

Universidad de las Ciencias Informáticas.

Facultad 2

Título: Estandarización de Jasper Server para el Centro

ISEC

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

Autora: Sachie Roca Gamboa

Tutor: MSc. Yadiel Ramos Rodríguez

La Habana, Junio del 2013.

“Año 55 de la Revolución”.



“Los seres humanos sin conocimientos, se asemejan a un meteoro viajando sin rumbo en el espacio sideral.”

Dedicatoria:

Dedico este trabajo de diploma a mis padres por su enorme apoyo y sacrificio durante todos estos años, por la educación que me han dado, por los valores y principios que me han inculcado en toda la etapa de mi vida. A ellos les debo todo lo que soy hoy.

A mi hermana por los buenos consejos que siempre me dio y su apoyo en los momentos difíciles de mi vida.

A mi familia y amistades que de una forma u otra estuvieron pendiente de mí en todo este tiempo en la universidad, que soportaron mi cambio de humor en esos momentos difíciles.

A Claudia que es como si fuera mi hermana, a Mercedes y familiares que son los que me han apoyado durante estos cinco años que he estado lejos de mi familia.

A mi novio Joel y familiares por darme su apoyo y ayuda incondicional, pues a pesar de estar poco tiempo a mi lado han sabido acogerme como si fuera parte de ellos.

Agradecimientos:

Agradezco a mis padres por todo el apoyo que me han dado a lo largo de estos cinco años, por confiar en mí, por darme la fuerza necesaria y decirme que yo sí podía llegar a ser una profesional en los momentos que dude.

A mi padrastro Agleidis por guiarme en buen camino y apoyarme en mis caprichos, por educarme y hacerme la persona fuerte que se muestra ante todos hoy aquí, la que no le tiene miedo a nada, la que nunca queda mal con sus compañeros y que siempre está atenta a todos para cuando la necesiten.

A mi papito lindo Heberto por comprenderme y ayudarme durante estos cinco años que de no ser por ti, por tu apoyo y tu comprensión hoy no estuviese aquí.

A mi mami linda Marlenis que es la mejor del mundo, que siempre me ha apoyado, dándome buenos consejos, guiándome por el camino correcto, todo lo que soy hoy y he logrado en mi vida mami, te lo debo a ti.

A mi hermana por siempre aconsejarme que estudiara, que debía trazarme metas y seguir adelante que yo iba a ser grande y tata gracias mira cogí tu consejo y me sirvió.

A mis abuelitos que creyeron en mí con orgullo que de cierto modo pensar en ellos, en la ilusión que siempre tuvieron de verme echa una profesional, me dio cada día fuerzas para seguir adelante y no defraudarlos.

A mis compañeros de aula que en los momentos más difíciles de la carrera me apoyaron y ayudaron en lo que necesitaba.

A Claudia, Anel, Rogelio, Josué, Alejandro, Dianet, Aile, Linny, Wendy, Rolando, Maylen y todos los demás que están y no están en la universidad en estos momentos para celebrar este preciado triunfo.

A mi novio por apoyarme noche tras noche y aguantar mis pesadeces de vez cuando, por levantarme el ánimo en esos días que no me sentía yo misma, por ayudarme a triunfar mis miedos, gracias.

A mi amigo y co - tutor Alexey que siempre me apoyo en lo que necesitaba, que me daba alientos de seguir adelante, que fue casi mi paño de lágrimas.

A mis profesores Darvis y Ariel que en momentos difíciles me ayudaron y alentaron para seguir adelante, que creyeron en mí y me aconsejaron siempre por buen camino.

A mi tutor por ayudarme cada vez que presentaba una duda y a todos aquellos profesores que estuvieron siempre atentos porque esta tesis estuviese bien.

A mis amigos de la Facultad 1 Alexis, Lairon, Alberto (alias el Indio), Yerandy porque cada vez que estaba estresada siempre estuvieron ahí para levantarme los ánimos y poder seguir adelante.

A todos Gracias.

14 de junio de 2013

Declaración de Autoría

Declaro que soy la única autora de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Sachie Roca Gamboa

Firma del Autor

MSc. Yadiel Ramos Rodríguez

Firma del Tutor

Resumen:

El presente trabajo tiene como objetivo diseñar y dotar de nuevas funcionalidades a la aplicación JasperServer para la generación de reportes en el centro ISEC de la Universidad de Ciencias Informáticas. Para ello se utilizó la librería JasperReports por las facilidades que brinda, así como todas aquellas de las que depende para su correcto funcionamiento. Se seleccionó la metodología Xtreme Programming (XP) como rectora del proceso de desarrollo obteniéndose una herramienta en los espacios de tiempo y costo esperados, que cumple con las características requeridas por el cliente y da solución a los problemas existentes dentro del centro con relación a la generación de reportes.

Palabras Clave: Librería, JasperReports, JasperServer.

Summary:

The present work aims to design and provide new functionality to the application JasperServer for reports generation in the center ISEC University Computer Science. We used this Seller JasperReports for the facilities it provides, as well as those on which it depends for its correct operation. XP methodology was selected as chancellor of the development process yielding a tool in the spaces of time and cost expected to comply with the characteristics required by the customer and provides a solution to the existing problems within the center with respect to the generation of reports.

Keywords: Library, JasperReports, JasperServer.

Índice

Introducción:	- 1 -
Estructura Capítular:	- 4 -
Capítulo 1. Fundamentación Teórica.....	- 5 -
1.1 Introducción	- 5 -
1.2 Sistemas Generadores de Reportes. Importancia en la actualidad	- 5 -
1.3 Ejemplos de Sistemas Generadores de Reportes	- 5 -
1.3.1 JasperServer	- 5 -
1.3.2 Crystal Reports:.....	- 7 -
1.3.3 Active Reports:	- 7 -
1.3.4 Oracle BI Publisher:.....	- 8 -
1.3.5 BIRT (Business Intelligence and Reporting Tools):.....	- 8 -
1.3.6 Selección de la Herramienta:.....	- 8 -
1.4 Herramientas de Desarrollo.....	- 9 -
1.4.1 Apache Tomcat 6.0	- 9 -
1.4.2 JasperServer 5.0	- 9 -
1.4.3 IReports 4.5.1	- 11 -
1.4.4 Java Development Kit (JDK) 1.6.0.....	- 12 -
1.5 Metodologías de Desarrollo.....	- 12 -
1.5.1 ¿Qué es una Metodología?	- 13 -
1.5.3 Rational Unified Process (RUP)	- 13 -
1.5.4 Feature Driven Development (FDD)	- 14 -
1.5.5 Xtream Programming (XP)	- 16 -
1.6 Lenguaje de Programación	- 19 -
1.6.1 Java	- 19 -
1.7 Gestores de Base de Datos	- 20 -
1.7.1 Oracle	- 21 -
1.7.2 PostgreSQL.....	- 23 -
1.7.3 MySQL	- 24 -
1.8 Conclusiones Parciales	- 25 -
Capítulo 2: Características del Sistema, Exploración y Planificación.....	- 26 -

2.1	Introducción	- 26 -
2.2	Propuesta del Sistema	- 26 -
2.3	Características no funcionales del Sistema	- 27 -
2.3.1	Usabilidad	- 27 -
2.3.2	Disponibilidad	- 28 -
2.3.3	Seguridad.....	- 28 -
2.3.4	Confidencialidad.....	- 28 -
2.3.5	Requisitos de Hardware:	- 28 -
2.3.6	Requerimientos de Software:	- 28 -
2.2	Usuarios relacionados con la aplicación.....	- 29 -
2.4	Exploración	- 29 -
2.4.1	Ciclo de vida de JasperReports-Server	- 30 -
2.6	Planeación	- 35 -
2.6	Conclusiones Parciales	- 41 -
Capítulo 3: Diseño, Implementación y Prueba		- 42 -
3.1	Introducción	- 42 -
3.2	Patrón de Arquitectura	- 42 -
3.2.1	Arquitectura Rest.....	- 42 -
3.3	Diseño.....	- 45 -
3.4	Implementación.....	- 48 -
3.5	Tareas Ingenieriles.....	- 54 -
3.6	Pruebas.....	- 56 -
3.7	Productización.....	- 59 -
3.8	Mantenimiento.....	- 59 -
3.10	Análisis de Factibilidad.....	- 60 -
3.10.1	Beneficios Tangibles e Intangibles	- 62 -
3.11	Visión Global del Trabajo Realizado.....	- 63 -
3.12	Conclusiones Parciales	- 64 -
Conclusiones		- 65 -
Recomendaciones		- 66 -
Referencias Bibliográficas.....		- 67 -
Glosario de Términos.....		- 69 -

Índice de Figuras:

ILUSTRACIÓN 1 FASES Y FLUJOS DE TRABAJOS EN RUP.....	- 14 -
ILUSTRACIÓN 2 PROCESO FDD.....	- 16 -
ILUSTRACIÓN 3 CICLO DE VIDA XP	- 18 -
ILUSTRACIÓN 4 PRINCIPALES COMPONENTES DE POSTGRESQL.....	- 23 -
ILUSTRACIÓN 5 PROPUESTA DEL SISTEMA.....	- 27 -
ILUSTRACIÓN 6 CICLO DE VIDA DE JASPERREPORTS	- 30 -
ILUSTRACIÓN 7 ARCHIVO .JRXML	- 31 -
ILUSTRACIÓN 8 DISEÑO DE CLASES QUE COMPONEN EL OBJETO JASPERDESIGN.....	- 33 -
ILUSTRACIÓN 9 PROCESO SEGUIDO EN LA FASE DE PLANEACIÓN.....	- 36 -
ILUSTRACIÓN 10 PRIMERA ITERACIÓN.....	- 39 -
ILUSTRACIÓN 11 SEGUNDA ITERACIÓN	- 40 -
ILUSTRACIÓN 12 CONFIGURACIÓN DE LA APLICACIÓN	- 45 -
ILUSTRACIÓN 13 NUEVO DIRECTORIO DE CARPETAS DE LA APLICACIÓN.....	- 46 -
ILUSTRACIÓN 14 CONFIGURACIÓN EL ARCHIVO LOG4J.PROPERTIES.....	- 49 -
ILUSTRACIÓN 15 CONFIGURACIÓN LDAP.....	- 49 -
ILUSTRACIÓN 16 CONFIGURACIÓN LDAP.....	- 49 -
ILUSTRACIÓN 17 CONFIGURACIÓN LDAP.....	- 50 -
ILUSTRACIÓN 18 CONFIGURACIÓN LDAP.....	- 50 -
ILUSTRACIÓN 19 CONEXIÓN A UNA BD	- 51 -
ILUSTRACIÓN 20 CONEXIÓN A UNA DETERMINADA BD.....	- 52 -
ILUSTRACIÓN 21 FUNCIÓN Y FORMULARIO DINÁMICO PARA ACCEDER A LA CLASE REDIRECT.PHP	- 53 -
ILUSTRACIÓN 22 PROGRAMAR EVENTO ONCLICK A UN LINK	- 54 -

Índice de Tablas:

TABLA 1 REQUERIMIENTOS DE HARDWARE.....	- 28 -
TABLA 2 REQUERIMIENTOS DE SOFTWARE (SERVIDOR).....	- 29 -
TABLA 3 REQUERIMIENTOS DE SOFTWARE (CLIENTE)	- 29 -
TABLA 4 USUARIOS RELACIONADOS CON LA APLICACIÓN.	- 29 -
TABLA 5 HU GENERAR REPORTE.....	- 37 -
TABLA 6 HU CONEXIÓN A UNA BD.....	- 38 -
TABLA 7 HU CONFIGURAR HERRAMIENTA.....	- 38 -
TABLA 8 HU AUTENTICAR POR DOMINIO.....	- 39 -
TABLA 9 PLAN DE ENTREGA	- 41 -
TABLA 10 ARQUITECTURA JASPERSERVER	- 44 -
TABLA 11 TARJETA CRC GENERAR REPORTE.....	- 48 -
TABLA 12 DATOS DE CONEXIÓN A UNA BD.....	- 52 -
TABLA 13 TAREA DE INGENIERÍA GENERAR REPORTE	- 54 -
TABLA 14 TAREA DE INGENIERÍA CONEXIÓN A UNA BD.....	- 55 -
TABLA 15 TAREA DE INGENIERÍA CONFIGURAR HERRAMIENTA	- 55 -
TABLA 16 TAREA DE INGENIERÍA AUTENTICAR POR DOMINIO.....	- 55 -
TABLA 17 CP AUTENTICAR	- 59 -

Introducción:

Con el pasar de los años las Tecnologías de la Información y las Comunicaciones (TIC) se van desarrollando y por ende perfeccionando, dando pie al surgimiento de los sistemas de información que hacen cada día un uso intensivo y extensivo de las tecnologías. Debido a que para muchas organizaciones se hace engorroso el manejo de grandes cantidades de información, aparecen los sistemas generadores de reportes facilitando el trabajo a las personas que harán uso de las mismas. La aparición de estas nuevas herramientas tienen el propósito de almacenar, recuperar y proveer todo tipo de documentos, los cuales han posibilitado reconocer su importancia como factor determinante del desarrollo para las instituciones que requieran de los mismos, estando obligadas, a hacer un uso cada vez más eficiente de todos sus recursos. Estos sistemas identificados como reportes, no son más que objetos que entregan información en un formato particular y que permiten realizar ciertas operaciones sobre ellos como imprimirlos, enviarlos por correo, guardarlos a un archivo, entre otras.

En la actualidad existen diferentes herramientas que tienen este tipo de propósito, entre ellas se encuentran: IReports, Crystal Reports, JasperServer, Oracle BI Publisher, entre otras. Estas aplicaciones tienen como característica común, la fácil generación de reportes en diversos formatos. Además es necesario conocer que prácticamente todas las que están soportadas para el lenguaje Java, utilizan iText, la cual es una librería de código abierto que soporta la generación de documentos HTML, RTF y XML, incluyendo documentos PDF. (1).

Debido a lo antes expuesto, el centro de Informatización de la Seguridad Ciudadana de la Universidad de las Ciencias Informáticas (ISEC), cuyo objeto social es facilitar los servicios de informatización asociada con aplicaciones de archivos y bibliotecas, cuenta con un grupo de proyectos de informatización en estas líneas. Este se encuentra definiendo un grupo de arquitecturas centralizadas que permitan la generación de proyectos de desarrollo de software con mayor agilidad, sin perder en calidad. Inicialmente los proyectos del centro una vez definidas las arquitecturas a utilizar para el desarrollo de las aplicaciones, integraban a las mismas la librería JasperReport, permitiendo esta la generación de documentos a diversos formatos. Incluir dicha librería no es eficiente debido a que requiere codificación extra para la generación de informes, conocimientos de programación para su uso, necesita la integración de forma manual con otras librerías para la generación de documentos de texto, gráficos y hojas de cálculo. Por lo que destina un grupo de recursos a la búsqueda de una aplicación

14 de junio de 2013

que genere reportes dinámicos, realizando un estudio previo para comprobar si admite integrarse con otras aplicaciones desarrolladas por el centro e indistintamente verificar su completo funcionamiento aparte de cualquiera de los aplicativos con los que se relacione.

Por lo planteado anteriormente se define como problema a resolver: ¿Cómo facilitar la generación de reportes y estadísticas para el Centro ISEC?

De acuerdo al presente problema, se define como *objeto de estudio*: los procesos de generación de reportes.

El *objetivo general* que persigue la investigación es: Configurar la herramienta JasperServer para que cumpla con los requerimientos asociados a la generación de reportes y estadísticas para los proyectos del centro ISEC.

Desglosándose el presente objetivo en los siguientes *objetivos específicos*:

- Caracterizar el marco teórico y conceptual de los métodos y herramientas para la generación de reportes.
- Implementar las funcionalidades faltantes en la herramienta JasperServer necesarias para la generación de reportes en el centro ISEC.
- Integrar dicha aplicación a nivel visual y tecnológico con las necesidades del centro ISEC.
- Documentar el desarrollo y el producto final del presente trabajo con vista a facilitar la construcción de futuras versiones.

Dentro del objeto de estudio se enmarca como *campo de acción*: la utilización de la herramienta JasperServer para la generación de reportes de los proyectos informáticos del centro ISEC.

La *idea a defender* plantea que con la estandarización de JasperServer para la generación de reportes se logrará mejorar la calidad y los tiempos de respuesta de los proyectos informáticos del centro ISEC.

Para dar cumplimiento a los objetivos propuestos se proponen las siguientes *tareas investigativas y prácticas*:

- Elaboración del marco teórico investigativo asociado con la generación de reportes para lograr un mejor entendimiento de las funcionalidades que brindan los mismos.
- Identificación de las herramientas existentes a nivel internacional que generen reportes dinámicos para seleccionar la más adecuada para el centro.

14 de junio de 2013

- Comparación de las herramientas existentes para enfatizar en las ventajas que presenta JasperServer sobre las demás.
- Investigación de la metodología, herramientas y lenguaje utilizados en el análisis, diseño e implementación de la herramienta JasperServer.
- Implementación de funcionalidades asociadas con la integración de JasperServer con las arquitecturas definidas para la generación de reportes por el centro ISEC.
- Implementación de funcionalidades asociadas con la conexión de JasperServer con gestores de bases de datos sobre la misma arquitectura de la aplicación para lograr que presente más de una conexión a una base de datos.
- Realización de un Manual de Usuario de la herramienta JasperServer una vez personalizada para dejar documentado todo el proceso realizado en la presente investigación.
- Realización de una Ayuda en Línea de la herramienta JasperServer una vez personalizada para que los usuarios que accedan a la aplicación tengan un mejor conocimiento de las funcionalidades que esta presenta.
- Despliegue de la aplicación en un servidor visible para todo el centro ISEC para integrar dicha herramienta con cualquier aplicación que la necesite.
- Elaboración de la documentación final del producto para la realización de futuras versiones.

