

Universidad de las Ciencias Informáticas

Facultad 2



Módulo de Reportes del Sistema Integral de Análisis de Información

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: José Luis Chamizo Hechavarría

Tutores: Ing: Martha Mesa Silva
Ing: Husseyn Despaigne Reyes

Co-tutor: MSc. Jorge Luis Olmedo Flores

“Año 55 de la Revolución”

La Habana, Julio 2013

Pensamiento

*Vivir no es sólo
existir,
sino existir y crear,
saber gozar y sufrir
y no dormir sin soñar.*

*Descansar, es empezar a
morir.*



Gregorio Marañón y Posadillo

DECLARACIÓN DE AUTORÍA

Declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

José Luis Chamizo Hechavarría

Autor

Ing. Martha Mesa Silva

Tutora

Ing. Husseyn Despaigne Reyes

Tutor

MsC. Jorge Luis Olmedo Flores

Co-tutor

AGRADECIMIENTOS

A mis padres María Teresa Hechavarría Cosme y José Higinio Chamizo Riera por su amor y dedicación.

A mis hermanos Risel María Chamizo Hechavarría y Adrián José Chamizo Enamorado ya que son lo más hermoso que tengo actualmente.

A mis abuelos Juana Elena y Manuel de Jesús por ser tan maravillosas personas.

A mis tíos y tías Mariela, Rita, Yuli, Yoni, Manuel, Tony, Marcia, Dani y los demás por su atención para conmigo.

A mis primos y primas Doris, Elianis, Yadira, Yudelmis, Yoandry, Alejandro, Liober y Marlon por ser como hermanos para mí.

A toda mi familia por su atención para conmigo.

A Ramona, Vivian y su esposo, Miguel y Eduardo por ser tan atentos conmigo desde que vine a estudiar en la UCI.

A mis tutores Martica, Yaily, Hussein y Sandy por su dedicación y entrega para conmigo. Por dedicar gran parte de su tiempo en lograr que pudiera graduarme como ingeniero en esta Universidad.

A equipo de desarrollo del SIAA por los maravillosos años que pasamos juntos. A Lilian, Martica, Norbelis, Yaily, Sandy, Eric, Hussein, Olmedo, Arián, Dany, Denis, Jova, Pedro y Rogelio.

A mis amigos de toda la vida principalmente a Idalberto y su familia ya que me han hecho sentir parte de su familia en cada momento.

A mis vecinos que siempre han estado conmigo en cada momento de mi vida, por su preocupación, dedicación y confianza.

A mis compañeros de aula y de escuela en cada una de mis etapas de enseñanza por ser mi apoyo.

A mis compañeros de apartamento por la buena convivencia.

A todos mis profesores por el conocimiento y las enseñanzas.

A cada una de las mujeres que me han soportado en cada momento de mi vida.

A todas las personas con las que he compartido.

Gracias a todos por estar presentes en mi vida, por hacer que valiera la pena vivirla, por tener un motivo para seguir adelante.

DEDICATORIA

A la persona que hizo que este trabajo fuera posible, a quien dedicó sus últimos años de vida en lograr que yo fuera alguien algún día. A la persona que luchó conmigo en los momentos más difíciles, a la persona que me felicitaba en las victorias y me alentaba a continuar en las derrotas. A la persona que siempre confió en mí sin importar el problema al cual me enfrentara. A la persona más fuerte que he conocido en la vida, que no le temía a nada, que siempre daba el paso al frente, a la que todos respetaban y admiraban, aquella que hasta en sus últimos momentos supo repartir alegría, felicidad, amor y esperanza. A la persona que aunque no se encuentra físicamente entre nosotros, siempre ha estado a mi lado. Para ella las gracias por hacer de mí quien soy hoy.

Para mi abuela:

Olivia Cosme Salas

RESUMEN

El presente trabajo está basado en la proposición de la realización de un módulo con el objetivo del manejo de los reportes por parte del personal de antifraude de ETECSA. En la empresa existen vías para obtener los reportes deseados de los diferentes sistemas por lo que se encuentra conformada la entidad, pero los mismo son muy complejos por no presentar un formato único por el cual se pueda obtener la información de forma comprensible y resumida, por tal motivo se hace necesario el desarrollo del módulo de reportes del SIAI. El módulo permite al usuario que se encuentre autenticado en el sistema que pueda obtener la información de forma comprensible y resumida mediante una interfaz agradable, además se le da la posibilidad de generar sus propios reportes de manera dinámica, por lo que hace que sea más agradable ante la vista del usuario debido a que el módulo se encuentra orientado a su trabajo. El sistema fue desarrollado empleando un grupo de técnicas, metodologías y tecnologías las cuales son explicadas con más detalles en el presente documento.

Palabras claves: Calidad, Información, Reportes.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1. Introducción	4
1.2. Principales conceptos asociados al dominio del sistema.	4
1.3. Generadores de Reportes.....	5
1.3.1. Herramientas a nivel mundial.....	5
1.3.2. Herramientas a nivel nacional.....	9
1.3.3. Análisis crítico de la situación actual.....	10
1.4. Metodología de desarrollo.....	10
1.5. Técnica de modelación.....	11
1.6. Lenguajes.....	12
1.6.1. Lenguaje de modelado.....	12
1.6.2. Lenguaje de programación.....	13
1.7. Tecnologías y herramientas.....	13
1.7.1. Herramientas de modelación.....	13
1.7.2. Herramientas de desarrollo.....	14
1.7.3. Servidor de aplicaciones.....	15
1.7.4. Servidor de base de datos.....	15
1.7.5. Marcos de trabajo.....	16
1.8. Conclusiones.....	17
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	18
2.1. Introducción.....	18
2.2. Propuesta del sistema.....	18
2.3. Modelo del negocio.....	18
2.3.1. Descripción de los procesos de negocio.....	19
2.4. Requisitos de software.....	21
2.4.1. Requisitos funcionales.....	21

2.4.2.	Requisitos no funcionales	22
2.5.	Definición de los casos de uso.....	25
2.5.1.	Definición de los actores del sistema.....	25
2.5.2.	Diagrama de caso de uso del sistema.....	26
2.5.3.	Descripción textual de casos de usos.....	26
2.6.	Conclusiones.....	40
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA		41
3.1.	Introducción.....	41
3.2.	Arquitectura del sistema.....	41
3.2.1.	Propuesta de arquitectura del sistema.....	41
3.2.2.	Patrón MVP de la capa de presentación.....	43
3.2.3.	Patrones de diseño.....	43
3.2.3.1.	Patrones de diseño empleado en el negocio.....	44
3.2.3.2.	Patrones de diseño empleado en la presentación.....	45
3.3.	Diagramas de interacción.....	48
3.3.1.	Diagrama de Paquetes.....	48
3.3.2.	Diagrama de clases del Diseño.....	48
3.4.	Modelo de datos.....	49
3.4.1.	Modelo Lógico de Datos.....	50
3.4.2.	Modelo Físico de Datos.....	51
3.5.	Diagrama de despliegue.....	52
3.5.1.	Descripción de los nodos del diagrama de despliegue.....	52
3.6.	Diagrama de componentes.....	53
3.7.	Conclusiones.....	54
CAPÍTULO 4: PRUEBAS		55
4.1.	Introducción.....	55
4.2.	Pruebas realizadas al sistema.....	55
4.2.1.	Pruebas de unidad.....	57

4.2.2. Pruebas de integración.....	58
4.2.3. Pruebas de sistema.	64
4.3. Conclusiones.....	64
CONCLUSIONES.....	65
RECOMENDACIONES.....	66
BIBLIOGRAFÍA.....	67
ANEXOS.....	70
Anexo 1.....	70
Anexo 2.....	¡Error! Marcador no definido.
Anexo 3.....	¡Error! Marcador no definido.
Anexo 4.....	¡Error! Marcador no definido.

INTRODUCCIÓN

La “Información” es un recurso vital para las empresas a nivel mundial que actualmente realizan grandes esfuerzos para tener un alto nivel de competitividad y posibilidades de desarrollo, que garanticen la supervivencia en el mercado. La importancia de medir la productividad y rendimiento de las diferentes áreas de una empresa conlleva al crecimiento exponencial de la información generada, por lo que controlar la misma se convierte en una necesidad y preocupación de cualquier entidad. En los últimos años se han incorporado numerosos avances tecnológicos para el acceso a los datos relevantes de manera frecuente y oportuna. Estos facilitan la realización de tareas mecánicas y rutinarias, evitando los errores humanos con el fin de elevar la calidad.

Los reportes informáticos son una excelente manera de organizar y presentar información que se encuentra contenida en una base de datos. Su función principal es aplicar un formato definido y mostrar los datos mediante un diseño atractivo, que sea fácil de interpretar por los usuarios. De esta forma se proporciona a los datos una mayor utilidad, ya que por lo general están acompañados de gráficos, diagramas y tablas de contenido. Es fundamental señalar que los resultados obtenidos a través de un reporte pueden ser manipulados pero sin alterar la fuente, ya que depende de otros aspectos o áreas del sistema. De esta forma, los reportes reflejan si se han cumplido los objetivos y metas de una empresa y en base a ello poder tomar las mejores decisiones que garanticen el crecimiento de las organizaciones.

La Empresa de Telecomunicaciones de Cuba S.A. es una organización de gran prestigio, que tiene como misión brindar servicios de telecomunicaciones que satisfagan las necesidades de los usuarios y la población. Desde su fundación, ETECSA ha ganado en eficiencia y compromiso con sus usuarios, resultado del buen funcionamiento de diversas áreas que integran la empresa. Sin embargo, actualmente la empresa es afectada por personas que hacen uso de los servicios a través de medios ilegales sin la intención de pagar por ellos, afectando no solo a la organización sino al país. En este sentido, los analistas de fraude para llevar a cabo las investigaciones a través del SIAI¹, necesitan de información que actualmente no se presenta en un formato adecuado, lo que provoca que resulte difícil de interpretar por

¹ Sistema Integral del Análisis de la Información (SIAI), desarrollado en proyecto de colaboración entre la Universidad de las Ciencias Informáticas (UCI) y la Empresa de Telecomunicaciones de Cuba S.A (ETECSA).

los analistas y afecte de forma considerable el tiempo previsto de las investigaciones. De igual forma, esta área de la empresa no tiene posibilidades de conocer mediante reportes personalizados, de fácil comprensión, si han disminuido o aumentado los indicios de comportamientos anómalos en las telecomunicaciones a partir de las investigaciones realizadas.

Partiendo de la problemática expuesta anteriormente se presenta el siguiente problema a resolver: ¿Cómo garantizar la obtención de la información estadística de forma comprensible y personalizada para contribuir a las investigaciones de fraude realizadas por los analistas en ETECSA?

A partir del problema planteado se establece como **objeto de estudio**: sistemas de información para la obtención, manipulación y procesamiento de los datos.

Para satisfacer el problema descrito se presenta como **objetivo general** del trabajo de diploma: “Desarrollar una aplicación para la web que permita obtener de forma comprensible y personalizada los reportes solicitados por los analistas y directivos de ETECSA desde el SIAI”.

Se establece como **campo de acción**: sistemas de información con tecnología Web para la gestión de reportes asociados al área de fraude en ETECSA.

Para lograr el objetivo trazado se propone como tareas de la investigación:

1. Estudio comparativo de sistemas de reportes para identificar las tendencias actuales.
2. Definición de la metodología, tecnologías y herramientas para el desarrollo de la aplicación.
3. Modelación de los procesos de negocio a tener en cuenta para el desarrollo de la aplicación de reportería.
4. Definición de los requerimientos funcionales y no funcionales del Módulo de reportes.
5. Realización de pruebas de integración entre el marco de trabajo GWT y las herramientas de reportería seleccionadas.
6. Diseño y desarrollo de la aplicación web que permita de forma embebida generar dinámicamente los reportes necesarios para la investigación.
7. Implementación de un conjunto de reportes previamente definidos para el trabajo de los analistas.
8. Configuración de los formatos de exportado de cada reporte a PDF, EXCEL y otros.
9. Configuración de los formatos de salida gráfica a pastel o barra de forma embebida en el aplicativo Web.
10. Diseño del modelo de datos lógico y físico perteneciente al área de reportes dinámicos.
11. Realización de pruebas de funcionalidad al sistema.

Al concluir la presente investigación se espera obtener una aplicación para la web que permita generar, obtener, graficar y exportar información a diferentes formatos, que sea de utilidad para ETECSA en la detección de actividades anómalas y a su vez permita tomar las decisiones correctas de acuerdo a los resultados obtenidos.

Para el desarrollo de la investigación se utilizaron diferentes métodos científicos dentro de los que se incluyen los métodos teóricos y empíricos que se muestran a continuación:

Métodos teóricos:

1. “Histórico-Lógico”: Empleado para establecer un estudio de los diferentes comportamientos y características específicas que brinda cada una de las herramientas de reportería, en la determinación y selección de la propuesta específica de las herramientas a utilizar.
2. “Analítico-Sintético”: Permitió el estudio de diferentes bibliografías con el fin de sintetizar, acotar las búsquedas, establecer cuáles fueron las fuentes fiables de información, y llegar a conclusiones aceptadas y bien balanceadas sobre el tema en cuestión.
3. “Modelación”: Permitió establecer esquemas, modelar ideas de forma gráfica, además de posibilitar la utilización de diagramas para expresar información de manera organizada. En el proceso de análisis y diseño el modelaje se convierte en una herramienta indispensable para el entendimiento de las diferentes etapas del desarrollo del software.

Método Empírico:

1. “Entrevista”: Para el proceso de entendimiento del negocio y los requerimientos que el sistema deberá satisfacer, además del proceso de familiarización con el personal capacitado sobre los temas a tratar.

Este trabajo se encuentra estructurado por 4 capítulos, los cuales se mostrarán a continuación:

Capítulo 1: Fundamentación Teórica. Se exponen los conceptos, las técnicas, tecnologías y tendencias de la propuesta presentada.

Capítulo 2: Características del sistema. Se ofrece una visión práctica del sistema, los requisitos funcionales y no funcionales, además de una propuesta del sistema.

Capítulo 3: Diseño e Implementación del sistema. Se presenta una vista interna del sistema, los diagramas de clases del diseño y los elementos empleados en la implementación del sistema con los artefactos generados.

Capítulo 4: Pruebas. Se presentan los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

El presente capítulo tiene como principal objetivo exponer los fundamentos teóricos generales, los cuales son necesarios para tomarlo como punto de partida a la solución del problema mencionado anteriormente. En el mismo se analizarán los principales conceptos relacionados con los “Reportes”, las diferentes tecnologías que intervienen, las características así como las ventajas y desventajas.

1.2. Principales conceptos asociados al dominio del sistema.

Módulo: Componente de un sistema, el cual se relaciona con otros a la hora de la construcción del mismo a fin de hacerlo más sencillo, regular y económico. Todo módulo, por lo tanto, forma parte de un sistema y suele estar conectado de alguna manera con el resto de los componentes. Es construido para ser fácil a la hora de ensamblar, presentar un acoplamiento flexible y realizarles reparaciones sin que el sistema sufra cambios (1).

Módulo (programación): En programación un módulo es una porción de un programa de computadora. Resuelve una o varias de las tareas que tiene que realizar el programa para cumplir con su función u objetivos (2).

Reporte: Informe que se emite o se presenta por un sistema, una entidad o una persona (3).

Reporte de Resultados: En marco de la presente investigación se llamará “Reporte de Resultados” a la capacidad con la que tiene que contar un sistema para que en un período determinado o de forma manual, pueda emitir los resultados estadísticos con los que cuenta hasta el momento la empresa.

Diseño de reporte: En el marco de la presente investigación se llamará “Diseño de reportes” a la plantilla o modelo que permite establecer la estructura que tomará al final los reportes que se generen. Su apariencia se encuentra conformada por varios componentes, estos se encuentran sobre distintas secciones delimitadas, las cuales definen el comportamiento a la hora de generar los componentes que se encuentran en ellas contenidos.

Personalizar: En el marco de la presente investigación se definirá “Personalizar” como la capacidad de configurar las opciones que ofrece una aplicación para acomodarla según la preferencia de los usuarios.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

Mediante la personalización el sistema tiene que ser capaz de responder a las necesidades particulares de los usuarios.

Con esto se puede decir entonces que un “Módulo de Reportes”, es el componente de un sistema automatizado, encargado de la emisión de los reportes de forma periódica (diario, semanal, quincenal, mensual, anual). Estos son capaces de gestionar gran cantidad de información con gran rapidez y confiabilidad en los datos. Los reportes personalizados son aquellos que se obtienen partiendo de las necesidades y exigencias de los usuarios o sea, va a estar constituido por los aspectos que sean de interés a los usuarios. De forma general el módulo y la personalización representan las piezas por la cual los usuarios del sistema pueden obtener y controlar la información mediante sus intereses.

