

Universidad de las Ciencias Informáticas



Título: Propuesta de una herramienta libre para la Gestión de Requisitos en proyectos productivos de la facultad 10.

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

Autores: Leidy Ramos González
Renier Socarrás Pérez

Tutores: Ing. Greidys Jorda Lueges
Ing. Yanko Hernández Valdés

**Ciudad de la Habana
2008**

DECLARACIÓN DE AUTORÍA

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2008.

Leidy Ramos González

Firma del Autor

Renier Socarrás Pérez

Firma del Autor

Greidys Jorda Lueges

Firma del Tutor

Yanko Hernández Valdés

Firma del Tutor



“Si el hombre crea solamente para sí mismo, podrá ser un famoso científico, un gran sabio, un excelente poeta, pero jamás será realmente un ser grande y perfecto.”

Carl Marx

Agradecimientos

Leidy

Para la realización del presente trabajo fue necesaria la colaboración de muchas personas, a quien dedico mis más modestos agradecimientos:

A mis padres, que nunca me abandonaron en este camino de hacerme profesional.

A mis tutores, Greidys Jorda Lueges y Yanko Hernández Valdés por confiar en mí.

A Yenisleidy Cariaga y Arismayda Dorado por guiarme metodológicamente en el desarrollo del trabajo.

A Andro Gilberto Mazanet Huart por el apoyo incondicional y los consejos útiles brindados en todo momento.

A Yaima y Dailín por toda la ayuda necesaria.

A los cuadros de la UJC por estar siempre dispuestos a ofrecerme información y ayuda en el trabajo.

A mis compañeros y amigos de la Universidad.

A todos Muchas Gracias.

Renier

A mi mamá, a mi hermana y a mi papá que siempre me dieron ese apoyo necesario para seguir adelante.

A mi novia y todos esos compañeros que me han ayudado, hasta en los tiempos más difíciles.

A todos aquellos que en algún momento dedicaron su tiempo para ofrecer su ayuda en el desarrollo de este trabajo.

Al Proyecto Futuro que confió en nosotros cuando solo éramos unos niños, pero que supimos dar el paso al frente.

A todos ellos, les agradezco que confiaran en mí.

Dedicatoria

Leidy

A Fidel Castro Ruz por ser el creador de esta casa de altos estudios y la UCI por permitirme ser parte de ella.

A mis padres por guiarme por el camino correcto y quererme tanto.

A mi hermana por apoyarme en todo.

A mi amor por su comprensión, ayuda y dedicación.

A mi familia por estar siempre a mi lado y ayudarme en todo lo que la necesité.

A todos, dedico este trabajo.

Renier

A mis padres María y Renato y a mi hermana Yayi por estar presente más de 20 años y guiarme en mis estudios, a ellos se la dedico para retribuirle esta alegría, de las tantas que ellos me dieron a mí.

A todos aquellos que les hace feliz que yo concluyera mis estudios, a mis profesores y compañeros.

A toda mi familia que siempre daba su apoyo y se preocupaba por lo que yo hacía para continuar, y a esas amistades que siempre las tengo presente, por las cosas que he hecho por ellos y las que ellos hicieron por mí.

A todos ellos se la dedico.

Resumen

La Gestión de Requisitos es un componente vital en el desarrollo de un proyecto software ya que provee la dirección y alcance del proyecto. El uso de herramientas para auxiliar la Gestión de Requisitos se ha convertido en un aspecto importante de la ingeniería, considerando el tamaño y la complejidad de cada proyecto, estas vienen siendo un eslabón esencial para lograr un producto con la calidad requerida y en el tiempo establecido.

En la actualidad, diversas son las compañías y empresas de desarrollo de software que tienen una Gestión de Requisitos ineficiente debido a que utilizan algún enfoque informal o ignoran los posibles riesgos que trae como consecuencia en el avance del proyecto. El objetivo de este trabajo es proponer el uso de una herramienta libre personalizada para la Gestión de Requisitos que cumpla con las necesidades de los proyectos productivos de la facultad 10.

El presente trabajo ofrece una alternativa para gestionar requisitos, para ello se realizó un estudio de las características generales y específicas de las principales herramientas existentes en el mundo. Se fundamentó el análisis y diseño de la funcionalidad que se le necesita agregar a dicha herramienta teniendo en cuenta las necesidades de la Universidad de las Ciencias Informáticas.

Para seleccionar la herramienta, la premisa fundamental la constituyó el software libre, contribuyendo al proceso de migración llevado a cabo en la Universidad de las Ciencias Informáticas.

Palabras claves

Ingeniería de Requisitos, Gestión de Requisitos, Herramienta, Proceso, Software Libre.

Índice

Agradecimientos	I
Dedicatoria.....	II
Resumen.....	III
Introducción	1
Capítulo I: Fundamentación Teórica.....	5
1.1 ¿Qué son los Requisitos?	5
1.2 Principales factores de éxito y fracaso de proyectos de software.....	6
1.3 El origen de la Ingeniería de Requisitos.....	7
1.4 Impacto de la Ingeniería de Requisitos.	7
1.5 ¿Qué es la Ingeniería de Requisitos?	8
1.6 Gestión de Requisitos	10
1.7 Tareas principales de la Gestión de Requisitos	10
1.8 La Gestión de Requisitos en la Práctica	11
1.9 Descripción de las Actividades de la Gestión de Requisitos.....	12
1.10 Panorámica	13
1.11 Ventajas y desventajas	14
1.12 Metodología y Herramienta CASE	16
Capítulo II: Análisis de la herramientas	18
2.1 Funcionalidades de las Herramientas de Gestión de Requisitos.....	18
2.2 Proceso de Gestión de Requisitos	20
2.3 La entrevista y sus resultados.....	21
2.4 Gestión de Requisitos en los proyectos productivos de la facultad 10.	22
2.5 Herramientas de Gestión de Requisitos en el mercado.....	23
2.6 Herramienta OSRMT Open Source Requirements Management Tool.	32

2.7 Aplicaciones adicionales	35
2.8 Seguridad de la información.....	35
Capítulo III: Análisis y Diseño del Módulo Reportes.	37
3.1 Modelo del Dominio.....	37
3.2 Diagrama de paquetes.	38
3.3 Descripción de los requisitos funcionales y no funcionales del paquete.....	41
3.4 Diagrama de Casos Usos del sistema.	44
3.5 Análisis y diseño del sistema	49
3.6 Diagramas de interacción.....	58
3.7 Descripción de las clases.....	65
3.8 Modelos de datos del sistema.....	71
Capítulo 4: Análisis de factibilidad del sistema.	73
4.1 Planificación del proyecto basada en el análisis de puntos de casos de uso.....	73
4.2 Análisis de costos y beneficios.....	79
Conclusiones	80
Recomendaciones	81
Referencias Bibliográficas	82
Bibliografía Consultada.....	83
Glosario de términos.....	91
Glosario de abreviaturas.....	95
Anexos.....	98

Introducción

En los últimos tiempos, el desarrollo del software ha tenido un auge considerable. Se conoce como la célula fundamental de muchos sistemas productivos. Las compañías y empresas mundiales, dependen cada día más de un buen software que le facilite su producto con mejores resultados, menos tiempo y menor costo.

El software ha revolucionado la manera de operar los negocios, industrias y hasta los individuos, al determinar los estándares de eficiencia personal y efectividad organizacional en todo el mundo. (FONT y VÁZQUEZ, 2007)

Uno de los principales problemas por lo que la mayoría de los proyectos no cumplen con las fechas establecidas y con sus objetivos es porque no se realiza de manera correcta la ingeniería de requisitos, los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada Gestión de Requisitos.

La Gestión de Requisitos es fundamental para el éxito de los proyectos que se realizan habitualmente en cualquier sector, saber lo que se va a hacer, cómo se va a hacer y dejar constancia de ello, trae como resultado un proyecto con mejor calidad y entrega en el tiempo establecido.

La sociedad cubana incrementa el desarrollo en la rama de la informática, aunque insipiente todavía, se encuentra inmersa en un proceso de migración de software privativo a software libre.

El software libre ha tomado buen curso y ha ganado en usuarios aunque queda mucho por recorrer, y si no esta más diseminado, es por la falta de conocimiento sobre el mismo en la sociedad, muchas personas no se suman a la comunidad de Software Libre, y principalmente por no saber el verdadero significado de la palabra “libre” en materia de software.

La libertad de la que se habla, no es la referente al no reconocimiento de cada autor de cualquier software; es la similar a la que se llama sociedad libre o libertad de expresión. El código debe ser algo transparente como las leyes que rigen un país.

En los últimos años, la Revolución Cubana se ha dado a la tarea de informatizar la sociedad, llevando las computadoras y los softwares hasta los lugares más intrincados de nuestro país. De ahí surge la brillante idea en el año 2002 de crear la primera universidad de la batalla de ideas llamada “Universidad de las Ciencias Informáticas” (UCI).

La UCI representa la infraestructura tecnológica más avanzada del país, la cual también se encuentra desarrollando un proceso de migración a software libre. El desarrollo tecnológico de Cuba necesita de esta migración y todos los procesos relacionados con el mismo, deben estar encaminados a esta

tarea; comenzando por el espacio universitario y luego haciéndolo extensivo al resto de los sectores económicos, políticos y sociales, para que llegue a cada rincón de la sociedad.

La universidad juega un papel fundamental en el desarrollo del software en el ámbito nacional, por lo que no se encuentra excluida de las deficiencias que existen cuando se desarrolla un producto; una de las dificultades que más puede afectar el resultado de un software, es los problemas en la recopilación, análisis y Gestión de Requisitos.

Para hacer la ingeniería de requisitos la Universidad utiliza herramientas privativas, aunque también se apoya en algunos modelos manuales. Actualmente la UCI carece de este tipo de herramienta libre, que ayude a elaborar de forma adecuada la Gestión de Requisitos; debido a esta carencia se hace necesario proponer una herramienta libre que apoye la Gestión de Requisitos en proyectos productivos de la facultad 10 sin la necesidad de utilizar modelos manuales que hagan el trabajo más engorroso y sin la calidad requerida, ya sea por el tamaño o la complejidad del proyecto.

Para la UCI es vital un sistema informático integral, que cumpla con los requerimientos necesarios para los proyectos productivos de la facultad 10, capaz de mejorar el procesamiento técnico de los documentos, reduciendo así, el costo y el tiempo de ejecución.

Por lo que el problema científico a resolver es:

¿Cómo facilitar la Gestión de Requisitos en los proyectos productivos de la facultad 10 mediante la utilización de una herramienta sobre plataforma libre?

Dada la problemática anterior el **objeto de estudio** lo constituye el proceso de Gestión de Requisitos para el desarrollo del software y el **campo de acción** del trabajo se centra en la Gestión de Requisitos sobre plataforma libre en proyectos productivos de la facultad 10.

Teniendo en cuenta el problema anterior, el **objetivo general** de este trabajo es proponer el uso de una herramienta libre personalizada para la Gestión de Requisitos que cumpla con las necesidades de la UCI en los proyectos productivos de la facultad 10.

Para llevar acabo el desarrollo de la investigación se trazaron los siguientes **objetivos específicos**:

1. Analizar los aspectos teóricos que sustentan el proceso de Gestión de Requisitos durante el desarrollo de un software.

2. Realizar un análisis de las principales herramientas para la Gestión de Requisitos existentes en el mundo.
3. Proponer una herramienta libre para la Gestión de Requisitos que satisfaga las necesidades de los proyectos productivos de la facultad 10.

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

1. Realizar un levantamiento bibliográfico sobre los principales aspectos teóricos, actividades y proceso que se manejan en la Gestión de Requisitos durante el desarrollo de un software.
2. Hacer un levantamiento de las diferentes herramientas para la Gestión de Requisitos que existen en la actualidad.
3. Fundamentar y personalizar la herramienta seleccionada para su uso en los proyectos productivos de la facultad 10.

Estructura del contenido con una breve explicación de sus partes.

El trabajo consta de una introducción, cuatro capítulos bien estructurados y legibles, conclusiones, recomendaciones y referencias bibliográficas para más información acerca del tema tratado.

Capítulo I. Fundamentación teórica.

En este capítulo se aborda todo lo relacionado con los conceptos más importantes de la ingeniería y Gestión de Requisitos de manera general, se tratan cuestiones como el origen de la ingeniería de requisitos, pasos, ventajas y desventajas de la Gestión de Requisitos así como la descripción de sus actividades.

Capítulo II. Análisis de las herramientas para la Gestión de Requisitos.

En este capítulo se muestran las funcionalidades básicas que debe tener una herramienta para la Gestión de Requisitos, aborda características de las principales herramientas y a partir de las mismas se establece una comparación entre cada una de ellas, ofreciendo elementos que justifican la preferencia de una por encima de las demás, teniendo en cuenta como eslabón principal que sea para

plataforma libre, se explica además todo lo relacionado con la herramienta propuesta Open Source Requirements Management Tool, sus características generales y específicas.

Capítulo III. Análisis y diseño del módulo Reportes.

En el presente capítulo se desarrolla el análisis y diseño de la funcionalidad que se le agregará a la herramienta OSRMT, dicho módulo es necesario para una correcta Gestión de Requisitos y para que satisfaga todas las necesidades de los proyectos productivos de la facultad 10.

Capítulo IV. Análisis de factibilidad del sistema.

En este capítulo se analiza, si es o no factible personalizar la herramienta OSRMT, agregándole el módulo de Reportes para un mejor funcionamiento de manera general y se hace por el método de estimación Análisis de Puntos de Casos de Usos.

Al finalizar, las conclusiones expresan los resultados obtenidos en la investigación y las recomendaciones manifiestan las posibles aplicaciones de la herramienta OSRMT, tanto en la docencia como en la producción, así como en otras ramas. Las referencias bibliográficas y la bibliografía consultada demuestran la profundidad y calidad de la investigación.

Capítulo I: Fundamentación Teórica

En este capítulo se tratarán los principales conceptos relacionados con la Ingeniería de Requisitos, su origen y surgimiento así como los pasos principales en los que esta se describe.

Se aborda todo lo relacionado con la Gestión de Requisitos, sus tareas principales, descripción de las actividades que se desarrollan en ella, todos los pasos y cosas que debemos tener en cuenta a la hora de aplicar en la práctica una herramienta para la Gestión de Requisitos.

Por último se ofrece una panorámica general relacionada con la aplicación de herramientas para la Gestión de Requisitos y las principales ventajas y desventajas que puede presentar el uso de estas herramientas.

1.1 ¿Qué son los Requisitos?

Los Requisitos constituyen el pilar fundamental de todo software, se inician cuando empieza cada proyecto. Ellos están presentes durante toda la ejecución del trabajo, hasta terminar el producto que se le entrega al cliente.

Los requisitos son:

- (1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- (3) Una representación documentada de una condición o capacidad como en (1) o (2).

Por lo que los requisitos son funcionalidades o características que debe cumplir tanto sistemas, hardware, software y usuario para dar cumplimiento a un problema determinado. (THAYER y DORFMAN, 1997)

Requisito funcional

Requisito que especifica una funcionalidad o capacidad que debe realizar o cumplir un sistema o un componente.

Requisito no funcional

Define las propiedades y cualidades que son necesarias tener en cuenta a la hora de diseñar e implementar un software y que el producto debe tener una vez culminado. Existen muchos tipos de requisitos no funcionales entre los que podemos encontrar (HURTADO, 2006):

- Requisito no funcional de software.
- Requisito no funcional de hardware.
- Requisito no funcional de usabilidad.
- Requisito no funcional de soporte.
- Requisito no funcional de costo.
- Requisito no funcional de eficiencia.
- Requisito no funcional de apariencia.
- Requisito no funcional de rendimiento
- Requisito no funcional de seguridad.
 - Integridad
 - Confiabilidad
 - Disponibilidad
- Requisito no funcional de desempeño
 - Tiempo
 - Espacio

1.2 Principales factores de éxito y fracaso de proyectos de software.

Investigaciones realizadas en Estados Unidos, por la Oficina de Cuentas de Gobierno Norteamericano (*Government Account Office, GAO*) (DIV, 1979), por el Grupo Standich en 1995 (STANDICH, 1995) y en 1996 en Europa, llevada a cabo por proyecto ESPITI (European Software Process Improvement Training Initiative) (KEPLER, 1996) demostraron la existencia de problemas en el desarrollo de software a nivel general. Las dificultades se centraban fundamentalmente en los temas relacionados con la ingeniería de requisitos, en las temáticas de especificación, gestión y documentación de los requerimientos de software. También se detectaron los principales factores de éxito y fracaso de este estudio los que se revelarán a continuación.

Principales factores de éxito:

- Implicación de los usuarios.
- Apoyo de los directivos.
- Enunciado claro de los requerimientos.

Principales factores de fracaso:

- Falta de información por parte de los usuarios.
- Especificaciones y requerimientos incompletos.
- Especificaciones y requerimientos cambiantes.

1.3 El origen de la Ingeniería de Requisitos

El origen del término Ingeniería de Requisitos, se atribuye a dos conferencias organizadas por la OTAN en 1967 y 1968. Fue pronunciado por primera vez por Fritz Bauer, en Gamisch, Alemania, en dicha conferencia sobre el desarrollo del software.

Allí se estableció el software como producto y aparecieron las empresas dedicadas al desarrollo y distribución masiva del mismo.

En estudios realizados se ha demostrado que casi un tercio de los proyectos de desarrollo del software se cancelan durante su desarrollo y que la gran mayoría presenta graves desviaciones respecto a plazos y presupuestos iniciales. Atribuyéndoles entre sus principales fuentes de frustración la incapacidad para resolver las necesidades de los usuarios, como consecuencia fundamental de una pobre Ingeniería de Requisitos. (FONT y VÁZQUEZ, 2007)

Los principales problemas que han dado origen a la crisis de desarrollo del software se encuentran en las primeras etapas, donde se hace necesario hacer una adecuada Ingeniería de Requisitos, obteniendo un producto con la calidad requerida.

1.4 Impacto de la Ingeniería de Requisitos.

Aunque la Ingeniería de Requisitos no es el concepto que resolverá todos los problemas, promete tener un impacto altamente positivo en la reducción del tiempo del proyecto y la disminución de los recursos implicados, lo que facilitaría la reutilización de requisitos cuando fuese necesario. A continuación se detallan algunos de los posibles beneficios que puede aportar una eficiente ingeniería de requisitos:

- Fomenta la calidad a través de una serie de características importantes, una de las cuales es la trazabilidad, la conexión de la información relacionada.
- Simplifica la evaluación de impacto de los cambios en los objetivos y requisitos.
- Valida la implementación de especificaciones técnicas de bajo nivel al demostrar cómo se cumplen requisitos de mayor nivel.

- Permite realizar un seguimiento del estado del proyecto, al controlar el progreso del trabajo que se está realizando con el fin de cumplir los requisitos.
- Garantiza que el servicio, el software o el sistema que desarrolle y entregue cumpla con los requisitos del cliente en el plazo establecido y dentro del presupuesto fijado.
- Aumenta la comunicación y la colaboración, garantizando que todos los participantes tengan acceso a los objetivos del negocio comunes que guían la empresa o a los requisitos del cliente que definen el proyecto de desarrollo.

1.5 ¿Qué es la Ingeniería de Requisitos?

La Ingeniería de Requisitos se define como:

“La Ingeniería de Requisitos trata con actividades en las cuales intenta comprender las necesidades exactas de los usuarios del sistema software, para traducir tales necesidades en instrucciones precisas y no ambiguas las cuales podrían ser posteriormente utilizadas en el desarrollo del sistema”

La Ingeniería de Requisitos, disciplina de la Ingeniería de Software, es donde se identifica el propósito del sistema, dirección y alcance. Consiste en un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada de los requerimientos del sistema siguiendo un determinado estándar. Los requisitos constituyen el enlace entre las necesidades reales de los clientes, usuarios y otros participantes vinculados al sistema (stakeholders). (KARAKOSTAS y LOUCOPOLOS, 1995)

El proceso de Ingeniería de Requisitos, es un conjunto de actividades que son seguidas con el objetivo de descubrir, modelar, validar y mantener un documento de requisitos. Este proceso debe lidiar con diferentes puntos de vista, usar una combinación de técnicas, herramientas y personas. Todo este proceso acontece en un universo de discurso con actores reales, por lo que se puede considerar un proceso centrado en las personas. (LANDAZURI, 2005)

Para una investigación más detallada nos apoyamos en Pressman (PRESSMAN, ROGER S, 2005) donde expresa que la ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando sus necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. El proceso de ingeniería de requisitos puede ser descrito en 6 pasos distintos:

- Identificación de requisitos
- Análisis y negociación de requisitos
- Especificación de requisitos
- Modelado del sistema
- Validación de requisitos
- **Gestión de requisitos**

La ingeniería de requisitos como plantea Pressman comienza con la identificación, durante la fase de inicio, los requisitos son escogidos mediante entrevistas con el cliente y son guardados en el documento Visión. En este también se recogen los cambios que se puedan introducir.

Posteriormente se procede al análisis y negociación de los mismos. Los requisitos pueden ser definidos en cada caso de uso, caso de prueba, características funcionales, no funcionales, etc. Los casos de uso se pueden examinar en un diagrama de casos de uso del negocio o del sistema en dependencia de los que se desee modelar.

Esta metodología también define la especificación de requisitos, actividad encargada de asignarle atributos a los requisitos. Entre los atributos principales que define RUP se encuentran: estado, estabilidad, riesgo, esfuerzo, prioridad, entre otros.

En la validación de los requisitos se realiza una revisión técnica los casos de prueba que son definidos para cada uno de los requisitos funcionales del sistema donde intervienen los desarrolladores y el cliente. En ellos se determina si han cumplido o no con su funcionalidad. Este proceso se lleva a cabo en el flujo de trabajo de prueba. Todo proyecto sea grande o pequeño debe abordar la validación de requisitos, para asegurar que todos han sido establecidos sin ambigüedad, inconsistencias y se ajusten a los estándares establecidos en un comienzo para el proceso, proyecto y producto.

No importa el cuidado tenido a la hora de definir los requisitos, en ocasiones estos cambiarán. Reemplazar un requisito no solo se reduce a quitar un requisito y sustituirlo por un nuevo, sino también hay que analizar el impacto que este cambio puede ocasionar en los ya existentes. Es necesario cerciorarse de dar a los requisitos una estructura que sea resistente a los cambios. Como solución a esta nueva interrogante RUP utiliza la trazabilidad para representar las dependencias éntrelos requisitos y otros artefactos del ciclo de vida del desarrollo y de esa manera **gestionarlos**.

1.6 Gestión de Requisitos

Los requisitos se inician cuando empieza un proyecto en las etapas de análisis y especificación de requisitos, posteriormente estos en el ciclo de vida de un proyecto, pueden ser modificados por lo que se establece el concepto de Gestión de Requisitos, que es el tratamiento y control de las actualizaciones y cambios a los mismos.

La Gestión de Requisitos en Ingeniería de Sistemas, es el proceso encargado de la identificación, asignación y seguimiento de los requisitos, incluyendo la interfaz, verificación, modificación y control del estatus a lo largo del ciclo de vida. (THAYER y DORFMAN, 1997) Es el conjunto de actividades que se concentra en el aseguramiento de las especificaciones, por ejemplo, los requisitos que son reunidos para la satisfacción del cliente.

Debido a que un proyecto informático es susceptible de cambios, habría que proceder a su actualización o a la incorporación de nuevas funcionalidades o eliminar otras, esto obliga a mantener controlado y documentado el producto.

Los cambios de requisitos deben ser gestionados para asegurar que la calidad de los mismos se mantenga, los problemas suscitados por los cambios de requisitos podrían incurrir en altos costos, siendo el requisito factor crítico de riesgo. (SOMMERVILLE y SAWYER, 1999)

1.7 Tareas principales de la Gestión de Requisitos

➤ **Recolección**

Recolección y documentación de requisitos es una actividad de comunicación iterativa entre clientes, gerentes y practicantes (stakeholders del proyecto), para descubrir, definir, refinar y registrar una representación precisa de los requisitos del producto. Varios métodos son utilizados para la recolección de requisitos. Algunos análisis iniciales como es la agrupación, categorización y priorización son desarrollados durante esta actividad.

➤ **Documentación**

Después que los requisitos han sido recolectados, hay que analizarlos al detalle y documentarlos en una especificación de requisitos. El resultado de la especificación de requisitos y de cualquier especificación de requisitos de componentes hardware/software derivado sirve como registro de convenio con el cliente y compromiso con el proveedor. Estas especificaciones son rastreadas

utilizando una matriz de trazabilidad de requerimientos y son sujetos a verificación y gestión de cambio a través del ciclo de vida del producto.

➤ **Verificación**

Una vez que la especificación de requisitos ha sido desarrollada, los requisitos son verificados.

La verificación de requisitos es un proceso para asegurar que la especificación de requisito del producto sea una representación exacta de las necesidades del cliente. Este proceso también asegura que los requisitos sean trazados y verificados a través de varias fases del ciclo de vida; particularmente en el diseño, implementación y pruebas. Los requisitos deben ser trazados desde fuentes externas, tales como los clientes, para derivar requisitos del nivel del sistema, para especificar requisitos del producto hardware/software. Además, todos estos requerimientos deben ser trazados al diseño, implementación y pruebas para asegurarse que los requerimientos han sido satisfechos.

➤ **Gestión de Cambios**

Gestión de cambios es un proceso formal para identificar, evaluar, trazar y reportar cambios propuestos y aprobados a la especificación del producto. Como el proyecto va evolucionando, los requerimientos pueden cambiar o expandirse para ajustar algunas modificaciones en el alcance o diseño del proyecto. Un proceso de gestión de cambios proporciona un rastreo completo y preciso de todos los cambios que son pertinentes al proyecto.

El proceso del ciclo de vida de la Gestión de Requisitos, debería ser flexible y adaptable para reunir las necesidades del proyecto. Las características del alcance e implementación del proceso del ciclo de vida de la Gestión de Requisitos en un proyecto, variará dependiendo de algunos factores claves: (LANDAZURI, 2005)

- Tamaño y complejidad del proyecto.
- Experiencia del personal del proyecto.
- Experiencia de los clientes del proyecto.
- Dominio de la aplicación.
- El propósito y uso de esta aplicación.

1.8 La Gestión de Requisitos en la Práctica

La Gestión de Requisitos es una actividad que consume tiempo y debería ser desarrollada por aquellos con un entrenamiento y/o experiencia adecuada. Antes de adentrarse en el proceso, se debe de considerar lo siguiente:

- El equipo del proyecto entiende y sigue un apropiado ciclo de vida del proyecto.
- Son involucrados en el proyecto clientes, usuarios, y todos los (stakeholders) importantes.
- La buena comunicación es importante durante el proceso de derivar requerimientos.
- Es desarrollado un prototipo o una primera versión del sistema software, adicional a un documento de especificación, de tal manera que los requerimientos pueden ser mejor definidos antes de ser considerados como línea base.
- Se debe de mantener una lista de requerimientos en una base de datos donde se puedan manejar más fácilmente.
- Se debe de implementar un proceso de gestión de la configuración de requerimientos.
- No añadir ni cambiar requerimientos sin llevar un análisis de riesgos que determine el impacto en un plan de proyecto y reestimación del costo y planeación del proyecto.
- Algunas capacidades operacionales son entregadas durante el proceso de desarrollo con la intención de añadir requerimientos nuevos.
- Es necesario utilizar un método de trazabilidad de requerimientos a través del ciclo de vida del producto. Este no es un proceso fácil, por tal motivo, se sugiere que se utilice una herramienta automatizada.

(LANDAZURI, 2005)

1.9 Descripción de las Actividades de la Gestión de Requisitos

- **Recolección de requisitos**
 - Planeación de Elicitación
 - Formatos de representación
 - Análisis de requerimientos inicial
 - Especificación de requerimientos

- **Documentación de requerimientos**
 - Formatos de representación
 - Análisis de requerimientos detallado

- Especificación de requerimientos
- Matriz de trazabilidad

- **Verificación de requerimientos**
 - Verificación – una actividad del ciclo de vida
 - Revisión de la verificación

- **Gestión de cambio de requerimientos**
 - Proceso de control de cambio
 - Aceptación de la línea base (baseline)

(LANDAZURI, 2005)

Muchos son los conceptos que se pueden encontrar sobre el tema, pero lo que se conoce es que la Gestión de Requisitos es el último paso que se tiene en cuenta en la Ingeniería de Requisitos.

Según Pressman la **Gestión de Requisitos** es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento. (PRESSMAN, ROGER S, 2005)

1.10 Panorámica

En el mundo actual, las compañías y empresas intentan producir software con buena calidad para satisfacer expectativas del cliente y que este obtenga el producto esperado, para lograr cierta eficiencia se hace necesario la utilización de herramientas informáticas para la Gestión de Requisitos. Las organizaciones con mejores posiciones en el mercado tienden a incrementar el uso de este tipo de herramienta, las más conocidas a nivel mundial se mueven entre propietarias, libres y multiplataforma, destacándose las siguientes herramientas:

- Requisite Pro
- REM
- Caliber RM
- DOORS
- IRqA
- OSRMT

➤ GatherSpace

En Cuba, las empresas productoras de software de manera general, no utilizan herramientas informáticas para la Gestión de Requisitos aunque en los últimos años se aprecian ligeros indicadores de utilización de alguna de estas herramientas.

Las dificultades que se han presentado oscilan entre los problemas de compatibilización de las mismas con los sistemas operativos utilizados en la elaboración del producto software, desconocimiento sobre la herramienta a utilizar y sobre que proceso de Gestión de Requisitos aplicarle.

Algunas empresas cubanas pese a que conocen la existencia de herramientas que facilitan la Gestión de Requisitos, no han dado el salto hacia el estudio, construcción, desarrollo y generalización de algunas herramientas propias para esta actividad, otras ni siquiera han modificado o han adaptado las existentes para su uso en la organización.

En entrevistas realizadas a líderes de proyectos y especialistas en la actividad de la Gestión de Requisitos en proyectos de la UCI y empresas asociadas a esta actividad pudimos corroborar la necesidad de la utilización, implementación o adaptación de algunas de estas herramientas informáticas.

En la UCI, por lo general, se tiene poco conocimiento y uso de las herramientas para la Gestión de Requisitos, aunque muy necesaria sería la utilización de ellas, ya que hasta el momento, esta actividad no se realiza de la manera más eficiente y correcta.

1.11 Ventajas y desventajas

El uso de herramientas de gestión de requisitos es alentado para mejorar tanto la productividad como la calidad en el desarrollo de un proyecto software. La gestión de requisitos es un componente vital en el desarrollo de un software ya que provee la dirección y alcance del proyecto. El uso de estas herramientas se ha convertido en un aspecto importante en la ingeniería de sistemas y el diseño, ellas mejoran tanto la productividad como la calidad en el desarrollo de un proyecto software. Su utilización ayuda a gestionar el proyecto disminuyendo el arduo trabajo en el mantenimiento de los requisitos, de

forma tal que proporcione grandes beneficios al reducir errores y mejorar tanto la productividad como la calidad en el desarrollo de un software. Además de permitir:

- Mejor comunicación interna de los aspectos referentes al proyecto.
- Documentación del proyecto siempre actualizada.
- Posibilidad de realizar análisis confiables sobre el progreso del proyecto, la estimación de los recursos y proyecciones futuras.

Lo que ha motivado a utilizar este tipo de herramientas es la complejidad para la gestión de los requisitos. Un estudio realizado por Meta Group (LANDAZURI, 2005) descubrió que aproximadamente el 60-70% de los proyectos de IT fallan por la pobre recopilación, análisis y gestión de requisitos, esto es porque el éxito de un proyecto software, es aquel que satisface al usuario, si sus requisitos no son completamente definidos y documentados puede afectar a todo el proyecto.

Las herramientas de gestión de requisitos son sofisticadas y complejas por la naturaleza del cual son responsables, son finamente detalladas, sensitivos al tiempo, altamente con dependencia interna y pueden estar continuamente en cambio. Estas herramientas que simplifican tareas complejas requieren de habilidades y un entendimiento total de sus capacidades.

En cualquier proyecto de ingeniería de sistemas, la gestión de requisitos es una tarea intensiva, debe de disponer con la habilidad de relacionar diferentes documentos, de obtener una visión sinóptica de esta relación de documentos, de crear reportes especiales de estos documentos, de controlar los cambios hechos a través del conjunto de documentos de una manera consistente, de acomodar los requisitos estructurados de documentos diversos y tipos de documentos.

El uso de una herramienta de gestión de requisitos proporciona a la organización algunas ventajas como son:

- Ahorro en costes de especificación y de desarrollo minimizando el impacto de errores.
- Mejora la calidad mediante un adecuado análisis y gestión de los requisitos.
- Facilita la reutilización real.
- Mejora la productividad facilitando la reutilización real desde la especificación.
- Reduce las no-conformidades del sistema.
- Permite controlar la especificación.
- Permite administrar más fácilmente la especificación.
- Ayuda a cumplir con estándares de calidad.
- Proporciona un repositorio no propietario de especificación.

- Permite centralizar toda la información del problema.
- Permite especificar sistemas de una forma estructurada y gráfica.
- Proporciona una trazabilidad completa de la especificación, etc.

(FINKELSTEIN y EMMERICH, 2000)

Las desventajas no se superponen ante las ventajas que presenta, pero es importante que se analicen porque pueden afectar el buen funcionamiento y el resultado de las herramientas. Entre ellas está el factor humano; la dependencia de la capacidad y la formación que debe tener el equipo de trabajo que está desarrollando el producto, influyen decisivamente en la calidad de los resultados.

Constituye una dificultad también la poca existencia de estas herramientas de códigos abiertos, y sobre todo para la comunidad de software libre, aunque existan diversas versiones hechas sobre plataforma libre, no todas cumplen con las 4 libertades de GNU/Linux.

1.12 Metodología y Herramienta CASE

RUP es una metodología de desarrollo de software que ayuda a organizar y planificar todo el proceso para poder obtener un producto de óptima calidad y clientes satisfechos con el resultado. En este epígrafe, se describirán las principales características de la metodología RUP (Rational Unified Process) pues es la que se utilizará en el desarrollo de próximos capítulos.

La metodología RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP divide el proceso de desarrollo en ciclos, obteniendo un producto al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante. Este se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). (REYNOSO, 2004)

La herramienta CASE que será utilizada es Visual Paradigm, pues la misma funciona sobre múltiples plataformas, tiene licencia gratuita, es un producto de muy buena calidad, soporta aplicaciones web,

las imágenes y reportes generados son de muy buena calidad, es fácil de instalar y actualizar y es compatible con otras ediciones. Además ofrece un entorno de creación de diagramas para UML 2.0, que es el lenguaje de modelado que se va a utilizar, diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad, uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, capacidades de ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, disponibilidad de integrarse en los principales IDEs y generación de código para Java principalmente y exportación de los proyectos como HTML. (PABLO, 2007)

Después de haber analizado algunos conceptos, definiciones y características de la Ingeniería de Requisitos y uno de sus principales pasos: la **Gestión de Requisitos**, sus tareas y actividades principales, se llega a la conclusión, de este es un paso muy importante a la hora de realizar un software, por lo que debe hacerse de la mejor manera para evitar posibles insatisfacciones del cliente, el uso de herramientas para la **Gestión de Requisitos**, es una de las soluciones propuestas para mejorar esta actividad en la realización de cada proyecto software.

Capítulo II: Análisis de las herramientas

En el transcurso de la investigación se encontraron diferentes herramientas para la Gestión de Requisitos, la mayoría propietarias, muchas de estas con versiones para plataforma libre pero sin la oportunidad de las 4 libertades del software libre.