Es por ello que se consideran como *posibles resultados*:

- Una aplicación mediante la cual se podrán producir estadísticas y reportes personalizados de forma dinámica y estática.
- Documentación del proceso y del producto, de manera que pueda continuarse el desarrollo de esta línea en el futuro.
- Información de retroalimentación sobre las funcionalidades de la aplicación dentro de la Universidad que pueda servir para su validación y publicación en Internet.
- La integración de la aplicación con cualquier aplicativo, de forma que pueda servir para cualquier entidad en el futuro.

Con el objetivo de alcanzar la solución deseada y de facilitar el desarrollo de la investigación, se utilizan los siguientes *métodos de investigación*:

Métodos empíricos:

Se utiliza el método de *análisis documental*: en la revisión de la literatura especializada para consultar la información necesaria en el proceso de investigación.

El método de la observación – externa – incluida: cuando una persona que forma parte del grupo de trabajo observa, guía y revisa de forma sistémica la investigación durante todo el proceso.

El método de *revisión de la documentación*: estudio de documentos y diferentes tipos de bibliografía, para seleccionar la información necesaria en la investigación.

Métodos teóricos:

Se utiliza el *método analítico – sintético*: que permite consultar la bibliografía especializada en cuanto al tema abordado e identificar elementos claves que contribuyan a la solución del problema científico planteado, permite sintetizar conceptos que ayudarán a comprender la solución del problema.

El método histórico – lógico: para el estudio crítico de los trabajos anteriores que constituyen referentes teóricos - prácticos en el tratamiento del problema planteado, y tomarlos como base de comparación con los resultados alcanzados.

Estructura Capitular:

Capítulo 1. Fundamentación Teórica: en él se analiza el concepto de reporte en cuestión; las herramientas existentes a nivel internacional y el aporte que puedan brindar durante la implementación; las metodologías de desarrollo del software y su papel primordial en el logro de un producto con calidad; para con estos conocimientos sintetizar en cómo se podrá mejorar la aplicación para la generación de reportes, de manera que cumpla con los requisitos planteados por el centro.

Capítulo 2. Exploración y Planificación: se hace énfasis en las características del sistema a desarrollar, la propuesta del mismo, se definen las iteraciones a realizar y la duración que presentarán; se describen también las historias de usuarios y la estimación de esfuerzo que presentan las mismas.

Capítulo 3. Diseño, Implementación, Prueba y Validación: según la metodología de desarrollo utilizada, este capítulo se refiere a las propias fases de la misma, definiendo la arquitectura del sistema y con ello patrones arquitectónicos empleados. Se procederá a abordar las historias de usuarios definidas y tareas de la ingeniería en cada iteración para luego hacer una revisión de las mismas. Se realizarán las pruebas pertinentes para dar cumplimiento a los objetivos propuestos por el cliente.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En la actualidad es muy común ver en las empresas un sistema que ayude a la generación de reportes, trayendo consigo la realización de nuevas herramientas dedicados a este tipo de tarea, por lo que el presente capítulo abordará un estudio del estado del arte acerca de los diferentes sistemas dedicados a la generación de reportes existentes en el ámbito internacional. Así como una breve explicación del porqué de su utilización y su necesidad actual. Además de explicar las herramientas, las base de datos con que la aplicación va a poder establecer una conexión, metodología y lenguaje a utilizar en la adaptación a JasperServer.

1.2 Sistemas Generadores de Reportes. Importancia en la actualidad

Los Sistemas generadores de reportes son herramientas que proveen una información necesaria para cualquier institución o persona que lo necesite, estos proporcionan la comunicación y el poder de análisis que muchos requieren para llevar a cabo los negocios a escala global. Tomando la información obtenida y presentándola en forma de resumen rutinario y de informes de excepción. Emplean modelos muy sencillos para presentar la información, facilitando una correcta y organizada documentación de lo que se requiera por lo que en la actualidad los mismos son de gran importancia. (2)

1.3 Ejemplos de Sistemas Generadores de Reportes

1.3.1 JasperServer

Aplicación web desarrollada en Java que permite generar reportes y acceder a ellos a través del explorador. Entre sus características encontramos que provee bastante flexibilidad y es muy fácil de usar. Se crean los reportes usando IReports, y cuando esté listo el archivo .jrxml, se puede subir a JasperServer. Los reportes pueden observarse a través de la web, y exportarse en diferentes formatos (Pdf, xls, etc.). Esta aplicación también permite trabajar con BI (Business Intelligence, denominado inteligencia empresarial, debido a que es un conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.) de manera bastante sencilla, usando un asistente para modificar la consulta MDX

14 de junio de 2013

gráficamente, solo se necesita importar un esquema Mondrian (un archivo xml) y luego ya tenemos una vista de análisis muy potente, capaz de generar gráficos estadísticos. (3).

Esta para su correcto funcionamiento utiliza diversos framework, por ejemplo: para la capa de presentación utiliza JQuery, Json y SpringMVC. Para la realización de template utiliza Tiles, la parte de seguridad es garantizada a través de Spring Security, la lógica de negocio con Spring, la capa de acceso a dato con Hibernate y para garantizar la generación de reportes utiliza JasperReport e iReport. Ofrece desde la versión 4.2 un API de servicios web basado en la arquitectura REST, permitiendo un acceso a los servicios de una forma mucho más simple que sus equivalentes tradicionales basados en SOAP. (4).

Utiliza internamente JasperReports que es la librería más popular de reportes de código libre en Java. Se puede incluir fácilmente en cualquier aplicación Java para la creación de sofisticados reportes, permitiéndonos crear subreportes, o reportes dentro de reportes. Es capaz de exportar los reportes a múltiples formatos, como PDF, XLS, RTF, HTML y XML. Además brinda soporte para los Sistemas Operativos Windows y Linux. (3).

Características principales de JasperServer:

1. Es el servidor de reportes de la suite.
2. Puede ser usado como aplicación independiente o desde otras aplicaciones.
3. Provee el punto de ingreso para los reportes y análisis de datos.
4. Posee una interfaz de usuario muy sencilla de usar y personalizable.
5. Se pueden ejecutar, programar y distribuir los reportes a usuarios y grupos.
6. Puede almacenar los reportes emitidos.
7. Administra los recursos compartidos.

Provee repositorio centralizado:

- ✓ La conectividad con los sistemas existentes de gestión de identidad para centralizar y asegurar informes y vistas de análisis.
- ✓ Informe de auditoría de acceso y uso para el cumplimiento.
- ✓ La seguridad de acceso granular hasta el nivel celular y la columna.

1.3.2 Crystal Reports:

Crystal Reports facilita la creación de informes simples y dispone también de herramientas poderosas necesarias para generar informes complejos o especializados. La flexibilidad del mismo no termina con la creación de informes, ya que éstos se pueden publicar en una variedad de formatos que incluyen Microsoft Word y Excel, correo electrónico e incluso en la Web. La elaboración avanzada de informes en Internet permite a otros miembros de su grupo de trabajo ver y actualizar informes compartidos en sus exploradores Web. (5)

Con Crystal Reports, se puede:

1. Aprovechar al máximo la elaboración de informes profesionales, a un precio atractivo para todo el mundo.
2. Capacitar al usuario final para que explore informes con parámetros y ordenación en informes.
3. Minimizar los recursos de TI y desarrollo en informes interactivos.
4. Desarrollar potentes aplicaciones Web híbridas de datos.
5. Ahorrar tiempo en el diseño de informes.
6. Incrustar informes de aspecto profesional en aplicaciones Java y .NET.
7. Adaptar su solución agregando herramientas de visualización y administración de informes.

1.3.3 Active Reports:

Es un componente de informes .NET para aplicaciones Windows Forms y de formularios Web. Entre sus características claves figuran la personalización, un rendimiento rápido, alta calidad y prestaciones multilingües: todas ellas contrastadas mediante su uso en decenas de miles de aplicaciones en todo el mundo. Admite exportaciones de datos a todos los formatos de archivo habituales, como PDF, Excel, RTF, TIFF, etc. Incluye un diseñador de informes Visual Studio .NET fácil de usar y una potente API. Los informes se crean dentro de Visual Studio .NET y se compilan directamente en el ejecutable.

Además se 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.

Presenta un diseño de página fijo ofreciendo a los desarrolladores una nueva forma de creación de informes en donde el formato es el objetivo principal. Excelente para crear facturas, informes,

formularios, folletos u otros informes altamente estructurados. El diseño de página continúa hace que sea fácil crear informes con tablas, referencias cruzadas y otras regiones de datos con diseños flexibles.

(6)

1.3.4 Oracle BI Publisher:

El generador de informes Business Intelligence de Oracle (BI Publisher) es una solución empresarial que permite crear, administrar y enviar todo tipo de documentos. Los usuarios finales pueden confeccionar fácilmente sus modelos de informes en el navegador Web o utilizando herramientas de escritorio cotidianas, lo que reduce enormemente el tiempo y el coste de elaboración, gestión y mantenimiento de informes. Al estar diseñado en código abierto, informáticos y programadores pueden crear modelos de datos compartibles prácticamente a partir de cualquier fuente de datos. BI Publisher es eficiente y escalable; puede generar decenas de miles de documentos por hora con el mínimo impacto en los sistemas de transacción. Está basado en el estándar W3C-XSL-FO, y es muy rápido. Ello permite manejar gran cantidad de entrada de datos y generar su salida en menos tiempo, utilizando niveles muy bajos de uso de CPU y de memoria. Genera documentos en varios formatos como: PDF, HTML, Excel, PowerPoint, RTF, Flash, eText (para EDI/EFT), CSV y XML. (7)

1.3.5 BIRT (Business Intelligence and Reporting Tools):

Sistema de reportes de código libre basado en Eclipse para aplicaciones Web, especialmente si están basadas en Java y J2EE. Tiene dos componentes principales, una herramienta para diseñar los reportes basada en Eclipse, y un componente que se puede agregar al servidor de la aplicación. Permite adicionar gráficos a la propia herramienta. Con BIRT, se puede incluir gran variedad de reportes en la aplicación, cumpliendo con las funcionalidades requeridas y generando reportes que cumplen con las normas de calidad impuestas. Posee variadas fuentes de datos que permiten al usuario obtener los mismos del lugar más indicado. Puede usarse en cualquier plataforma donde Eclipse corra. Tiene variados formatos de salida como HTML, PDF, WORD y XLS. (8)

1.3.6 Selección de la Herramienta:

Teniendo en cuenta lo descrito anteriormente tanto BIRT, Crystal Reports, Active Reports y Oracle BI Publisher son herramientas idóneas para elegir en la presente investigación. Aunque no es menos cierto que debido al previo conocimiento que presentan los usuarios finales sobre la librería JasperReport, se

selecciona finalmente JasperServer ya que esta incluye internamente dicha librería, brinda los reportes en menor tiempo de respuesta, presenta una arquitectura REST la cual permite agregar las funcionalidades faltantes, accediendo libremente al código fuente de la aplicación para cumplir con los requisitos previstos por el centro.

1.4 Herramientas de Desarrollo

Son muchas las herramientas reconocidas a nivel internacional para la generación de reportes, sin embargo, solo se hace alusión a las escritas en Java y de código libre. De ellas, se presenta un estudio previo del Apache Tomcat 6.0 y Java Development Kit 1.6, debido a que JasperServer coloca dichas aplicaciones por defecto. Además de la necesaria utilización del iReport para la creación de plantillas de reportes.

1.4.1 Apache Tomcat 6.0

Se desarrolla en un entorno abierto, participativo y publicado bajo la licencia Apache versión 2. Es un servidor Web que funciona como contenedor de servlets, desarrollado en código abierto por la Apache Software Foundation. Tomcat implementa las especificaciones de las tecnologías servlets Java y de páginas JSP. Puede funcionar como servidor Web por sí mismo. El hecho de que Tomcat fue escrito en Java, hace posible que funcione en cualquier sistema operativo que disponga de la máquina virtual Java (también se puede utilizar con XAMPP). (9) . Para el desarrollo de la aplicación se utilizará Apache 6.0 debido a que es la versión que soporta JasperServer 5.0.

Principales características:

1. Implementado de Servlet 2.5 y JSP 2.1.
2. Soporte para Unified Expression Language 2.1.
3. Diseñado para funcionar en Java SE 5.0 y posteriores.
4. Soporte para Comet a través de la interfaz Comet Processor.

1.4.2 JasperServer 5.0

Es una plataforma que ofrece capacidades de informes ricos, Inteligencia de negocios (BI) independiente y embebido de alto rendimiento. Una vez instalado propicia opcionalmente Apache Tomcat, Java y PostgreSQL en su sistema para que el servidor JasperReports esté listo para funcionar.

14 de junio de 2013

Presenta la librería más popular de reportes de código libre en Java. Esta librería se puede incluir fácilmente en cualquier aplicación Java para la creación de sofisticados reportes. Se utiliza además para crear archivos de salida que se emplearán en el futuro, como hojas de cálculo. (10)

Brinda las funcionalidades necesarias para dar cumplimiento a los requisitos impuestos. Además de los datos en forma de texto, es capaz de generar reportes profesionales incluyendo imágenes y gráficos, entre sus funcionalidades más importantes se encuentran:

- ✓ Provee de un diseño flexible de reportes.
- ✓ Es capaz de presentar los datos de forma gráfica o textual.
- ✓ Permite a los desarrolladores el suministro de datos de múltiples formas.
- ✓ Acepta datos de varias fuentes.
- ✓ Permite incluir marcas de agua.
- ✓ Puede generar subinformes.
- ✓ Es capaz de exportar los reportes a múltiples formatos.

Presenta una arquitectura cliente – servidor donde para acceder a las diversas capas de negocio y para la generación de documentos en los formatos PDF, HTML, CSV y Excel utiliza el framework Spring, para controlar la seguridad de la herramienta utiliza Spring Security, entre otros.

Debido a la utilización de la librería JasperReport, esta, también necesita de otras librerías para implementar algunas de sus funcionalidades, entre ellas:

Commons-digester.jar: La librería Common Digester incluye clases utilitarias que se usan para inicializar objetos de Java desde archivos XML. JasperReports aprovecha las ventajas del componente Digester del repositorio Jakarta Commons para implementar sus funcionalidades de parsear XML. Esta librería debe encontrarse en el CLASSPATH de la aplicación para que JasperReports funcione correctamente. (11)

Commons-collections.jar: Commons Collection es otro de los componentes de la suite de Jakarta Commons. Este componente brinda la funcionalidad de complementar el framework de colecciones de Java. JasperReports aprovecha las ventajas de este componente para implementar algunas de sus funcionalidades y lo incluye en su directorio de librerías.

Commons-logging.jar: La librería Common Logging incluye componentes que apoyan al desarrollo con el envío de datos a archivos de autenticación. JasperReports aprovecha las ventajas de este componente y lo incluye en su directorio de librerías.

14 de junio de 2013

Commons-beanutils.jar: La última librería que requiere JasperReports para compilar los reportes es Common BeanUtils. BeanUtils es una librería que ofrece wrappers fáciles de usar a través de Java reflection e introspection API. JasperReports la incluye en su directorio de librerías.

Ittext-2.1.7.js1.jar: Es la librería que se usa para generar y manipular PDF. Tiene además la habilidad de manipular documentos RTF, XML y HTML. JasperReports aprovecha las ventajas que ofrece iText para exportar los reportes a PDF y RTF.

Poi-ooxml-schemas-3.7.jar: Jakarta POI es una librería de clases de Java para crear y manipular varios formatos de Microsoft Office. JasperReports aprovecha las ventajas que ofrece POI para exportar los reportes a XLS (Excel). (12).

1.4.3 IReports 4.5.1

Herramienta visual que sirve para generar ficheros XML (plantillas de informe) que se puedan utilizar con la herramienta de generación de informes JasperReports.

Esta inicialmente se ha desarrollado en J++ por lo que su entorno de ejecución se limita a plataformas Microsoft. En sus últimas versiones se ha publicado una versión 100% Java con lo cual se limita la restricción existente. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, etc. (13).

La lista siguiente describe algunas de las características importantes de iReport:

- ✓ 100% escrito en JAVA y gratuito.
- ✓ Maneja el 98% de las etiquetas de JasperReports
- ✓ Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreportes.
- ✓ Soporta internacionalización nativamente.
- ✓ Navegador de la estructura del documento.
- ✓ Recopilador y exportador integrados.
- ✓ Soporta JDBC.
- ✓ Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).

- ✓ Incluye Wizard's (asistentes) para crear automáticamente informes.
- ✓ Tiene asistentes para generar los subreportes.
- ✓ Tiene asistentes para las plantillas.
- ✓ Facilidad de instalación.

Características y Beneficios

- ✓ Arquitectura de servidor flexible.
- ✓ Capa semántica para fuentes de datos relacionales y no relacionales.
- ✓ En las instalaciones, virtualizados, o Cloud (SaaS y PaaS) opciones de implementación.
- ✓ Los estándares abiertos, REST y SOAP basados en arquitectura de servicios web simplifica la integración de aplicaciones móviles y web.

Repositorio Centralizado

- ✓ La conectividad con los sistemas existentes de gestión de identidad para centralizar y obtener los informes de análisis y puntos de vista.
- ✓ Informar de acceso y auditoría de uso para el cumplimiento.
- ✓ Seguridad de acceso granular a nivel celular y columna.