1.3. Generadores de Reportes.

Los generadores de reportes son una potente herramienta que tiene como principal objetivo la creación de reportes. Presentan una interfaz que permite diseñar de manera instantánea y con un mínimo conocimiento y esfuerzo, la manera en la cual se desea que se muestre la información. Generan un grupo de sentencias SQL las cuales sirven para realizar las consultas a las diferentes tablas de una base de datos.

1.3.1. Herramientas a nivel mundial.

Microsoft SQL Server 2005 Reporting Services: Visor de informes desarrollado por Microsoft para el trabajo de reportes en la plataforma de Windows. Permite la toma de decisiones en tiempo real en toda la empresa, además de poder generar informes operacionales para la impresión y visualización en el navegador así como la exportación de los datos. Para un correcto funcionamiento de esta herramienta es necesario que el usuario tenga instalado en su sistema las actualizaciones de Microsoft SQL Server. Las actualizaciones tienen que hacerse de manera manual lo que trae consigo que el programador se encuentre en cada momento en que la empresa necesite realizar algún cambio en el flujo de datos que corre en su base de datos (4). Para su correcto funcionamiento el sistema tiene que contar con un conjunto de requisitos los cuales permiten su correcto desempeño. Por tal motivo es necesario tener instalado el Microsoft Office SharePoint Server 2007 o Windows SharePoint Services 2007 con compatibilidad con la programación en .NET, el Microsoft Windows Server 2003 con el Service Pack más

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

recientes (sólo x86), configurado para funcionar como un servidor de aplicaciones con ASP.NET² 2.0 habilitado, el SQL Server 2005 Reporting Services Service Pack 2 (SP2: por sus siglas en inglés), Microsoft Internet Explorer 6.0 con Service Pack 2 o posterior, siendo Microsoft Internet Explorer 7.0 muy recomendable. Entre sus principales características se pueden mencionar que los informes pueden ser exportados en los formatos HTML, PDF, TIFF, Excel, CSV, permite a los programadores integrar o extender procesamiento de datos e informes en aplicaciones personalizadas, los informes incluyen características interactivas basadas en la Web: informes de varios niveles de detalle, que permiten la navegación por distintas capas de datos; los informes con parámetros, que admiten el filtro de contenido en tiempo de ejecución; o los informes de formato libre, con diseños verticales, anidados o adyacentes, proporciona la capacidad de ver los informes, explorar páginas además de ampliar o disminuir diferentes áreas y permite administrar informes, modelos y orígenes de datos.

Active Reports: Diseñador de reportes desarrollado por Microsoft para el trabajo en la plataforma Windows. En tiempo de diseño el programador tiene que trabajar directamente desde el visual estudio. Es un complemento del lenguaje de programación Visual Basic, agregando la posibilidad de crear reportes desde la aplicación de manera muy sencilla. Incluye la funcionalidad del diseño de informes para el usuario final, siendo uno de los pocos con esta característica. Sirve para el trabajo en aplicaciones web y de escritorio, presenta una amplia gama de formatos de exportación como son PDF, Excel, RTF y TIFF además de incluir una vista previa de los informes. Soporta la conexión a las diferentes bases de datos estándares permitiéndole al usuario la opción de modificar los datos consultados (5). Entre sus principales características se pueden mencionar que es totalmente administrado ya que no presenta dependencias de aplicaciones de terceros, incluye un control de gráficos que admite tipos comunes de gráficos en 2D y 3D y proporciona prestaciones gráficas avanzadas, además de exportación nativa a diversos formatos de imagen. Incluye un control de visor de informes que admite ampliar o disminuir diferentes áreas, vista previa de informes, vistas divididas y de múltiples páginas, así como un panel de índice de contenidos, miniaturas, búsquedas de texto, anotaciones y personalización de la barra de herramientas.

JasperReports: Poderosa herramienta de creación de informes desarrollado por JasperSoft, siendo uno de los más populares motores de informes de código abierto. Está escrito completamente en Java y es

² ASP.NET: Framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y presentar los documentos con precisión de píxel que se puede ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel y Word. Puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. Otra de las funcionalidades que presenta es la generación de sub-reportes mediante los cuales se pueden diseñar y manipular reportes con grandes complejidades en su diseño. Es soportado en diferentes sistemas operativos siempre que exista una Máquina Virtual de Java para ese entorno. Cuenta con un grupo de librerías de apoyo las cuales le permiten un mejor diseño de los reportes, agregarles estilos y agregar nuevos formatos de salidas. Permite el paginado lo que hace que su trabajo delante de grandes cantidades de datos sea muy efectivo (6).

Crystal Reports: Sistema desarrollado por Microsoft diseñado para el trabajo en Windows solamente en la versión XP y Windows Server 2003. Se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic. Su versión para Java simplifica y agiliza el proceso de acceso a los datos, además de permitir el formateo e integración en las páginas Java Server Pages (JSP) y está disponible para utilizarlo directamente desde el entorno integrado de desarrollo (IDE) Eclipse. El software para Eclipse brinda una potente solución de generación de reportes basada en Java, que incluye un diseñador de reportes y un motor de tiempo de ejecución. Permite a los desarrolladores crear y modificar reportes, además de integrar el motor de ejecución en aplicaciones Java para clientes grandes y pequeños (7). Entre sus principales funcionalidades se puede mencionar que permite el trabajo en su entorno de diseño desde Eclipse o Crystal Reports XI, presenta una interfaz de diseño intuitiva para tener acceso a los datos y aplicar formato a los reportes con facilidad e incorpora reportes de manera sencilla y directa en aplicaciones grandes o basadas en la Web.

Agata Reports: Generador de reportes multiplataforma, es una herramienta de consulta y generación de gráficos que se conecta a diferentes Bases de Datos como son PostgreSQL, MySQL y Oracle. Permite exportar reportes en formatos como son HTML, PDF, XML y otros, además de permitir la definición de niveles de datos, subtotales y totales para el informe. Permite crear documentos, como cartas y conjugar dinámicamente con los datos provenientes del reporte, así como crear etiquetas de direccionamiento y hasta generar un diagrama Entidad-Relación (ER: por sus siglas en español) completo a partir de su banco de datos (8). La herramienta permite extraer los valores más significativos de una base de datos

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

permitiendo definir niveles de informe, así como aplicar un conjunto de funciones a cada nivel, como pueden ser suma, máximo, mínimo, contar y promedio. Genera gráficos de línea y de barras y permite la generación de reportes de manera dinámica en tiempo de ejecución, preguntando al usuario por los valores deseados en la generación del informe.

BIRT: Proveniente del inglés Business Intelligence and Reporting Tools es un proyecto de software de código abierto que proporciona capacidades de creación de informes y de inteligencia de negocio para clientes pesados (fat clients) y aplicaciones web, especialmente aquellas basadas en Java y Java EE. Es un proyecto de software de alto nivel dentro de la Fundación Eclipse ³(Eclipse Foundation), un consorcio de proveedores de la industria del software sin ánimo de lucro y una comunidad de código abierto. Los objetivos del proyecto son cubrir un ancho rango de necesidades de creación de informes dentro de una aplicación típica, abarcando desde informes operacionales o de empresa hasta procesamiento multi-dimensional analítico en línea (OLAP). Inicialmente, el proyecto se ha enfocado en capacidades que permitan a los desarrolladores de aplicaciones diseñar e integrar fácilmente informes dentro de aplicaciones. El proyecto recibe soporte desde una activa comunidad de usuarios en BIRT Exchange⁴ y desarrolladores de Eclipse.org Proyecto BIRT. Es un sistema multiplataforma teniendo como requisito principal de que el desarrollador debe trabajar con el eclipse, ya que este es el IDE de programación necesario para diseñar reportes con la herramienta (9). Entre sus principales características se destacan que permite exportar en varios formatos como son HTML, PDF, WORD y XLS, los diseños generados se encuentran en formato XML los cuales pueden ser empleados por otras librerías de desarrollo de reporte en tiempo de ejecución de la aplicación. Presenta un diseñador de informes visuales que puede ser trabajado desde el mismo Eclipse IDE con el fin de evitar el trabajo con varias herramientas al mismo tiempo para el desarrollo de los informes. Los componentes desarrollados pueden ser trabajados desde cualquier entorno de Java, además permite el filtrado de los datos y la personalización del informe.

³ Fundación Eclipse: Grupo de empresas desarrolladoras de software lideradas por IBM, encargadas de liderar el desarrollo de Eclipse, tanto el IDE como la plataforma de código abierto.

⁴ BIRT Exchange: Comunidad de desarrolladores para BIRT

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

1.3.2. Herramientas a nivel nacional.

Los generadores de reportes que se encuentran a nivel nacional, tienen el objetivo principal de resolver problemas específicos presentados en una entidad que necesite obtener reportes informáticos sobre sus datos, destacándose principalmente del desarrollo de las herramientas en la Universidad de las Ciencias Informáticas. Esto trae consigo que las herramientas que se desarrollen no sean de carácter general, sino que resuelvan solamente los problemas detectados internamente por los especialistas del equipo de desarrollo. De por sí, estas no son herramientas hechas puramente por la universidad, sino que toman el código fuente de otras disponibles en internet y tratan de integrarlas a su software quitándole las funcionalidades que no necesiten y programando otras en caso de ser necesario. Entre las herramientas con que se cuenta están:

GDR: Generador Dinámico de Reportes por sus siglas. Desarrollado en centro DATEC de la facultad 6. Entre sus principales características se encuentran que es una aplicación para la web, la cual necesita de la instalación de un servidor de aplicaciones web Apache, un sistema gestor de bases de datos PostgreSQL y un servidor de gráficos preferentemente el “*Rich Chap Server*”. Su código fuente es el de la herramienta PHP Reports y el trabajo con grandes cantidades de volúmenes de datos es deficiente al no tener implementado una solución de paginado para el trabajo con los mismos. Es un sistema multiplataforma y muy empleado por alguno de los servicios internos de la institución. El lenguaje de programación que se emplea para trabajar es php, y se cuenta con un grupo de desarrollo, trabajando en generar plantillas en formatos jrxml para lograr una integración con la librería Jasperreports y poder trabajar además con Java.

GRJ: Generador de Reportes Java por sus siglas. Desarrollado por el proyecto alasGrato en la facultad 6. Entre sus principales características se encuentra que es una aplicación de escritorio, desarrollado mediante la integración de iReports y Jasperreports, con el fin de diseñar y generar reportes desde la misma aplicación desarrollada. Se relaciona con diferentes bases de datos como PostgreSQL, MySQL y Oracle. Actualmente el proyecto se encuentra desintegrado, lo que trajo consigo la pérdida de toda la información que poseían al igual que el código fuente de la herramienta. No se siguió trabajando sobre la herramienta por lo que se encuentra muy atrasada con respecto a nuevas funcionalidades que se le han integrado al código fuente tanto del Jasperreports como el de iReports.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

Olympia: Aplicación desarrollada sobre el framework Symfony y escrita en PHP. Es multiplataforma y soporta imágenes, gráficas y sub-reportes, así como varios orígenes de datos. Proporciona a los usuarios, entre otras opciones, agilizar la toma de decisiones, generar reportes en varios formatos y con gran variedad de opciones en su diseño, marcando una diferencia entre los reportes tradicionales y los reportes dinámicos, objetos de este producto. Está compuesto por varias aplicaciones entre las que se encuentran el Visor de reportes, Diseñador de modelos y el Diseñador de reporte. Aunque permite abstraerse en parte de los conocimientos relacionados con los gestores de bases de datos, el usuario aún debe poseer conocimientos básicos de los mismos. Además, su entorno de trabajo está estructurado de forma que es difícil guiarse en la creación y generación de reportes. Las consultas que soporta la misma son las de SQL estándares, soportando como lenguaje de programación a PHP, no presenta buen rendimiento dentro de las bases de datos Oracle y necesita de un servidor web (Apache) para poder entrar en funcionamiento (26).

1.3.3. Análisis crítico de la situación actual.

La solución que se pretende aportar con el presente trabajo de diploma, es una “solución a la medida”, ya que por cuestiones de seguridad el cliente (ETECSA) solicita la realización de un generador de reportes propio de la entidad y que se integre al SIAI. El análisis de las herramientas anteriormente expuestas, permitió tener una idea global de las tendencias actuales en cuanto a generadores de reportes. Además permitirá aplicar un conjunto de buenas prácticas a la solución, como es el caso de la utilización de librerías y funcionalidades de dichas herramientas.

1.4. Metodología de desarrollo.

RUP: Rational Unified Procces por sus siglas en inglés. Metodología estándar utilizada para él para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. No es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí. Es una forma disciplinada de asignar tareas y responsabilidades en el desarrollo del software definiendo quien, como, cuando y que tiene que hacerse en un proyecto (12).

RUP personalizado para el Proyecto SIAI.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

RUP originalmente trabaja con 4 fases y 9 flujos de trabajo. Se trabajará con sus fases pero con la diferencia de que Elaboración y Construcción se tomarán como una sola, es decir que las fases serían 3 y quedarían de la siguiente forma:

- Inicio
- Elaboración-Construcción
- Transición

La idea de unir estas fases está dada por la necesidad de trabajar los flujos de diseño e Implementación a la par, pues queda demostrado en la práctica que ambos flujos dependen una del otro y al tener ambos mayor peso en las fases mencionadas anteriormente se decidió unir las mismas. Dentro de cada fase se desarrollan iteraciones que al final de un período de tiempo relativamente corto producen un resultado para el cliente. Dentro de cada iteración se usan algunos de los flujos que propone RUP, los cuales se numeran a continuación.

- Modelación del negocio
- Requerimiento
- Diseño e Implementación (los cuales se llevarán a la par como se mencionaba anteriormente).
- Prueba

1.5. Técnica de modelación.

IDEFO: Es un método diseñado para modelar las decisiones, acciones y actividades de una organización o sistema (11). Técnica para representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades. Permite modelar gráficamente sistemas de diferente propósito y a cualquier nivel de detalle. Basada en un estándar con especificaciones precisas y rigurosas es empleado para representar lo que se hace en una empresa. Es una técnica sencilla pero muy poderosa, fomenta el trabajo en equipo de manera disciplinada y coordinada. Aunque esto no significa que el modelo refleje el consenso del pensamiento del equipo. Es soportado por diferentes herramientas de cómputo, pudiéndose combinar con otras metodologías para agregar secuencia y sincronización de actividades. Entre sus ventajas se pueden mencionar:

- Técnica que obliga a mantener una jerarquía de relaciones entre las actividades / funciones descritas.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

- Facilita un análisis en profundidad de las entradas y salidas, así como los elementos de control y recursos de cada actividad.
- Es muy adecuado en el diseño de sistemas complejos y dinámicos.
- Permite un correcto control de los detalles comunicados en cada nivel (3 a 6 cuadros de funciones en cada nivel).
- Contexto acotado: sin omisiones o detalles dejados al azar.
- Conectividad del diagrama de interferencia: número de nodos, número de cuadros, detalle de expresiones con siglas.
- Etiquetas únicas y títulos: para evitar duplicados.
- Reglas de sintaxis para gráficos: cuadros y flechas.

Por las características mencionadas anteriormente se selecciona esta técnica de modelación para realizar el flujo de modelamiento de negocio, destacando principalmente la combinación de IDEF0 y RUP y la manera en que se describe el proceso de negocio detalladamente a través de su empleo.

1.6. Lenguajes.

1.6.1. Lenguaje de modelado.

Lenguaje Unificado de Modelado (UML por sus siglas en inglés): Tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, hasta la implementación y configuración con los diagramas de despliegue. Su necesidad radica principalmente en el hecho de que es un lenguaje que permite desarrollar un plan bien analizado que pueda ser comprensible tanto como para el cliente como para los o el realizador explicarlo, analizarlo y desarrollarlo (10). Sus objetivos son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

Se selecciona a UML por las facilidades que brinda el mismo como son la visualización, especificación, construcción y documentación de un sistema de software. Por el estándar que ofrece para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como son los procesos del negocio y las funciones del sistema, y aspectos concretos como son expresiones del lenguaje de programación, los esquemas de base de datos y los componentes de software reutilizables. Vale destacar que no se puede comparar con la programación estructurada, ya que no es programación lo que se realiza, sino que se diagrama la realidad de la utilización de un requerimiento. Además la metodología seleccionada (RUP) emplea UML como lenguaje de modelado.

1.6.2. Lenguaje de programación.

Java: Es un lenguaje de programación de alto nivel y además orientado a objetos. Se rige por una filosofía la cual se encuentra dividida varios objetivos principales destacándose principalmente:

- Deberá permitir la ejecución de un mismo programa en diferentes sistemas operativos.
- Deberá incluir por defecto soporte para trabajo en red.
- Deberá diseñarse para ejecutar código en sistemas remotos de forma segura.
- Deberá ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos como C++.

