

Universidad de las Ciencias Informáticas

Facultad 4



Título: “Análisis y diseño de la herramienta SeReq para el seguimiento de los requisitos en el proyecto ERP Cuba”

TRABAJO DE DIPLOMA PARA OPTAR POR EL TITULO DE
INGENIERO EN CIENCIAS INFORMATICAS

Autor: Sandy Machado Scull

Tutores: Ing. Yoan Arlet Carrascoso Puebla
Ing. Yanet Vega Miniet

Ciudad de La Habana

Mayo de 2009

AGRADECIMIENTOS

A mis padres y familia por el apoyo y la confianza depositados en mí.

A mis tutores porque sus consejos y ayuda fueron cruciales para desarrollar este trabajo.

A mis verdaderos amigos por su incondicionalidad mostrada en todo momento.

A todas las personas que de una forma u otra me han ayudado en esta etapa de la vida.

DEDICATORIA

*A mi madre,
A mi hermana.*

RESUMEN

El proceso de seguimiento de los requisitos de software representa un amplio beneficio en el desarrollo de aplicaciones pues permite mantener una traza directa de los requisitos identificados con el resto de los artefactos del sistema. En el presente trabajo se aborda la realización del análisis y diseño que servirá de base para la implementación de la herramienta SeReq, la cual automatizará el seguimiento de los requisitos en el proyecto ERP Cuba.

Se realizó un estudio de la actualidad llegándose a la conclusión de que las herramientas existentes para el seguimiento de los requisitos no suplen las necesidades de los analistas del proyecto ERP Cuba. Se estudiaron además, metodologías de desarrollo, herramientas y lenguaje para el modelado del sistema, así como consideraciones para la arquitectura del mismo.

En el trabajo se explicaron las principales características del sistema propuesto y se muestran los artefactos que posibilitan el análisis y diseño de la herramienta. Finalmente, se aplicaron métricas a estos artefactos para validar la calidad de los mismos, con lo cual se deja listo el camino para las futuras actividades de implementación.

Palabras clave

Seguimiento de los requisitos, traza, herramienta SeReq, proyecto ERP Cuba, análisis y diseño, métricas.

ÍNDICE

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPITULO 1: Fundamentación Teórica.....	5
Introducción.....	5
1.1 Ingeniería de requisitos.....	5
1.2 Gestión de requisitos.....	7
1.3 Seguimiento de los requisitos en el proyecto ERP Cuba.....	9
1.4 Herramientas para la gestión de requisitos.....	10
1.4.1 IBM Rational RequisitePro.....	11
1.4.2 IRqA 3.0.....	12
1.4.3 Borland® CaliberRM™.....	12
1.4.4 OSMRT.....	13
1.5 Modelo de desarrollo de software.....	13
1.6 Herramienta y lenguaje para el modelado.....	14
1.6.1 Herramientas CASE.....	14
1.6.1.1 Visual Paradigm.....	14
1.6.2 Lenguaje Unificado de Modelado.....	15
1.7 Especificaciones de la arquitectura.....	15
1.7.1 Arquitectura Cliente/Servidor.....	15
1.7.2 Patrón Modelo-Vista-Controlador (MVC).....	16

1.7.3 Lenguaje de programación.....	17
1.7.4 Sistema gestor de bases de datos.	18
1.7.6 Marcos de trabajo.	18
1.7.6.1 Zend Framework.	19
1.7.6.2 Zend_Ext Framework.	19
1.7.6.3 Doctrine Framework.	19
1.7.6.4 ExtJS Framework.	20
1.7.6.5 UCID Framework.	20
Conclusiones parciales.....	20
CAPITULO 2: Descripción de la solución propuesta.	22
Introducción.....	22
2.1 Modelo de dominio.	22
2.1.1 Modelo conceptual.	23
2.1.2 Glosario de términos.	24
2.2 Requisitos.....	25
2.2.1 Requisitos funcionales.	25
2.2.1.1 Modelo de Casos de Uso del Sistema.	25
2.2.1.2 Justificación de los Actores del Sistema.	26
2.2.1.3 Especificación de los Casos de Uso del Sistema.....	27
2.2.2 Requisitos no funcionales.	64
2.3 Modelo de diseño.	66
2.3.1 Fundamentación y aplicación de los patrones utilizados para el diseño.	66
2.3.2 Diagrama de componentes de la herramienta.....	67

2.3.3 Características comunes observadas en los diseños de clases.	68
2.3.4 Diagrama de clases de diseño.	70
2.3.5 Modelo de datos.	71
2.3.6 Seguridad de la herramienta propuesta.	72
2.3.7 Concepción general de la ayuda.	73
2.3.8 Tratamiento de errores.	73
Conclusiones parciales.	74
CAPITULO 3: Análisis de los resultados.	75
Introducción.	75
Análisis de la Especificación de requisitos.	76
3.1 Métricas para la Especificación de Requisitos.	76
3.1.1 Aplicación de la métrica.	76
3.1.2 Revisiones realizadas a especificaciones de requisitos.	77
3.1.3 Resultado del análisis.	79
3.2 Métricas para el Modelo de diseño orientado a objetos.	79
3.2.1 Métricas orientadas a Clases.	79
3.2.2 Métricas propuestas por Chidamber y Kemerer.	80
3.2.2.1 Aplicación de las métricas.	81
3.2.3 Métricas propuestas por Lorenz y Kidd.	82
3.2.3.1 Aplicación de las métricas.	83
Conclusiones parciales.	83
CONCLUSIONES GENERALES.	85
RECOMENDACIONES.	86

BIBLIOGRAFÍA REFERENCIADA	87
BIBLIOGRAFÍA CONSULTADA.....	90
ANEXOS.....	91
Anexo 1: Plantilla para la especificación de requisitos.....	91
Anexo 2: Prototipos de interfaz gráfica de usuario.....	93
Anexo 3: Diagramas de clases de diseño.....	106
Anexo 4: Descripción de las tablas del modelo de datos.	111

INTRODUCCIÓN

En la actualidad, un equipo de desarrollo de software compuesto por estudiantes y profesionales de la Universidad de las Ciencias Informáticas (UCI), en conjunto con especialistas de Organismos de la Administración Central del Estado (OACE), se encuentran inmersos en la creación de un Software Integral de Gestión que responda a la actual necesidad económica que presenta el país. Para el desarrollo de dicho sistema, cuyo nombre es Cedrux, este ha sido dividido en varios subsistemas, en cada uno de los cuales se capturan requisitos para su posterior implementación. Dado el alto nivel de conexión existente entre los procesos de una empresa, se puede entender que análogamente, los requisitos capturados en los diferentes módulos de cada uno de los subsistemas estarán relacionados.

Durante el flujo de trabajo de Levantamiento de Requisitos, realizado en la primera iteración del proyecto, los analistas comprobaron que los requisitos están interrelacionados y que pueden cambiar con mucha frecuencia. Por esta razón, se hace necesario conocer la repercusión que puede tener la realización de un cambio en un requisito en el resto de los artefactos, así como el impacto del mismo en la implementación. Teniendo en cuenta que el equipo de desarrollo es muy numeroso, con lo que la trasmisión de información se hace difícil, la manera óptima para realizar esta actividad es a través de una herramienta automatizada que permita mantener el seguimiento a cada uno de los requisitos elicitados.

Luego de estudiar algunas de las herramientas que pudieran resolver esta situación, se pudo verificar que estas no satisfacen las necesidades reales de los analistas del proyecto ERP Cuba para el seguimiento de los requisitos, puesto que no ofrecen un adecuado versionado de los requisitos, algunas presentan serios problemas de usabilidad dado que los usuarios a menudo no encuentran el flujo de las acciones mientras trabajan con la herramienta, además de que el marco de reportes sobre los requisitos aún es limitado y no ofrecen todas las funcionalidades que los analistas del proyecto necesitan para el seguimiento de los requisitos. Una explicación más exhaustiva del estudio se puede encontrar en la sección [1.4 Herramientas para la gestión de requisitos](#) del Capítulo 1 del presente trabajo de diploma.

Frente a las limitaciones que ofrecen las herramientas estudiadas, se hace necesario crear una nueva, basada en software libre, que sea capaz de facilitar el seguimiento de los requisitos que son capturados como parte del ciclo de desarrollo de software; así como permitir conocer el impacto, que sobre otros

artefactos tiene, la realización de un cambio en uno de los requisitos, y brindar dicha información inmediatamente a las personas afectadas. Por este motivo, se plantea la necesidad de realizar el análisis y el diseño, que servirá de base para la posterior implementación, de una nueva herramienta que supla las necesidades reales de los analistas.

El presente trabajo de diploma pretende dar solución a la **situación problemática** anteriormente expuesta, para lo cual se plantea el siguiente **problema de investigación**: ¿Cómo facilitar, en el proyecto ERP Cuba, el seguimiento de los requisitos de software y brindar la información al respecto al equipo de desarrollo?

Consecuentemente, el **objeto de estudio** es la Gestión de los Requisitos de software. Luego, el **campo de acción** en el cual se inserta el presente trabajo es el seguimiento de los requisitos mediante el uso de sistemas informáticos.

A partir del problema mencionado anteriormente el **objetivo general** es realizar el análisis y diseño de una herramienta que permita mantener el seguimiento a los requisitos de software en el proyecto ERP Cuba.

Con el fin de dar solución al problema planteado y alcanzar el objetivo trazado se proponen las siguientes **tareas de investigación**:

- Realizar el marco teórico de la investigación.
- Realizar el análisis y diseño de la herramienta.
- Evaluar la especificación de requisitos y el modelo de diseño de la herramienta.

Para completar dichas tareas quedan trazadas las siguientes **acciones**:

- Realizar un estudio del estado del arte de la ingeniería de requisitos, así como los sistemas que existen para la gestión de los mismos, tanto propietarios como libres, para analizar sus características e inconvenientes.
- Estudiar el fenómeno y comprender el problema.

- Estudiar y fundamentar la elección de la metodología de software usada para realizar el análisis y el diseño.
- Estudiar las distintas herramientas, técnicas y buenas prácticas que existen para modelar el análisis y el diseño.
- Realizar el análisis y diseño del sistema a partir de las herramientas, las técnicas y buenas prácticas estudiadas.
- Definir y aplicar métricas para evaluar los resultados.
- Redactar el informe final de la investigación.

El documento estará estructurado de la siguiente forma atendiendo al flujo lógico de las actividades desarrolladas.

En el Capítulo 1 se realizará el estado del arte de la ingeniería de requisitos, enfocándose a la gestión de requisitos y más particularmente al seguimiento de los requisitos. De este último proceso se analizará el modo en que se está realizando en el proyecto ERP Cuba y se revisarán las herramientas existentes que pudieran adoptarse para realizarlo de manera automatizada. Seguidamente, se estudiarán diferentes metodologías de desarrollo con el fin de adoptar una de ellas, la cual será la rectora en el proceso de desarrollo de la herramienta que propone el presente trabajo de diploma. Se definirá la herramienta y el lenguaje para el modelado, así con las especificaciones de la arquitectura en la se basará el diseño de la aplicación.

En el Capítulo 2 se expondrá la solución propuesta, la cual estará integrada por el modelo de dominio de la herramienta, sustentado en un modelo conceptual y el glosario de términos; el diagrama de casos de uso del sistema así como la descripción de los requerimientos funcionales y no funcionales. Conjuntamente se expone el modelo de diseño que incluye el diagrama de componentes de la herramienta, las características comunes observadas en los diseños de clases, los diagramas de clases del diseño y el modelo de datos. Además, se explicarán los aspectos relacionados con la seguridad, el tratamiento de errores y la ayuda del sistema.

En el Capítulo 3 se realizará la validación de los artefactos que componen el trabajo de diploma a través de técnicas de validación de requisitos y la aplicación de métricas orientadas a la calidad de las especificaciones de los requisitos y al modelo de diseño.

CAPITULO 1: Fundamentación Teórica.

Introducción

En este capítulo se presenta un estudio sobre el estado actual del seguimiento a los requisitos de software. Se definen los principales conceptos relacionados con la ingeniería de requisitos y la gestión de requisitos, además de indagar sobre herramientas que actualmente permiten mantener el seguimiento de los requisitos. Se estudian algunas metodologías que guían los procesos de desarrollo de software con el objetivo de definir la rectora del proceso. Se seleccionan la herramienta y el lenguaje para el modelado. Finalmente, se definen algunos aspectos de la arquitectura sobre la que se basará el desarrollo de la herramienta que propone el presente trabajo de diploma.

