UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Sistema de Incidencias MSIS de Mega Soft Computación C.A. basados en Help Desk y sistemas de ticketing (Issue tracking system o trouble ticketing).

TRABAJO ESPECIAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR Pablo Lugo, Roderick Rangel.

PROFESOR GUIA Antonio Castro.

FECHA Septiembre, 2007

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Sistema de Incidencias MSIS de Mega Soft Computación C.A. basados en Help Desk y sistemas de ticketing (Issue tracking system o trouble ticketing).

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: DIECISIETE (17) PUNTOS

J U R A D O E X A M I N A D O R

Firma:	Firma:	Firma:
Nombre: ANTONIO CASTRO G.	Nombre: Juan Sanchez	Nombre: ALFREDO GARCIA R.

REALIZADO POR	Pablo Lugo, Roderick Rangel.
PROFESOR GUIA	Antonio Castro.
FECHA	Septiembre, 2007

Agradecimientos

A Dios ante todo por darnos la oportunidad de poder estar vivos, con salud y fe en que todas nuestras metas son posibles, siempre que tengamos fe y esperanza.

A nuestros padres y hermanos, quienes nos dieron la vida y la grandiosa posibilidad de poder estudiar esta carrera. Gracias por ser nuestro apoyo durante toda la carrera y brindarnos su confianza para lograr nuestras metas.

A Mega Soft Computación C.A. por brindarnos las herramientas, conocimientos y ambiente laboral necesario para lograr el desarrollo de este trabajo. Específicamente a Antonio Castro (Tutor) y César Celis (Coordinador del departamento de Monitoreo), quienes fungieron como pilares y apoyos incondicionales durante el desarrollo del sistema.

A mis amigos Héctor Rodríguez, Humberto Moreno, Carlos Tolosa, Yasnair Angulo, Marisol Aguiar y Anaid Villamizar, quienes durante toda la carrera sirvieron de apoyo como compañeros en nuestro proceso de aprendizaje.

Lugo M. Pablo J.

Rangel L. Roderick A.

Tabla de Contenido

Resumen 4

Introducción 5

CAPITULO I 6

Planteamiento del Problema 6

Objetivo General 8

Objetivos específicos 8

Justificación 9

Limitaciones y Alcances 12

CAPITULO II 14

Marco Referencial 14

1. *Foundation2* 14
2. *Merchant Server (Última versión)* 17
3. *Monitor Merchant Server (MMS)* 19
4. *SMS* 20
5. *Help Desk* 21
6. *DSS Decision support system* 21
7. *Issue Tracking System* 22
8. *MSC - Scheduler* 22
9. *Eclipse* 23
10. *SQL Server 2000* 23
11. *Estructuras Cliente – Servidor* 24
12. *RMI* 25

CAPITULO III 27

Marco Metodológico 27

CAPITULO IV 33

Desarrollo 33

1. *Fase de Inicio (Levantamiento de Información)* 33
   - 1.1. *Actividades realizadas* 33
     - 1.1.1. Análisis de Requerimiento 33
     - 1.1.2. Estudio del Sistema existente 34
     - 1.1.3. Estudio de las Estructuras asociadas 35
   - 1.2. *Definición de Requerimientos y Alcance del Sistema* 36
     - 1.2.1. Requerimientos Funcionales 36
     - 1.2.2. Requerimientos no Funcionales 37
   - 1.3. *Identificación de Actores y Casos de Uso* 39
     - 1.3.1. Actores del sistema 39
     - 1.3.2. Casos de Uso 39
   - 1.4. *Elaboración de Lista de Riesgos* 41

2. Fase de Elaboración (Diseño del Sistema) *42*
2.1. Actividades realizadas *43*
2.1.1. Diseño de la estructura lógica del sistema 43
2.1.2. Diseño del Prototipo 43
2.1.3. Diseño del Modelo de datos 43
2.1.4. Diseño del Modelo de clases 44
2.2. Arquitectura del Sistema MSIS *45*
2.2.1. Representación de la arquitectura 45
2.2.2. Restricciones y objetivos de la Arquitectura 47
2.2.3. Vista de Casos de Uso 48
2.2.4. Vista Lógica 48
2.2.5. Vista de Implantación o Realización 52
2.2.6. Vista de Despliegue 52
2.2.7. Vista de Datos 54
2.3. Diseño y modelado de la Base de Datos del sistema *55*
2.4. Plan de desarrollo del software *56*
2.4.1. Descripción del Proyecto 56
2.4.2. Alcance, propósito, objetivos y restricciones 57
2.4.3. Etapa del Front End 58
2.4.4. Etapa del Back End 58
3. Fase de Construcción (Desarrollo del Sistema) *59*
3.1.1. Etapa 1 desarrollo del Front End 60
3.1.2. Etapa 2 desarrollo del Back End 60
4. Fase de transición (Implantación) *61*
4.1. Migración de datos *61*
4.2. Manual de instalación *62*
4.3. Producto *62*

CAPITULO V ***63***

Análisis de Resultados ***63***

Conclusión ***68***

Recomendaciones ***69***

Referencias bibliográficas ***70***

Documento Visión (Apéndice A) ***71***

Documento Especificaciones Suplementarias (Apéndice B) ***72***

Documento Plan de Desarrollo (Apéndice C) ***73***

Documento Plan de Acción (Apéndice D) ***74***

Documento Lista de Riesgos (Apéndice E) ***75***

Documento Modelo de Casos de Uso (Apéndice F) ***76***

Documento Especificaciones de Casos de Uso (Apéndice G) ***77***

Documento Modelo Lógico Relacional (Apéndice H)..... 78
Documento Modelo de Implementación (Apéndice I)..... 79
Documento Manual de Instalación (Apéndice J)..... 80

Resumen

El Sistema Mega Soft Issue System (**MSIS**) es una herramienta de gestión de Tickets centralizada, que controla, reestructura y mejora la resolución y flujo de eventos, esto es fundamental para la organización, ya que ésta se desenvuelve en un mercado delicado y sensible como lo es el de pagos electrónicos.

La herramienta además permitirá, el envío de mensajes o alertas por diversas vías (SMS, Mail y Mensajes directos a ordenadores), además de gestionar de manera automática toda una serie de Incidencias (tipo de Ticket), relacionadas de manera exclusiva con el **Monitor Merchant Server** (**MMS**) y el monitoreo de los **Merchant Server's** (**MS**).

La metodología empleada para la realización del **MSIS** fue **RUP**, esta metodología permitió trabajar en fases e iteraciones para el desarrollo efectivo del software, permitió generar toda una serie de artefactos especializados los cuales definen las funcionalidades del sistema; así como también su estructura, arquitectura, sistemas relacionados y casos de uso, entre otra serie de vistas. Cabe destacar que los artefactos desarrollados forman una configuración ideal para el desarrollo de proyectos pequeños.

El **MSIS** promete facilitar las actividades de los departamentos de Soporte, Monitoreo e Implementación de la organización, para el caso especifico de Monitoreo permitirá realizar la integración con el **MMS** completando así las herramientas necesarias para el óptimo desempeño del departamento.

Introducción

El propósito del presente TEG es explicar el desarrollo del sistema **Mega Soft Issue System** (de ahora en adelante **MSIS**) el cual es la solución para la problemática de la empresa Mega Soft computación CA. El sistema **MSIS** realiza el manejo de los eventos suscitados dentro de la organización de una manera fácil y eficiente, esto basándose en los Issue Tracking System o Trouble Ticket System (en español sistemas de gestión de tickets).

El **MSIS** se auto-describe como una herramienta centralizada, la cual permite controlar el flujo y resolución de los diversos tickets reportados por los usuarios del sistema. Este sistema está a la disposición de los departamentos de Monitoreo, Soporte e Implementación de la organización.

Además de permitir el control y administración de los tickets, cumple con funcionalidades alternas como: generación de reportes sobre el comportamiento de diversos elementos del sistema, asociación de los acuerdos de niveles de servicio (SLA) a los tickets, generación automática de Incidencias (tipo de Ticket), envio de alertas o mensajes por diversas vías a los usuarios del sistema.

Como propósito final, la herramienta permite reestructurar y mejorar los procesos de gestión de eventos, razón fundamental para la organización que se desenvuelve en un mercado tan delicado y sensible como es el de pagos electrónicos, con esto se garantiza mejorar los tiempos de respuesta a los problemas dando una mejor imagen a los clientes.

CAPITULO I

Planteamiento del Problema

La empresa Mega Soft Computación C.A. debido a su crecimiento en los últimos años a nivel comercial, se ha visto obligada a invertir en el desarrollo de nuevas formas de optimizar sus procesos de negocio, para así lograr mantener un control de todas sus actividades internas. Uno de sus problemas es el hecho de no poder controlar todos los cambios o incidencias que suceden dentro y fuera de la organización, debido a que deben atender un gran número de llamadas, alertas o fallos de los sistemas -entre otros problemas- durante las 24 horas del día, lo cual no permite llevar un control de todas las actividades realizadas, es decir no tienen manera de llevar un historial de los sucesos del día a día.

Luego de un breve análisis de los requerimientos y necesidades relacionadas con esta situación, se logro obtener el siguiente listado de conclusiones:

- Ausencia de control sobre quién realizó o atendió una incidencia.
- No existe forma alguna de establecer medidas de rendimiento sobre el personal encargado de atender, generar y seguir las incidencias.
- Ausencia de reportes que sirvan como indicadores para los clientes, con la intención de demostrar los esfuerzos realizados por el personal de atención de los eventos. Esta situación genera desconcierto en los clientes y dudas sobre la eficiencia de la organización.
- Descontrol sobre el estado de las incidencias o eventos. No existe forma de saber si un evento se encuentra sin atención y en el caso exclusivo de las incidencias, es necesario saber si se encuentra abierta, cerrada, en seguimiento, etc.

- Dificultad para saber cómo atacar ciertos eventos que ocurren de manera seguida o aislada, por la ausencia de un manual que tenga un procedimiento estandarizado para la resolución de dichos eventos.
- Dificultad al momento de mantener un control sobre qué componentes (Hardware y Software) tiene un determinado cliente de la empresa, razón por la cual es difícil definir cuando un reclamo le atañe a la compañía.
- No existe una agenda que permita establecer aquellos mensajes o asignaciones para el cumplimiento de tareas de mantenimiento o resolución de eventos.
- Ausencia de reportes gerenciales que permitan mantener informados a los coordinadores/gerentes sobre las ocurrencias e incidencias de la empresa.
- La creación de los tickets es únicamente manual, independientemente de los tipos (Incidencias, Administrativos, Control de Cambio, etc.), lo que ocasiona desorganización y la pérdida de los estándares que cada evento implica: a qué tipo de ticket corresponde, cuál es la jerarquía asignada y los niveles de riesgo implicados, etc.

A consecuencia de estos problemas, la empresa se ve en la imperante necesidad de desarrollar algún sistema y poder resolver la situación planteada. Es precisamente en estos casos cuando hablamos de sistemas como Help Desk, específicamente de TTS (Trouble Ticket System) o también llamados en algunos casos Issues Tracking Systems, que permiten establecer las directrices a seguir para solventar las deficiencias encontradas en el análisis previo.

Objetivo General

Diseñar, desarrollar e implantar un Sistema encargado del control de los eventos dentro de la empresa Mega Soft Computación C.A. (de nombre **Mega Soft Issue System**), basado en el funcionamiento de los Help Desk y sistemas de ticketing, específicamente Issue tracking system, el cual permita controlar y gestionar todos los eventos suscitados en la organización.

Objetivos específicos

- Desarrollar un Sistema de Ticket de nombre Mega Soft Issue System (**MSIS**) basado en los TTS (Trouble Ticket System) o también llamado en algunos casos Issue tracking system, el cual permita poder listar de acuerdo a diversas pautas o Services Levels Agreements (SLA) las incidencias que ocurren en la organización.
- Generar reportes gerenciales los cuales sirvan como indicadores de desempeño tanto del sistema como de los usuarios del mismo.
- Realizar todo un levantamiento de datos para determinar toda la funcionalidad del sistema, componentes que utilizará, etc. Además de determinar y localizar las fuentes informativas necesarias para el funcionamiento del **MSIS**.
- Realizar un Componente generador de alarmas, el cual sea capaz de enviar dichas alertas por diversos medios, como SMS, Mail's, entre otras posibles vías. Rigiéndose por niveles jerárquicos o prioridades de las incidencias y los SLA de la organización con sus clientes.
- Integrar los Services Levels Agreements (SLA) de los clientes al **MSIS**, mediante un módulo, para poder disparar las alarmas necesarias en caso de que exista la

violación del un SLA. Estos deberán ser configurables durante el tiempo y por cliente.

- Integrar al Sistema **MSIS** con sistemas como el **MMS** (Sistema de Monitoreo Merchant) para evitar la redundancia de datos y utilizarlo como posible fuente de información al sistema.

Justificación

Se plantea el desarrollo de un Sistema de Ticketing nuevo, comúnmente llamado en la empresa como Sistema de Incidencias, basado en las directrices de los Help Desk, además de incluir una nueva serie de módulos o capacidades fuera del estándar de éstos, garantizando así que el sistema esté adaptado a las necesidades de la organización.

Este nuevo sistema trabajará bajo las siguientes especificaciones:

- Estaría totalmente desarrollado e implementado en Foundation 2, framework propio de la Organización Mega Soft Computación C.A. (basado en J2EE, pero con muchas mejoras las cuales facilitan el diseño e implantación de aplicaciones Web).
- El diseño de la solución estará basado en MVC versión 2, patrón de arquitectura en que se basa el framework antes mencionado.
- Se planea generar una serie de tareas automatizadas dentro de un planificador de actividades (Scheduler) de la organización llamado MSC-Scheduler, el cual nos permitirá realizar la automatización de muchas acciones o actividades del Sistema de Ticketing **MSIS**. Dicho planificador está desarrollado en Java y es parametrizable por medio de archivos (XML).
- Existirá una tarea en el MSC-Scheduler que fungirá como capa intermedia o fachada (Middleware), encargada de realizar la búsqueda de los datos necesarios

para la generación parcial de los Tickets (incidencias, administrativos, controles de cambio, etc.). Dicha fachada se apoyará en sistemas como el **MMS** y en consultas a bases de datos para poder preformatear los tickets por generar o para realizar funciones de avisos por diversos medios.

- Se utilizarán características de los DSS (Sistema de Apoyo de Decisión) para poder generar reportes que sirvan de apoyo a la toma de decisiones a nivel de rendimiento del personal o de eventos suscitados en clientes, por medio de históricos o similares.
- Se utilizarán las características de los sistemas, como agendas electrónicas o cronogramas de actividades, para el diseño de un módulo que permita generar asignaciones propias y a terceros.
- Se planea utilizar tecnología relacionada con los SMS para disparar alarmas o alertas a los dispositivos móviles del personal de la empresa, utilizando los diversos protocolos relacionados con este servicio de mensajes cortos como (GSM, 3G, e incluso el propio TCP/IP).
- Existirán posiblemente almacenes de datos los cuales servirán para completar los datos recolectados por la fachada o Middleware mencionado anteriormente, mediante una estructura de datos para cubrir los requerimientos de información necesaria para el completo desarrollo del MSIS.
- Se utilizarán las características más representativas de los TTS dentro de los Help Desk, en primer lugar para realizar los listados y los SLA de manera organizada según reglas y niveles jerárquicos que serán establecidos a partir de las necesidades del cliente y además para facilitar la atención por medio de sistemas automatizados de generación de incidencias y call centers. Dentro de las características más representativas están:
    - Online case capture vía Web and email (Captura en línea).
    - Auto-response email (Respuestas Automáticas).
    - Automatic case assignment (Asignación Automática de casos).
    - Case management (Administración de casos).