Una comparación detallada entre las herramientas más importantes, permitió la preferencia de una por encima de las demás, tomando como premisa fundamental su funcionamiento sobre plataforma libre.

2.1 Funcionalidades de las Herramientas de Gestión de Requisitos

La mayoría de las herramientas de Gestión de Requisitos en el mercado realizan funciones similares. Estas permiten a los desarrolladores del sistema importar grandes documentos de una variedad de formatos estándar de procesadores de palabras. Los elementos del documento están sujetos a rigurosos cambios y a un régimen de control de versiones. Pueden ser generados una variedad de vistas de documentos utilizando tanto los atributos como las relaciones, generalmente vistas específicas de trazabilidad tales como matrices de trazabilidad. De la misma manera, plantillas de documentos pueden ser configuradas para crear nuevos documentos compuestos.

Las herramientas de gestión de requisitos son genéricas, esto es que necesitan ser configurados para soportar ingeniería de requisitos específicos y procesos de desarrollo de sistemas. Dichas configuraciones son soportadas por la creación de plantillas de documentos, esquemas, diseño de atributos y tipos de relación y vistas de documentos.

Funciones Básicas

INCOSE identificó las siguientes características básicas necesarias para una herramienta para ser considerada como una herramienta de **Gestión de Requisitos**: (JONES *et al.*, 2007)

- Identificación de requisitos "individuales"
- Asignación a un destino y clasificación de requisitos
- Grupo de requerimientos (recopilación), revisión, identificación / punto de arranque
- Proveer un interfaz de datos básicos.

Una herramienta de **Gestión de Requisitos** puede soportar la disciplina de ingeniería y la de gestión para gestionar requisitos, así mismo, debe poder coleccionar y gestionar los requisitos técnicos y

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

programáticos. Funciones comunes realizadas por la herramienta, consisten en la identificación de requisitos, revisión y edición, rastreo de requisitos a su origen, y generación de informes.

Funciones técnicas requeridas por las herramientas incluye análisis del impacto del cambio. Cuando un requisito es cambiado, se deben de identificar todos los requisitos afectados. Otra función beneficiosa ha ser utilizada es la verificación de la integridad y la consistencia. La función de gestión requerida por las herramientas consiste en la recopilación de métricas y supervisión de la estabilidad de los requerimientos a través del control de cambio. Control de cambio, consiste en mantener la pista de añadir, borrar o cambiar cualquier requisito existente.

Detalle de Funcionalidades

Tomando como referencia el estudio realizado por INCOSE, donde comparan más 35 herramientas comerciales en función de si soportan o no 14 funcionalidades generales divididas a su vez, en funcionalidades más específicas, también se analizaron las encuestas realizadas por Volere con más de 12 años de experiencia donde se especifican como herramientas de gestión 53 incluyendo las de INCOSE y agregando nuevas, se realizó esta investigación tomando como punto de partida para evaluar cada herramienta de las estudiadas las siguientes 7 funcionalidades:

1. Tipos de licencias bajo la que se publica el producto y que sistemas operativos soporta.
2. Captura e identificación de requisitos
3. Captura de la estructura de los elementos del sistema
4. Análisis de trazabilidad
5. Gestión de la configuración
6. Ambiente del sistema
7. Interfaz del usuario
8. Soporte y mantenimiento

Para escoger dichas funcionalidades estuvieron presente las necesidades de los proyectos productivos de la facultad 10 y demás factores como: entrevistas, criterios, opiniones entre otros, para que cada una de ella reflejara una información válida y certera y sirviera de apoyo, soporte y ayudará a tomar la decisión final de la mejor manera y con los argumentos necesarios.

2.2 Proceso de Gestión de Requisitos

El proceso de gestión que se define a continuación es independiente de cualquier metodología, es decir que se puede utilizar en cualquier proyecto. Fue definido a partir de un estudio del comportamiento de la gestión de requisitos en las diferentes metodologías actuales tales como: XP, MSF, Scrum, RUP; tomando sus mejores prácticas.

Dentro de este proceso propuesto los pasos a seguir son:

- Identificar las principales actividades que formarán parte del proceso.
- Construir el mapa de procesos. En él se representan gráficamente las actividades que forman parte del proceso así como estas aportan valor y contribuyen a los resultados.
- Describir cada una de las actividades del proceso.

Principales actividades del proceso de gestión de requisitos.

- Definir artefactos y tipos de requisitos relevantes para la trazabilidad.
- Definir las relaciones entre los artefactos y los tipos de requisitos.
- Definir atributos de los requisitos.
- Elaborar matrices de trazabilidad.
- Mantener y actualizar documentos.

Mapa del proceso

Proceso de gestión de requisitos.

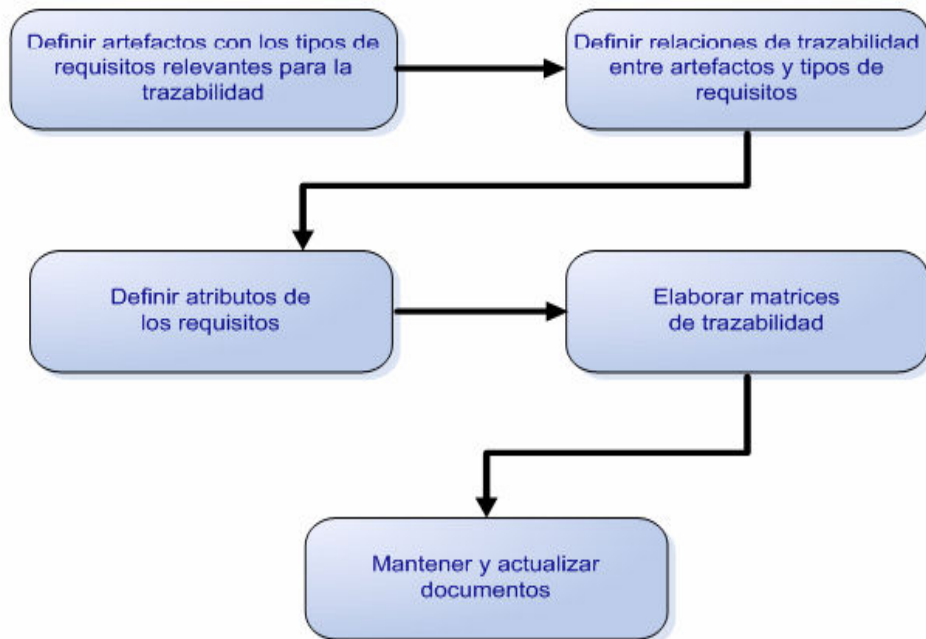


Figura 1. Mapa del Proceso

Mediante este proceso que se encuentra íntegramente en el trabajo de diploma del curso 2006-2007 (FONT y VÁZQUEZ, 2007) y conjuntamente con las entrevistas realizadas permitió establecer un punto de partida para la comparación entre las herramientas de Gestión de Requisitos y determinar cual sería la más adecuada para satisfacer las necesidades de los proyectos productivos de la facultad 10 de la UCI, en este proceso se establecen cuales son los pasos a seguir en la captura, validación y gestión de los requisitos en cualquier tipo de proyecto, independientemente de la metodología y plataforma que se utilice.

2.3 La entrevista y sus resultados.

Durante el desarrollo de la investigación se realizaron entrevistas a diferentes líderes de proyecto y especialistas en la actividad de Gestión de Requisitos. Realizar entrevistas toma tiempo; por lo que no es posible utilizar este método para recopilar toda la información que se necesite en la investigación.

La entrevista es una técnica eficaz para obtener datos relevantes y significativos desde el punto de vista de las ciencias sociales, para averiguar. La información que se obtiene es muy superior que cuando se limita a la lectura de respuesta escrita, a través de ella se pueden captar los gestos, tonos de voz, los énfasis, etcétera, que aportan una importante información sobre el tema y las personas entrevistadas.

La ventaja esencial de la entrevista consiste en que son los mismos actores sociales quienes proporcionan los datos relativos a sus conductas, opiniones, deseos, actitudes y expectativas. Cosas que por su misma naturaleza es casi imposible observar desde fuera.

Resultados

Muchos de los proyectos entrevistados muestran que de alguna forma u otra realizan la gestión de requisitos, algunos centran sus esfuerzos solo en las tareas de identificación y clasificación, sin llegar a gestionarlos en su plenitud. Por otro lado un gran por ciento no la realiza, argumentando como principales causas las siguientes.

- Falta de conocimiento relacionada con el tema.
- No es factible para proyectos pequeños.
- Están en la fase de captura de requisitos aún y no han concebido si la van a realizar o no.

Vale aclarar que ninguno de los proyectos entrevistados realiza el proceso de Gestión de Requisitos como es debido y no utilizan herramientas informáticas para este importante paso en la construcción de cualquier producto software.

2.4 Gestión de Requisitos en los proyectos productivos de la facultad 10.

En los proyectos productivos de la facultad 10 se agrupan estudiantes con conocimientos básicos, que guiados por un líder de proyecto trabajan en conjunto en la realización de un determinado producto software para resolver las necesidades del cliente y contribuir a un mejor desempeño.

Para llevar a cabo una buena gestión de requisitos ante cualquier tipo de software, es necesario seguir un proceso por la importancia que este tiene y si además utilizas una herramienta informática, el proyecto quedará justo a la medida del cliente debido a que los posibles errores que pudieran existir, en la parte de ingeniería del software, serían eliminados, con una buena aplicación de una herramienta y de un proceso para la gestión de requisitos.

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

Algunos consideran que la gestión de requisitos, es una tarea importante dentro la ingeniería, que solo se resume a la identificación y clasificación de los mismos. Otros plantean que solo es necesario en proyectos de mayor envergadura y que en aquellos de menor complejidad se puede prescindir de ella. Por otra parte, algunos dominan este paso de la ingeniería, sus tareas, actividades y los errores que trae consigo una inadecuada gestión de requisitos.

La gestión de requisitos no se resume solo a eso, sino que es una actividad que comprende desde el inicio del proyecto, en la etapa de identificación y análisis de los requisitos y se lleva a cabo durante todo su ciclo de desarrollo. Además no solo abarca las actividades de recolección y documentación, sino también el control y seguimiento de los cambios en los requisitos en cualquier momento. Es aplicable a cualquier tipo de proyecto, aunque de no hacerse en proyectos pequeños, las consecuencias serían menores, pero siempre existirían.

La gestión de requisitos es un proceso que no se tiene en cuenta en la totalidad de los proyectos productivos de la UCI, aunque en los entrevistados en la facultad 10 siempre estuvo presente por el tamaño y envergadura de los mismos, vale decir que, a pesar de aplicar la gestión de requisitos no se hizo en ninguno de la manera más correcta.

Es una actividad que consume tiempo y debería ser desarrollada por aquellos con una experiencia previa o quienes hayan recibido antes un entrenamiento. Hasta el momento, de los proyectos analizados, ninguno se guía por un proceso para gestionar los requisitos, siendo uno de los eslabones fundamentales para lograr la calidad y el factor fundamental a investigar, además de no utilizar herramienta informática alguna.

2.5 Herramientas de Gestión de Requisitos en el mercado

Las herramientas seleccionadas proporcionan casi todas las necesidades básicas exigibles a una herramienta de gestión de requisitos para que sea incorporada a proceso de gestión de requisitos por las empresas. Además, estas herramientas están ampliamente difundidas y son muy reconocidas, como demuestra el hecho de que aparecen siempre en todas las comparativas que se estudiaron, tienen un amplio soporte de las empresas que las desarrollan, y lo más importante es que algunas tienen la posibilidad de ampliar la funcionalidad.

En general, todas se basan en sistemas centralizados de gestión de bases de datos para almacenar la información correspondiente a los requisitos, que suelen consistir en párrafos de texto libre con una

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

serie de atributos predefinidos y a los que la mayoría de herramientas permiten asociar nuevos tipos de atributos por parte del usuario. Todas las herramientas asumen que la estructura de los requisitos es jerárquica, de forma que un requisito puede estar formado o tener asociados otros requisitos de nivel inferior, y la mayoría permite extraer párrafos de ficheros generados por procesadores de texto comerciales y convertirlos en requisitos.

Otras de las características comunes a la mayor parte de las herramientas es la posibilidad de realizar consultas sobre los requisitos en función de determinados valores de sus atributos.

Teniendo en cuenta las entrevistas, el proceso establecido en la UCI y la necesidad de la utilización de una herramienta para los proyectos y la docencia se seleccionaron 6 de las herramientas que cumplen con la mayoría de las funciones:

- **OSRMT v1_5**
- **GatherSpace**
- **Requisite Pro**
- **Caliber RM**
- **DOORS**
- **IRqA 3.0**

OSRMT v1_5:

Es una herramienta de software libre, el nombre completo es Open Source Requeriment Management Tool. Fue diseñada para servir el ciclo de vida completo del desarrollo del software. Es independiente de la plataforma que se vaya a utilizar, pues corre sobre Java. Permite la descripción de los requisitos, además de los documentos relacionados con la Ingeniería de Requisitos, también comprende las distintas matrices de trazabilidad, funcionalidades, casos de usos y casos de pruebas. (SMITH, 2007)

GatherSpace:

Es una herramienta de gran alcance, con toda la gerencia y el uso en línea simple de los requisitos que permite centralizar, modelar y compartir requisitos del software. Es capaz de proporcionar una intuitiva solución para pequeñas y grandes organizaciones. Es capaz de gestionar los requisitos del software utilizando las especificaciones funcionales, además de los casos de usos así como el modelado. Es un software no-instalable, está en línea 100% de las funcionalidades, solo se necesita un navegador.

RequisitePro:

Es una herramienta centrada en documentos, se almacenan los requisitos asociándolos a documentos (aunque también permite guardarlos directamente en la base de datos). Auxilia especialmente el

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

control de cambio de requisitos, con trazabilidad para especificaciones de software y pruebas. Está muy unido a MS Word ya que es partner de Microsoft Development. La herramienta permite el uso de Oracle sobre Unix o Windows como “back-end database” y también soporta SQL Server sobre Windows.

CaliberRM:

Es para sistemas grandes y complejos y proporciona una base de datos de requisitos con trazabilidad. La compañía ve a los requisitos como parte del proceso de gestión de la calidad del software al igual que las pruebas y el trazado de defectos. Caliber está basado en Internet y maneja referencia de documentos, responsabilidad de usuario, trazabilidad, prioridad y estado entre algunas otras características.

DOORS:

A diferencia del resto de las herramientas, considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a RequisitePro y Caliber-RM, que manejan solamente requisitos y sus atributos. Es una herramienta para organizaciones grandes que necesitan controlar complejos conjuntos de usuarios y requisitos de sistemas con una completa trazabilidad. Proporciona buena visualización de tales documentos como jerárquicos y su lenguaje de extensión, permite una gran variedad de soporte de herramientas a ser construidas.

IRqA 3.0:

Herramienta CASE de Ingeniería de Requisitos, diseñada para soportar las actividades realizadas en el proceso de especificación de sistemas. Proporciona y formaliza la comunicación entre cliente y proveedor y entre los distintos miembros del equipo de desarrollo. Facilita la captura, organización y análisis de los requisitos y la especificación de la solución mediante un apoyo metodológico adaptable a cada cliente.

2.5.1 Comparación entre las herramientas seleccionadas.

1. Tipos de licencias bajo la que se publica el producto y que sistemas operativos soporta.

OSRMT v1_5: Está publicado bajo Open Source General Public License (GPL), licencia de software libre. La aplicación es multiplataforma, es independiente del sistema operativo pues corre sobre Java.

GatherSpace: Tiene licencia comercial, se tienen versiones libres del producto y otras sujetas a pagos. Es una herramienta Web, no necesita instalación y es portable 100% a cualquier sistema operativo.

RequisitePro: IBM Rational RequisitePro está disponible para su adquisición mediante pagos a tres tipos de licencias de productos: una licencia de usuario autorizado, una Licencia de Término Fijo de Usuario Autorizado (FTL) y una licencia Flotante. La aplicación funciona sobre los siguientes sistemas operativos o plataformas recomendadas: Windows 2000, Windows NT y Windows XP.

CaliberRM: CaliberRM 2008 consta principalmente de dos licencias además de que existen licencias de solo lectura. Están la licencia para usuarios concurrentes y la licencia de usuario único. La aplicación funciona sobre los siguientes sistemas operativos o plataformas recomendadas: Windows 2000, Windows 2003 y Windows XP.

DOORS: La licencia de usuario final (End User License Agreement (EULA), donde ellos establecen su uso, al producto solo se le puede hacer una sola copia exclusiva para su uso en otro computador. La aplicación funciona sobre los siguientes sistemas operativos o plataformas recomendadas: Windows XP, Windows Vista y Windows 2003.

IRqA 3.0: La herramienta es propietaria publicada bajo Copyright© 2006 de TCP S.I. Todos los derechos reservados. La aplicación funciona sobre los siguientes sistemas operativos o plataformas recomendadas: Windows XP, Windows Vista, Windows 2003 y otros.

2. Captura e identificación de requisitos.

OSRMT v1_5: Se gestionan los requisitos completamente, características específicas, diseño, puesta en práctica, casos de prueba, y cualquier otro artefacto. Los requisitos se pueden importar directamente desde un XML. Los artefactos pueden tener una mejor clasificación si los usuarios especifican una lista de categorías.

GatherSpace: Es una herramienta de gran alcance en la gestión y para manejar y compartir los requisitos del software. En la parte inicial se puede comenzar agregando las características del proyecto y luego continuar con otros artefactos.

RequisitePro: Se gestionan los requisitos completamente, características específicas, diseño, puesta en práctica, casos de prueba, y cualquier otro artefacto. Los requisitos se pueden importar directamente desde un XML. Los artefactos pueden tener una mejor clasificación si los usuarios especifican una lista de categorías.

CaliberRM: Es una herramienta de gran alcance en la gestión y para manejar y compartir los requisitos del software. En la parte inicial se puede comenzar agregando las características del proyecto y luego continuar con otros artefactos.

DOORS: Se integra con Report Miner (RM) suite, para capturar, tracear, manejar y analizar una amplia información, para llevar a cabo una buena gestión de requisitos y mantener los estándares especificados. Se combina con base datos RM siendo una poderosa combinación, y un intuitivo documento de estilo.

IRqA 3.0: Se realiza la Captura de requisitos, Recopilación de la información facilitada por los usuarios, de forma sencilla y organizada. IRqA® permite al usuario definir distintas vistas para cada uno de los elementos de la especificación, de forma que la información a visualizar sea la que más le interesa en función de las operaciones que desee realizar en cada momento.

3. Captura de la estructura de los elementos del sistema.

OSRMT v1_5: Se pueden incorporar y mantener organizados los artefactos del diseño y la puesta en práctica de los mismos. Puede tener textos y unir documentos o cualquier accesorio de un archivo.

GatherSpace: Se pueden agregar e incorporar paquetes, se tienen en cuenta la característica de los requerimientos.

RequisitePro: Las opciones graficas son variadas, por ejemplo la matriz demuestran las relaciones entre los requisitos, existe un gran acoplamiento entre otras herramientas de modelado que permite un acoplamiento entre los elementos gráficos del modelo del análisis o del diseño y sus requisitos relacionados.

CaliberRM: El diseño es intuitivo y fácil de empleo. Proporciona un gestor central y asegura el la base de datos para los requisitos del proyecto. Se capturan los casos de usos para que los analistas, probadores, los encargados de la comercialización estén consciente del ciclo de vida.

DOORS: Puede capturar gráficamente la implementación del sistema como (la arquitectura, descomposición funcional) y mostrarlas de tal manera que los requisitos puedan asociarse a esta captura grafica.

IRqA 3.0: Se capturan los elementos estructurales genéricos siguientes: Diagramas de bloque: Representan bloques y las relaciones entre ellos, y se utilizan para navegar a través de la especificación y para demostrar la trazabilidad entre los elementos contenidos en bloques relacionados.

4. Análisis de trazabilidad.

OSRMT v1_5: Mantiene las matrices de trazabilidad, acoplamientos individuales de los artefactos y estos se pueden importar. Mediante los filtros o los reportes de los usuarios se puede identificar la terminación de los requisitos.

GatherSpace: Genera un informe de trazabilidad que demuestra el acoplamiento bidireccional entre las características y casos de usos, además de las características y los requisitos.

RequisitePro: Proporciona un fácil mecanismo de preguntas que permiten un eficiente acoplamiento en la trazabilidad. El árbol de trazabilidad permite darle un seguimiento completo a los requisitos y sus relaciones.

CaliberRM: Permite acceder a la base de datos y marcar la trazabilidad de los requisitos durante el ciclo de vida. Permite que los requisitos sean asociados a los artefactos. Analiza el impacto a través de los casos de usos y mediante la trazabilidad se muestra el alcance de los casos de usos.

DOORS: Una vez que las asociaciones estén completas los usuarios desearan verlas y estas pueden ser mostradas. Los requisitos pueden establecerse por los usuarios y establecer una trazabilidad para ver de donde vienen y hacia donde van.

IRqA 3.0: La matriz de trazabilidad permite que los usuarios examinen las relaciones entre los requisitos y otros elementos relacionados con el sistema (conceptos, panoramas de la prueba, servicios y diagramas).

5. Gestión de la configuración.

OSRMT v1_5: Se pueden establecer líneas bases, se guarda el historial de gestión de cambios y las versiones realizadas. Mantiene la seguridad mediante contraseñas para evitar que se cambien los artefactos por personas no autorizadas.

GatherSpace: Existen flujos básicos establecidos por el usuario y estos terminan al especificar cual es la respuesta del sistema.

RequisitePro: Se lleva un historial completo de cada requisito. Se almacena fecha, autor, tiempo y el contenido existente y los cambios realizados por el autor. Todo el historial se pone al día cuando un requisito cambia. Se incluyen líneas base de desarrollo.

CaliberRM: Se proporcionan capacidades de los requisitos basadas en su alcance que ayudan a los líderes de proyectos a planear el alcance, horario y recursos del proyecto. Captura versiones de los requisitos y se puede establecer líneas bases para el desarrollo.

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

DOORS: Una vez que los requisitos hayan sido capturados se necesita conocer cuando estos van siendo modificados, por quien, por que es que se cambian y estas funciones algunas son automáticas y otras no. En algún momento se necesitara establecer las líneas bases para los requisitos. Las líneas bases se muestran pero no se pueden modificar después de ser establecidas.

IRqA 3.0: Mantiene un historial de los cambios realizados a los requisitos de quien, cuando, que, donde, por que y como fueron hechos. El historial se puede mantener también con las versiones guardadas (registrando los cambios realizados). Esto permite que el usuario pueda comparar las diferentes versiones de los requisitos y ver las diferencias entre ellas.

6. Ambiente del sistema.

OSRMT v1_5: Permite varios usuarios conectados, la herramienta es multiplataforma, tiene soporte para múltiples gestores base datos comerciales como MS Access, Oracle, o SQL Server y otras libres como MySQL Server PostgreSQL y los requerimientos del hardware son mínimos.

GatherSpace: Se permite que un mismo proyecto tenga varios usuarios, el ambiente trabajo es vía Web y es muy intuitivo.

RequisitePro: Soporta múltiples usuarios. Soporta el uso para varias bases de datos comerciales como MS Access, Oracle, o SQL Server. Los requerimientos del hardware son medios.

CaliberRM: Soporta múltiples usuarios. Provee un repositorio central, seguro para administrar requerimientos de proyectos a través del ciclo de vida de las aplicaciones, mejorando la comunicación entre todos los miembros del equipo. Los requerimientos del hardware son medios.

DOORS: Muestra una ayuda múltiple (textos y visuales) para los usuarios. Los requerimientos de hardware dependen de la cantidad de usuarios y el tamaño de la base de datos establecida.

IRqA 3.0: Soporta la conexión de múltiples usuarios siempre y cuando el gestor de base datos lo permitan. Se puede utilizar en múltiples sistemas operativos.

7. Interfaz del usuario.

OSRMT v1_5: Se pueden abrir múltiples artefactos, las actualizaciones se reflejan en las vistas, todos los artefactos se pueden exportar y la interfaz de trabajo es un simple navegador Web o una aplicación de Escritorio.

GatherSpace: Por lo menos para comenzar se debe agregar un usuario para usar el GatherSpace. Se puede administrar todo el proyecto mediante este usuario que es un administrador. Para el uso de la herramienta se puede utilizar un navegador Web.

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

RequisitePro: La interfaz es muy intuitiva en lo que se debe hacer, se puede realizar reportes, trabajar en los documentos de requerimientos, estos se pueden editar vía WEB y todo es posible realizarlo al mismo tiempo.

CaliberRM: Está arquitectónicamente diseñado para comunicarse eficientemente vía TCP/IP, donde se da la posibilidad a los usuarios una comunicación instantánea. Tiene un entorno multiusuario basado en WEB manejando un versionado automático de requerimientos.

DOORS: Los usuarios tiene la posibilidad de de generar informes y trabajar con los requisitos al mismo tiempo. Cualquier cambio se refleja si afecta a otros componentes. La interfaz es basada en Windows. Tiene una vista WEB disponible que se puede utilizar con una conexión a la BD.

IRqA 3.0: La herramienta fue diseñada para que tenga una vista de ventanas independientes, esto permite que se puedan abrir al mismo tiempo diversas vistas del mismo proyecto y que estas se puedan manejar en paralelo.

8. Soporte y mantenimiento.

OSRMT v1_5: Es un producto libre publicado bajo licencia libre GNU General Public License (GNU GPL). Las actualizaciones del software no tienen costo adicional. El sitio oficial es <http://www.osrmt.com> y se puede encontrar soporte en el **SourceForge** en los foros de discusión.

GatherSpace: Se puede encontrar soporte en support@gatherspace.com, y tiene versiones de uso libre de la herramienta. El sitio oficial es www.gatherspace.com.

RequisitePro: La ayuda y mantenimiento de la herramienta están disponibles para su uso en el sitio <http://www.ibm.com/software/awdtools/reqpro/support/>. Las versiones de mejora del producto están disponibles alrededor de 18 meses y estas están disponibles para los usuarios que estén bajo contrato.

CaliberRM: Tiene una guía rápida de aprendizaje para disminuir los riesgos de perdida de tiempo en el proyecto. Permite la discusión en línea. Se puede encontrar soporte en <http://www.borland.com/us/products/caliber/index.html>.

DOORS: Tiene garantía establecida de 30 días. Las actualizaciones normalmente están disponibles de 6-12 meses facilitándoles el soporte a los usuarios. Se dan anualmente conferencia para grupos de usuarios y podemos encontrarlas en el sitio oficial, www.telelogic.com/products/doors/doorsanalyst/index.cfm.

IRqA 3.0: El soporte garantiza la utilización del producto por un periodo de 90 días según especifica el manual de usuario. Por las nuevas funcionalidades deseadas se solicitan y son sujetas a pagos. La ayuda se puede encontrar en <http://www.visuresolutions.com/products/irqa/irqa.php>.

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

Durante el desarrollo de este capítulo se fueron mostrando las principales características y funcionalidades de cada una de las herramientas de Gestión de Requisitos escogidas para la investigación. Después de haber analizado cada una, se puede afirmar que existen varias similitudes entre ellas.

Todas tienen un alto nivel de personalización desde cualquiera de sus ambientes de trabajo, permitiendo un buen desarrollo y manejo de la misma a cada empresa y cliente que requiere de su uso.

Aunque existan tales similitudes, también existen las diferencias que marcan la preferencia de unas por encima de otras.

Su principal diferencia radica en su disponibilidad para las diferentes plataformas virtuales que existen en el mundo, y su accesibilidad, como se muestra en la tabla comparativa antes expuesta.

En el caso de las herramientas IRqA y Requisite Pro son totalmente propietarias y están sujetas a pago y otras como CaliberRM y DOORS tienen determinadas restricciones y funcionan solo bajo plataformas totalmente privadas. Sin embargo GatherSpace es un navegador web y puede funcionar bajo cualquier plataforma aunque posee versiones libres y algunas sujetas a pago.

En cambio una herramienta como OSRMT que es gratis y está publicada bajo la licencia libre GPL, es multiplataforma y totalmente independiente del sistema operativo en que funcione, dado que su lenguaje de programación es Java, y que además posee un código completamente abierto, sería la herramienta más adecuada a aplicar en los proyectos productivos para la gestión de requisitos.

Esta herramienta desde su creación posee 5 versiones, cada una mejorando su funcionamiento y tratando de satisfacer al máximo las necesidades de sus usuarios y clientes. La versión OSRMT1.5 está muy bien elaborada y diseñada, cumple con las características básicas de una herramienta de gestión de requisitos además de poseer algunas características auxiliares. Sus funcionalidades son similares a las de muchas herramientas propietarias y su utilización le permite a cada proyecto controlar la especificación, minimizar el impacto de errores, reducir las no conformidades, mejorar la calidad mediante un adecuado análisis y gestión de los requisitos y además que proporciona una trazabilidad completa de la especificación.

La aplicación de esta herramienta contribuirá al buen desarrollo y al éxito de cualquier proyecto ya que para realizarlo y que quede con la calidad requerida y satisfaga las necesidades del cliente, es necesario entre otras características de la utilización de la gestión de requisitos. Cuanto mejor sea la

comunicación y gestión, mayor será la oportunidad de que los proyectos se entreguen en tiempo y dentro del presupuesto establecido.

2.6 Herramienta OSRMT Open Source Requirements Management Tool.

Características generales de OSRMT

La herramienta cumple con el proceso de Gestión de Requisitos establecido en la UCI y las necesidades de esta, contiene un interfaz amigable e intuitiva para el usuario, es independiente de la plataforma y cumple con una de las características determinantes en la comparación que es un producto de software libre publicado bajo licencia GPL.

Las características de OSRMT se corresponden con su facilidad para el manejo y excelente funcionalidad durante todo el ciclo de vida de un software.

Como inconveniente se tiene que el módulo de Reportes, no funciona correctamente y no cumple con todas las necesidades de la UCI.

Características específicas de OSRMT

Base de datos

La base de datos utilizada en la instalación fue PostgreSQL 8.1, aunque permite la migración completa de la herramienta sin dependencias de otras bases de datos a las siguientes: MS Access, MySQL, SQL Server y Oracle, que cualquiera de ellas actualmente son las más utilizadas en el mercado internacional según las necesidades específicas.

Licencia

Está publicado bajo Open Source General Public License (GPL) por Aron Smith, licencia de software libre. Que permite realizar cualquier tipo de cambio en el software mientras se mantenga los términos de la licencia.

Sistema Operativo

La herramienta es considerada multiplataforma debido a que es posible su instalación bajo cualquier sistema operativo, pues su código fuente es java para su instalación solo requiere la máquina virtual de java, la cual existe en casi todos los sistemas conocidos.

Lenguaje de programación

JAVA.

Interfaz en la que se utiliza

Java Swing browser.

Requerimientos de hardware y software

- **Hardware**

Para una PC cliente:

Computadora con cualquier tipo CPU a 1 MHz

Memoria RAM de 128 MB

Capacidad de Disco Duro en dependencia de documentos que se quieran guardar.

Para una PC servidor:

Computadora con cualquier tipo CPU a 1 MHz.

Memoria RAM de 256 MB

Capacidad de Disco Duro en dependencia de documentos que se quieran guardar.

- **Software**

Cualquier sistema operativo libre o privativo.

Se puede utilizar o no un servidor Web en dependencia de lo que se necesite.

Servidor Base de Datos.

El servidor Web y de base datos pueden o no estar en la misma máquina.

Se recomienda:

Para el servidor Web utilizar Apache 1.3.x ó Apache 2.x. En caso de no utilizar Apache, puede instalar cualquier servidor que sea capaz de ejecutar scripts hechos en SQL.

Para la base de datos utilizar PostgreSQL. La utilización de otras bases de datos será en dependencia de las necesidades de cada cliente.

Sistema de ayuda al usuario

A la herramienta se le desarrolló un manual de usuario general ya que el que tenía era una versión atrasada y algunos puntos esenciales no se tocaban, además de que se encontraba enfocado solamente al sistema operativo Windows.

Posee ahora un manual de usuario para ambas plataformas (Ver Anexo 6) y si le quedara alguna duda puede auxiliarse en foros internacionales de discusión acerca de esta herramienta y su funcionamiento.

Instalación

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

La instalación básica está sostenida en el manual de instalación que se desarrolló, permitiendo que se utilice la herramienta en los sistemas operativos GNU/Linux. Solo se necesita instalar la máquina virtual de java y luego proceder a la instalación recomendada. Finalmente con la base de datos deseada que se instale en la PC procedemos a establecer la conexión. (Ver Anexo 3)

Interfaz gráfica

En la interfaz gráfica que muestra en el escritorio, pueden ser abiertos múltiples artefactos, actualizados o creados. Presenta propiedades de configuración que permite la personalización de la aplicación de manera visual sin necesidad de modificar el código fuente.

Equipo de trabajo

Cuando se analizaban anteriormente las características generales, se plasmaba la importancia del equipo de trabajo y es una de las cosas por la que se aboga en el software libre (el trabajo en comunidad). Se puede determinar la agrupación de los especialistas en grupos de trabajos y crear un sistema más robusto durante el ciclo de vida del proyecto.

Usuarios

A medida que se necesite incorporar nuevos usuarios a la aplicación, se pueden agregar cuentas de usuarios, cambiarlas o invalidarlas. Una vez creadas se le asigna un rol y al menos un grupo al que va a pertenecer y la contraseña asociada al usuario, que por directivas de seguridad, esta debe ser cambiado cuando se entra en la aplicación por primera vez.

Grupos

Soporta grupos de trabajo con usuarios diferentes y con conexión al mismo equipo. Los grupos están compuestos por usuarios que tienen el mismo rol dentro del proyecto de trabajo.

Permisos

La concesión de permisos siempre es importante para lograr una mayor integridad y seguridad de los datos. En dependencia de la función que realice y al grupo de roles a que pertenezca le son asignados los permisos.

Acceso restringido a usuarios registrados.

Gestión de privilegios sobre determinadas tareas.

Estos se manejan a través de la base de datos, donde según el tipo de usuario y el rol que esta tiene dentro del equipo de trabajo así será el grupo donde será ubicado. Luego de acuerdo a los permisos así será el nivel de acceso que tendrá sobre todas las partes de la aplicación. Se puede establecer un sistema de autenticación a través de LDAP, viene implementado solo restaría ponerlo a funcionar.

Roles

La definición de roles está dirigida para lograr un mejor manejo de tareas, y contribuir con el control de permisos de acceso. La herramienta maneja el sistema de roles específico, donde los diferentes usuarios del sistema serán ubicados en estos grupos de roles según su función:

- Administradores
- Analista del negocio
- Desarrolladores
- Documentadores
- Jefes de proyectos
- Reguladores de la calidad
- Probadores de casos de prueba

2.7 Aplicaciones adicionales

La versión OSRMT 1.5 puede integrarse con otras aplicaciones que le permitan su buen funcionamiento.

Se utiliza **pgAdmin 3** para administrar la base datos: toda la administración posibles creaciones de nuevos campos pueden ser gestionados aquí.

El **JasperReports con el iReport**: para el manejo y creación de nuevos reportes.

La herramienta **iReport** es un constructor/diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes.

2.8 Seguridad de la información

OSRMT cuenta con un sistema de exportación e importación de artefactos que permite hacer copias de seguridad (backup), donde se les puede realizar a toda la información que se maneja como:

CAPÍTULO II: ANÁLISIS DE LAS HERRAMIENTAS

requisitos, casos de usos, casos de pruebas, características para en un determinado momento recuperar la información.

Además la base datos donde se guarda la información esta sujeta a todas las directivas de seguridad que establece PostgreSQL, y para la administración de la misma solo el administrador posee la contraseña y a través de pgAdmin3 puede administrar la misma. Toda la información de los artefactos que se generan son guardados en la base datos y bien identificados en el sistemas de tablas predefinidos.