1.4.4 Java Development Kit (JDK) 1.6.0

Se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje Java. Existen versiones del JDK para prácticamente todos los Sistemas Operativos y existen también distintos programas comerciales. Los JDK incorporan una herramienta de Debugger (detener la ejecución de un programa en la línea que se desee y poder conocer el valor de las variables en ese momento). Consta de una serie de aplicaciones y componentes, para realizar cada una de las tareas de las que es capaz de encargarse. Se trabajará con la versión 1.6 del JDK debido a que es la versión que soporta JasperServer 5.0. (14)

1.5 Metodologías de Desarrollo

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Las metodologías surgidas desde los 90 hasta la actualidad suelen tener otra mentalidad. Siendo conscientes de lo cambiante y amplio que es el mundo del software, una metodología debe ser lo suficientemente precisa

como para que todo el mundo la pueda seguir y sea de utilidad como pauta común, pero también debe ser lo suficientemente adaptable como para poder aplicarse en distintos proyectos, y lo suficientemente sencilla como para que no resulte muy compleja su utilización, pero lo adecuadamente completa y compleja como para que la utilización por parte del equipo sea provechosa. (15)

1.5.1 ¿Qué es una Metodología?

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. (16)

Se utiliza principalmente para:

- Facilitar la tarea de planificación
- Facilitar la tarea del control y seguimiento de un proyecto
- Mejorar la relación coste/beneficio
- Optimizar el uso de recursos disponibles
- Facilitar la evaluación de resultados y cumplimiento de los objetivos
- Facilitar la comunicación efectiva entre usuarios y desarrolladores

De acuerdo a lo antes planteado, se hizo un estudio de varias metodologías existentes, para encontrar cuál se adaptaría mejor a una aplicación de pequeña escala, con poco personal y requerimientos cambiantes. Para ello se analizaron: RUP (Rational Unified Process), FDD (Feature Driven Development) y XP (Xtream Programming).

1.5.3 Rational Unified Process (RUP)

El proceso unificado Rational (RUP) es un marco de trabajo de proceso de desarrollo de software iterativo, creado por Rational Software Corporation, una división de IBM desde 2003. RUP no es un proceso preceptivo concreto individual, sino un marco de trabajo de proceso adaptable, con la idea de ser adaptado por las organizaciones de desarrollo y los equipos de proyecto de software que seleccionarán los elementos del proceso que sean apropiados para sus necesidades.

RUP resultó de la combinación de varias metodologías y se vio influenciado por métodos previos como el modelo en espiral. Las consideraciones clave fueron el fallo de proyectos usando métodos monolíticos

del estilo del modelo en cascada y también la llegada del desarrollo orientado a objetos y las tecnologías GUI, un deseo de elevar el modelado de sistemas a la práctica del desarrollo y de resaltar los principios de calidad que aplicaban a las manufacturas en general al software.

El proceso de ciclo de vida de RUP se divide en cuatro fases llamadas Inicio, Elaboración, Construcción y Transición. Estas fases se dividen en iteraciones, cada una de las cuales produce una pieza de software demostrable. (17)

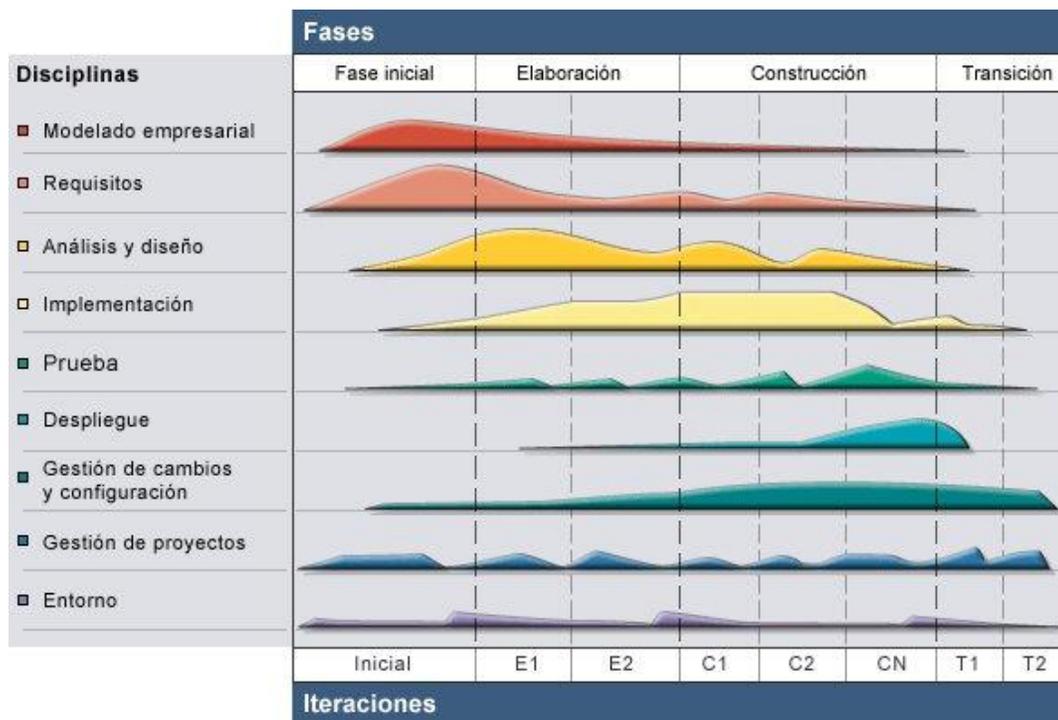


Ilustración 1 Fases y Flujos de Trabajos en RUP

La metodología RUP es poco factible para ser utilizada en proyectos pequeños, pues puede que no sea posible cubrir los costos de dedicación del equipo de profesionales, ya que tendrían que invertir una gran cantidad de horas y esfuerzo para hacer las cosas según lo que dicen los procesos.

1.5.4 Feature Driven Development (FDD)

Es un enfoque ágil para el desarrollo de sistemas. Dicho enfoque no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñado para trabajar con otras actividades de desarrollo de software y no requiere la utilización de ningún

14 de junio de 2013

modelo de proceso específico. Además, tiene en cuenta aspectos de calidad durante todo el proceso e incluye un monitoreo permanente del avance del proyecto. Al contrario de otras metodologías, FDD afirma ser conveniente para el desarrollo de sistemas críticos.

Los principios de FDD son pocos y simples: Se requiere un sistema para construir sistemas si se pretende escalar a proyectos grandes. Un proceso simple y bien definido trabaja mejor. Los pasos de un proceso deben ser lógicos y su mérito inmediatamente obvio para cada miembro del equipo. Vanagloriarse del proceso puede impedir el trabajo real. Los buenos procesos van hasta el fondo del asunto, de modo que los miembros del equipo se puedan concentrar en los resultados. Los ciclos cortos, iterativos y orientados por rasgos son mejores.

Hay tres categorías de rol en FDD: roles claves, roles de soporte y roles adicionales. Los seis roles claves de un proyecto son:

1. Administrador del proyecto, quien tiene la última palabra en materia de visión, cronograma y asignación del personal.
2. Arquitecto jefe (puede dividirse en arquitecto de dominio y arquitecto técnico).
3. Manager de desarrollo, que puede combinarse con arquitecto jefe o manager de proyecto.
4. Programador jefe, que participa en el análisis del requerimiento y selecciona rasgos del conjunto a desarrollar en la siguiente iteración.
5. Propietarios de clases, que trabajan bajo la guía del programador jefe en diseño, codificación, prueba y documentación, repartidos por rasgos.
6. Experto de dominio, que puede ser un cliente, patrocinador, analista de negocios o una mezcla de todo eso.

FDD consiste en cinco procesos secuenciales durante los cuales se diseña y construye el sistema. La parte iterativa soporta desarrollo ágil con rápidas adaptaciones a cambios en requerimientos y necesidades del negocio. Cada fase del proceso tiene un criterio de entrada, tareas, pruebas y un criterio de salida. (18)

14 de junio de 2013

en tareas de programación y se asignan a los programadores para ser implementadas durante una iteración

Roles XP: los roles de acuerdo con la propuesta de Beck son:

Programador: el programador escribe las pruebas unitarias y produce el código del sistema.

Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuales se implementan en cada iteración centrándose en apoyar mayor valor al negocio.

Encargado de pruebas (tester): ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para las pruebas.

Encargado de seguimiento (tracker): proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (coach): es el responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las practicas XP y se siga el proceso correctamente.

Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (big boss): es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

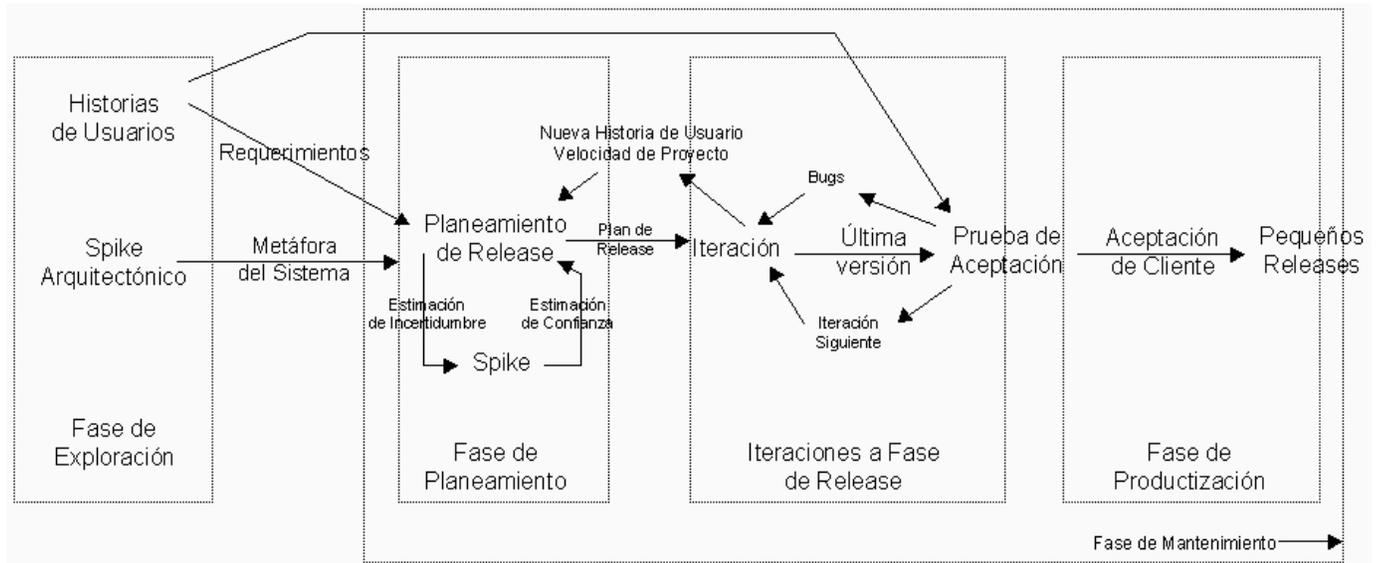


Ilustración 3 Ciclo de vida XP

Selección de la Metodología:

Para la selección de la metodología de desarrollo a utilizar, se hace necesario analizar el entorno en el que se construirá el producto. En primer lugar es necesario destacar que los usuarios finales son los propios desarrolladores del centro ISEC, pues son los que tienen la necesidad de un componente que los abstraiga de toda la lógica relacionada con reportes, lo cual significa que en todo momento se estará interactuando con los mismos.

Además, se hace necesario una solución rápida, pues la herramienta formará parte de los cimientos del sistema de ISEC y por lo tanto deberá estar funcional desde el momento en que el mismo se vaya a utilizar. La misma será sometida a constantes pruebas por parte de los desarrolladores, lo cual significa que se le estará dando mantenimiento al código a la vez que se desarrolle el resto de las iteraciones identificadas. Lo antes descrito evidencia cómo las características del presente trabajo se inclinan a adoptar las prácticas y valores de la metodología XP por lo cual se seleccionó la misma como rectora del proceso de desarrollo del software. Los pasos planteados por XP fueron moldeados a las características especiales del desarrollo. Se adoptaron los valores de XP comunicación, simplicidad, retroalimentación, coraje y respeto. La comunicación entre el equipo de desarrollo y los clientes será maximizada y efectuada de forma directa e interpersonal con el fin de evitar los problemas y errores causados por la mala adopción de la misma. Se realizará un trabajo tan simple como sea posible, favoreciendo la

comunicación, reduciendo el código sin utilizar y garantizando la calidad. Se estará constantemente midiendo el sistema para conocer cuánto se acerca a las funcionalidades necesarias mediante pruebas automáticas y por parte de los usuarios. Los valores antes mencionados contribuirán a enfrentar con coraje cada cambio en los requerimientos y refactorizaciones al sistema. Todo esto implica respeto hacia la organización y las personas cuya vida cambiará con el componente que se está proponiendo.

1.6 Lenguaje de Programación

1.6.1 Java

Fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos se agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red. (20). Las características principales que nos ofrece Java son:

1.- Lenguaje Simple: Se le conoce como lenguaje simple porque viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de java por eso utiliza determinadas características de C++ y se han eliminado otras.

2.- Orientado a Objeto: Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras en estructuras encapsuladas es más fácil su manipulación.

3.- Distribuido: Permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

4.- Robusto: Es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

5.- Seguro: La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.

14 de junio de 2013

6.- Indiferente a la arquitectura: Java es compatible con los más variados entornos de red, cualesquiera sean estos desde Windows 95, Unix a Windows Nt y Mac, para poder trabajar con diferentes sistemas operativos. Es muy versátil ya que utiliza byte-codes que es un formato intermedio que sirve para transportar el código eficientemente o de diferentes plataformas (Hardware - Software).

7.- Portable: Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java.

8.- Interpretado y compilado a la vez: Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.

9.- Multihebra o Multihilos: Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multa hilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo.

10.- Dinámico: El lenguaje java es muy dinámico en la fase de enlazado, sus clases solamente actuaran en medida en que sean requeridas o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red.

11.- Produce Applets: En java se pueden crear aplicaciones independientes y applets. Independientes porque se pueden comportar como cualquier programa escrito en cualquier lenguaje. Por otra parte los applets considerados pequeños programas, tienen la capacidad de ejecutar funciones muy complejas.

12.- Alto rendimiento: Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.

1.7 Gestores de Base de Datos

Para dar cumplimiento a unas de las tareas requeridas por el centro, JasperServer debe ser capaz de conectarse a diversos Gestores de Base de Datos por lo que se exponen a continuación los utilizados para cumplir con dicho objetivo:

1.7.1 Oracle

Una base de datos Oracle es una colección de datos tratada como una unidad. El propósito general es almacenar y recuperar información relacionada. Oracle puede abrir y utilizar solo una base de datos en cualquier punto y momento. (21)

Es un sistema gestor de base de datos con característica objeto-relacionales, que pertenece al modelo evolutivo de SGBD. Sus características principales son las siguientes:

- ✓ Entorno cliente / servidor.
- ✓ Gestión de grandes bases de datos.
- ✓ Usuarios concurrentes.
- ✓ Alto rendimiento en transacciones.
- ✓ Sistema de alta disponibilidad.
- ✓ Gestión de la seguridad.
- ✓ Compatibilidad.
- ✓ Contestabilidad

Ventajas de Oracle

1. Oracle es el motor de base de datos relacional más usado a nivel mundial.
2. Puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador.
3. Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
4. El software del servidor puede ejecutarse en multitud de sistemas operativos.
5. Oracle es la base de datos con más orientación hacía Internet.

La estructura lógica de Oracle esta determinada por:

- ✓ Uno o más *tablespaces* (área lógica de almacenamiento).
- ✓ Los objetos de los distintos *esquemas* (colecciones de objetos).

Las estructuras de almacenamiento lógico, incluyendo tablespaces, segmentos y extensiones, dictan la forma de utilización del espacio físico. Los objetos del esquema y las relaciones entre ellos forman el diseño relacional de la base de datos.

Tablespaces

Una base de datos se divide en unidades de almacenamiento lógico llamadas tablespaces. Un tablespace se usa para agrupar estructuras lógicas relacionadas.

Esquemas y objetos del esquema

Un esquema es una colección de objetos de la base de datos. Los objetos del esquema son estructuras lógicas que hacen referencia directa a datos de la base de datos (tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, clusters y enlaces con otras bases de datos).

Bloques de datos, extensiones y segmentos

Los bloques de datos Oracle son las estructuras lógicas de más bajo nivel. Una base de datos se almacena en bloques de datos, que se corresponden con un número específico de bytes de espacio en disco. Una extensión es un número específico de bloques de datos contiguos, que se emplea para almacenar un tipo de información específico. Un segmento es un conjunto de extensiones reservadas para una determinada estructura lógica (por ejemplo, segmentos de datos o segmentos de índices).

El servidor Oracle provee de *control de accesos discrecional*, es decir, acceso restringido a la información basado en privilegios.

Oracle gestiona la seguridad de la base de datos usando:

- ✓ Usuarios y esquemas de la base de datos.
- ✓ Privilegios.
- ✓ Roles.
- ✓ Ajustes de rendimiento y cuotas.
- ✓ Límites sobre los recursos.
- ✓ Auditoría.

Cada usuario tiene un dominio de seguridad, que determina cosas como:

- ✓ Acciones (privilegios y roles) disponibles para el usuario.
- ✓ Cuotas sobre tablespaces.
- ✓ Límites en los recursos del sistema.