- Alerts and escalation (Escalabilidad y Alertas SLA).
- Asset Management (Control de Activos).
- Service metrics and analytics (SLA).

Gracias a estas características y beneficios que poseerá el sistema **MSIS**, podemos aseverar que éste apoyará por completo la gestión adecuada y eficiente de los eventos de la organización, además de mantener informados a los gerentes y coordinadores sobre los sucesos por medio de resúmenes y reportes que podrán fungir como herramientas para la toma de decisiones sobre los entes antes mencionados, facilitando la solución de los problemas en forma macro.

Limitaciones y Alcances

Las limitaciones del sistema MSIS son las siguientes:

- Será implantado únicamente para el departamento de Monitoreo de la empresa Mega Soft Computación C.A., con motivos de facilitar su implantación y pruebas, pero está estipulado que a largo plazo deberá ser utilizado por toda la organización.
- Al sólo estar implantado dentro del departamento de Monitoreo, sólo podrá atender Incidencias relacionadas con dicho departamento, esto nos limita a productos como Merchant y otros pocos clientes los cuales pueden ser monitoreados.
- Solamente estará disponible para los browser de Firefox y Explorer debido a que son el estándar de la organización. Se planea a largo plazo poder hacerla totalmente compatible con más browser.

Los alcances del sistema son los siguientes:

- Desarrollar dentro del Sistema de Ticket, tareas encargadas de generar plantillas de forma automatizada sobre los posibles fallos o estatus de las aplicaciones asociadas al MSIS, además de poder asignarle una prioridad y nivel, de acuerdo al tipo de incidencia, permitiendo atender funciones relacionadas a los call center.
- Desarrollar dentro de las funciones del Sistema de Ticket, la posibilidad de asignarle incidencias a diversos grupos de usuarios de acuerdo a los niveles y prioridades del sistema, para generar una distribución balanceada de las incidencias, en aquellos casos cuando un cliente esté generando muchas incidencias.
- Desarrollar módulos de definición de grupos, usuarios, proyectos, instancias de componentes del Merchant Server.

- Desarrollar un modulo de control de acceso y resumen de actividades de los usuarios del sistema.
- Realizar por medio de las funciones del Sistema de ticketing incidencias automatizadas encargadas de informarnos sobre qué software está dentro de los clientes del MSIS, además de informarnos sobre los cambios que se susciten dentro de estos.
- Desarrollar un módulo de agenda el cual permita establecer periodos de avisos sobre tareas que deberán ser cumplidas, cambios pendientes, establecimiento de incidencias operativas las cuales son recurrentes, y listas de chequeo para el óptimo monitoreo del desempeño de los usuarios del sistema. Además de tomar en cuenta las posibles alarmas que deberán ser disparadas en caso de no cumplirse estos avisos o tareas pendientes.
- Podrá ser visualizado y emitirá alarmas o avisos a dispositivos PDA, lo cual permite un alcance mayor y alto nivel de supervisión de parte de la alta gerencia.
- Deberá ser capaz de integrarse con otras aplicaciones de la organización, logrando así obtener la información necesaria de cada uno de los sistemas autónomos internos, para poder realizar su seguimiento y levantar incidencias relacionadas con estos.
- Será capaz de integrarse con diversos almacenes de datos de forma simultánea y de diversas compañías o tipos como ORACLE, SQL Server, entre otros, de manera transparente para el usuario, garantizando la portabilidad del sistema.

CAPITULO II

Marco Referencial

1. Foundation2

Framework para el desarrollo de aplicaciones Web propio de la organización. Éste está basado en J2EE y permite realizar las aplicaciones de forma rápida y efectiva. Foundation2 o la Fundación está basado en MVC V.2, dentro de esta arquitectura la Fundación o Foundation2 juega el papel de controller o controlador de los eventos, ya sea de parte del cliente (Browser) y del servidor encargado de la lógica del negocio.

Antes de profundizar en el framework es necesario explicar conceptos como:

a) MVC

*El **Model-View-Controller** (Modelo-Vista-Controlador, en adelante MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80.* [1] MVC es un patrón de arquitectura de software que separa los datos de una aplicación (Modelo), la interfaz de usuario (Vista), y la lógica de control (Controlador), en tres componentes distintos.

El patrón MVC en aplicaciones Web, es estructurado de la siguiente manera, la vista, es la página HTML y el código que provee de datos dinámicos a la misma, el controlador, es el Sistema de Gestión de Base de Datos (Sistemas encargados de la organización y modelado de datos que serán transportados de la vista al modelo y viceversa) y el modelo, es el modelo de datos o base de datos.

---

[1] *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. Ing. Febe Ángel Ciudad Ricardo http://www.informaticahabana.com/evento_virtual/files/MUL053.pdf.*

- ***Modelo (Model):*** *Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.*
- ***Vista (View):*** *Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.*
- ***Controlador (Controller):*** *Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests") para el modelo o la vista.*[2]

b) J2EE

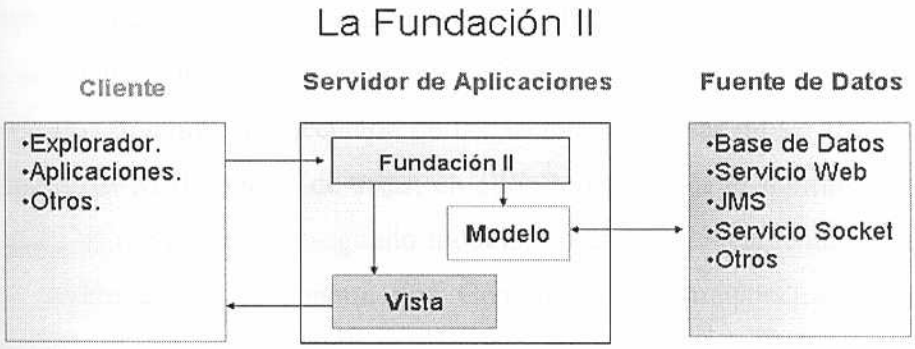
Java Platform, Enterprise Edition o Java EE (J2EE), es una plataforma de programación parte de la plataforma Java para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java. Es robusta, segura, portable y escalable, construida en las bases sólidas de la plataforma Standard Edition de Java (Java SE). La plataforma Java EE está definida por una especificación, similar a otras especificaciones del Java Community Process. [3]

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, entre otros, y define cómo coordinarlos y combinarlos. Java EE también configura algunas especificaciones de componentes únicos de él. Estas incluyen Enterprise JavaBeans, servlets, portlets, JavaServer Pages y varias tecnologías de servicios Web. Esto permite crear aplicaciones de empresa portables entre plataformas y escalables en el tiempo, además de integrables con tecnologías anteriores.[4]

---

[2] *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. Ing. Febe Ángel Ciudad Ricardo http://www.informaticahabana.com/evento_virtual/files/MUL053.pdf.*
[3] *Sun Developer Network http://java.sun.com/javaee/index.jsp*
[4] *Sun Developer Network http://java.sun.com/javaee/overview/whitepapers.*

En las aplicaciones Web convencionales el trabajo es realizado por medio de Servlet's, JSP, ASP, PHP, etc. En la Fundación gracias a la estructura MCV compuesta por el modelo, la vista y como controlador el framework, para ejercer el trabajo podemos proporcionar una base para el desarrollo rápido de aplicaciones Web.



- La Fundación II provee una base de desarrollo.
- Los Modelos plasman el Negocio y el Acceso a Datos.
- Las vistas no requieren Programación en el común de los casos.

Imagen 1, Foundation2 esquema funcional. [5]

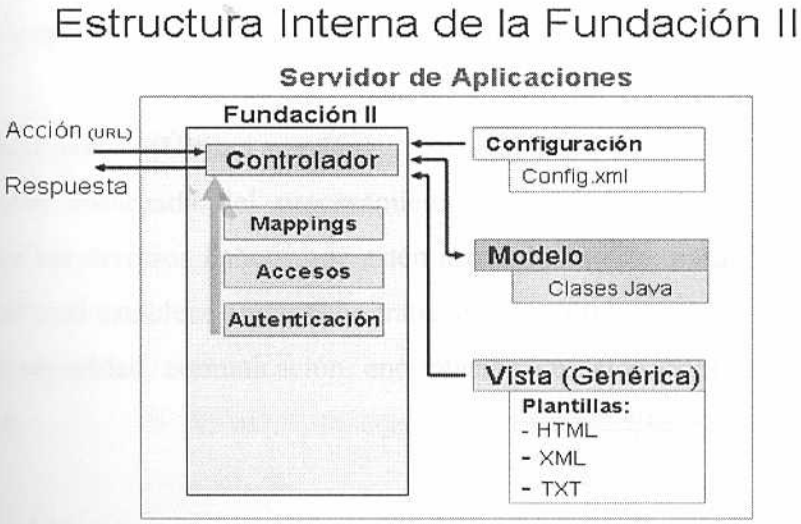


Imagen 2, Foundation2 estructura. [6]

[5] *Foundation2 introducción y funcionamiento.*
[6] *Foundation2 introducción y funcionamiento, propiedad de Mega Soft Computación CA.*

Su estructura interna se basa en:

- Un controlador, que está en capacidad de dirigir la acción a realizar por medio de un archivo XML de nombre "Action-map.xml", además controla los accesos a las acciones por medio de roles, localiza y organiza la ejecución de cada acción por medio de archivos de configuración XML, y realiza los mappings o mapeos para la sustitución dinámica de los datos en la vista final o template resultado.
- Los archivos llamados "Config.xml" (Propios de cada acción) que son los encargados organizar la ejecución de las acciones, además de ser los encargados de aplicar el MVC dentro de estas; en ellos se especifica el nombre del modelo (Model); el nombre de un segundo modelo, encargado únicamente de generar la vista dinámica (Por lo general uno Genérico del framework); el nombre del témplate que deberá personalizar el modelo de vista, entre otros parámetros los cuales permiten realizar la acción.
- Por último la Fundación se basa exclusivamente en un tipo de objeto genérico conocido como DataSet para el transporte de los datos de los modelos a las vistas, estos objetos cuentan con clases u estructuras de apoyo las cuales facilitan, los accesos a BD, la construcción de dichos objetos, etc.

2. Merchant Server (Última versión)

Aplicación encargada del procesamiento de transacciones financieras de un cliente hacia los diversos bancos que estén asociados a éste, basados en el estándar ISO 8583, el cual establece cómo debe tratarse todo lo relacionado con transacciones financieras, seguridad, comunicación, encriptación de datos, confidencialidad de los mismos, etc.

Estructura del Merchant Server (a nivel general)

- **Terminal**: Es la conexión que existe entre un Vterminal y un Payment específico, ésta es única y no pueden existir dos iguales dentro de un mismo Vterminal.

- **Vterminal**: Es la caja o post encargado de ejecutar las transacciones financieras, cada uno de estos mantiene, dentro de si, cierto número de Terminals, esto permite que las transacciones sean distribuidas de forma específica a su Payment correspondiente.
- **Payment**: son los diversos tipos o formas de pago que manejan los bancos (Acquirers), estas formas de pago deben estar asociadas a un Terminal, esto permite saber, con qué bancos se encuentra asociado un determinado cliente.
- **Acquirer**: son los bancos o adquirientes relacionados con cada cliente del software Merchant.
- **Merchant Client**: Programa o aplicativo de la organización que permite realizar las conexiones entre aquellas cajas o Vterminals que no pueden conectarse de forma directa contra el Merchant Server.
- **Merchant Server**: Es el aplicativo encargado de manejar, procesar, guardar todas aquellas transacciones ocurridas; además de llevarlas o ejecutarlas en su banco correspondiente. Sólo existe uno por cada cliente, es decir existe un sólo gran servidor el cual controla todas las transacciones ejecutadas en cada una de las sucursales de este cliente.

Funcionamiento (básico)

- Cada vez que se ejecuta una transacción por un Vterminal éste se lo comunica al Merchant Server por medio del Terminal y desde este punto, después de registrar en la base de datos todo lo relacionado con la transacción, es llevada hacia el banco correspondiente, para su ejecución.
- En el caso de, que no se pueda conectar contra un banco específico, está en capacidad de llevarlo a otro que se encuentre asociado al cliente.
- De no poder conectarse un Vterminal directamente con el aplicativo, se apoya en el Merchant Client para esta tarea.

3. Monitor Merchant Server (MMS)

Está basado en 3 componentes básicos, primero está el Cliente MMS, el cual se encuentra desarrollado bajo el MSC-Scheduler de la organización. Éste se encarga de ejecutar tareas de recolección de datos en los diferentes servidores Merchant que existen; de segundo está el Servidor MMS desarrollado también bajo el MSC-Scheduler, encargado de recibir estos datos emitidos por todos los clientes, organizarlos, depurarlos e insertarlos en un BD central de la organización; por último está el tercer componente que es un Monitor Web MMS desarrollado bajo Foundation2, este aplicativo se encarga de consultar en la BD antes mencionada para mostrar los datos actuales de todos los servidores Merchant y apoyar al área de monitoreo de la organización.

El **MMS** está compuesto de una serie de actividades o funcionalidades específicas entre las cuales tenemos:

- Recopilación de datos y envío al servidor central por medio de vías como FTP, Socket y Mail.
- Extracción de datos transaccionales para ser enviados al servidor y luego ser resumidos y transformados en volúmenes transaccionales por bloques de una hora.
- Transformación y depuración de los datos recopilados en el servidor central, para realizar reportes automáticos los cuales son enviados vía mail a los integrantes del departamento de monitoreo de la organización, entre otros involucrados.
- Realiza las diversas vistas y reportes acerca de los estados de los Merchant a nivel de la interfaz Web, resúmenes estadísticos, volúmenes transaccionales, y el monitoreo de los clientes Merchant para verificar su conectividad y buen funcionamiento en todos los sentidos.

Funcionamiento

El funcionamiento del **MMS** está segmentado de la siguiente manera:

- ***Etapa de recolección de datos***, realizada por las tareas que son ejecutadas por el Cliente MMS, esta recolección o recopilación de datos se basa en localizar y extraer los datos teóricos o que componen al sistema de los clientes y datos transaccionales los cuales son enviados cada horas y son migrados a la base de datos local del sistema (proceso de datamining).
- ***Etapa de transformación y elaboración de datos estadísticos***, se realiza en el Servidor MMS, justo en el momento en que se reciben los datos de cada cliente del sistema, son depurados y en su mayoría transformados a valores numéricos para luego con ellos poder elaboras reportes.
- La tercera y cuarta etapa están formadas por el Monitor Web MMS:
  - ***Tercera etapa generación de reportes estadísticos, volúmenes y alertas***, como caídas de clientes, fallos, transacciones ocurridas, reportes, entre otras, para esto se manipulan los datos almacenados para luego ser mostrados en reportes adaptados según las necesidades del personal.
  - ***Cuarta etapa seguimiento o monitoreo de los clientes***, se basa en revisar los estados, volúmenes y transacciones de las horas anteriores a la actual, así como también las posibles alarmas y fallas presentadas en tiempo real, esto por medio de acciones que permiten ver el estado actual de la plataforma Merchant en un cliente.