Su diseño, robustez, el respaldo de la industria y su fácil portabilidad han hecho que este lenguaje sea uno de los que mayor crecimiento y amplitud haya tenido dentro de la industria de la informática a nivel mundial. Es por esto que lo podemos ver en dispositivos móviles y sistemas empujados, en navegadores web, en sistemas de servidor, en aplicaciones de escritorio y en muchas plataformas. Las principales características de este lenguaje es el nivel de seguridad que presenta en el desarrollo de aplicaciones web, y el poder emplearlo desde cualquier sistema. El empleo de este lenguaje posibilita el trabajo con GWT el cual no es más que una librería que permite el desarrollo de las interfaces para proyectos web desarrollados en Java y actualmente es empleado por los desarrolladores del proyecto SIAI.

1.7. Tecnologías y herramientas.

1.7.1. Herramientas de modelación.

Microsoft Visio: Software de dibujo vectorial para Microsoft Windows. Las herramientas por lo que se encuentra conformado permite el desarrollo de diagramas de oficinas, base de datos, diagrama de flujos de programas, UML y más que permiten iniciar al usuario en los lenguajes de programación. Facilita la

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

visualización, comunicación, análisis de información bastante compleja, es fácil de actualizar y permite integrar diagramas con la información de varias fuentes. Permite la conexión entre elementos de manera dinámica, por lo que los diagramas se distribuyen y alinean también de manera dinámica (13). Permite la representación gráfica de la técnica de modelado IDEF0, por lo que facilita realizar los diagramas necesarios para modelar el toda la información recopilada en las entrevistas realizadas.

Visual Paradigm 5.0: Visual Paradigm para UML es una herramienta profesional la cual soporta completamente el ciclo de vida del desarrollo del software. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Propicia un conjunto de ayudas para desarrollar los programas desde la planificación hasta la generación del código fuente de los programas y la documentación (14). Entre sus principales características se pueden mencionar:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Soporta aplicaciones Web.
- Generación de código para Java y exportación como HTML.
- Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.

Por las comodidades que brinda esta herramienta es seleccionada para el diseño de los artefactos que son generados en cada etapa del desarrollo del software, lo que traería consigo un excelente resultado los cuales hagan una descripción de los diferentes procesos de la entidad.

1.7.2. Herramientas de desarrollo.

Eclipse Índigo: Entorno de Desarrollo Integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”. Incluye las herramientas de desarrollo de Java, por lo que cuenta con un compilador interno y un modelo de los archivos fuentes de Java, un correcto análisis del código que se encuentra generando el desarrollador. Provee al desarrollador un grupo de frameworks los cuales son muy ricos para el trabajo con aplicaciones tanto gráficas como web, lo que además de poder ser empleados con otros plugins, se puede hacer uso de su interfaz gráfica la cual es personalizable y muy profesional. El empleo de esta herramienta ofrece ventajas como el empleo de

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

módulos para proporcionar toda su funcionalidad al frente de la Plataforma de Cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.

Por todas las características y ventajas que presenta este entorno es considerado como la mejor opción para el desarrollo del proyecto. Es una herramienta que además de ser multiplataforma, tiene una buena comunicación con el desarrollador, lo que permite que sean tratados los errores cometidos por un programador, brindándole principalmente una serie de recomendaciones y ayuda para obtener una mayor información (15).

1.7.3. Servidor de aplicaciones.

JBoss 7.1.0: Proyecto de código abierto el cual se encuentra programado completamente en Java, por lo que puede ser empleado en cualquier sistema operativo que lo soporte. Tiene una implementación de todos los paquetes de servicios brindados por J2EE como es el caso de EJB y sobre todo el mismo posee una licencia libre, lo que permite que pueda ser empleado sin ningún costo. Entre sus principales características se destacan el poseer código abierto, poseer una estructura modular lo que proporciona aislamiento de la aplicación real, confiable para las empresas, posee servicios que lo asisten para que pueda interactuar o comunicarse con diferentes aplicaciones y sobre todo lo que más destaca es su habilidad para reaccionar y adaptarse sin perder calidad alguna a la hora de manejar el crecimiento continuo de trabajo de manera fluida. Al ser inicializado los servicios inician también con el objetivo de eliminar las esperas innecesarias. Emplea un grupo de módulos para aislar las clases que no son empleadas por la aplicación y solo son visibles en caso de que sean referenciadas, lo que esto aumenta el nivel de seguridad hacia la aplicación. Permite una administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portales y la persistencia de los datos (16).

Se selecciona esta herramienta por el nivel de seguridad que promete a los clientes, su respuesta ante diferentes situaciones que puedan presentarse, la estructura que presenta y la gestión de memoria.

1.7.4. Servidor de base de datos.

Oracle 11g: Ha sido desarrollada principalmente para que los clientes aborden los retos que se derivan de su entorno empresarial cada vez más competitivo y cambiante, el rápido crecimiento de los volúmenes de datos y la necesidad de ofrecer un mejor servicio. El desarrollo de sus funcionalidades garantiza un alto

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

rendimiento, alta seguridad, fiabilidad y escalabilidad. Permite realizar pruebas de cambio en aplicaciones mediante una simulación de las cargas reales de los usuarios. Se considera que es uno de los sistemas más completos por su soporte de transacciones, el soporte multiplataforma y su estabilidad. Permite la gestión de todo tipo de datos como son las imágenes, ficheros de texto, objetos en 3D, archivos XML entre otros. Permite un alto rendimiento en las operaciones de lectura y escritura además de permitir una gestión segura de todo su contenido (17).

Se selecciona este servidor por ser comprensiva para almacenes de datos y la inteligencia del negocio, por combinar escalabilidad, rendimiento, integración embebida, calidad de datos y capacidades de análisis.

1.7.5. Marcos de trabajo.

Ext GWT⁵(GXT): Es un framework desarrollado por Google el cual tiene como principal función la transformación de un código Java en HTML y JavaScript. Se encuentra dividido en dos modos, el modo de desarrollo el cual ejecuta a la aplicación como código bytecode de java dentro de la Máquina Virtual de Java, y el modo web el cual ejecuta la aplicación como código JavaScript y HTML puro, compilados a partir del código Java. El proyecto SIAI se encuentra trabajando en el modo web, ya que entre sus objetivos se encuentra el desarrollo de una aplicación web que permita realizar un análisis total de la información con la que tienen que trabajar los analistas del fraude en ETECSA, además que este modo es el que se suele emplear para el despliegue del sistema. Es compatible con varios navegadores mediante el desarrollo de un plugin de integración, lo que esto es notorio ya que en su totalidad los navegadores necesitan de código especiales para poder mostrar correctamente los contenidos de las aplicaciones web sin ningún tipo de distorsión en su contenido. Entre sus principales características se puede mencionar que sus componentes gráficos son dinámicos y reusables lo que permite modificar a gusto de desarrollador cada uno de los componentes que necesita para desarrollar su aplicación. Otra de las características es que mantiene un control de las diferentes características del navegador, es de código abierto, el código JavaScript generado puede ser ofuscado para optimizar el rendimiento de la aplicación y el navegador y la existencia de gran cantidad de librerías que permiten ampliar grandemente las funcionalidades de GWT (18).

⁵ GWT del inglés Google Web Toolkit, traducido al español Set de Herramientas Web de Google.

Capítulo 1: “Fundamentación Teórica”

“Módulo de Reportes del SIAI”

EJB: (del inglés: *Enterprise JavaBeans*) Proporcionan un modelo de componentes distribuidos estándar del lado del servidor. Su objetivo es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables (19). En el proyecto SIAI es empleado por los servicios que presta como son el manejo de las transacciones, seguridad, concurrencia, servicios de red, gestión de recursos, persistencia, escalabilidad y sobre todo la adaptación en tiempo de despliegue, lo que esto ahorraría en tiempo y complejidad en la generación de las diferentes clases con las características mencionadas anteriormente.

JPA: (del inglés: *Java Persistence API*) proporciona un estándar para la gestión de los datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de los datos. Para su desarrollo se han tomado varias ideas y conceptos de los principales frameworks de persistencia como es el caso de Hibernate, JDo y Toplink, además de la incorporación de EJB para su trabajo. La relación entre entidades Java y las tablas de la base de datos se realizan mediante anotaciones en las mismas clases de entidad. Permite al desarrollador mapear, almacenar, recuperar y actualizar datos de las tablas de una base de datos en objetos Java y viceversa (20).

1.8. Conclusiones.

En el presente capítulo se realizó un estudio de los generadores de reportes internacionales como nacionales, lo cual permitió tener una visión de las tendencias actuales para aplicarlo a la solución. Además se definió la metodología, técnica y herramientas necesarias, las cuales darán soporte a todo el proceso de desarrollo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción.

En el presente capítulo se exponen los elementos fundamentales del análisis y modelación del Módulo de Reportes del SIAI. Específicamente se establecen los requerimientos funcionales y no funcionales, los actores, casos de uso del sistema y se realizará una descripción de los mismos.

2.2. Propuesta del sistema.

Debido a que no se cuenta con una herramienta, que muestre mediante una interfaz agradable y de forma resumida la información necesaria para los analistas de fraude de ETECSA, es necesaria la creación de una aplicación la cual permita obtener de forma concreta el comportamiento de los sucesos anómalos en las telecomunicaciones y no se afecte considerablemente el tiempo previsto para realizar las investigaciones. El sistema deberá mostrar la información solicitada de forma comprensible y resumida, además de darle la posibilidad al usuario de obtener la información mediante diferentes tipos de gráficas y poderla exportar en varios formatos. La información se deberá obtener de forma directa mediante un grupo de reportes que se encuentren definidos previamente por los analistas de fraude de manera colectiva y también tiene que dar la posibilidad de obtenerla mediante la creación de reportes de manera dinámica respondiendo a intereses individuales surgidos al momento de interactuar con la aplicación. Actualmente el departamento de Dirección de Operaciones de Seguridad (DOPS) de ETECSA, encargado de la gestión de actividades anómalas en las telecomunicaciones, para llevar a cabo su proceso de investigación, desarrolla en conjunto con la Universidad de las Ciencias Informáticas (UCI), el Sistema Integral de Análisis de Información (SIAI). El desarrollo actual del SIAI no cuenta con un generador de reportes, por lo que el módulo a desarrollar en la presente investigación se incorporará al sistema como parte del mismo.

2.3. Modelo del negocio.

Para la descripción del modelo de negocio se hace uso de la técnica de modelado IDEF0 con el objetivo de capturar y modelar de forma detallada el funcionamiento del negocio. Mediante la misma se obtendrán las entradas, los controles, las salidas y los mecanismos a emplear para poder darle solución al Módulo Reportes del SIAI. El negocio a desarrollar es una parte del negocio general del SIAI por lo que se toma

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

como proceso principal “*Gestionar Información del Negocio*” siendo la caja A5 del negocio general. Destacar además que las cajas que poseen una tonalidad diferente al azul claro, se colorearon intencionalmente para indicar que no tiene que ser implementados para realizar el módulo, sino que forman parte de otros módulos. Las entradas (I3, I5), los controles (C2, C5), las salidas (O5, O7, O9) y los mecanismos (M1, M4) son elementos del sistema general por lo que para entenderlas se recomienda consultar el documento “*Modelo del Negocio con IDEF0 v1.0 - Módulo Plataforma Web*”.

2.3.1. Descripción de los procesos de negocio

Proceso Gestionar Información del Negocio:

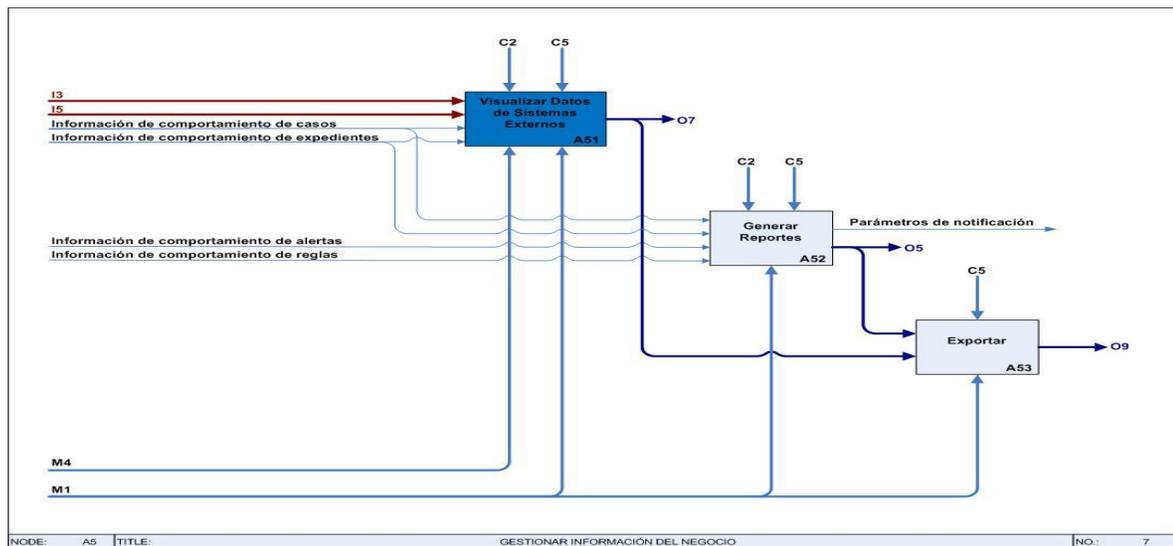


Fig. 1: Diagrama Gestionar Información del Negocio.

Tabla 1: Ficha del proceso A5.

Ficha de Proceso		
Proceso	“Gestionar Información del Negocio”	
Notación:	A5	
Entradas:	I3	Fuentes de datos Externas.
	I5	Número Telefónico.
		Información de comportamiento de reglas.
		Información de comportamiento de alertas.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

		Información de comportamiento de casos.
		Información de comportamiento de expedientes.
Controles:	C2	Tiempo.
	C5	Solicitud.
Salidas:	O5	Reportes Personalizados.
	O7	Información de Sistemas Externos.
	O9	Información Exportada.
		Parámetros de Notificación.
Mecanismos :	M1	Personal ETECSA antifraude.
	M4	Sistemas Externos.
Reglas del Negocio:		<ul style="list-style-type: none"> Las notificaciones solo se generarán cuando sean solicitadas.

Descripción del Proceso

El proceso hace alusión a las actividades que se llevan a cabo para la gestión de Información del Negocio. Parte de las diferentes informaciones de comportamiento de los distintos procesos mencionados anteriormente como son Gestión de las Reglas, Alertas, Casos y Expedientes, las cuales se analizan a solicitud y en ocasiones teniendo en cuenta períodos de tiempo específicos. Como resultado se obtienen reportes personalizados e informaciones de los sistemas externos a conveniencia del Personal de ETECSA de antifraude, que pueden ser además exportados a los formatos deseados.

Proceso Generar Reportes:

El proceso hace alusión a las actividades que se realizan para poder gestionar parte de la información del negocio, recibiendo la información de comportamiento de varios procesos del sistema como son la Gestión de Reglas, Alertas, Casos y Expedientes, los cuales son analizados para realizar reportes de manera dinámica los cuales sean diseñados por el Personal de ETECSA de antifraude o de manera estática, los cuales se definirán en las entrevistas con el cliente a la hora de solicitar requisitos funcionales respecto a reportes que los mismos necesiten que sean estándares para la organización. Influyendo en cada momento los factores de tiempo y solicitud como formas de control. Ver [Anexo 1](#), Tabla 9.

Proceso Exportar:

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

El proceso hace alusión a la posibilidad que se le brinda al usuario de poder exportar la información obtenida de los procesos de *Generar Reportes* y *Visualizar Datos de Sistemas Externos* en diferentes tipos de formatos. Teniendo como control las solicitudes realizadas y como mecanismos el personal antifraude de ETECSA. Ver [Anexo 1](#), Tabla 10.

Proceso Generar Reportes Estáticos:

El proceso hace alusión a la actividad de generar un reporte de manera estática donde el usuario selecciona uno de los reportes definidos con el objetivo de obtener su resultado. Ver [Anexo 1](#), Tabla 11.

Proceso Generar Reportes Dinámicos:

El proceso hace alusión a las actividades que se llevan a cabo para la gestión de los reportes que se generan de manera dinámica. Ver [Anexo 1](#), Tabla 12.

Proceso Graficar Reportes:

El proceso hace alusión a la posibilidad que brinda el sistema de obtener la información de forma gráfica con el objetivo de obtener una mejor comprensión del resultado obtenido. Ver [Anexo 1](#), Tabla 13.

2.4. Requisitos de software.

Los requisitos de software actualmente se han convertido en un elemento clave para el desarrollo de soluciones informáticas. La mala definición de los mismos puede traer consigo el naufragio del proyecto completamente. Mediante los mismos se identifican las necesidades que presenta el cliente, lo que traerá consigo el regimiento de la aplicación. A continuación se muestran los requisitos tanto funcionales como no funcionales que involucran al presente módulo:

2.4.1. Requisitos funcionales

RF1: Generar reportes. Requisitos encargados de generar reportes que han sido definidos previamente.