Con el estudio que se realizó en el presente capítulo se pudo determinar que existen una gran variedad de herramientas de Gestión de Requisitos, que básicamente sus funcionalidades son similares y que las diferencias están dadas en cuanto a las necesidades de la facultad y de la UCI. Se hizo énfasis en el Proceso de Gestión de Requisitos utilizado como guía en el desarrollo de la investigación y como resultado, la propuesta de la herramienta libre a utilizar en la Gestión de Requisitos de proyectos productivos de la facultad 10, así como sus funcionalidades, características generales y específicas. Como continuidad del estudio se encontraron algunas inconformidades en el módulo de Reportes, al no cumplir con todas las necesidades de la UCI y se le hace la propuesta de personalización en el Capítulo III.

Capítulo III: Análisis y Diseño del Módulo Reportes.

Después de un análisis profundo de las herramientas de gestión propuesta, se determino que se necesita un modulo dentro de la herramienta que se encargue de los reportes, la versión actual no da una solución a tal problema y se propone desarrollar un sistema general de reportes que aporte soluciones reales, eficientes y satisfactorias a las necesidades de la universidad. Brindándole a los usuarios la posibilidades de trabajo, que tienes todas las herramientas de gestión. El sistema estará confeccionado de tal manera que las opciones que se le muestran al usuario sean las mas intuitivas posibles de los tipos de reportes a automatizar. Este sistema automatizado debe cumplir con las normas y estándares para que continúe siendo un producto de software libre bajo la licencia que fue publicado. Debe ser capaz de crear, buscar y actualizar reportes además de que puedan ser guardados en los formatos establecidos en un directorio dado.

3.1 Modelo del Dominio

Actualmente no se ha construido ni formalizado cuales serán los desarrolladores para el proyecto OSRMT y el proceso de reportes no se encuentra establecido debido a la inexistencia de un procedimiento que describa los pasos lógicos que debe seguir el proceso. Esto implica que no se encuentren bien establecidas las fronteras para tales procesos de negocio. Por tales motivos se obtiene a través de un Modelo de Dominio, la relación que se establece entre los conceptos fundamentales que serán manipulados por el sistema.

3.1.1 Conceptos fundamentales del modelo de dominio del módulo Reportes.

Usuario: Es la persona que hace uso de la herramienta, que necesita ciertos privilegios del sistema para poder acceder a los reportes.

Crear Reporte: Es la operación de emitir un nuevo reporte para llevar el control de determinadas acciones o actividades del proyecto.

Crear Formato Reporte: Es la acción de crear un nuevo formato de Reporte (.XML) con el iReport.

Actualizar Formato Reporte: Es la acción de modificar un Reporte (.XML) que haya sido creado anteriormente en el iReport.

Eliminar Formato Reporte: Es la acción de eliminar un Reporte (.XML).

HTML-PDF: Es un convertidor interno para convertir a estos formatos.

BD: Es la que representa la persistencia de datos en la Base Datos.

Matriz, Artefactos, Líneas Bases, Administración: Son los tipos de Reportes (.XML) ya creados con el iReport.

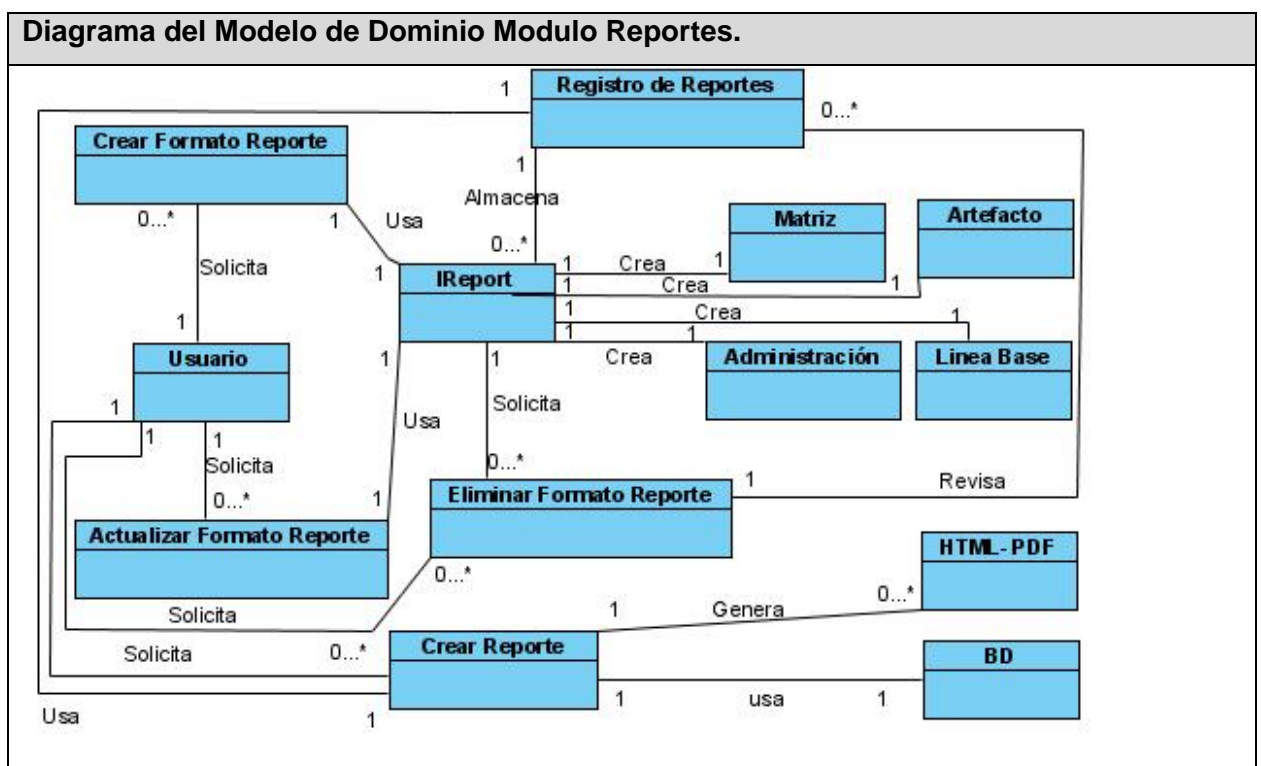


Figura 2. Diagrama del Modelo de Dominio Modulo Reporte.

3.2 Diagrama de paquetes.

Para un mejor entendimiento del sistema se presenta el siguiente diagrama de paquetes. El que se encuentra de color azul es el que propone implementar y se destacó con otro color para que se entienda como queda su ubicación dentro de la herramienta. Partiendo de la necesidad de implementación del módulo de reportes para el sistema propuesto.

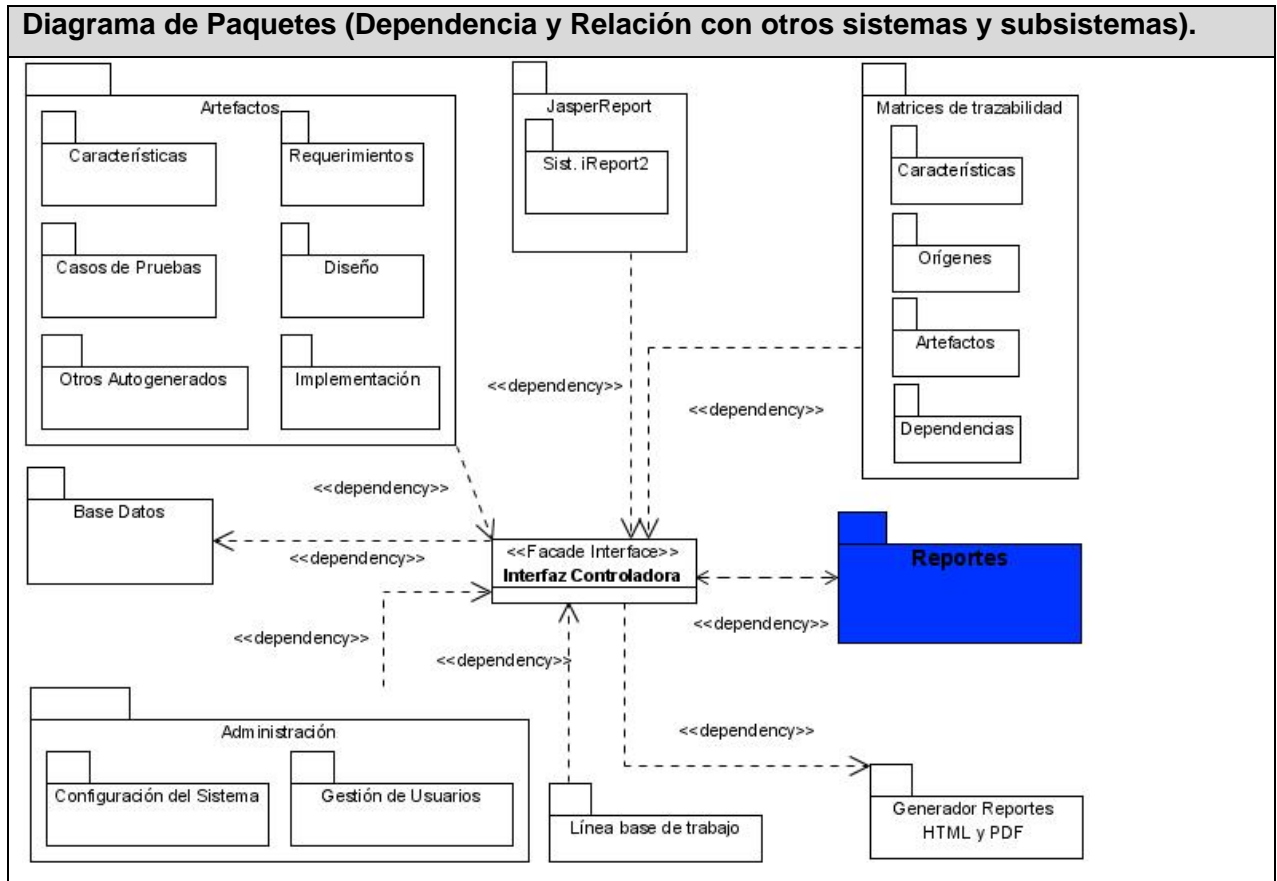


Figura 3. Diagrama de Paquetes (Dependencia y Relación con otros sistemas y subsistemas).

3.2.1 Dependencias y Relaciones con otros softwares.

De acuerdo con el diagrama de paquetes que esta en la **Figura 3** se representa de forma general como se encuentra la organización de la herramienta. A continuación se presenta una descripción de cada paquete por separado incluyendo la propuesta que es el paquete de Reportes.

JasperReport

El sistema para Generar Reportes para la herramienta OSRMT se va a relacionar con otros sistemas para que las prestaciones que brinda funcionen correctamente. A continuación se describe el sistema del cual este depende o se relaciona.

Herramienta iReport con JasperReport.

Es una herramienta visual que sirve para generar ficheros XML (plantillas de informe) que se puedan utilizar con la herramienta de generación de informes JasperReport. Esta herramienta se integra con OSRMT de la siguiente manera: Mediante esta herramienta se crean una versión compilada para OSRMT de reportes .jrxml y la versión generada es .jasper y aquí podemos personalizar los reportes que se realizarán (Ver Anexo Manual de Usuario).

Artefactos

En el paquete de Artefactos se encuentran todos los tipos de artefactos que se pueden generar en un proyecto de software, aquí se pueden generar Requisitos, Características, Casos de Pruebas, Diseño e Implementación de un software, este incluye todos los atributos: relaciones, dependencias, nombres y otros. Estos son los que determinan el posible funcionamiento del proyecto manteniendo una organización y guía en el trabajo de los desarrolladores.

Matrices de Trazabilidad

Este paquete representa una de las principales necesidades que lleva al uso de un software para realizar la Gestión de Requisitos, ya que es casi imposible realizar la trazabilidad de los requisitos de manera manual y a medida que crece el sistema a desarrollar los errores pueden ser mayores. Aquí es donde se representa de manera visual cual es la relación que puede existir entre los artefactos de un sistema. Se generan de manera automática las matrices siguiendo un patrón establecido.

Administración

Este paquete es el encargado de manejar la administración de los usuarios que tienen permiso para hacer uso de la herramienta. En él se realizan todas las gestiones de permisos de los usuarios, los roles que estos van a cumplir dentro del proyecto y los permisos que estos van a tener para hacer uso de las distintas partes de la herramienta.

Líneas Bases

En este paquete se manejan las líneas de trabajo que guiarán el trabajo del proyecto durante todo su desarrollo.

Base Datos

El paquete de Base Datos es el que contendrá toda la información necesaria para la realización de cualquier reporte, aquí se gestionan todos los datos guardados de manera persistentes y necesarios para el normal desarrollo del proyecto.

Generador Reportes HTML y PDF

Luego de tener creado el reporte deseado este paquete se encargará de generar el reporte a los formatos PDF o HTML según las necesidades del usuario.

Reportes

El paquete de Reportes se encuentra marcado en azul por que es la propuesta a automatizar y dentro de el es donde se hará todo el proceso de creación, generación y almacenado de los reportes. A continuación se explica más detalladamente como funcionará este paquete y cuales son sus funciones.

3.3 Descripción de los requisitos funcionales y no funcionales del paquete.

Requisitos funcionales.

Los requerimientos del sistema definen las funciones que el sistema será capaz de realizar. Son el conjunto de propiedades que debe cumplir el software para ser exitoso en el entorno en el cual se usará. Estos deben ser comprensibles por clientes, usuarios y desarrolladores, deben tener una sola interpretación y estar definidos en forma medible y verificable. (Jacobson 2004). Para mantener el estándar de la aplicación se tuvo en cuenta los requisitos funcionales de los demás paquetes.

Los requisitos funcionales del sistema son:

R1: Permitir crear un reportes de todas las matrices de trazabilidad, de cualquier tipo de artefacto, teniendo acceso al método generar matriz del módulo Matrices de Trazabilidad dando la posibilidad de guardarlo en un directorio o imprimirlo.

R1.1: Crear Reporte Matriz de seguimiento de orígenes (identifica el origen de cada requisito) dando la posibilidad de guardarlo en un directorio o imprimirlo.

R1.2: Crear Reporte Matriz de seguimiento de características (es definida por el cliente) dando la posibilidad de guardarlo en un directorio o imprimirlo.

R1.3: Crear Reporte Matriz de seguimiento de dependencias (indica cómo se relacionan los requisitos entre sí) dando la posibilidad de guardarlo en un directorio o imprimirlo.

R1.4: Crear Reporte Matriz de seguimiento de artefactos (vincula a los requisitos con los artefactos que los manejan) dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2: Crear Reporte de todos los artefactos del proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.1: Crear Reporte de Características que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.2: Crear Reporte de Requisitos que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.2: Crear Reporte de Casos de Usos que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.2: Crear Reporte de Diseño que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.2: Crear Reporte de Implementación que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R2.2: Crear Reporte de Casos de Pruebas que se generan en el proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R3: Crear Reportes de la administración del proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R4: Crear Reporte de las Líneas Bases del proyecto dando la posibilidad de guardarlo en un directorio o imprimirlo.

R5: Crear Formato Reportes con la herramienta IReport.

R5.1 Crear Formato Reportes con la herramienta IReport de Administración.

R5.2 Crear Formato Reportes con la herramienta IReport de Matrices.

R5.3 Crear Formato Reportes con la herramienta IReport de Líneas Base.

R5.4 Crear Formato Reportes con la herramienta IReport de Artefactos.

R6: Actualizar Formato Reportes con la herramienta IReport.

R6.1 Actualizar Formato Reportes con la herramienta IReport de Administración.

R6.2 Actualizar Formato Reportes con la herramienta IReport de Matrices.

R6.3 Actualizar Formato Reportes con la herramienta IReport de Líneas Base.

R6.4 Actualizar Formato Reportes con la herramienta IReport de Artefactos.

R7 Eliminar Formato de Reporte.

3.3.1 Requerimientos no Funcionales

Los requisitos no funcionales especifican propiedades, que de una forma u otra restringen el entorno del sistema o de la implementación como por ejemplo rendimiento, interfaz de usuario, facilidad de mantenimiento, dependencias de la plataforma, entre otros. (JACOBSON, 2004) Para mantener el estándar de la aplicación se tuvo en cuenta los requisitos no funcionales de los demás paquetes.

Apariencia o interfaz externa

La interfaz de usuario debe ser apropiada para un control de operaciones efectivo. Esta debe ser sencilla, amigable, intuitiva y de fácil para el usuario, con el objetivo de evitar la resistencia humana al

uso del nuevo sistema, ya que el factor humano determina en gran medida el éxito o el fracaso del mismo.

Usabilidad

El sistema será flexible y de fácil aprendizaje, pues se trata en todo lo posible de mantener un estándar de operatividad que logre que las interacciones del usuario con el sistema sean predecibles y familiares.

Rendimiento

La aplicación se mantendrá dentro de lo establecido para mantener el rendimiento que hasta ahora ha tenido. Mantendrá el poco consumo de memoria y tiempo de trabajo.

Portabilidad

El sistema mantendrá su programación en Java para que sea independiente de plataforma Y continúe siendo portable hacia cualquier sistema operativo que se pueda instalar la Máquina Virtual de Java (Windows, Linux, Unix...). Continuará siendo independiente del SGBD ya que puede utilizar cualquier gestor de base datos (PostgreSQL, MySQL, SQL, MS Access, Oracle), con la posibilidad de migración a estos gestores sin afectar la operatividad del sistema.

Seguridad

El sistema tendrá acceso diferenciado con contraseña para cada usuario, con permiso de acceso al sistema. Además tendrá un registro individual de todas las transacciones efectuadas en tablas, con indicación de fecha, horario, usuario (historial de los cambios efectuados).

Software

EL software será totalmente adaptable a las necesidades de los usuarios. Continuará siendo Libre de acuerdo a las licencias bajo las cuales fue publicado.

Hardware

Los requerimientos mínimos del hardware son: para una PC cliente 128 MB y un servidor alrededor de 256 MB, velocidad mínima de 1 MHz para cliente y servidor. Esto no incluye los documentos guardados en la base de datos.

Ayuda y documentación en línea.

El sistema contará con una ayuda de funcionamiento del software, manual de usuario e instalación, además de la existencia online de los foros de discusión y la ayuda online que se podrá encontrar en internet.

3.4 Diagrama de Casos Usos del sistema.

En la siguiente figura se muestra el diagrama de casos de uso del sistema a automatizar para el paquete de Reportes.

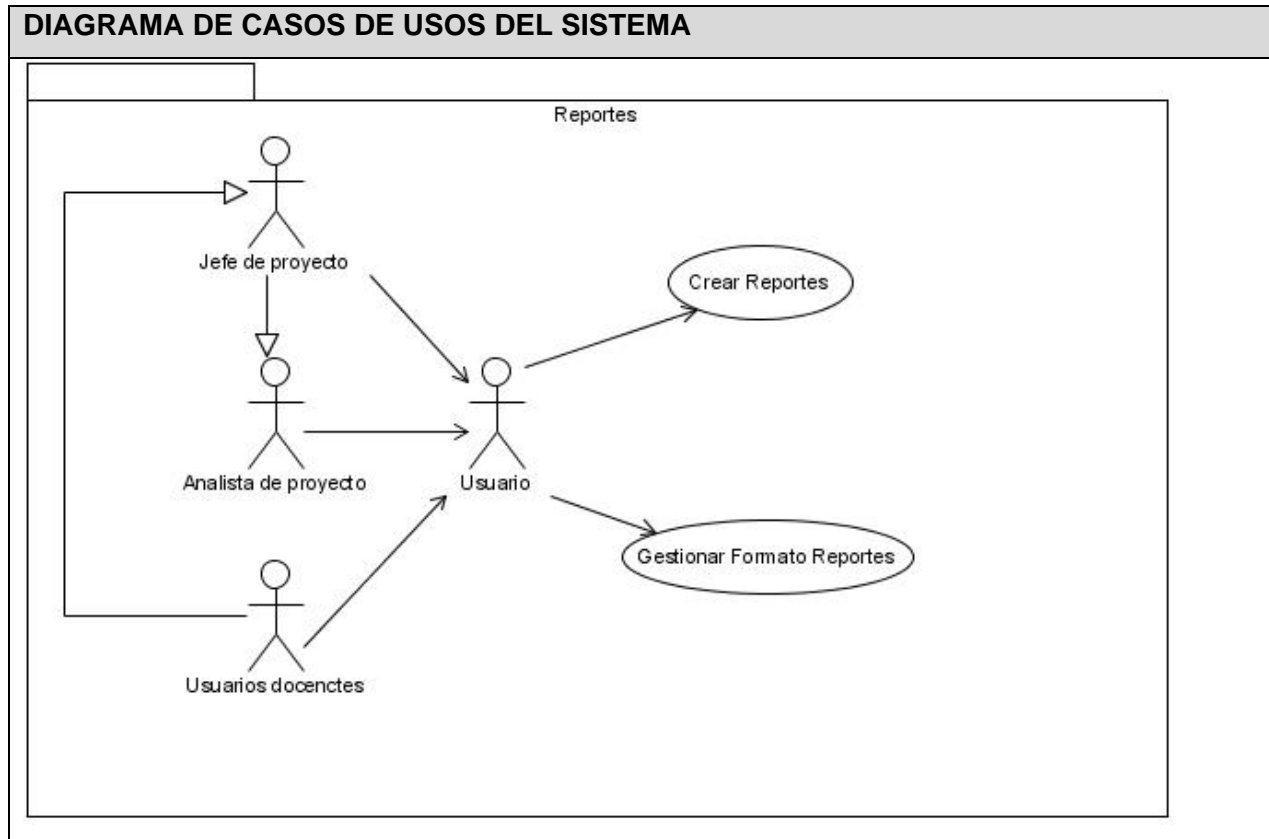


Figura 4. Diagrama de Casos Usos del Sistema.

Por la importancia que tienen todos los casos de usos del sistema se desarrollaran en la primera iteración.

Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Crear Reportes	Reportes	Permite crear un nuevo reporte en el sistema cuando se necesite nuevas entradas para el control de estadísticas, donde este reporte

			utilizará los formatos de plantillas XML, además este caso de uso incluirá si el usuario desea imprimir este reporte o lo desea guardar en un directorio específico en formato HTML o PDF.
CU-2	Gestionar Formato de Reportes	Reportes	Este caso de uso se divide en secciones o escenarios los cuales van a ser los crear, actualizar y eliminar formatos. Luego de que estos formatos sean creados con la herramienta IReport el caso de uso se encargara de actualizar la lista de formatos disponibles para que pueda ser usada por el CU-1 Crear Reporte.

Figura 5. Casos Usos del Sistema.

3.4.1 Descripción detallada de los Casos de Uso del Sistema

	Crear Reporte	
Propósito	Permite que se pueda crear un reporte de los datos que se manejan en el sistema y se encuentran almacenados en la Base de Datos.	
Actores:	Usuario(Jefe de Proyecto, Analista de Proyecto y Usuarios de Docencia)	
Resumen:	El caso de uso comienza cuando un actor decide crear un nuevo reporte de los datos que se manejan en el proyecto. El actor puede crear un nuevo reporte (matriz, artefacto, administración y líneas base del proyecto). Para esto hace uso de los formatos de reportes (.XML) que tiene que haber sido creados anteriormente o usar los que propone por defecto la herramienta, terminando así es caso de uso.	
Referencias	R1, R2, R3, R4 (y las secciones que contiene).	
Precondiciones	1. Que se encuentre autenticado el actor.	
Postcondiciones		
Sección "Crear Reporte"		
Acción del actor		Respuesta del sistema

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL MÓDULO REPORTES

1. El actor accede al sistema de reportes.	2. El sistema le muestra los tipos de reportes disponibles y la opción de crear un nuevo reporte.
3. El actor escoge la opción correspondiente y oprime el botón crear reporte.	4. El sistema muestra un formato (.XML) con el nuevo esquema del reporte a crear, dando la opción de utilizar los reportes creado con el iReport para obtener datos específicos de la base de datos solo marcando atributos deseados para el reporte sin la necesidad de crear códigos XML o SQL para acceder a los datos de la base datos, además muestra los reportes correspondientes a todas la matrices, a las líneas base del proyecto artefactos y al sistema de administración.
5. El actor introduce los datos correspondientes en el esquema de reportes o selecciona los deseados y oprime el botón ok.	6. El sistema muestra el directorio para que el usuario escoja la carpeta específica donde desea guardar el reporte que ya ha sido convertido a HTML-PDF.
7. El actor escoge el directorio y presiona el botón guardar.	8. El sistema envía un mensaje informativo de que se realizó la acción.
Flujo Alternativo	
Acción 4	Se emite un mensaje para que no deje todos los campos en blanco.
Acción 6	Se emite un mensaje para que llene los campos obligatorios.
Acción 6	Si el usuario no desea guardar el reporte también muestra la opción de imprimir el reporte.
Acción 7	El usuario escoge la opción de imprimir el reporte.
Prioridad	Crítica

Figura 6. Descripción detallada (CU Crear Reporte).

	Gestionar Formato Reportes
Propósito	Este Gestionar incluye en sus secciones a Crear, Actualizar y Eliminar Formato de Reportes.
Actores:	Usuario(Jefe de Proyecto, Analista de Proyecto y Usuarios de Docencia)
Resumen:	El caso de uso comienza cuando un actor decide crear un formato reporte. El actor crear

formatos para (matriz, artefacto, administración, líneas base del proyecto). Para esto hace uso de sistemas externos como iReport en conjunto con el JasperReports donde el sistema le tiene que mostrar al usuario todos los atributos disponibles a utilizar para generar el reporte sin necesidad que el usuario tenga que escribir sentencias SQL o XML para coger los datos de la base de datos terminando así es caso de uso.	
Referencias	R5 (y las secciones que contiene)
Precondiciones	1. Que se encuentre autenticado el actor.
Postcondiciones	
Sección "Crear Formato Reporte"	
Acción del actor	Respuesta del sistema
1. El actor accede al sistema de reportes.	2. El sistema le muestra los tipos de reportes disponibles y la opción de crear un nuevo formato de reporte.
3. El actor escoge la opción que desea crear formato de reporte.	4. El sistema muestra la herramienta para crear los formatos de reportes IReport.
5. El actor el actor crea el formato y lo guarda.	6. El sistema muestra el mensaje de confirmación.
	7. La herramienta muestra el nuevo formato de reporte y da la opción de seleccionar cuales son los atributos que tendrá ese reporte.
8. El usuario selecciona los atributos deseados y le da la opción guardar.	9. El sistema muestra un mensaje de confirmación.
Flujo Alternativo	
Acción 6	Se emite un mensaje de error en caso que el formato haya sido mal diseñado.
Prioridad	Crítica

Figura 7. Descripción detallada (CU Gestionar Formatos Reportes sección Crear).

	Gestionar Formato Reportes
Propósito	Este Gestionar incluye en sus secciones a Crear, Actualizar y Eliminar Formato de Reportes.
Actores:	Usuario(Jefe de Proyecto, Analista de Proyecto y Usuarios de Docencia)
Resumen:	El caso de uso comienza cuando un actor decide crear un formato reporte. El actor

crear formatos para (matriz, artefacto, administración, líneas base del proyecto). Para esto hace uso de sistemas externos como iReport en conjunto con el JasperReports donde el sistema le tiene que mostrar al usuario todos los atributos disponibles a utilizar para generar el reporte sin necesidad que el usuario tenga que escribir sentencias SQL o XML para coger los datos de la base de datos terminando así es caso de uso.

Referencias	R6 (y las secciones que contiene)
Precondiciones	1. Que se encuentre autenticado el actor.
Postcondiciones	
Sección "Actualizar Formato Reporte"	
Acción del actor	Respuesta del sistema
1. El actor accede al sistema de reportes.	2. El sistema le muestra los tipos de reportes disponibles y la opción de actualizar un formato de reporte.
3. El actor escoge la opción que desea actualizar formato de reporte.	4. El sistema muestra la herramienta para actualizar los formatos de reportes IReport.
5. El actor el actor actualiza el formato y lo guarda.	6. El sistema muestra el mensaje de confirmación.
	7. La herramienta muestra el nuevo formato de reporte y da la opción de seleccionar cuales son los atributos del reporte.
8. El usuario selecciona que atributos desea quitar, o poner al reporte y le da la opción guardar.	9. El sistema muestra un mensaje de confirmación.
Flujo Alternativo	
Acción 6	Se emite un mensaje de error en caso que el formato haya sido mal diseñado.
Acción 9	Emite un mensaje de error en caso de que los atributos no sean los correctos.
Prioridad	Crítica

Figura 8. Descripción detallada (CU Gestionar Formatos Reportes sección Actualizar).

	Gestionar Formato Reportes
Propósito	Este Gestionar incluye en sus secciones a Crear, Actualizar y Eliminar

	Formato de Reportes.	
Actores: Usuario(Jefe de Proyecto, Analista de Proyecto y Usuarios de Docencia)		
Resumen: El caso de uso comienza cuando un actor decide crear un formato reporte. El actor crear formatos para (matriz, artefacto, administración, líneas base del proyecto). Para esto hace uso de sistemas externos como iReport en conjunto con el JasperReports donde el sistema le tiene que mostrar al usuario todos los atributos disponibles a utilizar para generar el reporte sin necesidad que el usuario tenga que escribir sentencias SQL o XML para coger los datos de la base de datos terminando así es caso de uso.		
Referencias	R6 (y las secciones que contiene)	
Precondiciones	1. Que se encuentre autenticado el actor.	
Postcondiciones		
Sección "Eliminar Formato Reporte"		
Acción del actor	Respuesta del sistema	
1. El actor accede al sistema de reportes.	2. El sistema le muestra los tipos de reportes disponibles y la opción de eliminar un formato de reporte.	
3. El actor escoge el formato de reporte que desea eliminar.	4. El sistema muestra un mensaje de confirmación que se eliminó.	
Flujo Alternativo		
Acción 6	Se emite un mensaje de error en caso que el formato no haya sido eliminado y cual fue el problema.	
Prioridad	Media	

Figura 9. Descripción detallada (CU Gestionar Formatos Reportes sección Eliminar).

3.5 Análisis y diseño del sistema

Para comenzar a dar cumplimiento a la fase de elaboración del sistema automatizado para la realización de reportes, en este capítulo se realizaran los diagramas de análisis y de clases del diseño para dar respuesta a la solución que se propone. Se le da seguimiento y realización a los casos de usos del sistema para dar cumplimiento a los requerimientos funcionales. Se tendrá en cuenta los principios de diseño durante el desarrollo de la aplicación.

3.5.1 Análisis del sistema.

La parte de análisis de sistemas es uno de los flujos de trabajo que le da cumplimiento a la fase de elaboración. Aquí se verán como las funcionalidades del sistema pueden ser más exactas.

Tipos de clases:

CI_<Nombre de la clase>: estas clases modelan la interacción entre los usuarios y el sistema, es decir, ventanas, formularios, dispositivos, sistemas externos, etc. Ejemplo CI_Reporte de Matrices.

CC_<Nombre de la clase>: estas clases encapsulan el comportamiento de cada caso de uso y coordinan el trabajo de las clases interfaz y entidad. Ejemplo CC_Realizar Reporte.

CE_<Nombre de la clase>: estas clases modelan toda la información del sistema que posee una vida larga y que puede ser persistente. Ejemplo CE_Reporte de Matrices.

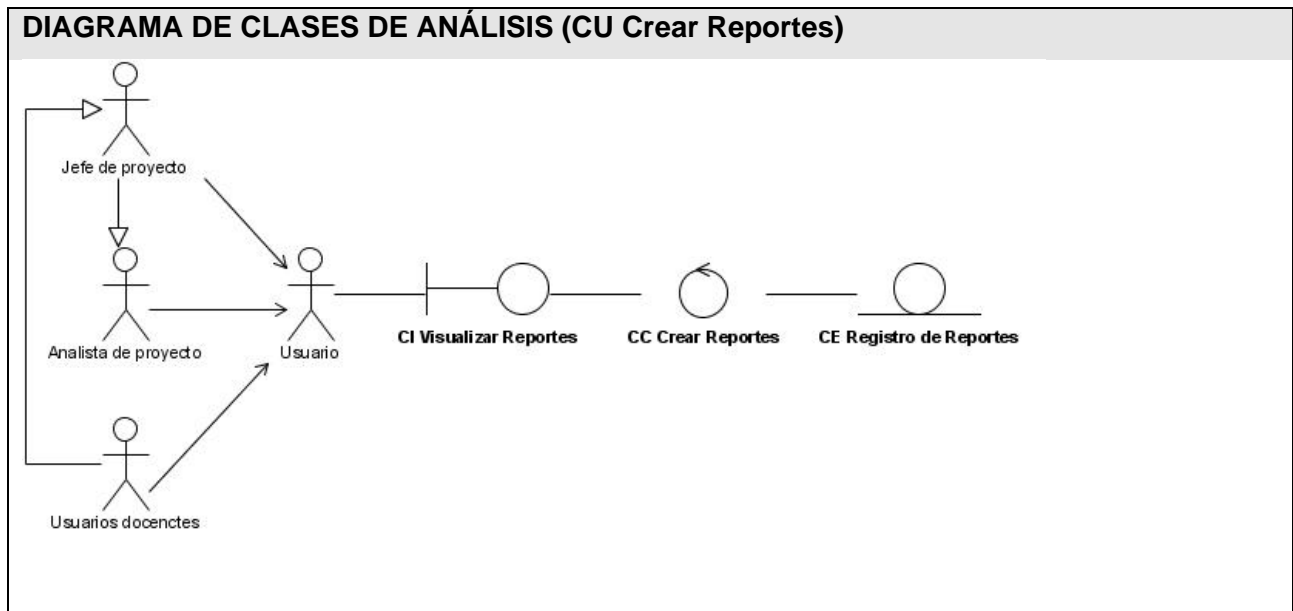


Figura 10: Diagrama de Análisis (CU Crear Reportes).

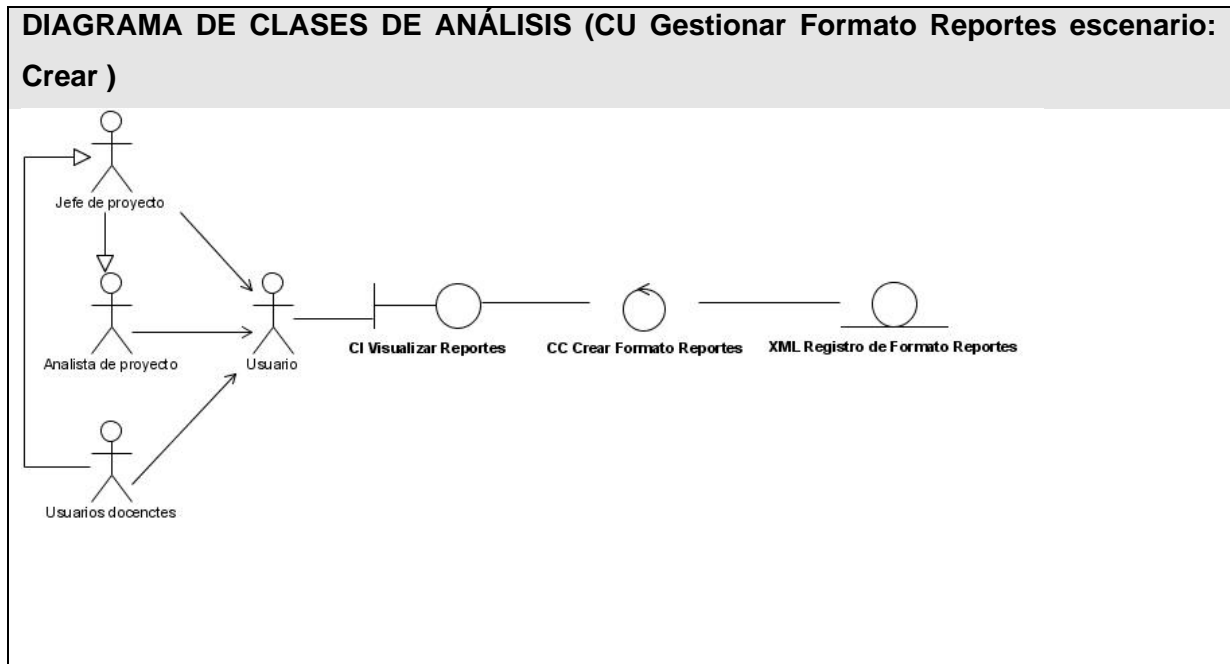


Figura 11: Diagrama de Análisis (CU Gestionar Formato Reportes escenario: Crear).