4. SMS

El servicio de mensajes cortos o SMS (Short Message Service) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad está disponible en una amplia

variedad de redes, incluyendo las redes 3G, existe un ente encargado de regular e impartir todos los estándares relacionados con los mensajes cortos llamado 3GPP, las especificaciones 23.040 y 24.011 son las encargadas de estandarizar el proceso de envío y como hacer que sean punto a punto.[7]

5. Help Desk

Los help desk son un conjunto de servicios, que de manera integral, bien sea a través de uno o varios puntos de contacto, ofrecen la posibilidad de gestionar y solucionar todas las posibles incidencias, a esto se le añade la atención de requerimientos TICS (Tecnologías de Información y Comunicaciones).[8] Como su nombre lo dice, es una Mesa de Ayuda, donde los responsables ofrecen Servicios acerca de soporte técnico ayudando a incrementar la productividad y satisfacción de los usuarios internos y externos. *En 1980 comenzaron a aparecer las computadoras personales en las corporaciones, hasta ese momento todo era manipulado por el departamento de IT (Tecnología de Información), esta aparición causó la necesidad de generar en estas organizaciones el departamento de soporte técnico, expertos no entrenados, encargados de solucionar problemas sobre las estaciones de trabajo, con la llegada de Internet en 1990 estos departamentos podían desligarse de las organizaciones y así poder prestar sus servicios a más empresas, es ahí donde comenzamos a hablar de los Help Desk.*[9]

6. DSS Decision support system

Los sistemas de soporte a la decisión, DSS (*Decision Support Systems*) por sus siglas en inglés, *son definidos por* ***Turban*** *como sistemas de información basados en computadoras los cuales combinan modelos y datos para intentar resolver problemas*

---

[7] *3GPP – Organización que mantiene las especificaciones http://www.3gpp.org/ftp/Specs/html-info/23040.htm, http://www.3gpp.org/ftp/Specs/html-info/24011.htm.*
[8] *Introduction to Help Desk Concepts and Skills by Susan Sanderson http://faculty.necc.mass.edu/eschuster/cis114_s2006/chapter_1_helpdesk.pdf*
[9] *Introduction to Help Desk Concepts and Skills by Susan Sanderson http://faculty.necc.mass.edu/eschuster/cis114_s2006/chapter_1_helpdesk.pdf*

no estructurados utilizando una interfaz amigable para el usuario. ***Bonckzek*** *los define como un sistema basado en computadora, que consiste en 3 componentes interactivos: sistema de lenguaje, sistema de conocimiento y sistema de procesamiento de problemas. Por último* ***Olson y Courtney*** *dicen que un sistema de soporte a la decisión es el uso de cómputo interactivo para aprender acerca de problemas de decisión, frecuentemente a través de accesos a datos y modelos.*[10]

7. Issue Tracking System

Son sistemas encargados de resolver eventos o incidencias (Issue's) ocurridas en tiempo real, ya sea a través de un call center, vía mail, entre otras posibles vías. La funcionalidad básica de estos sistemas es el poder generar soluciones a estos eventos gracias a un knowledge base. Estos sistemas permiten hacer el seguimiento constante de las incidencias, hasta que estas son solucionadas por completo.[11] Estos sistemas son capaces de organizar el flujo de trabajo y saber como asignar los asuntos según el estado actual, son capaces de reasignar y seguir las incidencias además de poder editarlas y utilizarlas como posibles respuestas a una situación común para diversos clientes, entre otras funcionalidades, de forma de facilitar el seguimiento y manejo de dichas solicitudes así como cualquier otra interacción con sus clientes o usuarios.

8. MSC - Scheduler

Planificador de tareas desarrollado por la organización, para ejecutar actividades que deban ser programadas en ciertos momentos del día, ya sea de manera simple o recursiva durante el tiempo.

---

[10] *Turban, Efraim, Aronson, Jay E., Liang Ting-Peng, Decision Support Systems and Intelligent Systems, Seventh Edition*

[11] *Copyright © 2003-2007 Edgewall Software. All rights reserved. http://trac.edgewall.org/ http://trac.edgewall.org/wiki/TracWiki. Copyright © 2002-2007 TrackStudio, Ltd http://www.trackstudio.com/*

Este planificador se maneja con un XML en el cual se deben explicar datos como clases que van a ser corridas, tiempos de ejecución, períodos de pausa y de repetición, entre otros datos.

9. Eclipse

Eclipse es una plataforma de software de Código abierto, este es independiente de una plataforma para desarrollar lo que comúnmente se llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, ha sido usada anteriormente para la creación de IDE's (Integrated Development Environment), interfaces para el desarrollo de aplicaciones, como el IDE de Java llamado Java Development Toolkit y el compilador que se embarca como parte de Eclipse y es usado en su propio desarrollo.[12] Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización sin fines de lucro que fomenta una comunidad de código abierto y un conjunto de productos para satisfacer las necesidades de organizaciones y desarrolladores.

10. SQL Server 2000

SQL Server 2000 es un sistema de gestión de bases de datos relacionales (SGDBR o RDBMS: Relational Database Management System) diseñado para trabajar con grandes cantidades de información y con la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios Web. Ofrece el soporte de información para las tradicionales aplicaciones Cliente/Servidor, las cuales están conformadas por una interfaz a través de la cual los clientes acceden a los datos por medio de una LAN.

---

[12] *Copyright © 2007 The Eclipse Foundation. All Rights Reserved*
http://wiki.eclipse.org/index.php/Eclipse_FAQs#Part_I__--_The_Eclipse_Ecosystem

Microsoft SQL Server es un sistema basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. [13]

Sus características más representativas podrían ser:

- Permite trabajar en modo cliente-servidor donde los datos se alojan en el servidor y las terminales o clientes de la red sólo pueden observar y realizar muy pocos cambios sobre estos.
- Soporta un gran volumen de transacciones de manera simultánea.
- Soporta procedimientos, funciones y triggers almacenados.
- Permite administrar información de otros servidores de datos
- Provee escalabilidad, estabilidad y seguridad.

11. Estructuras Cliente – Servidor

Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores. [14]

---

[13] *Erica Isabel Enriquez. Prof. Director: Mgter. David Luis la Red Martínez*
http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/SQL.pdf
[14] *Félix Murillo Alfaro Instituto Nacional de Estadística e Informática*
http://www.inei.gob.pe/biblioineipub/bancopub/inf/lib5038/defi.HTM

Esta arquitectura consiste básicamente en un programa "el Cliente" realiza peticiones a otro programa "el servidor" que les da respuesta y donde por lo general reside la mayor parte de la lógica del negocio de ese sistema. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora tiene mayor validez y efectividad en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Ventajas de la arquitectura cliente-servidor:

- **Centralización del control**: los recursos, accesos e integridad de los datos, es decir, los componentes que conforman la lógica o corazón del sistema, son controlados por "el servidor" de forma que un "cliente" defectuoso o no autorizado no pueda corromper los datos o la funcionalidad del todo.
- **Escalabilidad**: se puede aumentar la capacidad de clientes y servidores por separado.

12. RMI

RMI (Java Remote Method Invocation) es un mecanismo ofrecido en Java para invocar un método remotamente, es decir invocar a otra Virtual Machine para correr un método sobre ésta. Al ser RMI parte estándar del entorno de ejecución Java, usarlo provee un mecanismo simple en una aplicación distribuida que solamente necesita comunicar servidores codificados para Java. Si se requiere comunicarse con otras tecnologías debe usarse CORBA o SOAP en lugar de RMI.[15]

Al estar específicamente diseñado para Java, RMI puede darse el lujo de ser muy amigable para los programadores, permitiendo pasar objetos por referencia (cosa que no hace SOAP), "recolección de basura" distribuida en cada una de las Virtual

[15] *Sun Microsystems, Inc. http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp*

Machine's sobre las cuales trabaja, y pasaje de tipos arbitrarios (funcionalidad no provista por CORBA).

Por medio de RMI, un programa Java puede exportar un objeto, siempre y cuando éste sea serializable. A partir de esa operación este objeto está disponible en la red, esperando conexiones en un puerto TCP. Un cliente puede entonces conectarse e invocar los métodos que este objeto posea (respetando sus permisos claro está). La invocación consiste en el "marshaling" de los parámetros (utilizando la funcionalidad de "serialización" que provee Java), luego se sigue con la invocación del método (cosa que sucede en el servidor). Mientras esto sucede el cliente que realiza la petición está en espera bloqueado a que regrese una respuesta. Una vez que termina la ejecución el valor de retorno (si lo hay) es serializado y enviado al cliente. El código cliente recibe este valor como si la invocación hubiera sido local.[16]

[16] *Sun Microsystems, Inc. http://java.sun.com/docs/books/tutorial/rmi/index.html, http://java.sun.com/j2se/1.5.0/docs/guide/rmi/index.html, http://java.sun.com/j2se/1.5.0/docs/guide/rmi/faq.html,*

CAPITULO III

Marco Metodológico

Al momento de realizar un sistema es importante escoger la metodología, debido a que ésta será la herramienta que permitirá organizar el flujo de trabajo. El Sistema **MSIS** estará basado en una adaptación a RUP.

Rational Unified Process (RUP) es un proceso de desarrollo de software; éste constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. [17]

El RUP está basado en 6 principios claves:

- **Adaptación:** El proceso de desarrollo deberá estar adaptado a las necesidades de la organización. El tamaño del proyecto, su tipo o regulaciones de éste son suministradas por la empresa e influyen en su diseño.
- **Balance:** Los requerimientos propuestos por los diversos Stakeholders suelen ser diferentes, contradictorios y hasta tienden a disputar recursos limitados de la organización. Se debe encontrar un balance y categorizar o calificar estos requerimientos para luego medir su probabilidad de desarrollo.
- **Colaboración:** El desarrollo de software, procesos relacionados con el levantamiento de información y diseño, no debe ser realizado por una única persona sino por múltiples equipos. Para ello debe existir una comunicación fluida para la coordinación de los requerimientos, desarrollos, evaluaciones, planes, resultados, etc.

[17] *TEG de Mar Bru Soler, Universidad Politécnica de Valencia Diseño y Construcción de un sistema para la monitorización de campos electromagnéticos*

- **Iteraciones:** Los proyectos se entregan, aunque sea de un modo interno para evaluaciones de los Stakeholders, a estas entregas se les da el nombre de **iteraciones**. En cada iteración los Stakeholders junto con los grupos de desarrollo analizan, la estabilidad, calidad y facilidad de uso del producto, gracias a éstas, la refinación del proyecto es constante.
- **Abstracción:** Este principio se basa en el concepto de la reutilización de elementos como patrones de desarrollo de software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Esto facilita el trabajo de los ingenieros de software permitiéndoles ir directamente de los requisitos suministrados por los Stakeholders a la codificación de software. Un nivel alto de abstracción también permite discusiones que incentíven la participación de los Stakeholders en el proceso de diseño y desarrollo, esto facilita la generación de niveles arquitectónicos los cuales pueden estar acompañados por las representaciones visuales de la arquitectura, con ayuda de gráficos en UML.
- **Calidad:** El control de calidad debe realizarse en todas las fases e iteraciones.

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuyentes claves de RUP colaboró con Boehm en la investigación. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivan Jacobson. El primer resultado de esta fusión la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten. [18] El gráfico mostrado a continuación muestra la línea de tiempo de creación de la metodología RUP.

[18] *TEG de Mar Bru Soler, Universidad Politécnica de Valencia Diseño y Construcción de un sistema para la monitorización de campos electromagnéticos.*

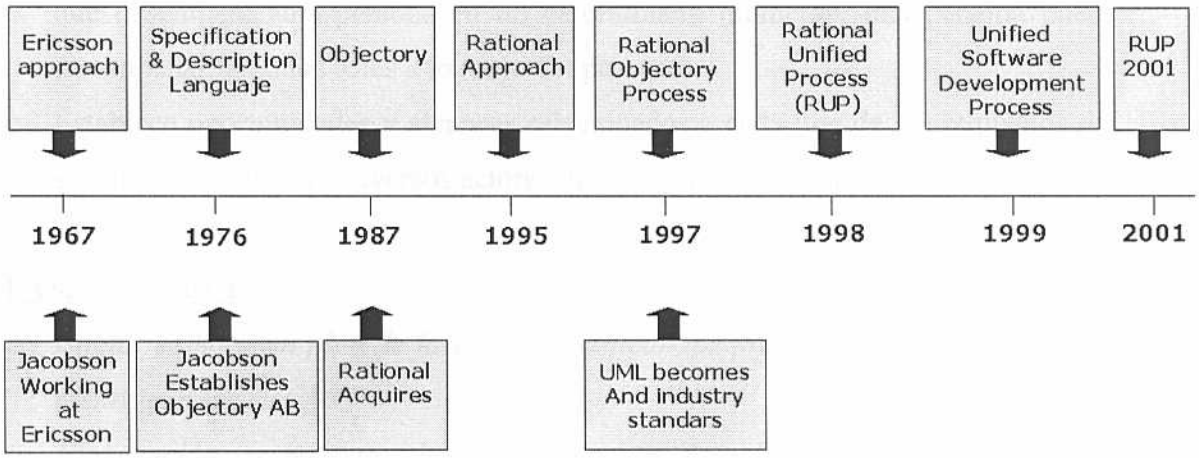


Tabla 1. Esquema histórico de etapas para la creación de RUP [19]

La metodología RUP destaca que para un desarrollo eficiente de un sistema se deben de aplicar las siguientes tres características esenciales: estar dirigido por los Casos de Uso, estar centrados en la arquitectura, y proponer un desarrollo iterativo e incremental, apoyado en una constante revisión.

Las características más representativas de esta metodología son:

- Impone organización o forma disciplinada de asignar tareas y responsabilidades.
- Plantea la utilización o implementación de las mejores prácticas de Ingeniería del Software las cuales son: desarrollo iterativo, administración de requisitos planteados, uso de arquitecturas basadas en componentes, manipulación de controles de cambios, y controles de versiones en el desarrollo, modelado visual del software y verificación constante de la calidad del software elaborado.
- Impone la creación y manipulación de artefactos (que son los productos tangibles del proceso de desarrollo como por ejemplo, el modelo de casos de uso, el código fuente, etc.), los cuales manejan iteraciones y revisiones internas, y roles (papel

[19] *Departamentos de sistemas informáticos y computación Universidad Politécnica de valencia https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc*

que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

- Establece oportunidades y alcances relacionados a cada uno de los requisitos del sistema: identifica los diversos actores del sistema; identifica los casos de uso.