RF2: Exportar informe. Requisito encargado de exportar la información una vez obtenido el resultado del reporte.

RF3: Graficar. Requisito encargado de graficar la información una vez obtenido el resultado del reporte. Es aplicable solo para algunos de los requisitos que se encuentran dentro del RF1.

RF4: Administrar reporte. Requisitos encargados de gestionar los reportes de manera dinámica en tiempo de ejecución.

RF5: Autenticar. Requisito general del sistema SIAI.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Es importante destacar que los requisitos funcionales 1 y 4 se encuentran conformados por varios requisitos funcionales los cuales se encuentran en la planilla “Especificación de requisitos de software v2.0”, por lo que se recomienda el estudio de dicho documento para una mejor comprensión de los requisitos expuestos. El requisito funcional 1 está conformado por los requisitos desde el 1 hasta el requisito 52 y el requisito funcional 4 está conformado por los requisitos desde el 55 hasta el requisito 58.

2.4.2. Requisitos no funcionales

Seguridad:

RNF Infraestructura de seguridad J2EE.

RNF Seguridad del servidor de aplicaciones.

RNF Seguridad de las interfaces externas.

Usabilidad:

RNF Idiomas distintos, español e inglés.

RNF Habilidades mínimas de usuarios.

RNF Entrenamiento a usuarios.

RNF Documentación de usuarios.

Fiabilidad:

RNF Exactitud. La exactitud debe estar alrededor de un 98%.

RNF Máximo de errores: La cantidad máxima de errores debe promediar en 10 errores por cada mil líneas de código.

RNF Clasificación de errores: Los errores estarán clasificados en un nivel menor, un nivel significativo y un nivel muy significativo.

Eficiencia:

RNF Capacidad: El número de clientes concurrentes es de 300.

RNF Utilización de recursos: La memoria en los servidores del despliegue no debe pasar del 95% de consumo. La capacidad disco en los servidores del despliegue no debe pasar del 80% de consumo. Las comunicaciones por red deben estar disponibles un 99%.

Soporte:

RNF Pruebas de calidad: Al finalizar cada flujo de trabajo que genere un entregable será controlado por ambas partes (ETECSA – UCI) y aceptados en dependencia de su calidad.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

RNF Pruebas de funcionalidad: Como política de seguridad del centro de Telemática de la Facultad 2, se realiza la solicitud al Laboratorio de Seguridad Informática (LabSI) del mencionado centro, para que realice al proyecto SIAI las pruebas de seguridad pertinentes, por ser consideradas de interés para el mismo.

RNF Entorno de pruebas: Debido a que el proceso de desarrollo se realizará en el Departamento de Antifraude de ETECSA se hace necesario que todas las pruebas se ejecuten es dicha entidad por el personal de LIPS y LabSI.

RNF Estándares de codificación: Para reforzar el soporte o mantenimiento del sistema a construir se definieron normas de codificación para el lenguaje de programación Java. (Se especifican en el documento ubicado en EXPEDIENTE DE PROYECTO V.202 (SIAI)\0. GENERALES\PAUTAS DE CODIFICACIÓN\Estándar de codificación Java.doc).

Restricciones de diseño:

RNF Estándares para el desarrollo de GUI: Seguirá el estándar J2EE para aplicaciones empresariales, específicamente MVC. El estándar MVP se utilizará para la integración de GWT en las interfaces gráficas.

Requisitos para la documentación de usuarios en línea y ayuda del sistema:

Se debe poder contar con un manual para usuarios, donde el interesado pueda documentarse sobre el uso de la herramienta, sus características, funcionalidades bien detalladas para una mejor comprensión y uso de la misma de forma eficiente.

RNF Interfaz:

RNF Interfaces de usuario:

Las interfaces de usuarios tienen que ser de manera general amigable y fácil de entender sin mucho entrenamiento por parte de los clientes, emplear colores suaves e información legible que evite cualquier pérdida por parte de un usuario sin conocimiento de la aplicación y el usuario puede acceder a las diferentes opciones sin tener que navegar por muchos puntos intermedios.

RNF Interfaces de Hardware:

Estaciones clientes:

- Disco duro con 10 GB libres o más.
- Memoria RAM de 1 GB o más.
- CPU con microprocesador a 2 GHz o más.

Servidores:

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

- De 8 a 16 procesadores.
- De 24 a 32 GB RAM.
- De 80 a 120 GB disco duro.

Servidor del banco de datos históricos:

- De 8 a 16 procesadores.
- De 24 a 32 GB RAM.
- De 80 a 120 GB disco duro.

RNF Interfaces Software:

El sistema operativo a utilizar debe ser de preferencia debido a que la aplicación es multiplataforma. Se debe contar con la Máquina virtual de Java 1.6.x instalada.

Estaciones clientes:

- Navegador web recomendado, Mozilla Firefox cualquiera de sus versiones a partir de su versión 3.0.x o emplear el Internet Explorer.
- Sistemas Operativos Linux, kernel 2.6.24 o superior, Windows XP SP2 o superior.
- Navegador web tiene que tener habilitado el uso de JavaScript.

Servidores:

- Sistema operativo RedHat Enterprise Linux v5.0 o superior.
- JDK 1.6.x.
- Python 2.6.x instalado.
- Oracle 11g.
- Bibliotecas Boost para C++.
- JasperReports Server.
- JBoss 4.2.3 o 5.1.0, full installation (incluye JBoss Messaging, JBoss JBPM, JBoss Drools).

Servidor del banco de datos históricos:

- Sistema operativo RedHat Enterprise Linux v5.0 o superior.
- JDK 1.6.x.
- Python 2.6.x instalado.
- Oracle 11g.
- Oracle BI EE 10.1.3.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

RNF Interfaces de Comunicación:

SIAI de forma general debe constituir un aplicativo compatible con sistemas actuales que se encuentren en explotación, lo cual le permitirá ajustarse al escenario donde se desplegará el sistema.

Dentro de los sistemas externos fundamentales con los cuales debe presentarse una integración se encuentran:

- Servidor de correos, móviles y otros.

Se prevé el envío de mensajes de correo electrónico, mensajes SMS u otro tipo de notificación a servidores de correos y móviles, con posibilidad de extenderse en el futuro; todos ellos a través del subsistema “Servidor de Colas”.

Requisitos de licencia:

- Para JBoss es LGPL (GNU Lesser General Public Licence).
- Para PostgreSQL es LGPL.
- Para JasperReports es LGPL.
- Para SQLite es CPL (Common Public Licence).
- Para las máquinas virtuales Java también es LGPL.
- Para Eclipse es EPL (Eclipse Public Licence).
- Para OracleBI es comercial y depende del número de servidores, la cantidad de procesadores de cada servidor y/o el número de conexiones hacia el servidor de aplicaciones.

En caso de comercialización del sistema se incluirá al precio del producto el monto de las licencias OracleBI y OracleDB, o se le informa al comprador que debe negociar las licencias con las características anteriormente descritas.

Estándares aplicables:

- NC ISO/IEC 25000: Ingeniería de software - Requisitos de Calidad y Evaluación de productos de software (SQUARE) -Guía para SQUARE.
- NC - ISO/IEC 9126: Ingeniería de Software. Calidad del Producto.

2.5. Definición de los casos de uso.

2.5.1. Definición de los actores del sistema.

Tabla 2: Actor del sistema.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Actor	Descripción
Usuario	Interactúa con la Plataforma WEB, realizando todas aquellas operaciones que le fueron asignadas en dependencia de los privilegios y permisos que se le otorgaron.

2.5.2. Diagrama de caso de uso del sistema.

Los casos de usos que se muestran en la figura 2 engloban las funcionalidades afines, con el objetivo de agrupar las mismas y respondiendo a cada uno de los requisitos funcionales expuestos anteriormente. Los casos de usos que poseen una tonalidad diferente al azul, se colorearon intencionalmente para indicar que se aplican al módulo pero forman parte de las funcionalidades del SIAI.

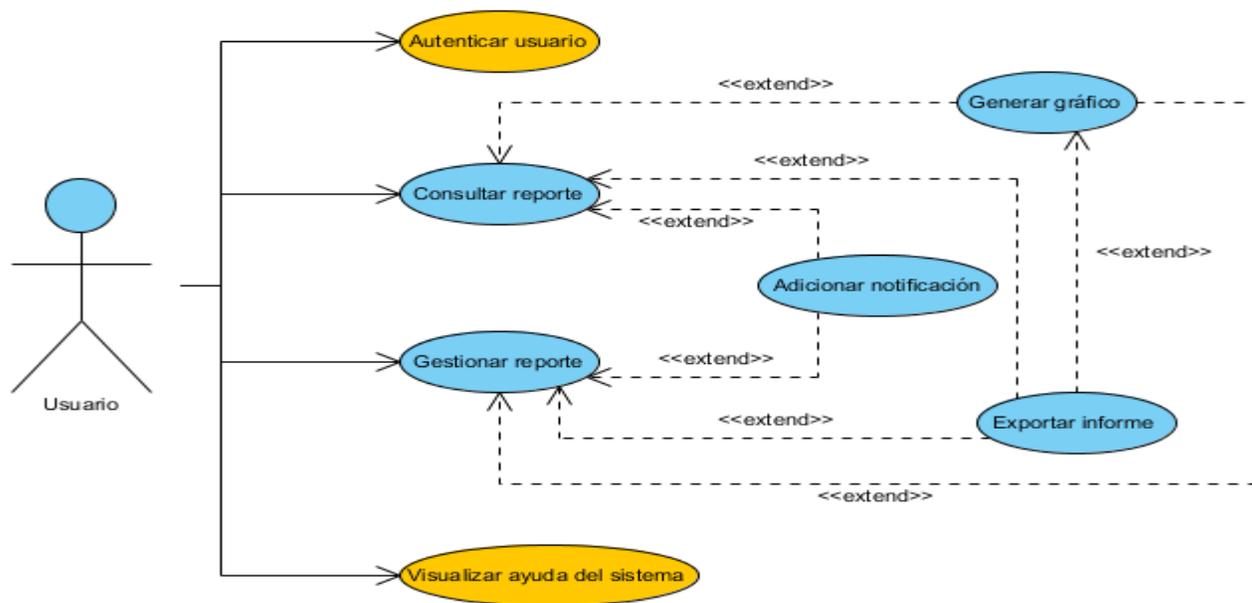


Fig. 2: Diagrama de caso de uso del sistema

2.5.3. Descripción textual de casos de usos.

La descripción textual de los caso de usos se realiza con el objetivo de ilustrar la interacción del actor con el sistema, permitiendo un mejor análisis y comprensión de cada funcionalidad. A continuación se expone

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

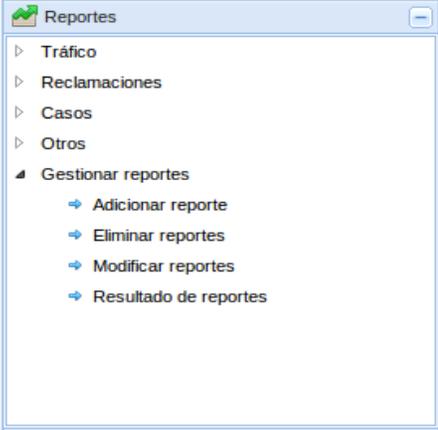
en la Tabla 12 la descripción para el CU4, mientras que para el CU1, CU2 y CU3 sus descripciones se podrán encontrar en el [Anexo 2](#), específicamente en la Tabla 14, Tabla 15 y Tabla 16.

Tabla 3: Descripción textual CU Gestionar reporte.

Caso de Uso:	Gestionar reporte	
Actores:	Usuario (Inicia)	
Resumen:	El CU se inicia cuando el usuario desea gestionar un reporte.	
Precondiciones:	<ol style="list-style-type: none"> 1. El usuario tiene que estar autenticado en el sistema. 2. El usuario debe tener permisos para acceder a esta parte del sistema. 	
Referencias:	RF55, RF56, RF57, RF58. CU Generar gráfico. (Extensión) CU Exportar informe. (Extensión) CU Adicionar notificación. (Extensión)	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Sección “Gestionar reportes”		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción del submenú “Gestionar reporte”	2. Muestra las siguientes opciones: <ul style="list-style-type: none"> • Adicionar reporte. (Ir a la sección de Adicionar reporte) • Eliminar reportes. (Ir a la sección de Eliminar reportes) • Modificar reportes. (Ir a la sección de Modificar reportes) • Resultado de reportes. (Ir a la sección de Resultado de 	

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

reportes)	
Prototipo de Interfaz	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prototipo de Interfaz	
Flujo Normal de Eventos	
Sección “Adicionar reporte”	
Acción del Actor	Respuesta del Sistema
1. Accede a la opción “Adicionar reportes”.	2. Muestra la interfaz solicitando los siguientes datos: <ul style="list-style-type: none"> • Nombre. • Descripción. • Campos de salida. • Tablas de la BD.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

	<ul style="list-style-type: none">• Campos de la tabla seleccionada.• Aceptar.• Campos a mostrar.• Condiciones de salida.• Operadores lógicos.• Campos de la base de datos.• Operadores operacionales.• Valor de la condición.• Eliminar condición.• Adicionar condición.
3. Introduce los datos solicitados. Selecciona la opción “ <i>Aceptar</i> ”.	4. Verifica la validez de los datos introducidos.
	5. Genera una consulta SQL con los valores de los Campos a mostrar y las Condiciones.
	6. Guarda en la BD los valores del nombre del reporte, Descripción y la consulta SQL generada.
	7. Muestra el siguiente mensaje: “ <i>Reporte generado y almacenado correctamente</i> ”
Prototipo de Interfaz Datos de entrada	

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Datos generales

Nombre:

Descripción:

Condiciones de salida

Campos de salida

Tablas de la BD:
Seleccionar... ▼

Campos de la tabla seleccionada

Aceptar

Campos a mostrar

Aceptar Cancelar

Nombre:

Descripción:

Condiciones de salida

Campos de salida

Tablas de la BD:
Seleccionar... ▼

Campos de la tabla seleccionada

- accounttype
- accountname
- creditlimit
- client
- clientstatus
- clienttype
- clientname

Aceptar

Campos a mostrar

- Account.accounttype
- Account.accountname

Condiciones de salida

AND ▼ Account.accountname ▼ equals ▼ account

Aceptar Cancelar

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

<div data-bbox="618 321 1070 474" style="border: 1px solid gray; padding: 5px; width: fit-content; margin: auto;">Informe de servidores Reporte generado y almacenado correctamente</div>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1. Selecciona la opción “ <i>Cancelar</i> ”.	3.2. Limpia los valores introducidos en los elementos visuales.
	4.1. Muestra un señalamiento en los datos que no se corresponden con el formato indicado o en los campos obligatorios que se hayan dejado vacíos.
	4.2. Muestra el mensaje “ <i>Existen valores incorrectos</i> ” al no seleccionar campos para visualizar.
	6.1. No puede guardar los datos por problemas con la conexión a la BD.
Prototipo de Interfaz	
<div data-bbox="280 1215 1409 1438" style="border: 1px solid gray; padding: 10px; width: fit-content; margin: auto;"><p>Condiciones de salida</p><div style="display: flex; align-items: center; gap: 10px;"><div style="border: 1px solid gray; padding: 2px;">AND</div><div style="border: 1px solid gray; padding: 2px;">✖</div><div style="border: 1px solid gray; padding: 2px;">Account.accountname</div><div style="border: 1px solid gray; padding: 2px;">equals</div><div style="border: 1px solid gray; padding: 2px; width: 100px; height: 20px;"></div><div style="border: 1px solid gray; padding: 2px;">!</div></div><div style="display: flex; align-items: center; gap: 10px; margin-top: 5px;"><div style="border: 1px solid gray; padding: 2px;">+</div></div></div>	
<div data-bbox="618 1482 1070 1640" style="border: 1px solid gray; padding: 5px; width: fit-content; margin: auto;">Advertencia Existen valores incorrectos</div>	
Flujo Normal de Eventos	

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Sección “Eliminar reportes”	
Acción del Actor	Respuesta del Sistema
1. Accede a la opción “ <i>Eliminar reportes</i> ”.	2. Muestra una interfaz con los siguientes campos: <ul style="list-style-type: none">• Eliminar.• Reportes.
3. Selecciona los reportes a eliminar. Selecciona la opción “ <i>Eliminar</i> ”.	4. Muestra el mensaje de confirmación “¿ <i>Desea realizar la operación seleccionada?</i> ”
5. Selecciona la opción “Yes”	6. Elimina de la base de datos el reporte seleccionado en la interfaz.
Prototipo de Interfaz	
Datos de entrada	