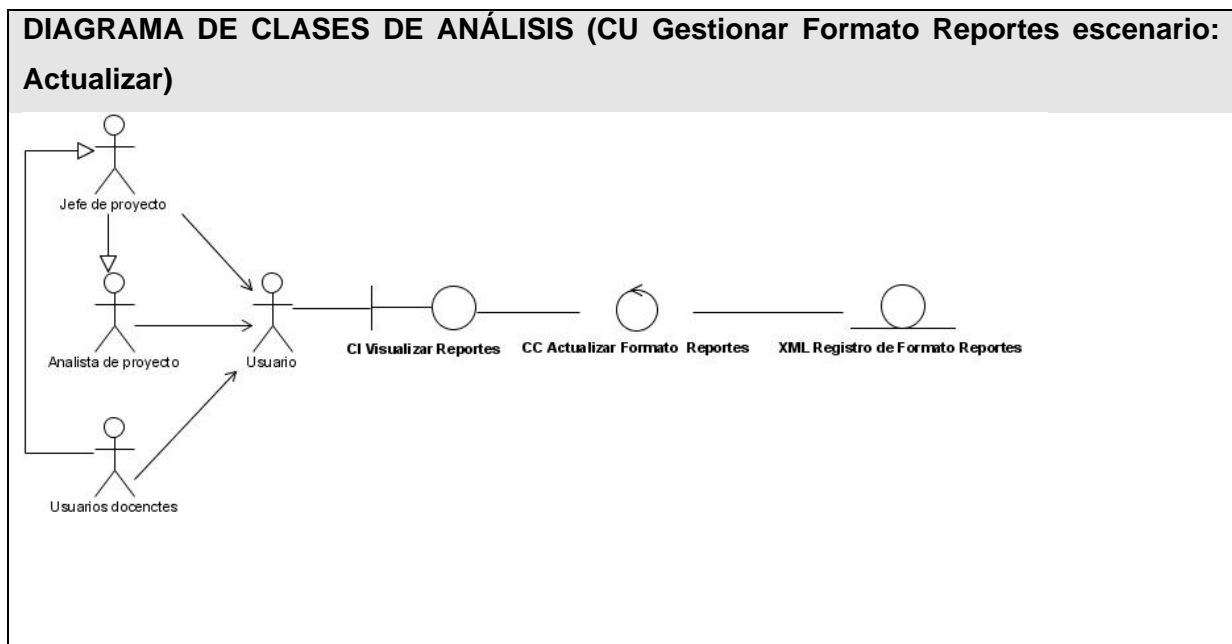


Figura 12: Diagrama de Análisis (CU Gestionar Formato Reportes escenario: Actualizar).

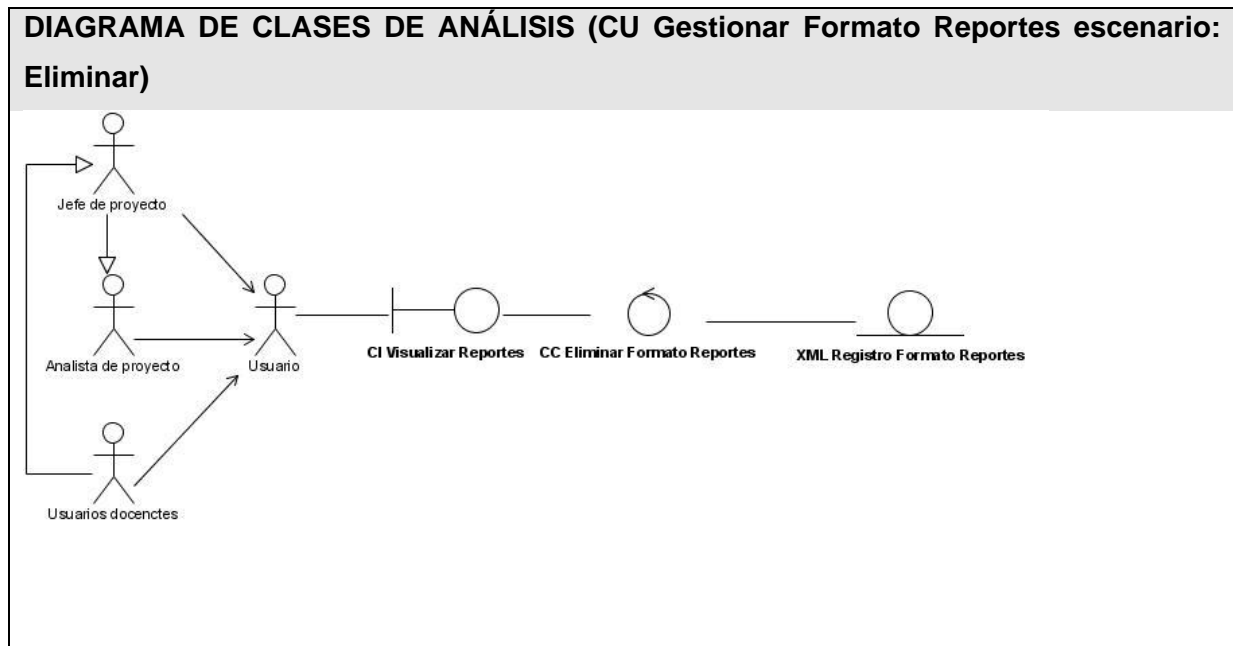


Figura 13: Diagrama de Análisis (CU Gestionar Formato Reportes escenario: Eliminar).

3.5.2 Diagramas de clases del diseño del sistema.

El diseño del sistema es otro flujo de trabajo que se realiza al final de la fase de elaboración. Este permite mediante el modelo de diseño describir la realización física de los casos de uso centrándose en el impacto que tienen en el sistema los requisitos funcionales, no funcionales y otras restricciones relacionadas con el entorno de implementación. (JACOBSON, 2004) A continuación se describen algunas de las principales clases que serán utilizadas durante la implementación del sistema para el Reporte de la información existente en el proyecto.

➤ Clases de la capa Presentación

FU_<Nombre de la Forma de usuario>: Son las formas que van a funcionar como interfaz a los usuarios. Se construirán dinámicamente para ser visualizadas en las ventanas de los usuarios.

FR_<Nombre del formulario>: Son los formularios que se utilizan para obtener los datos introducidos por el usuario en cada una de las actividades que se realiza durante el procesamiento de un documento.

FP_<Nombre Forma principal>: Son las clases que construyen las formas que funcionarán como interfaz de usuarios y tienen la lógica de la presentación.

➤ Clases de la capa Lógica del Negocio

<<Façade Interface>>: Es la clase que permite separar físicamente las capas de presentación y lógica de la aplicación. Aquí se hace real el uso del patrón fachada.

CC_<Nombre de la clase>: Son las clases controladora que brindan los servicios necesarios a la clase interfaz. Contienen la lógica de la aplicación, en forma de transacciones de negocio. Utiliza una interfaz de fachada (Facade Interface) donde se exponen todos los servicios necesarios para la capa de presentación referidos a la lógica, lo que permite separar físicamente ambas capas.

CCM_<Nombre de la clase>: Son las clases que contienen las reglas del negocio de la aplicación relacionadas con una entidad del negocio, controlan y brindan los servicios a las clases CC_ por que son las encargadas de interactuar con las entidades y con la capa de persistencia de datos.

CE_<Nombre de la clase>: Son las entidades del negocio, clases que contienen todos los datos, no tienen lógica de negocio. Contienen colecciones de atributos en forma de XML.

➤ Clases de las capas Persistencia y Fuente de Datos

No es necesario describir las clases de las capas de Persistencia y fuente de Datos, ya que estas no serán modificadas solo son necesarias para acceder a la información que estas contienen. Serán representadas en un paquete de Acceso a Datos.

DIAGRAMA DE CLASES DEL DISEÑO (CU Crear Reportes: Escenario Crear Reportes Matrices, Artefactos, Líneas Bases y Administración)

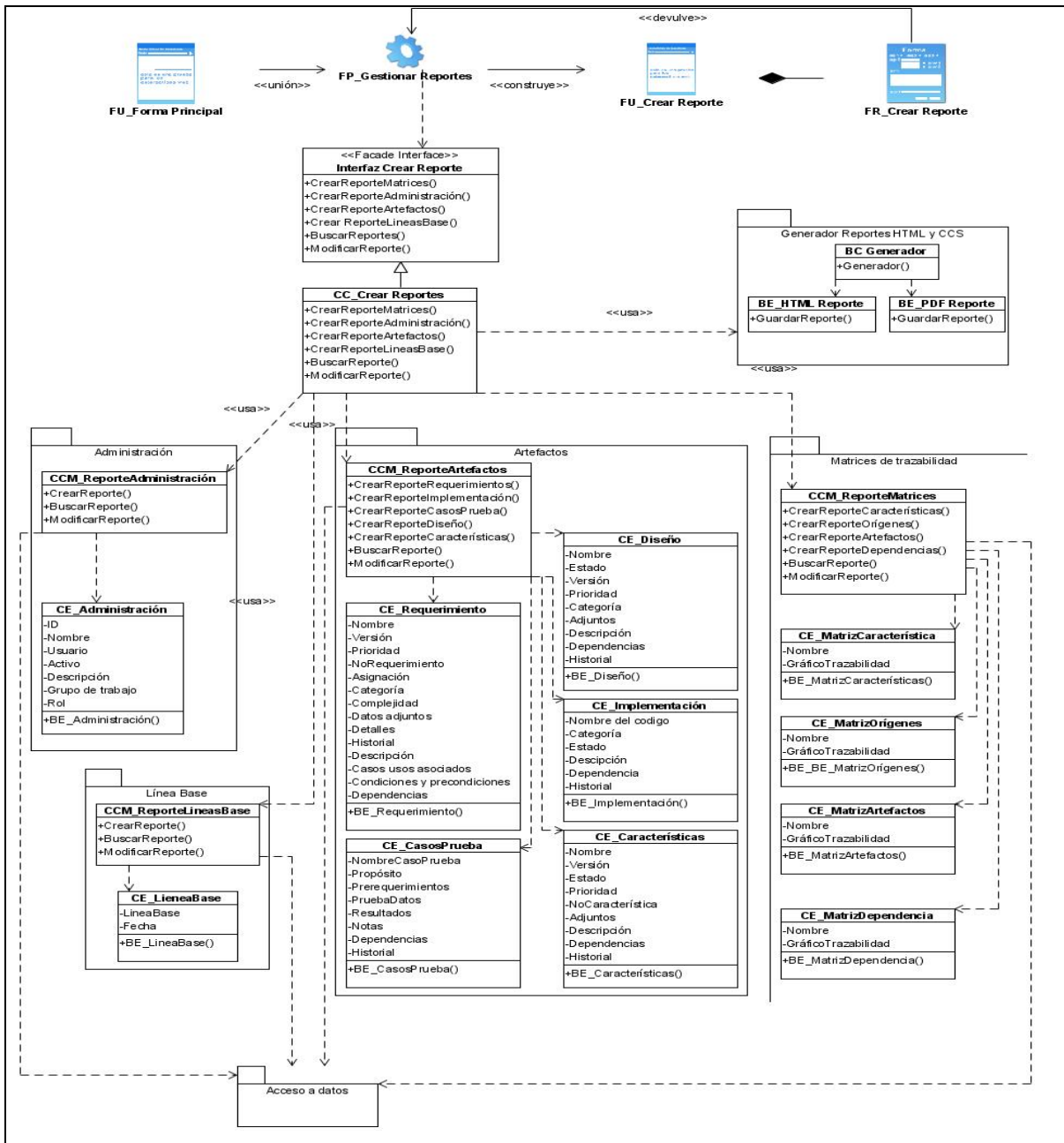


Figura 14: Diagrama de Diseño (CU Crear Reportes: Escenario Crear Reportes Matrices, Artefactos, Líneas Bases y Administración).

DIAGRAMA DE CLASES DEL DISEÑO (CU Gestionar Formato Reportes: Escenario Crear Reportes Matrices, Artefactos, Líneas Bases y Administración)

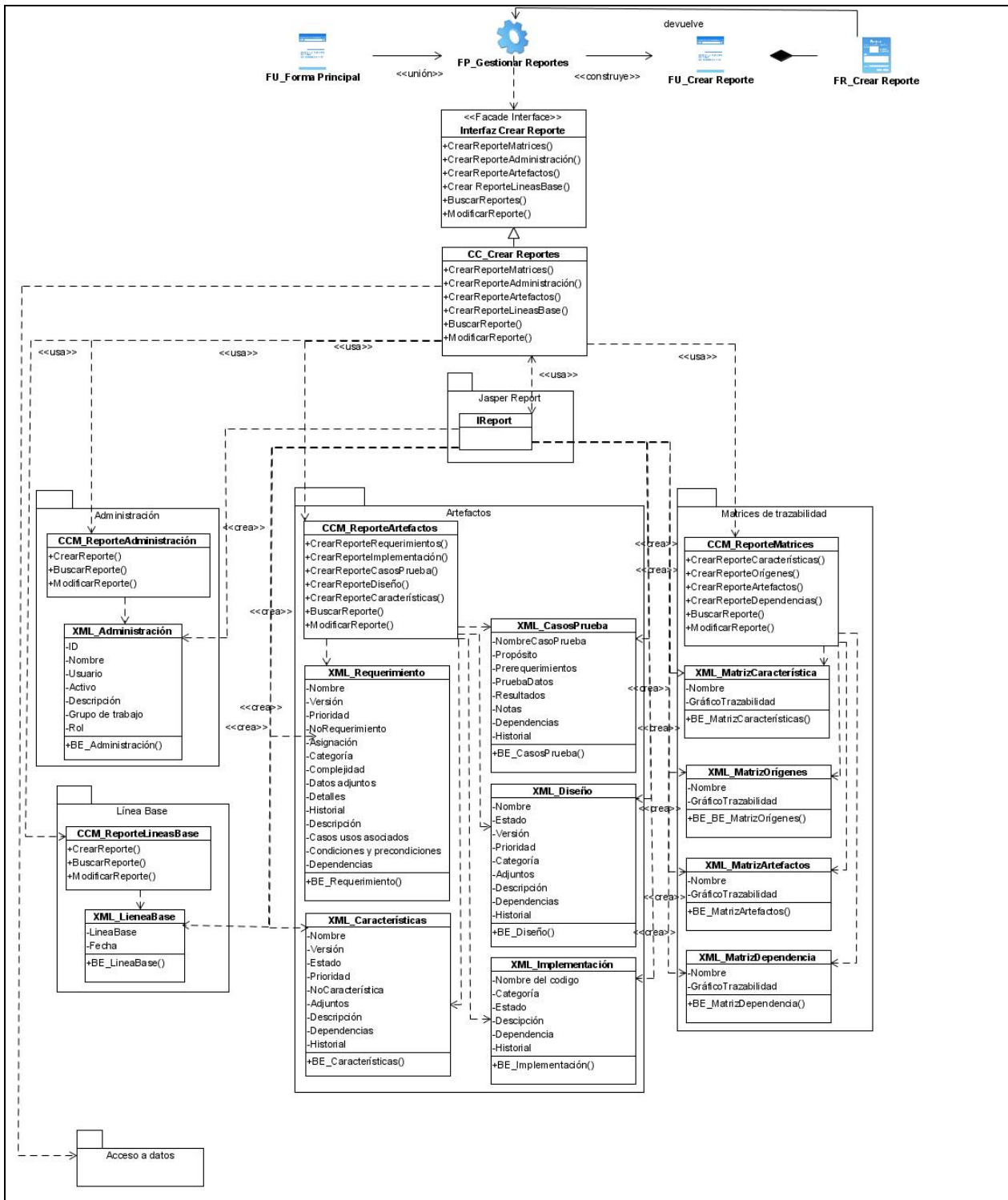


Figura 15: Diagrama de Diseño (CU Gestionar Formato Reportes: Escenario Crear Reportes Matrices, Artefactos, Líneas Bases y Administración).

DIAGRAMA DE CLASES DEL DISEÑO (CU Gestionar Formato Reportes: Escenario Actualizar Reportes Matrices, Artefactos, Líneas Bases y Administración)

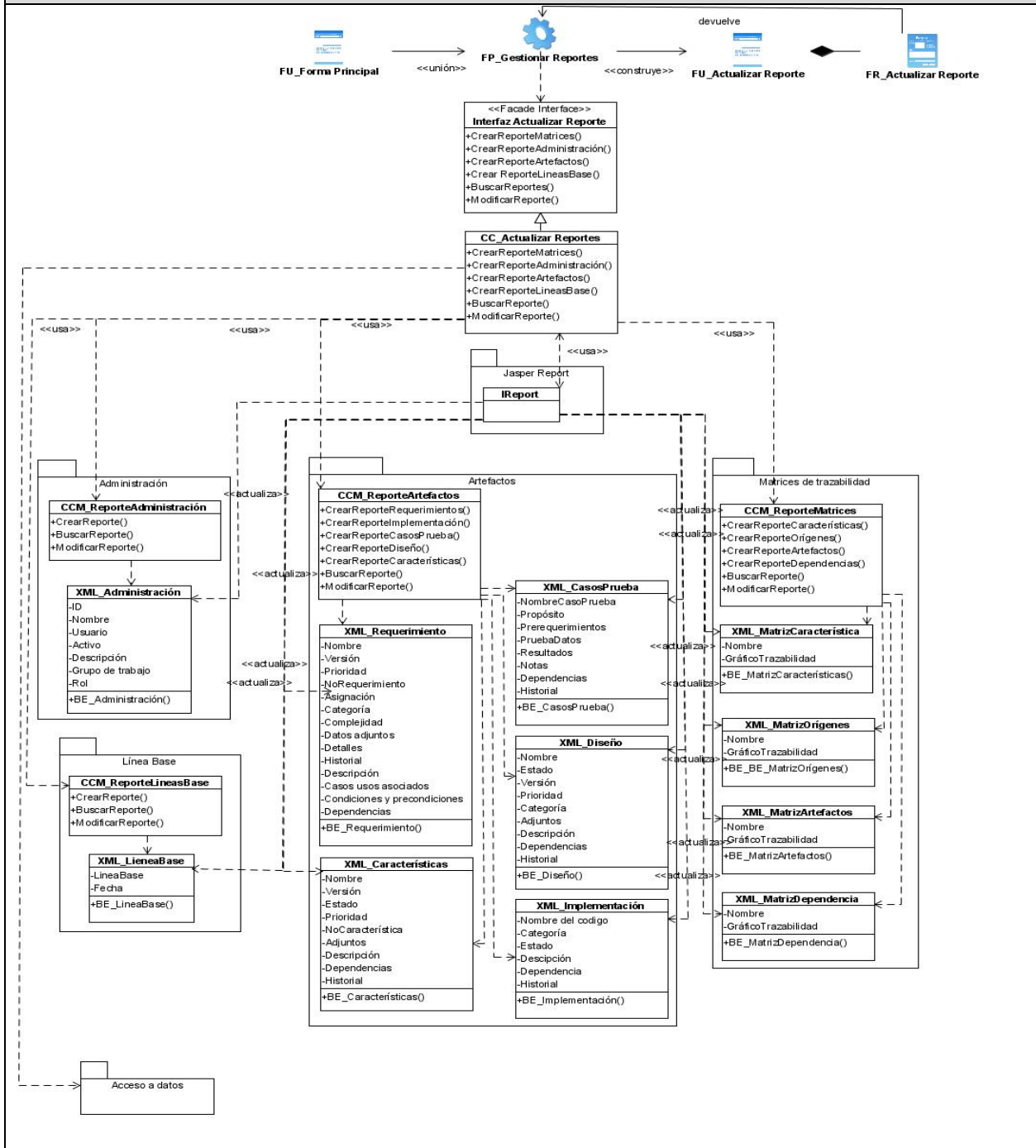


Figura 16: Diagrama de Diseño (CU Gestionar Formato Reportes: Escenario Actualizar Reportes Matrices, Artefactos, Líneas Bases y Administración).

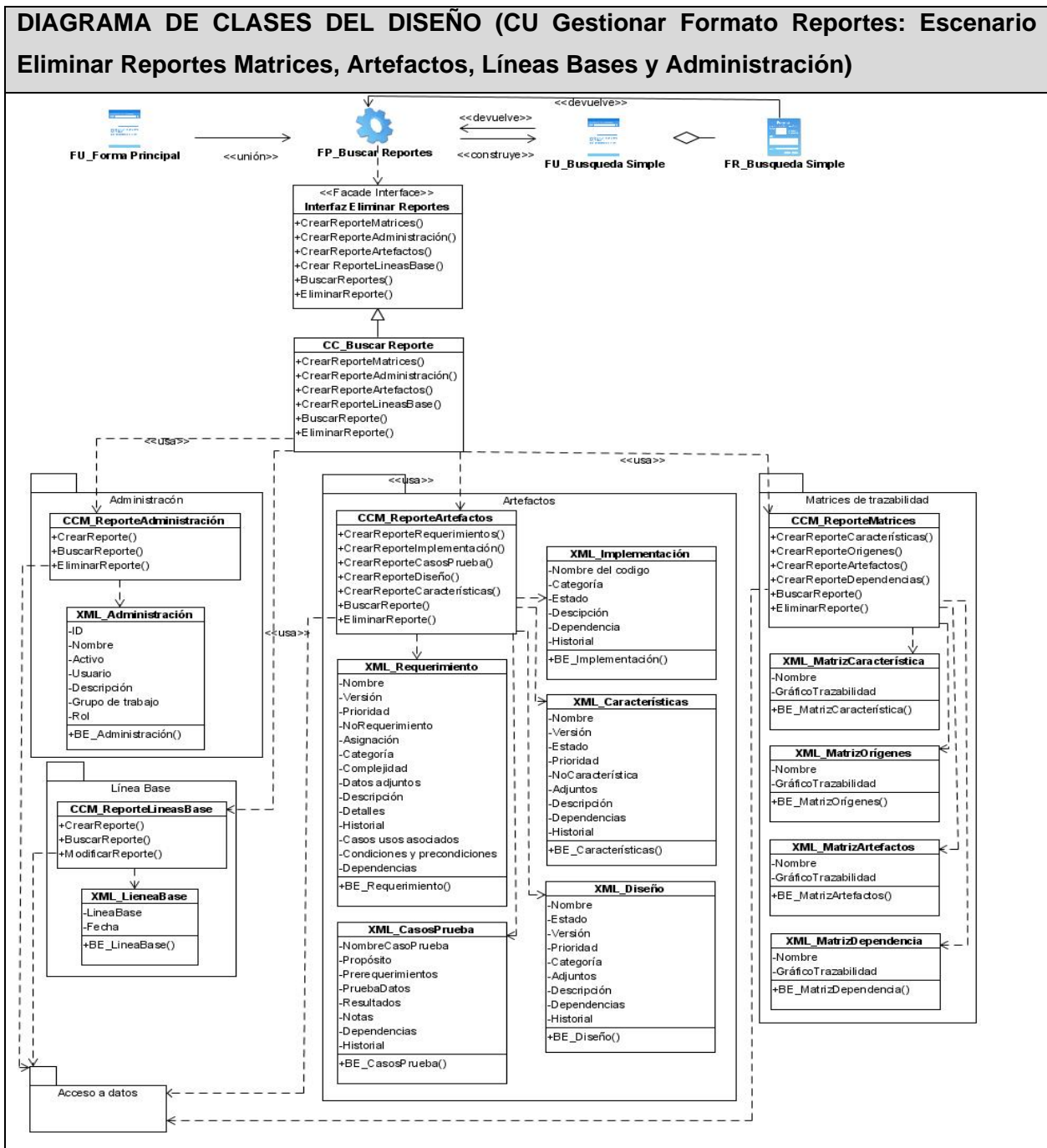


Figura 17: Diagrama de Diseño (CU Gestionar Formato Reportes: Escenario Eliminar Reportes Matrices, Artefactos, Líneas Bases y Administración).

3.6 Diagramas de interacción.

Los diagramas de secuencia muestran las interacciones entre los objetos que estarán presentes en la aplicación, ordenadas en secuencia temporal durante un escenario concreto. Para mayor entendimiento los casos de usos con más de un flujo se representaron en diagramas diferentes.

3.6.1 Métodos que van a unir el paquete de reportes con los demás paquetes.

El código de la aplicación viene integrado con el paquete base de instalación y la clase que se encuentra dentro de este paquete específicamente en:

`src/com/com/osrmt/appclient/reqmanager/RequirementManagerController.java` es la encargada de inicializar la aplicación con todos sus componentes.

Dentro de las acciones del menú se establecen los parámetros que se utilizarán para las acciones de cada submenú y dentro de estas acciones se encuentra la funcionalidad reportes que a continuación se muestra:

```
menuManager.addItem("Menu Tools - Reports", actions.getAction  
(ActionGroup.REQMGRTOOLSREPORTS), false, Keys.getCtrl_R());
```

Con este parámetro `REQMGRTOOLSREPORTS` y otros, se utilizarán para crear la acción del menú.

El método que específicamente tiene que llamar al paquete de reporte que se propone es: `public ApplicationAction getToolsReports()` que se encuentra dentro de la clase `RequirementManagerActions.java` dentro de la misma carpeta `reqmanager`. Dentro del método se debe llamar al wizard que será el encargado de hacer todas las funciones de reportes.

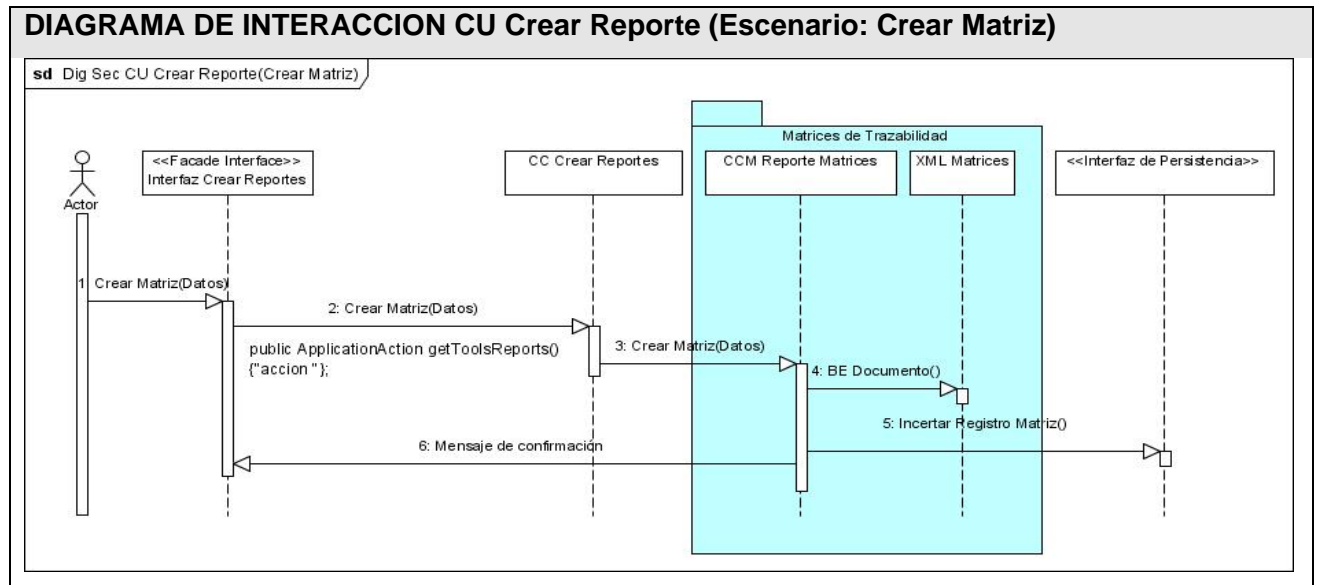


Figura 18: DIAGRAMA DE INTERACCION CU Crear Reporte (Escenario: Crear Matriz).

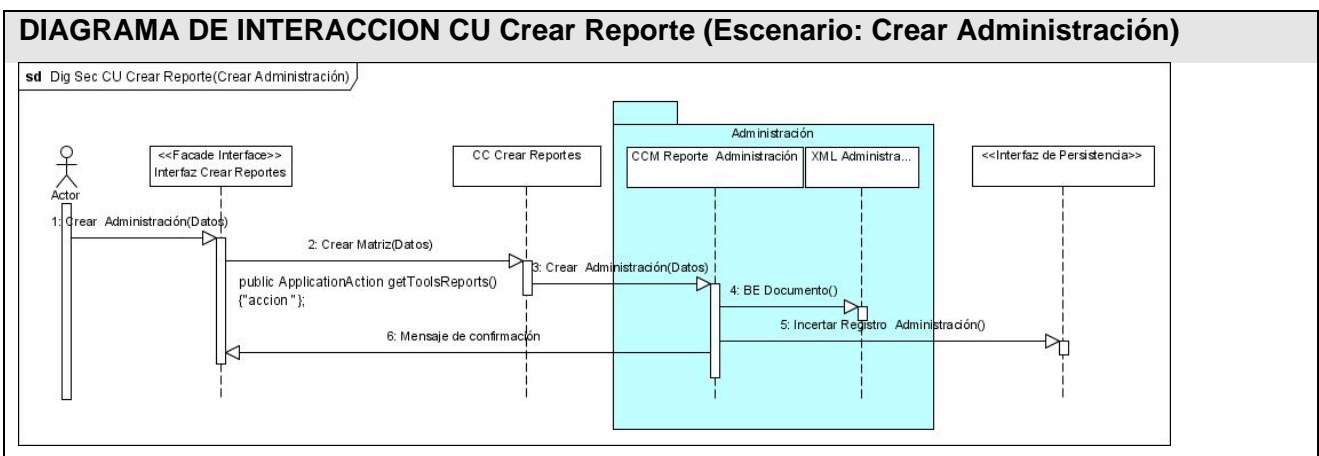


Figura 19: DIAGRAMA DE INTERACCION CU Crear Reporte (Escenario: Crear Administración).

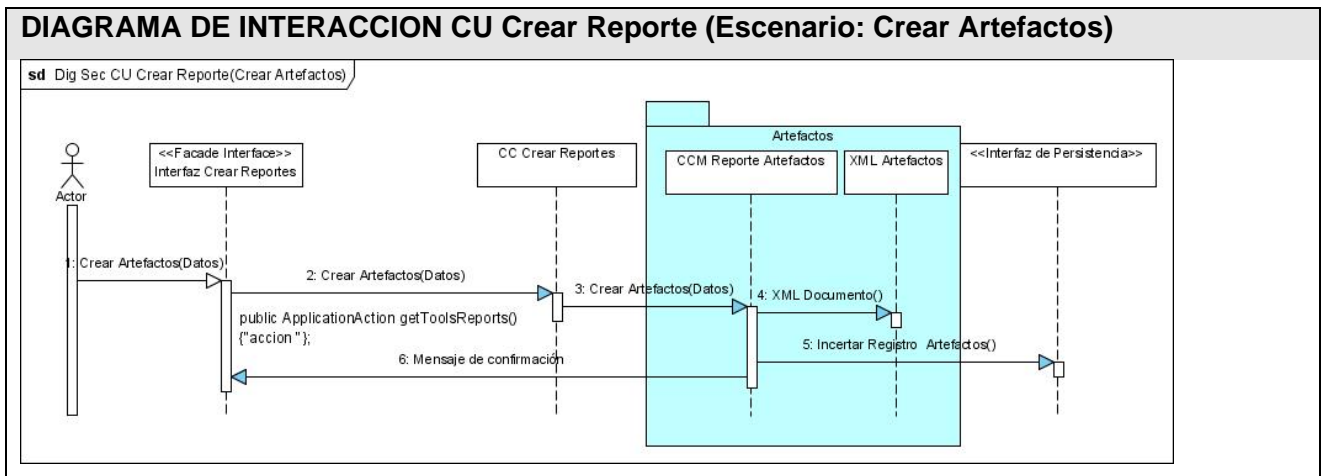


Figura 20: DIAGRAMA DE INTERACCION CU Crear Reporte (Escenario: Crear Artefactos).

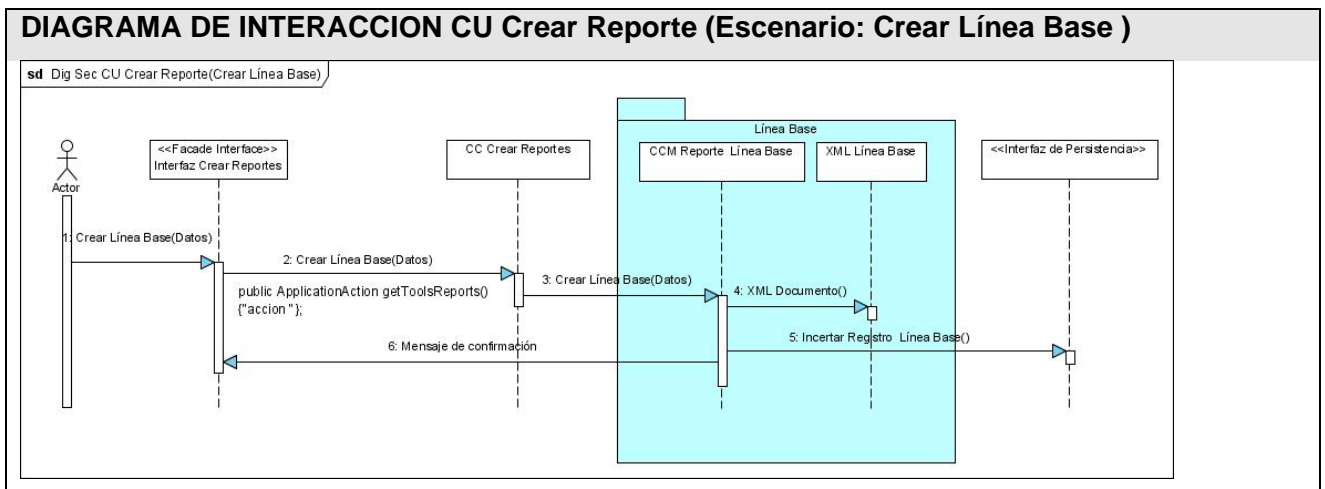


Figura 21: DIAGRAMA DE INTERACCION CU Crear Reporte (Escenario: Crear Línea Base).