FASES DEL RUP

- ***Inicio:*** *se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos*
- ***Elaboración****: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos*
- ***Construcción:*** *se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.*
- ***Transición:*** *se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.* [20]

Debido a que la metodología aplicada al sistema es una adaptación, utilizaremos los siguientes artefactos los cuales conforman una buena configuración para un proyecto de este tamaño (Pequeño).

Entregables del Sistema

A continuación se describen brevemente cada uno de los artefactos que se generarán y usarán durante el proyecto.

1. ***Documento Visión:*** *Este permite obtener una visión general de los requerimientos del proyecto, características clave y restricciones principales.*

[20] *TEG de Mar Bru Soler, Universidad Politécnica de Valencia Diseño y Construcción de un sistema para la monitorización de campos electromagnéticos.*

2. ***Especificaciones suplementarias:*** *Este documento contiene las especificaciones de los requerimientos no funcionales del sistema, además de aplicar filtros para determinar la efectividad, factibilidad, entre otros puntos, del sistema.*
3. ***Plan de Desarrollo:*** *Proporciona la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.*
4. ***Plan de Acción:*** *Refleja el cronograma estipulado para el desarrollo completo del sistema.*
5. ***Lista de Riesgos:*** *Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.*
6. ***Glosario:*** *Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.*
7. ***Modelo de Casos de Uso:*** *El modelo de Casos de Uso presenta la funcionalidad del sistema y los actores que hacen uso de ella. Se representa mediante Diagramas de Casos de Uso.*
8. ***Especificaciones de Casos de Uso:*** *Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no-funcionales asociados.*
9. ***Modelo Lógico Relacional:*** *Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Tablas donde se muestran las tablas, claves, etc.*
10. ***Modelo de Implementación:*** *Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.*

11. ***Manual de Instalación:*** *Este documento incluye las instrucciones para realizar la instalación del producto.*
12. ***Producto:*** *Todos los ficheros fuente y ejecutable del producto.*[21]

Justificación de la metodología empleada

- Con el uso de lo artefactos asociados garantizamos una amplia documentación en los niveles jerárquicos de la organización relacionados con el sistema.
- Permite mantener una constante revisión, evitando la pérdida o desvío de los requerimientos del sistema.
- El desarrollo de las actividades es paralelo, esto nos permite ir avanzando en muchos aspectos de sistema al mismo tiempo.
- El desarrollo en fases y flujos de trabajo, facilita la visualización global del proyecto.
- Permite mantener una organización de las asignaciones y responsabilidades, manteniendo informados constantemente a los Stakeholders de la evolución de la solución. Esto origina un ambiente de responsabilidad compartida dentro de los grupos de desarrollo.
- Aunque para el óptimo desarrollo de la metodología se recomienda que se desarrolle dentro de grandes grupo de trabajo, es fácilmente adaptable para el desarrollo de proyectos pequeños con poco personal.

[21] *Departamentos de sistemas informáticos y computación Universidad Politécnica de Valencia https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc*

CAPITULO IV

Desarrollo

El desarrollo del sistema **MSIS** fue concebido en una serie de etapas según la metodología de RUP, estas etapas son las siguientes:

- Inicio (Levantamiento de Información).
- Elaboración (Diseño del sistema).
- Construcción (Desarrollo del sistema).
- Transición (Implantación).

Ahora bien todas estas fases o etapas antes mencionadas poseen toda una serie de actividades internas las cuales estudiaremos en el resto del capítulo, para detallar el proceso de desarrollo del sistema.

1. Fase de Inicio (Levantamiento de Información)

En esta fase del desarrollo del sistema **MSIS**, se describen todos los procedimientos o actividades necesarias para recopilar la información necesaria (requerimientos, sistemas existentes y estructuras asociadas) con la idea de obtener visión general del ambiente o entorno en el cual se encuentra la organización, esta etapa o fase permite establecer las bases para idear el diseño de la solución.

1.1. Actividades realizadas

1.1.1. Análisis de Requerimiento

Esta actividad permite recabar las necesidades de la organización, sobre el manejo y gestión de los Tickets (Dentro de la organización se les da la notación de "Tareas"). Por medio de reuniones con los Stakeholders y entrevistas individuales con estos, se

obtuvo una vista general del problema planteado, y ésta se fraccionó a su vez, en un grupo de requerimientos los cuales indagan más en la problemática.

Este listado de requerimientos globales sirvió para profundizar en entrevistas y discusiones con los usuarios finales o personal operativo del aplicativo actual antes del desarrollo del **MSIS**, estas conversaciones permitieron generar directrices específicas sobre los requerimientos, permitiendo detallarlos y desmembrarlos en nuevos requerimientos funcionales más específicos, en conclusión este último proceso de conversaciones permitió detallar los resultados obtenidos en previas conversaciones con los Stakeholders.

1.1.2. Estudio del Sistema existente

Este estudio permitió analizar al sistema que empleaba la empresa antes del desarrollo del **MSIS**, que sólo estaba conformado por un aplicativo Web incapaz de manejar históricos sobre los tipos de tickets, ni históricos de los estados por los cuales un ticket había pasado durante su resolución, entre otras deficiencias que no permitían que fuese calificado como un **sistema de gestión de tickets**. Por otra parte el sistema sirvió de base para entender el funcionamiento de la generación de tickets, facilitando también la comprensión de las estructuras asociadas a esta generación.

Las estructuras más representativas para el desarrollo del sistema **MSIS,** basado en los sistemas de Gestión de Tickets (Ticketing System), que fueron extraídas de este análisis fueron las siguientes:

- Clientes
- Sucursales
- Instancias
- Proyectos
- Roles
- Usuarios

- Grupos
- Componentes

Cabe destacar que la administración de estas estructuras, dentro del sistema, es muy deficiente. Gracias a los Stakeholders y a los usuarios finales del sistema durante el proceso de entrevistas, y al proceso de análisis del mismo, se determinaron todas las fallas sobre el look and feel, funcionalidad y facilidad de uso, de estas estructuras, la falla más importante a nivel general que podemos extraer, es el hecho de no saber coordinar el flujo de los eventos de administración, esto causa molestias e incomprensión de estos módulos.

1.1.3. Estudio de las Estructuras asociadas

En esta actividad se estudió la Base de Datos del sistema antes de emprender el desarrollo del **MSIS**, el **MMS** (Monitor Merchant Server) y para una mejor comprensión de éste, el **MS** (Merchant Server). Esto con el fin de completar el levantamiento de información necesaria.

El sistema **MMS** fue analizado, con el fin de facilitar su entendimiento, además de sentar las bases para el proceso de integración de este sistema con el **MSIS**. El estudio del **MS** fue con el fin de comprender con un mayor nivel de detalle al **MMS** y los procesos que éste monitorea. Todos estos datos recabados permiten la definición de los tipos de Incidencias automatizadas que deberá generar el **MSIS** (entiéndase por Incidencias como un tipo de Ticket).

Acerca de la BD del sistema que la organización poseía, podemos resaltar el hecho de que contenía gran cantidad de des-normalizaciones en su construcción, además de una serie de detalles en las tablas y sus relaciones, estos detalles eran la causa de algunos de los problemas planteados dentro del aplicativo Web antes analizado.

Otro detalle de suma importancia en la fase de inicio, fue la completa ausencia de manuales, instructivos o documentos relacionados al antiguo sistema, lo cual originó problemas en el análisis.

1.2. Definición de Requerimientos y Alcance del Sistema

1.2.1. Requerimientos Funcionales

Los requerimientos funcionales, son aquellos requerimientos obtenidos después del análisis completo del problema planteado por la organización, en otras palabras son todas aquellas funcionalidades que el sistema debe cumplir, para satisfacer y mejorar la calidad de los afectados, en este caso específicamente los Stakeholders y los departamentos de **Monitoreo**, **Soporte** e **Implementación**.

Dentro de los requerimientos funcionales obtenidos en el proceso de análisis, el más representativo es el siguiente: Necesidad de controlar todos los Tickets suscitados dentro de los departamentos de **Monitoreo**, **Soporte** e **Implementación**, de manera integral, centralizando información para que los usuarios puedan solventar de manera rápida y efectiva estos eventos. Este requerimiento general puede ser subdividido en el siguiente listado de requerimientos funcionales.

- Administrar Tickets.
- Administrar Consultas personalizadas.
- Administrar Prioridades de Tickets.
- Administrar Tipos de Tickets.
- Administrar Títulos de Tickets.
- Administrar Estados de Tickets.
- Administrar Soluciones.

Estos, junto a otra serie de requerimientos funcionales, están especificados dentro del documento de **Especificaciones Suplementarias** (**Apéndice B**), documento que cumple la función de definir aquellos detalles que no están descritos en documentos

como **Modelos de Casos de Uso (Apéndice G)** y **Especificaciones de Casos de Uso (Apéndice H)**, complementando el proceso de determinación de requerimientos y formulación del sistema solución (**MSIS**). Por otra parte los requerimientos generales establecidos por los Stakeholders se encuentran descritos dentro del documento **Visión (Apéndice A)**.

1.2.2. Requerimientos no Funcionales

Los requerimientos no funcionales del sistema son aquellos aspectos visibles para los usuarios que no se encuentran relacionados de ninguna forma con el funcionamiento del sistema, dentro de estos requerimientos se podrían listar detalles como:

- Documentación relacionada al sistema, manuales, etc.
- Características o situaciones suscitadas en la ejecución como usuarios concurrentes, cargas de trabajo, volumen de datos manejados.
- Interfaces sencillas y amigables, que estén desarrolladas bajo los estándares establecidos.
- Usuarios capacitados, experiencia en el área y alto nivel de compresión del fin a alcanzar.

Para lograr cumplir de manera efectiva estos requerimientos se realizan una serie de preguntas a nivel general sobre el producto para medir su nivel de calidad.

Gráfico 1 Dimensiones de Calidad[22]

La funcionalidad, usabilidad, confiabilidad, desempeño y soportabilidad son los puntos a atacar para poder responder a estos requerimientos. Como bien muestra el gráfico anterior, estas pruebas deben realizarse sobre la solución con la finalidad de garantizar su eficiencia.

Las descripciones y análisis de cada uno de los puntos antes mencionados se encuentran detallados dentro del documento **Especificaciones Suplementarias** (**Apéndice B**). Cabe destacar que estas descripciones fueron desarrolladas después de implantado el sistema, es decir, ya realizadas todas las pruebas de aceptación e iteraciones del desarrollo del sistema.

[22] *Prof. Amelia Soriano http://carolina.terna.net/ingsw3/datos/Semana2_Calidad.pdf*

1.3. Identificación de Actores y Casos de Uso

1.3.1. Actores del sistema

Los actores son los diversos tipos de usuarios que un sistema posee, cada tipo de actor que esté definido en el sistema maneja toda una serie de permisos y casos de uso que pueden ser accedidos o no por otros tipos de usuario, los actores permiten agrupar y organizar al personal operativo de un sistema. Para el caso del **MSIS** los actores definidos son los siguientes:

- Administrador.
- Autor.
- Monitor.
- Soporte.
- Implementación.

Estos actores definidos dentro del **MSIS** son el resultado de estandarizar a los diversos usuarios según sus funciones y estas, transformarlas en roles dentro del sistema. El detalle de las funciones, permisos y casos de uso accesibles de estos actores, se encuentra descrito dentro de los documentos **Visión (Apéndice A)**, **Modelo Casos de Usos (Apéndice F)** y **Especificaciones de Casos de Uso (Apéndice G)**.

1.3.2. Casos de Uso

Los Casos de Uso son las transformaciones de los requerimientos funcionales dentro del sistema en actividades definidas; éstas capturan los requerimientos y los transforman en una sucesión de procedimientos, acciones o funciones. El proceso de definir todos los parámetros necesarios para realizar esta transformación es el caso de uso.

Dentro de un caso de uso existen una serie de parámetros los cuales facilitan el proceso de traducción del requerimiento. Estos parámetros fijos son los siguientes:

- Información característica: está conformada por la precondición, condición final de éxito y fallo, el disparador y actor relacionado al caso de uso.
- Flujo de eventos: lista todos los pasos por los cuales debe pasar el usuario para realizar el caso de uso.
- Planificación: compuesta por la prioridad, fecha tentativa del desarrollo y comentarios acerca del caso de uso.

Los casos de uso a generales definidos para el sistema son los siguientes:

- Administrar Alarmas.
- Administrar Tickets.
- Gestión de Seguridad.
    - Administrar Usuario.
    - Administrar Rol.
    - Administrar Sesión.
- Gestión de Entidades.
    - Administrar Cliente.
    - Administrar Sucursal.
    - Administrar Instancia.
    - Administrar Componente.
- Administrar Soluciones.
- Administrar SLA's.
- Gestión de Herramientas.
    - Administrar Agenda.
    - Administrar Asignaciones.
    - Administrar Mensajes.

El listado de casos de uso antes mencionado forma al Front End del Sistema. Éste es el aplicativo Web desarrollado bajo el Framework Foundation2. El listado de casos de uso perteneciente al Back End es el siguiente:

- Registro de Usuarios.
- Activar y desactivar tareas.
- Sincronizar datos del **MMS** hacia **MSIS.**

Además de los casos de uso del Back End, éste componente posee actividades configurables, las cuales pueden ser solo editadas por usuarios con rol administrativo las cuales son:

- Enviar mensajes SMS.
- Reportar SLA's violados.
- Reportar Alarmas.
- Balancear Tickets dentro del sistema.
- Generar Tickets automatizados.
- Actualizar datos dentro del sistema.
- Reportar Avisos.

Para tener una idea de la comunicación y detalle general de esos casos de uso consultar el documento de **Modelo de Caso de Uso (Apéndice F)** y para profundizar en su funcionamiento consultar las **Especificaciones de Casos de Uso (Apéndice G)**.

1.4. Elaboración de Lista de Riesgos

Al momento de emprender un sistema, existen toda una serie de temores asociados al cumplimiento o efectividad del desarrollo propuesto, estos temores pueden ser catalogados, calificados e inclusive hasta prevenidos, para delimitar el rango de incertidumbre y se les conoce como riesgos del sistema.

En esta fase y con ayuda de los Stakeholders, se lograron determinar todos los posibles riegos asociados al sistema **MSIS**. El riesgo de mayor importancia encontrado en esta etapa fue, no lograr generar y desarrollar todos los requerimientos funcionales y no funcionales, es decir no lograr extraer toda la información necesaria para el completo diseño del sistema solución, o también el no interpretar correctamente las funcionalidades planteadas, ya que esto podría causar retrasos e inclusive el rechazo del sistema.

Como medida de prevención, tanto los Stakeholders como los desarrolladores, sostuvieron toda una serie de reuniones y revisiones, con la idea de no perder de vista o malinterpretar los requerimientos y funcionalidades del sistema.

El resto de este listado, junto con sus calificaciones, impactos y demás detalles que permiten delimitar al riesgo, lo podrán encontrar dentro del documento de **Lista de Riesgos (Apéndice E)**.

En resumen la fase de Inicio (Levantamiento de información), fue la etapa de recopilación de datos y en ésta se logró la construcción básica de la solución y el estudio del ambiente en el cual deberá desenvolverse el producto final. Cabe destacar que finalizando la fase se sostuvo una reunión con los Stakeholders, quienes revisaron y supervisaron el completo y efectivo desarrollo de la misma con la idea de seguir la planificación establecida. Ésta se refleja dentro de los documentos **Plan de Desarrollo (Apéndice C)** y **Plan de Acción (Apéndice D)**.