1.1 Ingeniería de requisitos.

En la actualidad, la producción de software se está convirtiendo en una industria con amplio progreso en Cuba. En el éxito del proceso de desarrollo de un sistema informático incide totalmente el trabajo que se realice durante la elicitación de los requisitos de dicho sistema. Sobresale entonces un desafío observable en cada equipo de ingenieros que se encuentra inmerso en esta primera etapa de desarrollo de software. Se trata del modo de asegurar que el sistema que se está describiendo a partir de especificaciones, supla las expectativas de los clientes, completando las necesidades de estos. La certeza sobre este aspecto es difícil de lograr; no obstante, ofrece una profunda garantía si un sólido proceso de ingeniería de requisitos se realiza.

La IEEE (*Institute of Electrical and Electronics Engineers*) ha definido el concepto de requisito como:

- i) Una condición o capacidad que necesita el usuario para resolver algún problema o alcanzar un objetivo.*
- ii) Condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.*
- iii) Una representación documental de una condición o capacidad como en i o en ii.” (IEEE, 1998)*

Esta definición enmarca el objetivo de lo que algunos autores han referido acerca de la ingeniería de requisitos.

Según Roger Pressman la ingeniería de requisitos se puede definir como *“un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software.”* (Pressman, 2005)

Para Thayer y Dorfman *“la ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando 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.”* (Thayer, et al., 1997)

En tanto, Donald Reifer ha plasmado que *“la ingeniería de requisitos es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo de un sistema que satisfaga las necesidades del usuario.”* (Reifer, 1994)

A partir de las citas anteriores se puede resumir que la ingeniería de requisitos es un proceso centrado en la búsqueda de las necesidades del usuario, con el fin de realizar una especificación de estas que conduzca al desarrollo de un sistema informático que soporte los procesos de negocio en cuestión. Vale destacar que cobra suma importancia para el éxito o fracaso de numerosos proyectos y su mala gestión tiene una gran incidencia en relación con el desbordamiento de costes o el incumplimiento de plazos de finalización.

Lo expuesto revela que el proceso de ingeniería de requisitos no es trivial, sino que es amplio y necesita ser organizado en otros subprocesos que tributen al resultado que se persigue. En relación con esto, Ian Sommerville y Pete Sawyer han definido cinco pasos distintos, los cuales son: Identificación de Requisitos, Análisis de Requisitos y Negociación, Especificación de Requisitos, Validación de Requisitos y Gestión de Requisitos. (Sommerville, et al., 1997)

La identificación de requisitos consiste en interactuar con el cliente, los usuarios y los que están involucrados en los objetivos del sistema, para ayudar a solucionar los problemas de alcance, de comprensión y de volatilidad de los requisitos. Es una comunicación iterativa con los clientes y los

interesados del proyecto, para descubrir, definir, refinar y registrar una representación precisa de los requisitos del futuro sistema.

Durante el análisis y negociación de los requisitos, estos son agrupados por categorías y se organizan en subconjuntos; al mismo tiempo, son examinados en su consistencia, completitud y ambigüedad, y se clasifican en base a las necesidades de los clientes y usuarios. (Sommerville, et al., 1997)

Con la información conseguida hasta el momento es posible realizar la especificación de requisitos, que a menudo se recomienda sea realizada en una plantilla estándar. La especificación constituirá un convenio formal con el cliente y un compromiso para el equipo de desarrollo de construir exactamente lo que ha solicitado el cliente.

El siguiente paso del proceso es realizar la validación de requisitos, que mediante revisiones técnicas formales, se examinan las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias y sin omisiones. Tras este proceso de verificación se comprueba que los errores detectados han sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proyecto.

1.2 Gestión de requisitos.

Es bien conocido que los requisitos cambian con frecuencia, lo cual se manifiesta a lo largo del desarrollo del software. Los cambios pueden estar dados por avances tecnológicos, modificaciones en leyes o regulaciones, cambios en las estrategias o ambiente del negocio, o bien porque los usuarios cambiaron su forma de pensar. Adicionalmente, se tiene que nuevos requisitos pueden surgir a medida que avanza el desarrollo del sistema o como efecto de la inclusión de otros requisitos. Cada vez que los requisitos sean modificados deben actualizarse los atributos de los requisitos y sus dependencias.

Por estas razones, el proceso de gestión de los requisitos adquiere una importancia vital dentro de la ingeniería de requisitos, puesto que permite a los analistas del sistema supervisar y dar seguimiento a los cambios que se produzcan en ellos y de esta manera controlar el peligroso impacto que esto pueda causar sobre el sistema.

Roger Pressman describe la gestión de requisitos como *“un conjunto de actividades que ayudan al equipo de desarrollo a identificar, controlar y darle seguimiento a los requisitos y los cambios en cualquier momento”*. (Pressman, 2005)

Por otra parte, Bárbara McDonald Landazuri afirma que la gestión de requisitos es un componente esencial en el desarrollo de un proyecto de software ya que provee la dirección y alcance del proyecto. Expone además, que la gestión de requisitos es el proceso encargado de la identificación, asignación y seguimiento de los requisitos, incluyendo la verificación, la modificación y el control de su estado a lo largo del ciclo de vida del proyecto. (McDonald Landazuri, septiembre 2005)

Dichas definiciones sobre la gestión de requisitos permiten asumir que la gestión de requisitos está enfocada a las tareas de identificación o recolección de requisitos, su documentación, verificación y control de los cambios, que sobre ellos se produzcan. El impacto de los cambios en los requisitos puede incidir sobre otros artefactos, por ejemplo, los modelos producidos en el trabajo de análisis, diseño y pruebas. Se hace necesario entonces identificar las relaciones de los requisitos con el resto de los artefactos desarrollados o en proceso de desarrollo. Como consecuencia, la gestión de requisitos quedará a cargo de definir los procedimientos que guiarán el análisis previo a la aceptación de un cambio, y luego de realizado, facilitar su seguimiento.

Con la gestión efectiva de los requisitos será posible entender, desarrollar y entregar las funcionalidades que el cliente requiere, con un mínimo de afectación en el plazo y los recursos acordados garantizando la calidad en la producción de software. *“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.”* (McDonald Landazuri, septiembre 2005)

Se propone que la gestión de requisitos sea flexible y adaptable para reunir las necesidades del proyecto. Las características del alcance de esta actividad (Energy, abril 2000) variarán dependiendo de algunos factores claves como el tamaño y complejidad del proyecto, la experiencia de los clientes y del personal que trabaja en él, así como del propósito y uso del sistema que se construirá.

A pesar de que el proceso de gestión de requisitos es amplio, este trabajo de diploma se centrará en el seguimiento respecto a los cambios que se generan en los requisitos durante el ciclo de vida del software y las herramientas de gestión de requisitos que auxilian y/o automatizan estas tareas.

Específicamente, el seguimiento a los requisitos está sustentado en el concepto de trazabilidad de los requisitos que algunos autores han ofrecido.

Para Francisco Pinheiro, *“la trazabilidad es la habilidad de definir, capturar y seguir las trazas dejadas por los requerimientos en otros elementos del ambiente de desarrollo de software, así como las trazas que estos elementos dejan en los requisitos”*. (Pinheiro, julio 2000)

Asimismo, la definición de trazabilidad, que se reconoce como la más difundida, *“se concentra en la especificación de requisitos de software y requiere en primer lugar, que permita remontarse hacia atrás en el origen de los requerimientos y que también posibiliten referenciar a cada requerimiento en la documentación de desarrollo”*. (IEEE, 1984)

A partir de estas definiciones se puede entender el proceso de seguimiento de los requisitos como una actividad que posibilitará mostrar la relación existente entre los requisitos identificados y de estos con otros artefactos del sistema. Esto supone un amplio beneficio, en tanto ayudará a realizar un análisis del impacto de los cambios, lo que favorecerá la reducción de los riesgos y paralelamente será posible demostrar que se implementaron todos los requerimientos.

1.3 Seguimiento de los requisitos en el proyecto ERP Cuba.

En el proyecto ERP Cuba actualmente se realiza el control y seguimiento de los requisitos de software. Para ello, los analistas principales registran cada requisito en una hoja de cálculo donde escriben el nombre del requisito y el tipo asociado, que puede ser funcional o no funcional, así como el estado y la fase en que se encuentra el requisito. Los analistas principales además, agrupan los requisitos en dependencia de la funcionalidad que contemplan y definen la complejidad de cada requisito para luego, en conjunto con el arquitecto, precisar la prioridad con que serán implementados los requisitos.

Los analistas principales necesitan escribir fórmulas en la hoja de cálculo para obtener información estadística en relación con el estado de los requisitos y la fase en que estos se encuentran. Con esta información pueden observar el avance en la implementación de los requisitos que fueron identificados inicialmente. Pero con estas fórmulas no les es posible definir qué otros artefactos se ven afectados cuando un requisito cambia, por lo que la identificación de los riesgos y del impacto de los cambios de los requisitos en el proceso de desarrollo no es óptima.

El registro de los cambios en los requisitos se tiene que realizar de forma manual en el proyecto, esto implica que los analistas principales tengan que visitar los puestos de trabajo de analistas en cada uno de los diferentes módulos y revisar detenidamente los documentos que contienen esta información para posteriormente actualizarla. Atendiendo a la gran dimensión del proyecto, esta actividad se torna engorrosa y pueden cometerse errores.

Por otro lado, se debe tener en cuenta que el equipo de desarrollo del proyecto ERP Cuba es muy numeroso y los módulos están muy interrelacionados, de manera que los cambios en los requerimientos necesitan ser difundidos dentro de un amplio equipo, pues son de interés colectivo.

En la actualidad, para la transmisión de esta información los analistas necesitan reunirse sistemáticamente en largas reuniones de integración para dar a conocer los cambios que se generan en los requisitos en los diferentes módulos. Estas reuniones ocupan el tiempo de las personas implicadas, lo cual incide en una disminución de su tiempo de productividad.

Tras analizar el modo en que se realiza el seguimiento de los requisitos en el proyecto ERP Cuba, se concluye que este proceso está necesitando mucho esfuerzo y tiempo de las personas que están vinculadas con el mismo, unido al hecho de que no se está realizando eficientemente, lo cual se puede solucionar con una herramienta automatizada que facilite las actividades anteriormente mencionadas.

1.4 Herramientas para la gestión de requisitos.

El uso de herramientas de gestión de requisitos es alentado para mejorar tanto la productividad como la calidad en el desarrollo de un proyecto de software.

Lo que ha motivado a utilizar este tipo de herramientas es la complejidad al gestionar los requisitos. Aproximadamente del sesenta al setenta por ciento de los proyectos informáticos fallan por la pobre recopilación, análisis y gestión de requisitos. Dado que el éxito de un proyecto de software radica en la satisfacción del usuario, si sus requisitos no son completamente definidos y documentados, puede verse afectado todo el proyecto. (McDonald Landazuri, septiembre 2005)

Una herramienta de gestión de requisitos debe ser capaz de realizar algunas funciones como el registro, edición y rastreo de requisitos a su origen así como la generación de informes.

Otra función requerida por una herramienta de gestión de requisitos incluye el análisis del impacto del cambio: cuando un requisito es cambiado, se deben identificar todos los artefactos afectados. 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 cambios. El control de cambios, consiste en mantener la pista de añadir, borrar o cambiar cualquier requisito existente. (INCOSE, August 1997)

Durante la investigación para conformar el estado del arte del presente trabajo de diploma fueron estudiadas algunas herramientas que existen en el mundo para la gestión de los requisitos: Borland® CaliberRM™, DOORS, IBM Rational RequisitePro, IRqA 3.0 y OSMRT. De ellas se pudo constatar que proporcionan casi todas las necesidades básicas exigibles a una herramienta de gestión de requisitos para que sea incorporada por las empresas. Además, se encuentran ampliamente difundidas y son muy reconocidas, como demuestra el hecho de que aparecen en la gran mayoría de los materiales consultados en la actualidad. A continuación se explican las funcionalidades de algunas de estas herramientas.