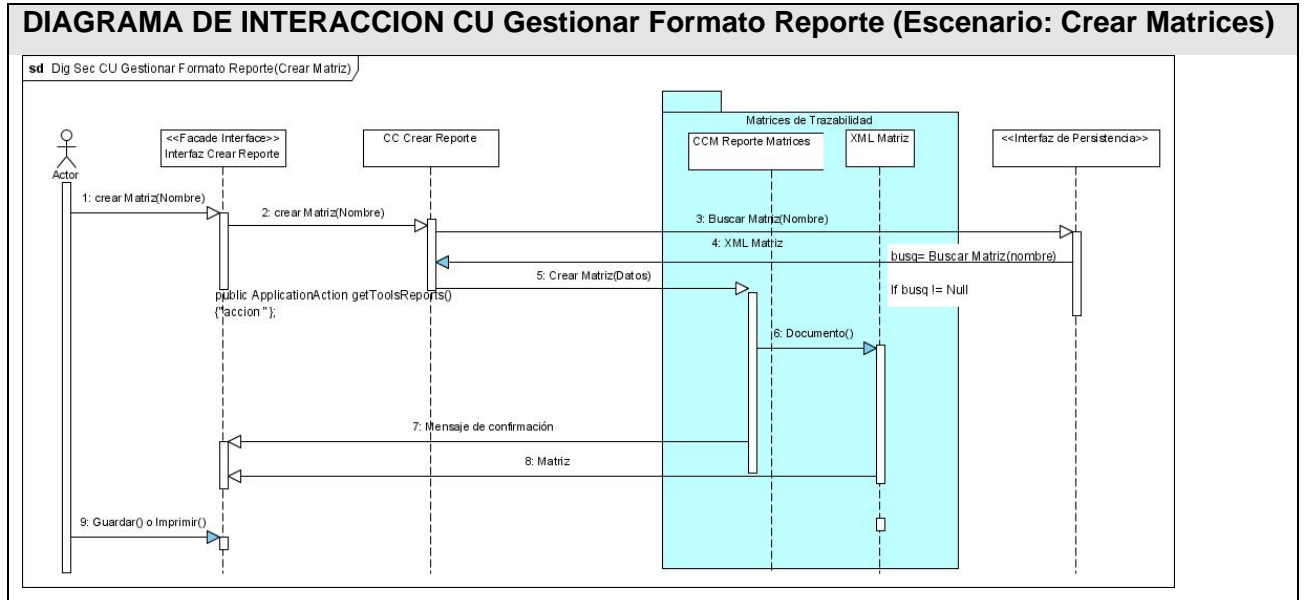


Figura 22: DIAGRAMA DE INTERACCION CU Formato Reporte (Escenario: Crear Matrices).

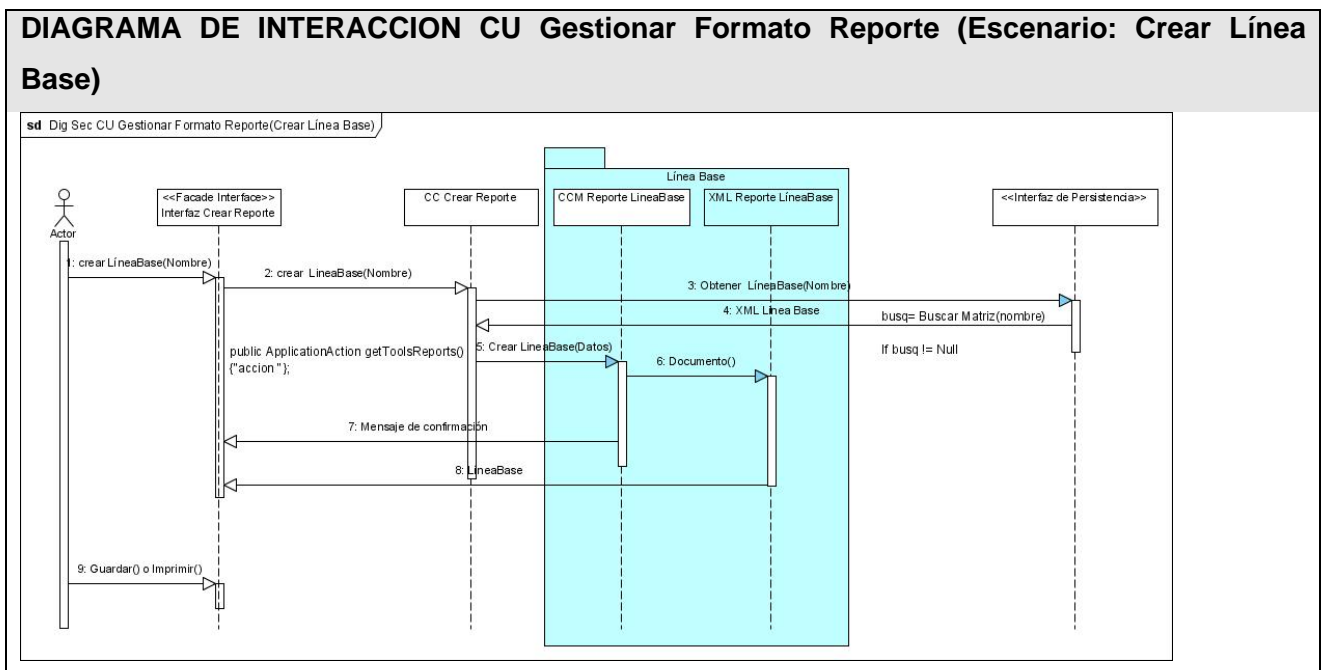


Figura 23: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Crear Línea Base).

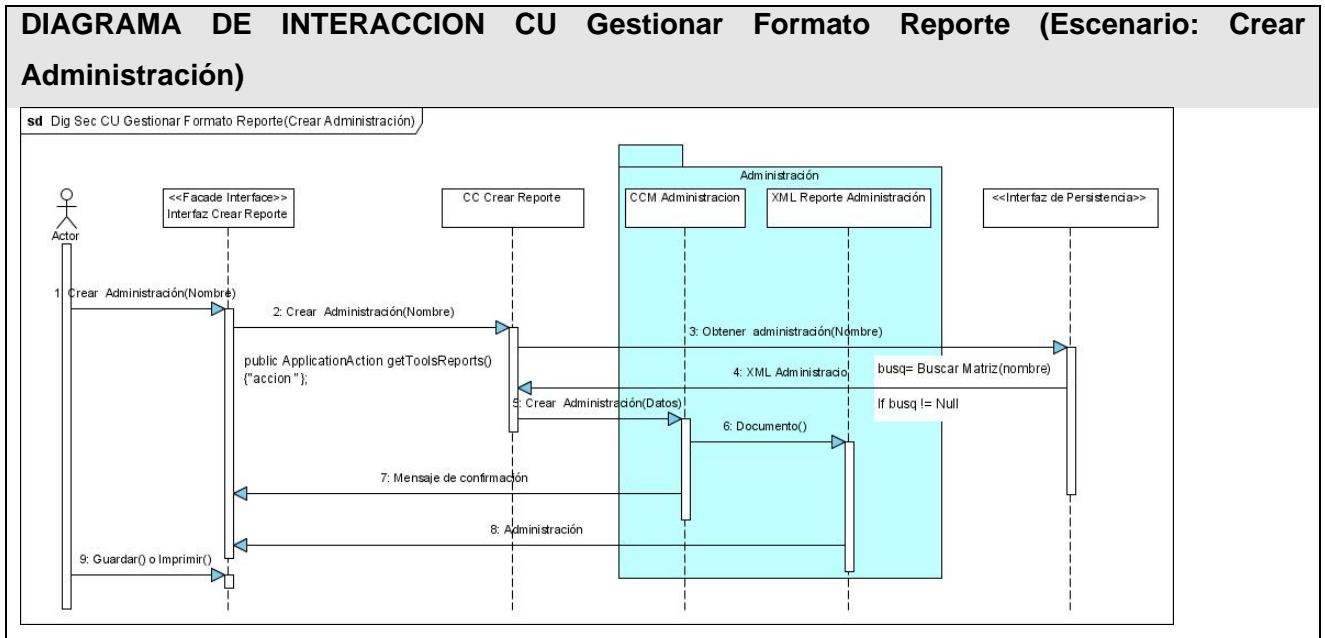


Figura 24: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Crear Administración).

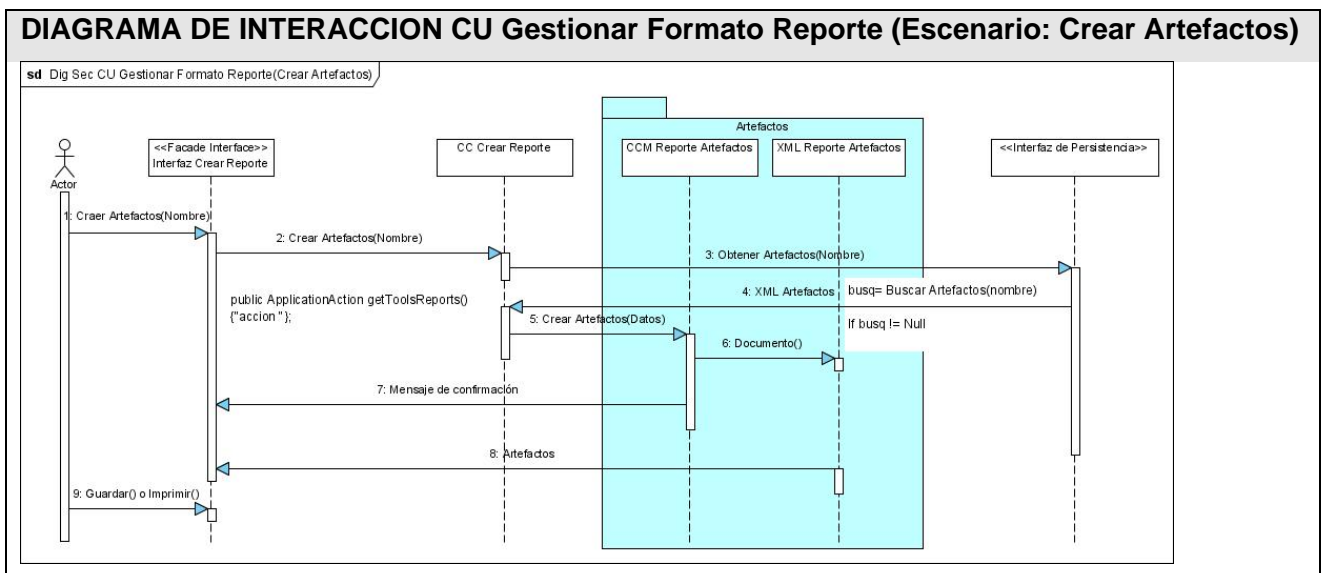


Figura 25: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Crear Artefactos).

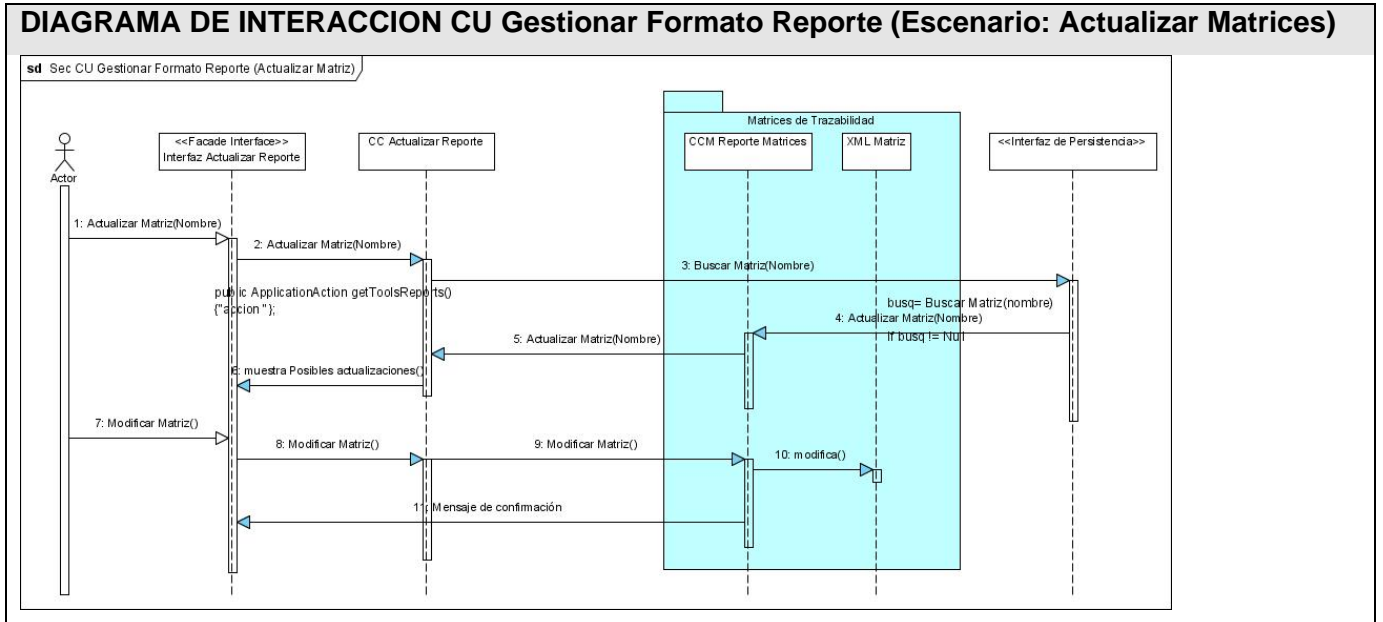


Figura 26: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Actualizar Matrices).

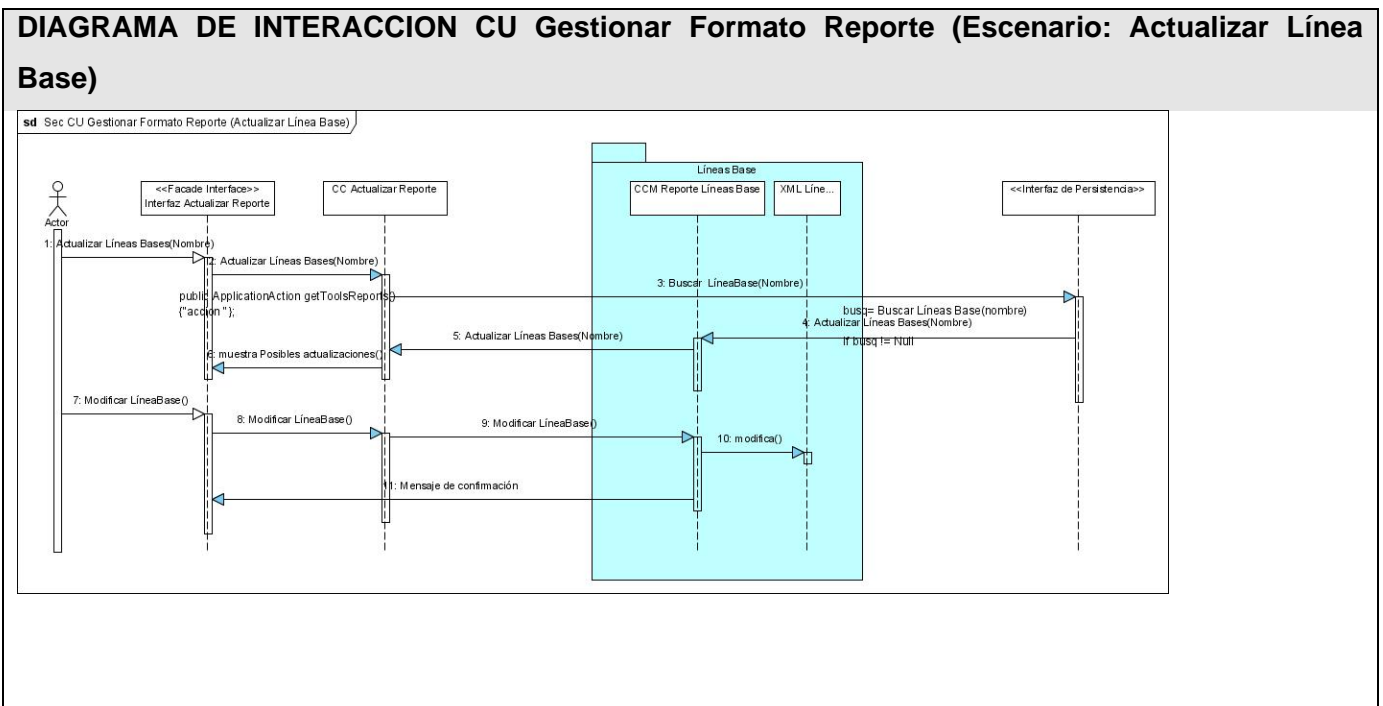


Figura 27: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Actualizar Línea Base).

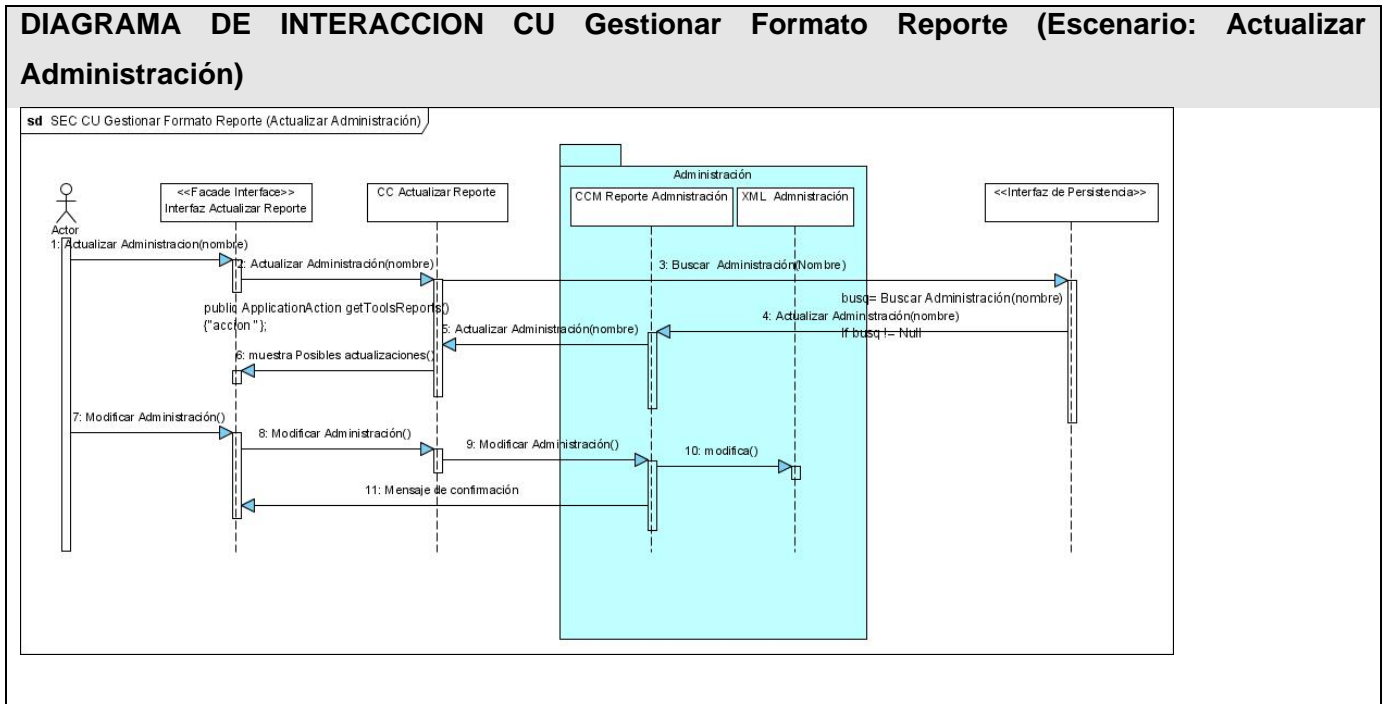


Figura 28: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Actualizar Administración).

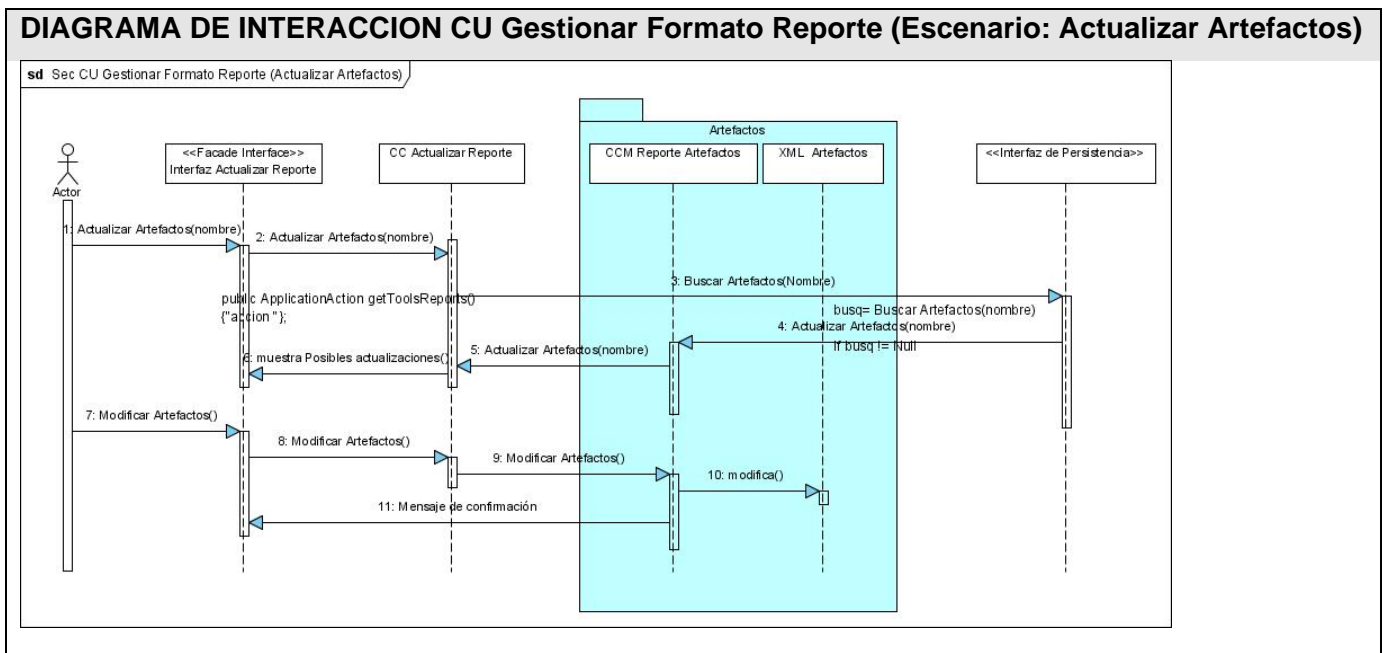


Figura 29: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Actualizar Artefactos).

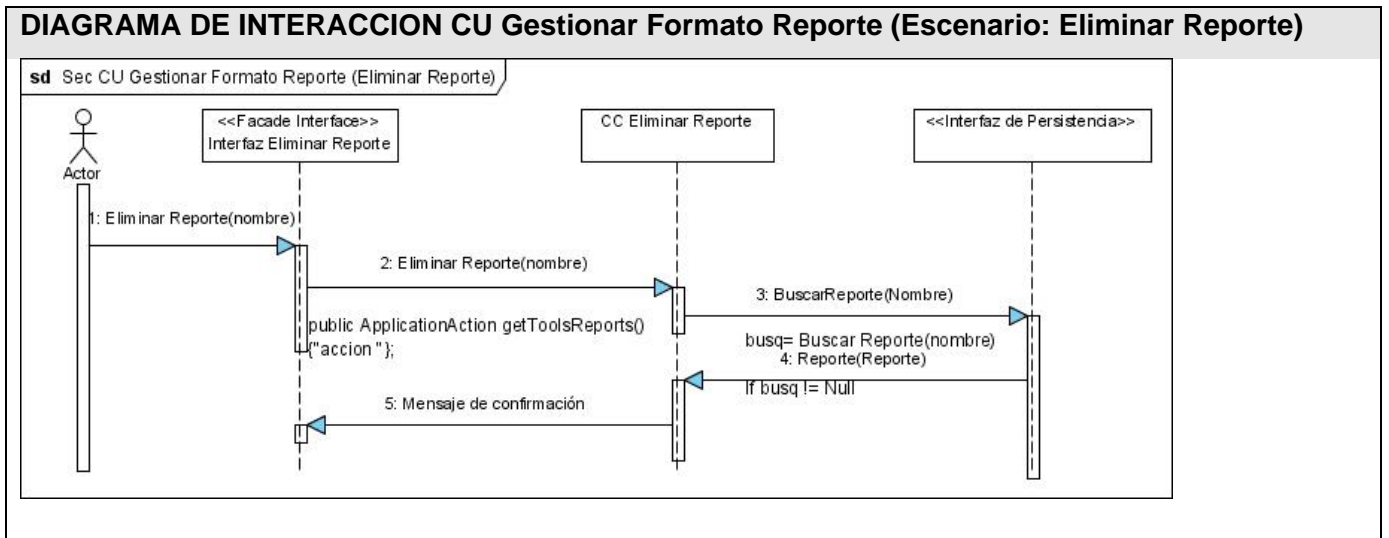


Figura 30: DIAGRAMA DE INTERACCION CU Gestionar Formato Reporte (Escenario: Eliminar Reporte).

3.7 Descripción de las clases.

Nombre: CC_Gestionar Reportes	
Tipo de Clase: Clase de servicio o clase controladora (capa de servicio)	
Para cada responsabilidad:	
Nombre:	Crear Formato Reporte Administración.
Descripción:	Esta es la clase encargada de llamar al la herramienta IReport que es la encargada de gestionar los formatos de reportes de Administración. Luego de esto llevar el archivo a la carpeta Reportes correspondientes para que puedan ser usados en Crear Reportes.
Nombre:	Crear Formato Reporte Línea Base.
Descripción:	Esta es la clase encargada de llamar al la herramienta IReport que es la encargada de gestionar los formatos de reportes de Línea Base. Luego de esto llevar el archivo a la carpeta Reportes correspondientes para que puedan ser

	usados en Crear Reportes.
Nombre:	Crear Formato Reporte Artefactos.
Descripción:	Esta es la clase encargada de llamar al la herramienta IReport que es la encargada de gestionar los formatos de reportes de Artefactos. Luego de esto llevar el archivo a la carpeta Reportes correspondientes para que puedan ser usados en Crear Reportes.
Nombre:	Crear Formato Reporte Matrices.
Descripción:	Esta es la clase encargada de llamar al la herramienta IReport que es la encargada de gestionar los formatos de reportes de Matrices. Luego de esto llevar el archivo a la carpeta Reportes correspondientes para que puedan ser usados en Crear Reportes.
Nombre:	Actualizar Formato Reportes Administración.
Descripción:	Permite modificar los reportes de administración según los datos que se muestran y los que se mostraran dependiendo de los datos que hayan sido guardados en la base de datos.
Nombre:	Actualizar Formato Reportes Línea Base
Descripción:	Permite modificar los reportes de las líneas bases de trabajo según los datos que se muestran y los que se mostrarán dependiendo de los datos que hayan sido guardados en la base de datos.
Nombre:	Actualizar Formato Reportes Artefactos
Descripción:	Permite modificar los reportes de artefactos según los datos que se muestran y los que se mostrarán dependiendo de los datos que hayan sido guardados en la base de datos.
Nombre:	Actualizar Formato Reportes Matrices
Descripción:	Permite modificar los reportes de matrices según los datos que se muestran y los que se mostrarán dependiendo de los datos que hayan sido guardados en la base de datos.
Nombre:	Eliminar Formato Reportes
Descripción:	Es la clase encargada de eliminar un formato de reportes que ya no va ser utilizado.

Figura 31: Descripción de Clases (CC_Gestionar Reportes).

Nombre: CC_Crear Reportes	
Tipo de Clase: Clase de servicio o clase controladora (capa de servicio)	
Para cada responsabilidad:	
Nombre:	Crear Reporte Administración.
Descripción:	Permite crear un nuevo reporte de administración según los datos que se deseen tener en dicho reporte y la salida que se le dará.
Nombre:	Crear Reporte Línea Base.
Descripción:	Permite crear un nuevo reporte de línea base de trabajo según los datos que se deseen tener en dicho reporte y la salida que se le dará.
Nombre:	Crear Reporte Artefactos.
Descripción:	Permite crear un nuevo reporte de artefactos que se generen según los datos que se deseen tener en dicho reporte y la salida que se le dará.
Nombre:	Crear Reporte Matrices.
Descripción:	Permite crear un nuevo reporte de matrices según los datos que se deseen tener en dicho reporte y la salida que se le dará.

Figura 32: Descripción de Clases (CC_Crear Reportes).

Nombre: XML_Administración	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
ID	Int
Nombre	Varchar
Usuario	Varchar
Activo	Int
Descripción	Varchar
Grupo de trabajo rol	Varchar
Nombre:	XML_Administración()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento administración (objeto) con todos sus atributos.

Figura 33: Descripción de Clases (XML_Administración).

Nombre: XML_Línea Base	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Línea Base	varchar
Fecha	timestamp
Nombre:	XML_LíneaBase()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Línea Base (objeto) con todos sus atributos.

Figura 34: Descripción de Clases (XML_ Línea Base).

Nombre: XML_Artefactos Requerimientos	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Prioridad	int
Nombre	varchar
Versión	float
No Requerimiento	int
Asignación	varchar
Categoría	varchar
Complejidad	int
Descripción	varchar
Datos Adjuntos	varchar
Historial	varchar
Casos Usos Asociados	varchar
Condiciones y Precondiciones	varchar
Dependencias	int
Detalles	varchar
Nombre:	XML_Administración()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Artefactos (objeto) con todos sus atributos.

Figura 35: Descripción de Clases (XML_ Requerimientos).

Nombre: XML_Artefactos Casos Pruebas	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Propósito	varchar
Nombre	varchar
Prerequerimientos	varchar
PruebaDatos	varchar
Resultado	varchar
Nota	varchar
Dependencia	int
Historial	varchar
Nombre:	XML_CasosPrueba()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Casos Prueba (objeto) con todos sus atributos.

Figura 36: Descripción de Clases (XML_ Casos Prueba).

Nombre: XML_Artefactos Características	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Versión	float
Nombre	varchar
Estado	float
Prioridad	int
NoCaracterística	int
Adjunto	varchar
Descripción	varchar
Dependencia	int
Historial	varchar
Nombre:	XML_Características ()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Características (objeto) con todos sus atributos.

Figura 37: Descripción de Clases (XML_ Características).

Nombre: XML_Artefactos Implementación	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Nombre	varchar
Estado	float
Descripción	varchar
Categoría	int
Dependencia	int
Historial	varchar
Nombre:	XML_ Implementación ()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Implementación (objeto) con todos sus atributos.

Figura 38: Descripción de Clases (XML_ Implementación).

Nombre: XML_Artefactos Diseño	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Nombre	varchar
Estado	float
Prioridad	int
Adjunto	varchar
Versión	float
Descripción	varchar
Categoría	int
Dependencia	int
Historial	varchar
Nombre:	XML_ Diseño ()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Diseño (objeto) con todos sus atributos.

Figura 39: Descripción de Clases (XML_ Diseño).

Nombre: XML_Matriz	
Tipo de clase: Entidad del negocio o clase entidad (capa de dominio)	
Nombre	varchar
Gráfico Trazabilidad	img
Nombre:	XML_Matriz()
Descripción:	Es el constructor de la clase, permite crear un nuevo documento Matriz (objeto) con todos sus atributos.

Figura 40: Descripción de Clases (XML_Matriz).

3.8 Modelos de datos del sistema.

Principio de diseño.

Esta aplicación va dirigida a los analistas de proyecto y se considera que estos tengan un mínimo de conocimientos del tema, el diseño está centrado en lograr una interfaz clara, de fácil uso por parte de los usuarios. Además de que puede ser utilizada en la docencia por parte de estudiantes y profesores.

Patrones de Diseño.

Se usa un patrón de diseño con el objetivo de que el sistema sea reusable y de gran fortaleza, para proporcionar una interfaz unificada. El objetivo que se persigue es reducir la complejidad y minimizar las dependencias. Para el desarrollo del siguiente paquete utilizaremos el siguiente patrón:

Fachada (Facade): Proporcionar una interfaz unificada para un conjunto de interfaces en un subsistemas, haciéndolo más fácil de usar. Reduce la complejidad y minimiza las dependencias. Es muy intuitivo si se quiere proporcionar una interfaz sencilla para un subsistema complejo y si se quieren dividir los niveles la fachada sería el punto de entrada a esos niveles.

La funcionalidad principal del patrón fachada es delegar las peticiones de los clientes a los subsistemas. Los clientes hacen las peticiones a la fachada y esta le responde, esto permite que los clientes no accedan directamente a los objetos del sistema.

Los flujos de trabajo análisis y diseño son muy importantes en el proceso de desarrollo del sistema pues proporciona un dominio total del problema así como una visión amplia del tamaño y la

complejidad del mismo. Con la finalización de este capítulo se da a conocer cómo debe ser implementado el software. Se conoció cómo llevar a cabo las funcionalidades que requieren el sistema. Se puede concluir que esto puede utilizarse como una guía de fácil comprensión para los desarrolladores ya que con los flujos de trabajo realizados se puede entender de forma general como es el funcionamiento de lo que se debe desarrollar.

Capítulo 4: Análisis de factibilidad del sistema.

El objetivo fundamental, cuando utilizamos una herramienta para la Gestión de Requisitos es contribuir a que los productos queden con la calidad requerida, utilizando un proceso de gestión eficiente y obteniendo un de él un costo razonable, de ahí la importancia que tiene la etapa de planificación de proyectos durante el proceso de desarrollo del software.

La planificación de proyectos consiste en estimar los resultados del proyecto desde la primera fase de trabajo hasta la entrega del producto final al cliente y los valores de costo, tiempo de desarrollo y recursos requeridos. (JACOBSON, 2004)

4.1 Planificación del proyecto basada en el análisis de puntos de casos de uso

Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso. El método de Análisis de Puntos de Casos de Uso es un método de estimación y cálculo del tamaño, costo y tiempo de desarrollo del software basado en cuentas hechas sobre los casos de uso para un sistema de software. Este exige la existencia de un modelo de casos de uso y la descripción detallada textual de los mismos, por lo que la labor deberá ser hecha cuando exista algún entendimiento del dominio del problema o cuando se esté realizando las labores de arquitectura y dimensionamiento del tamaño del sistema.(HERNÁNDEZ, SIGIFREDO E. BADANI, 2002b)

A continuación se detalla la secuencia de pasos que se deben desarrollar cuando se va a realizar la estimación del proyecto utilizando el método de Análisis de Puntos de Casos de Uso.

Cálculo de Puntos de Casos de Uso sin ajustar

Se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de Peso de los Actores sin ajustar

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de Actor	Descripción	Factor de Peso	Cantidad Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación API (Application Programming Interface)	0	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3	9

Figura 41: Factor de Peso de los Actores sin ajustar

Finalmente, se cuentan los actores de acuerdo a su clasificación, multiplicando cada subtotal por su factor de peso y sumando cada producto, obteniéndose el peso de los actores sin ajustar (UAW) igual a **9**.

Factor de Peso de los Casos de Uso sin ajustar

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad Casos de Uso	Total
Simple	El caso de uso contiene de 1 a 3 transacciones	5	4	20
Medio	El caso de uso contiene de 4 a 7 transacciones	10	3	30
Complejo	El caso de uso contiene más	15	0	0

	de 8 transacciones			
--	--------------------	--	--	--

Figura 42: Factor de Peso de los Casos de Uso sin ajustar

Finalmente, se cuentan los casos de uso de acuerdo a su clasificación a partir de la cantidad de transacciones que tienen los mismos, multiplicando cada subtotal por su factor de peso y sumando cada producto, obteniéndose el peso de los casos de uso sin ajustar (UUCW) igual a **50**.

Una vez obtenido los valores del factor de peso de los actores y casos de uso sin ajustar, se calcula entonces los puntos de casos de uso sin ajustar (UUCP), siendo este último resultado igual a **59**.