2. Fase de Elaboración (Diseño del Sistema)

En esta fase del desarrollo se plantea el diseño del prototipo inicial del sistema, estructuras relacionales y lógica del sistema, hasta lograr un modelo de clases de la solución propuesta. A grandes rasgos se plasma la arquitectura completa del sistema; esta fase es crucial en cualquier desarrollo debido a que formulará todas las

estructuras, funciones, procedimientos y clases para el cumplimiento de los requerimientos obtenidos en la etapa anterior.

2.1. Actividades realizadas

2.1.1. Diseño de la estructura lógica del sistema

En esta actividad se recopiló la lógica del sistema empleado antes del desarrollo de la solución y se extrajeron aquellos objetos, modelos y clases que podrían ser reutilizados, de esta recopilación no se logró obtener mucha información debido a que el sistema empleado no poseía gran robustez, ni estaba bien estructurado, esto implicó realizar un diseño nuevo imaginando que nunca existió tal sistema y gracias a los documentos desarrollados en la primera fase, se logró realizar el diseño de los paquetes, estructuras y objetos del sistema manteniendo el estándar establecido por la organización y el framework Foundation2.

2.1.2. Diseño del Prototipo

Esta actividad fue desarrollada por solicitud de los Stakeholders asociados al proyecto, con la idea de poder intervenir en el proceso de diseño de las interfaces reflejando su opinión sobre el look and feel del Front End; éste fue realizado sobre el framework Foundation2, lo cual ayudó a profundizar los conocimientos del mismo.

El prototipo contenía casi todas las vistas disponibles del sistema así como también el flujo de las actividades. Esto ayudó en el proceso de refinar el diseño del Front End y permitió establecer el estándar de las plantillas, vistas y acciones relacionadas. Éste fue evaluado junto con el diseño de la estructura lógica por uno de los Stakeholders (César Celis) finalizando la primera iteración dentro de esta fase.

2.1.3. Diseño del Modelo de datos

En esta actividad se recopiló el modelo de datos o modelo relacional del sistema empleado antes del desarrollo y se extrajeron de éste todas aquellas entidades que

deberían formar parte del nuevo sistema; este modelo de datos poseía muchas inconsistencias, des-normalizaciones, y ausencias de reglas de integridad, entre otras fallas. La utilidad que se le dio a esta recopilación fue reconocer las entidades básicas del sistema, partiendo de este listado se desarrolló el nuevo modelo relacional el cual contendría nuevas tablas para satisfacer los requerimientos solicitados en la fase anterior.

Al modelo relacional desarrollado, se le añadieron toda una serie de reglas, validaciones y especificaciones con la idea de liberar a los modelos del sistema de ciertas funciones de validación y restricciones.

2.1.4. Diseño del Modelo de clases

En esta actividad se aprovechó, la experiencia obtenida de la realización del prototipo, sobre el framework Foundation2 y de las reglas, restricciones y estándares impuestos por éste, para realizar el diseño del Modelo de Clases del Front End del sistema, con la idea de mejorar las cualidades del framework y realizar una estructuración organizada de los objetos, modelos y clases de éste, sin perder de vista las pautas marcadas, aplicando las mejores prácticas de desarrollo.

El diseño del Back End de la solución, está adaptado a los estándares del MSC Scheduler de la organización, el cual obliga a utilizar ciertas clases e interfaces para el óptimo desarrollo de las tareas. Fuera de éstas, los objetos y estructuras desarrollados para Back End, están basados en las mejores prácticas de desarrollo. Estas estrategias están impuestas por lo desarrolladores.

Estas actividades realizadas fueron revisadas por los Stakeholders, así como también los entregables desarrollados en esta fase.

2.2. Arquitectura del Sistema MSIS

2.2.1. Representación de la arquitectura

La arquitectura del sistema **MSIS**, es de manera general una **arquitectura centralizada débilmente acoplada**, entre tres componentes o subsistemas, dos de estos conforman el Back End del sistema y el otro el Front End del mismo. A continuación se explican de manera más detalla los subsistemas que conforman al **MSIS**.

- **Back End**
  1. *Cliente MSIS:* Este componente cumple con la funcionalidad de registrar a los usuarios dentro del subsistema Servidor MSIS y a su vez permite desplegar los diversos mensajes de alertas y avisos emitidos por el componente antes mencionado.
  2. *Servidor MSIS:* Este componente cumple con las tareas de generar tickets de manera automática dependiendo del comportamiento del sistema **MMS**, balancear y distribuir los tickets de los usuarios, encargarse del registro de los clientes e informar a éstos con avisos y alarmas por diversas vías sobre actividades pendientes, notas agendadas y tickets cuyos SLA estén por vencerse.

- **Front End**
  1. *Aplicativo Web MSIS:* Este aplicativo maneja todo el proceso asociado a la gestión de tickets, es decir la creación, seguimiento, cierre, manejo de proyectos, clientes, instancias, creación de avisos, establecimiento de SLA's, entre otra gran cantidad de funciones; este sistema operacional además permite controlar los tipos de usuario del sistema, crearlos y asociarles roles de trabajo además de permitir generar reportes especializados que son de suma utilidad para la alta gerencia.

A continuación mostramos un diagrama general de la arquitectura planteada:

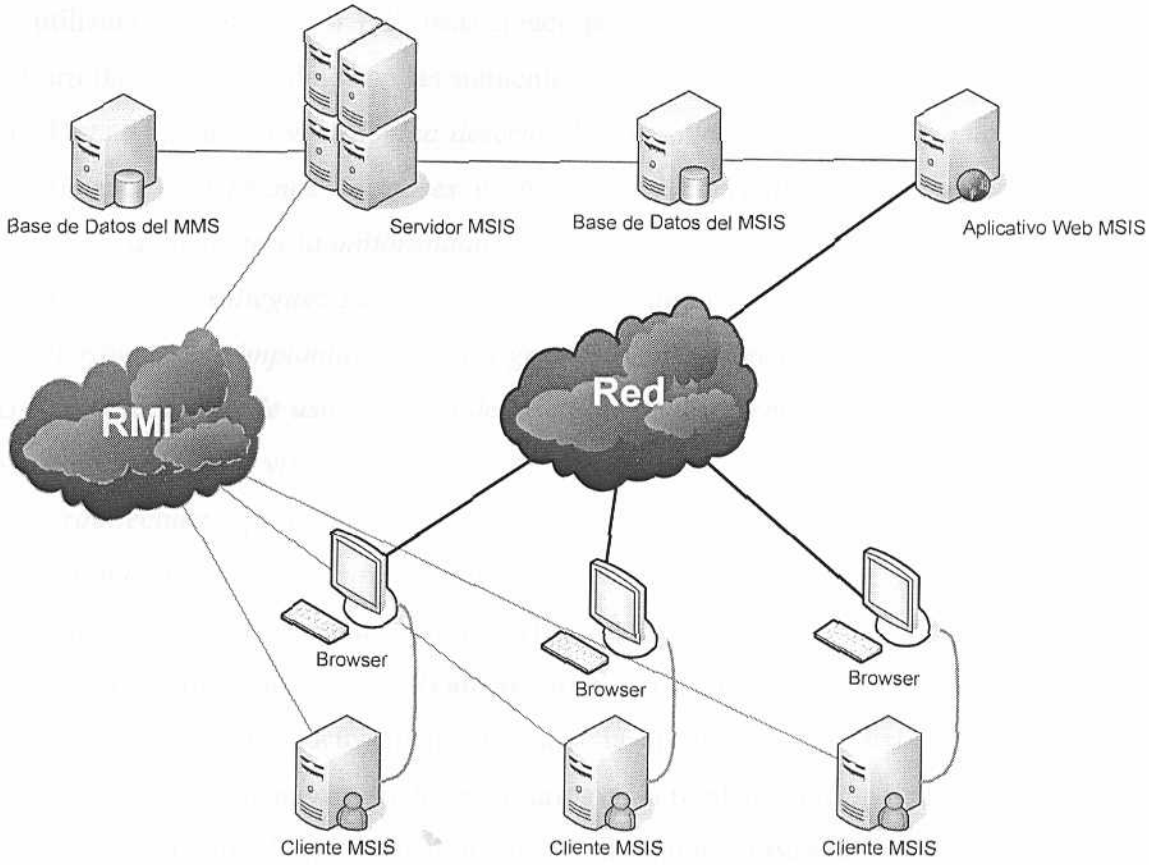


Gráfico 2 Arquitectura General del Sistema MSIS [23]

Es importante destacar que todos los subsistemas están basados en una arquitectura de tres capas (**Three-Tier** o **Three-Layer**), específicamente las capas son: la de datos, Lógica y presentación.

La comunicación entre el **Cliente MSIS** y el **Servidor MSIS** se realizó a través del mecanismo RMI (**Remote Method Invocation**), gracias a este mecanismo es posible la realización de diversas acciones como la suscripción de los usuarios dentro del **Servidor MSIS** y así habilitar el envío de mensajes puntuales a estos.

[23] *Desarrollado por los autores.*

Para lograr un mayor nivel de detalle acerca de la arquitectura del sistema **MSIS**, se utilizará el modelo de 4 + 1 vistas creado por Philippe Krutchen. Las vistas que se desarrollan en este modelo son las siguientes:

- ***Vista Lógica:*** *La vista lógica describe los aspectos estáticos y dinámicos de un sistema en términos de clases y objetos y se concentra en la abstracción, el encapsulamiento y la uniformidad.*
- ***Vista de Despliegue:*** *La vista de despliegue describe los diferentes recursos de hardware y la implantación del programa en dichos recursos.*
- ***Vista de Casos de uso:*** *La vista de los casos de uso forma el adhesivo que unifica el resto de las vistas. Los casos de uso motivan y justifican las opciones de arquitectura, permiten identificar las interfaces críticas, fuerzan a los diseñadores a concentrarse sobre los problemas concretos y demuestran y validan las otras vistas de la arquitectura.*[24]
- **Vista de Implantación o Realización:** Se preocupa de la organización de los módulos del producto final. Muestra la asignación de los Componentes, subsistemas y demás artefactos necesarios en la implementación de la solución.
- **Vista de Datos:** Representa el proceso de diseño de la Base de Datos del sistema, además en esta vista por lo general se explican las entidades principales del antes mencionado diseño.

2.2.2. Restricciones y objetivos de la Arquitectura

La arquitectura del sistema se desarrolló para que éste cumpliera las siguientes funcionalidades críticas que garanticen su éxito:

- El sistema debe permitir el manejo efectivo de los tickets suscitados en el día. Con esto lo que se plantea es proveer una herramienta adaptada a las necesidades de resolución de eventos dentro de los departamento de Monitoreo, Soporte e Implementación de la organización.

---

[24] *Introducción al UML Mari Carmen Otero Vidal*
http://www.vc.ehu.es/jiwotvim/IngenieriaSoftware/Teoria/BloqueII/UML-1.pdf

- La comunicación entre los **Clientes MSIS** y el **Servidor MSIS** debe ser totalmente transparente e independiente, no importa su localidad, para el buen funcionamiento de la solución.
- Los tickets generados deben mantenerse dentro de históricos que permitan la elaboración de reportes que ayuden a la gerencia y también para mantener un historial de los diversos eventos suscitados en todos los clientes de la organización.
- El sistema debe soportar un gran volumen de datos y clientes recurrentes, es decir manejar concurrencia de eventos y mantener un comportamiento estable efectivo para la resolución rápida de los tickets.
- El sistema debe estar disponible para todos los usuarios pertenecientes a los departamentos afectados, todo el tiempo.
- El **MSIS** debe permitir la posibilidad de poder encontrarse distribuido en diversos servidores o computadores, y darle la ilusión al usuario de estar ubicado en uno solo. Esto en cuanto a tiempos de respuesta, accesos, etc.

2.2.3. Vista de Casos de Uso

En esta vista se debe detallar las funciones y casos de uso que le pertenecen a cada uno de los actores del sistema mediante diagramas y, para un mayor detalle de estos casos, una descripción del funcionamiento de cada uno de estos. Para visualizar el análisis de esta vista, su despliegue y detalle favor consultar los documentos de **Modelo de Casos de Uso (Apéndice F)** para los diagramas, y **Especificaciones de Caso de Uso (Apéndice G)** para el detalle de su funcionalidad.

2.2.4. Vista Lógica

En esta vista se describen los subsistemas, paquetes y detalles de ciertos componentes funcionales que conforman al **MSIS**. Esto a través de diagramas de paquetes y clases que se detallan a continuación:

- Paquetes del Front End (Subsistema **Aplicativo Web**):

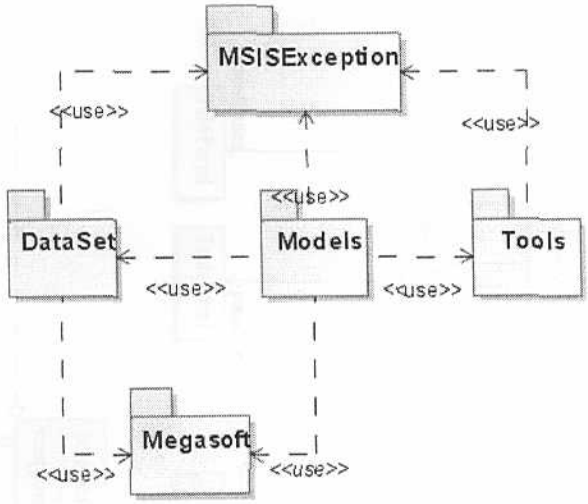


Gráfico 3 Diagrama de Paquetes del Front End [25]

- Paquetes Back End (Subsistema **Servidor MSIS** y **Cliente MSIS**):

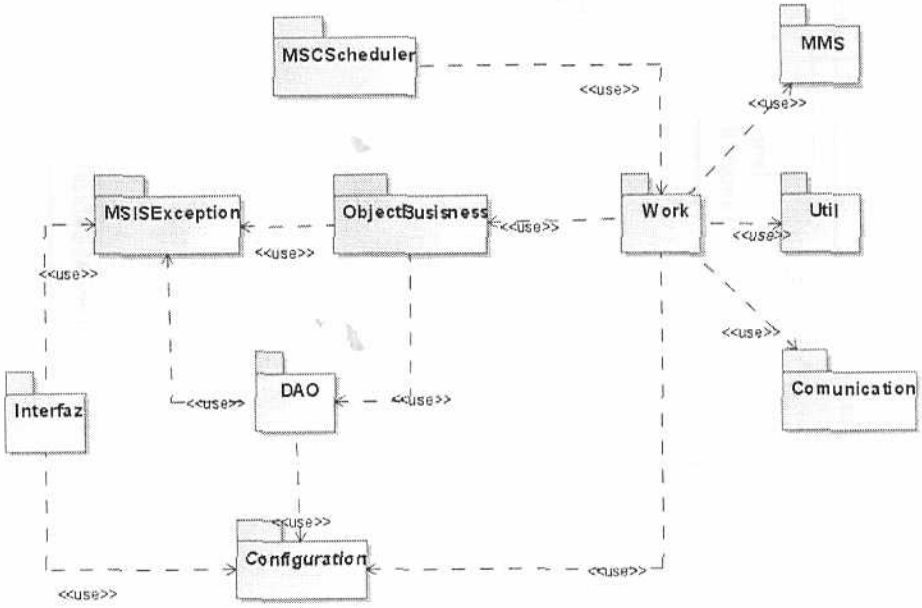


Gráfico 4 Diagrama de Paquetes del Back End [26]