1.4.1 IBM Rational RequisitePro.

Es un software propietario con varias ventajas como la de mantener los equipos de proyecto actualizados gracias a la creación, análisis y gestión de los requisitos de los casos de uso. Además, proporciona a los equipos la posibilidad de comprender el impacto de los cambios, brinda acceso basado en web para los equipos distribuidos y garantiza que todos los componentes del equipo estarán informados de los requisitos más actuales para asegurar la coherencia.

IBM Rational RequisitePro propicia una mejor comunicación dentro del equipo de desarrollo y reduce el riesgo de los proyectos, pero para su explotación debe ser integrado con otras aplicaciones que son también software propietario.

1.4.2 IRqA 3.0.

Es una herramienta para la gestión de requisitos diseñada para soportar las actividades realizadas en el proceso de especificación de requisitos. Entre sus funcionalidades se encuentran la captura, organización y análisis de los requisitos. En las últimas versiones se observa el soporte a la reutilización de requisitos y la posibilidad de definir gráficamente flujos de trabajo asociados a los requisitos (Malumbres, 2008). IRqA 3.0 emplea matrices de seguimiento por lo que permite conocer la trazabilidad de los requisitos durante el ciclo de desarrollo.

1.4.3 Borland® CaliberRM™.

Herramienta que facilita la colaboración, el análisis de las repercusiones y la comunicación entre los miembros del equipo de desarrollo, brindando información, exactitud y previsibilidad. Sus funcionalidades están soportadas en un repositorio centralizado, permite consultar el seguimiento de los requisitos durante la construcción de los sistemas, ofrece apoyo a las decisiones y se integra a otros productos de Borland®.

A pesar de estos potentes beneficios, dichas herramientas constituyen software propietario lo cual no se corresponde con el principio de independencia tecnológica en el cual está basado actualmente el desarrollo de software de la UCI. Por esta razón no serán adoptadas por el proyecto ERP Cuba para el seguimiento de los requisitos de software.

1.4.4 OSMRT.

Por otro lado, OSMRT es una herramienta basada en software libre que gestiona los requisitos del sistema durante todo el ciclo de vida del software. Puede ser explotada como aplicación de escritorio pero presenta problemas con la usabilidad, puesto que su diseño gráfico resulta incómodo al realizar algunas de sus funcionalidades. Además, no es posible realizar impresiones aunque un botón esté dispuesto para ello en la interfaz. La versión para su empleo a través de la web es aún menos amigable para el usuario, quienes a menudo no encuentran el flujo de las acciones mientras trabajan con la herramienta. Luego, dado que es necesario contar en el proyecto con una herramienta web, que con sólo disponer de un navegador web los usuarios puedan interactuar con ella y tras observar las limitaciones que OSMRT presenta, no es factible adoptar dicha herramienta para el seguimiento de los requisitos en el proyecto ERP Cuba.

Frente a las limitaciones que ofrecen las herramientas se hace necesario desarrollar una nueva herramienta, basada en software libre, capaz de darle seguimiento a los requisitos y que proporcione toda la información a los interesados inmediatamente después de la realización de un cambio en los requisitos, con lo que estará apoyando a la determinación del impacto de este cambio. La herramienta que se propone lleva como nombre SeReq.

A lo anterior se le une la necesidad del equipo de desarrollo del proyecto de probar el nuevo modelo de desarrollo basado en componentes que han definido, para lo cual la herramienta SeReq será tomada como piloto de estas pruebas. Es importante destacar que el alcance del presente trabajo de diploma abarca hasta el análisis y diseño de la herramienta dejando listas las bases para la futura implementación.

El desarrollo de productos de software es un proceso complejo que a menudo exige el empleo de una metodología de desarrollo que conduzca dicho proceso.

1.5 Modelo de desarrollo de software.

Para el desarrollo del presente trabajo de diploma se seguirá el nuevo modelo de desarrollo definido por el proyecto ERP Cuba para la especificación de requisitos y el modelo de diseño. La decisión está basada en el interés que presenta el equipo de desarrollo en probar dicho modelo en la generación de los

artefactos mencionados. El nuevo modelo de desarrollo está basado en componentes, guiado por casos de uso y es iterativo e incremental.

1.6 Herramienta y lenguaje para el modelado.

1.6.1 Herramientas CASE.

La ingeniería de software asistida por computadora (*CASE: Computer Aided Software Engineering*) ayuda a los ingenieros de software en todas las actividades asociadas a los procesos de software.

Las herramientas CASE automatizan las actividades de gestión de proyectos, gestionan todos los productos de los trabajos elaborados a través del proceso y ayudan a los ingenieros en el trabajo de análisis, diseño y codificación. La importancia de su uso radica en permitir reducir la cantidad de esfuerzo que se requiere para producir un producto o para alcanzar un hito en el proceso de desarrollo. Además, contribuyen a la calidad del software ya que proporcionan nuevas formas de observar la información de la ingeniería del software. (Pressman, 2005)

1.6.1.1 Visual Paradigm.

Visual Paradigm 6.1 es una herramienta CASE, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una más rápida construcción de aplicaciones de calidad, con un menor costo. Permite construir todos los tipos de diagramas necesarios para la documentación del software, así como importar proyectos realizados en Rational Rose.

Además, presenta un diseño enfocado al negocio que genera un software de calidad. Permite transformar diagramas Entidad-Relación en esquemas de base de datos y viceversa. Permite realizar ingeniería directa e inversa, obtener el código a partir de diagramas, así como diagramas a partir de un código previamente escrito.

1.6.2 Lenguaje Unificado de Modelado.

El Lenguaje Unificado de Modelado (*UML: Unified Modeling Language*) es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. Fue diseñado para ser un lenguaje de modelado de propósito general, que ofrece gran flexibilidad y expresividad y puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes. Igualmente es empleado para modelar aplicaciones de muy diversos dominios de aplicación (telecomunicaciones, comercio, sanidad, entre otros) y plataformas de objetos distribuidos como por ejemplo J2EE, .NET o CORBA.

1.7 Especificaciones de la arquitectura.

Teniendo en cuenta que el presente trabajo de diploma está centrado en el análisis y diseño de una herramienta para la gestión de requisitos, no es propósito del autor crear una nueva arquitectura para el desarrollo de la herramienta y por tanto adopta la arquitectura definida por el proyecto ERP Cuba, la misma está basada en desarrollo por componentes. La decisión está basada en el hecho de que esta arquitectura se encuentra actualmente en explotación, se le brinda soporte y con ella se han obtenido resultados concretos que hoy en día están siendo aplicados en entidades cubanas. Además, tener en cuenta que la herramienta SeReq será utilizada como piloto para probar la integración de los componentes que se han desarrollado en el proyecto ERP Cuba.

1.7.1 Arquitectura Cliente/Servidor.

La arquitectura Cliente/Servidor es una nueva tendencia en el desarrollo de redes, que tiene como objetivo optimizar el uso tanto del hardware como del software, a través de la separación de funciones: el cliente, quien inicia una determinada petición y el servidor, dedicado a responder dichas peticiones.

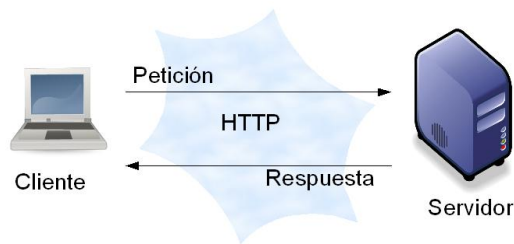


Figura 1 Arquitectura Cliente/Servidor

Puede presentarse como uno o varios clientes y servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información almacenada localmente.

Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

1.7.2 Patrón Modelo-Vista-Controlador (MVC).

El Modelo-Vista-Controlador es un patrón de arquitectura utilizado en sistemas web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para insertar, actualizar y consultar información de la base de datos.

La **Vista** es la información presentada al usuario. Una vista puede ser una página web o una parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página. Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

MVC es particularmente apropiado para aplicaciones web interactivas, aplicaciones donde un usuario web interactúa con un sitio web.

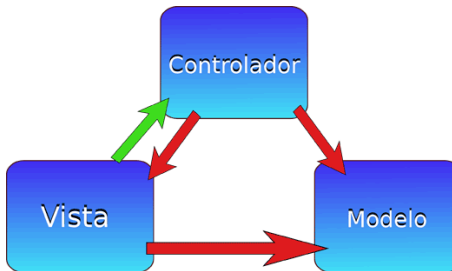


Figura 2 Patrón Modelo-Vista-Controlador.

1.7.3 Lenguaje de programación.

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

PHP (*Hypertext Pre-processor*) es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones.

Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la web.

- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.

De acuerdo con las facilidades expuestas anteriormente, PHP 5.2.4 será el lenguaje de programación en que se desarrollará la herramienta que propone el presente trabajo de diploma.

1.7.4 Sistema gestor de bases de datos.

PostgreSQL 8.3 es un gestor de bases de datos relacional orientado a objetos, libre y gratuito.

Presenta las siguientes propiedades:

- Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.
- Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- Aislamiento: Mediante un sistema de acceso concurrente multiversión (*MVCC: Multiversion concurrency control*) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.

1.7.6 Marcos de trabajo.

Los marcos de trabajo (frameworks) no son más que arquitecturas definidas para un determinado dominio de la aplicación que contiene un conjunto de componentes implementados y sus interfaces bien definidas, estos componentes se pueden utilizar, redefinir y crear nuevos componentes.

Los objetivos principales que persigue un framework son, entre otros, acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

1.7.6.1 Zend Framework.

Es un marco de trabajo de código abierto que está diseñado para PHP 5. Dentro de sus principales características están:

- Módulos para manejar archivos en formato de documento portátil, canales de sindicación de noticias y servicios web.
- Incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar la base de datos.
- Completa documentación y pruebas de alta calidad.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web.

1.7.6.2 Zend_Ext Framework.

Es un marco de trabajo de código abierto diseñado para PHP 5, elaborado a partir de Zend Framework cumpliendo con todas sus características. Lo novedoso consiste en un controlador vertical para el control de las acciones realizadas por las vistas hacia el controlador, un motor de reglas para las validaciones en el servidor y se le incluyó el IoC para la comunicación entre los módulos o componentes. Además, se le incorporó la integración con el Doctrine Framework para trabajo en la capa de abstracción de la base de datos y con el ExtJs Framework para el desarrollo de las vistas.

1.7.6.3 Doctrine Framework.

El framework Doctrine es un potente, multiplataforma y completo sistema para el mapeo de objeto-relacional (*ORM: Object Relational Mapper*) para PHP 5.2 ó superior.

1.7.6.4 ExtJS Framework.

ExtJS Framework facilita las herramientas necesarias para la creación de aplicaciones web con excelentes gráficos, ya que posee una considerable colección de elementos para el diseño de interfaces: ventanas, pestañas, menús, tablas, entre otros.

Brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

1.7.6.5 UCID Framework.

Es un framework encargado del trabajo con las vistas. Abarca la integración de ExtJS Framework con el sistema incluyendo el integrador de interfaz, el generador de interfaz dinámica y la impresión de documentos. Integra la iconografía, los diferentes temas de escritorio de la aplicación y el multilinguaje.

Conclusiones parciales

La realización de cambios a los requisitos de software es controlada por la gestión de requisitos y se plantea que puede afectar a otros artefactos, e incluso influir sobre el avance del proyecto. En el proyecto ERP Cuba no existe una forma óptima de informar al equipo de desarrollo la realización de cambios a los requisitos y el impacto que estos cambios causan, pues no se cuenta con una herramienta automatizada que facilite esta actividad.

Tras el estudio realizado en este capítulo se pudo constatar que las herramientas disponibles para el seguimiento de los requisitos no satisfacen las necesidades de los analistas. Luego de definir el modelo de desarrollo a seguir, investigar acerca de las metodologías de desarrollo de software, las herramientas y lenguaje para el modelado y concretar algunos aspectos sobre arquitectura de software, están creadas las condiciones para realizar el análisis y diseño de la herramienta SeReq la cual automatizará el seguimiento de los requisitos en el proyecto ERP Cuba.

CAPITULO 2: Descripción de la solución propuesta.

Introducción

En el presente capítulo se elaboran los artefactos que posibilitan realizar el análisis y el diseño de la herramienta SeReq para el seguimiento de los requisitos a partir del modelo de desarrollo ágil basado en la metodología RUP definido en el Capítulo 1.