1.7.2 PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. A continuación se muestra un gráfico que ilustra de manera general los componentes más importantes en un sistema PostgreSQL. (22)

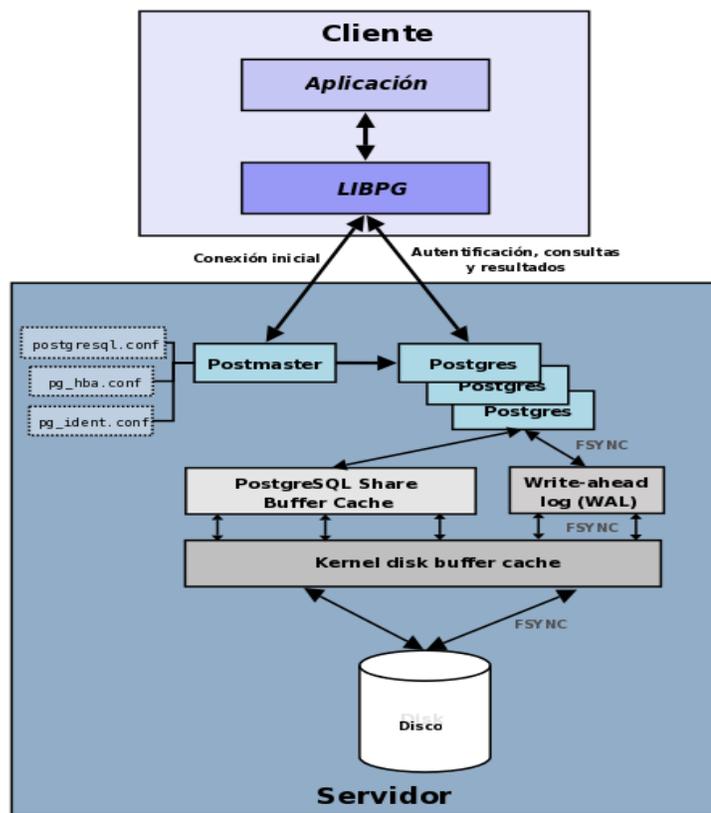


Ilustración 4 Principales Componentes de PostgreSQL

- **Aplicación cliente:** Es una aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.

- **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos postgres:** Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)
- **Kernel disk buffer cache:** Caché de disco del sistema operativo
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione

1.7.3 MySQL

Es un sistema de gestión de bases de datos relacional, bajo la licencia GPL de GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (23).

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.

14 de junio de 2013

- ✓ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas

- ✓ Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- ✓ Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✓ Facilidad de configuración e instalación.
- ✓ Soporta gran variedad de Sistemas Operativos
- ✓ Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- ✓ Conectividad y Seguridad.

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

Debido a que JasperServer puede integrarse con cualquier gestor de base de datos, no necesariamente tiene que ser con los antes mencionados, solamente basta con agregarle el driver de conexión con dicho gestor y la forma de configuración del mismo.

1.8 Conclusiones Parciales

En el presente capítulo se analizaron los conceptos principales asociados a la investigación para seleccionar las herramientas de trabajo, las librerías de clases y las metodologías de desarrollo adecuadas para la construcción del producto final. Después de estudiada la bibliografía sobre las herramientas de generación de reportes existentes, se decidió escoger la herramienta JasperServer para adaptarla a las necesidades que se presentan en el centro ISEC. Se trabajará con las herramientas Apache Tomcat 6.0, iReport 4.5 y el JDK 1.6. Se establecerán las conexiones a los gestores de base de datos Postgres, MySQL y Oracle. Además se estudiaron las metodologías pertenecientes a los diferentes enfoques de desarrollo de software, para arribar a la conclusión de que el proceso que más se adapta al presente trabajo es el de XP, eligiéndose como metodología rectora del proceso de desarrollo.

Capítulo 2: Características del Sistema, Exploración y Planificación

2.1 Introducción

En este capítulo se describen las nuevas funcionalidades que serán incluidas en el sistema. Se hace alusión a las fases de Exploración y Planificación, propias de la metodología de desarrollo utilizada, donde se confeccionan las historias de usuarios importantes para cada iteración definida por el equipo de desarrollo.

2.2 Propuesta del Sistema

JasperServer inicialmente solo presenta conexión a la BD Postgres, exporta reportes a formatos HTML, XML, RTF y PDF. Solo admite la entrada al sistema con dos tipos de usuarios, generando reportes y gráficos estáticos. Cuenta con diversas librerías las cuales son las encargadas de ejecutar todas las acciones que se requieran para cumplir con diversas funciones como: crear, exportar y almacenar reportes, permitiéndoles a los usuarios un mejor manejo de la información.

Debido a que la presente investigación tiene como objetivo dotar de nuevas funcionalidades a dicha aplicación. Se propone integrarle una serie de funcionalidades para que cumpla con las acciones relacionadas con la generación de reportes dinámicos, además de lograr que sea segura y fiable para el usuario, se procura exportar los reportes a otros formatos como Flash, ODT, ODS, DOCx y XLSX, para facilitar su análisis y almacenamiento fuera del ámbito de la aplicación. Podrá contar con dos tipos de usuarios donde uno es un usuario normal y el otro es administrador, en dependencia del tipo de usuario que se autentique este tendrá acceso a las funcionalidades que pueda realizar. Ejemplo de esto el administrador puede configurar la herramienta, crear, eliminar o administrar una determinada carpeta dándole a estas los permisos requeridos para un determinado usuario. Además de que podrá realizar las acciones que realiza un usuario normal de la aplicación y establecerá la conexión a la base datos que se desee utilizar en el entorno que se utilice dicha herramienta, mientras que un usuario normal solo puede imprimir, almacenar o exportar un determinado reporte y generar gráfico. Además se añadirá la conexión a LDAP de forma que cualquier institución o centro pueda utilizar dicha herramienta con solo establecer sus datos en el fichero adecuado. Ver Ilustración 6.

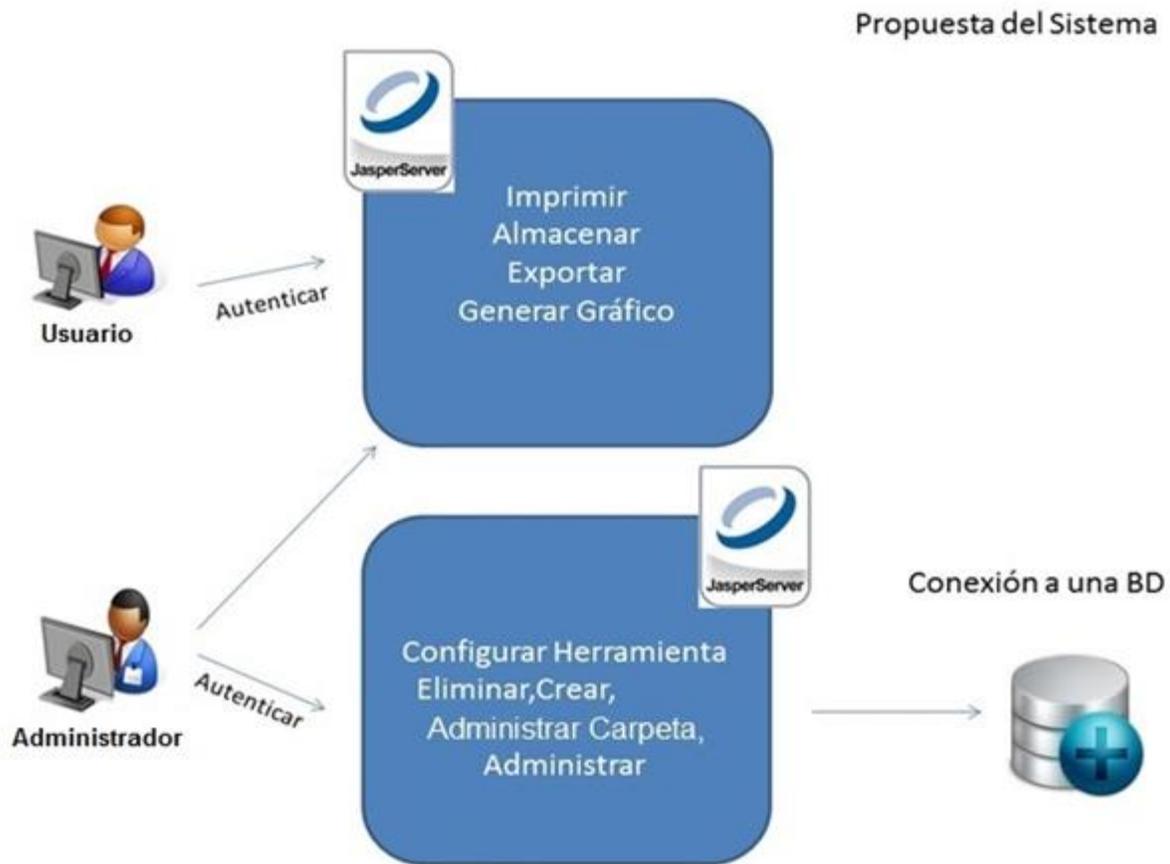


Ilustración 5 Propuesta del Sistema

2.3 Características no funcionales del Sistema

2.3.1 Usabilidad

Se requiere una preparación previa para interactuar con el sistema, aunque el manejo de la aplicación es sencillo, permitiendo una fácil comprensión por el usuario.

2.3.2 Disponibilidad

Una vez que el usuario se autentique en el sistema, tendrá el acceso garantizado a las funcionalidades que le correspondan según su rol, siempre que los datos introducidos en la autenticación sean los correctos y que la conexión con el servidor esté disponible.

2.3.3 Seguridad

El acceso a las funcionalidades estará determinado por los permisos establecidos por el administrador según el tipo de usuario. Autenticarse será obligatorio para el uso de cualquier servicio de la aplicación. Las contraseñas de cada usuario se almacenarán con el uso de un algoritmo de encriptación que imposibilite su lectura, para ello utiliza el framework Spring Security.

2.3.4 Confidencialidad

La información manejada en el sistema estará protegida de acceso no autorizado y cada usuario con acceso, sólo debe acceder a los datos que le correspondan por su rol y cuenta registrada. El rol de administrador tendrá acceso a todas las funcionalidades del sistema, mientras que el cliente, solo contará con algunos servicios para generar cualquier tipo de reporte.

2.3.5 Requisitos de Hardware:

Servidores	Especificaciones
Servidor de Aplicaciones Web	<ol style="list-style-type: none"> RAM: 256 MB o superior. Disco Duro: 80 GB o superior. UPS: 1.
Servidor de Base de Datos	<ol style="list-style-type: none"> RAM: 1 GB o superior. Disco Duro: 80 GB o superior. UPS: 1.

Tabla 1 Requerimientos de Hardware

2.1.6 Requerimientos de Software:

Servidores	Especificaciones
Servidor de Aplicaciones Web	<ol style="list-style-type: none"> Sistema Operativo: Multiplataforma.

	2. Servidor Web: Apache 6.0 o superior.
Servidor de Base de Datos	1. Sistema Operativo: Multiplataforma 2. Sistema Gestor de Base de Datos: PostgreSQL 9.1, MySQL 5.1.18 y Oracle.

Tabla 2 Requerimientos de Software (Servidor)

Cliente	Especificaciones
Cliente Web	✓ Navegador Web: Cualquier navegador Web.

Tabla 3 Requerimientos de Software (Cliente)

2.2 Usuarios relacionados con la aplicación

Se define como usuarios relacionados con la aplicación a aquellos que interactúan de una u otra forma para generar cualquier tipo de reporte y obtienen un resultado de todos los procesos que se ejecutan en ella.

Usuario	Justificación
Administrador	Es la persona encargada de gestionar todo tipo de acción que se requiera para lograr que se genere cualquier tipo de reporte. En caso de existir algún problema en el JasperServer, el administrador podrá solucionarlos sin ningún problema. Este puede darle acceso a cuantos usuarios quieran acceder a dicha aplicación, así como efectuar cualquier cambio ya sea de diseño o programación de la misma.
Usuario	Es la persona que realiza cualquier tipo de actividad relacionada para generar un reporte sin tener privilegios a efectuar algún cambio a la aplicación.

Tabla 4 Usuarios relacionados con la aplicación.

2.4 Exploración

Para iniciar la producción de un producto de software es preciso asegurarse de que es factible y posible hacerlo. Se debe poseer confianza en que las herramientas elegidas ayudarán a la culminación del trabajo. Se debe creer que una vez que el código se haga, éste puede utilizarse cada día. Cada miembro del equipo debe confiar en sus propias habilidades y en las habilidades de los otros.

La fase de exploración ayuda a resolver todos estos conflictos. Es en ella donde los clientes plantean a grandes rasgos las historias de usuario que son de interés para el producto. Durante la misma, los desarrolladores interactúan con las herramientas a utilizar durante todo el proceso, exploran activamente las posibilidades de arquitectura y experimentan los límites de las tecnologías a utilizar.

Durante la fase de exploración el equipo de desarrollo procedió a realizar un estudio en profundidad en cuanto a las herramientas y librerías de desarrollo seleccionadas para explorar las posibles soluciones a generar. (24).

2.4.1 Ciclo de vida de JasperReports-Server

Para un mejor entendimiento de las librerías de desarrollo seleccionadas, se estudiaron los estados por los que pasa un reporte en el proceso de generación.

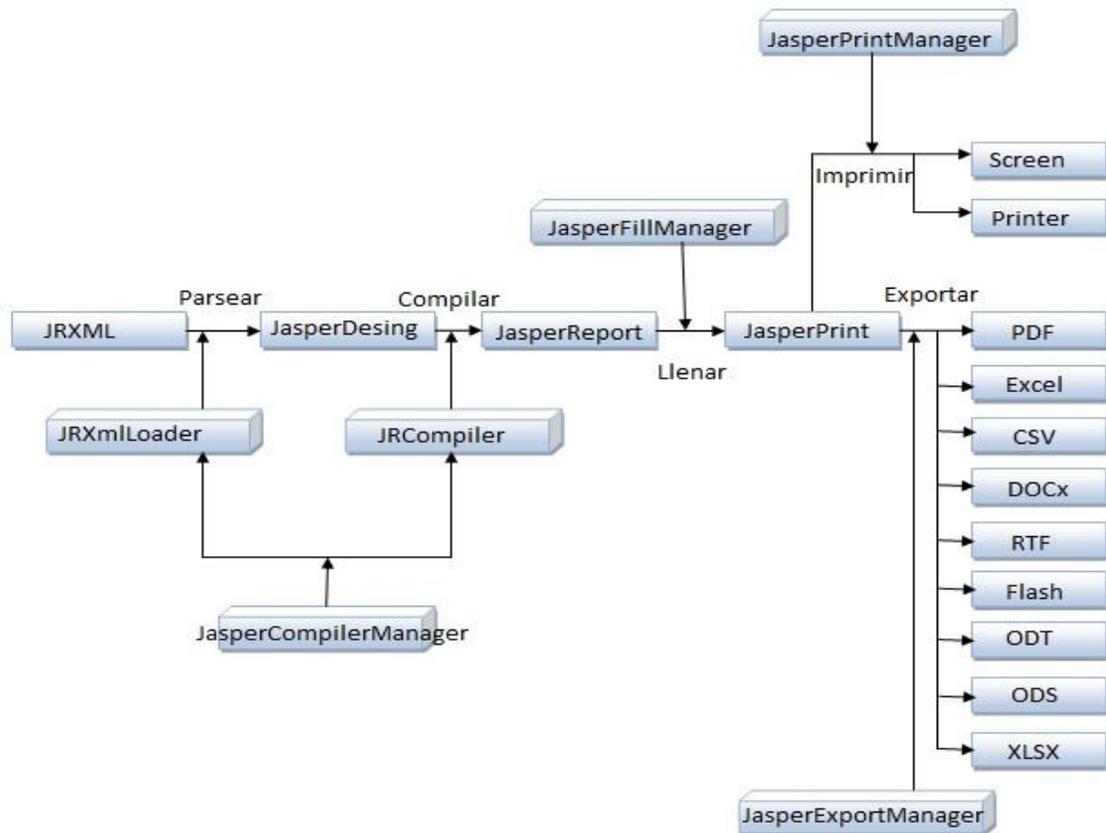


Ilustración 6 Ciclo de vida de JasperReports

Como se muestra en la figura y se explicó en el capítulo anterior, el proceso de generación de reportes parte de un archivo *.jrxml*, el cual pertenece a la familia de XML basado en el DTD de JasperReports y que contiene el diseño del reporte a generar. El DTD de JasperReports es el siguiente:

```
<! DOCTYPE jasperReport PUBLIC "-//JasperReports/DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">
```

Un ejemplo de archivo *.jrxml* se muestra en la Ilustración 8:

```
- <!--
  Created with iReport - A designer for JasperReports
-->
- <jasperReport name="simple_template" columnCount="1" printOrder="Vertical" orientation="Portrait" pageWidth="595" pageHeight="842"
columnWidth="535" columnSpacing="0" leftMargin="30" rightMargin="30" topMargin="20" bottomMargin="20" whenNoDataType="NoPages"
isTitleNewPage="false" isSummaryNewPage="false">
  <property name="ireport.scriptlethandling" value="2"/>
  <property name="ireport.encoding" value="UTF-8"/>
  <import value="java.util.*/>
  <import value="net.sf.jasperreports.engine.*/>
  <import value="net.sf.jasperreports.engine.data.*/>
- <background>
  <band height="0" isSplitAllowed="true"> </band>
</background>
- <title>
  <band height="50" isSplitAllowed="true"> </band>
</title>
- <pageHeader>
  <band height="50" isSplitAllowed="true"> </band>
</pageHeader>
- <columnHeader>
  <band height="30" isSplitAllowed="true"> </band>
</columnHeader>
- <detail>
  <band height="100" isSplitAllowed="true"> </band>
</detail>
- <columnFooter>
  <band height="30" isSplitAllowed="true"> </band>
</columnFooter>
- <pageFooter>
  <band height="50" isSplitAllowed="true"> </band>
</pageFooter>
- <lastPageFooter>
  <band height="50" isSplitAllowed="true"> </band>
</lastPageFooter>
- <summary>
  <band height="50" isSplitAllowed="true"> </band>
</summary>
</jasperReport>
```

Ilustración 7 Archivo *.jrxml*

Este ejemplo ilustra los principales elementos de un archivo *.jrxml*, todos son opcionales con la excepción del elemento raíz `<jasperReport>`. Cada uno de estos elementos contiene como hijo un elemento `<band>`. Las bandas contienen los datos que se muestran en el reporte. La función de cada uno de los elementos presentados es la siguiente:

Title. Esta es la primera sección del reporte, generada solo una vez durante el proceso de llenado del reporte y ubicada al comienzo del documento resultante.