[25] *Desarrollado por los autores.*
[26] *Desarrollado por los autores.*

- Interacción de Back End:

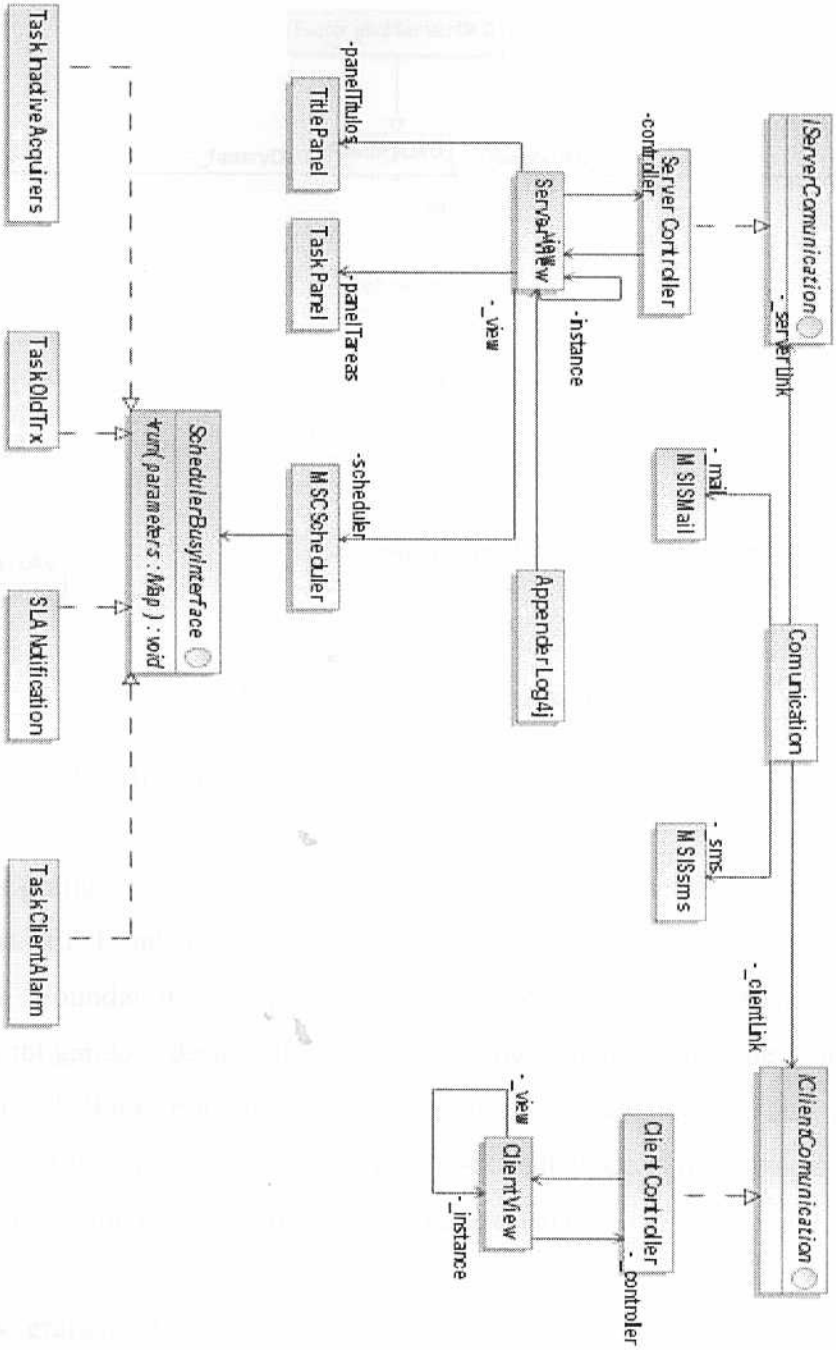


Gráfico 5 Diagrama de Clases del Back End (Ejemplo) [27]

[27] *Desarrollado por los autores.*

o Funcionamiento del DAO:

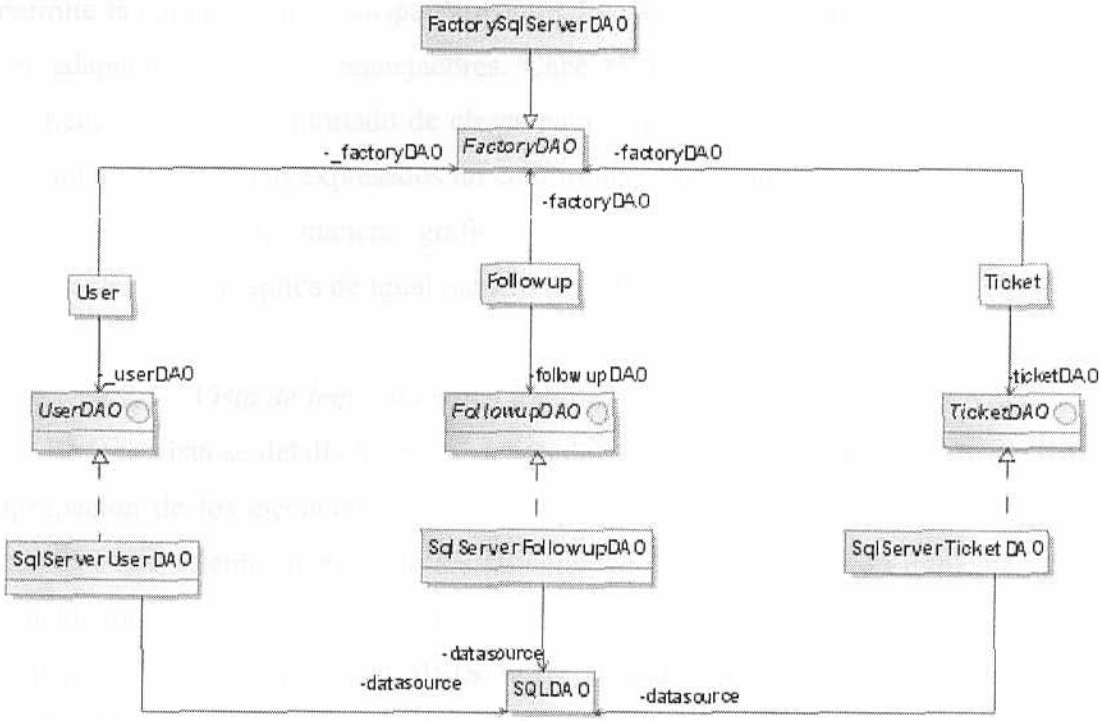


Gráfico 6 Diagrama de Clases del DAO (Ejemplo) [28]

Los gráficos 3 y 4, reflejan la estructura y organización de paquetes de los subsistemas. El Front End posee una organización muy simple por causa de Framework Foundation2; éste limita las posibilidades de generar estructuras complejas obligando a desarrollar cada acción del sistema dentro de la arquitectura antes vista. El Back End si se encuentra más estructurado ya que no existen limitaciones de ningún Framework y el MSC-Scheduler de la organización se adapta fácilmente a cualquier tipo de arquitectura desarrollada.

Los gráficos 5 y 6, describen con mayor detalle el funcionamiento de estructuras de suma importancia como lo son: el Back End con sus procesos de

[28] *Desarrollado por los autores.*

comunicación, envío de mensajes y alertas y el funcionamiento del DAO, el cual permite la comunicación transparente entre diversas bases de datos e inclusive puede ser adaptada a diversos manejadores. Cabe resaltar que los dos últimos gráficos contienen un número limitado de clases para simplificar su entendimiento, es decir, los objetos de negocio expresados no conforman el todo del sistema, más bien sirven para representar de manera gráfica y poco recargada la interacción de los componentes, esto aplica de igual manera para el resto de los objetos.

2.2.5. Vista de Implantación o Realización

En esta vista se detalla la organización de la solución, es decir como se realizó la agrupación de los ejecutables, documentos, fuentes y subsistemas del **MSIS**. Cabe destacar que el empaquetado de los modelos de diseño lógico y de implementación son idénticos, es por ello que para obtener un mayor detalle acerca de la vista de implantación o realización del **MSIS**, se recomienda consultar el documento **Modelo de Implementación** (**Apéndice I**) en éste se detalla el contenido almacenado en cada unos de estos módulos y las razones de su organización.

2.2.6. Vista de Despliegue

En esta vista se describe cuál será la estructura física encarga de sostener al sistema, esto incluye los enlaces, redes y demás facilidades provistas por la organización para garantizar el buen desenvolvimiento del sistema solución.

A continuación se muestra el gráfico que engloba el despliegue del **MSIS**:

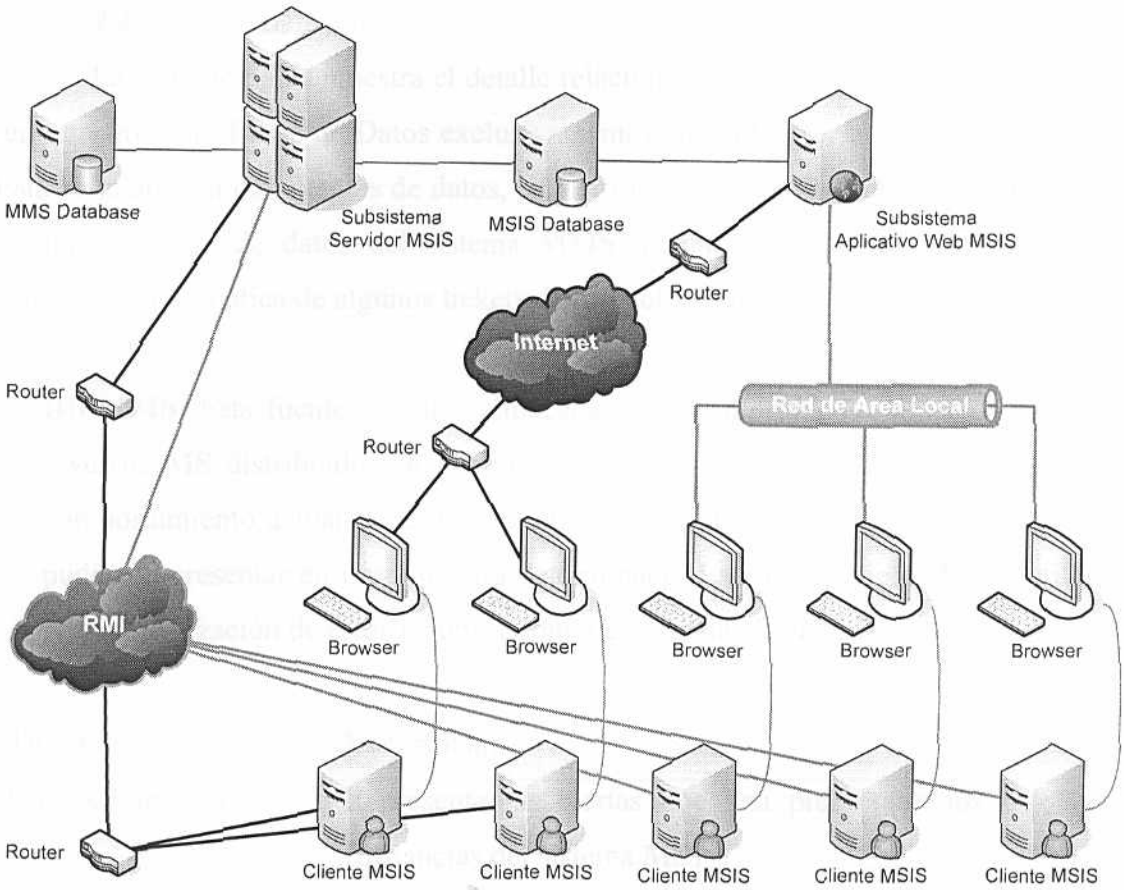


Gráfico 7 Vista de Despliegue general del Sistema MSIS [29]

En el gráfico anterior podemos notar que el sistema es capaz de estar dentro y fuera de una red de área local y funcionar de igual manera manteniendo la transparencia hacia los usuarios. Cabe destacar que actualmente el sistema solamente se encuentra implantando dentro de una red local dentro de la organización, por lo cual todos los enlaces están dentro de la red privada interna.

[29] *Desarrollado por los autores.*

2.2.7. *Vista de Datos*

La vista de datos muestra el detalle relacionado con todas las fuentes de datos (en nuestro caso, Bases de Datos exclusivamente) utilizadas por el sistema solución. Esta vista analiza dos fuentes de datos, primero la base de datos del sistema **MSIS** y segundo la base de datos del sistema **MMS**, que permite el monitoreo para la generación automática de algunos tickets dentro del sistema.

- **DB MMS**: Esta fuente de datos almacena toda la información relacionada a los diversos **MS** distribuidos en los clientes de la organización, en esta se aloja el comportamiento actual de dichos sistemas así como también las fallas que estos pudiesen presentar en un momento determinado. Las tablas que el **MSIS** utiliza para la realización de estos tickets automáticos son las siguientes:

Tablas	**Descripción**
ClientsWarnings	Representa las alertas que han presentado los Clientes (instancias del sistema **MS**)
ConnectionStatusAC	Representa el estado de la conexión entre un Cliente y un Acquirer
ConnectionStatusMMC	Representa el estado de la conexión entre un Merchant y un MerchantClient
DownTimeReports	Representa los Reporte de "caídas" o sin funcionamiento de las diferentes instancias del sistema **MS**
MerchantWarnings	Representa las alertas que han presentado los diferentes Merchants de las instancias de los sistemas **MS**
PendingCloseReports	Representa los reportes de Cierre Pendiente de cada Terminal de las diferentes instancias del sistema **MS**
TerminalWarnings	Representa las alertas que han presentado los diferentes Terminals de las instancias de los sistemas **MS**

VTerminalWarnings	Representa las alertas que han presentado los diferentes VTerminal de las instancias de los sistemas **MS**

Tabla 1 Tablas del **MMS** utilizadas por el **MSIS** [30]

- **DB MSIS**: Ésta es la fuente de datos del **MSIS**, se encarga del almacenamiento, de los tickets y demás datos del sistema. El detalle de ésta será analizado más adelante en este capítulo.

2.3. Diseño y modelado de la Base de Datos del sistema

La base de datos del **MSIS** fue ideada con la idea de salvaguardar los datos del sistema, éstos incluyen desde sus tickets, abarcando todo lo relacionado con sus clientes, mensajes, SLA, hasta llegar a los usuarios y roles de estos dentro del sistema.