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

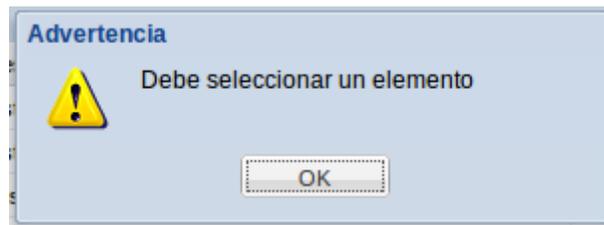
<p>Operaciones</p> <p></p>																							
<p>Reportes</p> <p> Preferencias ▾</p> <table border="1"><thead><tr><th><input type="checkbox"/> Nombre</th><th>Descripción</th></tr></thead><tbody><tr><td><input type="checkbox"/> reportAccount</td><td>test account</td></tr><tr><td><input type="checkbox"/> reportCentral</td><td>test this report</td></tr><tr><td><input type="checkbox"/> reportFMCaseExtention</td><td>Test this report</td></tr><tr><td><input type="checkbox"/> reportFMFile</td><td>test this repory</td></tr><tr><td><input type="checkbox"/> reportPolicyRuleExtension</td><td>test this report</td></tr><tr><td><input type="checkbox"/> reportSubscriber</td><td>assasasasasas</td></tr><tr><td><input type="checkbox"/> ghjghj</td><td>ghjghj</td></tr><tr><td><input type="checkbox"/> ghjghj</td><td>ghjghj</td></tr><tr><td><input type="checkbox"/> chamizo1</td><td>fghfhdfh</td></tr><tr><td><input type="checkbox"/> REPORT</td><td>TEST REPORT</td></tr></tbody></table> <p>Page 1 of 1 </p>		<input type="checkbox"/> Nombre	Descripción	<input type="checkbox"/> reportAccount	test account	<input type="checkbox"/> reportCentral	test this report	<input type="checkbox"/> reportFMCaseExtention	Test this report	<input type="checkbox"/> reportFMFile	test this repory	<input type="checkbox"/> reportPolicyRuleExtension	test this report	<input type="checkbox"/> reportSubscriber	assasasasasas	<input type="checkbox"/> ghjghj	ghjghj	<input type="checkbox"/> ghjghj	ghjghj	<input type="checkbox"/> chamizo1	fghfhdfh	<input type="checkbox"/> REPORT	TEST REPORT
<input type="checkbox"/> Nombre	Descripción																						
<input type="checkbox"/> reportAccount	test account																						
<input type="checkbox"/> reportCentral	test this report																						
<input type="checkbox"/> reportFMCaseExtention	Test this report																						
<input type="checkbox"/> reportFMFile	test this repory																						
<input type="checkbox"/> reportPolicyRuleExtension	test this report																						
<input type="checkbox"/> reportSubscriber	assasasasasas																						
<input type="checkbox"/> ghjghj	ghjghj																						
<input type="checkbox"/> ghjghj	ghjghj																						
<input type="checkbox"/> chamizo1	fghfhdfh																						
<input type="checkbox"/> REPORT	TEST REPORT																						
<p>Información</p> <p> ¿Desea realizar la operación seleccionada?</p> <p><input type="button" value="Yes"/> <input type="button" value="No"/></p>																							
<p>Informe de servidores</p> <p>Reporte eliminado correctamente</p>																							
<p>Flujos Alternos</p>																							
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>																						

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

3.1. Selecciona la opción “ <i>Eliminar</i> ”.	3.2. Muestra el siguiente mensaje: “ <i>Debe seleccionar un elemento</i> ”.
5.1. Selecciona la opción “ <i>No</i> ”	5.2. Se cierra la ventana con el mensaje de confirmación.

Prototipo de Interfaz



Flujo Normal de Eventos

Sección “Modificar reportes”

Acción del Actor	Respuesta del Sistema
1. Accede a la opción “ <i>Modificar reportes</i> ”.	2. Muestra una interfaz con los siguientes campos: <ul style="list-style-type: none"> • Modificar. • Reportes.
3. Selecciona el reporte a modificar. Selecciona la opción “ <i>Modificar</i> ”.	4. Se visualizan los siguientes campos: <ul style="list-style-type: none"> • Nombre. • Descripción. • Elementos actuales. <ul style="list-style-type: none"> • Campos a mostrar. • Condiciones de salida. • Campos de salida. <ul style="list-style-type: none"> • Tablas de la BD.

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

	<ul style="list-style-type: none">• Campos de la tabla seleccionada.• Aceptar.• Campos a mostrar.• Condiciones de salida.<ul style="list-style-type: none">• Operadores lógicos.• Campos de la base de datos.• Operadores operacionales.• Valor de la condición.• Eliminar condición.• Adicionar condición.• Aceptar• Cancelar
5. Modifica los valores deseados. Selecciona la opción “ <i>Aceptar</i> ”.	6. Genera un nuevo reporte dinámico con los valores especificados.
	7. Actualiza en la BD los valores.
	8. Muestra el siguiente mensaje: “ <i>Reporte modificado y almacenado correctamente</i> ”.
Prototipo de Interfaz Datos de entrada	

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Operaciones

 Modificar

Reportes

 Preferencias ▾

<input type="checkbox"/> Nombre	Descripción
<input type="checkbox"/> reportAccount	test account
<input type="checkbox"/> reportFMFile	test this repory
<input type="checkbox"/> reportPolicyRuleExtension	test this report
<input type="checkbox"/> reportSubscriber	assasasasasas
<input type="checkbox"/> ghjghj	ghjghj
<input type="checkbox"/> ghjghj	ghjghj
<input type="checkbox"/> chamizo1	fghfhdfh
<input type="checkbox"/> REPORT	TEST REPORT
<input type="checkbox"/> reportTest	
<input type="checkbox"/> chamizo	probando

Page 1 of 1 

Datos generales

Nombre:

Descripción:

Condiciones de salida

Campos de salida

elementos actuales

Campos a mostrar

- FMFile.state,
- FMFile.name,
- FMFile.finalresult,
- FMFile.employee

Condiciones de salida

- FMFile.name != 'kgkgkg'

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Condiciones de salida

AND

✖

Seleccionar...

=

+

Campos de salida

Tablas de la BD:

Seleccionar...

Campos de la tabla seleccionada

Aceptar

Campos a mostrar

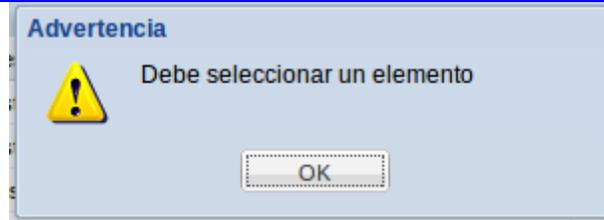
Flujos Alternos

Acción del Actor	Respuesta del Sistema
3.1. Selecciona la opción “ <i>Modificar</i> ”	3.2. Muestra el siguiente mensaje: “ <i>Debe seleccionar un elemento</i> ”.
5.1. No selecciona ningún dato de salida de los existentes y no entra ninguno nuevo. Selecciona la opción “ <i>Aceptar</i> ”.	5.2. Muestra el siguiente mensaje: “ <i>Tiene que especificar al menos un campo de salida</i> ”.
5.3. Selecciona la opción “ <i>Cancelar</i> ”	5.4. Se oculta el campo donde se cargan los valores del reporte a modificar.

Prototipo de Interfaz

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”



Flujo Normal de Eventos

Sección “Resultado de reportes”

Acción del Actor	Respuesta del Sistema
1. Accede a la opción “Resultado de reportes”	2. Muestra una interfaz con los siguientes campos: <ul style="list-style-type: none">• Aplicar.• Reportes.
3. Selecciona el reporte del cual se desea el resultado. Selecciona la opción “Aplicar”.	4. Realiza una búsqueda en la BD.
	5. Se visualizan los siguientes campos: <ul style="list-style-type: none">• Resultado del reporte.• Cancelar.

Prototipo de Interfaz

Datos de entrada

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Operaciones

Aplicar

Reportes

 Preferencias ▾

<input type="checkbox"/> Nombre	Descripción
<input type="checkbox"/> reportAccount	test account
<input type="checkbox"/> reportFMFile	test this repory
<input type="checkbox"/> reportPolicyRuleExtension	test this report
<input type="checkbox"/> reportSubscriber	assasasasasas
<input type="checkbox"/> ghjgjh	ghjgjh
<input type="checkbox"/> ghjgjh	ghjgjh
<input type="checkbox"/> chamizo1	fghfhdfh
<input type="checkbox"/> REPORT	TEST REPORT
<input type="checkbox"/> reportTest	
<input type="checkbox"/> chamizo	probando

Page 1 of 1

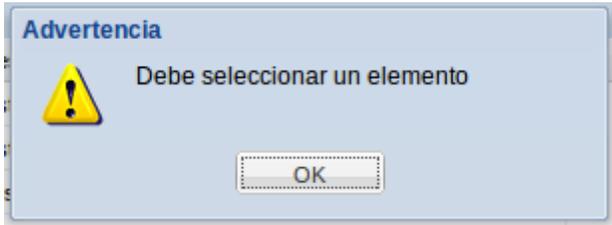
Resultado del reporte

Account.accounttype	Account.accountname	Account.creditlimit	Account.client	Account.clientstatus	Account.clienttype
0	WTVTH	null	null	null	null
0	622JH	null	null	null	null
0	FIFSZ	null	null	null	null
0	S87YT	null	null	null	null
0	YHJZZ	null	null	null	null
0	YY9J0	null	null	null	null
0	OYXYN	null	null	null	null
0	C305J	null	null	null	null
0	7VYWH	null	null	null	null
0	POACL	null	null	null	null
0	ELO1X	null	null	null	null
0	MG73W	null	null	null	null

Cancelar

Capítulo 2: “Características del sistema”

“Módulo de Reportes del SIAI”

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1. Selecciona la opción “Aplicar”	3.2. Muestra el mensaje: “Debe seleccionar un elemento”.
Prototipo de Interfaz	
	
Poscondiciones:	<ol style="list-style-type: none">1. Se crea el reporte de manera dinámica y se guarda en la base de datos.2. Se elimina de la base de datos el reporte seleccionado.3. Se modifican los valores en la base de datos del reporte seleccionado.4. Se obtiene el resultado del reporte seleccionado.

2.6. Conclusiones.

En el presente capítulo se definieron los requisitos funcionales y no funcionales del sistema, elaborados minuciosamente a través de las entrevistas realizadas a los especialistas del negocio. Los mismos se agruparon en 4 casos de uso, los cuales fueron representados en el diagrama de caso de uso del sistema y descritos posteriormente, atendiendo en todo momento las necesidades de los clientes.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1. Introducción.

El presente capítulo contiene el diseño del sistema como parte de la solución propuesta. Se describe el estilo arquitectónico y patrones del diseño seleccionados, así como los diagramas de clases que satisfacen a cada uno de los requisitos detectados. También se muestra el modelo de datos lógico y físico de la base de datos con la descripción de cada uno de los atributos por lo que se encuentran conformados.

3.2. Arquitectura del sistema.

La arquitectura del sistema es uno de los elementos que tiene un gran impacto directo en la capacidad para satisfacer los atributos de calidad del sistema. Estos atributos forman parte de los requerimientos no funcionales, por lo que una buena definición de arquitectura ayuda a un buen desempeño de la aplicación. También influye directamente en la guía del desarrollo del sistema, además de que puede ser empleada para el desarrollo de diferentes sistemas en caso de ser positivos sus resultados. Los patrones arquitectónicos son los que definen la estructura que presentará un sistema de software, los cuales a sus veces se encuentran conformados por subsistemas con responsabilidades específicas. También presentan una serie de directivas con el objetivo de organizar los diferentes componentes con el objetivo de facilitar la tarea de diseño (21).

3.2.1. Propuesta de arquitectura del sistema.

La arquitectura por capas es un estilo de programación el cual tiene como principal objetivo separar las capas de presentación, lógica del negocio y acceso a datos. Mediante la misma los datos y servicios estarán completamente separados, por lo que el usuario recibe la información de manera indirecta mediante un servidor. Por sus principales ventajas este estilo arquitectónico es empleado por el SIAI como se muestra en la figura 4, por lo que para el desarrollo del módulo es también empleado con el objetivo de continuar una sola línea de desarrollo.

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

La capa de “*Presentación*” es la que se le muestra al usuario, con el objetivo de presentarle la aplicación y gestionar la información manejada. La principal característica que tiene que poseer, es que debe ser amigable y lograr que el usuario se sienta motivado para emplear la aplicación.

La capa de “*Lógica del negocio*” es la encargada de manejar todas las reglas que tienen que cumplirse con el objetivo de manejar las funcionalidades del sistema.

La capa de “*Acceso a datos*” es la encargada de contener las entidades necesarias para el trabajo del sistema, con el objetivo de comunicar a la lógica del con la base de datos.

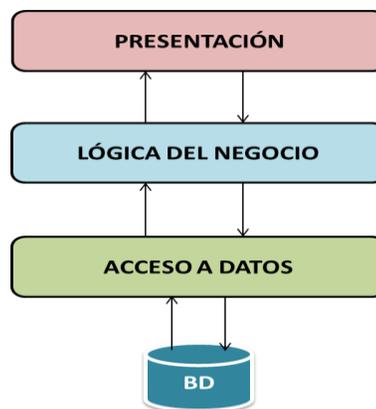


Fig. 3: Arquitectura del SIAI

La arquitectura a emplear para el desarrollo del módulo es la misma a emplear por el SIAI, desarrollando cada una de las clases en la respectiva capa que le corresponde, con el objetivo de garantizar un alto nivel de seguridad y mantener una sola línea en el desarrollo de la plataforma final.

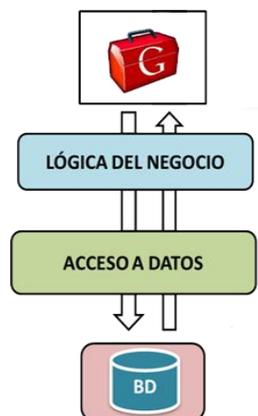


Fig. 4: Arquitectura del Sub Módulo Reportes del SIAI

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

3.2.2. Patrón MVP de la capa de presentación.

El patrón a emplear en la presentación es el MVP⁶ al igual que en el SIAI como de muestra en la figura 6. Mediante el mismo se puede desarrollar las interfaces del sistema de una manera más sencilla teniendo la menor cantidad de código para desarrollarlas. Por su parte el presentador se encarga de manejar toda la lógica de las interfaces sin depender completamente de las mismas, realizando el enlace entre las vistas y los modelos, dotando a las vistas de una mayor inteligencia. Cada presentador tiene como entrada la vista a la cual debe implementar, manejando el tráfico de los datos y los eventos que se realizan.

Vista: Es donde se encuentran las interfaces de usuario, en las mismas se encuentran los campos y/o controles.

Presentador: Es donde se manejan las entidades, se llenan los datos de la vista y se manejan los eventos. Es importante tratar de que todo el código se desarrolle en este miembro.

Modelo: Son las clases que representan las entidades, donde solo el presentador tiene acceso a las mismas para luego ser mostradas por las vistas.

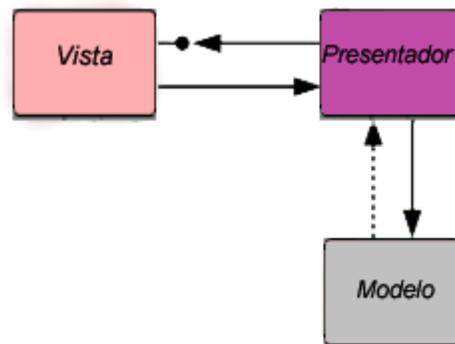


Fig. 5: Patrón MVP

3.2.3. Patrones de diseño.

Los patrones de diseño son una solución repetible general de un problema que ocurre comúnmente en el diseño de software. Es una descripción o plantilla para la forma de resolver un problema que se puede utilizar en muchas situaciones diferentes. Pueden acelerar el proceso de desarrollo, proporcionando,

⁶ MVP: Modelo Vista Presentador: Patrón derivado del Modelo Vista Controlador (MVC) que ayuda a ofrecer una clara separación entre la vista, el modelo y el controlador

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

paradigmas de desarrollo probadas. El diseño eficaz de software requiere considerar los problemas que pueden no hacerse visibles hasta más tarde en la ejecución. La reutilización de los patrones de diseño ayuda a evitar problemas sutiles que pueden causar grandes problemas y mejora la legibilidad del código para los programadores y arquitectos están familiarizados con los patrones (23).