Cálculo de Puntos de Casos de Uso ajustados

Se calcula a partir de la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema, cada uno de los factores se cuantifican con un valor de 0 (no presente o sin influencia) hasta 5 (fuerte influencia o presencia).

Factor	Descripción	Factor de Peso	Valor asignado	Total
T1	Sistema distribuido	2	4	8
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	5	5
T4	Procesamiento interno complejo	1	4	4
T5	Código reutilizable	1	5	5
T6	Facilidad de instalación	0.5	2	1
T7	Facilidad de uso	0.5	4	2

T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	1

Figura 43: Factor de Complejidad Técnica

Para calcular **TCF** se realiza mediante la siguiente fórmula:

Totali = Factor Pesi * Valor Asignadoi

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Totali})$$

Finalmente, el Factor de complejidad técnica (TCF) tiene un valor igual a **1.14**.

Factor de ambiente

Este valor se calcula mediante la cuantificación de un conjunto de factores ambientales. Cada uno de los factores se cuantifican con un valor de 0 (aporte irrelevante) hasta 5 (aporte muy importante).

Factor	Descripción	Factor de Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10

E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3

Figura 44: Factor de ambiente

Para calcular **EF** se realiza mediante la siguiente fórmula:

$$\text{Totali} = \text{Factor Pesi} * \text{Valor Asignadoi}$$

$$\text{EF} = 1.4 - 0.03 * \Sigma (\text{Totali})$$

Finalmente, el Factor de ambiente (EF) tiene un valor igual a **0.68**.

Una vez obtenido los valores del factor de complejidad técnica y el factor de ambiente, se calcula entonces los puntos de casos de uso ajustado (UCP), siendo este último resultado igual **45.7368**.

Cálculo del Factor de Conversión

Este valor se calcula mediante la siguiente fórmula:

$$\text{TotalEF} = \text{CantEF} < 3 \text{ (entre E1 y E6)} + \text{CantEF} > 3 \text{ (entre E7, E8)}$$

$$\text{CF} = 20 \text{ horas-hombre (si TotalEF} \leq 2)$$

$$\text{CF} = 28 \text{ horas-hombre (si TotalEF} = 3 \text{ o TotalEF} = 4)$$

$$\text{CF} = \text{abandonar o cambiar proyecto (si TotalEF} \Rightarrow 5)$$

Finalmente, el Factor de Conversión (CF) tiene un valor igual a 20 horas-hombre.

Cálculo del Esfuerzo estimado en horas-hombres

Este valor se calcula mediante la siguiente fórmula:

$$\text{E} = \text{UCP} * \text{CF}$$

donde:

E: Esfuerzo estimado en horas-hombre

Finalmente, el Esfuerzo (E) tiene un valor igual a 914.736 horas-hombre.

El valor del esfuerzo calculado anteriormente no es más que el esfuerzo estimado del proyecto durante la implementación, una estimación más completa del esfuerzo total del proyecto, se puede obtener agregando a la estimación del esfuerzo obtenida anteriormente, las estimaciones de esfuerzo de la

demás actividades relacionadas con el desarrollo de software. En la tabla siguiente se muestra la distribución del esfuerzo entre las diferentes actividades de un proyecto, según datos estadísticos.

Actividad	Porcentaje (%)	Esfuerzo (horas-hombres)
Análisis	10	228.684
Diseño	20	457.368
Implementación	40	914.736
Pruebas	15	343.026
Sobrecarga (otras actividades)	15	343.026
Total	100	2286.84

Figura 45: Esfuerzo por actividades de un proyecto

A partir del valor obtenido anteriormente, se puede decir, que un hombre puede desarrollar el proyecto en **914.736 horas**. Los integrantes del equipo de desarrollo del proyecto serán profesores, que no solo trabajarán en el desarrollo del proyecto, sino que también realizarán otras tareas, por lo que dedicarán al proyecto aproximadamente 6 horas diarias, 5 días de la semana, lo cual representa 20 días al mes, lo que equivale a un total de **120 horas mensuales**. Finalmente, con un **equipo de 2 profesores**, se puede **desarrollar el proyecto en 4 meses aproximadamente**.

Hoy en día, existe una fuerte demanda en cuanto a las herramientas para la Gestión de Requisitos, debido a que ellas proporcionan a cada empresa y compañía una seguridad en el avance productivo de cada proyecto software que se elabora.

Esta herramienta será de gran utilidad para todos los proyectos, pues permitirá realizar con una mayor eficiencia, calidad y seguridad la gestión de requisitos; paso importante este en el desarrollo de todo proyecto software.

Además con la automatización de todas las actividades que se realizan durante la elaboración de cualquier software se garantiza un trabajo organizado y con menos posibilidad de cometer errores que imposibiliten al final, obtener un producto que satisfaga las necesidades de cliente y con la calidad

requerida .Se ahorrarán grandes volúmenes de papel, tinta, presillas y el procesamiento de los documentos será menos tedioso para los trabajadores de proyecto, pues tendrán todas las herramientas necesarias para desarrollar estas actividades disponibles desde sus puestos de trabajo.

4.2 Análisis de costos y beneficios

Anteriormente, se mencionaron todos los beneficios tanto tangibles como intangibles que reportará el uso de herramientas para la Gestión de Requisitos en cada proyecto. A pesar de que la herramienta no se ha utilizado en la universidad y se está proponiendo por primera vez, se puede decir con seguridad que será viable y fácil de utilizar, además que sus funcionalidades son similares a la herramienta que se está utilizando actualmente en la universidad para algunos proyectos. Sin embargo con la herramienta OSRMT no se va incurrir en gastos de recursos pues funciona bajo la licencia libre GPL, y su base de datos y lenguaje de programación son tecnologías de licencias libres con el objetivo de aprovechar todas las ventajas que estas proporcionan y cooperar con la migración del país a software libre.

Una vez realizado el análisis de factibilidad del sistema se puede concluir diciendo, que la estimación del tiempo para el desarrollo del proyecto es de 4 meses aproximadamente. Además la herramienta que se propone personalizar, brinda una serie de beneficios para los proyectos de la facultad 10 y para el resto de los proyectos en la universidad, la misma contribuye a minimizar los posibles errores en el proceso de Gestión de Requisitos, a obtener un producto con la calidad requerida y que satisfaga las necesidades del cliente, entre otros beneficios, por lo tanto, se puede decir, que es factible implementar dicha funcionalidad.

Conclusiones

En la actualidad, existe una fuerte demanda en cuanto al uso de las herramientas para la Gestión de Requisitos, debido a que ellas proporcionan a cada empresa y compañía una seguridad en el avance productivo de cada proyecto software que se elabora.

Con la realización de este trabajo se le dio cumplimiento a todos los objetivos planteados inicialmente. La herramienta propuesta es de gran utilidad para todos los proyectos, pues permitirá realizar con una mayor eficiencia, calidad y seguridad la Gestión de Requisitos; paso importante este, en el desarrollo de todo proyecto software.

La herramienta Open Source Requirement Management Tool propuesta en el presente trabajo posee características y funcionalidades similares a muchas otras herramientas, pero una de las premisas para su elección es la licencia libre bajo la cual está publicada y en los sistemas operativos que funciona, ya que es una herramienta multiplataforma y muy completa de manera general.

Sin embargo para que la misma satisfaga todas las necesidades de la facultad 10 y para que realice una Gestión de Requisitos lo más completa posible, se hace necesaria su personalización, agregándole un nuevo módulo llamado Reportes, dicho módulo podrá generar cualquier tipo de reporte que el cliente necesite y podrá ser guardado en un directorio específico en el formato HTML o PDF o puede ser impreso como estime conveniente el usuario.

El trabajo investigativo realizado constituye un manual de conocimiento y estudio, que resume las características principales de las herramientas para la Gestión de Requisitos utilizadas en el mundo y que puede ser usado por los proyectos para el desarrollo de sus productos, teniendo como ventaja esencial el ofrecimiento de información en español, ya que la mayoría de la documentación se encuentra en inglés. Es esencial para su utilización en la docencia como vía de aprendizaje ya que no existía ninguna en la facultad 10.

Recomendaciones

La realización de este trabajo posee una gran importancia, en aras de ofrecer una alternativa a la facultad y a la Universidad, del uso de herramientas libres para la Gestión de Requisitos y se recomienda:

- Promover los conocimientos del lenguaje Java, a través de los cursos optativos pertinentes.
- Utilizar la herramienta OSRMT para gestionar requisitos en todos los proyectos productivos de la facultad.
- Utilizar la dicha herramienta como material de estudio y trabajo en la docencia de la facultad para cursos venideros.
- Realizar un sitio de web que contenga toda la información de la herramienta OSRMT, su manual de usuario e instalación para que sirva de documentación y ayuda a todos los estudiantes y proyectos de la facultad.
- Poner en práctica la utilización del servidor Web que contiene la herramienta OSRMT.

Referencias Bibliográficas

FINKELSTEIN, A. y EMMERICH, W. *The Future of Requirements Management Tools*. London: [Consultado el: 21 de Abril de 2008]. Disponible en: <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/formtools.pdf>.

FONT, Y. C. y VÁZQUEZ, M. C. *Gestión de requisitos en el desarrollo del software*. Tutor: Delgado, I. R. C. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007. [Consultado el: 5 de Enero de 2008]. Disponible en: http://bibliodoc.uci.cu/TD/TD_0708_07.pdf

DIV, G. A. O. W. D. F. A. G. M. S. I. *Contracting for Computer Software Development Serious Problems Require Management Attention to Avoid Wasting Additional Millions*. USA: [Consultado el: 20 de Noviembre]. Disponible en <http://stinet.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA327099>

GASCA, G. P. *Requisitos No Funcionales* Madrid: [Consultado el: 22 de Abril de 2008]. Disponible en: <http://www.calidaddelsoftware.com/>.

HERNÁNDEZ, S. E. B. *Métricas de estimación de tamaño: Puntos de Caso de Uso. Ingeniería de Ejecución en Computación* Santiago de Chile: Universidad Central de Chile, [Consultado el: 10 de Abril de 2008]. Disponible en: http://eiec.ucecentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo_PCU.doc

JACOBSON, I. *El Proceso unificado de desarrollo de software*. 2004. [Consultado el: 20 de Abril de 2008]. Disponible en: <http://biblioteca.uci.cu>

JONES, D. A.; KAR, P. C., et al. *Interfacing Requirements Management Tools. In The Requirements Management Process. A First Look*. Texas: [Consultado el: 29 de Enero de 2008]. Disponible en: <http://www.itmweb.com/essay544.htm>.

KARAKOSTAS, V. y LOUCOPOLOS, P. *System Requirements Engineering* New York: [Consultado el: 2007 de 20 de Noviembre]. Disponible en: <http://citeseer.comp.nus.edu.sg/context/65419/0>

KEPLER, J. *ESPITI: European Software Process Improvement* Asturias: University Linz Systems Engineering and Automation. [Consultado el: 7 de Mayo de 2008]. Disponible en: <http://www.sea.uni-linz.ac.at/>

LANDAZURI, B. A. M. *Definición de Perfiles en Herramientas de Gestión de Requisitos*. Tutor: Caro, E. T. Tesis de Doctorado, Universidad Politécnica de Madrid, 2005. [Consultado el: 30 de Noviembre de 2008]. Disponible en: <http://lucio.ls.fi.upm.es/doctorado/Trabajos20042005/Mcdonald.pdf>

PABLO. *Visual Paradigm for UML* [Consultado el: 20 de Enero de 2008]. Disponible en: <http://www.tolontolon.es/foro/creaciun-ediciun-y-diseso/10128-visual-paradigm-5-professional-key.html>.

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. Quinta Edición ed. Madrid: McGraw-Hill, 2005. 992 p. ISBN 97-010-5473-3. [Consultado el: 23 de Mayo de 2007]. Disponible en: <http://www.mnlibros.com.ar/DespLibro.asp?Libro=9701054733>

REYNOSO, C. *Rational Unified Process* [Consultado el: 10 de Marzo de 2008]. Disponible en: <http://www.norulesatschool.com/process.php?url=uggc%3A%2F%2Ff.jvxvcrqvn.bet%2Fjvxv%2F>.

SMITH, A. *Open Source Requirements Management Tool* [Consultado el: 6 de Junio de 2008]. Disponible en: <http://www.osrmt.com>

SOMMERVILLE, I. y SAWYER, P. *Requirements Engineering: A Good Practice Guide*. USA: 1999. ISBN 0 471 97444 7. [Consultado el: 5 de Enero de 2007]. Disponible en: <http://www.comp.lancs.ac.uk/computing/resources/re-gpg/>

STANDICH, G. *Programando sin Sorpresas. Planeación de Software (tesis para obtener el grado de Maestro en Sistemas Computacionales)*. [Consultado el: 28 de Abril de 2007]. Disponible en: <http://www.agilies.com/tesis/>.

THAYER, R. H. y DORFMAN, M. *IEEE Software Requirement Engineering, Second Edition*. New York: IEEE Computing Society, 1997. [Consultado el: 30 de Mayo de 2007]. Disponible en: <http://www.ieeexplore.ieee.org/jel5/52/20258/00936219.pdf>

Bibliografía Consultada

ACTIS, B. Y. *Software: Santa Fe es una de las principales productoras Argentina*: [Consultado el: 24 de Abril de 2008]. Disponible en: <http://www.laopinion-rafaela.com.ar/opinion/2007/10/14/r7a1404.php>.

AHMED, E. *Business News Americas Colombia: Productora de Software* [Consultado el: 24 de Abril de 2008]. Disponible en: <http://www.psl.com.co/>

ALEXANDER, I. *Requirements Tools* [Consultado el: 12 de Enero de 2008]. Disponible en: http://easyweb.easynet.co.uk/~iany/other/other_sites.htm.

ALEXANDER, I. *The Requirements Engineering Specialist Group of the British Computer Society* [Consultado el: 20 de Febrero de 2008]. Disponible en: <http://www.resg.org.uk/newsletter.html>.

ANAYA, V. y LETELIER, P. *SmarTTrace: Una Herramienta para Trazabilidad de Requisitos en Proyectos basados en UML*. [Consultado el: 12 de Enero de 2008]. Disponible en: http://www.wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER02/anaya.pdf

BROOKS, F. P. *The Mythical Man-Month: Essays on Software Engineering*. North Carolina: Addison-Wesley, 1995. ISBN 0-201-83595-9. [Consultado el: 15 de Enero de 2008]. Disponible en: http://en.wikipedia.org/wiki/The_Mythical_Man-Month

COMMUNITY, C. C. *CaseComplete®* [Consultado el: 24 de Abril de 2008]. Disponible en: <http://www.casecomplete.com/default.aspx>.

COMMUNITY, S. N. *SourceForge.NET* Chicago: [Consultado el: 12 de Febrero de 2008]. Disponible en: <http://sourceforge.net/>.

COMPANY, G. *GatherSpace.com* California: [Consultado el: 25 de Enero de 2008]. Disponible en: <http://www.gatherspace.com/>.

COMPANY, I. *Telelogic An IBM Company* [Consultado el: 22 de Febrero de 2008] Disponible en: <http://www.telelogic.es/index.cfm>.

COMPANY, I. B. M. C. I. *Rational RequisitePro* España: [Consultado el: 2 de Mayo de 2008]. Disponible en: http://www.ibm.com/developerworks/downloads/r/rrp/?S_CMP=eCAT.

COMPANY, V. S. *Visure the requirements company* Madrid: [Consultado el: 20 de Febrero de 2008]. Disponible en: <http://www.visuresolutions.com/>.

CORPORATION, B. S. *Borland® Caliber® Analyst Collaborative Software Requirements Engineering* [Consultado el: 5 de Diciembre de 2007]. Disponible en: <http://www.borland.com/us/products/caliber/index.html>.

CRUZ, A. R. B. y MINIET, Y. V. *Procedimiento para desarrollar la Ingeniería de Requisitos en el proyecto Sistema de Gestión Penitenciaria*. Tutor: Pérez, M. H. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007. [Consultado el: 12 de Noviembre de 2007]. Disponible en: http://bibliodoc.uci.cu/TD/TD_0488_07.pdf

CUSUMANO, M. y SHIRKY, C. *Perspectives on Free and Open Source Software*. London: 2005. 571 p. ISBN 0-262-06246-1. [Consultado el: 20 de Noviembre de 2007]. Disponible en: <http://mitpress.mit.edu/catalog/item/default.asp?tid=10477&ttype=2>

CHINYAVONG, L. *Nace Visure, Empresa Española Especializada en Herramientas de Gestión de Requisitos* Madrid: Abanico Publicidad Integral. [Consultado el: 22 de Noviembre]. Disponible en: <http://www.noticias.info/Asp/aspComunicados.asp?nid=317325>.

DIV, G. A. O. W. D. F. A. G. M. S. I. *Contracting for Computer Software Development Serious Problems Require Management Attention to Avoid Wasting Additional Millions*. USA: [Consultado el: 20 de Noviembre]. Disponible en <http://stinet.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA327099>

DUQUE, J. G.; MARTÍNEZ, A. B. B., *et al.* *Modelo de reutilización de requisitos funcionales utilizando técnicas formales*. 2000. 138 p. ISBN 84-8427-074-2. [Consultado el: 5 de Diciembre]. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=1200490>

ESCALONA, M. J. y KOCH, N. *Ingeniería de Requisitos en Aplicaciones para la Web. Un estudio comparativo* Sevilla: [Consultado el: 22 de Noviembre de 2007]. Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>.

FIGUEROA, P. *Metodologías de Desarrollo Software* Bogotá: [Consultado el: 3 de Marzo de 2008]. Disponible en: <http://kybele.escet.urjc.es/documentos/ISG/Estructurado/%5BISG-2006-07%5DMetodologiasDesarrolloSW.pdf>.

FINKELSTEIN, A. y EMMERICH, W. *The Future of Requirements Management Tools*. London: [Consultado el: 21 de Abril de 2008]. Disponible en: <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/formtools.pdf>.

FONT, Y. C. y VÁZQUEZ, M. C. *Gestión de requisitos en el desarrollo del software*. Tutor: Delgado, I. R. C. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007. [Consultado el: 5 de Enero de 2008]. Disponible en: http://bibliodoc.uci.cu/TD/TD_0708_07.pdf

FORERO, P. A. F. *Metodología de desarrollo de software Orientado por Objetos* Bogotá: [Consultado el: 20 de Febrero de 2008]. Disponible en: <http://www.cs.ualberta.ca/~pfiguero/soo/metod/>.

GARCÍA, V. G. *Estudio de herramientas de análisis*. [Consultado el: 10 de Enero de 2008]. Disponible en: <http://petra.euitio.uniovi.es/~i1656157/>.

GEMMELL, M. y APHANG, J. *REM Solutions* [Consultado el: 24 de Abril de 2008]. Disponible en: <http://www.remsolu.com/>.

GODA SOFTWARE, I. *To Goda Software* [Consultado el: 5 de Marzo de 2008]. Disponible en: <http://www.analysttool.com/>.

- HERNÁNDEZ, J. *Ciclo de vida del Software. Gestión de requisitos* [Consultado el: 15 de Noviembre de 2007]. Disponible en: <http://www.danysoft.info/free/gesrequisitos.pdf>
- HERNÁNDEZ, S. E. B. *Métricas de estimación de tamaño: Puntos de Caso de Uso. Ingeniería de Ejecución en Computación* Santiago de Chile: Universidad Central de Chile, [Consultado el: 10 de Abril de 2008]. Disponible en: http://eiec.uccentral.cl/ftp/material/apuntes/iec61/Diseno/Metodo_PCU.doc
- HERRERA, Y. E. y MUÑOZ, L. C. *Sistema de Gestión de Información del DDC-Ingeniería y Gestión de Software UCI*. Tutor: Pérez, M. Y. F. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007. [Consultado el: 4 de Abril de 2008]. Disponible en: http://bibliodoc.uci.cu/TD/TD_0527_07.pdf
- HURTADO, G. P. G. *Requisitos No Funcionales* Madrid: [Consultado el: 22 de Abril de 2008]. Disponible en: <http://www.calidaddelsoftware.com/>.
- JACOBSON, I. *El Proceso unificado de desarrollo de software*. 2004. [Consultado el: 20 de Abril de 2008]. Disponible en: <http://biblioteca.uci.cu>
- JONES, D. A.; KAR, P. C., et al. *Interfacing Requirements Management Tools. In The Requirements Management Process. A First Look*. Texas: [Consultado el: 29 de Enero de 2008]. Disponible en: <http://www.itmweb.com/essay544.htm>.
- KARAKOSTAS, V. y LOUCOPOLOS, P. *System Requirements Engineering* New York: [Consultado el: 20 de Noviembre de 2007]. Disponible en: <http://citeseer.comp.nus.edu.sg/context/65419/0>
- KEPLER, J. *ESPITI: European Software Process Improvement* Asturias: University Linz Systems Engineering and Automation. [Consultado el: 7 de Mayo de 2008]. Disponible en: <http://www.sea.uni-linz.ac.at/>
- LANDAZURI, B. A. M. *Definición de Perfiles en Herramientas de Gestión de Requisitos*. Tutor: Caro, E. T. Tesis de Doctorado, Universidad Politécnica de Madrid, 2005. [Consultado el: 30 de Noviembre de 2008]. Disponible en: <http://lucio.ls.fi.upm.es/doctorado/Trabajos20042005/Mcdonald.pdf>

LETELIER, P. *Introducción a Rational Unified Process (RUP)* Valencia [Consultado el: 3 de Marzo de 2008]. Disponible en: <http://pid.dsic.upv.es>

LÓPEZ, O. P. *Ejemplo de desarrollo software utilizando la metodología RUP*. Valencia: [Consultado el: 13 de febrero de 2008]. Disponible en: <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/Requisitos.html>.

LOVELLE, J. M. C. *Herramientas de Herramientas de Desarrollo*. Asturias: [Consultado el: 23 de Mayo de 2008]. Disponible en: <http://petra.euitio.uniovi.es/~hd/herramientas/index.html>.

LUEGES, G. J. *Diseño del Módulo de Catalogación para el Departamento de Procesos Técnicos de la Biblioteca Nacional "José Martí"*. Tutor: Leyva, M. D. L. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007. [Consultado el: 10 de Junio de 2008]. Disponible en: http://bibliodoc.uci.cu/TD/TD_0896_07.pdf

MARTÍNEZ, J. C. M. *Análisis Comparativo de Técnicas, Metodologías y Herramientas de Ingeniería de Requerimientos*. Centro de Investigación y Estudios Avanzados del IPN CINVESTAV, 2004. [Consultado el: 4 de Marzo de 2008]. Disponible en: <http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2004/resumenJuanCarlosM.html>

MOLPECERES, A. *Procesos de Desarrollo: RUP, XP y FDD*. Alemania: [Consultado el: 16 de Marzo de 2008]. Disponible en: www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF

ORTIZ, A. *Conferencia sobre Ingeniería de Software, Perspectivas de una nueva profesión*. Santo Domingo Este: [Consultado el: 22 de Noviembre de 2007]. Disponible en: <http://chapters.computer.org/dominicana/index.htm>.

PABLO. *Visual Paradigm for UML* [Consultado el: 20 de Enero de 2008]. Disponible en: <http://www.tolontolon.es/foro/creaciun-ediciun-y-diseso/10128-visual-paradigm-5-professional-key.html>.

PALACIO, J. *Gestión y procesos en empresas de software*. [Consultado el: 22 de Abril de 2008]. Disponible en: <http://www.navegapolis.net/content/view/156/59/>.

PERALTA, M. *Estimación del esfuerzo basada en casos de uso*. Buenos Aires: [Consultado el: 5 de Mayo de 2008]. Disponible en: <http://www.itba.edu.ar/capis/webcapis/planma.html>.

PIEDRA, N. *La ingeniería de requisitos...* Barcelona: [Consultado el: 22 de Noviembre de 2007]. Disponible en: <http://barceludena.wordpress.com/2007/09/28/slideshare-id119787docingeniera-de-requisitos2621w425/>.

PINO, F. J.; GARCÍA, F., et al. *Herramienta de Soporte a la Valoración Rápida de Procesos Software*. Ciudad Real: [Consultado el: 14 de Diciembre de 2007]. Disponible en: <http://ieeexplore.ieee.org/iel5/9907/4378505/04378509.pdf?isnumber=4378505&prod=JNL&arnumber=4378509&arSt=218&ared=223&arAuthor=Pino%2C+Francisco+J.%3B+Garcia%2C+Felix%3B+Piattini%2C+Mario>.

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. Quinta Edición ed. Madrid: McGraw-Hill, 2005. 992 p. ISBN 97-010-5473-3. [Consultado el: 23 de Mayo de 2007]. Disponible en: <http://www.mnlibros.com.ar/DesplLibro.asp?Libro=9701054733>

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. Quinta Edición ed. Madrid: Mc-Graw-Hill, 2002. 601 p. ISBN 84-481-3214-9. [Consultado el: 5 de Mayo de 2007]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>

REYNOSO, C. *Rational Unified Process* [Consultado el: 10 de Marzo de 2008]. Disponible en: [http://www.norulesatschool.com/process.php?url=uggc%3A%2F%2Ff.jvxvcrqvn.bet%2Fjvxv%](http://www.norulesatschool.com/process.php?url=uggc%3A%2F%2Ff.jvxvcrqvn.bet%2Fjvxv%2F).

SMITH, A. *Open Source Requirements Management Tool* [Consultado el: 6 de Junio de 2008]. Disponible en: <http://www.osrmt.com>

SOMMERVILLE, I. y SAWYER, P. *Requirements Engineering: A Good Practice Guide*. USA: 1999. ISBN 0 471 97444 7. [Consultado el: 5 de Enero de 2007]. Disponible en: <http://www.comp.lancs.ac.uk/computing/resources/re-gpg/>

STANDICH, G. *Programando sin Sorpresas. Planeación de Software (tesis para obtener el grado de Maestro en Sistemas Computacionales)*. [Consultado el: 28 de Abril de 2007]. Disponible en:

<http://www.agilies.com/tesis/>.

THAYER, R. H. y DORFMAN, M. *IEEE Software Requirement Engineering, Second Edition*. New York: IEEE Computing Society, 1997. [Consultado el: 30 de Mayo de 2007]. Disponible en:

<http://www.ieeeexplore.ieee.org/iel5/52/20258/00936219.pdf>

TORO, A. D. *Herramienta REM* Sevilla: [Consultado el: 2 de Mayo de 2008]. Disponible en:

http://www.lsi.us.es/descargas/descarga_programas.php?id=3.

TOVAR, E. *La Ingeniería de Requisitos Software como factor de éxito en Proyectos Software*. Madrid: [Consultado el: 22 de Noviembre de 2007]. Disponible en: [http://is.ls.fi.upm.es/doctorado/Curso2007-](http://is.ls.fi.upm.es/doctorado/Curso2007-2008/Asignaturas/IngReq.html)

[2008/Asignaturas/IngReq.html](http://is.ls.fi.upm.es/doctorado/Curso2007-2008/Asignaturas/IngReq.html).

TOVILLAS, D. R. *Borland CaliberRM™* Buenos Aires: [Consultado el: 4 de Febrero de 2008]. (Daniel Rodolfo Tovillas). Disponible en: <http://www.samsistemas.com.ar>.

WIEGERS, K. E. *More About Software Requirements Thorny Issues and Practical Advice*. Washington: Microsoft Press 2006. 216 p. ISBN 0735622671. [Consultado el: 30 de Octubre de 2007]. Disponible en: <http://www.amazon.com/More-About-Software-Requirements-Practical/dp/0735622671>

WIEGERS, K. E. *Software Requirements. Second Edition*. Washington: Microsoft Press, 2003. 516 p. [Consultado el: 2 de Mayo de 2007]. Disponible en: <http://www.amazon.com/Software-Requirements-Second-Pro-Best-Practices/dp/0735618798>

Glosario de términos

Apache: Servidor de páginas Web desarrollado Apache Software Foundation, organización de miles de voluntarios que colaboran para la creación de software de libre distribución. Actualmente se considera líder en el mercado de los servidores, por encima de las soluciones propietarias.

Asignación: Establecimiento de lo que corresponde a algo o alguien específico.

Backup: Copia de seguridad. Es la acción de copiar documentos, archivos o ficheros de tal forma que puedan recuperarse en caso de fallo en el sistema.

Base de datos: Conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior.

Browser: Es un navegador, programa de aplicación que proporciona una manera de mirar y de obrar recíprocamente con toda la información sobre el World Wide Web (www)

Business: En la economía, es un negocio (también llamado firma o empresa), entidad de organización legalmente reconocida diseñada para proporcionar mercancías y/o servicios a los consumidores o a las entidades corporativas.

Calidad: Es la percepción que el cliente de un producto o servicio, es una fijación mental del consumidor que asume conformidad con dicho producto o servicio y la capacidad del mismo para satisfacer sus necesidades.

CaliberRM: Es un Sistema Enterprise de Gerenciamiento de requerimientos, solución sencilla de usar de gestión de requerimientos altamente colaborativa preparada para accesos a través de Internet que permite a los equipos de desarrollo definir, gestionar y comunicar requisitos cambiantes a lo largo del ciclo de vida.

Ciclo de vida: Período de tiempo que comienza con la concepción del producto del software y termina cuando el producto esta disponible para su uso.

Copyright o Derechos de Autor: Derecho de propiedad que se genera de forma automática por la creación de diversos tipos de obras y que protege los derechos e intereses de los creadores de trabajos literarios, dramáticos, musicales y artísticos, grabaciones musicales, películas, emisiones radiadas o televisadas, programas por cable o satélite y las adaptaciones tipográficas de los libros, folletos, impresos, escritos y cualesquiera otras obras de la misma naturaleza. Una vez que un trabajo ha sido creado de una forma los creadores o titulares de derechos pueden guardarse para sí mismos (o autorizar a otros) la potestad exclusiva de copiar, publicar, representar, emitir por radio o televisión, o adaptar su obra.

Crisis del Software: Término utilizado para definir a los problemas que existían y existen en el desarrollo de sistemas de software.

DOORS: Herramienta de gestión de requisitos líder del sector, maximiza el valor de las iniciativas de optimización de procesos de negocio y aumenta la calidad de los proyectos de desarrollo de software e ingeniería de sistemas mejorando la comunicación de los requisitos y la colaboración.

GatherSpace: Es una herramienta que permite la edición, compartición y distribución de requisitos on-line, un poco al estilo de Google Docs y otras aplicaciones Web 2.0.

Gestión de la configuración: conjunto de procesos destinados a asegurar la integridad de todos los productos obtenidos durante cualquiera de las etapas del desarrollo de un sistema de Información (SI), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo.

Hipervínculos: Es un enlace, normalmente entre dos páginas web.

Impacto: Conjunto de posibles efectos que puedan ocasionar la modificación o cambio del entorno natural, como consecuencia de obras u otras actividades.

Internet: Método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominados TCP/IP y garantiza que las redes físicas heterogéneas funcionen como una red (lógica) única.

Interfaz: Es el conjunto de comandos y/o métodos que permiten la intercomunicación del programa con cualquier otro programa o entre partes (módulos) del propio programa o elemento interno o externo.

iReport: Es un diseñador de informe de la fuente abierta que ayuda al revelador con esa tarea. Permite diseñar el informe visualmente y ofrece una inspección previa de la herramienta.

IRqA: Herramienta CASE de Ingeniería de Requisitos, diseñada para soportar las actividades realizadas en el proceso de especificación de sistemas.

JasperReports: Es una poderosa herramienta de creación de informes Java open source que tiene la habilidad de entregar contenido rico en el monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Metodología: Definición del proceso de investigación que sigue a la iniciación y permite sistematizar los métodos, procedimientos y las técnicas necesarias para llevarlas a cabo.

MS Access: Software de Microsoft para la creación y mantenimiento de Bases de Datos avanzadas. Es muy popular entre los desarrolladores que mantienen Bases de Datos en línea con Windows NT.

Oracle: Sistema de gestión de base datos relacional. Es uno de los motores de base de datos usado en el mercado y más potentes.

Plantilla: Forma de dispositivo que proporciona una separación entre la forma o estructura y el contenido. Es un medio o un instrumento que permite guiar, portar o consumir un diseño o esquema predefinido.

Producto de software: Conjunto de programas, procedimientos y opcionalmente documentación asociada que se entrega al usuario como resultado.

PostgreSQL: Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD.

Repositorio: Sitio centralizado donde se almacena y mantiene información digital, habitualmente base de datos o archivos informáticos.

Requisite Pro: Herramienta de trabajo que mantiene los equipos de proyectos al día gracias a la creación, análisis y gestión de los requisitos de aplicaciones y casos de uso.

Requisito: Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

Reutilización de software: Proceso de implementar o actualizar sistemas de software usando activos de software existentes.

Riesgo: Posibilidad de que un peligro pueda llegar a materializarse.

Scripts: Guión. Secuencia de código sin compilar que puede ejecutar acciones, usualmente poco complejas. Están escritos en lenguaje XML y no se ejecutan solos, necesitan de un software que los interprete.

Scrum: Metodología ágil, aplicada originalmente por Jeff Sutherland y elaborada mas formalmente por Ken Schwaber. Scrum aplica principios de control industrial, junto con experiencias metodológicas de Microsoft, Borland y Hewlet Packard.

Sistema software: Conjunto de programas de ordenador, procedimientos y opcionalmente la documentación y datos asociados, necesarios para el funcionamiento de un sistema.

Swing: Es una biblioteca gráfica para Java que forma parte de las Java Foundation Classes (JFC). Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas.

Trazabilidad: Conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada producto desde su origen hasta su destino final.

Validación: Confirmación mediante examen y aportación de pruebas objetivas de que se cumplen los requisitos concretos para un uso determinado.

Viabilidad: Condición que hace posible el funcionamiento del sistema, proyecto o idea el que califica, atendiendo a sus características tecnológicas y a las leyes de la naturaleza involucradas.

Windows: Sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft.

Glosario de abreviaturas

BD: Base de Datos.

CMMI: Siglas de “Capability Maturity Model Integration”, modelo desarrollado por el instituto de ingeniería de software (SEI) para medir el nivel de capacidad o madurez de una organización en la ejecución de los procesos relacionados con el desarrollo de un producto de software.

CPU: Central Processing Unit, unidad central de procesamiento, el procesador es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de computadora.

EULA: End User License Agreement, es una licencia por la cual el uso de un producto sólo está permitido para un único usuario (el comprador).

GB: Gigabyte es una unidad de medida informática puede equivalerse a 2^{30} bytes o a 10^9 bytes, según el uso.

GNU: Significa «ñu» en inglés, es un acrónimo recursivo de «GNU No es Unix» y en español se pronuncia fonéticamente.

GPL: Licencia General Pública creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

HP/UX: Es la versión de Unix desarrollada y mantenida por Hewlett-Packard desde 1983.

IBM: Siglas de internacional Business Machines. Empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

IEEE: The Institute of Electrical and Electronics Engineers, es una asociación técnico-profesional mundial, formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática e ingenieros en telecomunicación.

INCOSE: The International Council on Systems Engineering, es un consejo internacional de ingeniería de sistemas que tiene como misión avanzar el estado y la práctica de la ingeniería de sistemas en industrias, academias, y el gobierno, etc.

J2EE: Es una plataforma que añade a Java la funcionalidad necesaria para convertirse en un lenguaje orientado al desarrollo de servicios en Internet.

LDAP: Lightweight Directory Access Protocol, es un Protocolo Ligero de Acceso a Directorios de tipo cliente-servidor para acceder a un servicio de directorio.

Linux: Es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU o GPL, es decir que es software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds.

MB: Megabyte es una unidad de medida de cantidad de datos informáticos.

MSF: Microsoft Solutions Framework, marco para el desarrollo de sistemas de software basado en principios, modelos, disciplinas, conceptos, prácticas y recomendaciones propias, derivada de la experiencia de Microsoft. Empleadas en los entornos de metodologías ágiles.