PageHeader. Aparece en la parte superior de cada página del documento resultante.

ColumnHeader. Aparece al inicio de cada columna definida en el reporte

Detail. En esta sección es donde se encuentran los datos principales que contendrá el reporte. Para cada tupla del origen de datos, esta sección será generada.

ColumnFooter. Aparece al terminar cada columna.

PageFooter. Aparece en la parte inferior de cada página.

Summary. Esta sección es generada sólo una vez por reporte y aparece al final del documento generado, pero no es necesariamente la última sección.

Haciendo uso de la clase `JRXmlLoader` el archivo *.jrxml* es parseado y convertido en un objeto `JasperDesign`. Dicho objeto mantiene la estructura del reporte descrita anteriormente, pero en forma de objetos. Brinda la posibilidad de insertar, modificar o eliminar cualquier elemento contenido en el reporte. El objeto `JasperDesign` representa la estructura del reporte a través de dos clases, `JRBaseReport`, que mediante una jerarquía de clases contiene todos los elementos del reporte, dígame las bandas mencionadas anteriormente y éstas a su vez los distintos elementos que pueden contener como textos estáticos, textos dinámicos, gráficos, tablas, entre otros. `JasperDesign` hereda directamente de `JRBaseReport` y por lo tanto contiene todos los elementos del mismo. La otra clase sería `JRDesignDataSet` que contiene las expresiones, parámetros, campos, variables y grupos que componen el reporte. `JasperDesign` contiene un elemento de tipo `JRDesignDataSet`. La relación descrita se puede apreciar en la Ilustración 9.

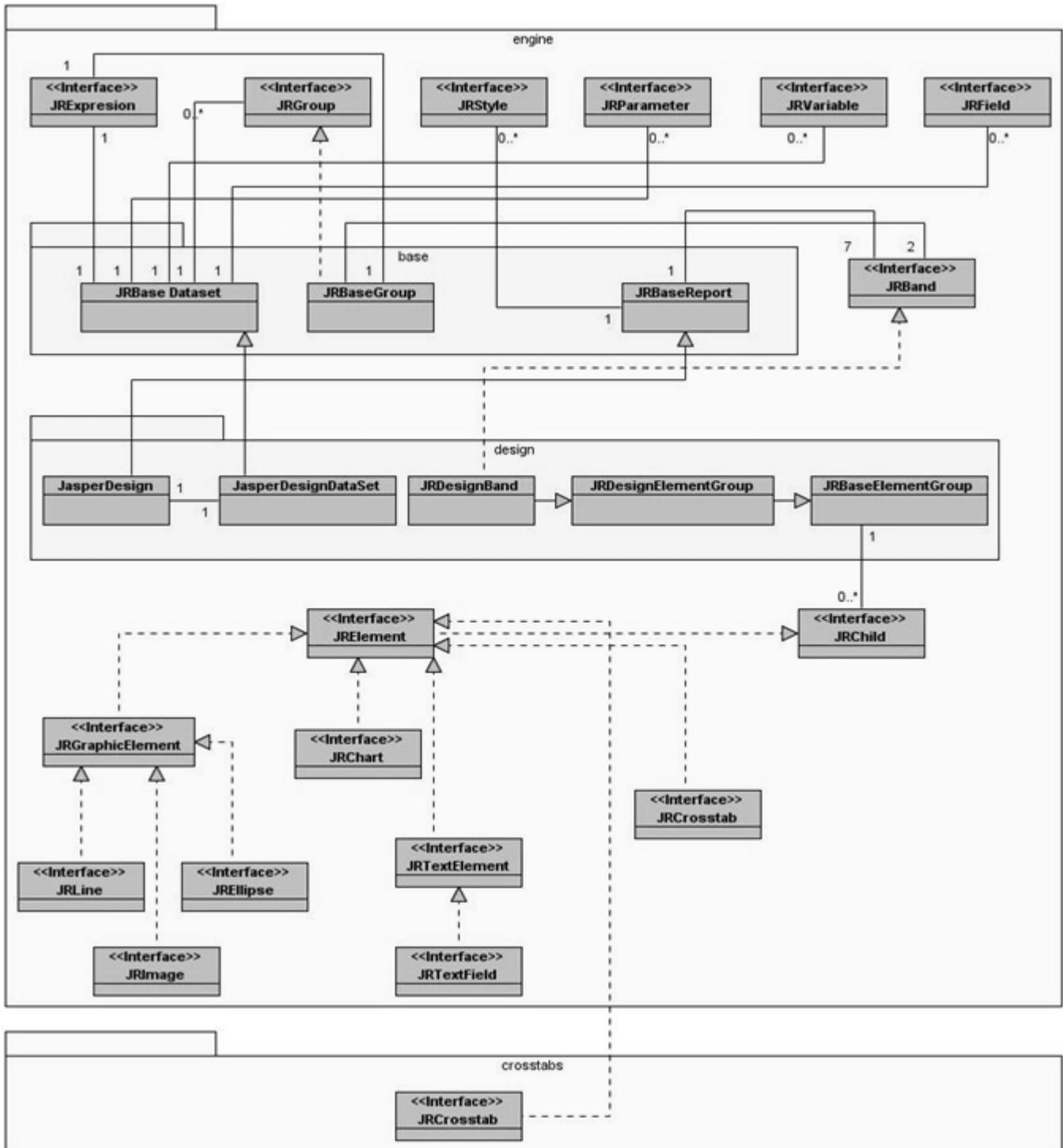


Ilustración 8 Diseño de Clases que componen el objeto JasperDesign

2.5 Historias de Usuarios

Las Historias de Usuarios: son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas. Las historias de usuario se descomponen en tareas de programación y se asignan a los programadores para ser implementadas durante una iteración. (25).

Durante esta fase los clientes realizaron estudios del levantamiento de requisitos del centro ISEC para derivar de los mismos las historias de usuario, obteniendo los siguientes resultados:

Generar Reporte: Al seleccionar la opción para generar reporte, se debe procesar la lista de datos haciendo uso de un origen de datos generalizado que posibilite mostrar textos con estilo en el reporte, generar el reporte atendiendo a un formato previo definido y mezclado con el diseño original y exportar el mismo a los formatos PDF, XLS, Excel, CSU, DOCx, ODS, RTF, ODT o Flash en dependencia de la selección del usuario.

Exportar reporte a un formato: El sistema permite al usuario la opción de exportar el reporte a PDF, CSU, DOCx, RTF, Flash, ODT, ODS, XLSX.

Generar gráfico: Al seleccionar la opción para generar un gráfico, se debe interpretar la información enviada por el usuario y generar un gráfico que puede ser de los siguientes tipos: pastel, pila, barras y área, en dependencia de la selección del usuario, para luego insertarlo como un elemento de una banda específica del reporte.

Conexión a una Base de Datos: Al seleccionar la opción para conectarse a una base de datos determinada, se interpreta la información introducida por el usuario y con solo agregar el driver de la base de datos deseada el sistema permite conectarse a la misma.

Conexión a la BD Postgres: El sistema le permite al usuario conectarte a la base de datos Postgres con solo agregar el driver de conexión para la misma.

Conexión a la BD Oracle: El sistema le permite al usuario conectarte a la base de datos Oracle con solo agregar el driver de conexión para la misma.

Conexión a la BD MySQL: El sistema le permite al usuario conectarte a la base de datos Postgres con solo agregar el driver de conexión para la misma.

Configurar Herramienta: El sistema le permite al administrador del sistema la opción de configurar la herramienta a su gusto o adaptarla de tal manera que sea más fácil para el usuario.

Agregar Carpeta: El sistema le permite al administrador la opción de Agregar una carpeta.

Eliminar Carpeta: El sistema le permite al administrador la opción de Eliminar una determinada carpeta.

Copiar Carpeta: El sistema le permite al administrador la opción de Copiar una determinada carpeta.

Autenticar por dominio: A la hora de autenticarse, el sistema debe ser capaz de darle acceso a cualquier usuario de un dominio determinado, donde antes debe ser configurado por el programador del sistema en la clase LDAP de la herramienta para poder acceder satisfactoriamente a la aplicación.

Administrar Usuarios: El sistema le permite al administrador la opción de Administrar Usuarios. Donde puede ver o crear usuarios, dando las opciones de Activarlos, Desactivarlos o Eliminarlos. Les agrega a los usuarios que determine las funciones necesarias ya sea para administrar, o ser un usuario anónimo o normal.

Cortar Carpeta: El sistema le permite al administrador cortar una determinada carpeta.

Permisos de Carpeta: El sistema le permite al administrador darle los permisos requeridos que presentara un usuario sobre una determinada carpeta.

Propiedades de Carpeta: El sistema le permite al administrador ver las propiedades que posee una determinada carpeta.

Administrar Funciones: El sistema le permite al administrador agregarle funciones a los usuarios creados, funciones como ROLE_AMINISTRATOR, ROLE_USER.

Generar reporte dinámico: El sistema le permite al usuario cargar cualquier reporte para exportarlo a un determinado formato o hacer consultas sobre ese reporte para filtrar parámetros específicos.

2.6 Planeación

La intención de esta fase es establecer un acuerdo entre los clientes y los desarrolladores en el menor tiempo para que la mayor cantidad de historias de usuario puedan ser realizadas. Para ello se realizó el juego de planeación con el objetivo de maximizar el valor del software producido a partir de la puesta en producción de las características más importantes lo antes posible, en el mismo participan en conjunto desarrolladores y clientes con el fin de determinar rápidamente el alcance de la versión a construir; esto se logra además gracias a tener al cliente siempre disponible, lo cual facilita la interacción continua entre los involucrados. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación

de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. (26). A partir de las especificaciones anteriores se procedió a seguir el proceso representado por la Ilustración 5.

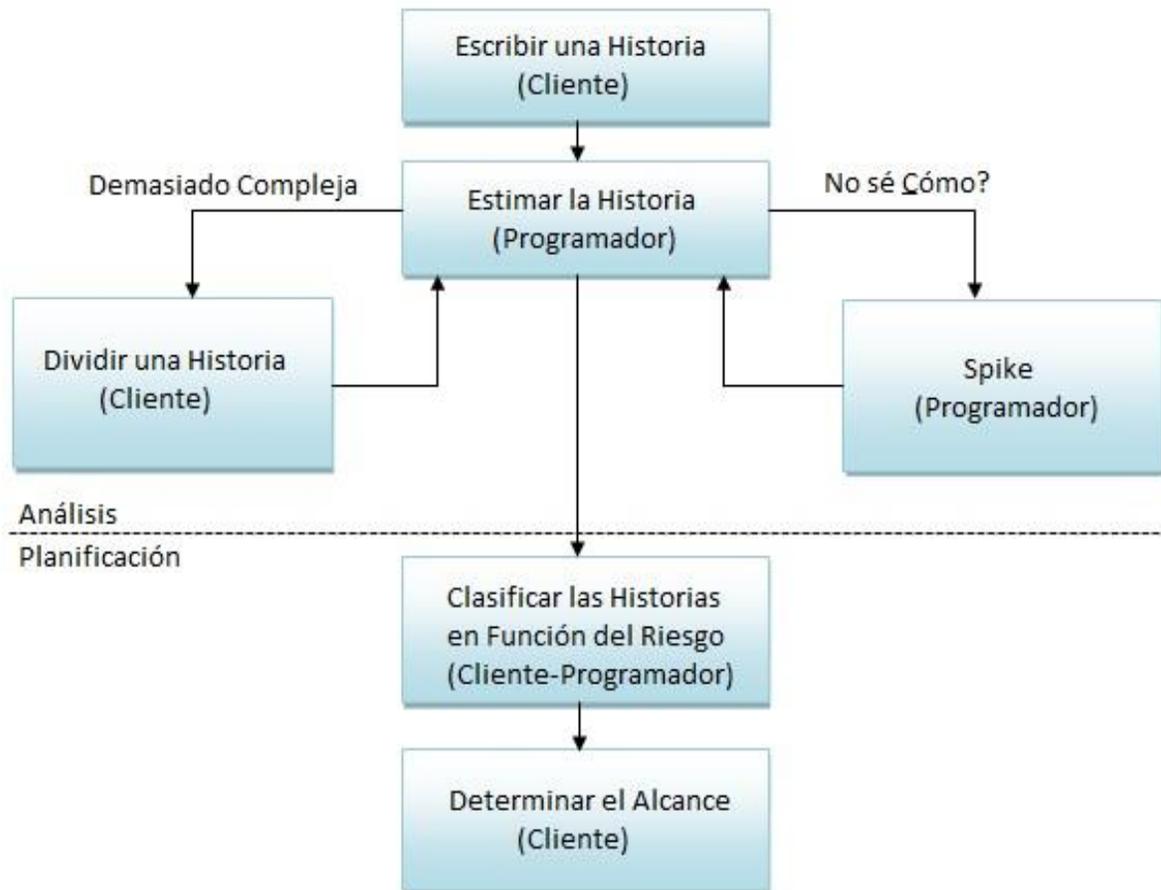


Ilustración 9 Proceso seguido en la fase de Planeación.

Ya el cliente escribió las historias de usuario, por lo tanto, es tarea del programador realizar la estimación de las mismas. Teniendo en cuenta que por primera vez se enfrenta a una tecnología de este tipo, se hizo necesario para algunas historias de usuarios anteriores realizar un spike para tener una noción a grandes rasgos de la complejidad de las tareas asociadas a la historia de usuario y lograr una mejor estimación del desarrollo de las mismas. El uso del spike dio como resultado que las historias 1,4, y 12 serían demasiado complejas realizarlas como un todo, se le explicó al cliente que en ambos casos se engloban funcionalidades que por sí solas pueden necesitar un gran esfuerzo de implementación y la

necesidad de dividirlos. El cliente procedió entonces a realizar la división. Luego de la división, ya se estuvo en condiciones de realizar una estimación de las nuevas historias de usuario, lo cual se llevó a cabo a partir del conocimiento existente sobre las tecnologías a utilizar y las características de las mismas. Es válido aclarar que dicha estimación es necesaria ir adaptando a través del proceso de desarrollo, puesto que para su realización no se tuvo un punto de partida; sin embargo se espera que la repetición de dicho proceso en el futuro arroje resultados más reales a partir de la experiencia adquirida.

Las historias más importantes son aquellas que contienen un mayor valor para el negocio. Se debe tener cuidado con la secuencia de historias basadas en dependencias técnicas, pues la mayoría de las veces estas dependencias son menos importantes que el valor. Las historias de usuario de mayor valor para el negocio, según las necesidades del cliente, son aquellas que incluyen funcionalidades imprescindibles para seguir todo el proceso de generación de reportes, aun cuando los mismos no cuenten con elementos finales como gráficos y formatos. Al analizar el riesgo técnico de las historias de usuario se arribó a la conclusión de que todas presentaban riesgo técnico, pues requerían un profundo conocimiento por parte del desarrollador de la lógica asociada a los reportes así como de las herramientas técnicas a utilizar. Sobre las dependencias entre ellas, se concluyó que se partiría de la historia “generar reporte”, para a través de la misma construir todas las demás funcionalidades.

Posteriormente se procedió a realizar un plan de entregas entre desarrolladores y clientes basado en los datos obtenidos en el proceso anterior. Una vez que el desarrollador estimó la historia de usuario y proporcionó al cliente un tiempo estimado de construcción de la misma; el cliente estuvo en condiciones de decidir cuánto tiempo debe durar cada iteración a realizar y cuáles historias se implementarán en cada una de ellas. Los puntos estimados se refieren a días, teniendo en cuenta que la mayoría de las historias tienen una duración menor a una semana.

Una vez obtenidos todos estos datos, se presentan las HU con prioridad Alta, para ver las demás [ver Anexos 1:](#)

Tabla 5 HU Generar Reporte

Historia de Usuario:	
Número: 1	Usuario: Desarrollador.
Nombre de Historia: Generar Reporte.	

14 de junio de 2013

Prioridad del negocio: Alta.	Riesgo del desarrollo: Alto.
Puntos Estimados: 5	Iteración asignada: 1
Descripción: Al seleccionar la opción para generar reportes, el sistema debe procesar la lista de datos y generar el reporte.	
Observaciones: La lista de datos debe ser procesada como un origen de datos de valor para JasperServer.	