Las tablas más importantes dentro del sistema de MSI son las siguientes:

Tablas	**Descripción**
Client	Representa el listado de Clientes que mantienen contratos con la organización.
Component	Representa a los elementos físicos que conforman a todos los productos comercializados e internos de la organización.
Establishment	Representa las Sucursales o Establecimientos de los Clientes.
Followup	Representa los Seguimientos que se le realizan sobre los Tickets.
Instante	Es la representación física del Componente.
Note	Representa recordatorios para Tickets, Usuarios o Grupos de Usuarios según sea el caso.
Portal_Role	Representa los distintos roles y niveles de acceso que puede tener un

[30] *Desarrollado por los autores.*

	Usuario en el **MSIS**
Portal_User	Representa a cada Usuario del **MSIS**
Project	Representa los Proyectos y productos ofrecidos por la organización.
SLA	Representa los Niveles de Acuerdo de Servicios
Solution	Representa las Soluciones recomendadas para ciertos Tickets.
Ticket	Representa el evento suscitado que requiere atención.

Tabla 2 Listado de Entidades principales del **MSIS** [31]

La Base de datos del **MSIS** está completamente desarrollada en SQL Server 2000 Service Pack 4 y fue diseñada gracias al aplicativo ERwin Data Modeler, de la compañía AllFusion, versión 4.1.2522.

El detalle acerca de las tablas, atributos, explicaciones y modelo ER, se encuentra dentro del documento **Modelo Lógico Relacional** (**Apéndice H**), en éste se podrá apreciar en detalle el resto de la información, acerca de la BD del sistema y la del **MMS**.

2.4. Plan de desarrollo del software

El plan de desarrollo consiste en el proceso de organizar y estructurar cronológicamente, por medio de parámetros como, prioridad, dificultad, etc., las etapas, iteraciones y revisiones relacionadas con la creación de la solución, partiendo de las herramientas generadas durante el levantamiento de información.

2.4.1. Descripción del Proyecto

El sistema **MSIS** busca como fin agilizar el proceso de gestión de tickets dentro de la organización y está formado por tres subsistemas, los cuales trabajan en conjunto para mejorar la calidad de trabajo de los departamentos de Monitoreo,

[31] *Desarrollado por los autores.*

Soporte e Implementación. Una de las tareas más importantes de esta solución, es el hecho de poder plasmar y manipular, dentro de las diversas Incidencias suscitadas, los SLA establecidos entre los clientes y la empresa.

La organización además busca con este sistema, la posibilidad de manipular por medio de reportes especializados el comportamiento de: los clientes, usuarios del sistema, tiempos de atención, tipos de tickets más frecuentes, desempeño de los grupos de trabajo, entre otros.

Como resumen el sistema **MSIS**, se desenvolverá dentro de la organización como un sistema de gestión de tickets personalizado para cada usuario, adaptado a las necesidades de la empresa la cual requiere resolver los eventos suscitados en sus productos de pago electrónico, de manera rápida y efectiva.

2.4.2. Alcance, propósito, objetivos y restricciones

El alcance, propósito, objetivos y restricciones perseguidos por este sistema se pueden resumir en este listado:

- Permitirá a los usuarios de los departamentos de Monitoreo, Soporte e Implementación, la gestión de tickets.
- Realizará la creación de Incidencias automáticas relacionadas al comportamiento de los **MS** (Merchant Server) de cada uno de los clientes, gracias a la BD del sistema MMS (Monitor Merchant Server) encargado de recopilar el comportamiento de estos.
- Generará reportes los cuales podrán ser utilizados por la alta gerencia para evaluar el desempeño tanto de los clientes del sistema, como de los usuarios.
- Permitirá avisar por diversos medios, notas personales, asignaciones y avisos o alertas sobre los acuerdos de niveles de servicio (SLA).

- Controlará las diversas etapas por las cuales puede pasar un ticket e inclusive asociarle a este una solución estándar, impuesta ya sea por la organización o departamento según convenga.
- Permitirá a los usuarios personalizar su ambiente de trabajo, ya que podrán modificar a su conveniencia toda una serie de opciones para mejorar o facilitar su trabajo.

El plan se estructuró con la idea de desarrollar en dos etapas todo el software, en la primera etapa se planeó el desarrollo del Front End y en la segunda el Back End. El plan contiene 4 iteraciones y al final de cada una de estas una revisión de los avances de la misma.

2.4.3. Etapa del Front End

En esta primera etapa se planteó desarrollar este aplicativo Web bajo el framework Foundation2. Para un efectivo desarrollo de esta etapa se planeó realizar dos iteraciones, la primera al terminar el desarrollo de los componentes fundamentales para la creación de tickets y la segunda al terminar la administración de estos. Por otro lado también se planeó realizar reuniones con los Stakeholders al finalizar cada iteración, con la finalidad de revisar los avances y emitir correcciones rápidas, las cuales no afecten el cronograma de actividades.

2.4.4. Etapa del Back End

En esta etapa se plantearon las actividades para realizar el desarrollo del segundo y tercer subsistema del **MSIS**, los cuales conforman el Back End del sistema; el **Servidor MSIS**, se desarrollará bajo el MSC-Scheduler de la organización, éste permitirá generar tareas de monitoreo sobre la BD del **MMS** para la creación de ciertas Incidencias de manera automática.

Otros puntos importantes a resaltar sobre la planificación de esta etapa dentro del plan para el desarrollo del software son: las actividades relacionadas a los diversos medios para la emisión de mensajes y avisos o alertas, los medios establecidos en la planificación son: mail, mensajes directos a los usuarios (esto apoyados en el mecanismo RMI) y mensajes SMS, estos últimos sólo para casos muy especiales. Estas vías permitirán mantener informados tanto a los usuarios, como a los diversos Stakeholders, sobre los SLA y sobre las actividades almacenadas o asignadas.

Por ultimo una funcionalidad del Back End de suma importancia, es la reasignación de los tickets, esta actividad reducirá la posibilidad de que existan tickets sin asignación u olvidados, este requerimiento específico, es vital para la organización debido a que con su exitosa implantación mejorará la imagen frente a los clientes, además de balancear y mantener prácticamente con similares cargas de trabajo a los usuarios del sistema en todo momento.

Para obtener un mayor detalle sobre el cronograma desarrollado se recomienda consultar los documentos de **Plan de Acción (Apéndice D)** y **Plan de Desarrollo (Apéndice C)** y para detallar las actividades internas de estas etapas, revisar el documento de **Especificaciones de Caso Uso (Apéndice G)**.

3. Fase de Construcción (Desarrollo del Sistema)

El desarrollo del software **MSIS** consta de dos etapas internas, cuatro iteraciones y cuatro revisiones, cada una de estas al final de cada iteración. Esto fue establecido en la planificación para el desarrollo del mismo. A continuación se describen los procesos de desarrollo de cada una de estas etapas junto con sus revisiones e iteraciones:

3.1.1. Etapa 1 desarrollo del Front End

En esta etapa se desarrolló todo el **Aplicativo Web MSIS** basado en el framework Foundation2 el cual conforma uno de los tres subsistemas del **MSIS**, este aplicativo debe permitir a los usuarios realizar la gestión de tickets, proponer soluciones estándar a tipos de tickets, permitir asentar los niveles de acuerdo de servicio (**SLA**) establecidos entre la organización y los clientes, asignar actividades y almacenar actividades (Modulo de Agenda), entre otras muchas funciones.

El cronograma exacto del proceso de desarrollo dentro de esta epata se puede apreciar dentro del documento **Plan de Acción** (**Apéndice D**). Cabe destacar que, dentro de esta etapa se realizaron dos iteraciones y revisiones, la primera de éstas fue antes de comenzar el desarrollo del Modulo de Atención de Tickets. En esta iteración se revisaron, probaron y evaluaron todas las actividades que suministran los datos necesarios para la creación de los tickets.

En el proceso de desarrollar el Front End del sistema existe un punto de suma importancia para el aplicativo; se trata del Módulo de Atención de Tickets. En éste se desarrolla todo el proceso de creación, seguimiento, visualización, cierre y reapertura de los tickets, lo cual implica que estas acciones deben concentrar el resto de la información operacional (Cliente, Sucursal, instancia, proyecto, entre otra serie de datos) que el sistema provee, para suministrarle a los usuarios una serie de plantillas que faciliten el proceso de gestión de estos eventos y así optimizar los procesos. La segunda iteración fue realizada al finalizar el desarrollo de este módulo antes comentado, junto con las revisiones y evaluaciones esperadas para determinar su efectividad.

3.1.2. Etapa 2 desarrollo del Back End

El desarrollo del Back End del **MSIS**, fue concebido con la idea de poder automatizar toda el proceso manual de creación de ciertas incidencias dentro del

sistema, generar avisos a los usuarios por diversas vías acerca del cumplimiento de los SLA sobre las incidencias (tipo de ticket) o sobre avisos, asignaciones y mensajes almacenados desde el Front End, balancear o distribuir los tickets para evitar la no resolución u olvido de éstos, entre otras tareas.

Este etapa fue constantemente supervisada por los Stakeholders del sistema, los cuales promovían las reglas para el buen funcionamiento de las actividades, estos permitieron determinar cuáles deberían ser los tipos de incidencias generados de manera automática, cuales deberían ser las restricciones y condiciones para balancear y reasignar las incidencias de los usuarios, así como también en su asignación al momento de su creación. En resumen suministraron gran cantidad de información para la construcción y establecimiento de los alcances de las tareas.

Dentro del proceso del Back End se realizaron las dos últimas iteraciones de la del plan de desarrollo del software para tener detalles cronológicos sobre éstas, consultar el documento **Plan de Acción (Apéndice D)**.

4. Fase de transición (Implantación)

En el proceso de transición se realizaron los ajustes y modificaciones recomendados por los Stakeholders sobre el software en general, además de realizar todo el proceso de implantación de la solución desarrollada.

4.1. Migración de datos

Al inicio del proceso de transición se planteó la idea de realizar una migración de los datos almacenados en el aplicativo anterior, debido a las múltiples inconsistencias que existían entre el sistema antiguo y el **MSIS** se tuvo que emprender una actividad con la ayuda de algunos Stakeholders para calificar y definir los tipos de tickets, además de realizar el proceso de traspaso de esa información.

La actividad de calificar los tickets del antiguo sistema, se realizó debido a que en el **MSIS** los títulos de los tickets están estandarizados y esto ocasiona la necesidad de clasificarlos para así poder recopilar la información.

4.2. Manual de instalación

Para la efectiva implantación del sistema **MSIS** se deben realizar ciertas implantaciones y configuraciones. Toda esta explicación se encuentra dentro del documento **Manual de Instalación** (**Apéndice J**), donde se especifica todo el proceso relacionado para la implantación efectiva del sistema.

4.3. Producto

Para una mayor compresión del sistema **MSIS** se desarrolló un CD con toda la documentación generada en proceso de desarrollo, con los ejecutables o aplicativos de cada uno de los subsistemas y las fuentes del mismo, con la intención de facilitar la compresión, mantenimiento y desarrollo de nuevas funcionalidades o módulos dentro del mismo. Este CD conforma el último artefacto desarrollado con el nombre de **Producto**.

CAPITULO V

Análisis de Resultados

El desarrollo del **MSIS**, permitió la creación de una herramienta para la gestión de tickets, completamente adaptada a las necesidades de la organización, además de facilitar y controlar de manera automática muchos de los procesos manuales anteriormente ejecutados por los usuarios finales.

Los resultados alcanzados en el desarrollo de esta solución se detallan a continuación bajo el siguiente esquema:

Primer objetivo perseguido

Desarrollar un Sistema de Ticket de nombre Mega Soft Issue System (**MSIS**) basado en los TTS (Trouble Ticket System) o también llamado en algunos casos Issue tracking system, el cual permita poder listar de acuerdo a diversas pautas o Services Levels Agreements (**SLA**) las incidencias que ocurren en la organización.

Resultado del objetivo

Se desarrolló un sistema de nombre **MSIS**, basado en tres subsistemas, los cuales trabajan en conjunto para manipular tickets de manera efectiva y eficiente; está basado en los Issue Tracking System o Sistemas de gestión de Tickets. Una de sus funcionalidades es el poder asociar a las incidencias (Entiéndase por éstas como un tipo de ticket específico) a los niveles de acuerdo de servicio o SLA, los cuales permiten controlar los tiempos de atención estipulados entre la organización y los clientes.

Estas asociaciones permiten plasmar los contratos de atención establecidos por la organización y, gracias a estos, controlar los tiempos de atención y desempeño de los usuarios en los procesos de gestión exclusiva de las incidencias donde apliquen estos SLA.

Segundo objetivo perseguido

Generar reportes gerenciales los cuales sirvan como indicadores de desempeño tanto del sistema como de los usuarios del mismo.

Resultado del objetivo

Con la ayuda del framework Foundation2 y de las librerías de diseño gráfico JasperReport, para la creación de reportes en aplicaciones Java, se logró el desarrollo de un módulo de reportes gerenciales, con la idea de poder generar reportes gráficos y detallados sobre el comportamiento de los clientes, sistema y usuarios del mismo. Entre los reportes más importantes que podemos resaltar se encuentran: tickets gestionados por grupos de trabajo, por usuarios, tipos de tickets generados al mes, tiempos de respuestas relacionados a los tickets, cumplimiento de SLA, entre otros.

El módulo de reporte desarrollado, aportará ayuda exclusivamente a la alta gerencia y Stakeholders de la organización relacionados al sistema, para el resto de los usuarios tan sólo podrá ofrecer reportes sobre su desempeño y el de sus grupos.

Tercer objetivo perseguido

Realizar todo un levantamiento de datos para determinar toda la funcionalidad del sistema, componentes que utilizará, etc. Además de determinar y localizar las fuentes informativas necesarias para el funcionamiento del **MSIS**.

Resultado del objetivo

Gracias a la metodología aplicada (RUP), se desarrollaron toda una serie de artefactos los cuales permitieron realizar el levantamiento de información para determinar las funcionalidades de la solución.

Además de los artefactos, las entrevistas, reuniones y constantes revisiones sostenidas con los Stakeholders y usuarios del sistema permitieron tanto el desarrollo de los artefactos, como el de las funciones que el sistema tendría. Unos de los factores más importantes en la fase de inicio (levantamiento de información), fue el proceso de discusiones con los Stakeholders, debido a que estos ya manipulaban una idea general de la solución que planeaban desarrollar y fueron los encargados de suministrar la información, además de ser los revisores de los artefactos desarrollados.

Cuarto objetivo perseguido

Realizar un Componente generador de alarmas, el cual sea capaz de enviar dichas alertas por diversos medios, como SMS, Mails, entre otras posibles vías. Rigiéndose por niveles jerárquicos o prioridades de las incidencias y los SLA de la organización con sus clientes.

Resultado del objetivo

Uno de los tres subsistemas del **MSIS**, es el **Servidor MSIS**. Éste se desarrolló con la idea de automatizar muchas de las tareas del sistema, además de realizar el envío de avisos y alarmas. Estos avisos o alarmas están atados a dos módulos del sistema, los SLA y los avisos o asignaciones.

Para los SLA se desarrolló un modulo de niveles jerárquicos parametrizable. Éste, permite realizar el envío de alertas y alarmas a un grupo específico de usuarios o Stakeholders del sistema, cada SLA posee sus propios niveles jerárquicos.