OSRMT: Open Source Requirements Tool, herramienta multiplataforma, diseñada para servir el ciclo de vida completo del desarrollo del software para el funcionamiento de proyectos.

OTAN: Siglas de la Organización del Tratado del Atlántico Norte.

PDF: Portable Document Format, Formato de Documento Portátil es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Formato más extendido en Internet para el intercambio de documentos.

RAM: Random Access Memory, la memoria de acceso aleatorio, o memoria de acceso directo, más conocida como memoria RAM, se compone de uno o más chips y se utiliza como memoria de trabajo para programas y datos.

REM: REquisite Management, es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software.

RM: Es una orden de Unix usada para eliminar archivos y directorios del sistema de archivos. Proviene de la palabra remove que significa "borrar" en inglés.

RUP: Rational Unified Process, proceso de ingeniería del software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades en las organizaciones de desarrollo del software. Se trata de un proceso integrado en un producto, desarrollo y mantenido por Rational Software, e integrado en su conjunto de herramientas de desarrollo.

SEI: Software Engineering Institute, es un instituto federal estadounidense de investigación y desarrollo, para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares.

Solaris: Es un sistema operativo desarrollado por Sun Microsystems. Es un sistema certificado como una versión de UNIX. Aunque Solaris en sí mismo aún es software propietario, la parte principal del sistema operativo se ha liberado como un proyecto de software libre denominado Opensolaris. Solaris

puede considerarse uno de los sistemas operativos más avanzados. Su denominación así a su sistema operativo.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN).

UCI: Universidad de las Ciencias Informáticas

UML: Unified Modeling Language, lenguaje unificado de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

URL: Uniform Resource Locator, Localizador Universal de Recursos y se refiere a la dirección única que identifica a una página Web en Internet.

Windows NT: Windows NT (Nueva Tecnología) es una familia de sistemas operativos producidos por Microsoft, de la cual la primera versión fue liberada en julio de 1993. Fue diseñado para ser un poderoso sistema operativo multiusuario, basado en lenguaje de alto nivel, independiente del procesador, con rasgos comparables con Unix.

XML: Extensible Markup Language, lenguaje de marcas extensible, en realidad es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

XP: EXtreme Programming, metodología heterodoxa de programación. Es la más popular de las denominadas metodologías ágiles. Surgida a partir de la metodología de trabajo empleada por Kent Beck, Wark Cunningham y Martin Fowler en el desarrollo del proyecto C3 para Chrysler.

XSLT o Transformaciones XSL: Es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Anexos**Anexo1: Entrevista a proyectos productivos de la Facultad 10****Entrevista a proyectos productivos de la Facultad 10****Tema: Gestión de los Requisitos del Software****Proyecto:** _____**Nombre del Entrevistado:** _____**Rol:** _____

1. ¿En su proyecto realizan la Gestión de Requisitos?

Sí ___ No___

Seleccione por qué no la realiza:

- Desconocen que existe
- No lo consideran importante
- No existe la herramienta indicada
- Otras causas. Menciónelas: _____

2. ¿Ha tenido su proyecto alguna dificultad por una inadecuada Gestión de Requisitos?

Sí ___ No___

Mencione la(s) dificultad(es) _____

3. ¿En la actualidad, mediante que forma su proyecto realiza la Gestión de Requisitos?

Manual___ Mediante una herramienta informática ___

4. Conoce usted alguna(s) de la(s) herramienta(s) para la Gestión de Requisitos.

Sí ___ No ___

Menciónelas _____

5. ¿Sabe cuál(es) herramienta(s) para la Gestión de Requisitos funciona(n) sobre plataforma libre?

Sí ___ No ___

Menciónelas _____

6. ¿Utilizan alguna herramienta para la Gestión de Requisitos en el proyecto?

Sí ___ No ___

a.) En caso de utilizarla, diga cuál:

b.) En el caso de que usted realice la Gestión de Requisitos de otra manera, describa brevemente como lo ejecuta.

c.) En el caso de no utilizarla, diga porqué no lo hace:

7. ¿Conoce usted los pasos principales que se realizan durante la Gestión de Requisitos?

Sí ___ No ___

Menciónelos. _____

8. ¿Desearía utilizar en su proyecto una herramienta que respalde la Gestión de Requisitos sobre plataforma libre?

Sí ___ No ___

¿Por qué?

Anexo 2: Versiones de la herramienta OSRMT

Versiones de OSRMT		
Versiones	Fecha	Comentarios
0.5	16 de abril del 2006	Versión preliminar.
1.0	15 de mayo del 2006	Primera versión estable.
1.1	20 de junio del 2006	Algunos cambios en la interfaz.
1.2	28 de agosto del 2006	Se modifican las formas para los requisitos y los controles de usuarios se hacen más fáciles.
1.3	10 de octubre del 2006	Adiciona un módulo de trazabilidad.
1.4	12 de diciembre del 2006	Mejoras significativas el funcionamiento para manejar grandes cantidades de artefactos y generar informes en XML.
1.5	26 de marzo del 2007	Mejoras significativas de la información, junto con un cliente web.

Anexo 3: Manual de instalación de la herramienta OSRMT

Elaborado por:

Leidy Ramos González

Renier Socarras Pérez

Fecha de última revisión:

24 de abril 2008

Índice de contenido

1. Instalación de la versión 5.0 estándar de Java.

Entorno de trabajo.

Es necesaria la instalación de Java puesto que la herramienta esta programada sobre lenguaje Java y el instalador es un archivo .jar que necesita ser ejecutado con la máquina virtual de Java.

De las versiones anteriores a la 5.0 de Java solo se puede instalar hasta la 1.5, que permite el funcionamiento de la herramienta OSRMT.

Para verificar cual versión de Java es la que usted tiene funcionando solo con los comandos en la consola cmd para Windows y un terminal para Linux, podrá determinarla y si es menor a la versión Java1.5 debe ser cambiada y recomendamos la versión Java 5.0.

Si se siente inseguro de cual instalar recomendamos el link JRE.

Para más información visitar el sitio oficial:

<http://java.sum.com>

2. Instalación del Cliente OSRMT para la versión 1.5

Instalación del software.

Descargamos del sitio

<http://www.osrmt.com>

La versión osrmt_v1_5.zip

Cuando se extraiga el archivo osrmt.jar los ejecutamos mediante líneas de comandos:

```
java -jar osrmt15.jar
```

O simplemente clic sobre el archivo. Ver figura 1.

**Figura 1**

Seleccione el lenguaje de instalación y presione OK.

Después de leer y estar de acuerdo con el Readme.txt presione siguiente y seleccione la trayectoria que tomara la instalación (Ver figura 2), si lo instala sobre un directorio existente SOBRESCRIBE la base datos y el connection. xml y se pierden los datos ya existentes.

**Figura 2**

Luego se seleccionan los paquetes que se desean instalar. Ver figura 3

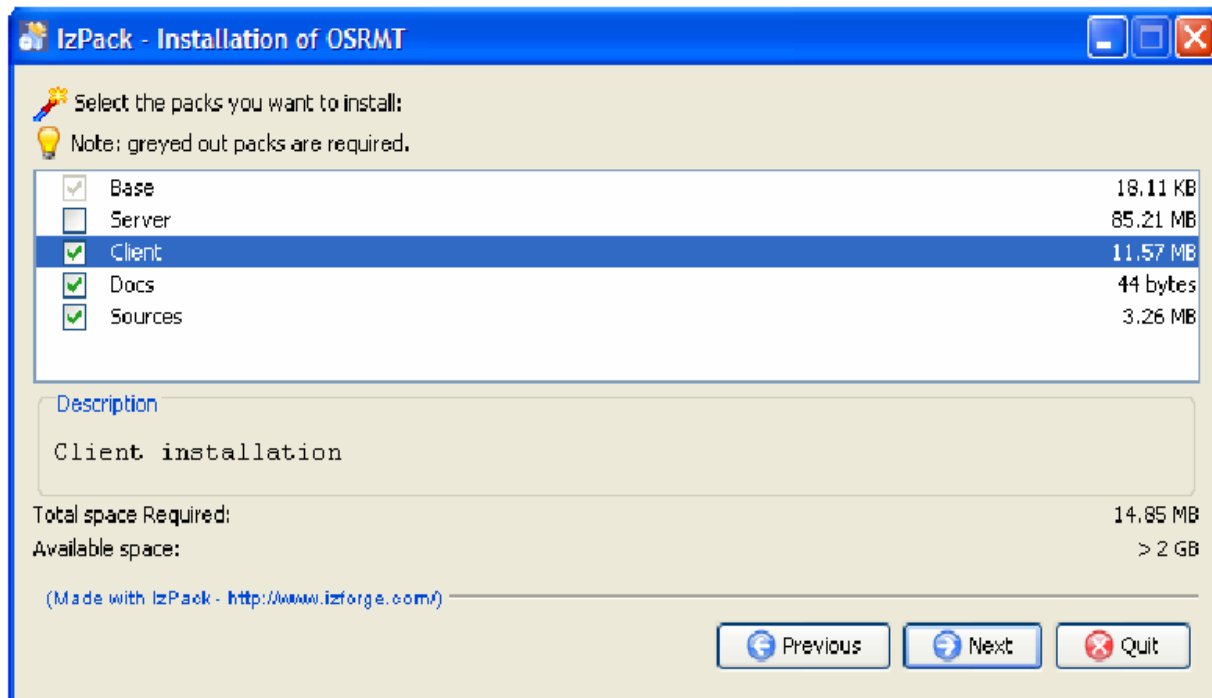


Figura 3

Ejecutamos el acceso directo creado o solamente vamos al directorio client/ y para Windows se ejecuta run.bat y para Linux se ejecuta run.sh dándole los permisos necesarios a run.sh para la ejecución.

Si se desea se pueden crear accesos directos en el escritorio. Ver figura 4

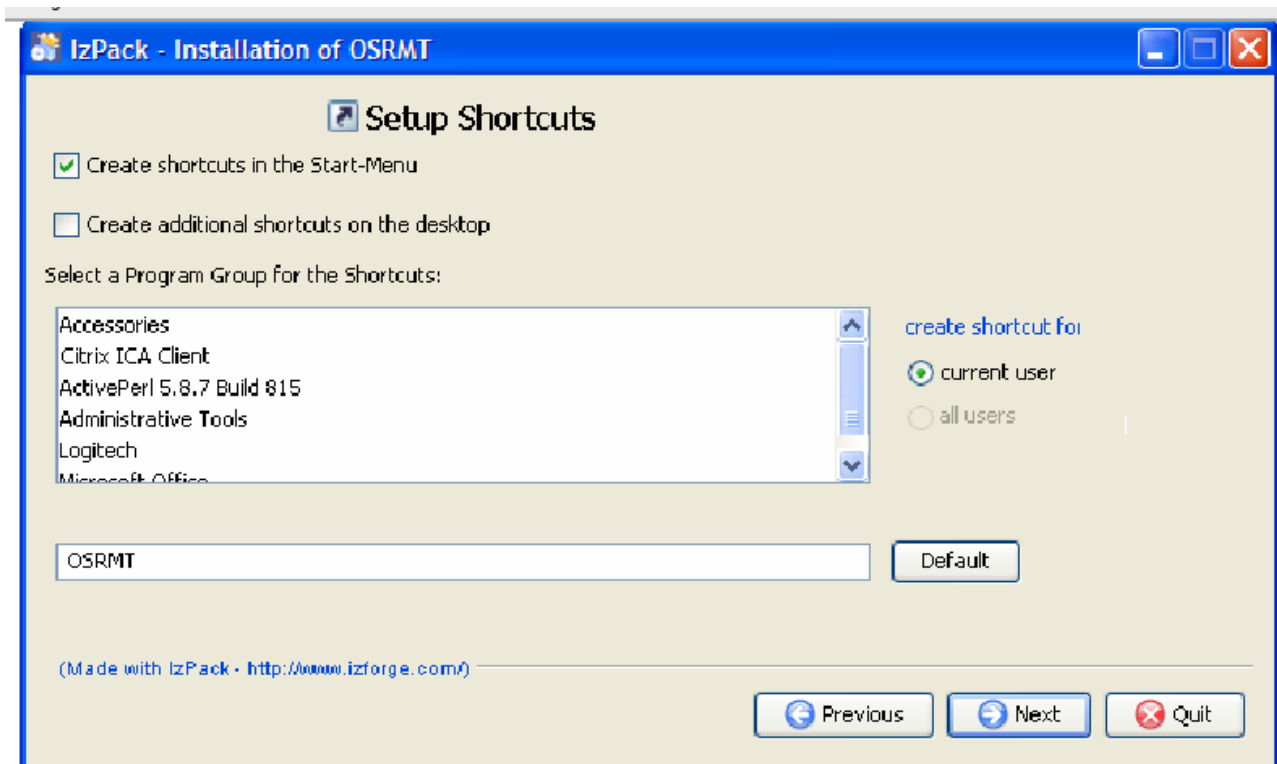


Figura 4

3. Creando nuevas bases datos para su uso.

Utilizando base datos libre para su uso.

Dentro del sistema de directorios, escoger en "client" el fichero "connection.xml" que nos interese en función de la base de datos que se desee utilizar. Ver figura 5

Podemos encontrar:

connection.access.xml

connection.mysql.xml

connection.oracle.xml

connection.postgresql.xml

connection.sqlserver.xml

Basado en que será un sistema para el uso dentro del software libre, se escogió el connection.postgresql.xml y se copia el código dentro de "connection.xml" sin modificar valores específicos a menos que sea necesario para el uso en específico dentro de un ambiente de trabajo. Se pueden mantener las opciones por defecto.

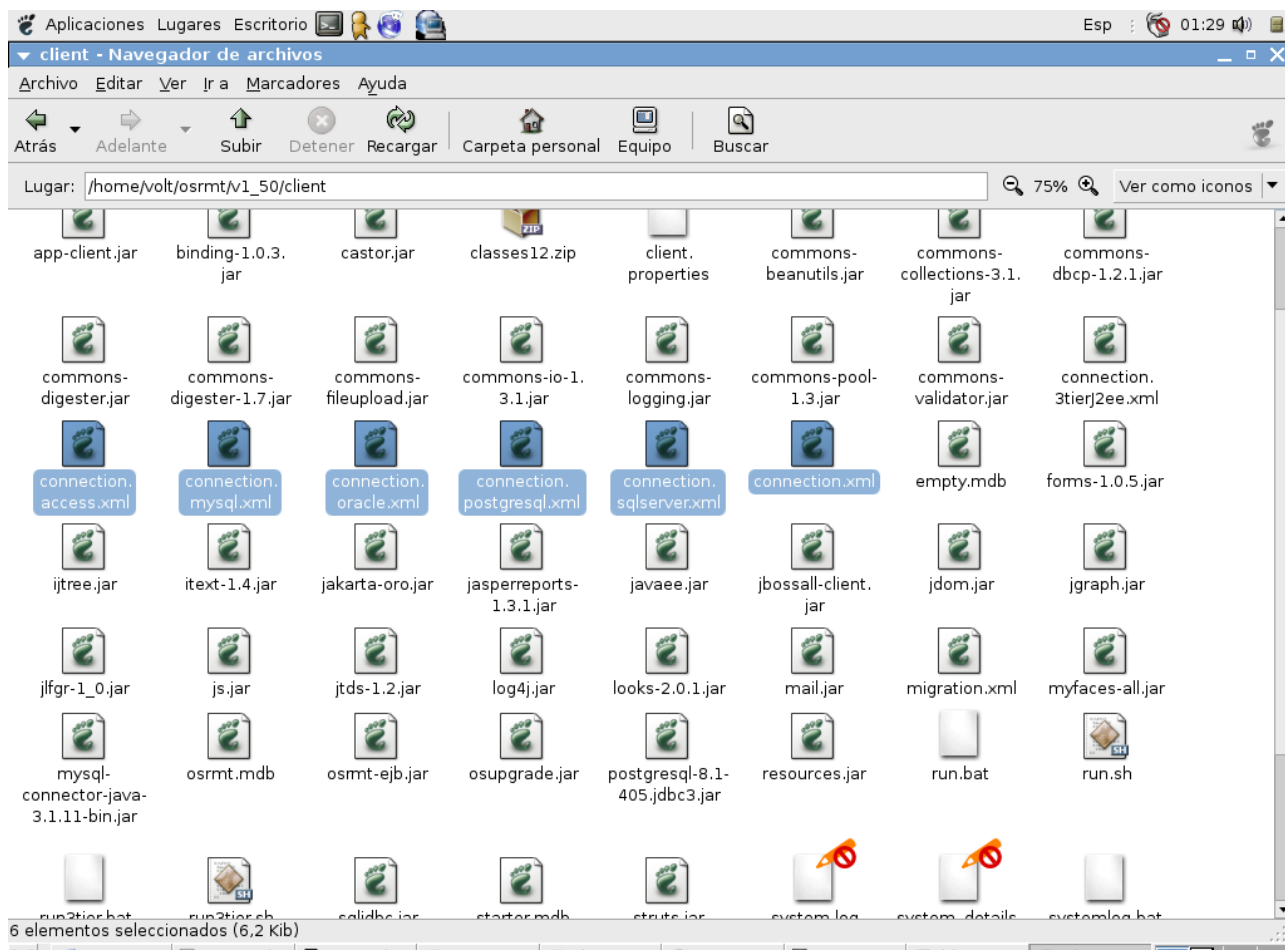


Figura 5

Servidor necesario para el funcionamiento de la base datos.

Se necesita la instalación de PostgreSQL 8.1.x o cualquier otra versión más reciente. Además de utilizar el gestor de bases de datos que sea de su agrado para manejar y mantener el control de la base de datos.

Proponemos el uso de pgadmin3. Ver figura 6

También en líneas de comandos se puede hacer el trabajo aunque es más tedioso.

```
user:-$
```

```
user:-$ psql -U postgres template1
```

Ejecutar los script siguiendo el orden

Entrar en el directorio `osrmt/v1_50/client/schema` ahí se encuentran los script específicos según la base de datos a utilizar.

El primer archivo a ejecutar sería `postgresql_create_user.sql` que permitirá establecer cual va a ser el usuario. De forma visual a través de la herramienta de gestión de base datos puede ser establecido este usuario.

El segundo archivo a ejecutar sería `postgresql_create_schema.sql` que mediante este se establece el sistema de tablas que tendrá la base de datos necesaria para el buen funcionamiento.

Por último crear la vista asociada `postgresql_create_views.sql`

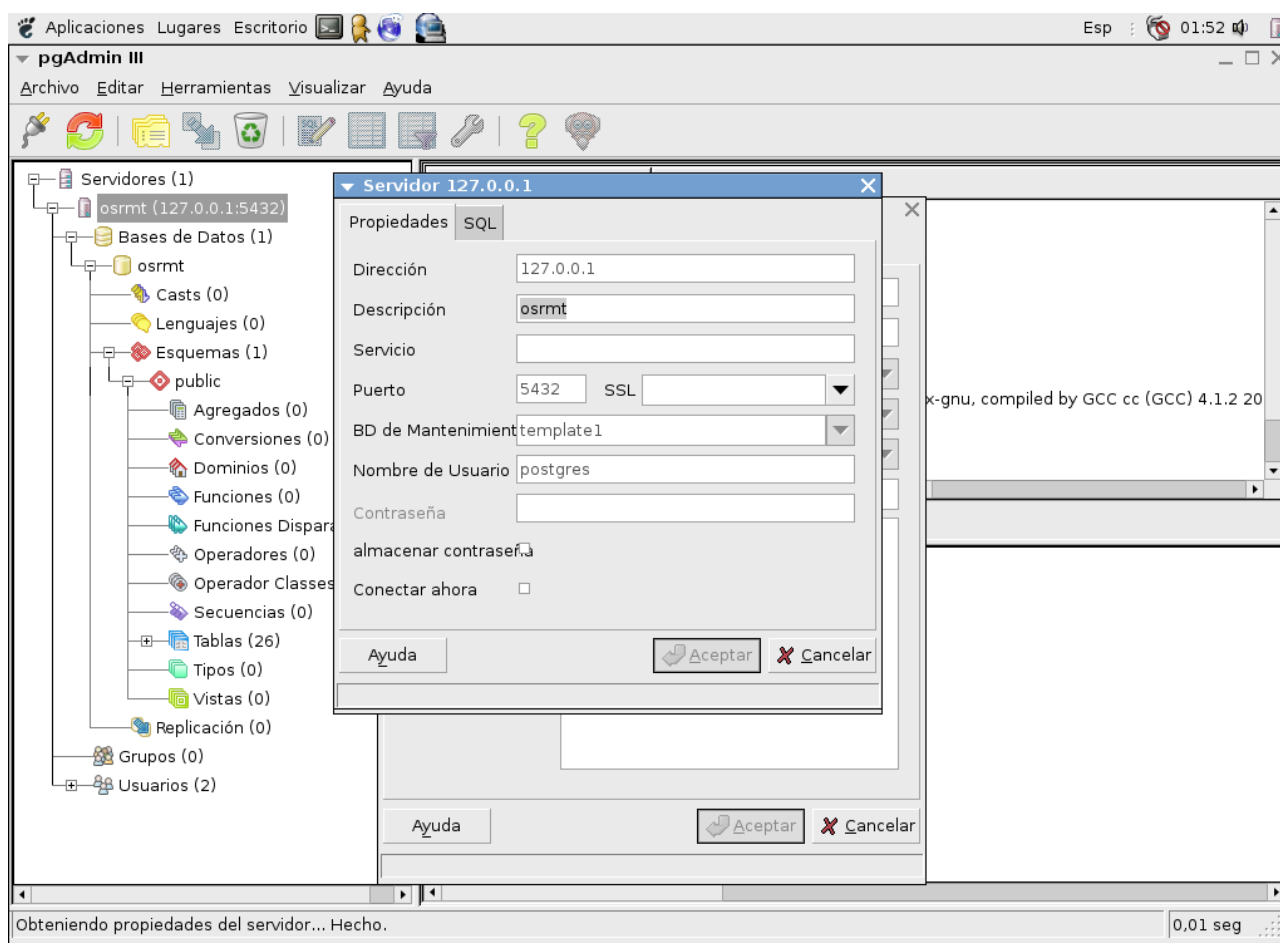


Figura 6

4. Importar contenido de la base datos

Se corrige `connection.xml` según los pasos siguientes. Para esto se ejecuta el archivo `clientupgrade.sh`

Seleccione la opción 4

Select configuration option 1, 2, 3 or 4

- 1) Define a new connection
- 2) Test the connection
- 3) Save the new connection
- 4) Initialize a new database
- 5) Upgrade 1.3 to 1.4 database
- 6) Migrate database contents
- 7) Export language file
- 8) Import language file
- 9) Exit

Enter option number [Exit]

Se confirma si quiere inicializar la base datos. Ver figura 7.

Enter option number [Exit]: 4

initializing database defined in connection.xml:

jdbc:mysql://localhost/osrmt?useUnicode=true&characterEncoding=UTF-8

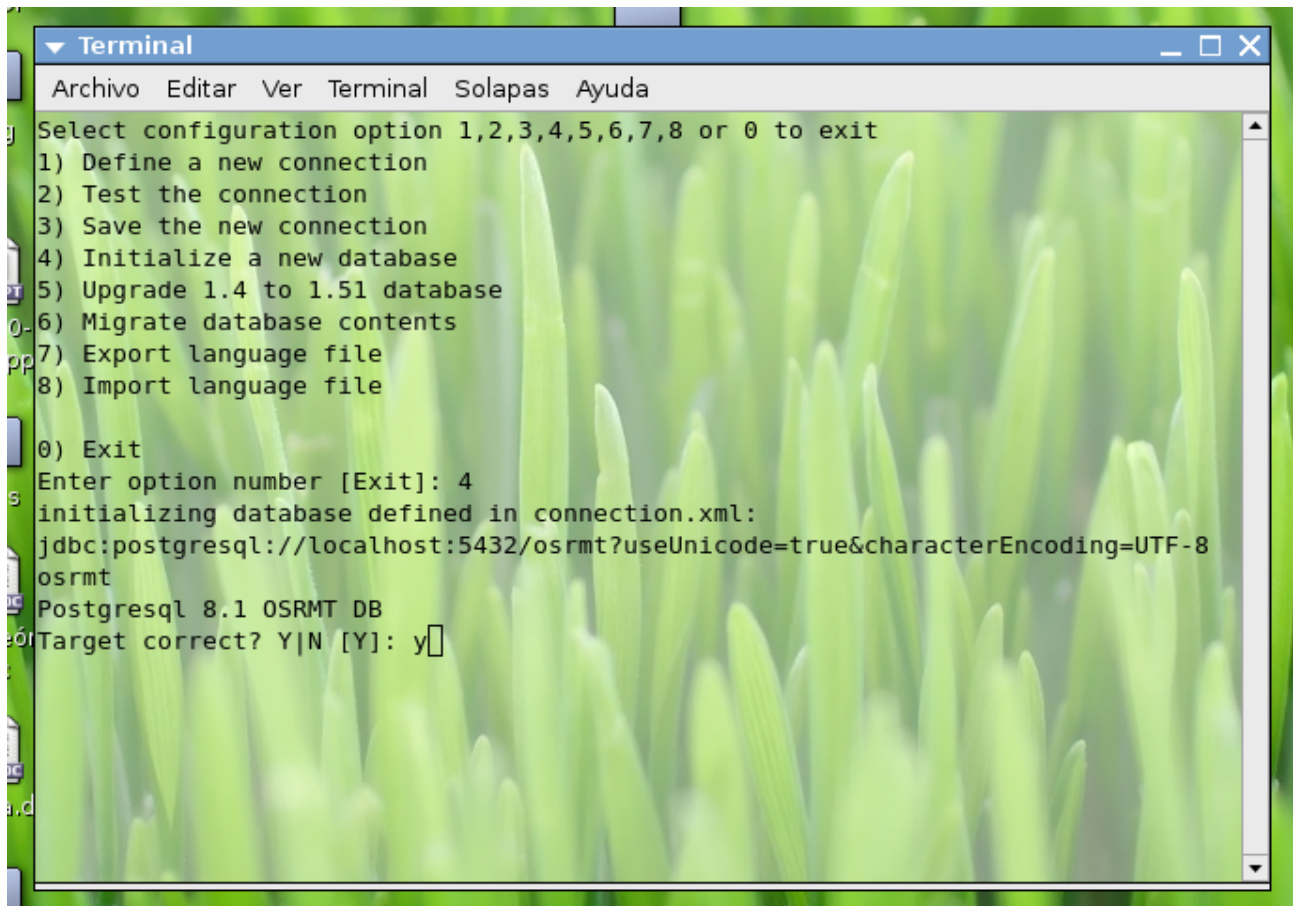
osrmt

pgsql

Target correct? Y|N [Y]: y

Empty schema located - initialize and populate schema? [Y]:

Solo presione y se actualizaran los datos, luego regresará al menú principal.



```
Terminal
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
Select configuration option 1,2,3,4,5,6,7,8 or 0 to exit
1) Define a new connection
2) Test the connection
3) Save the new connection
4) Initialize a new database
5) Upgrade 1.4 to 1.51 database
6) Migrate database contents
7) Export language file
8) Import language file
0) Exit
Enter option number [Exit]: 4
initializing database defined in connection.xml:
jdbc:postgresql://localhost:5432/osrmt?useUnicode=true&characterEncoding=UTF-8
osrmt
Postgresql 8.1 OSRMT DB
Target correct? Y|N [Y]: y
```

Figura 7

Solo resta iniciar el servidor "server/jboss-4.0.3/bin/run" y comprobar que no hay errores de permisos, ni puertos ocupados.

Ejecutar el cliente "client/run", usuario DEMO/demo, viene con 2 ejemplos vacios que puede utilizar para ver el funcionamiento.

Anexo 4: Manual de Usuario

Elaborado por:

Leidy Ramos González

Renier Socarras Pérez

Fecha de última revisión:

24 e abril 2008

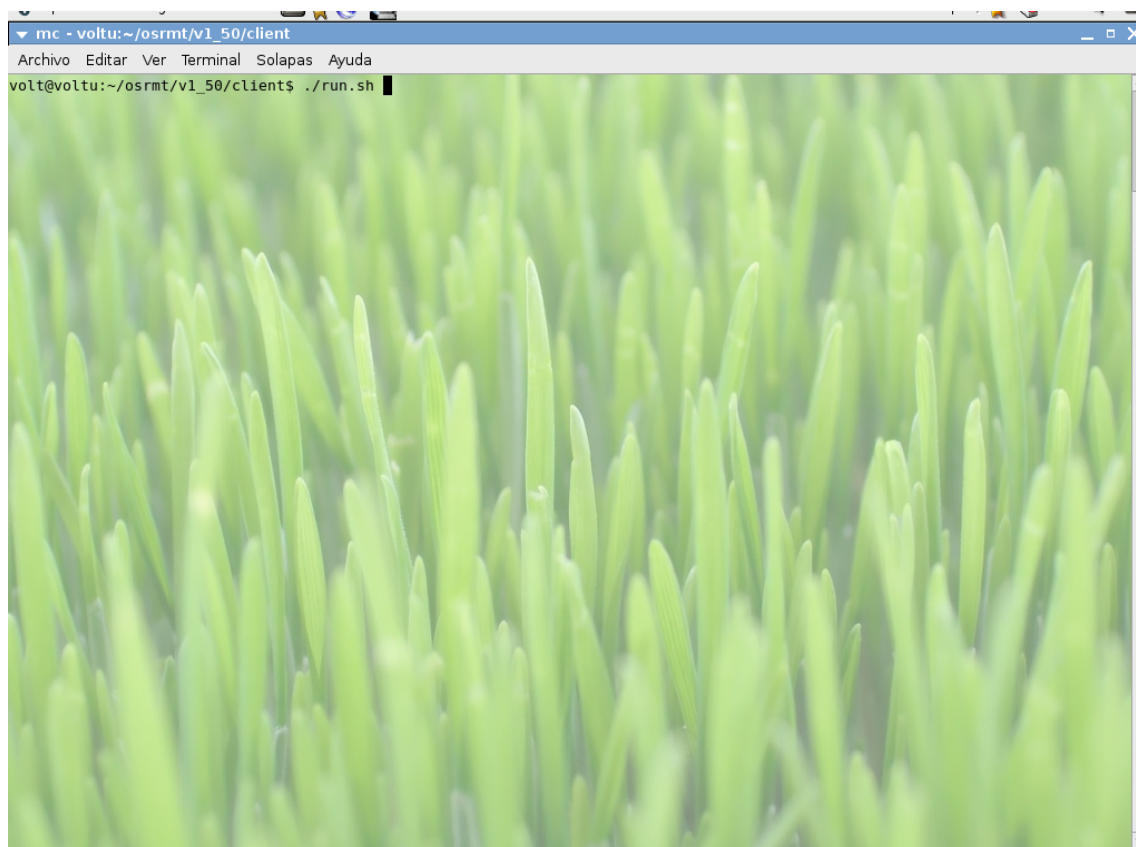
1. Buscar el inicio.

1.1 Inicializar la aplicación.

1) Iniciando con un acceso directo en el Escritorio.

2) Usando comandos en consola.

Moveirse por el directorio hasta encontrarse la carpeta client y ejecutar run.sh.



1.2 Entrar en la aplicación.

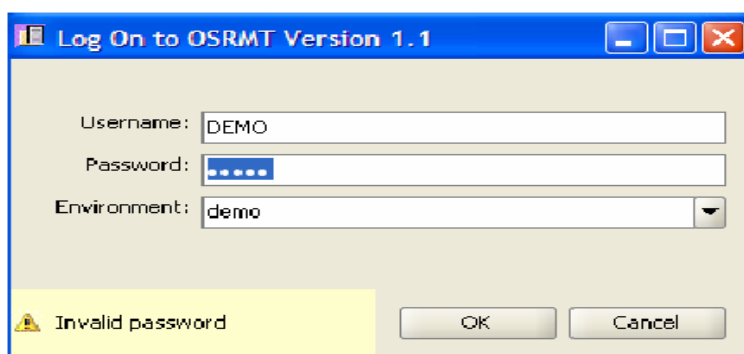
Para entrar debe contar con un usuario, contraseña, y un entorno.

El ambiente se genera de un archivo local de configuración.

El ambiente será de selección y en caso que solo **haya** un ambiente activo este se encontrará bloqueado (como por defecto).

Toma como usuario por defecto el último que fue registrado al iniciar la aplicación.

Mostrará un error en caso que el usuario o la contraseña sean incorrectos, y si quiere tener mas detalles técnicos del error surgido solo haga click en el cuadro del error.

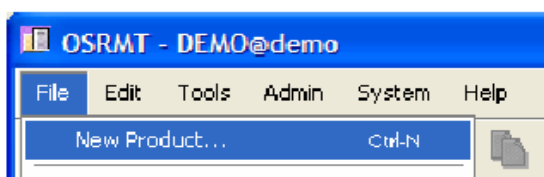


Luego presionar la tecla enter u OK si ya ha sido escrito el usuario y la contraseña correctamente.

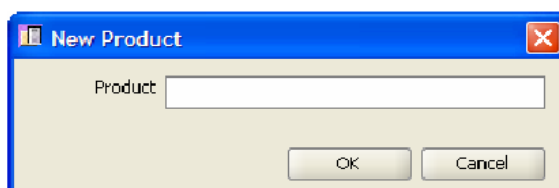
1.3 Crear un nuevo producto.

El primer paso es crear un nuevo producto, que no es más que asignarle un nombre al proyecto donde se guardarán todos los requisitos y los artefactos que se generarán.

File -> New Product...

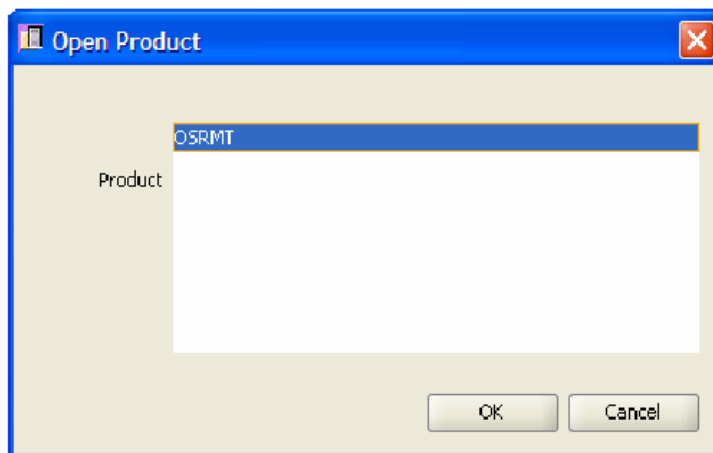


Entre el nombre que será mostrado posteriormente en la aplicación.



1.4 Abriendo el Proyecto.

Abrir en el menú File Open.



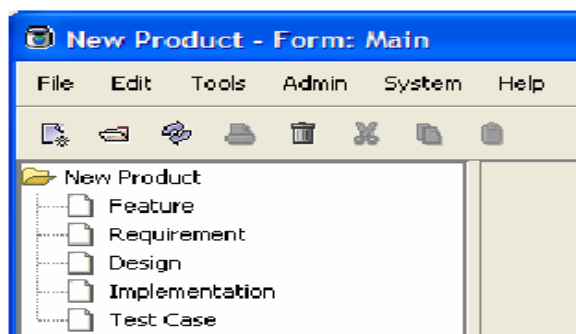
Selecciona el proyecto y haga click en OK.

2. Manejo de artefactos.

2.1 Árbol de artefactos.

Los artefactos pueden ser características del producto, requisitos, casos de usos de las aplicaciones, módulos, paquetes, o cualquier cosa que se necesite.