Tabla 6 HU Conexión a una BD

Historia de Usuario:	
Número: 4	Usuario: Desarrollador.
Nombre de Historia: Conexión a una Base de Datos.	
Prioridad del negocio: Alta.	Riesgo del desarrollo: Alto.
Puntos Estimados: 5	Iteración asignada: 2
Descripción: Al seleccionar la opción para conectarse a una base de datos determinada, se interpretará la información introducida por el usuario y con solo agregar el driver de la Base de Datos deseada el sistema permite conectarse a la misma.	
Observaciones:	

Tabla 7 HU Configurar Herramienta

Historia de Usuario:	
Número: 8	Usuario: Desarrollador.
Nombre de Historia: Configurar herramienta.	
Prioridad del negocio: Alta.	Riesgo del desarrollo: Alto.
Puntos Estimados: 5	Iteración asignada: 2
Descripción: El sistema le permite al administrador del sistema la opción de configurar la herramienta a su gusto o adaptarla de tal manera que sea más fácil para el usuario.	
Observaciones:	

Tabla 8 HU Autenticar por dominio

Historia de Usuario:	
Número: 12	Usuario: Desarrollador.
Nombre de Historia: Autenticar por dominio.	
Prioridad del negocio: Alta.	Riesgo del desarrollo: Alto.
Puntos Estimados: 5	Iteración asignada: 2
Descripción: El sistema debes ser capaz de conectarse a cualquier dominio haciendo uso del LDAP	
Observaciones:	

Se identificaron dos iteraciones. La primera iteración, con duración de 5 semanas, se propone construir la infraestructura necesaria para generar los reportes de todo el sistema de ISEC, que incluye las funcionalidades básicas para el paso del reporte por todo su ciclo de vida y su presentación al usuario en el formato requerido.

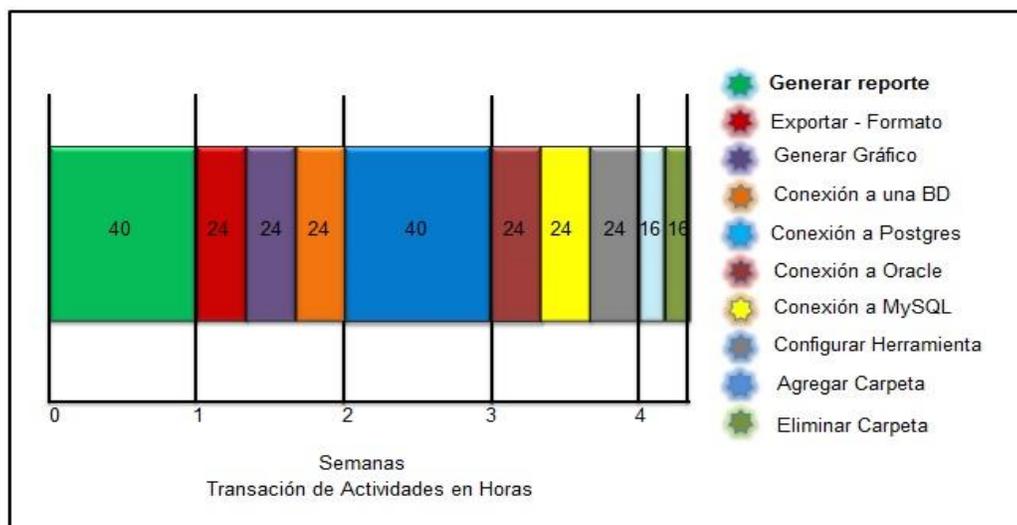


Ilustración 10 Primera Iteración

La segunda iteración, con duración de 4 semanas, se propone agregar al componente todas las funcionalidades restantes para generar los reportes de todo el sistema de ISEC.

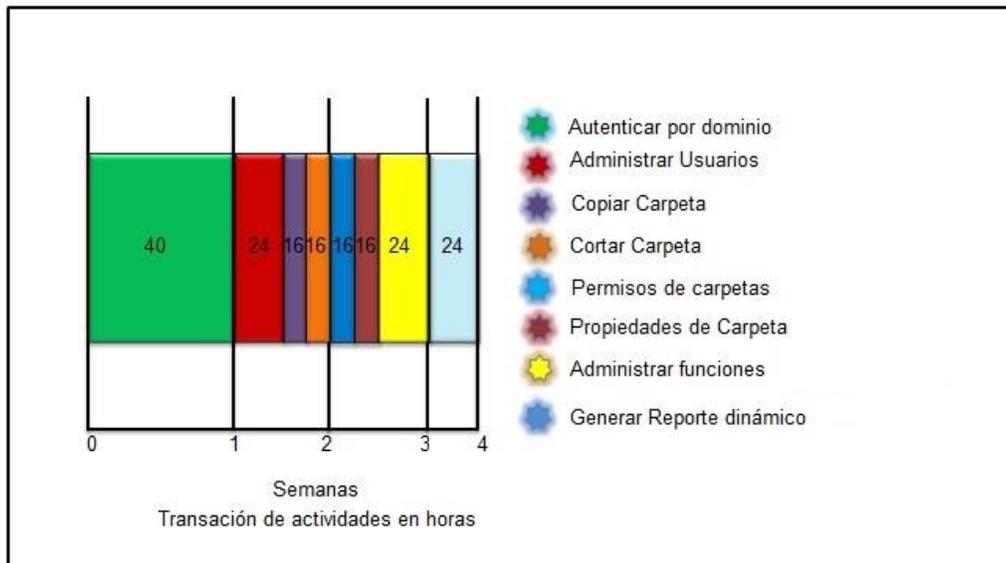


Ilustración 11 Segunda Iteración

El plan de entrega se presenta en la siguiente Tabla:

Historia de Usuario	Tiempo Estimado (horas laborables)	Iteración Asignada	Entrega Asignada
Generar Reporte	40	1	1
Exportar reporte a un formato	24	1	1
Generar Gráfico	24	1	1
Conexión a una BD	40	1	1
Conexión a la BD Postgres	24	1	1
Conexión a la BD Oracle	24	1	1
Conexión a la BD MySQL	24	1	1
Configurar Herramienta	40	1	1
Agregar Carpeta	16	1	1
Eliminar Carpeta	16	1	1

14 de junio de 2013

Copiar Carpeta	16	2	2
Autenticar por dominio	40	2	2
Administrar Usuarios	24	2	2
Cortar Carpeta	16	2	2
Permisos de Carpeta	16	2	2
Propiedades de Carpeta	16	2	2
Administrar Funciones	24	2	2
Generar Reporte dinámico	24	2	2

Tabla 9 Plan de Entrega

2.6 Conclusiones Parciales

En el presente capítulo se examinaron las funcionalidades a incluir en la herramienta, se identificaron 18 historias de usuarios, donde algunas de ellas debido a ser muy complejas fue necesario dividir las para lograr una mejor estimación en su desarrollo, todas estas van a ser desarrolladas en período de 9 semanas persiguiendo el proceso de desarrollo propuesto por XP.

Capítulo 3: Diseño, Implementación y Prueba

3.1 Introducción

En este capítulo se describen las fases de diseño, implementación y prueba, propias de la metodología de desarrollo XP. Se identifican y organizan las clases relevantes para las funcionalidades del sistema así como el patrón arquitectónico utilizado para la aplicación Web. Se procederá al diseño de las base de datos a utilizar y abordar las tareas de ingenierías definidas. Por último, y no menos importante se realizarán las pruebas al software las cuales se derivan de las Historias de Usuario y Tareas de la Ingeniería que se han implementado como parte del lanzamiento del software. Además se realizó un estudio de los aspectos favorables y desfavorables que tuvieron lugar durante el desarrollo de la solución y el impacto social de la misma haciendo énfasis en los aportes alcanzados sobre los procesos para la generación de reportes y las herramientas existentes para llevarlos a cabo.

3.2 Patrón de Arquitectura

Los sistemas de software crecen proporcionalmente, hoy se han perfeccionado tanto los algoritmos, que los códigos han dejado de convertirse en un problema. Actualmente el diseño de los sistemas constituye un nuevo reto, los patrones arquitecturales expresan la estructura fundamental para el sistema de software a desarrollar. Para el desarrollo del presente sistema el patrón de arquitectura que se seguirá es REST (Representational State Transfer), arquitectura que sigue JasperServer como guía de su implementación.

3.2.1 Arquitectura Rest

La Transferencia de Estado Representacional (REST) es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web. En la actualidad se usa en el sentido más amplio para describir cualquier interfaz Web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios Web SOAP. (27).

Un concepto importante en REST es la existencia de recursos (elementos de información), que pueden ser accedidos utilizando un identificador global. Para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian

14 de junio de 2013

representaciones de estos recursos (los ficheros que se descargan y se envían), no obstante, si la distinción entre recursos y sus representaciones es demasiado platónica para su uso práctico en la red. La petición puede ser transmitida por cualquier número de conectores (por ejemplo clientes, servidores, cachés, túneles, etc.) pero cada uno lo hace sin "ver más allá" de su propia petición (lo que se conoce stateless, otra restricción de REST, que es un principio común con muchas otras partes de la arquitectura de redes y de la información). Así, una aplicación puede interactuar con un recurso conociendo el identificador del recurso y la acción requerida, no necesitando conocer si existen cachés, cortafuegos, túneles o cualquier otra cosa entre ella y el servidor que guarda la información. La aplicación, sin embargo, debe comprender el formato de la información devuelta (la representación), que es por lo general un documento HTML o XML, aunque también puede ser una imagen o cualquier otro contenido. REST afirma que la Web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

- ✓ Un protocolo cliente/servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
- ✓ Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones se equiparan a las operaciones CRUD que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
- ✓ Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- ✓ El uso de hipermedias, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

Arquitectura de JasperServer:



Tabla 10 Arquitectura JasperServer

JasperServer cuenta con tres capas donde en cada una de ellas utiliza diversos framework, la capa de presentación utiliza JQuery, Json y SpringMVC, para controlar la seguridad de la herramienta utiliza Spring Security, la de lógica de negocio hace uso de Spring y la de acceso a dato de Hibernate. En la capa de presentación es donde el usuario puede interactuar con la aplicación o cualquier otro servicio que esta brinde. La lógica del negocio se encarga de hacer peticiones a la base de datos para conformar los reportes, de validar, parsear los ficheros de configuración y confeccionar los reportes para brindarlos en la capa de presentación. Finalmente en la capa de acceso a dato se define la conexión a una determinada base de datos y en dependencia de las consultas que se realicen esta brindara los datos necesarios para confeccionar los reportes solicitados. (28)

3.3 Diseño

Para lograr que la aplicación JasperServer cumpliera con los requisitos pedidos por el centro, se hizo necesario configurar dicha aplicación, esto se logra modificando y creando algunas hojas de estilos CSS para cargarlas en la configuración anterior que tenía la aplicación, quedando así un nuevo diseño de la herramienta, logrando un mejor entorno de trabajo para el cliente. Finalmente la aplicación queda con un nuevo estilo y un nuevo directorio de carpetas, como se muestra en la Ilustración 12 y 13 respectivamente.

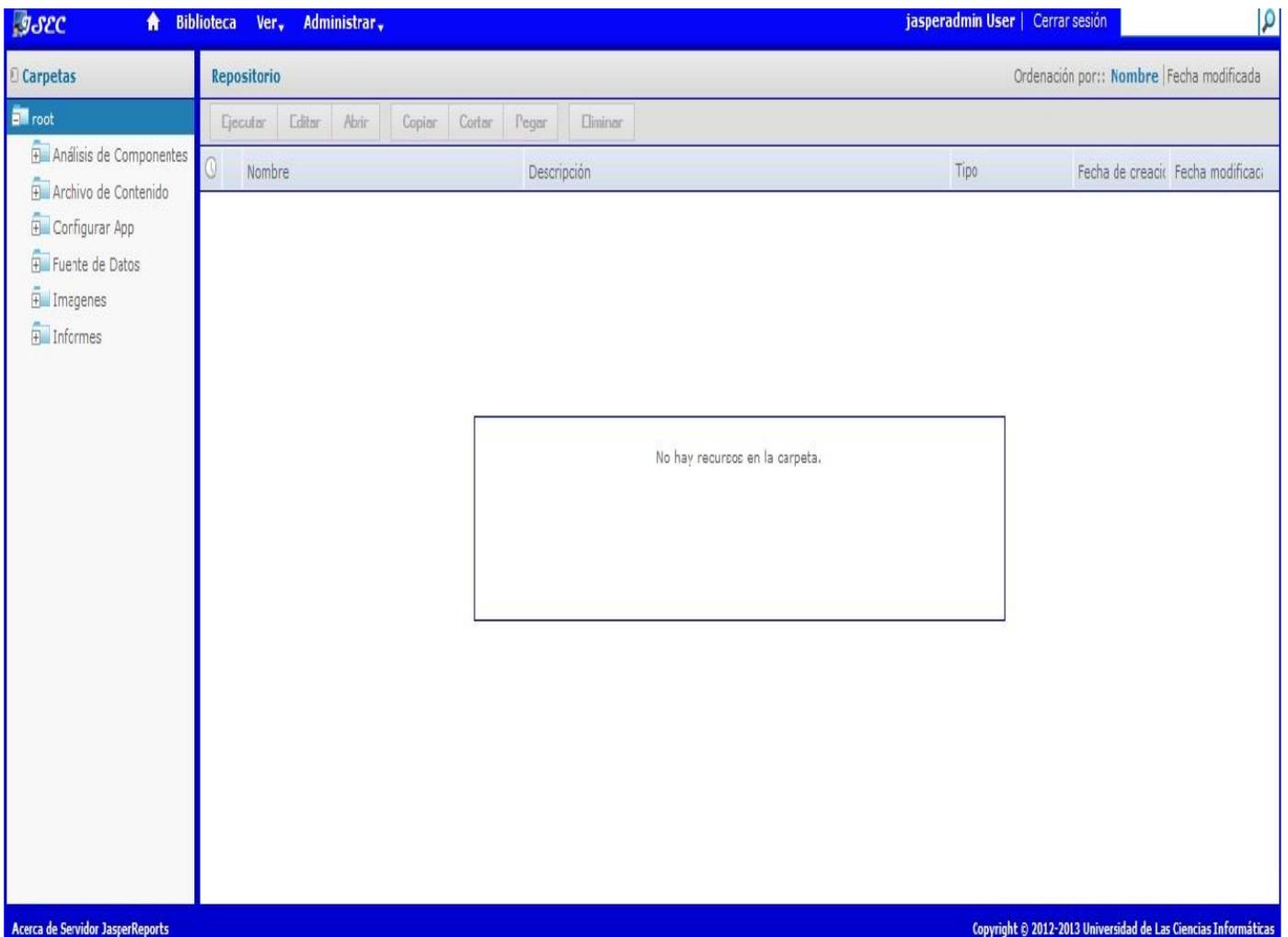


Ilustración 12 Configuración de la Aplicación

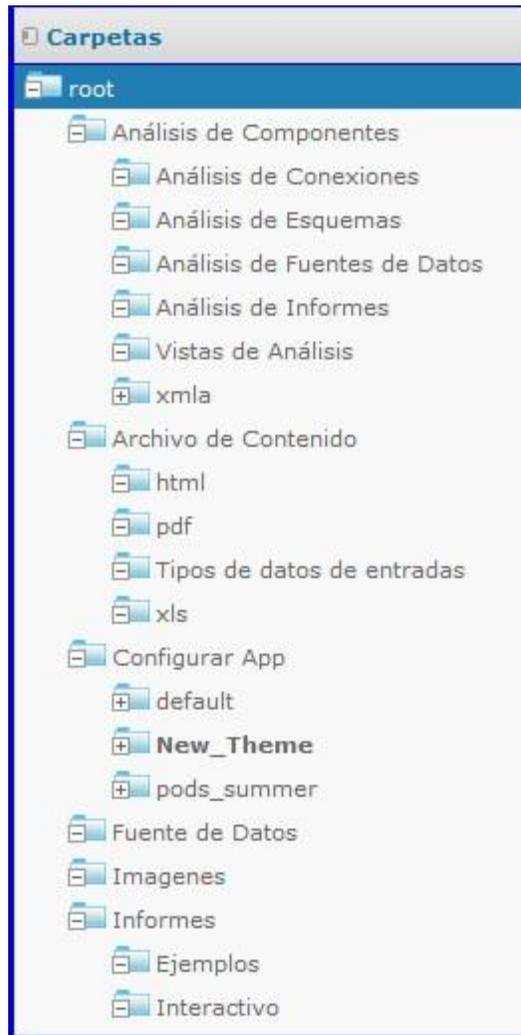


Ilustración 13 Nuevo directorio de carpetas de la Aplicación

Para comenzar a implementar cada una de las tareas se realizó un diseño simple previo, que incluía solamente las clases necesarias que utiliza JasperServer para realizar sus principales funciones. El mismo se hizo utilizando tarjetas CRC las cuales son usadas para representar las responsabilidades de las clases y su interacción. El nombre CRC viene dado por clase, responsabilidad y colaboración, elementos que sus creadores vieron como las dimensiones esenciales del modelo orientado a objetos. Las mismas se utilizaron para resumir el significado de las clases y estructurar su conjunto simulando escenarios. Para ver las demás tarjetas CRC. [Ver anexo 4.](#)

Clase GenerarReporteImpl	
Descripción: Clase que sirve como fachada para la generación de reporte.	
Nombre	Colaborador
Generar los reportes generales de la aplicación	JasperReportResource JasperDesign Formato JasperReport JasperCompilerManager
Generar un reporte a partir de uno compilado	JasperReport JasperReportResource DefaultSimpleDataSource JasperPrint JasperFillManager GenerarReporteExcel GenerarReportePDF GenerarReporteCSU GenerarReporteDOCx GenerarReporteRTF GenerarReporteFlash GenerarReporteODT GenerarReporteXLSX
Cargar Imágenes	URL JasperReportResourceImpl
Exportar reporte a un formato.	JasperPrint GenerarReportePdfImpl GenerarReporteExcelImpl GenerarReporteCsulImpl

	<p>GenerarReporteOdtImpl</p> <p>GenerarReporteXlsxImpl</p> <p>GenerarReporteOdsImpl</p> <p>GenerarReporteFlashImpl</p> <p>GenerarReporteRtfImpl</p> <p>GenerarReporteDocxImpl</p>
--	---

Tabla 11 Tarjeta CRC Generar Reporte

Como es de esperar cada uno de los reportes que se generen tienen su diseño específico, pues muestran datos diferentes y responden a documentos oficiales con características propias. Sin embargo, todos deben seguir una estructura que varía en dependencia del grupo de documentos al que pertenezca. Esta estructura generaliza los encabezados y pie de página de cada grupo de reportes, por lo que se hace posible la creación de archivos generales con estos formatos, lo cual posibilita que a la hora de diseñar un reporte, no sea necesario incluir las bandas pageHeader y pageFooter, limitando el diseño solamente a aquellos elementos específicos del reporte, lo cual ahorra tiempo de diseño. Además, un cambio en un elemento determinado de uno de estos formatos, provocaría un cambio solamente en el archivo general; de lo contrario, sería necesario cambiar el archivo de cada reporte perteneciente a dicho grupo.