Como artefactos que posibilitan el entendimiento y análisis del proceso que se desea automatizar se encuentran el modelo de dominio de la herramienta, compuesto por un modelo conceptual y un glosario de términos, además el diagrama de casos de uso del sistema, las especificaciones de los requerimientos funcionales y un listado de los requerimientos no funcionales.

Posteriormente, como artefactos creados durante el diseño de la herramienta se encuentran: el diagrama de componentes de la aplicación, que muestra la organización estructural de los elementos del diseño; las características comunes observadas en los diseños de clases; los diagramas de clases del diseño y el modelo de datos. Asimismo, se ha descrito la solución adoptada para gestionar la seguridad y el tratamiento de errores del sistema.

2.1 Modelo de dominio.

El primer paso en el proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimientos sobre el problema en cuestión.

En el presente trabajo de diploma se intenta capturar los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, por lo que decide realizar un modelo de dominio. Para conformar dicho modelo se ha elaborado un modelo conceptual que contiene los objetos y conceptos que se enmarcan en la problemática expuesta, así como un glosario de términos para clarificar el vocabulario menos conocido.

2.1.1 Modelo conceptual.

El modelo conceptual es una representación de conceptos del mundo real, no de componentes de software. El objetivo de la creación de este artefacto es aumentar la comprensión del problema y contribuir a esclarecer la terminología o nomenclatura del dominio. Puede verse como un modelo que comunica a los interesados cuáles son los términos importantes y cómo se relacionan entre sí. (Betancourt Cruz, et al., 2007)

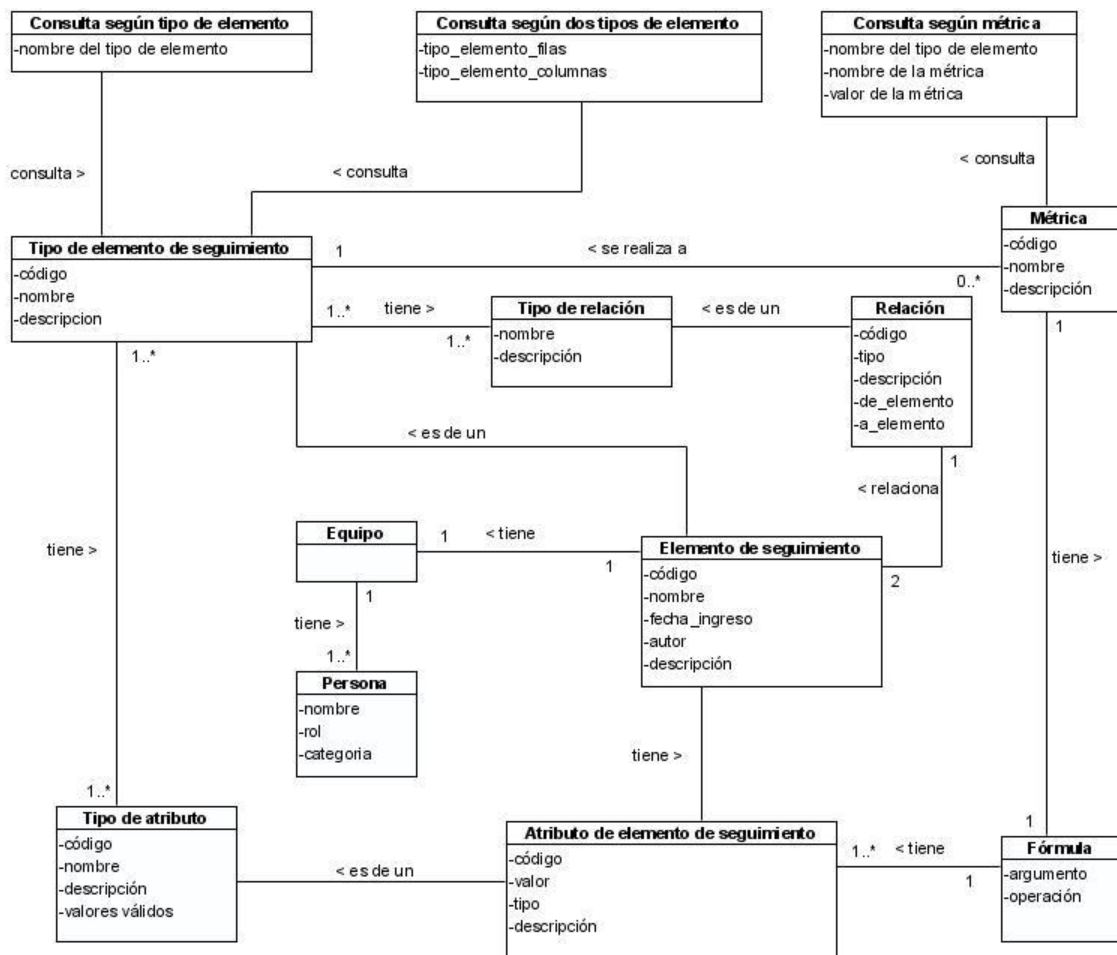


Figura 3 Modelo conceptual.

2.1.2 Glosario de términos.

El glosario de términos es una lista de conceptos asociados al negocio que son usados comúnmente y que deben ser del dominio del equipo de desarrollo para poder modelar el negocio y dar una solución a la problemática planteada.

Argumento: En la fórmula, se refiere a cada uno de los factores que la conforman.

Atributo de elemento de seguimiento: Es una propiedad del elemento de seguimiento.

Consulta: Es un conjunto de información relacionada con un concepto. Puede estar basada en un tipo de elemento de seguimiento para conocer la información de todas las instancias asociadas a éste, o puede expresarse en una matriz de seguimiento para conocer las relaciones existentes entre las instancias de dos tipos de elementos de seguimiento. Por otra parte, es posible realizar consultas empleando métricas previamente definidas.

Elemento de seguimiento: Es cualquier elemento textual o modelo que es necesario seguir explícitamente durante el desarrollo de software.

Métrica: Es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (IEEE, 1998) En la herramienta que trata el presente trabajo de diploma las métricas pueden ser definidas por el usuario a través de fórmulas que arrojan información relacionada con los parámetros involucrados.

Operación: Se refiere a una operación matemática. Por ejemplo: adición, multiplicación, entre otras.

Rol: Papel que desempeña un usuario frente al sistema. Está asociado a las acciones que el usuario puede llevar a cabo en el sistema.

Seguimiento: Es el proceso de mantener una traza directa entre los artefactos generados durante el proceso de desarrollo de un software.

2.2 Requisitos.

El flujo de trabajo de Levantamiento de Requisitos radica su mayor esfuerzo en el establecimiento de un acuerdo entre los clientes y los desarrolladores, sobre lo que el sistema debe hacer. Se precisa el ámbito del sistema y se pretende entender el comportamiento del software definiendo una interfaz enfocada a las necesidades y metas del usuario. Este flujo de trabajo intenta proveer a los desarrolladores de un mejor entendimiento de los requerimientos del sistema. (Jacobson, et al., 2004)

Las técnicas utilizadas para el Levantamiento de Requisitos de la herramienta SeReq fueron las entrevistas realizadas a los clientes y los talleres impartidos por estos, ámbito que fue propicio para desencadenar lluvias de ideas.

Existen dos clasificaciones para los requisitos de software: funcionales y no funcionales.

A continuación se expone el modelo de casos de uso del sistema donde se han modelado las funcionalidades de la herramienta SeReq, así como la descripción textual de estos casos de uso. Luego se explican los requisitos no funcionales de la aplicación.

2.2.1 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

2.2.1.1 Modelo de Casos de Uso del Sistema.

La Figura 6 muestra el diagrama de casos de uso del sistema de herramienta SeReq. La descripción de cada caso de uso se explica en la sección [2.2.1.3 Especificación de los Casos de Uso del Sistema](#) del presente capítulo.

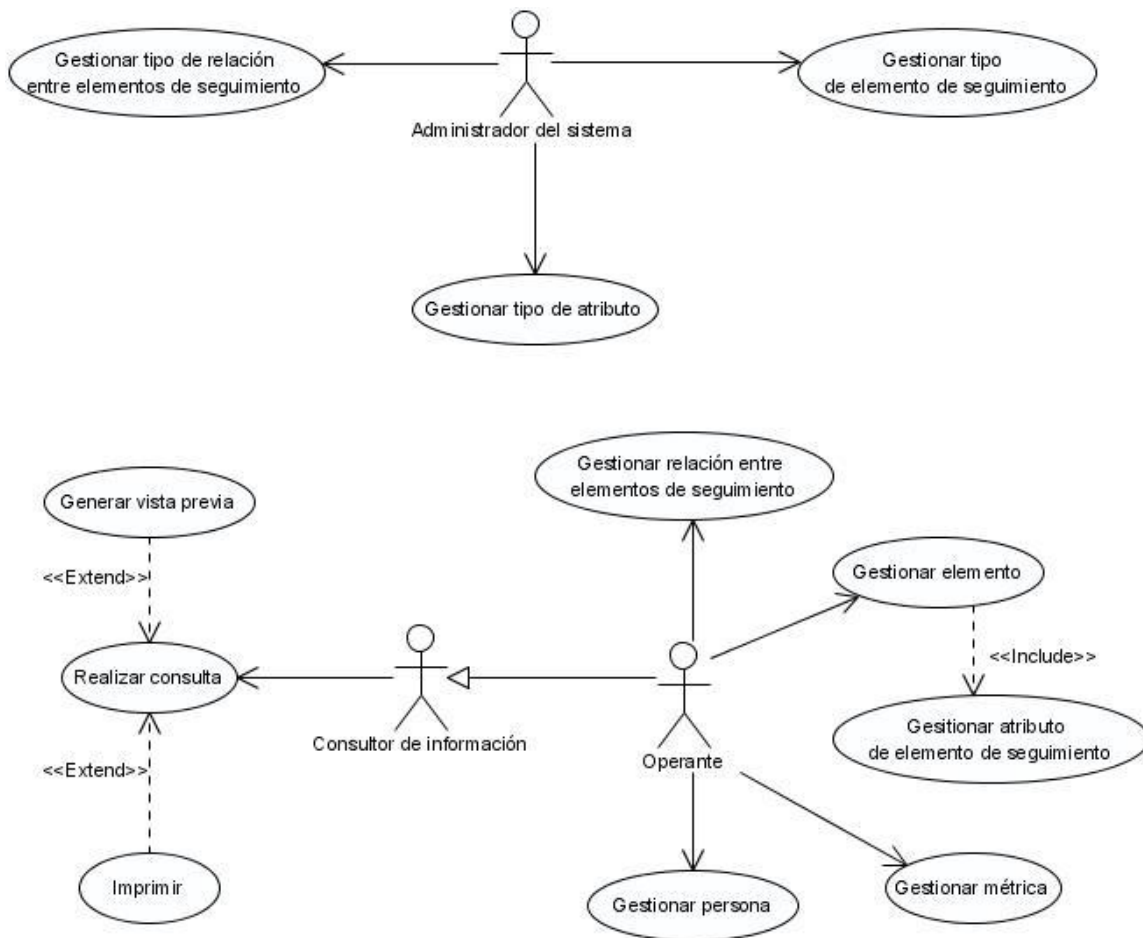


Figura 4 Diagrama de Casos de Uso del Sistema.

2.2.1.2 Justificación de los Actores del Sistema.

La Tabla 1 expone la justificación de los actores del sistema.

Actor	Descripción
Administrador del sistema	Encargado de gestionar los tipos de objetos genéricos, los que posteriormente

	servirán de base para la creación de objetos concretos.
Operante	Encargado de realizar las funcionalidades que brinda el sistema. Gestiona los objetos concretos que se manipulan en el sistema y realiza operaciones con ellos.
Consultor de información	Encargado de consultar la información disponible en el sistema sin posibilidad de modificarla o eliminarla.

Tabla 1 Justificación de los Actores del Sistema.

2.2.1.3 Especificación de los Casos de Uso del Sistema.

A continuación se especifican los casos de uso del sistema que propone el presente trabajo de diploma. Se ha utilizado la Plantilla DCS - Especificación de Requisitos 3.0 empleada en el proyecto ERP Cuba, puesto que contiene los campos necesarios para documentar los requisitos eficientemente (Vega Miniet, 2008). Con esto se logra que el cliente comprenda de una mejor forma los requisitos del futuro sistema y se evita la generación de gran cantidad de información que a menudo el resto de los desarrolladores no revisan para continuar el desarrollo. Dicha plantilla se puede consultar en el [Anexo 1](#).