3.2.3.1. Patrones de diseño empleado en el negocio.

Para la capa de negocio, se emplearon algunos patrones de diseño que ayudaron sustancialmente al diseño de dicha capa. A continuación se muestran los mismos:

Business Interface:

Se abstrae los detalles de implementación de la lógica de negocio, lo que permite tanto facilitar las pruebas de las cosas que utilizan las interfaces, y la reutilización de la lógica de negocio de manera arbitraria. Los EJB se componen por una interfaz remota y la clase de implementación del EJB, además de la clase de entidad del EJB. La clase de implementación del EJB debe implementar los métodos definidos en la interfaz remota, surgiendo de esta manera el patrón Interfaces de Negocio. La Interface de Negocio, como toda interfaz, define un modelo a seguir por las clases que la implementen, logrando de esta manera que la interfaz remota y la clase de implementación del EJB cuenten con los mismos métodos antes de ser compiladas, evitando que se arrastren errores hasta la compilación total del proyecto o inclusive hasta su despliegue.

DAO (Data Access Object):

Para el desarrollo del módulo se emplea el patrón, con el objetivo de acceder a la base de datos mediante los mecanismos de acceso implementados por el DAO. De esta forma se oculta completamente el origen de los datos que son trabajados desde la aplicación. Para el correcto funcionamiento del patrón se implementó en el negocio las clases que interactúan entre el acceso a los datos y la aplicación cliente, dichas clases son interfaces remotas para ser accedidas desde el servidor en GWT. Por otra parte se generaron los objetos de transferencia, los cuales son las entidades que representan las tablas por la cual se encuentra conformada la base de datos y por último se implementaron las clases de acceso a dato, las cuales son el objeto principal del patrón, ya que son las clases que permiten el acceso de manera transparente hacia la información almacenada. En la Fig. 6 se puede apreciar de manera general la estructura empleada para el acceso a los datos por parte del módulo.

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

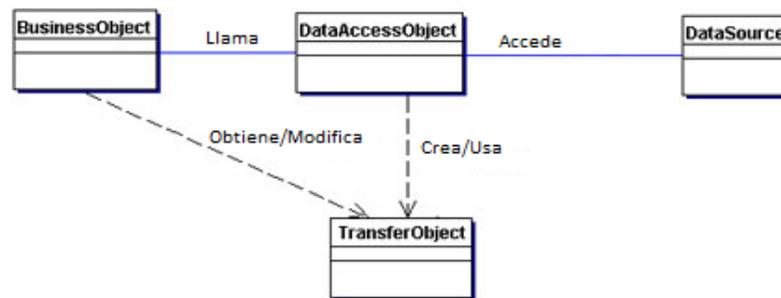


Fig. 6: Diagrama de clases del patrón DAO.

BusinessObject: Representa al cliente los datos. Objeto que requiere el acceso a la fuente de datos para obtener y almacenar datos.

DataAccessObject: Abstrae la implementación del acceso a los datos subyacentes para el BusinessObject para permitirle el acceso transparente a la fuente de datos. También delega la carga de datos y operaciones de almacenamiento.

DataSource: Representa una implementación de fuente de datos la cual en para el desarrollo del SIAI se emplea una base de datos Oracle

TransferObject: Representa el objeto de transferencia que se emplea como soporte de los datos. Puede ser utilizado por el DataAccessObject con el objetivo de devolver los datos al cliente. También puede recibir los datos del cliente con el objetivo de actualizar la información en la base de datos.

3.2.3.2. Patrones de diseño empleado en la presentación.

Patrones GRASP

Los patrones GRASP (*del inglés: General Responsibility Assignment Software Patterns*) son un grupo de patrones lo cuales tiene como objetivo principal la descripción de los principios fundamentales para la asignación de responsabilidades a objetos. Entre los principales patrones que se encuentran dentro de los GRASP se destacan los 5 principales que son: “*Experto*”, “*Creador*”, “*Alta cohesión*”, “*Bajo acoplamiento*” y “*Controlador*”. A continuación se describen los empleados para el desarrollo del sub módulo.

Experto:

Para el desarrollo del módulo, fue necesario el empleo de dicho patrón con el objetivo de que las clases presentadoras fueran las encargadas de conocer solamente los valores de la vista a las que manejan, de esta forma pueden informar y modificar toda la información que se maneja en las interfaces y tienen el

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

objetivo principal de crearlas. De esta forma se conserva el encapsulamiento, ya que los objetos mediante la información que presentan pueden realizar las operaciones que le sean encomendados.

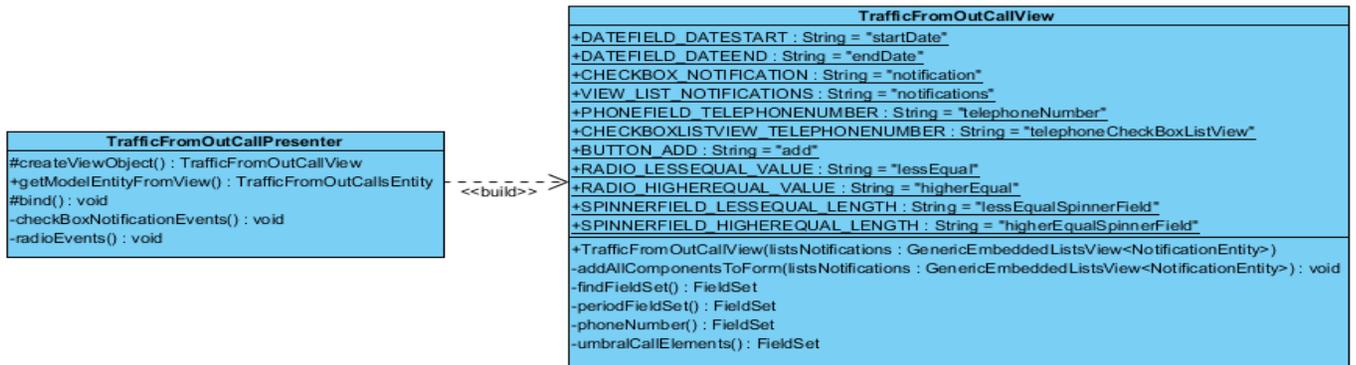


Fig. 7: Patrón Experto en el Módulo de reportes

Creador:

Para el desarrollo del módulo fue necesario el empleo de dicho patrón para la comunicación entre los presentadores y las interfaces, ya que en los presentadores es donde se encuentran los datos de inicialización necesarios para construir las vistas, por lo que los presentadores son los creadores de las vistas y utiliza sus objetos para trabajarlos.

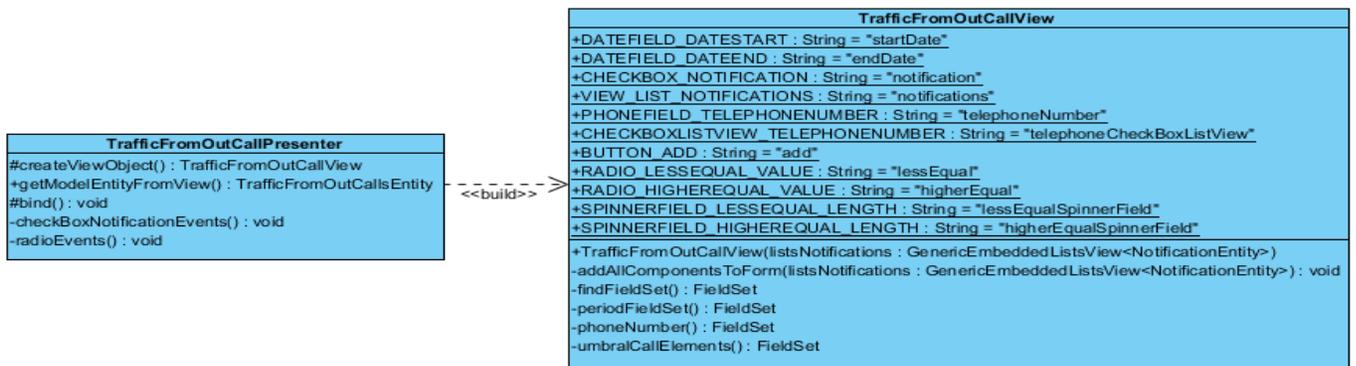


Fig. 8: Patrón Creador en el Módulo de reportes

Bajo acoplamiento:

Para el desarrollo del módulo fue necesario el empleo de dicho patrón con el objetivo de evitar una gran cantidad de conexiones entre las diferentes clases, permitiendo principalmente la independencia de cada

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

uno para reducir el impacto en los futuros cambios que se realicen. Por lo general las clases desarrolladas extenderán principalmente de las clases genéricas desarrolladas en el paquete en el cual se encuentra.

TrafficFromOutCallsEntity
-serialVersionUID : long = 1L
+TrafficFromOutCallsEntity() +TrafficFromOutCallsEntity(numberOrigin : String, numberDestiny : String, cantOfCallsDestiny : String) +getNumberOrigin() : String +setNumberOrigin(numberOrigin : String) : void +getNumberDestiny() : String +setNumberDestiny(numberDestiny : String) : void +getCantOfCallsDestiny() : String +setCantOfCallsDestiny(cantOfCallsDestiny : String) : void

Fig. 9: Patrón Bajo acoplamiento en el Módulo de reportes

Alta cohesión:

Para el desarrollo de módulo fue necesario el empleo de dicho patrón con el objetivo de minimizar las responsabilidades que presentan las clases, por lo que las mismas solo presentarán la información necesaria para su trabajo, evitando los métodos y atributos que pueden sobrecargarlas y dificulte su entendimiento y reutilización.

TrafficFromOutCallView
+DATEFIELD_DATESTART : String = "startDate" +DATEFIELD_DATEEND : String = "endDate" +CHECKBOX_NOTIFICATION : String = "notification" +VIEW_LIST_NOTIFICATIONS : String = "notifications" +PHONEFIELD_TELEPHONENUMBER : String = "telephoneNumber" +CHECKBOXLISTVIEW_TELEPHONENUMBER : String = "telephoneCheckBoxListView" +BUTTON_ADD : String = "add" +RADIO_LESSEQUAL_VALUE : String = "lessEqual" +RADIO_HIGHEREQUAL_VALUE : String = "higherEqual" +SPINNERFIELD_LESSEQUAL_LENGTH : String = "lessEqualSpinnerField" +SPINNERFIELD_HIGHEREQUAL_LENGTH : String = "higherEqualSpinnerField"
+TrafficFromOutCallView(listsNotifications : GenericEmbeddedListView<NotificationEntity>) -addAllComponentsToForm(listsNotifications : GenericEmbeddedListView<NotificationEntity>) : void -findFieldSet() : FieldSet -periodFieldSet() : FieldSet -phoneNumber() : FieldSet -umbralCallElements() : FieldSet

Fig. 10: Patrón Alta cohesión en el Módulo de reportes

3.3. Diagramas de interacción.

3.3.1. Diagrama de Paquetes.

Mediante la representación del diagrama de paquetes se puede reflejar la manera en la que quedan organizados los paquetes y los elementos por los que se encuentra conformado el módulo de reportes.

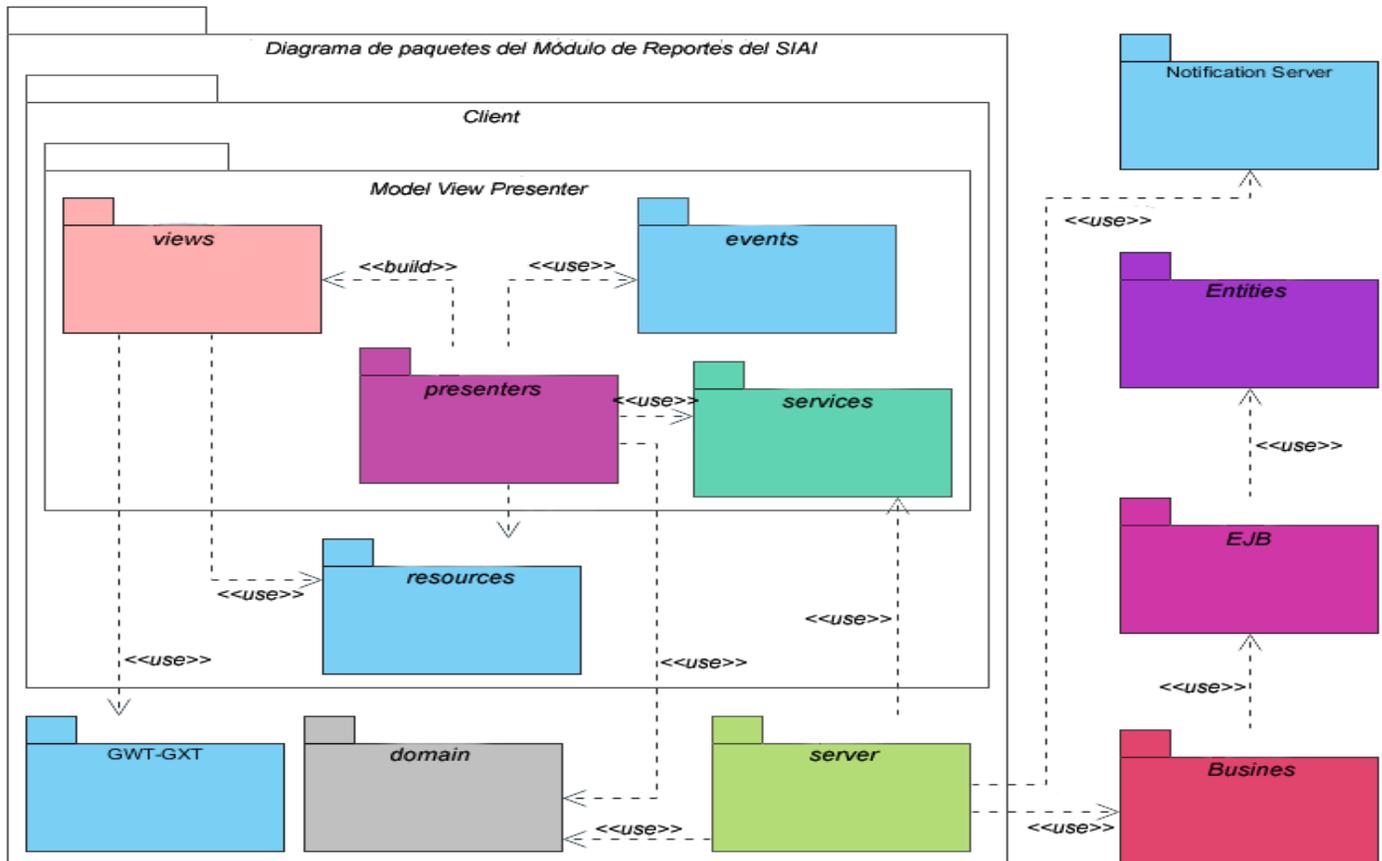


Fig. 11: Diagrama de paquetes del Sub Módulo Reportes del SIAI

3.3.2. Diagrama de clases del Diseño.

Los diagramas de clases del diseño se realizarán por caso de uso, con el objetivo de representar un flujo completo y para ello se empleará el mismo caso de uso que se vio en la descripción y los restantes se encontrarán en los anexos. Para el diagrama de clases mostrado en la Fig. 12 se muestra específicamente el flujo de la sección “Adicionar reporte”.

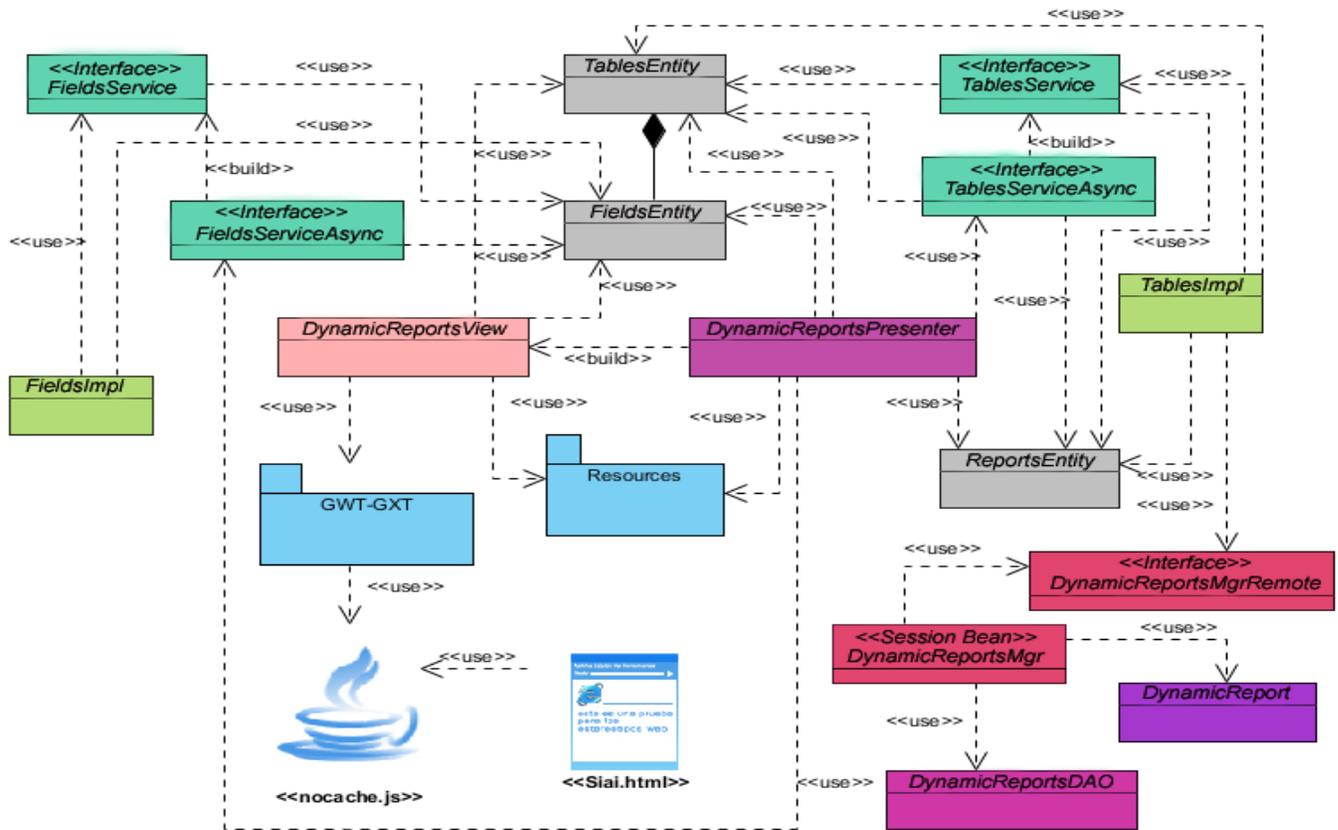


Fig. 12: Diagrama de clases del diseño del CU Gestionar reporte, sección “Adicionar reporte”.