3.4 Implementación

Para implementar las funcionalidades detalladas por el cliente anteriormente, se procedió a realizar las HU: autenticar por dominio y conexión a una determinada BD. Donde para lograr que la aplicación se conectara al dominio uci (dominio requerido inicialmente) se hizo necesario configurar varios archivos. Esto no quiere decir que no se pueda hacer para cualquier tipo de dominio, solamente basta con configurar los archivos ubicados en jasperserver\WEB-INF, identificados por los nombres de: log4j.properties y applicationContext-security.xml, quedando dichas configuraciones como se muestra en las Ilustraciones de la 14 a la 18.

Primeramente se configura el archivo log4j.properties, donde se debe agregar la siguiente línea de código:

```
# Ldap
log4j.logger.org.springframework.security.ldap=DEBUG, stdout, fileout
```

Ilustración 14 Configuración el archivo log4j.properties

Luego se configura el archivo applicationContext-security.xml, agregándole los parámetros que son requeridos para autenticarse a un dominio en específico:

```
<!-- ===== AUTHENTICATION ===== -->
<bean id="authenticationManager" class="org.springframework.security.providers.ProviderManager">
  <property name="providers">
    <list>
      <ref local="ldapAuthenticationProvider"/>
      <ref bean="{bean.daoAuthenticationProvider}"/>
      <ref bean="anonymousAuthenticationProvider"/>
      <!--ref local="jaasAuthenticationProvider"/-->
    </list>
  </property>
</bean>
```

Ilustración 15 Configuración ldap

For LDAP authentication-->

```
<bean id="ldapContextSource" class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
  <constructor-arg value="ldap://10.0.0.3:389/dc=uci,dc=cu"/>
  <!--
  You may not need the next properties-->
  <property name="userDn"><value>cn=ad search, ou=Systems, ou=UCI Domain Impersonals, dc=uci, dc=cu</value></property>
  <property name="password"><value>nF2SODWAHiW0eJboFFQEAvVzJ</value></property>
</bean>
```

Ilustración 16 Configuración ldap

```

<!--
For LDAP authentication
This bean is not used by default
-->
<bean id="userSearch" class="org.springframework.security.ldap.search.FilterBasedLdapUserSearch">
  <constructor-arg index="0">
    <value></value>
  </constructor-arg>
  <constructor-arg index="1">
    <value>(sAMAccountName={0})</value>
  </constructor-arg>
  <constructor-arg index="2">
    <ref local="ldapContextSource" />
  </constructor-arg>
  <property name="searchSubtree">
    <value>true</value>
  </property>
  <property name="derefLinkFlag">
    <value>true</value>
  </property>
</bean>

```

Ilustración 17 Configuración ldap

```

<!--
For LDAP authentication-->

<bean id="ldapAuthenticationProvider" class="org.springframework.security.providers.ldap.LdapAuthenticationProvider">
  <constructor-arg>
    <bean class="org.springframework.security.providers.ldap.authenticator.BindAuthenticator">
      <constructor-arg><ref local="ldapContextSource"/></constructor-arg>
      <property name="userSearch" ref="userSearch"/>
    </bean>
  </constructor-arg>
  <constructor-arg>
    <bean class="org.springframework.security.ldap.populator.DefaultLdapAuthoritiesPopulator">
      <constructor-arg index="0"><ref local="ldapContextSource"/></constructor-arg>
      <constructor-arg index="1"><value></value></constructor-arg>
      <property name="convertToUpperCase"><value>true</value></property>
      <property name="groupRoleAttribute"><value>cn</value></property>
      <property name="rolePrefix"><value></value></property>
      <property name="groupSearchFilter"><value>(member={0})</value></property>
      <property name="searchSubtree"><value>true</value></property>
      <!--<property name="defaultRole"><value>ROLE_USER</value></property>-->
    </bean>
  </constructor-arg>
</bean>

```

Ilustración 18 Configuración ldap

Para lograr que la herramienta se conectara a diversas bases de datos se hizo necesario incluir los drivers de cada una de las bases de datos Oracle, Postgres y MySQL, los mismos se agregaron en los archivos de la aplicación, en la siguiente dirección: apache-tomcat\lib. Para poder establecer dichas conexiones es necesario el conocimiento previo de algunos datos que son pedidos por la herramienta a la hora de establecer una determinada conexión. Ver Ilustración 19 y 20. Para más información dirigirse al Manual de Usuarios que se encuentra con toda la documentación final del trabajo realizado.

Establezca el tipo de origen de datos y sus propiedades

Primero, seleccione el tipo de origen de datos que desea agregar, después introduzca los valores de propiedad necesarios.

Tipo:

Nombre (necesario):

ID de recurso (necesario):

Descripción:

Controlador (necesario):

Consejo: org.postgresql.Driver

Ilustración 19 Conexión a una BD

14 de junio de 2013

URL (necesario):

Consejo: jdbc:postgresql://localhost:5432/mydb

Nombre de usuario (necesario):

Contraseña:

Zona horaria:

▼

Consejo: No cambie los ajustes de la zona horaria a menos que esté seguro de que los datos de marca de hora de la base de datos no son correctos.

Guardar ubicación:

Ilustración 20 Conexión a una determinada BD

Base de Datos	Driver y URL
Microsoft SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver jdbc:sqlserver://<host>:1433;databaseName=<database>
Microsoft SQL Server with jTDS driver	net.sourceforge.jtds.jdbc.Driver jdbc:jtds:sqlserver://<host>:1433/<database>
MySQL	com.mysql.jdbc.Driver jdbc:mysql://<host>:3306/<database>
Oracle (thin driver)	oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@<host>:1521:<database-instance>
PostgreSQL	org.postgresql.Driver jdbc:postgresql://<host>:5432/<database>
Vertica	com.vertica.jdbc.Driver jdbc:vertica://<host>:5433/<database>

Tabla 12 Datos de conexión a una BD

Integración de JasperServer con servicios web:

Como una de las tareas en las que se emerge el presente trabajo es la integración de JasperServer con cualquier aplicación, se hizo un profundo estudio para lograr unos de los principales objetivos previstos por el centro ISEC. Para ello se crea una aplicación Web para dicho estudio, llegando a la conclusión que es una operación bastante sencilla y que facilita a grandes rasgos el trabajo de las personas, ya que si se encuentran trabajando en cualquier otra aplicación no es necesario que tengan que salir para generar cualquier tipo de reporte. Solamente se debe asegurar la autenticación y seguridad del sitio, para luego darle la opción de acceder al JasperServer. Una vez asegurada la autenticación se debe incluir los códigos correspondientes a las ilustraciones 21 y 22 e incluir la clase redirect.php, donde la misma asegura en dependencia del tipo de usuario que sea los privilegios que este va a poseer una vez autenticado. Esto se puede ejecutar de dos formas, la primera mediante un vínculo, y la segunda a través de un botón, para las dos formas es necesario agregarle a la página que desee, el vínculo o el botón necesario para dar acceso al reporteador JasperServer. Si es un link, es necesario agregarle el evento "onclick" llamando en él, una función, la cual contendrá usuario y contraseña que posteriormente se envían por el método POST a la página redirect.php donde en dependencia del tipo de usuario que se autentique así serán las funcionalidades que este tendrá una vez accedido al JasperServer.

```

<form id="frmsubmit" method="post" action="controladora/redirect.php" style="display:none;"
<input type="hidden" name="usuario" id="usuario"/>
<input type="hidden" name="contrasena" id="contrasena"/>
</form>

<script type="text/javascript">
function sub(par1,par2){
    document.getElementById('usuario').value =par1 ;
    document.getElementById('contrasena').value =par2 ;
    document.getElementById('frmsubmit').submit();
}
</script>

```

Ilustración 21 Función y formulario dinámico para acceder a la clase redirect.php

```
<li>
  <a onclick="sub('<?php echo $_SESSION['usuario']; ?>', '<?php echo $_SESSION['contrasena']; ?>')">SGR - ISEC</a>
</li>
```

Ilustración 22 Programar evento onclick a un link

3.5 Tareas Ingenieriles

XP plantea que la implementación de un software se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Exploración. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente.

Las tareas de la ingeniería serán representadas mediante tablas divididas por secciones. A continuación las tareas correspondientes a las HU de prioridad más alta, consultar las demás en el [Anexo 2](#) para las tareas correspondientes a la primera iteración y el [Anexos 3](#) para las de la segunda iteración.

Tarea de Ingeniería:	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de Tarea: Generar Reporte	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6/6
Fecha Inicio: 21-01-2013	Fecha Fin: 25-01-2013
Programador Responsable: Sachie Roca Gamboa	
Descripción: El sistema debe ser capaz de cargar una lista de datos que debe ser procesada con un origen de datos de valor para JasperServer.	

Tabla 13 Tarea de Ingeniería Generar Reporte

Tarea de Ingeniería:	
Número de Tarea: 4	Número de Historia de Usuario: 4
Nombre de Tarea: Conexión a una BD	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5/6

14 de junio de 2013

Fecha Inicio: 05-02-2013	Fecha Fin: 09-02-2013
Programador Responsable: Sachie Roca Gamboa	
Descripción: Al seleccionar la opción para conectarse a una base de datos determinada, se interpretar la información introducida por el usuario y con solo agregar el driver de la Base de Datos deseada el sistema permite conectarse a la misma.	

Tabla 14 Tarea de Ingeniería Conexión a una BD

Tarea de Ingeniería:	
Número de Tarea: 8	Número de Historia de Usuario: 8
Nombre de Tarea: Configurar Herramienta	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6/6
Fecha Inicio: 21-02-2013	Fecha Fin: 28-02-2013
Programador Responsable: Sachie Roca Gamboa	
Descripción: El sistema le permite al administrador del sistema la opción de configurar la herramienta a su gusto o adaptarla de tal manera que sea más fácil para el usuario.	

Tabla 15 Tarea de Ingeniería Configurar Herramienta

Tarea de Ingeniería:	
Número de Tarea: 12	Número de Historia de Usuario: 12
Nombre de Tarea: Autenticar por Dominio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6/6
Fecha Inicio: 08-03-2013	Fecha Fin: 15-03-2013
Programador Responsable: Sachie Roca Gamboa	
Descripción: El sistema debes ser capaz de conectarse a cualquier dominio haciendo uso del LDAP	

Tabla 16 Tarea de Ingeniería Autenticar por Dominio

3.6 Pruebas

Dentro de los instrumentos capaces de medir el estado de calidad de un producto se encuentran las pruebas. El proceso de pruebas se dirige fundamentalmente a componentes del software o al sistema de software en general, con el objetivo de medir hasta cuando el software cumple las funcionalidades establecidas por el cliente. Propio de la metodología XP, se lleva a cabo la Fase de Prueba. Durante el desarrollo de software, XP establece probar constantemente tanto como sea posible, esto permite un aumento de la calidad del sistema desarrollado reduciendo el número de errores no detectados. (29).

Se efectuarán tres tipos de pruebas:

Pruebas Unitarias: las pruebas unitarias son las encargadas de verificar el código y son diseñadas por los programadores. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación de un sistema, para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente.

Las pruebas unitarias fueron desarrolladas constantemente cada vez que se terminaba de implementar alguna funcionalidad probándola directamente en el entorno real.

Pruebas de Aceptación: las pruebas de aceptación son pruebas de caja negra, también llamadas pruebas del cliente, las especifica el cliente y se enfocan en las características generales y las funcionalidades del sistema. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado.

Las pruebas de aceptación correspondiente a cada una de las funcionalidades que presenta JasperServer y serán representadas mediante tablas divididas por las siguientes secciones:

Clases Válidas: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas válidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.

Clases Inválidas: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las posibles entradas inválidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado y cómo responde el sistema.

Resultado Esperado: se hará una breve descripción del resultado que se espera ya sea para entradas válidas o entradas inválidas.

Resultado de la Prueba: se hará una breve descripción del resultado que se obtiene.

Observaciones: algún señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

Las pruebas de aceptación se llevarán a cabo redactando los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades. Luego se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo, para así reunir los miembros del proyecto seleccionados para realizarlas. Finalmente, se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba. Como se muestra en la siguiente tabla. Para ver las demás pruebas de aceptación dirigirse al [Anexo 5](#).

Autenticar por dominio

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El administrador accede a la interfaz principal e introduce los datos:</p> <ul style="list-style-type: none"> - ID de usuario: jasperadmin - Contraseña: ***** <p>Y presiona el botón: “inicio de sesión”</p>		<p>El sistema verifica que los datos sean correctos y da acceso a la aplicación con las funcionalidades pertinentes para este usuario.</p>	<p>El sistema da acceso al usuario satisfactoriamente.</p>	
	<p>El administrador accede a la interfaz principal e introduce solamente:</p> <ul style="list-style-type: none"> - ID de usuario: jasperadmin <p>Y presiona el botón: “inicio de sesión”</p>	<p>El sistema muestra un mensaje de error: “Credenciales introducidas no válidas. No se pudo iniciar sesión en JasperReports Server.”</p>		
	<p>El administrador</p>	<p>El sistema muestra</p>		

	<p>accede a la interfaz principal e introduce solamente:</p> <ul style="list-style-type: none"> - Contraseña: ***** <p>Y presiona el botón: "inicio de sesión"</p>	<p>un mensaje de error: "Credenciales introducidas no válidas. No se pudo iniciar sesión en JasperReports Server."</p>		
<p>El usuario accede a la interfaz principal e introduce los datos:</p> <ul style="list-style-type: none"> - ID de usuario: sroca - Contraseña: ***** <p>Y presiona el botón: "inicio de sesión"</p>		<p>El sistema verifica que los datos sean correctos y da acceso a la aplicación con las funcionalidades pertinentes para este usuario.</p>	<p>El sistema da acceso al usuario satisfactoriamente.</p>	
	<p>El usuario accede a la interfaz principal e introduce solamente:</p> <ul style="list-style-type: none"> - ID de usuario: sroca <p>Y presiona el botón: "inicio de sesión"</p>	<p>El sistema muestra un mensaje de error: "Credenciales introducidas no válidas. No se pudo iniciar sesión en JasperReports Server."</p>		
	<p>El usuario accede a la interfaz principal e introduce solamente:</p> <ul style="list-style-type: none"> - Contraseña: ***** <p>Y presiona el botón:</p>	<p>El sistema muestra un mensaje de error: "Credenciales introducidas no válidas. No se pudo iniciar sesión en</p>		

	“inicio de sesión”	JasperReports Server.”		
--	--------------------	---------------------------	--	--

Tabla 17 CP Autenticar

Pruebas Factibilidad: Están dirigidas esencialmente a medir los tiempos de repuestas que JasperServer demora en generar un reporte. Además de medir cuanto se demora consultar un parámetro en una base de datos extremadamente grande e imprimir un reporte de esta BD. Para ello se hace uso del programa Jmeter: es un proyecto de Apache Jakarta que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones Web. Puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios Web, JMS, HTTP y conexiones TCP genéricas. También puede ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo. Soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes. (30).

3.7 Productización

La fase de productización requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. (31).

La configuración e implementación de la herramienta se ha llevado a cabo dentro del entorno real donde será utilizado el mismo, por lo cual las pruebas de unidad realizadas al código se han realizado con datos reales y con el funcionamiento del resto del proyecto incluido. Es por esta razón que se pudieron detectar los fallos de integración y subsanar los mismos con anterioridad y una vez terminada cada iteración no fue necesario realizar pruebas adicionales.

3.8 Mantenimiento

El mantenimiento preventivo de software es el proceso por el cual se mejora y optimiza el software que se ha instalado, este mantenimiento se realiza para la prevención de posibles problemas que puedan llegar a surgir a medida que se utiliza el mismo. Esta fase es un estado normal en un proyecto XP pues

se deben producir nuevas funcionalidades y mantener el sistema corriendo simultáneamente. Se intentan grandes refactorizaciones no realizadas en la iteración culminada, se insertan nuevas tecnologías o se migra a versiones más actuales de las que ya se están usando, se experimentan nuevas ideas arquitectónicas y el cliente puede escribir nuevas historias en pos de buscar mejoras para su empresa.

Durante esta fase no se detectó ninguna falla en la implementación, esto se verificó mediante las pruebas funcionales realizadas por el cliente.