Dada la adopción de la plantilla mencionada anteriormente, los casos de uso son especificados como requisitos de la aplicación. Por ejemplo, los casos de uso cuyos nombres comienzan con la palabra gestionar, han sido especificados como requisitos en correspondencia con los cuatro escenarios fundamentales: adicionar, modificar, buscar y eliminar.

Caso de Uso del Sistema del Sistema Gestionar tipo de elemento de seguimiento.

Especificación del Requisito Adicionar tipo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
---------------------------	------------------	------------------

	Tipo de elemento de seguimiento.	Código, nombre, descripción.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se introduce el código, el nombre y la descripción del tipo de elemento de seguimiento que se desea adicionar.</p> <p>Se asocian tipos de atributo al tipo de elemento de seguimiento que se está adicionando.</p> <p>Si se asocia un tipo de atributo no deseado el sistema permite a la vez desasociar tipo de atributo.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de elemento de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	El campo nombre no admite números.	
Post-condiciones	Se ha adicionado un nuevo tipo de elemento de seguimiento.	
Post-requisito	No procede.	

Especificación del Requisito Modificar tipo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de elemento de seguimiento existente.</p> <p>Se modifican los datos del tipo de elemento de seguimiento seleccionado.</p> <p>Se pueden asociar o desasociar tipos de atributos al tipo de elemento de seguimiento.</p> <p>Si se asocia un tipo de atributo no deseado el sistema permite a la vez desasociar tipo de atributo.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de elemento de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	<p>El campo nombre no admite números.</p> <p>Se ha seleccionado un tipo de elemento de seguimiento para modificarlo.</p>	

Post-condiciones	Se ha modificado un tipo de elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Eliminar tipo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de elemento de seguimiento existente.</p> <p>Se elimina el tipo de elemento de seguimiento seleccionado.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene los tipos de elemento de seguimiento.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione tipo de elemento de seguimiento a eliminar.</p>	
Validaciones	No han sido creados elementos de seguimiento del tipo de elemento de	

	<p>seguimiento que se desea eliminar.</p> <p>No han sido creadas métricas sobre el tipo de elemento de seguimiento que se desea eliminar.</p> <p>Se ha seleccionado un tipo de elemento de seguimiento para eliminarlo.</p>
Post-condiciones	Se ha eliminado un tipo de elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Buscar tipo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de los tipos de elementos de seguimiento que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un tipo de elemento del listado y modificarlo o eliminarlo si lo desea.</p>	

	Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.
Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Caso de Uso del Sistema del Sistema Gestionar tipo de atributo.

Especificación del Requisito Adicionar tipo de atributo.

Conceptos tratados	Conceptos	Atributos
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se introduce el código, el nombre y la descripción del tipo de atributo que se desea adicionar.</p> <p>Se introducen los valores válidos para el tipo de atributo que se desea adicionar.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de atributo.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario</p>	

	permitiéndole que introduzca los datos correctos.
Validaciones	El campo nombre no admite números.
Post-condiciones	Se ha adicionado un nuevo tipo de atributo.
Post-requisito	No procede.

Especificación del Requisito Modificar tipo de atributo.

Conceptos tratados	Conceptos	Atributos
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de atributo existente.</p> <p>Se modifican los datos del tipo de atributo seleccionado.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de atributo.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	El campo nombre no admite números.	

	<p>No han sido creados atributos de elemento de seguimiento en el sistema que sean del tipo de atributo que se desea eliminar.</p> <p>Se ha seleccionado un tipo de atributo para eliminarlo.</p>
Post-condiciones	Se ha modificado un tipo de atributo.
Post-requisito	No procede.

Especificación del Requisito Eliminar tipo de atributo.

Conceptos tratados	Conceptos	Atributos
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
	Tipo de elemento de seguimiento.	Código, nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de atributo existente.</p> <p>Se elimina el tipo de atributo seleccionado.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p>	

	<p>Se actualiza la tabla que contiene los tipos de atributo.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione un atributo a eliminar.</p>
Validaciones	<p>El tipo de atributo que se desea eliminar no ha sido asociado a ningún tipo de elemento de seguimiento.</p> <p>No han sido creados atributos de elemento de seguimiento en el sistema que sean del tipo de atributo que se desea eliminar.</p> <p>Se ha seleccionado un tipo de atributo para eliminarlo.</p>
Post-condiciones	Se ha eliminado un tipo de atributo.
Post-requisito	No procede.

Especificación del Requisito Buscar tipo de atributo.

Conceptos tratados	Conceptos	Atributos
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de los tipos de atributo que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un tipo de atributo del listado y modificarlo o</p>	

	eliminarlo si lo desea. Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.
Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Caso de Uso del Sistema del Sistema Gestionar tipo de relación entre elementos de seguimiento.

Especificación del Requisito Adicionar tipo de relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de relación.	Nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se introduce el nombre y la descripción del tipo de relación que se desea adicionar.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de relación entre elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	

Validaciones	El campo nombre no admite números.
Post-condiciones	Se ha adicionado un tipo de relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Modificar tipo de relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de relación.	Nombre, descripción.
	Relación.	Código, descripción, tipo, de_elemento, a_elemento.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de relación entre elementos de seguimiento existente.</p> <p>Se modifican los datos del tipo de relación entre elementos de seguimiento seleccionado.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los tipos de relación entre elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	No han sido creadas relaciones entre elementos de seguimiento que sean del	

	<p>tipo de relación que se desea modificar.</p> <p>El campo nombre no admite números.</p> <p>Se ha seleccionado un tipo de relación entre elementos de seguimiento para modificarlo.</p>
Post-condiciones	Se ha modificado una relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Eliminar tipo de relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de relación.	Nombre, descripción.
	Relación.	Código, descripción, tipo, de_elemento, a_elemento.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de relación entre elementos de seguimiento existente.</p> <p>Se elimina la relación entre elementos de seguimiento seleccionada.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene los tipos de relación entre elementos de seguimiento.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al</p>	

	usuario para que seleccione una relación a eliminar.
Validaciones	<p>No existen relaciones entre elementos de seguimiento que sean del tipo de relación que se desea eliminar.</p> <p>Se ha seleccionado un tipo de relación entre elementos de seguimiento para eliminarlo.</p>
Post-condiciones	Se ha eliminado un tipo de relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Buscar tipo de relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Tipo de relación.	Nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de los tipos de relaciones entre elementos de seguimiento que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un tipo de relación del listado y modificarlo o eliminarlo si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>	

Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Caso de Uso del Sistema del Sistema Gestionar elemento de seguimiento.

Especificación del Requisito Adicionar elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
	Tipo de elemento de seguimiento	Nombre, descripción.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
Precondiciones	Precondiciones	Pre-requisito
	Se han adicionado tipos de elemento de seguimiento y tipos de atributo.	Adicionar tipo de elemento de seguimiento. Adicionar tipo de atributo.
Descripción	Se introduce el nombre y la descripción del elemento de seguimiento que se desea adicionar. Se selecciona el tipo de elemento al que pertenece.	

	<p>Si el elemento de seguimiento que se está adicionando es un requisito de software el sistema permite al usuario que conforme el grupo de trabajo que estará inmerso en el desarrollo de este requisito.</p> <p>Se gestionan los atributos del elemento de seguimiento que se está adicionando.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>
Validaciones	<p>El campo nombre no admite números.</p> <p>Si el elemento de seguimiento que se desea adicionar es un requisito el usuario el equipo de trabajo asociado a ese requisito no debe estar vacío.</p> <p>Para gestionar atributos es necesario seleccionar el tipo de elemento de seguimiento previamente.</p>
Post-condiciones	Se ha adicionado un elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Modificar elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.

	Tipo de elemento de seguimiento	Código, nombre, descripción.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de elemento de seguimiento existente.</p> <p>Se modifican los datos del elemento de seguimiento seleccionado.</p> <p>Se selecciona el tipo de elemento al que pertenece.</p> <p>Si el elemento de seguimiento que se está modificando es un requisito se realizan las modificaciones necesarias en el equipo de trabajo asociado al requisito</p> <p>Se gestionan los atributos del elemento de seguimiento que se está adicionando.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	

Validaciones	<p>El campo nombre no admite números.</p> <p>Si el elemento de seguimiento es un requisito el equipo de trabajo asociado al requisito no debe estar vacío.</p> <p>Para gestionar atributos es necesario seleccionar el tipo de elemento de seguimiento previamente.</p> <p>Se ha seleccionado un elemento de seguimiento para modificarlo.</p>
Post-condiciones	Se ha modificado un elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Eliminar elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
	Relación.	Nombre, descripción, tipo, de_elemento, a_elemento.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona un tipo de elemento de seguimiento existente.</p> <p>Se elimina el tipo de elemento de seguimiento seleccionado.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p>	

	<p>Se actualiza la tabla que contiene los elementos de seguimiento.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione un elemento a eliminar.</p>
Validaciones	<p>El elemento de seguimiento que se desea eliminar no debe tener relaciones con otro elemento.</p> <p>Se ha seleccionado un elemento de seguimiento para eliminarlo.</p>
Post-condiciones	<p>Se ha eliminado un elemento de seguimiento.</p>
Post-requisito	<p>No procede.</p>

Especificación del Requisito Buscar elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
	Tipo de elemento de seguimiento	Código, nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de los elementos de seguimiento que existen según el tipo de elemento seleccionado.</p>	

	<p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un elemento del listado y modificarlo o eliminarlo si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>
Validaciones	No procede.
Post-condiciones	No procede.
Post-requisito	No procede.

Caso de Uso del Sistema del Sistema Gestionar atributo de elemento de seguimiento.

Especificación del Requisito Adicionar atributo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	Se selecciona el tipo de atributo para el atributo que se desea adicionar.	

	<p>Se registra el código, el valor y la descripción del atributo que se desea adicionar.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los atributos del elemento de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>
Validaciones	El campo nombre no admite números.
Post-condiciones	Se ha adicionado un nuevo atributo de elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Modificar atributo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.

Descripción	<p>Se selecciona un atributo de elemento de seguimiento existente.</p> <p>Se modifican los datos del atributo de elemento de seguimiento seleccionado.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los atributos del elemento de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>
Validaciones	<p>El campo nombre no admite números.</p> <p>Se ha seleccionado un atributo de elemento de seguimiento para modificarlo.</p>
Post-condiciones	Se ha modificado un atributo de elemento de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Eliminar atributo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
	Tipo de atributo.	Código, nombre, descripción, valores válidos.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha	No procede.

	autenticado en el sistema.	
Descripción	<p>Se selecciona un atributo de elemento de seguimiento existente.</p> <p>Se elimina el atributo de elemento de seguimiento seleccionado.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene los atributos del elemento de seguimiento.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione un atributo a eliminar.</p>	
Validaciones	<p>El atributo de elemento de seguimiento que se desea eliminar no ha sido asociado a ningún elemento de seguimiento.</p> <p>Se ha seleccionado un atributo de elemento de seguimiento para eliminarlo.</p>	
Post-condiciones	Se ha eliminado un atributo de elemento de seguimiento.	
Post-requisito	No procede.	

Especificación del Requisito Buscar atributo de elemento de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Atributo de elemento de seguimiento.	Código, descripción, tipo, valor.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
Precondiciones	Precondiciones	Pre-requisito

	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de los atributos de elementos de seguimiento que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un atributo del listado y modificarlo o eliminarlo si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Caso de Uso del Sistema del Sistema Gestionar persona.

Especificación del Requisito Adicionar persona.

Conceptos tratados	Conceptos	Atributos
	Persona	Nombre, categoría, rol de proyecto.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.

Descripción	<p>Se introduce el nombre, la categoría y el rol de proyecto de la persona que se desea adicionar.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene las personas.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>
Validaciones	El campo nombre no admite números.
Post-condiciones	Se ha adicionado una persona.
Post-requisito	No procede.

Especificación del Requisito Modificar persona.

