamline escalation procedures and ensure timely incident hand-offs between various support groups
- Tight integration and proactive controls between supporting processes; for example, automatic logging of incidents during unapproved changes to configuration records
- The ability to accumulate incident data for decision-making purposes, such as:
 - Total number of incidents
 - Average incident resolution time (by Customer and Priority)
 - Incidents resolved with agreed Service Levels (by Customer and Priority)
 - Incidents resolved by front line support or through access to the Knowledge Base (with escalation and routing to subject matter experts)
 - Breakdown of incidents by classification, department, business service, etc.
 - Number of incidents resolved by analyst group/individual analyst/SME group, etc.

Avoiding Potential Incident Management Issues

The following are some common issues to be conscious of, in order to avoid problems in the Incident Management process:

BYPASSING INCIDENT MANAGEMENT If users attempt to resolve incidents themselves, IT cannot gauge service levels and the number of errors. Technology can help by centralizing the Service Desk function — essentially acting as the clearinghouse for all incidents, and integrating Incident Management within a broader Incident, Problem, Change and Configuration Management process. Incident Management bypass can also happen when users informally approach the SME groups for help. From a process perspective, however, the SME group should not take on the work until the Service Desk function has logged the incident.

HOLDING ON TO INCIDENTS Some organizations mistakenly fuse Information Management and Problem Management into a hybrid Incident Management process. This is detrimental from the perspective of metrics and the ability to prioritize the problems properly. There should be a clear separation between the two processes, and incidents should be closed once the customer confirms that the error condition has gone away. Based on business rules, the analyst can make the decision as to whether a related problem record should be created to look for a permanent solution.

TRAFFIC OVERLOAD This occurs when there are an unexpected number of incidents. This may result in the incorrect recording of incidents, leading to lengthier resolution times and degradation of overall service. Technology can help by automating procedures to deploy spare capacity and resources.

TOO MANY CHOICES There is the temptation to classify incidents in detail and make the analyst navigate through many sub-levels to select the incident type. This increases the time it takes to create

the incident, and will often lead to the incorrect classification because the analyst gives up searching for the most correct match.

LACK OF A SERVICE CATALOG If IT services are not clearly defined, it becomes difficult to refuse to provide help. A Service Catalog can help by clearly defining IT services, the configuration components that support the service, together with agreed service levels.

Benefits of Incident Management Best Practices

The benefits of implementing an Incident Management process in line with ITIL best practices include:

- Timely resolution of incidents resulting in reduced business impact
- Improved user satisfaction
- More efficient utilization of Service Desk and other staff
- Enhanced ability to measure and monitor IT performance relative to SLAs
- Better data to support executive decisions regarding service quality
- Improved ability to track incidents and service requests efficiently
- Proactive identification of process enhancements

The objective of Incident Management is to rapidly restore services in support of service level agreements. Unlike Problem Management, whose focus is on finding the root cause of problems, Incident Management is essentially about getting things back up and running quickly, even if this means performing workarounds and quick fixes.

Technology can play a critical role in optimizing the Incident Management process, by automating the actual process activities themselves (such as incident recording and classification), and by accessing the outputs from other related processes. Integration with other processes (especially Problem Management, Change Management, Configuration Management and Service Level Management) is vitally important to ensure that incidents are kept to a minimum and that the highest levels of availability and service are maintained.

To learn more about the CA ITIL solutions, visit ca.com/itil.

CA, one of the world's largest information technology (IT) management software companies, unifies and simplifies complex IT management across the enterprise for greater business results. With our Enterprise IT Management vision, solutions and expertise, we help customers effectively govern, manage and secure IT.

Learn more about how CA can help you transform your business at ca.com

BIOGRAFÍA

DATOS PERSONALES

Nombre: Galo Esteban Solís Gallo

Lugar y Fecha de Nacimiento: Quito, 28 de Mayo de 1984

EDUCACIÓN

Primaria: Colegio Ecuatoriano Suizo.

Secundaria: Colegio San Gabriel.

Superior: Escuela Politécnica del Ejército.

TÍTULOS

Bachiller en Ciencias, especialización Físico Matemático (2002)

Suficiencia en el idioma extranjero inglés (2005)

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADA POR

Galo Esteban Solís Gallo

COORDINADOR DE LA CARRERA

Ing. Ramiro Delgado

Lugar y fecha: Sangolquí, __ de diciembre de 2008