3.10 Análisis de Factibilidad

Una vez integrado la aplicación en el entorno de desarrollo, el mismo se utilizó para analizar el tiempo que se demora en generar un reporte determinado, filtrar parámetros de acuerdo a un patrón de búsqueda y la exportación del reporte a un formato específico. Para ello se estableció la conexión a la Base de Datos central de SIIPOL para obtener los datos, proceder a la realización de una plantilla de reporte en el iReport, cargar en el JasperServer el archivo .jrxml y realizar las pruebas pertinentes usando como archivo de comparación un reporte ya existente en la aplicación.

Se validó además que el tiempo de respuesta del sistema estuviera entre los límites señalados por las normas internacionales de calidad. Para ello se aplicó una métrica definida por la norma ISO 9126-3 (Métricas Internas de la Calidad del Software) enfocada al comportamiento del sistema denominada “Tiempo de respuesta” y que tiene como objetivo determinar cuál es el tiempo estimado para completar una tarea. (32).

Para ello, se evalúa la eficiencia de las llamadas al sistema operativo y a la aplicación y se estima el tiempo de respuesta basado en ello, midiéndose la ruta completa de una transacción. Calculando la variable:

$X = \text{tiempo (calculado o simulado)}$

Estas pruebas se hicieron de forma probabilística, imprimiendo en la consola el tiempo que arrojaba un timer colocado para medir cuanto se demoraban las funcionalidades en ejecutarse. Se realizaron iteraciones para cada funcionalidad seleccionada y se analizaron los valores quedando los resultados como se muestra en la figura 26.

14 de junio de 2013

Después de acceder a la Base de Datos central del Siipol, se hizo una estimación de cuanto era que se demoraba si se consultaba una persona específica, arrojando esto un tiempo de 1.753 segundos la primera vez, luego como Oracle ya cachea dicha consulta solamente demora 0.905 segundos. A la hora de imprimir un listado de persona por primera vez se demora 4.335 segundos y después de eso solo demora 2.135 segundos. Lo cual constituye un valor aceptable para un sistema que maneja una gran cantidad de datos y mantiene una conexión constante con el servidor de Base de Datos.

En JasperServer cuando se genera un reporte este arroja un tiempo de 0.896 segundos la primera vez, luego arroja un tiempo de 0.818 segundos, Cuando exporta el reporte a un determinado formato se ejecuta en 0.450 segundos y cuando se filtra en reporte dado un parámetro específico este se ejecuta la primera vez en 1.36 segundos y la segunda vez en 0.593 segundos.

El siguiente gráfico muestra los datos obtenidos en milisegundos a partir del análisis realizado en la base de datos de Siipol y en el reporteador JasperServer. Llegando a la conclusión que es más factible y eficiente la utilización de este último.

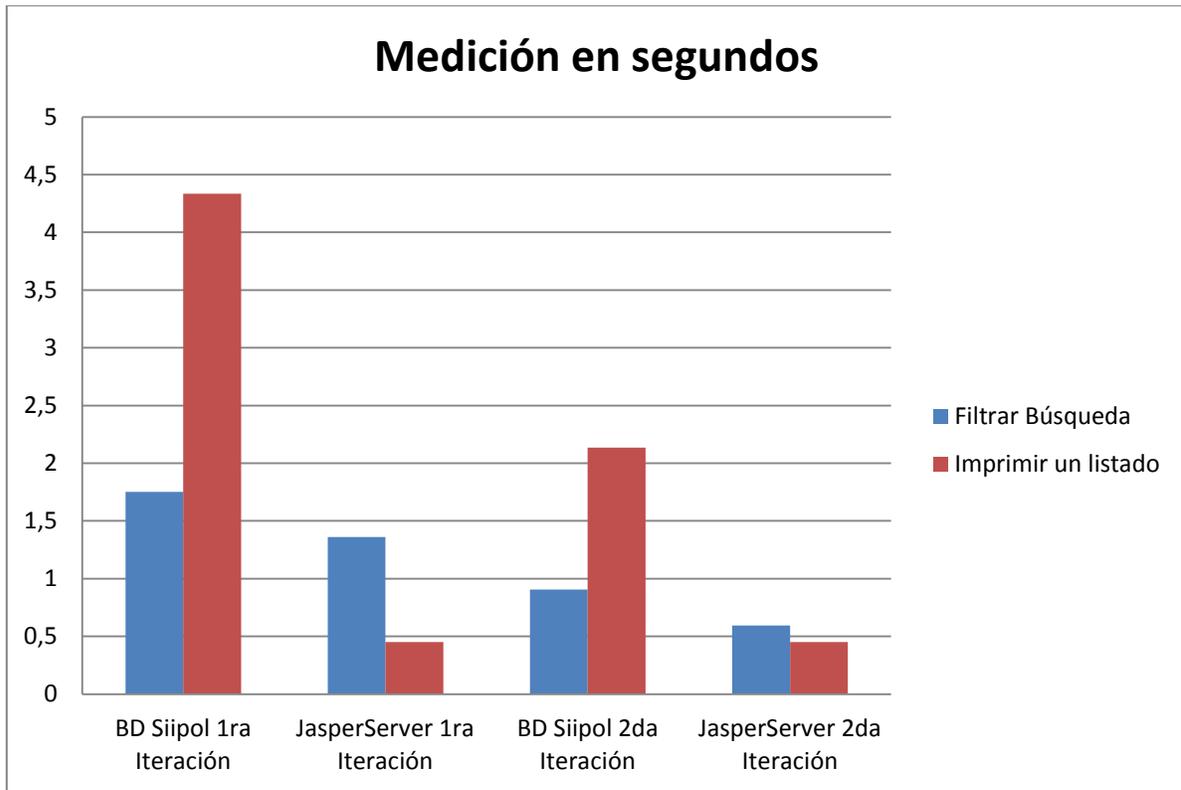


Gráfico 1 Análisis de Factibilidad

3.10.1 Beneficios Tangibles e Intangibles

Tangibles:

- ✓ Disminución del almacenamiento de información en papel, y al mismo tiempo la posibilidad de obtener y manipular un gran volumen de datos, llegando a ser miles de registros los procesados diariamente.
- ✓ Depuración de la información que se ha ido acumulando a lo largo de los años.
- ✓ Mayor control de las actividades desarrolladas en las áreas de trabajo del centro ISEC, permitiendo la evaluación, seguimiento y control de sus procesos.
- ✓ Generación eficiente de reportes, tanto predefinidos como informes del estado de cada parte del proceso que permiten mostrar la información detallada para facilitar análisis y planificaciones posteriores.

14 de junio de 2013

- ✓ Establecimiento de estándares (metodológicos, de código, diseño y documentación) para futuros proyectos.

Intangibles:

- ✓ Incremento de la cultura informática de la institución en el uso de la tecnología como instrumento de trabajo diario.
- ✓ Eficiencia en la Gestión de Información en los Procesos de Capacitación y Reentrenamiento.
- ✓ Aumento del alcance y la credibilidad del centro ISEC.
- ✓ Mayor satisfacción del cliente.
- ✓ Aumento de la efectividad organizacional.
- ✓ Información más relevante para el análisis y la toma de decisiones.

3.11 Visión Global del Trabajo Realizado

Debido a la gran necesidad que presentaba el centro ISEC a la hora de generar un reporte, se procedió a realizar un análisis de los principales aspectos para lograr dicho objetivo. Se estudiaron las herramientas existentes a nivel internacional, escogiendo finalmente entre todas JasperServer, ya que esta aplicación Web trae consigo los programas y librerías necesarios para la generación de documentos a diversos formatos. JasperServer genera gráficos y reportes estáticos, solamente establece conexión con la base de datos Postgres, exporta documentos a formato: PDF, HTML, XML, Excel y RTF. Luego de que el cliente nos diera las principales características que el programa debe presentar, se procedió a estandarizar la herramienta, escogiendo como metodología rectora para el proceso del desarrollo del software, XP. Agregándole nuevas funcionalidades como: la exportación de documentos a formatos ODS, ODT, CSV, Flash y XLSX. También se hace uso de LDAP, permitiendo esto, la conexión a un determinado dominio (en este caso para el dominio de la UCI). Si otro centro o institución desea integrar dicha aplicación o utilizarla, solamente debe establecer su configuración específica en el fichero applicationContext-security.xml o de lo contrario utilizar los usuarios por defecto que esta posee. Se adaptaron las configuraciones pertinentes para lograr un mejor entorno de trabajo, así como un nuevo directorio raíz para mejorar la navegabilidad en la aplicación. Se le agregaron los driver de conexión con las base de datos MySQL y Oracle de manera que si existe alguna institución con este tipo de BD, pueda utilizar la misma o de lo contrario para establecer una conexión a otra BD que no

sean las antes mencionadas, solamente debe agregar el driver de conexión de la misma y a establecer los valores requeridos a la hora de conectarse. Se agregaron las funcionalidades pertinentes para que a la hora de cargar un reporte, este permita filtrar parámetros por un determinado valor. Obteniendo con la nueva configuración una aplicación que cumple con todos los objetivos previstos por el cliente, agregando a todo este proceso la documentación detallada de la misma para futuras versiones, además de una ayuda en línea de la aplicación y un manual de usuario sobre todos los pasos que se siguieron para lograr mejorar dicha herramienta.

3.12 Conclusiones Parciales

En este capítulo se realizó un profundo análisis sobre el patrón arquitectónico sobre el que se desarrolla la aplicación. Se identificaron y describieron las tareas de la ingeniería listas para su implementación, correspondiente a las HU descritas en el capítulo anterior. Se identificaron las clases que posee JasperServer. Por último, y no menos importante, se llevaron a cabo las pruebas de aceptación logrando la satisfacción del cliente con el software desarrollado. Así como los tiempos de respuestas a la hora de generar un reporte cargado de una Base de Datos extremadamente grande.

Conclusiones

Con el desarrollo de este trabajo se profundizó en el conocimiento de los sistemas generadores de reportes. Luego de escoger JasperServer como aplicación para utilizar en el centro ISEC, se realizó un estudio de las metodologías que podían guiar la investigación escogiendo finalmente XP. Llegando a la conclusión de que se cumplió el objetivo general propuesto, obteniéndose una herramienta que genera reportes dinámicamente. Para ello:

- Se realizó la implementación de todas las funcionalidades faltantes a la herramienta JasperServer permitiendo esto la fácil generación de reportes en el centro ISEC.
- Se realizó la integración de dicha aplicación a nivel visual y tecnológico para cumplir el objetivo propuesto por el centro.
- Se realizó la documentación del desarrollo final al presente trabajo en pos de facilitar información para futuras versiones, así como la realización de un manual de usuario de todo el proceso realizado y una ayuda en línea para cualquier duda inmediata que le pueda surgir al cliente

Con la estandarización de la herramienta JasperServer se facilitó la generación de reportes y estadísticas para el Centro ISEC. Por lo que se puede concluir que se dio cumplimiento al objetivo general trazado.

Recomendaciones

Con la culminación del presente trabajo, se dio cumplimiento a los objetivos propuestos, dotando al centro ISEC de una aplicación para la generación de reportes que cumple con todas las funcionalidades requeridas por el mismo. Sin embargo, se realizan una serie de recomendaciones para enfocar el trabajo futuro en el mejoramiento de la propuesta realizada.

1. Generalizar el conocimiento vinculado a la generación de reportes teniendo en cuenta que es un tema poco estudiado en la Universidad.
2. Integrar dicha aplicación en otros centros de la universidad o en instituciones que requieran de una aplicación que genere reportes.

Referencias Bibliográficas

1. **Puldón Joan, Ing. Jaime, y otros, y otros.** “Librería iText para la generación de PDF dinámicos.”. [En línea] ccia.cujae.edu.cu/index.php/siia/siia2008/paper/download/.../290.
2. Administración y Finanzas. [En línea] www.monografias.com.
3. **Melina, Jesica.** **tallersoft2011-JASPERSERVER**. [En línea] http://mistock.lcompras.biz/index.php?option=com_content&view=article&id=1276:javasperserver&catid=59:tallersoftware&Itemid=116.
4. JasperServer y el API de servicios web basado en REST. [En línea] 2012. <http://blog.gkudos.com/2012/05/31/jasperserver-y-el-api-de-servicios-web-basado-en-rest/>.
5. **SAP Crystal Reports.** SAP Crystal Reports. [En línea] <http://www.sap.com/spain/solutions/sap-crystal-solutions/query-reporting-analysis/sapcrystalreports/index.epx>.
6. **ActiveReports** 7. **ActiveReports** 7. [En línea] <http://www.componentsource.com/products/activeresports-6/index-es.html>.
7. Oracle Business Intelligence Publisher. [En línea] Oracle Business Intelligence Publisher.
8. Business Intelligence and Reporting Tools. [En línea] www.stratebi.com/birt.
9. HISPASEC. [En línea] <http://unaaldia.hispasec.com/2012/12/solucionadas-tres-vulnerabilidades-en.html>.
10. JasperReports Server. [En línea] <http://community.jaspersoft.com/project/jasperreports-server>.
11. Apache Commons. [En línea] commons.apache.org/digester.
12. **Heffel, David R.** JasperReports for Java Developers. *Birmingham : Packt Publishing Ltd.* s.l. : ISBN 1-904811-90-6, 2006.
13. **Reporting., Project Statistic for JasperReports - Java.** SourceForge.net. [En línea] 2009. http://sourceforge.net/project/stats/detail.php?group_id=36382&ugn=jasperreports&type=prdownload&mode=week&package_id=28579&release_id=626893&file_id=1628202..
14. APÉNDICE I. EL JDK (Java Development Kit). [En línea] http://zarza.usal.es/~fgarcia/doc/tuto2/A_I.htm.
15. **Amaro Calderón, Sarah Dámaris, Valverde Rebaza, Jorge Carlos.** Metodologías Ágiles. Trujillo-Perú : Universidad Nacional de Trujillo. s.l. : Facultad de Ciencias Físicas y Matemáticas, Escuela de Informática, 2007.
16. **Respuestas.** [En línea] <http://espanol.answers.yahoo.com/question/index?qid=20070409191212AA6esHE>.
17. **Metodologías del desarrollo de software.** [En línea] http://www.ecured.cu/index.php/Metodolog%C3%ADas_de_desarrollo_de_software.
18. **Garzás, Javier.** La Metodología ágil FDD. [En línea] <http://www.javiergarzas.com/2012/09/metodologia-gil-fdd-1.html>.
19. **Publicaciones.** *RUP vs XP.*
20. Qué es Java. [En línea] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
21. **Velasco, Roberto Hernando.** El SGBDR Oracle. [En línea] <http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html>.
22. Postgres. [En línea] www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.

23. **Enríquez Toledo Alma, Maldonado Ayala Jesús, Nakamura Ortega Yunko, Nogueron Toledo Goretty.** MySQL.
24. **Beck, Kent.** Extreme Programming Explained. [aut. libro] s.l. : Addison Wesley. s.l. : ISBN: 0201616416., 1999.
25. **Fowler, Kent Beck y Martin.** Planning Extreme Programming. s.l. : ISBN: 0-201-71091-9., 2000.
26. **Fowler., Kent Beck y Martin.** Planning Extreme Programming. s.l. : s.l. : Addison Wesley.
27. Ingeniería del Software en el siglo XXI. [En línea] <http://wp.me/pQAdW-4K>.
28. **JasperSoft community.** JasperReports Server Architecture. [En línea] <http://community.jaspersoft.com/wiki/jasperreports-server-architecture>.
29. *Pruebas de software.* 2000.
30. Jmeter. [En línea] <http://es.wikipedia.org/wiki/JMeter>.
31. **González, Carlos Sánchez.** *ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras.* s.l. : s.l. : UNIVERSIDADE DA CORUÑA, 2004.
32. **Mendoza, Gonzalo Mena.** ISO 9126-3: Métricas Internas de la Calidad del Producto de Software. s.l. : Querétaro : Universidad Autónoma, 2006.
33. Crystal Reports. [En línea] [Crystal Reports.msdn.microsoft.com](http://CrystalReports.msdn.microsoft.com).
34. Introducción a JasperReports. [En línea] <http://www.campus.fi.unju.edu.ar>.
35. [En línea] www.componentsource.com/products/activereports-6/index-es.html.
36. [En línea] <http://www.gravitar.biz/index.php/bi/integracion-hyperion-oracle-bi/>.
37. **Vélez, Andrés Arenas.** JasperReports Server. [En línea] community.jaspersoft.com/.../jasperreports-serve... .
38. **Hernández, Dayana Daniel.** *Componente Arquitectónico para la generación de reportes.*
39. Respuestas. [En línea] <http://espanol.answers.yahoo.com/question/index?qid=20070409191212AA6esHE>.
40. **Intelligence, BI-Business.** [En línea] <http://bi-businessintelligence.blogspot.com/>.

Glosario de Términos

Apache Software Foundation: Organización no lucrativa creada para dar soporte a los proyectos de software bajo la denominación *Apache*, incluyendo el popular servidor HTTP Apache.

API: Una interfaz de programación de aplicaciones es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

JDBC: API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

SOA: Arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

XML: Metalenguaje extensible de etiquetas. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

.JRXML: Archivo que te permite configurar el informe como se desee y es necesario para generar un reporte en JasperReports.

BI: Se denomina **inteligencia empresarial**, **inteligencia de negocios** o **BI** (del inglés *business intelligence*) al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

XAMPP: es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

SOAP: siglas de Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

HTML: (Hyper Text Markup Language) Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de etiquetas diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas web.

JAVA: Lenguaje de programación multiplataforma desarrollado por Sun Microsystem.

XML: (eXtensible Markup Language) Lenguaje de Marcas Extensible. Es un metalenguaje extensible de etiquetas.

XP: (Extreme Programming) Programación Extrema. Es una metodología ágil de desarrollo de software.