Conceptos tratados	Conceptos	Atributos
	Persona	Nombre, categoría, rol de proyecto.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una persona existente.</p> <p>Se modifican los datos de la persona seleccionada.</p>	

	<p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene las personas.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>
Validaciones	El campo nombre no admite números.
Post-condiciones	Se ha modificado una persona.
Post-requisito	No procede.

Especificación del Requisito Eliminar persona.

Conceptos tratados	Conceptos	Atributos
	Persona	Nombre, categoría, rol de proyecto.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una persona existente.</p> <p>Se elimina la persona seleccionada.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene las personas.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al</p>	

	usuario para que seleccione un elemento a eliminar.
Validaciones	La persona que se desea eliminar no debe formar parte de ningún equipo de trabajo asociado a un requisito. Se ha seleccionado una persona para eliminarla.
Post-condiciones	Se ha eliminado una persona.
Post-requisito	No procede.

Especificación del Requisito Buscar persona.

Conceptos tratados	Conceptos	Atributos
	Persona	Nombre, categoría, rol de proyecto.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de las personas que existen según el tipo de elemento seleccionado.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar un elemento del listado y modificarlo o eliminarlo si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>	
Validaciones	No procede.	

Post-condiciones	No procede.
Post-requisito	No procede.

Especificación del Requisito Conformar equipo de trabajo asociado a un requisito.

Conceptos tratados	Conceptos	Atributos
	Equipo de trabajo.	Listado de personas.
	Persona.	Nombre, categoría, rol de proyecto.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El sistema muestra una interfaz con dos tablas. Una de las tablas muestra a las personas que están incluidas en el equipo de trabajo del requisito. La otra de las tablas muestra a las personas que pertenecen al proyecto y no están incluidas en el equipo de trabajo del requisito en cuestión.</p> <p>En la primera tabla el usuario selecciona y quita del equipo de trabajo a las personas que desea.</p> <p>En la segunda tabla el usuario selecciona e incluye a las personas que desea en el equipo de trabajo.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene los equipos de trabajo.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario</p>	

	permitiéndole que introduzca los datos correctos.
Validaciones	No procede.
Post-condiciones	Se ha conformado un equipo de trabajo.
Post-requisito	No procede.

Caso de Uso del Sistema del Sistema Gestionar relación entre elementos de seguimiento.

Especificación del Requisito Adicionar relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Relación.	Código, descripción, tipo, de_elemento, a_elemento.
	Tipo de relación.	Nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se introduce el código y la descripción de la relación que se desea adicionar.</p> <p>Se seleccionan los dos elementos de seguimiento que quedarán relacionados.</p> <p>Se selecciona el tipo de relación para la relación que se desea adicionar.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene las relaciones entre elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario</p>	

	permitiéndole que introduzca los datos correctos.
Validaciones	No procede.
Post-condiciones	Se ha adicionado una nueva relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Modificar relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Relación.	Código, descripción, tipo, de_elemento, a_elemento.
	Tipo de relación.	Nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una relación entre elementos de seguimiento existente.</p> <p>Se modifican los datos de la relación entre elementos de seguimiento seleccionada.</p> <p>Si los datos son correctos se muestra un mensaje de información al usuario sobre el éxito de la acción.</p> <p>Se actualiza la tabla que contiene las relaciones entre elementos de seguimiento.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	

Validaciones	Se ha seleccionado una relación entre elementos de seguimiento para modificarla.
Post-condiciones	Se ha modificado una relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Eliminar relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
	Relación.	Código, descripción, tipo, de_elemento, a_elemento.
	Tipo de relación.	Nombre, descripción.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una relación entre elementos de seguimiento existente.</p> <p>Se elimina la relación entre elementos de seguimiento seleccionada.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene las relaciones entre elementos de seguimiento.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione una relación a eliminar.</p>	
Validaciones	Se ha seleccionado una relación entre elementos de seguimiento para	

	eliminarla.
Post-condiciones	Se ha eliminado una relación entre elementos de seguimiento.
Post-requisito	No procede.

Especificación del Requisito Buscar relación entre elementos de seguimiento.

Conceptos tratados	Conceptos	Atributos
		Tipo de relación.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de las relaciones entre elementos de seguimiento que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar una relación del listado y modificarla o eliminarla si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Caso de Uso del Sistema del Sistema Gestionar métrica.

Especificación del Requisito Adicionar métrica.

Conceptos tratados	Conceptos	Atributos
	Métrica	Código, nombre, tipo de elemento, descripción, fórmula.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se introduce el nombre y la descripción de la métrica que se desea adicionar.</p> <p>Se selecciona el tipo de elemento de seguimiento sobre el que se desea crear la métrica.</p> <p>Se introduce la fórmula que contendrá la métrica.</p> <p>Si los datos son correctos se muestra una interfaz con el resultado de la métrica.</p> <p>Se actualiza la tabla que contiene las métricas.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	<p>La fórmula de la métrica debe contener operaciones y/o funciones matemáticas.</p> <p>El campo nombre no admite números.</p>	
Post-condiciones	Se ha adicionado una métrica.	
Post-requisito	No procede.	

Especificación del Requisito Modificar métrica.

Conceptos tratados	Conceptos	Atributos
	Métrica	Código, nombre, tipo, descripción, fórmula.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una métrica existente.</p> <p>Se modifican los datos y/o la fórmula de la métrica seleccionada.</p> <p>Si los datos son correctos se muestra una interfaz con el resultado de la métrica.</p> <p>Se actualiza la tabla que contiene las métricas.</p> <p>Si los datos no son correctos se muestra un mensaje de error al usuario permitiéndole que introduzca los datos correctos.</p>	
Validaciones	<p>La fórmula de la métrica debe contener operaciones y/o funciones matemáticas.</p> <p>El campo nombre no admite números.</p> <p>Se ha seleccionado una métrica para modificarla.</p>	
Post-condiciones	Se ha modificado una métrica.	
Post-requisito	No procede.	

Especificación del Requisito Eliminar métrica.

Conceptos tratados	Conceptos	Atributos
---------------------------	------------------	------------------

	Métrica	Código, nombre, tipo, descripción, fórmula.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el sistema.	No procede.
Descripción	<p>Se selecciona una métrica existente.</p> <p>Se elimina la métrica seleccionada.</p> <p>Si la acción de eliminar tiene éxito se muestra un mensaje con esta información al usuario.</p> <p>Se actualiza la tabla que contiene las métricas.</p> <p>Si no se produce la selección correctamente se muestra un mensaje de error al usuario para que seleccione una métrica a eliminar.</p>	
Validaciones	Se ha seleccionado una métrica para eliminarla.	
Post-condiciones	Se ha eliminado una métrica.	
Post-requisito	No procede.	

Especificación del Requisito Buscar métrica.

Conceptos tratados	Conceptos	Atributos
	Métrica	Código, nombre, tipo, descripción, fórmula.
Precondiciones	Precondiciones	Pre-requisito
	El usuario se ha autenticado en el	No procede.

	sistema.	
Descripción	<p>El usuario completa los datos según el criterio elegido para realizar la búsqueda.</p> <p>El sistema realiza una búsqueda de las métricas que existen.</p> <p>El sistema muestra un listado con el resultado de la búsqueda.</p> <p>El usuario puede seleccionar una métrica del listado y modificarla o eliminarla si lo desea.</p> <p>Si la búsqueda no tiene éxito se muestra un mensaje de información al usuario.</p>	
Validaciones	No procede.	
Post-condiciones	No procede.	
Post-requisito	No procede.	

Caso de Uso del Sistema del Sistema Realizar consulta.

Especificación del Requisito Realizar consulta.

Conceptos tratados	Conceptos	Atributos
	Consulta.	
	Consulta según métrica.	Nombre del tipo de elemento, nombre de la métrica, valor de la métrica.
	Consulta según dos tipos de elemento.	Tipo_elemento_filas, tipo_elemento_columnas.
	Consulta según tipo de elemento.	Nombre del tipo de elemento.

	Tipo de elemento de seguimiento.	Código, nombre, descripción.
	Elemento de seguimiento.	Nombre, fecha_ingreso, autor, fecha, descripción.
	Métrica.	Código, nombre, tipo, descripción, fórmula.
Precondiciones	Precondiciones	Pre-requisito
	Se han adicionado elementos de seguimiento.	Adicionar elemento de seguimiento.
Descripción	<p>Se selecciona el criterio por el que se desea realizar la consulta. Estos pueden ser tres:</p> <ul style="list-style-type: none"> - dado un tipo de elemento de seguimiento - entre dos tipos de elementos de seguimiento - dada una métrica. <p>Se seleccionan los datos requeridos según el criterio de para realizar la consulta.</p> <p>El sistema mostrará una interfaz con el resultado de la consulta.</p> <p>Si no es seleccionado el criterio para realizar la consulta el sistema mostrara un mensaje de error para exigir que se seleccione el criterio.</p>	
Validaciones	No procede.	
Post-condiciones	Se ha realizado una consulta.	

Post-requisito	No procede.
-----------------------	-------------

Caso de Uso del Sistema del Sistema Generar vista previa.

Conceptos tratados	Conceptos	Atributos
	Consulta	Nombre, valores de entrada.
Precondiciones	Precondiciones	Pre-requisito
	Se ha realizado una consulta.	Realizar consulta.
Descripción	Se genera la vista previa de la consulta.	
Validaciones	No procede.	
Post-condiciones	Se ha generado la vista previa de la consulta.	
Post-requisito	No procede.	

Caso de Uso del Sistema del Sistema Imprimir.

Conceptos tratados	Conceptos	Atributos
	Consulta	Nombre, valores de entrada.
Precondiciones	Precondiciones	Pre-requisito
	Se ha generado la vista previa de la consulta.	Generar vista previa de consulta.
Descripción	Se imprime la consulta.	
Validaciones	No procede.	

Post-condiciones	Se ha impreso una consulta.
Post-requisito	No procede.

2.2.2 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales. Los requisitos no funcionales de la aplicación han sido agrupados por categorías y se muestran a continuación.

Software

Cliente:

- Sistema operativo con interfaz gráfica y soporte para red.
- Las interfaces deben ser compatibles con Internet Explorer 5 ó superior y con Mozilla Firefox 2.0 ó superior.

Servidor:

- Sistema operativo: Ubuntu Server 8.04.
- Servidor web Apache 2.0 o superior.
- Gestor de base de datos: PostgreSQL 8.3 o superior.

Hardware

Cliente:

- Pentium a 133 MHz o superior.

- 128 MB como mínimo, de memoria RAM. 256 MB de memoria RAM recomendada.

Servidor:

- Dual Core Pentium 4 a 3.00 GHz por cada procesador.
- 512 MB de memoria RAM.
- 80 GB de disco duro.

Servidor de Base de Datos:

- Dual Core Pentium 4 a 3.00 GHz por cada procesador.
- 1 GB de memoria RAM.
- 80 GB de disco duro.

Apariencia o interfaz externa

El sistema debe poseer una interfaz amigable al usuario, basada en web, brindando facilidades que permitan interactuar y navegar con el sistema, de forma fácil y rápida.

El sistema debe cumplir con los estándares de diseño establecidos en el proyecto ERP Cuba según se especifica en ERP-ARQ Estándar para el diseño de interfaces v1.1. (UCID, mayo de 2008)

Seguridad

El servidor de bases de datos y el servidor de aplicación estarán instalados en computadoras diferentes.

La comunicación entre el servidor y el cliente será a través del protocolo HTTPS.

Cada usuario realizará operaciones en la aplicación en dependencia de sus privilegios o niveles de acceso.

Usabilidad

La aplicación debe poseer mapa de navegación disponible para los usuarios.

Soporte

Se debe brindar capacitación a los usuarios de la herramienta para el adecuado uso y explotación de la misma.

Se debe disponer de manual de usuario para apoyar la capacitación y el soporte a los clientes.