3.4. Modelo de datos.

Los modelos de datos son una estructura abstracta que documenta y organiza la documentación. Para el desarrollo de los reportes dinámicos fue necesario el desarrollo de una nueva tabla en la base de datos, la cual se encargará de guardar la información correspondiente de los reportes que se generen de manera dinámica. La misma solo tendrá relación con la tabla de notificaciones, ya que el usuario tendrá la oportunidad de solicitar notificaciones una vez realizado el reporte generado. Por las características del generador de reportes dinámicos, el módulo trabaja también con las otras tablas debido a que, para realizar los reportes se debe interactuar con toda la base de datos para ofrecer la información solicitada por los usuarios.

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

3.4.1. Modelo Lógico de Datos.

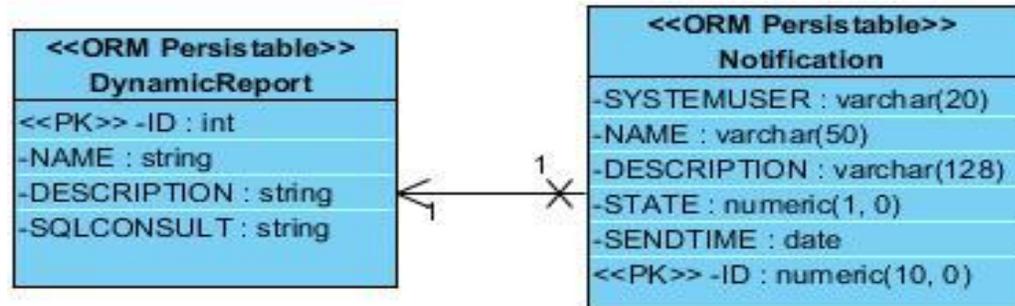


Fig. 13: Diagrama de clases persistentes.

Tabla 4: Descripción de los atributos del Modelo Lógico de Datos.

Nombre: DynamicReport		Nombre: Notification	
Descripción: Contiene la información de los reportes que se generan de manera dinámica.		Descripción: Contiene la información de las notificaciones realizadas.	
Atributos:		Atributos:	
Nombre:	Tipo de Dato:	Nombre:	Tipo de Dato:
ID	int	SYSTEMUSER	varchar(20)
NAME	String	NAME	varchar(50)
DESCRIPTION	String	DESCRIPTION	varchar(128)
SQLCONSULT	String	STATE	numeric(1, 0)
		SENDTIME	date
		ID	numeric (10, 0)

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

3.4.2. Modelo Físico de Datos.

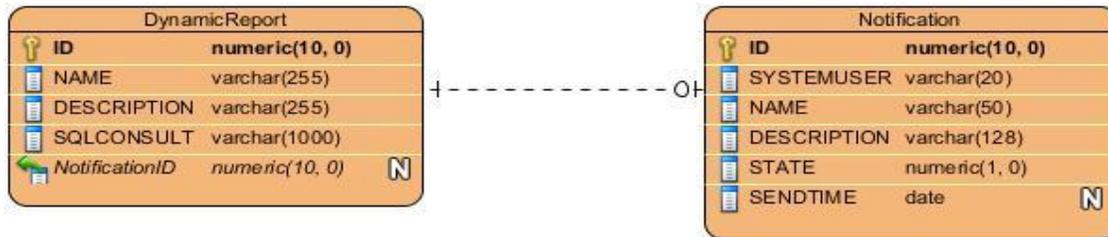


Fig. 14: Diagrama del modelo entidad relación.

Tabla 5: Descripción de los atributos del Modelo Físico de Datos.

Nombre: DynamicReport				
Descripción: Almacena todos los reportes generados por los usuarios en el sistema.				
Nombre:	Tipo de Dato:	PK/FK	¿Nulo?	Descripción
ID	numeric(10, 0)	PK	No	Identificador del reporte generado.
NAME	varchar(255)		No	Nombre del reporte.
DESCRIPTION	varchar(255)		No	Descripción del reporte.
SQLCONSULT	varchar(1000)		No	Consulta generada para desarrollar el reporte.
NotificationID	numeric(10, 0)	FK	Si	Identificador de la notificación.
Nombre: Notification				
Descripción: Almacena todas las notificaciones solicitadas por los usuarios en el sistema.				
Nombre:	Tipo de Dato:	PK/FK	¿Nulo?	Descripción
ID	numeric(10, 0)	PK	No	Identificador de la notificación generada.
SYSTEMUSER	varchar(20)		No	Usuario del sistema al cual realizar la notificación.

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

NAME	varchar(50)		No	Nombre de la notificación.
DESCRIPTION	varchar(128)		No	Descripción de la notificación.
STATE	numeric(1, 0)		No	Estado de la notificación.
SENDTIME	date		Si	Tiempo definido para generar la notificación.

3.5. Diagrama de despliegue.

Con el objetivo de modelar el hardware empleado en la implementación del sistema, así como la relación entre sus diferentes componentes, se muestra en la Fig. 15 el diagrama de despliegue empleado para el desarrollo del módulo.

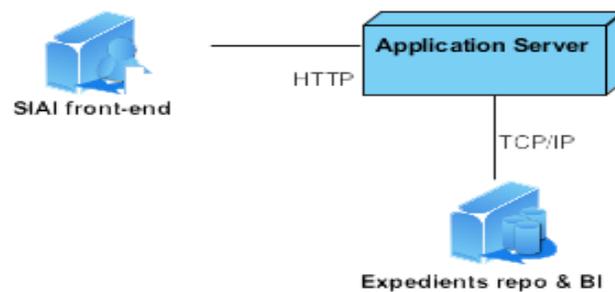


Fig. 15: Diagrama de despliegue

3.5.1. Descripción de los nodos del diagrama de despliegue.

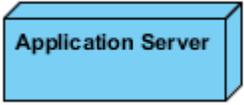
A continuación se procede con la descripción de los nodos por los que se encuentra conformado el diagrama de componentes.

Tabla 6: Descripción de los nodos del diagrama de despliegue.

Nodo	Descripción

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

 <p>Expedients repo & BI</p>	Aloja un servidor de base de datos para el repositorio de expedientes creados para la investigación de los casos. Además en este nodo se realizarán las tareas de inteligencia de negocio (BI).
 <p>Application Server</p>	Servidor de aplicaciones J2EE. Servirá de plataforma base al “SIAI front-end”. Además alojará el servidor Jasper Server para gestionar los reportes predefinidos de la aplicación de detección con el SW10 ⁷ , los servicios web que conforman la Capa SOA para el intercambio de información con el entorno empresarial (SW11).
 <p>SIAI front-end</p>	Cliente desde el cual se accede al “front-end” de SIAI (SIAI web application).

3.6. Diagrama de componentes.

Los diagramas de componentes se realizarán por caso de uso, con el objetivo de representar un flujo completo y para ello se empleará el mismo caso de uso que se vio en la descripción y los restantes se encontrarán en el [Anexo 3](#).

⁷ SW10: Sistema de reportes predefinidos, convertor de formatos.

Capítulo 3: “Diseño e implementación del sistema”

“Módulo de Reportes del SIAI”

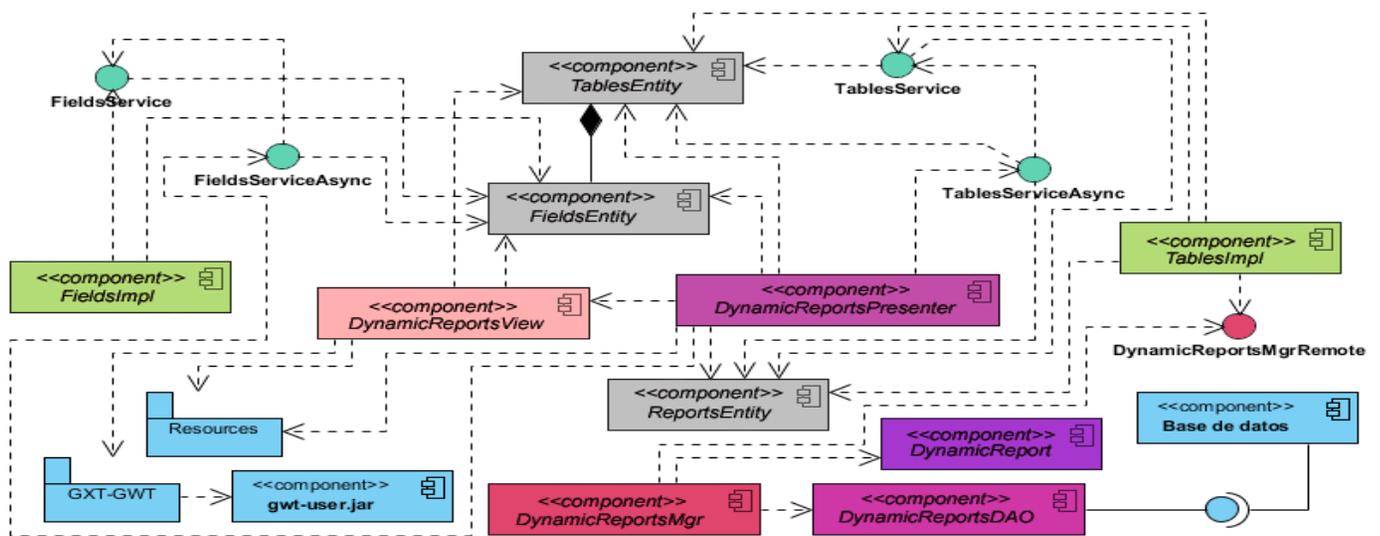


Fig. 16: Diagrama de componentes del CU Gestionar reporte, sección “Adicionar reporte”.

3.7. Conclusiones.

En este capítulo se mostró el diseño y la implementación del sistema, mostrando principalmente los diagramas de clases del diseño del módulo de reportes y el diagrama de paquetes por el cual se encuentra conformado. También se definió la arquitectura del sistema al igual que los patrones necesarios para lograr una aplicación más eficiente, fácil de entender y eficiente.

CAPÍTULO 4: PRUEBAS

4.1. Introducción.

El presente capítulo tiene como objetivo la descripción de las pruebas realizadas al sistema al igual que la muestra de los resultados obtenidos al ser aplicadas.

4.2. Pruebas realizadas al sistema.

En el desarrollo de software es muy común que se cometan algunos errores, los cuales nacen principalmente de la complejidad del producto a desarrollar. Mientras mayor dificultad exista, se multiplican las posibilidades de que persistan los errores aún después de terminado y entregado completamente el sistema, manifestándose principalmente por la manera en cómo el cliente lo emplea. Por tal motivo se hace necesaria la evaluación del sistema continuamente a medida que va pasando cada una de sus etapas de desarrollo, permitiendo detectar los problemas lo antes posible y lograr disminuir el impacto en el tiempo y esfuerzo del desarrollo. Para ello es necesario el apoyo en un grupo de técnicas las cuales permiten evaluar la calidad del producto a desarrollar.

Técnicas de evaluación estáticas.

Las técnicas de evaluación estática de los artefactos del desarrollo se les conocen de modo genérico por “Revisiones”. Estas pretenden detectar de forma manual defectos en cualquier producto del desarrollo. Generalmente las mismas se desarrollan de forma manual, refiriéndose que el producto (pueden ser requisitos, código, diseño y otros) se encuentra impreso en un papel y los revisores lo analizan mediante lecturas al mismo (24).

Dentro de las revisiones existentes para detectar mediante esta técnica algunos errores, se realizaron: Revisiones informales: Mediante la generación de opiniones entre los profesores del proyecto y el desarrollador sobre diferentes casos relacionados a la generación del módulo.

Revisiones formales: Mediante la reunión de algunos profesores del proyecto y el desarrollador, para analizar el producto a través de la lectura de diferentes documentos generados que se relacionan directamente con la generación del módulo.

Técnicas de evaluación dinámicas.

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

Las técnicas de evaluación dinámicas, también denominadas como pruebas del software generan entradas al sistema con el objetivo de detectar los fallos cuando el sistema ejecuta dichas entradas. Los fallos se observan cuando se detectan incoherencias entre las salidas obtenidas y las esperadas. Estas se aplican principalmente sobre código desarrollado hasta el momento en el cual se realizan las pruebas (24).

Las pruebas a desarrollar en las aplicaciones se agrupan principalmente en dos tipos, las de caja blanca y las de caja negra. Las de caja blanca se realizaron en el examen del código a evaluar, por su parte las de caja negra se realizaron sobre las interfaces con el objetivo de verificar el cumplimiento de cada una de sus funcionalidades obteniéndose como resultado si se cumplieron los requisitos funcionales solicitados. Para el desarrollo del módulo se emplearon pruebas tanto de caja blanca como de caja negra, con el apoyo de distintos criterios que permitieron verificar el estado de calidad de producto. Para el diseño de pruebas caja blanca se empleó el criterio de cobertura lógica “*Cobertura de Caminos*”, con el objetivo de verificar el funcionamiento de la secuencia desde la entrada hasta la salida del programa. Por otra parte para el diseño de las pruebas de caja negra se empleó el criterio de “*Particiones de equivalencia*”, con el objetivo de verificar las funcionalidades con valores tanto válidos como no válidos.

Para un mejor entendimiento de lo antes expuesto, se muestra un resumen en la Fig. 17.

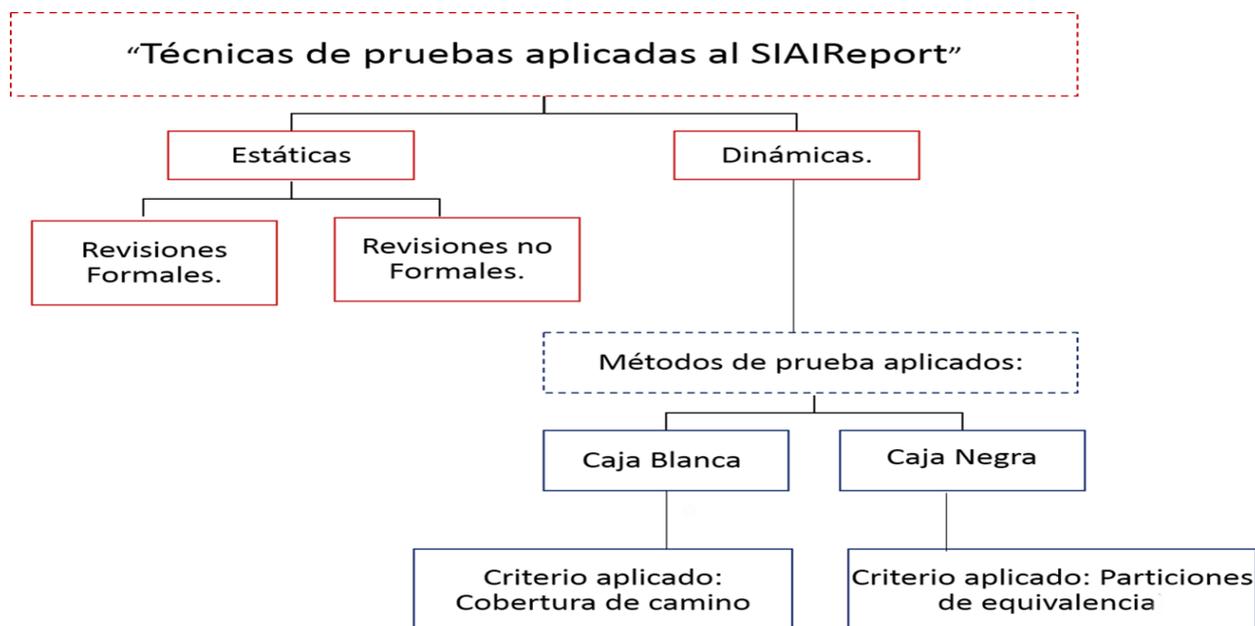


Fig. 17: Técnicas de pruebas empleadas al módulo.