En el caso de los avisos o asignaciones se desarrolló una especie de módulo de agenda, en el cual los usuarios almacenan tareas para ser recordadas en un determinado momento, aplica de igual manera para las asignaciones.

El proceso de envío fue concebido para nunca perder ningún tipo de aviso, es decir siempre que un usuario esté dentro del sistema, sus avisos llegarán directamente a su ordenador, en caso contrario llegarán vía mail y sólo para aquellos casos que lo requieran podrán ser enviados vía SMS. Estos avisos por lo general serán exclusivos para los Stakeholders y administradores del sistema.

Quinto objetivo perseguido

Integrar los Services Levels Agreements (SLA) de los clientes al **MSIS**, mediante un módulo, para poder disparar las alarmas necesarias en caso de que exista alguna violación del algún SLA. Estos deberán ser configurables durante el tiempo y por cliente.

Resultado del objetivo

Dentro del Front End (**Aplicativo Web MSIS**) se desarrolló un módulo exclusivo para la definición de los SLA. En éste se indican los elementos de un cliente que estarán asociados al SLA (Instancias asociadas) además de definir las trazas de tiempo internas de éste y sus niveles jerárquicos. Cabe destacar que el módulo permite la modificación y reemplazo de SLA definidos.

Estos SLA son piezas fundamentales para el envío de alertas y avisos. El **Servidor MSIS** monitorea todos los tickets que guarden relación con una instancia. Si sobre esta existe un SLA existirá la posibilidad de realizar los envíos, avisos y alertas pertinentes relacionados al ticket.

Sexto objetivo perseguido

Integrar al Sistema **MSIS** con sistemas como el **MMS** (Sistema de Monitoreo Merchant) para evitar la redundancia de datos y utilizarlo como posible fuente de información al sistema.

Resultado del objetivo

El **Servidor MSIS** (subsistema del **MSIS**) es el encargado de realizar la integración con el **MMS**, dentro de este subsistema se ejecutan una serie de tareas, entre las cuales está la extracción de datos de la BD del **MMS** hacia el **MSIS** con la finalidad de evitar la redundancia de información, entre otras de las actividades está la de generar de manera automática incidencias sobre el comportamiento de los **MS**, que son monitoreados por el **MMS**.

La integración de estos dos sistemas se desarrolló mediante un DAO, el cual permite realizar las conexiones a BD de las dos y mediante procesos de análisis generar las incidencias relacionadas a los comportamientos anormales de los **MS**.

Conclusión

Como conclusiones del desarrollo del **MSIS** podemos listar las siguientes:

- Se desarrolló una herramienta capaz de manipular, controlar y gestionar los eventos suscitados dentro de la organización, basándose en los Issue Tracking System.
- Se integró al **MSIS** con el **MMS** para la automatización de algunas incidencias, recopilación de la data de todos los **MS** con el motivo de evitar la redundancia y finalizar el desarrollo de las herramientas de trabajo para el **CMM** (Centro de Monitoreo Merchant).
- Se desarrolló por medio de avisos en diversos medios, controles para los SLA y alarmas del sistema, con la finalidad de mantener informados a todos los usuarios y evitar la pérdida o falla en la gestión de los tickets.
- Se recopilaron toda una serie de artefactos y documentos para lograr la determinación de todas las funcionalidades del sistema. Toda esta información podrá servir para futuros desarrollos de la empresa.
- Se idearon toda una serie de reportes los cuales permitirán a los Stakeholders, tener resúmenes informativos y gráficos acerca del comportamiento de diversos elementos del sistema. Éstos podrán ayudar a tomar decisiones con respecto al funcionamiento o problemas presentados en los clientes del sistema.
- Se desarrolló todo un proceso de administración de elementos informativos para la óptima creación de tickets, englobando en cada uno de éstos toda la información necesaria para la resolución rápida y sencilla de los eventos enmarcados, es decir los tickets contienen toda la información necesaria del entorno donde ocurre el evento que se está reportando.

Recomendaciones

El trabajo de grado ofrece oportunidades amplias para su perfeccionamiento y profundización, pero difícilmente para ser tomado como otro posible tema de tesis, esto debido a que ya la base y estructura del sistema están completamente desarrolladas, lo cual indica que la profundización puede ser vista a nivel de nuevos reportes, automatizaciones de tareas y funcionalidades nuevas relacionadas al funcionamiento principal del sistema (Gestión de tickets).

Dentro de las implicaciones inmediatas se puede comentar el proceso de recuperación de los datos del antiguo sistema, sólo de algunas entidades, debido a que éste no es del todo compatible con el **MSIS,** para esto se deben desarrollar tareas en la aplicación o funciones y procedimientos dentro de la BD para el proceso de migración de la información. Cabe destacar que esta información antes debería ser analizada, con la idea de no recuperar aquella información viciada o mal gestionada.

A largo plazo se planea el desarrollo de nuevas tareas automáticas y procesos de avisos más especializados, así como también la posibilidad de extender el sistema al resto de las sucursales de la organización.

Las áreas en las cuales el sistema desarrollado podrá ser utilizado, son todas aquellas cuya funcionalidad pueda ser asociada a los call center's o departamentos relacionados con la gestión de eventos. Para el caso exclusivo de la organización es para los departamentos de Monitoreo, Implementación y Soporte, los cuales deben estar constantemente atendiendo peticiones y solventando eventos en tiempos preestablecidos, manipulando toda una serie de datos o elementos los cuales permiten simplificar el buen desenvolvimiento de sus labores.

Referencias bibliográficas

- Ing. Febe Ángel Ciudad Ricardo (2000). *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos.*
- *Sun Developer Network forums* [Documentos WWW]. URL inicial: *http://forum.java.sun.com/index.jspa.*
- Susan Sanderson (2001). Introduction to Help Desk Concepts and Skills (1a edición).
- Turban, Efraim, Aronson, Jay E., Liang Ting-Peng (2002). Decision Support Systems and Intelligent Systems (7a edición).
- Mar Bru Soler (2003). *Diseño y Construcción de un sistema para la monitorización de campos electromagnéticos.* Universidad Politécnica de Valencia. [Documento PDF] URL: *https:/.../C1/Material/Documentos%20Disponibles/PFC-ETSIA UPV%20Ejemplo2%20de%20aplicación%20de%20RUP.pdf*
- *Introducción a la metodología RUP Departamentos de sistemas informáticos y computación.* Universidad Politécnica de Valencia. [Documento WWW]. URL: https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc
- Andrew S. Tanenbaum (2003). *Sistemas operativos modernos.* (5ª Edición): Pearson Educación.
- W. Stallings (2005). *Sistemas Operativos* (5a edición): Prentice-Hall.
- A. Silberschatz, P. Galvin, G. Gagne (2003). *Conceptos de Sistemas Operativos* (6a edición): Limusa Wiley.
- Philippe Kruchten (2000). *"The Rational Unified Process. An Introduction".* (2a edición): Addison Wesley.

Tabla de Contenidos

1. Introducción 4
1.1 Propósito 4
1.2 Alcance 4
1.3 Referencias 4
1.4 Descripción 4
2. Definiciones 5
2.1 MMS 5
2.2 MS 5
2.3 MSIS 5
2.4 CMM 5
2.5 SLA 5
2.6 Ticket 5
2.7 Instancia 5
2.8 Componente 5
2.9 Proyecto 6
2.10 Entidades Externas 6
2.11 ISO8583 6
2.12 Front End 6
2.13 Back End 6
2.14 RUP 6
2.15 JasperReport 6
2.16 IReport 7
2.17 GPL 7
2.18 Stakeholder 7
2.19 Scheduler 7
2.20 DSS 7
2.21 TTS 7
2.22 SMS 7
2.23 Three – Tier 7
2.24 Modelo 4 + 1 vistas 7

Glosario

1. Introducción

El presente documento tiene como finalidad, proveer información acerca de toda una serie de términos relacionados al **Sistema de Incidencias MSIS,** los cuales no son familiares a los usuarios o lectores, con la idea de facilitar el entendimiento del resto de los documentos y brindarles una especie de diccionario especializado de toda la terminología empleada.

1.1 Propósito

Funcionar como una herramienta de apoyo para familiarizarse y entender, todos aquellos términos propios del sistema, los cuales están presentes tanto en el tomo del TEG como en el resto de los documentos.

1.2 Alcance

Este documento contendrá términos asociados exclusivamente al sistema, las terminología descrita en el mismo por lo general son breves y claras, con la idea de proporcionar un entendimiento fácil y rápido. Desde un esquema global contiene conceptos asociados al **MMS**, **MS** y **MSIS** (siendo éste el Sistema propuesto).

1.3 Referencias

En el documento, alguno de los conceptos descritos, son resúmenes extraídos del marco referencial del tomo, además de otras referencias bibliográficas las cuales serán descritas a continuación:

- http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=15871. **ISO 8583** (International Organization for Standarization).

1.4 Descripción

A continuación se presentan todos los términos del documento, algunos con sus acrónimos asociados y otros tan sólo con una breve explicación.

2. Definiciones

2.1 MMS

Siglas de "**Monitor Merchant Server**", sistema conformado por tres componentes el **Cliente MMS**, **Servidor MMS** y el **Aplicativo Web Monitor MMS**. Su función principal radica en monitorear el comportamiento del **MS** (**Merchant Server**). El **MMS** envía información recolectada de los distintos servidores (por medio del ***Cliente MMS***) *donde se aloja el* **MS**, *a un repositorio de datos (controlado por el* **Servidor MMS**), con el propósito de manipular esta información (por medio del **Aplicativo Web**) para facilitar y estructurar reportes amigables a los usuarios finales, con la intención de facilitar la gestión de problemas asociados al **MS**.

2.2 MS

Siglas de "**Merchant Server**", sistema transaccional basado en el estándar **ISO8583** el cual estipula, el estándar a seguir para el envió y recepción de transacciones financieras. El **MS** permite realizar por un mismo punto *de venta transacciones y llevarlas a sus destinos correspondientes.*

2.3 MSIS

Siglas de "**Mega Soft Issue System**", nombre del sistema a ser desarrollado, el cual fungirá como un sistema de gestión de **Ticket's**.

2.4 CMM

Este el Centro de Monitoreo Merchant o también llamado Departamento de Monitoreo de la Organización.

2.5 SLA

Siglas de "**Service Level Agreements**" o Acuerdo de niveles de Servicio, son contratos establecidos entre la organización y sus clientes, en los cuales se estipulan, de manera global, los tiempos, mínimos y máximos, de respuesta a un problema asociado al producto que está en venta.

2.6 Ticket

Son llamados así los eventos que ocurren en una organización que son manejados dentro de un sistema de gestión. Para el **MSIS** los tipos de tickets con los cuales trabajamos son los siguientes: Administrativos, Control de *cambio, Incidencias, Operativas, Error. Estos tipos pueden crecer dentro del sistema.*

2.7 Instancia

Una instancia es a un componente como un objeto es a una clase, es decir, son extensiones físicas ubicadas dentro de los clientes, las cuales guardan relación con los diversos componentes que pertenecen a los productos de la organización.

2.8 Componente

Son los elementos, a modo genérico, que controla u organiza el sistema. Estos elementos por lo general

forman parte del Sistema **MS**, es decir, forman parte de la infraestructura necesaria para el buen funcionamiento del sistema, o del **MMS**. Para este caso hablamos a nivel de los tres posibles componentes que conforman al sistema.

2.9 Proyecto

Son los diversos productos que provee la empresa Mega Soft Computación a sus clientes, estos proyectos se están asociados a cada sucursal que lo amerite.

2.10 Entidades Externas

Son aquellas empresas u organizaciones, las cuales sostienen relaciones con la organización los proyectos.

2.11 ISO8583

Estándar establecido por la ISO (**International Organization for Standarization**) el cual regula cómo debe ser el proceso de comunicación, códigos de respuesta y seguridad en los datos enviados al momento de realizar una transacción financiera por medio de tarjetas o pagos en línea. En un nivel más técnico, especifica las interfaces, procesos y funciones que se deben utilizar para entablar las comunicaciones entre los adquirientes y los clientes que realizan pagos con tarjetas. Este proceso de envió de datos debe ser seguro y manejar ciertos protocolos establecidos por el estándar.

2.12 Front End

Puede ser una sección de código, componente de un sistema o fachada de un aplicativo, la cual interactúa directamente con los usuarios del sistema y provee una fuente amigable de visualización de los datos, además de facilitar el proceso de manipulación de los sistemas para aquellos usuarios no expertos, por lo general los front end son desarrollados para facilitar las actividades de los usuarios finales.

2.13 Back End

Es el controlador o manejador de la lógica del negocio o sistema, esta encargados de suministrar información al Front End, además utiliza la información para activar acciones sobre el sistema y su lógica. En ellos residen las verdaderas cargas de trabajo. Por lo general éstos son los encargados de todas aquellas operaciones de respaldo, administración de datos, etc. que deben ser transparentes para los usuarios del sistema.

2.14 RUP

Acrónimo de Rational Unified Process. Este representa la metodología utilizada en el desarrollo del sistema.

2.15 JasperReport

Librería gráfica encargada de la generación de reportes o informes, por medio de la interpretación de un tipo de archivo JRXML, basado en los XML. Se encuentra escrita en java y es libre (GPL). Esta capacitada para integrarse de manera fácil con la librería JFreeChart, los formatos posibles de salida son los siguientes: PDF, XML, HTML, CSV, XLS, RTF, TXT. Para la fácil generación de las plantillas JRXML se puede trabajar con la herramienta de diseño IReport.

2.16 IReport

Editor gráfico adaptado para la generación de archivos JRXML, de manera más sencilla, capaz de generar, de manera visual, reportes, gráficos, sub-reportes, etc. También esta herramienta es GPL.

2.17 GPL

Siglas de General Public License, está orientada principalmente a proteger la libre distribución, modificación y uso de software; permite el libre uso de éstos, sin la necesidad de cancelar matriculas o licencias de alto costo.

2.18 Stakeholder

Término empleado para describir al público altamente interesado, en el desarrollo o planificación de un proyecto, dentro de una organización pueden fungir como Stakeholders: los accionistas, dueños, coordinadores, gerentes, etc.

2.19 Scheduler

Término asignado a los programas planificadores de tareas. Dentro de la organización existe el MSC-Scheduler, que forma parte esencial en el desarrollo del Back End del **MSIS**.

2.20 DSS

Acrónimo otorgado a los programas de soporte a la toma de decisiones.

2.21 TTS

Acrónimo otorgado a los Trouble Ticket System en español a sistemas de gestión de Tickets.

2.22 SMS

Acrónimo otorgado a los Short Message Service, o mensajes de texto.

2.23 Three – Tier

Arquitectura de tres capas, esta se basa en la construcción de tres grandes componentes (capas), los cuales realicen y faciliten el desarrollo del software. Estas capas se comunican entre ellas, pero no deben ser dependientes, permitiéndoles poder ser removidas y reubicadas con mucha facilidad.

2.24 Modelo 4 + 1 vistas

Modelo para el análisis o fácil explicación de la arquitectura de un sistema, basado en 5 vistas que permiten dar de manera general un enfoque global del sistema, mediante estas vistas, los interesados pueden profundizar en detalles del sistema de manera amigable y fácil de comprender.