2.3 Modelo de diseño.

En el proceso de desarrollo de software, el diseño permite adquirir una comprensión de los aspectos asociados a los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución, concurrencia y tecnologías de interfaz gráfica de usuario. El diseño se propone crear una entrada apropiada y un punto de partida para las actividades de implementación, descomponer los trabajos de implementación en partes más manejables y capturar las interfaces entre los subsistemas. (UCI, 2006-2007)

2.3.1 Fundamentación y aplicación de los patrones utilizados para el diseño.

Se entiende por patrón una solución probada que se puede aplicar con éxito a un determinado tipo de problema que aparece repetidamente en algún campo. El establecimiento de estos patrones comunes es lo que posibilita el aprovechamiento de la experiencia acumulada en el diseño de aplicaciones. (Morales, 2007)

Para hacer un diseño eficiente se tomaron en cuenta un conjunto de patrones, que al ser experiencias de diseñadores expertos, permiten dar solución a problemas de diseño facilitando notablemente el trabajo posterior. Además, se debe tener en cuenta que los marcos de trabajo que se han utilizado llevan implícito la aplicación de gran cantidad de patrones. Aunque el presente trabajo de diploma, no persigue como objetivo un estudio minucioso de los patrones de diseño, a continuación exponen algunos patrones que están presentes en la solución.

En la definición del diseño del trabajo se utilizaron los patrones de asignación de responsabilidades GRASP (*General Responsibility Assignment Software Patterns*), entre ellos, los patrones Experto, Controlador, Bajo acoplamiento y Alta Cohesión.

El patrón Experto plantea que se debe asignar la responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir la responsabilidad.

El patrón Controlador establece que se debe asignar la responsabilidad del manejo de los eventos de un sistema a una clase controladora.

El patrón Bajo Acoplamiento persigue tener las clases lo menos ligadas entre sí que se pueda. De esta manera, en caso de producirse una modificación en alguna de ellas, se tiene la mínima repercusión posible en el resto de las clases. Con este patrón se potencia la reutilización y se disminuye la dependencia entre las clases.

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas estén las responsabilidades de una clase. La aplicación del patrón Alta Cohesión facilita un diseño donde las clases tienen responsabilidades estrechamente relacionadas y no realizan un trabajo enorme.

2.3.2 Diagrama de componentes de la herramienta.

Para el desarrollo de la herramienta SeReq se han definido componentes en los que se organizan los elementos del diseño de la aplicación.

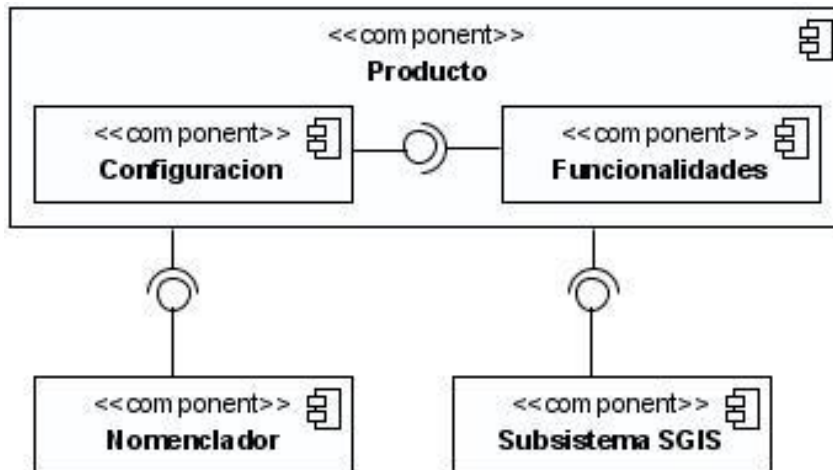


Figura 5 Diagrama de componentes.

En la Figura 8 se muestra la aplicación SeReq dividida en componentes que facilitan su organización en partes más manejables. El componente Configuración se encarga de la gestión de los objetos que manejan los usuarios de la herramienta. El componente Funcionalidades, por su parte, abarca las funciones que se pueden realizar con los objetos registrados en el sistema, entre ellas se encuentran: la elaboración de métricas y la realización de consultas.

En el diagrama se representa el componente Nomenclador, que es el encargado de gestionar de manera genérica los tipos de objetos de los que se derivarán posteriormente los objetos concretos que manejarán los usuarios de la herramienta, además del subsistema SIGIS, que posibilita la gestión de la seguridad del sistema. Estos no están incluidos en la herramienta SeReq en sí, sino que han sido implementados por el equipo de arquitectura del proyecto ERP Cuba y sus servicios son utilizados por la herramienta que propone el presente trabajo. Una descripción más detallada acerca del subsistema SIGIS se expone en la sección [2.3.6 Seguridad de la herramienta propuesta](#) del presente capítulo.

2.3.3 Características comunes observadas en los diseños de clases.

Esta sección tiene como objetivo describir algunas características que se observan en cada uno de los diseños de clases de la aplicación y de esta forma lograr claridad en los diagramas de diseño.

A continuación se presentan las características identificadas.



Figura 6 Controlador frontal del marco de trabajo Zend_Ext.

En la Figura 9 se representa el uso del controlador frontal del marco de trabajo Zend_Ext para la aplicación: el fichero index.php. El mismo tiene una relación de instanciación con los controladores en cada uno de los componentes mediante la cual los invoca en dependencia de la acción a realizar. Luego, como indica el patrón Modelo-Vista-Contolador, estos controladores cargarán las vistas o los modelos en cada componente.

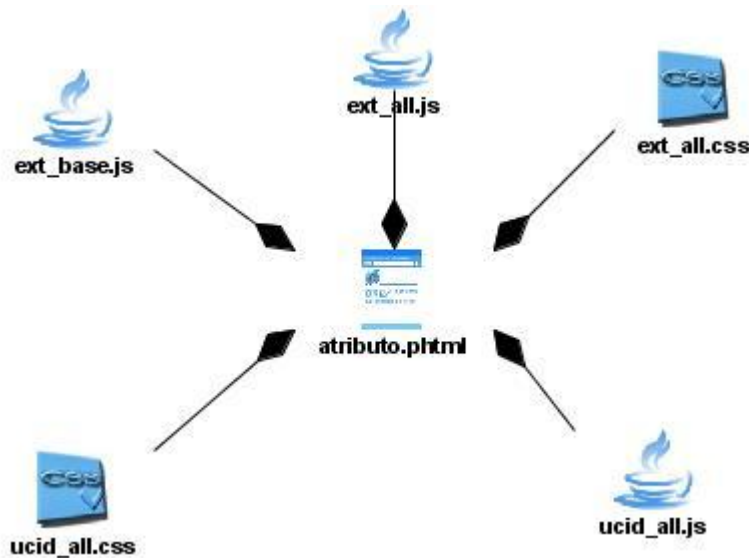


Figura 7 Clases para el trabajo con las interfaces de usuario.

La Figura 10 muestra las clases que brindan los marcos de trabajo ExtJS y UCID para el trabajo con las interfaces de usuario.

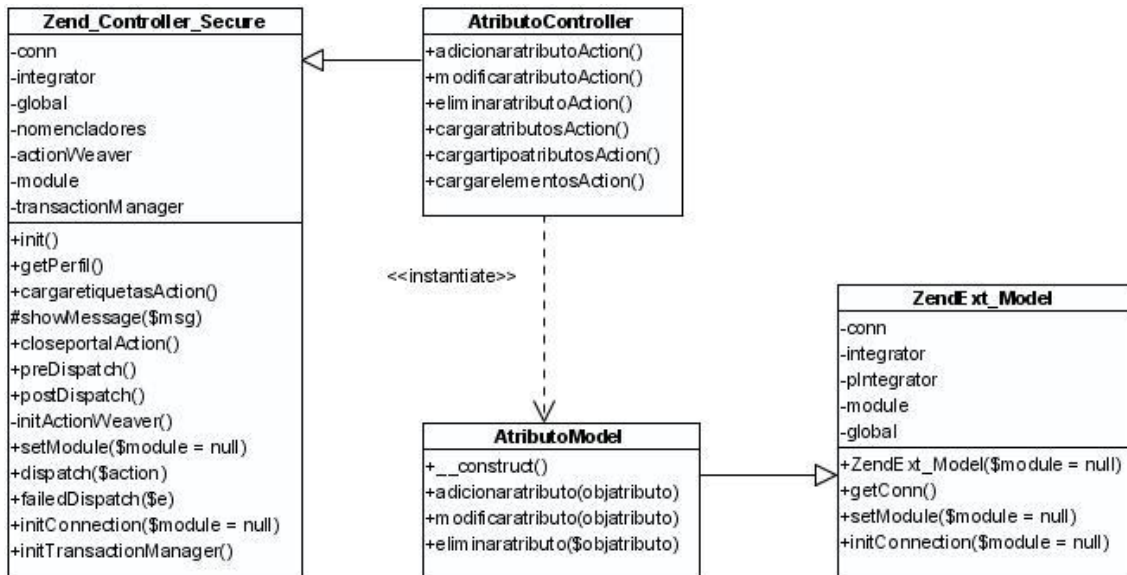


Figura 8 Clases de Zend_Ext usadas en los diseños de la aplicación.

En la Figura 11 se representan las clases del marco de trabajo Zend_Ext que son usadas por las clases controladoras y modelos de la aplicación.

2.3.4 Diagrama de clases de diseño.

Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido y tienen correspondencia directa con las clases en la implementación. Un diagrama de clases de diseño representa la parte estática del sistema: las clases y sus relaciones.

En la Figura 12 se muestra el diagrama de clases de diseño para el requisito Gestionar atributo especificado anteriormente. En él se representa el controlador atributo.php, el cual es el encargado de construir la página cliente: atributo.phtml mediante la relación <<build>>. Esta página cliente contiene al script atributo.js pues este tiene implementado el código de ejecución de la página y a su vez contiene al formulario form_atributo, el cual toma los datos que el usuario ingrese en la página y los envía al controlador atributo.php para su posterior procesamiento.

El controlador atributo.php tiene una relación <<instantiate>> con la clase controladora AtributoController, encargada de la gestión de atributo. Esta clase a su vez, tiene una relación <<instantiate>> con la clase del modelo AtributoModel, pues esta última tiene la responsabilidad de realizar las operaciones con cierta complejidad sobre la información de las tablas de la base de datos y enviar el resultado a la controladora. Esto lo realiza accediendo a las clases del paquete Domain mediante la relación <<instantiate>>.

Dentro del paquete Domain se han mapeado las tablas de la base de datos y las clases con el prefijo "Base_" contienen la información de esas tablas. El framework Doctrine ofrece las funcionalidades necesarias para el mapeo de las tablas de la base de datos.

La clase controladora AtributoController puede acceder directamente al paquete Domain por medio de la relación <<instantiate>>, esto puede ocurrir en caso de que se requieran datos sin necesidad de procesarlos previamente.

El resto de los diagramas de clases de diseño se pueden encontrar en el [Anexo 3](#).

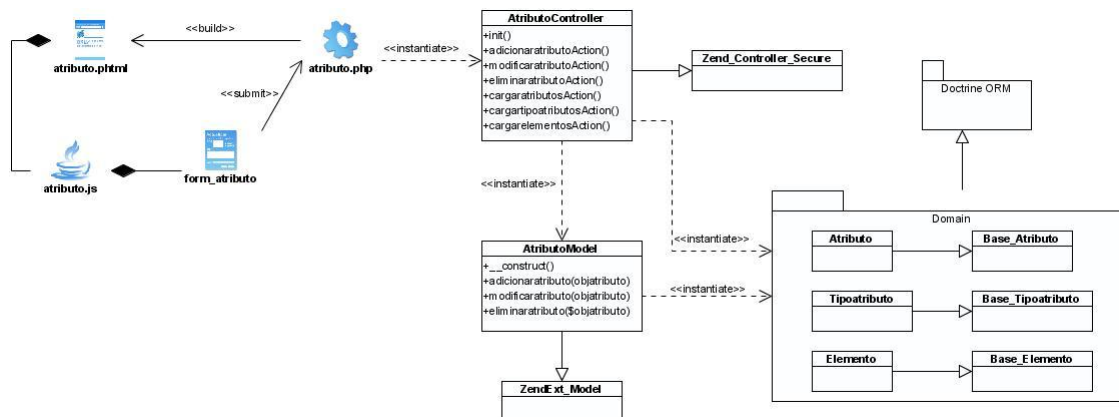


Figura 9 Diagrama de clases de diseño Caso de Uso Gestionar atributo.

2.3.5 Modelo de datos.

En el modelo de datos las tablas representan la información que se obtiene y es almacenada durante los procesos de la aplicación. En el mismo se puede apreciar la relación entre los datos. La Figura 13 muestra el modelo de datos propuesto durante el diseño de SeReq.