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

4.2.1. Pruebas de unidad.

Con el objetivo de verificar la correcta codificación del módulo, se le realizan las pruebas de unidad, comprobando principalmente el funcionamiento de las diferentes funcionalidades de las clases por separado, para verificar si las mismas dan como resultado lo esperado por el usuario. Para realizar la prueba se empleó el método de caja blanca para el trazo de las funcionalidades y el de caja negra para obtener los resultados directamente desde la aplicación. Se realizaron 5 iteraciones en las que se detectaron un total de 135 errores, siendo resueltos los 135.

Tabla 7: Iteraciones de las pruebas de unidad realizadas al módulo.

Iteración	Cantidad de errores	Cantidad de errores resueltos	Cantidad de errores no resueltos
1	54	30	24
2	30	27	3
3	16	13	3
4	25	17	8
5	10	10	0

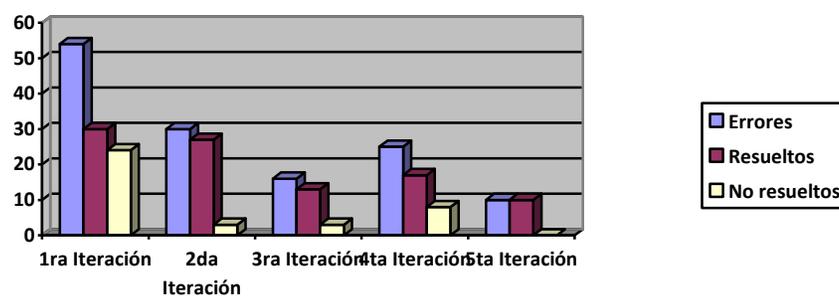


Fig. 18: Resultado gráfico de las iteraciones de las pruebas de unidad.

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

4.2.2. Pruebas de integración.

Luego de probada las funcionalidades del sistema por separado, se hizo necesario la prueba de todas las funcionalidades integradas, comprobándose principalmente si existirían conflictos entre las diferentes funciones. Para llevar a cabo las pruebas de integración se fueron integrando de manera incremental cada una de las funcionalidades para detectar los fallos que surgieran de manera más fácil. Como punto de partida se tomaron los requisitos funcionales los cuales sirvieron como guía en el desarrollo de la integración de manera incremental. Al aplicar las pruebas de integración no se detectaron problemas entre las funcionalidades por lo que se pudo montar todo el sistema de manera correcta.

En la Tabla 8 se puede observar la descripción de la prueba realizada en la integración del módulo de reportes con el SIAI.

Tabla 8: Descripción de la prueba de integración.

Módulo actual	Módulo integrado	Condiciones de ejecución	Funcionalidad	Escenarios de prueba	Resultado previsto	Resultado real
Sistema Integral de Análisis de Información.	Módulo de Reportes del Sistema Integral de Análisis de Información.	El usuario tiene que estar autenticado en el sistema.	Llamadas de entrada de LDI.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte.
			Llamadas de salida de LDI.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

					obtener el resultado del reporte especificado.	resultado del reporte.
			Reclamaciones de clientes.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte.
			Respuesta a reclamaciones de clientes.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte.
			Informe de servidores.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

					obtener el resultado del reporte especificado.	resultado del reporte.
			Casos por estado.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado y brinda la oportunidad de graficar el reporte.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte permitiendo que lo pueda graficar.
			Casos de fraude por tipos de reglas.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado y brinda la oportunidad de	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte permitiendo que lo pueda graficar.

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

					grafica el reporte.	
			Historial de casos.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte.
			Costos por concepto de fraude.	Obtener el resultado del reporte.	El sistema valida los datos y luego realiza una solicitud hacia la BD con el objetivo de obtener el resultado del reporte especificado y brinda la oportunidad de graficar el reporte.	El sistema valida los datos y luego realiza la solicitud hacia la BD y muestra al usuario el resultado del reporte permitiendo que lo pueda graficar.
			Adicionar reporte.	Crea un reporte dinámico y lo guarda en la	El sistema verifica la validez de los datos y luego	El sistema verifica la validez de los datos, luego genera la

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

				BD.	genera una entidad de reporte dinámico la cual es guardada en la BD.	entidad de reporte dinámico y lo guarda en la BD.
			Eliminar reportes.	Elimina el/los reporte(s) seleccionado(s).	El sistema verifica que elija al menos un reporte a eliminar, lanza un mensaje de confirmación, en caso de ser positivo el resultado elimina el/los reporte(s) seleccionado(s).	El sistema verifica que elija al menos un reporte a eliminar, lanza un mensaje de confirmación y en caso de ser positivo elimina el/los reporte(s) seleccionado(s).
			Modificar reportes.	Modifica valores del reporte seleccionado.	El sistema verifica que se seleccione un reporte para modificar, muestra los valores del reporte especificado, brinda la posibilidad de	El sistema verifica que se seleccione un reporte para modificar, muestra los valores del reporte especificado, brinda la posibilidad de

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

					agregar nuevas condiciones y campos de salida, verifica si existe al menos un campo de salida, genera la entidad del reporte dinámico y modifica los valores del reporte seleccionado en la BD.	agregar nuevas condiciones y campos de salida, verifica si existe al menos un campo de salida, genera la entidad del reporte dinámico y modifica los valores del reporte seleccionado.
			Resultado de reportes.	Obtiene el resultado del reporte seleccionado.	El sistema verifica si se seleccionó un reporte para mostrar su resultado, ejecuta la sentencia SQL que presenta el reporte y obtiene el resultado, muestra una lista en la cual se encuentra el resultado	El sistema verifica si se seleccionó un reporte para mostrar su resultado, ejecuta la sentencia SQL que presenta el reporte y obtiene el resultado, muestra una lista en la cual se encuentra el resultado obtenido.

Capítulo 4: “Pruebas”

“Módulo de reportes del SIAI”

					obtenido.	
--	--	--	--	--	-----------	--

4.2.3. Pruebas de sistema.

Para verificar si el sistema se integró de manera correcta se realizaron las pruebas de sistema, comprobando principalmente si se cumplían los requisitos funcionales definidos previamente, el correcto funcionamiento de las interfaces además del rendimiento y respuesta del sistema en condiciones límites y de sobrecarga. Para la generación de las pruebas se emplearon técnicas de caja negra, desarrolladas principalmente por la líder del proyecto por la parte de la UCI. Para un mejor manejo se definieron 4 casos de prueba principalmente, los cuales son de gran ayuda para el probador a la hora de realizar las pruebas. Se recomienda ver el [Anexo 4](#) para encontrar los casos de prueba generados que son aplicables al módulo.

4.3. Conclusiones.

En este capítulo se describieron las pruebas realizadas al sistema así como los resultados obtenidas de la misma, siendo satisfactoria tanto en las de integración como en la de sistema por los que el módulo se encuentra en perfecta condiciones de trabajo de acuerdo a los resultados obtenidos.

CONCLUSIONES

El desarrollo actual tanto de las telecomunicaciones como el informático, ha permitido un gran mejoramiento en términos de calidad. Actualmente no se concibe el mundo sin telecomunicación, lo que trae consigo que se genere gran cantidad de información en el uso de cada uno de los servicios que son brindados por empresas especializadas en el mundo de la comunicación. El presente trabajo está dirigido fundamentalmente al control de la información mediante reportes informáticos, con el fin de obtener de una manera más resumida y comprensible las grandes cantidades de información correspondiente al fraude telefónico con el que cuenta la empresa de ETECSA. Tras un estudio del problema que presentaba la empresa en cuanto al trabajo con los reportes de su información, se determinó llevar a cabo la generación de un módulo el cual le diera solución, reflejándose en el presente informe de forma resumida y detallada, los principales elementos de las principales etapas para el desarrollo que fueron necesarios para una correcta elaboración del producto.

Se puede expresar que se le dio solución de manera satisfactoria al objetivo general propuesto, contando con la creación de una aplicación para la web mediante el cual se obtiene de manera comprensible los reportes solicitados por el personal de antifraude de ETECSA. También se le brinda al usuario la oportunidad de generar reportes dinámicos de manera sencilla y sin mucho esfuerzo. La principal característica que se destaca es que el módulo se encuentra orientado al cliente y no al desarrollador como lo son gran cantidad de generadores de reportes a nivel mundial.

Otros resultados obtenidos son los siguientes:

1. Documentación de manera detallada de cada una de las etapas de desarrollo.
2. Generación de una base de datos de apoyo para el sistema SIAI.
3. Se demostró el por qué no se seleccionó una herramienta de reportes y se realizó el módulo de reportes del sistema SIAI.
4. Se realizaron pruebas a la solución propuesta, con el objetivo de verificar el correcto funcionamiento de la misma, obteniéndose resultados positivos que aprueban la incorporación del módulo al sistema general.

RECOMENDACIONES

Con la realización del siguiente trabajo, se recomienda la incorporación al módulo la funcionalidad de generar plantillas de forma dinámicas en la aplicación, con el objetivo de mostrar al usuario la información en plantillas personalizadas. También se recomienda la incorporación de nuevas funcionalidades relacionadas con otros módulos del sistema.

BIBLIOGRAFÍA

1. Definicion.de *Definicion.de* <http://definicion.de/modulo/>
2. **carlospes.com** carlospes.com *carlospes.com* <http://www.carlospes.com/minidiccionario/modulo.php>
3. **SIPEC** SIPEC *SIPEC* <http://sipec.sep.gob.mx/WebHelp/reportes/reporte.htm>
4. **Microsoft** Microsoft *Microsoft* <http://www.microsoft.com/en-us/download/details.aspx?id=1848>
5. **ComponentSource**. ComponentSource. *ComponentSource*. [En línea] 01 de 07 de 2012. [Citado el: 07 de 12 de 2012.] <http://www.componentsource.com/products/activeresports-6/index-es.html>
6. **JasperSoft Community**. JasperSoft Community. *JasperSoft Community*. [En línea] [Citado el: 28 de 11 de 2012.] <http://community.jaspersoft.com/>.
7. **msdn**. msdn. *msdn*. [En línea] 2005. [Citado el: 25 de 11 de 2012.] <http://msdn.microsoft.com/es-es/library/ms225593%28v=vs.80%29.aspx>.
8. **solis** solis *solis* <http://www.solis.org.br/projetos/labs/agata>
9. **O'Reilly** *O'Reilly* <http://www.onjava.com/pub/a/onjava/2006/07/26/deploying-birt.html>
10. **osmosislatina** osmosislatina *osmosislatina* <http://www.osmosislatina.com/lenguajes/uml/basico.htm>
11. **IDEF** IDEF *IDEF* <http://www.idef.com/IDEF0.htm>
12. **Bermeo Pérez, Fabián**. Fabian Bermeo Perez. *Fabian Bermeo Perez*. [En línea] 08 de 12 de 2010. [Citado el: 25 de 11 de 2012.] <http://fabianbermeop.blogspot.com/2010/12/metodologia-rup-desarrollo-de-software.html>.
13. **aliftitellez**. aliftitellez. *aliftitellez*. [En línea] 21 de 10 de 2011. [Citado el: 28 de 11 de 2012.] <http://aliftitellez.wordpress.com/2011/10/21/microsoft-visio/.14>.
14. **Visual Paradigm**. Visual Paradigm. *Visual Paradigm*. [En línea] 16 de 08 de 2010. [Citado el: 28 de 11 de 2012.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

15. **GENBETA.** GENBETA. *GENBETA*. [En línea] 23 de 06 de 2011. [Citado el: 25 de 11 de 2012.] <http://www.genbetadev.com/herramientas/eclipse-37-indingo-ya-disponible>.
16. **jboss.** jboss. *jboss*. [En línea] 14 de 12 de 2012. [Citado el: 20 de 01 de 2013.] <https://docs.jboss.org/author/display/AS71/Documentation>.
17. **Bravo, Edison.** oracle11g-newfeatures. *oracle11g-newfeatures*. [En línea] 08 de 2011. [Citado el: 20 de 01 de 2013.] <http://oracle11g-newfeatures.blogspot.com/p/nuevas-caracteristicas-de-rendimiento.html>.
18. **Vaughan, Daniel.** Ext GWT 2.0 Beginner's Guide. Olton : Packt Publishing Ltd, 2010. ISBN 978-1-849511-84-1.
19. **Zukowski, John.** *Programación Java 2*. Madrid : Artes Graficas Guemo; S, L., 2004. ISBN: 88-415-1559-X.
20. **Keith, Mike y Schincariol, Merrick.** *Pro JPA 2*. United States of America : Springer-Verlag New York, Inc, 2009. ISBN: 978-1-4302-1957-6.
21. **John Wiley & Sons., Buschmann, Frank.** *Pattern-Oriented Software Architecture*. 1996. ISBN/0471958697.
22. *Software Architecture Design Pattern in Java*. London : AUERBACH PUBLICATIONS, 2004. ISBN: 0-8493-2142-5.
23. Source Making. *Source Making*. [En línea] [Citado el: 23 de 04 de 2013.] http://sourcemaking.com/design_patterns.
24. **Juristo, Natalia, Moreno, Ana M y Vegas, Sira.** Unidad Docente de Ingeniería del Software (UDIS). *Unidad Docente de Ingeniería del Software (UDIS)*. [En línea] 17 de 10 de 2006. [Citado el: 02 de 05 de 2013.] http://is.ls.fi.upm.es/docencia/erdsi/Documentacion_Evaluacion_7.pdf.
25. Programación. *Programación*. [En línea] 14 de 09 de 2011. [Citado el: 04 de 05 de 2013.] <http://j2ee.ibsi.cl/desarrollo/java-j2ee/patron-de-diseno-dao/>.

26. Hernández, Yanoski Agneri Martínez and Olivares, José Rolando Lafaurie. Sistema para la generación de reportes en la plataforma alasGRATO. Ciudad Habana: s.n., 2008.
27. **Cortéz, Gloria y Casallas, Rubby.** Universidad de los Andes. *Universidad de los Andes.* [En línea] 1998. [Citado el: 07 de 04 de 2013.] <http://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf>.
28. **Kuchana, Partha.** *Software Architecture Design Pattern in Java.* London: AUERBACH PUBLICATIONS, 2004. ISBN: 0-8493-2142-5.
29. **Rondon Grados, Luis.** JAVA J2EE. *JAVA J2EE.* [En línea] 03 de 04 de 2009. [Citado el: 21 de 01 de 2013.] <http://luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>.
30. **Weathersby, Jason.** O'Reilly. *O'Reilly.* [En línea] 26 de 07 de 2006. [Citado el: 25 de 11 de 2012.] <http://www.onjava.com/pub/a/onjava/2006/07/26/deploying-birt.html>.
31. **Cervantes, Humberto.** SG. SG. [En línea] 02 de 2010. [Citado el: 26 de 11 de 2012.] <http://sg.com.mx/content/view/922>.
32. **Empresa de telecomunicaciones cubana.** Empresa de Telecomunicaciones de Cuba. [En Línea] 19 de octubre de 2011. [Citado el: 16 de 11 de 2012.] <http://www.etcса.cu/index.php?sel=nuestraempresa>.

ANEXOS

Anexo 1.

Tabla 9: Ficha del proceso “Generar reporte”.

Ficha de Proceso		
Proceso	“Generar reportes”	
Notación:	A52	
Entradas:		Información de comportamiento de reglas
		Información de comportamiento de alertas
		Información de comportamiento de casos
		Información de comportamiento de expedientes
Controles:	C2	Tiempo
	C5	Solicitud
Salidas:	O5	Reportes Personalizados
		Parámetros de notificación
Mecanismos:	M1	Personal ETECSA antifraude
Reglas del Negocio:	<ul style="list-style-type: none"> Los reportes pueden ser gestionados por todo el personal de antifraude. 	
Descripción del Proceso		
<p>El proceso hace alusión a las actividades que se realizan para poder gestionar parte de la información del negocio, recibiendo la información de comportamiento de varios procesos del sistema como son la Gestión de Reglas, Alertas, Casos y Expedientes, los cuales son analizados para realizar reportes de manera dinámica los cuales sean diseñados por el Personal de ETECSA de antifraude o de manera estática, los cuales se definirán en las entrevistas con el cliente a la hora de solicitar requisitos funcionales respecto a reportes que los mismos necesiten que sean estándares para la organización. Influyendo en cada momento los factores de tiempo y solicitud como formas de control.</p>		