Cada tipo de artefacto tiene un nombre que puede ser modificado por el administrador. Lista que aparece abajo es la que está por defecto. Para requisitos específicos se pueden modificar los iconos de los artefactos.



Nuevo artefacto.

Para crear el artefacto seleccione el tipo y haga click en nuevo artefacto.



Complete el formulario (vea la entrada de datos, guarde los nuevos datos entrados y cierre el formulario).

Esto creará un artefacto hijo debajo del artefacto seleccionado.

Eliminar artefacto.

Para eliminar un artefacto que fue adicionado, seleccionamos el artefacto y con Edit->Delete o los botones de la barra puede ser eliminado.



Nota: Cuando se selecciona un artefacto para ser eliminado este si no es una rama del árbol será eliminado solo, si es un padre se eliminarán todos los descendientes y en el cuadro de confirmación saldrá la cantidad de artefactos que se eliminarán.

Mover un artefacto.

Se puede alterar la jerarquía de los artefactos arrastrando de un padre a otro. Lo que no se puede es arrastrar los tipos artefactos. Cuando se mueve un artefacto a otro tipo de artefacto solo crea una dependencia entre estos, sin mover el artefacto.

Ver la lista de artefactos.

Seleccione un artefacto en el árbol y en la derecha aparecerá todos los artefactos hijos.

Name	Priority	Status	Version	Description
Artifact Description	Must have	Completed	1.0	Shall be able to c
Custom numbering system	Medium	Submitted	2.0	Shall allow admin
Export artifacts	Must have	Submitted	2.0	Export artifacts
Manual artifact entry	Must have	Completed	1.0	Shall be able to c
Custom artifact attributes	Must have	Completed	1.0	Shall be able to c
Hierarchy of artifacts	Must have	Completed	1.0	Shall be able to c
Status report	Must have	Approved	1.0	Shall generate a

La forma de mostrar las columnas (la organización) puede ser cambiada por un administrador.

Para corregir un artefacto solo con darle doble click mostrará los datos y estos podrán ser corregidos.

Se pueden organizar los artefactos haciendo click encima de una columna, esta tomándola como criterio de organización.

Actualizar el árbol.

Se puede actualizar el árbol utilizando el botón correspondiente o File Refresh



Esto actualiza directamente de la base datos y se ve cualquier cambio que haya hecho otro usuario y en caso de que exista un error muestra el estado real del sistema.

2.2 Entrada de datos de los artefactos.

Cada tipo del artefacto (característica, requisito, diseño etc.) puede tener una configuración diferente de entrada de datos según lo establecido por el administrador.

Vaya completando todos los datos que usted necesite y presionando Next para seguir con todo el ciclo general.

Personalizar las formas.

El administrador puede agregar o eliminar pestañas.

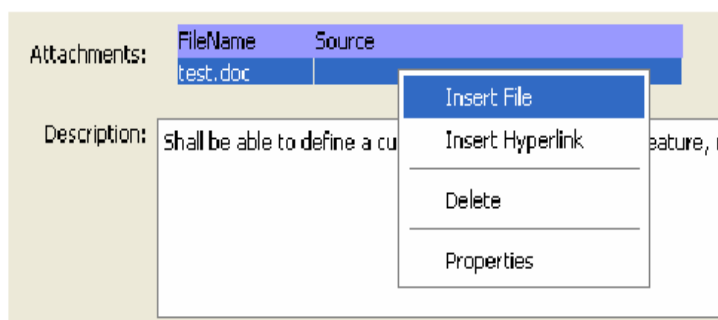
Los controles pueden tener valores por defecto o una lista de valores referenciados a una tabla.

El administrador puede escribir un JavaScript para personalizar el formato o inicializar un control.

Personalizar un control puede ser con el objetivo de guardar nuevos datos en la base de datos.

Adjuntar y unir documentos.

Haga click derecho en la lista de archivos adjuntos y saldrá un submenú con las opciones adicionales.



Seleccione un archivo y luego Insert y le mostrará un explorador para buscar un archivo local o de la red.

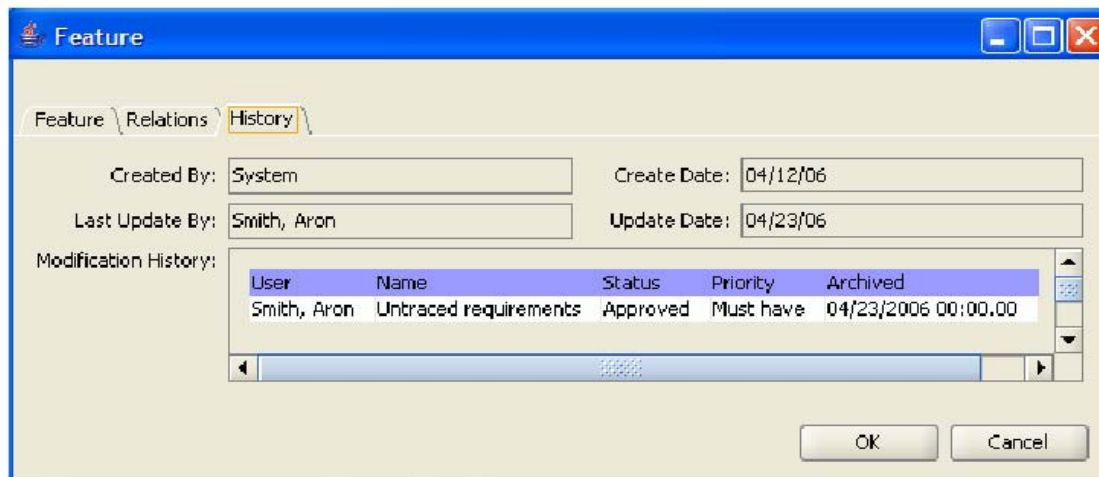
Seleccione un archivo y luego le haga click en Upload donde se subirá una copia al servidor y se guardará en un lugar establecido por el administrador.

Una vez que el archivo haya sido mostrado, el nombre y la descripción pueden ser cambiados usando el submenú.

Los hipervínculos pueden ser cambiados usando Insert Hyperlink.

Modificaciones del historial

Todas las modificaciones y creaciones de artefactos son guardadas en el historial, además de que se encuentra disponible la hora y los datos de cuando fue actualizado.



3.
Reporte
s y
Trazabili
dad.

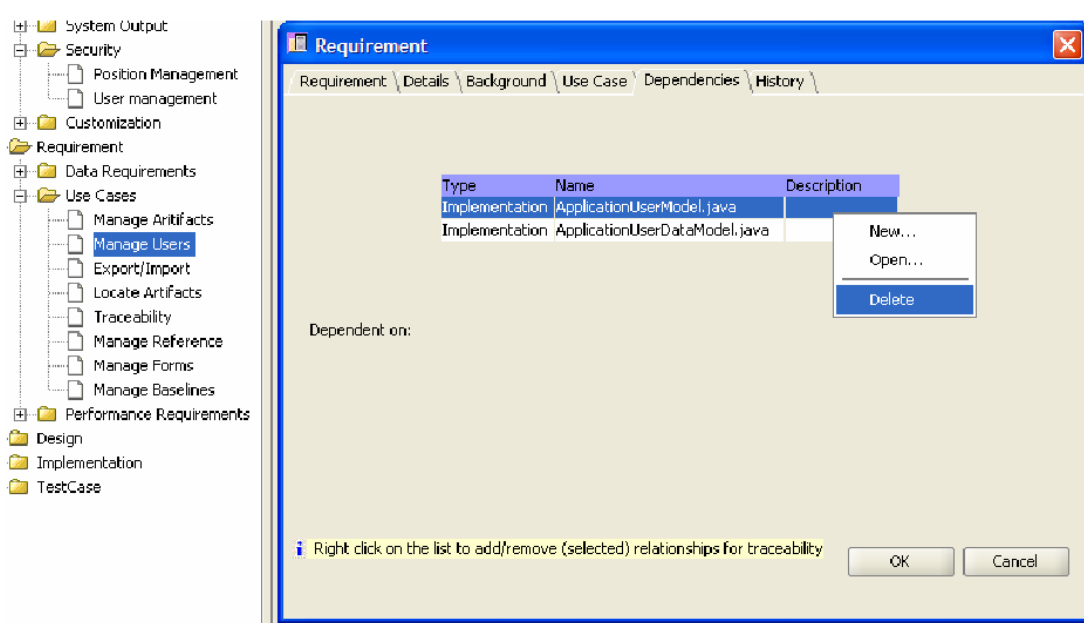
3.1 Los

artefactos hijos no son dependientes.

Si selecciona un artefacto ya existente y creamos un hijo en el árbol, esto automáticamente creará una relación entre los dos artefactos. Sin embargo esto no creará una dependencia.

Adicionando dependencias.

Mediante las dependencias se define la trazabilidad entre los artefactos. Para determinar nuevas dependencias solo haga click derecho encima de dependencias y selecciona New, y así se van agregando todos los deseados.

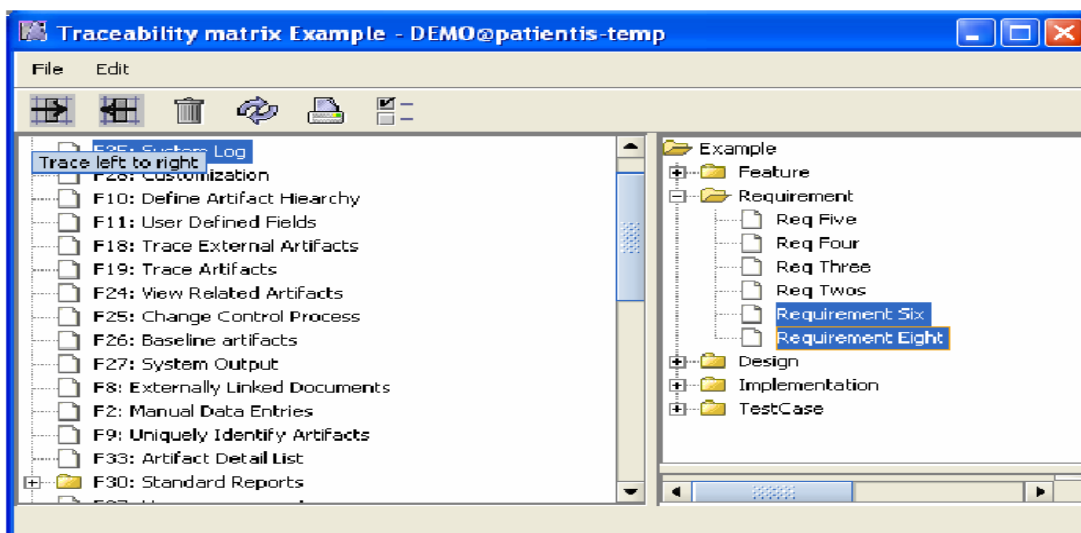


3.2 Árbol de trazabilidad.

Se puede abrir el árbol de trazabilidad usando los iconos de la barra o usando las opciones View - Trazabilidad.



El árbol permite seleccionar los artefactos y luego presionar el botón trazar.



Se pueden crear dependencias entre los artefactos que se encuentran en la izquierda y en la derecha, marcando los artefactos en la derecha y luego los de la izquierda o viceversa y luego dándole a los botones de crear trazabilidad.



Como se muestra en la figura se crean 2 dependencias:

System Log -- Requirement Six

System Log -- Requirement Eight

Solo queda que según la dirección del botón será donde se creará la dependencia hacia la derecha o hacia la izquierda.

Solo se debería crear dependencias de una en una sino crearía un número confuso de dependencias. Por ejemplo podría crear dependencia para una característica con múltiples requisitos.

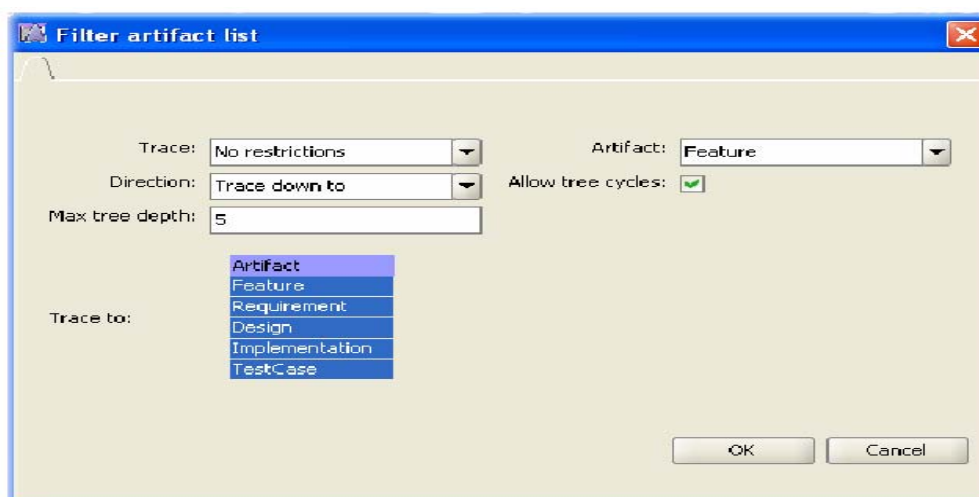
3.2.1 Opciones del árbol de trazabilidad.

La parte derecha presenta todo el árbol de artefactos.

Para cambiar la forma de ver el árbol de trazabilidad utilice el siguiente botón y corrija los criterios.



Aquí se encuentran los criterios que pueden ser modificados.

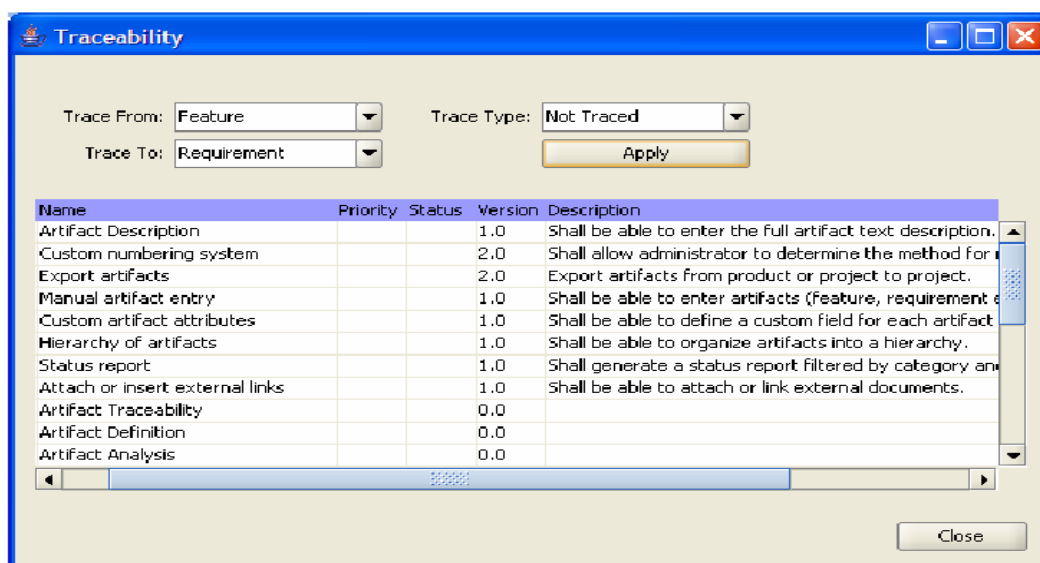


Campo	Descripción
Traza	Permite revisar la lista de artefactos.
	'Child dependency' está la lista de artefactos y los rastros existentes (dependencias) con los artefactos hijos. 'No Child dependencies' muestra todos los artefactos sin dependencias hacia algún hijo.

	<p>'Parent dependencies' restringe los artefactos de la lista a aquellos que sean padres y tengan alguna dependencia.</p> <p>'No Parent dependencies' padres sin ningún rastro de dependencia.</p>
Artefacto	Define el nivel superior de los artefactos en el árbol.
Dirección	<p>La trazabilidad hacia abajo mostrará la dependencia entre los artefactos hijos.</p> <p>La trazabilidad hacia arriba mostrará la dependencia con los artefactos de los niveles superiores.</p> <p>Si usted acaba de crear una dependencia pero no en esta dirección el árbol no se modificará.</p>
Ciclos en el árbol	Con esto permitirá que ningún artefacto agregado al árbol sea exhibido dos veces y se evitan los ciclos engañosos
Profundidad en el árbol	Siempre se debe evitar que las dependencias circulares ganen en profundidad.
Trazabilidad	El tipo de artefacto a trazar desde abajo o hasta el árbol.

3.3 Vistas de trazabilidad.

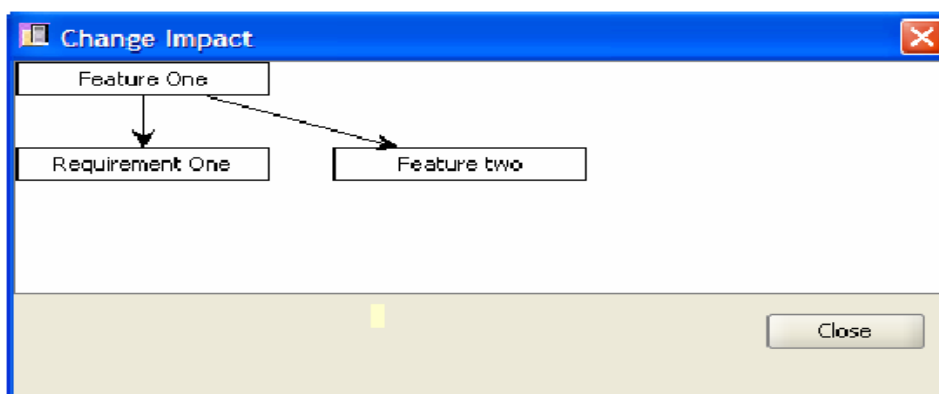
La matriz de trazabilidad, el trazado y el no trazado están disponibles en la forma de trazado.



Cambie el trazado y el tipo y haga click en Apply para obtener una nueva vista de trazado.

Impacto de trazabilidad.

Seleccione un artefacto del árbol y en el menú Tools ---Trace Impact, y verá un gráfico de los artefactos dependientes del seleccionado.



3.4 Reportes estándares.

Los reportes

se encuentran en el menú Tools Reports.

Selecciona el reporte de la lista y presione Next.

Finalmente selecciones las extensiones HTML o PDF para generar el reporte.

El programa ejecutará el archivo el navegador local HTML o el visor de PDF. Esto puede fallar en algunos sistemas operativos lo cual deberá comenzar el visor de PDF o el navegador y abrir el archivo generado. El nombre del archivo se exhibe en la barra de estado. Típicamente los archivos se crean en /temp/nombre.

3.5 Personalizar Reportes.

Guarda cualquier reporte que usted no desee perder.

3.5.1 Copiar Reportes.

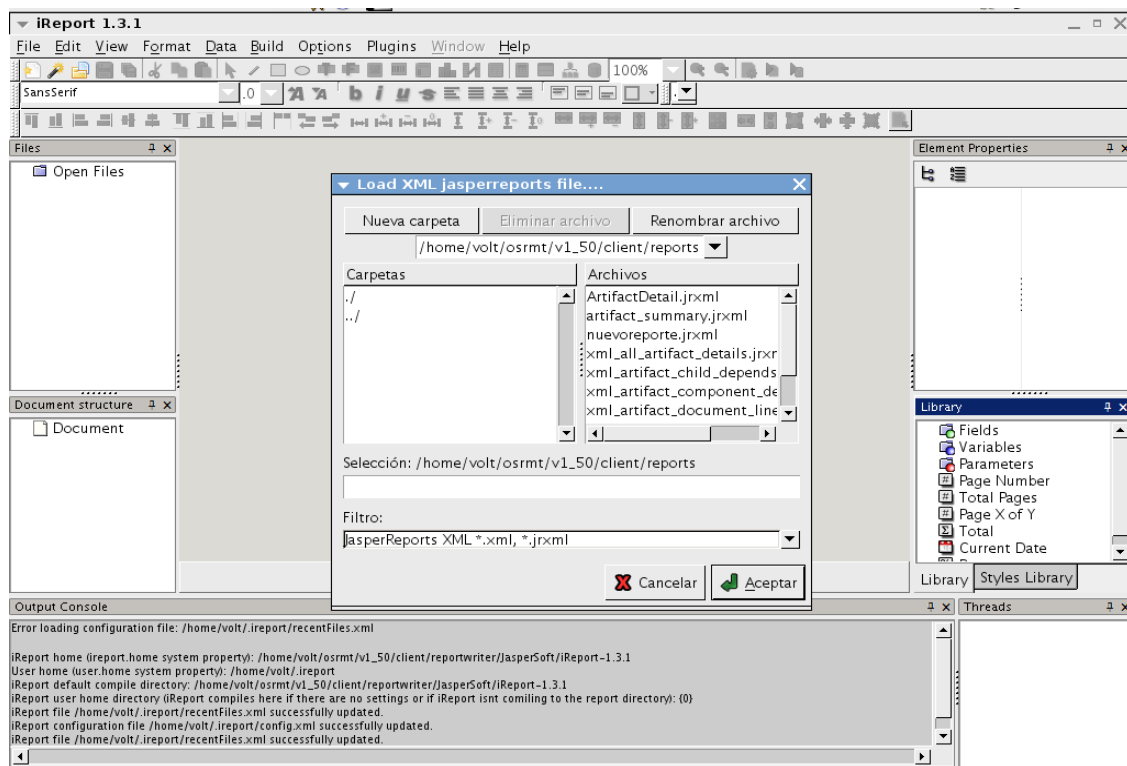
Navegar en el directorio de reportes \osrmt\v1_50\client\reports.

Copie un archivo existente a un nuevo nombre.

Copie ArtifactDetail.jrxml newreport.jrxml

Abra el informe escrito mediante un acceso directo o la forma de los reportes que se encuentra en el directorio.

osrmt\v1_50\client\reportwriter\JasperSoft/iReport-1.3.1\bin\startup.sh



3.5.2 Modific ar el Report

e

Al modificar el reporte y guardar los datos se modificará la versión anterior.

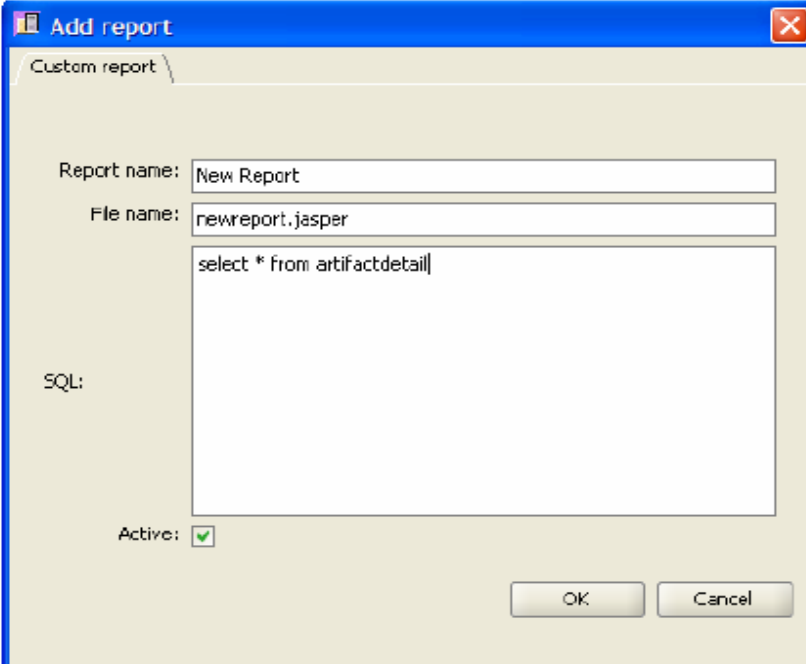
Usando de iReport en el menú Build – Compile para generar una versión compilada para osrmt reports\newreport.jasper

3.5.3 Adicionar un nuevo reporte

Dirijase al menú, en Tools –Reports y de click en Add Report.

Debe llenar todo el formulario asegurando que el nombre del archivo no se encuentre en el directorio.

El administrador puede crear una vista para un SQL complejo y compartirlo a través de múltiples formas.



Report name: New Report

File name: newreport.jasper

SQL: select * from artifactdetail

Active:

OK Cancel

4. Cliente para un navegador Web.

4.1 URL

La URL la provee el administrador del sistema.

<http://localhost:8080/osrmt>



open source REQUIREMENTS MANAGEMENT TOOL

Username: DEMO

Password:

Submit Reset

El administrador tiene cierto control para modificar los colores, el tamaño de la fuente y quien será el administrador para determinados requisitos.

4.2 Navegación.

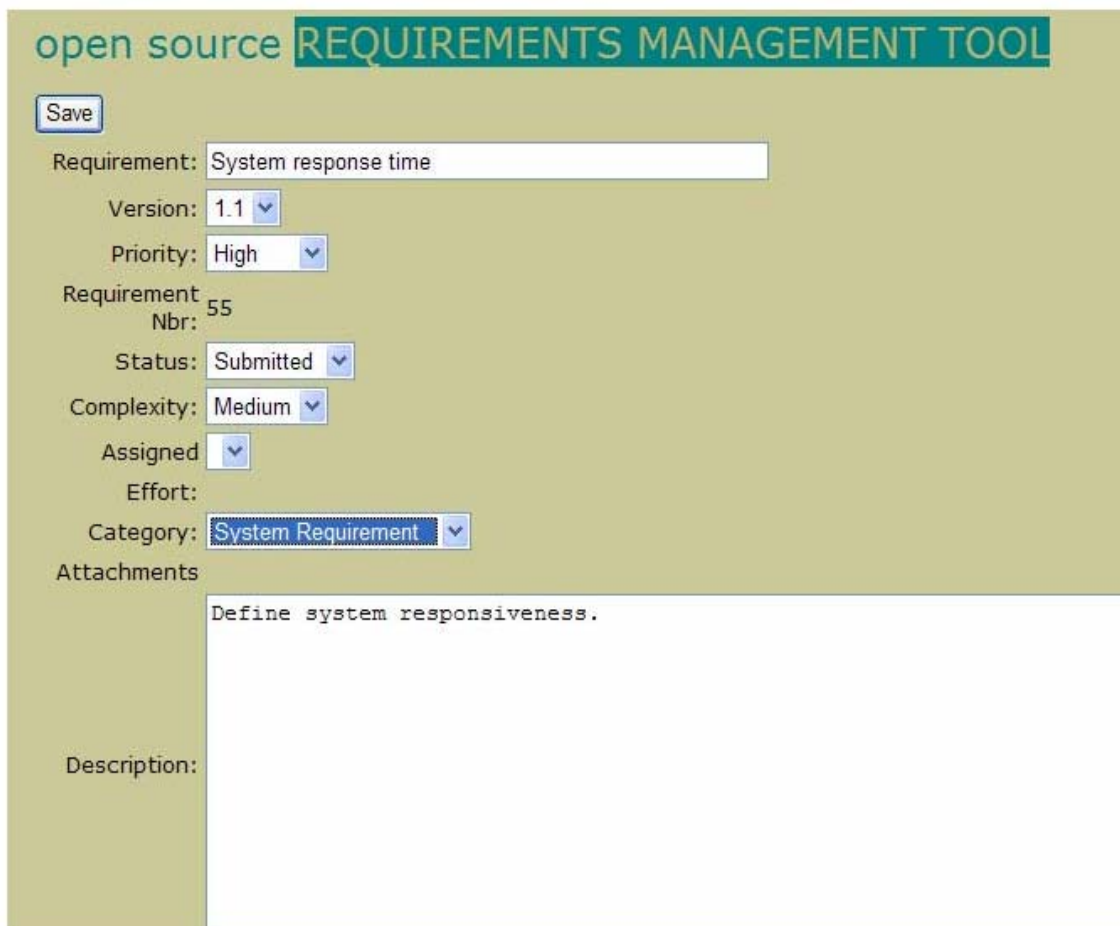
Haga click en los link para navegar por el árbol y click en el nombre del producto para retornar a la raíz.

4.3 Seguridad.

El administrador del sistema puede modificar la seguridad para los tipos de artefactos y/o solamente darle permisos de lectura a los tipos de artefactos.



Los campos específicos se pueden leer solamente u ocultarlos para que no sean vistos por la web del cliente.



The screenshot shows the 'open source REQUIREMENTS MANAGEMENT TOOL' interface with a 'Save' button and a form for editing a requirement. The form fields are:

- Requirement: System response time
- Version: 1.1
- Priority: High
- Requirement Nbr: 55
- Status: Submitted
- Complexity: Medium
- Assigned: [Dropdown]
- Effort: [Empty]
- Category: System Requirement

Below the form is an 'Attachments' section with a text area containing the text: 'Define system responsiveness.'

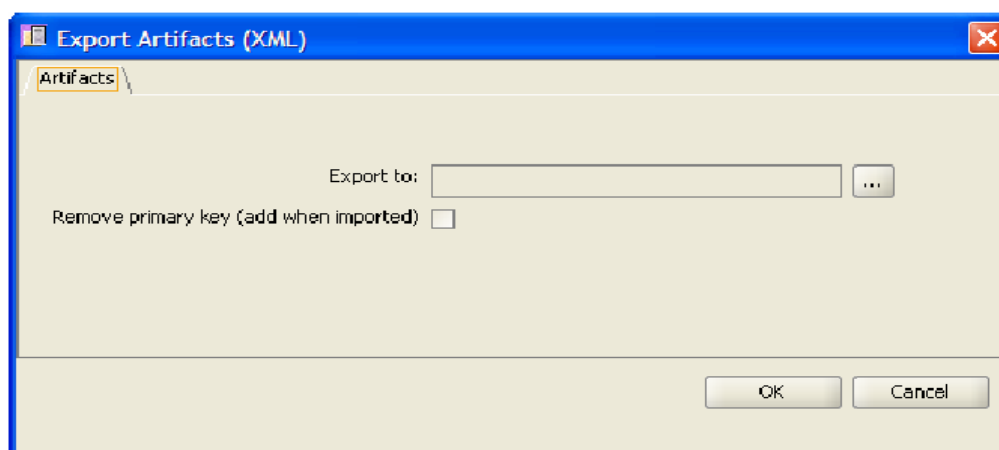
At the bottom, there is a 'Description:' label followed by a large empty text area.

5. Exportar/Importar.

5.1 Exportando artefactos.

Busque los requisitos en el árbol, filtre la lista y/o busque los artefactos creados y personalice la lista de artefactos.

File – Export puede también exportar solo los artefactos de la lista filtrada.



Seleccione la opción para quitar la llave primaria a

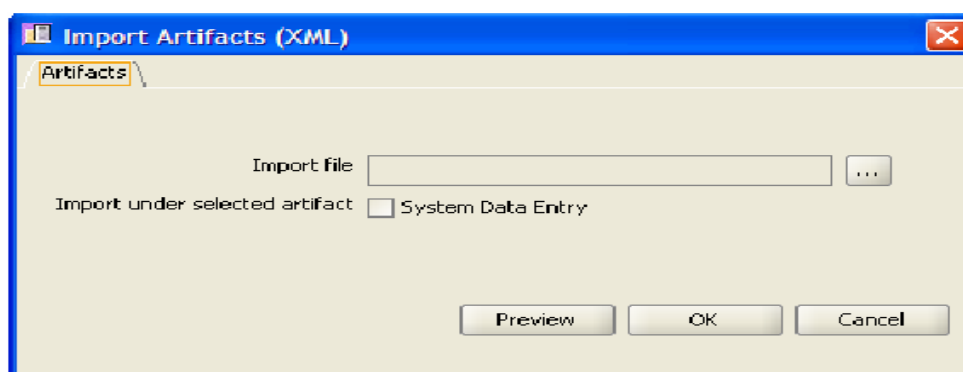
la importación en una nueva base de datos.

Exportar para CSV.

La exportación a CSV no se apoya directamente, aunque puede haber utilidades libres en Internet para convertir XML al CSV¹.

5.2 Importar artefactos.

Puede navegar a través de una rama del árbol e importar artefactos específicos para esa rama.



1
repre

documento sencillo para

Para asegurar que los artefactos tengan la misma ubicación el archivo que se va a importar debe tener el mismo formato que el que fue exportado.

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0_04" class="java.beans.XMLDecoder">
  <object class="com.osrmt.modellibrary.reqmanager.ArtifactModel">
    <void property="artifactName">
      <string>System Data Entry</string>
    </void>
    <void property="artifactId">
      <int>0</int>
    </void>
    <void property="artifactNbr">
      <int>0</int>
    </void>
    <void property="artifactRefDisplay">
      <string>Feature</string>
    </void>
    <void property="artifactRefId">
      <int>129</int>
    </void>
    <void property="categoryRefDisplay">
      <string>Do not print</string>
    </void>
    <void property="categoryRefId">
      <int>1567413</int>
```

```
</void>
<void property="createUserName">
  <null/>
</void>
<void property="priorityRefDisplay">
  <string>Must have</string>
</void>
<void property="priorityRefId">
  <int>14</int>
</void>
<void property="productRefDisplay">
  <string>OSRMT</string>
</void>
<void property="productRefId">
  <int>1000001</int>
</void>
<void property="recordTypeRefDisplay">
  <string>User Record</string>
</void>
<void property="recordTypeRefId">
  <int>323</int>
</void>
<void property="statusRefDisplay">
  <string>Completed</string>
</void>
<void property="statusRefId">
  <int>19</int>
</void>
<void property="systemAssignedVersionNbr">
  <int>1</int>
</void>
<void property="updateCount">
  <int>8</int>
```

```
</void>
<void property="updateUserId">
  <int>1</int>
</void>
<void property="updateUserName">
  <string>Demo, Osrmt</string>
</void>
<void property="versionRefDisplay">
  <string>1.0</string>
</void>
<void property="versionRefId">
  <int>472</int>
</void>
</object>
</java>
```

6. Apéndice.

6.1 Forma de registro del sistema.

La lista de errores de compilación, información, advertencia y los mensajes de error están disponibles en el menú.

Help -- System Log

6.2 Error en el comienzo de la aplicación.

Dependencias.

Deben existir las condiciones siguientes para que comience la aplicación.

1. Tener la instalación de la versión de Java 1.5 o mayor, preferiblemente la 5.0.
2. Deben estar todos los archivos requeridos .jar antes de ejecutar run.sh.
3. Debe estar lista y conectada la base de datos.
4. La base de datos debe haber sido inicializada.
5. Cualquier error será mostrado en una caja de texto grande y contiene el rastro de cual fue el error.
6. El servidor OSRMT debe estar corriendo.

7. Para que el cliente se conecte al host el servidor debe estar corriendo por el puerto específico.
8. De connection.xml es J2ee el archivo de configuración.

6.3 Conectarse al servidor.

El host y el puerto por el cual el cliente debe conectarse se especifica en el archivo:

```
jndi.properties.OSRMTv05\client\dd\clienbt>type jndi.properties
java.naming.factory.initial=org.jboss.security.jndi.LoginInitialContextFactory
#org.jnp.interfaces.NamingContextFactory
java.naming.provider.url=jnp://127.0.0.1:1099
java.naming.factory.url.pkgs=org.jboss.naming.client
java.naming.security.principal=200
java.naming.security.credentials=j2ee
java.naming.security.protocol=client-login
j2ee.clientName=osrmt-client
```

Pasos y cosas a tener en cuenta:

1. La dirección IP se destaca en amarillo y el puerto en rojo.
2. Asegúrese que puede usar ese ip y ese puerto.
3. Si su conexión no se realiza puede ser debido a estos errores.
 1. El servidor OSRMT no se encuentra corriendo.
 2. El servidor OSRMT no se encuentra corriendo por ese IP y ese puerto.
 3. La red local o el servidor se encuentran restringidos para IP específicos o subredes.