En el [Anexo 4](#) se describen las tablas que componen dicho modelo.

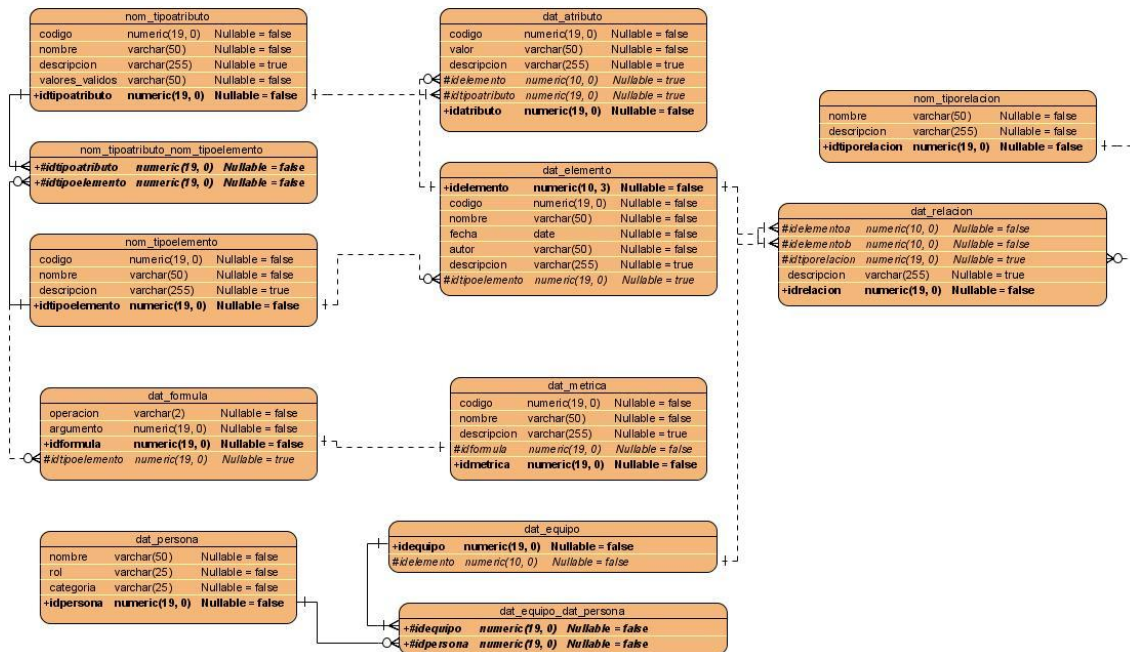


Figura 10 Modelo de datos.

2.3.6 Seguridad de la herramienta propuesta.

Con el objetivo de minimizar la inversión de tiempo y de recursos materiales y humanos en la implementación de la seguridad de la herramienta SeReq, ha sido reutilizada, de la arquitectura de seguridad definida en el proyecto ERP Cuba, la forma de garantizar dicha seguridad, la cual es provista y administrada por el Sistema de Gestión Integral de Seguridad (SGIS).

SGIS provee los servicios de seguridad a los subsistemas que se suscriben a él en un entorno seguro. De manera general, SGIS permite controlar las funcionalidades de la herramienta y de cada una de ellas, las acciones que las soportan. Igualmente, permite controlar los roles y usuarios con permisos para acceder a cada una de estas funcionalidades. (Gómez Baryolo, et al., 2008)

De esta forma se puede permitir o denegar el acceso tan profundamente como se requiera y la seguridad de la herramienta queda gestionada a través de un sistema de administración de seguridad centralizado. Cada usuario tendrá su perfil y podrá configurarlo según desee. El sistema permitirá la administración de conexiones ya que cada sistema puede configurar a qué servidores, gestores, bases de datos y esquemas conectarse. (Gómez Baryolo, et al., 2008)

2.3.7 Concepción general de la ayuda.

El sistema deberá contar con una ayuda en línea, capaz de informar al usuario las funcionalidades de la aplicación. Se proveerá información sobre cómo acceder a cada una de ellas, la descripción de su propósito y los pasos necesarios para llevar a cabo las operaciones obteniendo un resultado óptimo. La ayuda contará con imágenes que faciliten al usuario la comprensión de la información que trasmite.

2.3.8 Tratamiento de errores.

Para lograr que el usuario obtenga resultados satisfactorios en la ejecución de una tarea, se deberán pronosticar los posibles errores que el mismo pueda cometer. Para el tratamiento de errores de la herramienta SeReq se propone reutilizar la implementación desarrollada para este fin en el proyecto ERP Cuba. En dicho proyecto, el framework Zend_Ext resuelve el tratamiento de excepciones para el sistema Cedrux con la ayuda del componente Exception. (Díaz Peña, 2009)

De manera general, se validará en la entrada de los datos la completitud de los mismos, así como su formato de entrada. El usuario tendrá conocimiento de los errores ocurridos mediante mensajes informativos. En este sentido, se dará la información necesaria al usuario para corregir el error, informándole en qué consiste y cómo solucionarlo.

Con la reutilización explicada anteriormente se optimizan recursos materiales y tiempo en el desarrollo de la herramienta y se estandariza el mecanismo de tratamiento de errores propuesto por el proyecto ERP Cuba.

Conclusiones parciales

En este capítulo fue elaborado el modelo de dominio de la herramienta, compuesto por un modelo conceptual y un glosario de términos, así como el diagrama de casos de uso del sistema, las especificaciones de los requerimientos funcionales y un listado de los requerimientos no funcionales.

Además se describieron algunos patrones aplicados al diseño de la solución, se realizó el diagrama de componentes de la herramienta SeReq y se explicaron las características comunes observadas en los diseños de clases. Se elaboraron los diagramas de clases del diseño y se expuso el modelo de datos como parte del diseño de la base de datos. Finalmente, se describieron brevemente las características de la seguridad, la ayuda del sistema y de cómo se realizará el tratamiento de errores en el mismo. Todos estos artefactos servirán de base para la futura implementación de los requisitos especificados en el presente capítulo.

CAPITULO 3: Análisis de los resultados.

Introducción

En este capítulo se hace un análisis de los resultados obtenidos durante el desarrollo de la solución propuesta.

Los modelos obtenidos durante el desarrollo no pueden probarse en el sentido convencional, ya que no pueden ejecutarse. Sin embargo, existen otras formas de evaluar estos modelos, por ejemplo, mediante la aplicación de métricas. El objetivo de las métricas en general, es entender mejor la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo realizado a nivel de proyecto. (Pressman, 2005)

Las métricas internas se aplican a un producto de software no ejecutable durante todas las etapas de su desarrollo. Permiten medir la calidad de los entregables intermedios y predecir la calidad del producto final. Además, le permiten al usuario iniciar tempranamente las acciones correctivas en el ciclo de desarrollo.

Por la importancia que posee obtener una especificación de requisitos clara, completa y consistente se ha empleado la técnica de validación de requisitos por los prototipos de interfaz gráfica de usuario.

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución. Un prototipo puede revelar errores u omisiones en los requerimientos propuestos, favorece la comunicación entre clientes y desarrolladores y provee una primera visión del producto. (UCI, 2008-2009)

Los prototipos de interfaz gráfica de usuario asociados a cada caso de uso se pueden consultar en el [Anexo 2: Prototipos de interfaz gráfica de usuario.](#)

Análisis de la Especificación de requisitos.

La etapa de Levantamiento de Requisitos es fundamental en el desarrollo de un software. Puesto que uno de los principales objetivos de esta etapa es el de llegar a un acuerdo entre clientes y desarrolladores sobre las necesidades que debe satisfacer el sistema a implementar, la calidad de la especificación de los requisitos es un elemento crucial para el desarrollo del proyecto y asegurar así la calidad del producto final. Por esta razón se han ideado formas de evaluación para dichas especificaciones. La verificación de requisitos puede considerarse como la actividad cuyo objetivo primordial es alcanzar la calidad interna, mientras que la validación tiene como objetivo asegurar que los requisitos elicitados, documentados, analizados y verificados representen realmente las necesidades de clientes y usuarios.

3.1 Métricas para la Especificación de Requisitos.

Esta métrica, propuesta por Alan Davis en 1993 (Pressman, 2005), corresponde a la calidad de la especificación de requerimientos. Incluye características como: especificidad, corrección, comprensión, verificación, consistencia, concisión y exactitud. En el presente trabajo se aplicarán las métricas que contribuyen a determinar la especificidad y el grado de validación de los requisitos.

3.1.1 Aplicación de la métrica.

Para determinar la especificidad de los requisitos primeramente se define a nr que representa el total de requisitos de la especificación:

$$nr = nf + nnf$$

donde nf es el número de requisitos funcionales y nnf es el número de requisitos no funcionales. Luego se puede medir la especificidad o ausencia de ambigüedad con la fórmula:

$$Q_e = nui/nr, \text{ donde}$$

nui es el número de requerimientos donde todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca esté Q_e del valor 1, menor ambigüedad habrá en la especificación. (Pressman, 2005)

Para la evaluación de la métrica de la especificación de los requisitos se llevaron a cabo dos revisiones con el objetivo de comparar el trabajo realizado y mejorar su calidad.

3.1.2 Revisiones realizadas a especificaciones de requisitos.

La Tabla 1 muestra una comparación de las especificaciones de requisitos entre las revisiones realizadas en dos momentos del desarrollo. Los resultados han sido representados en el Gráfico 1 que se muestra seguidamente.

Revisión 1	Revisión 2
$nf = 30,$ $nnf = 20;$ $nr = nf + nnf = 30 + 20,$ $nr = 50.$ entonces $Q_e = nui / nr,$ donde $nui = 50 - 12(F, NF);$ por lo que $nui = 38$ $Q_e = 38 / 50 = 0.76$	$nf = 36,$ $nnf = 21;$ $nr = nf + nnf = 36 + 21,$ $nr = 57.$ entonces $Q_e = nui / nr,$ donde $nui = 57 - 2(F);$ por lo que $nui = 55$ $Q_e = 55 / 57 = 0.95$

Tabla 2 Comparación de las revisiones realizadas a las especificaciones.

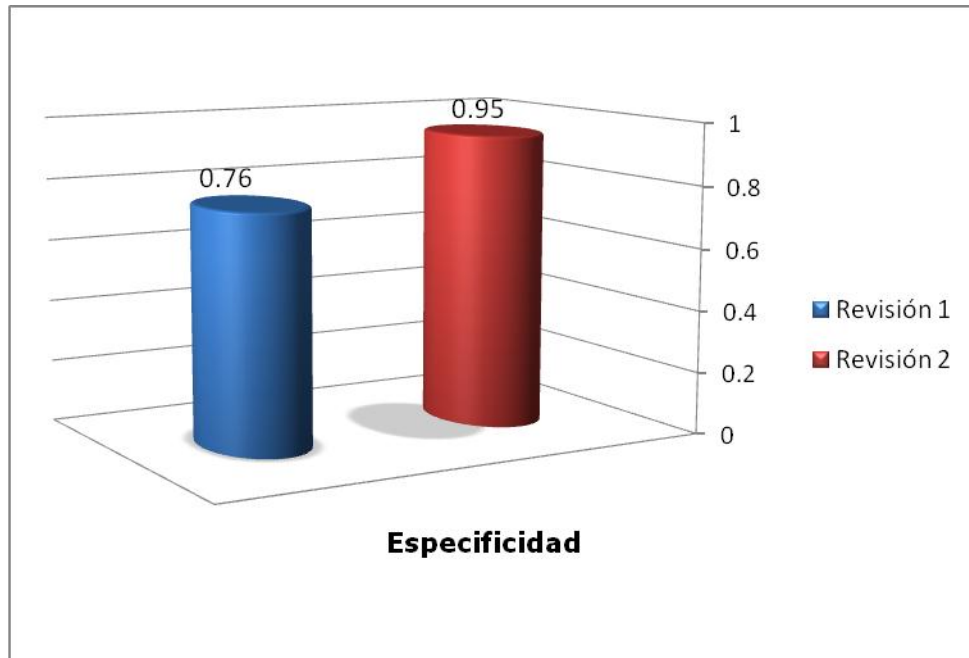


Gráfico 1 Control de la Calidad de la Especificación de Requisitos.

Por otra parte, para determinar el grado de validación de los requisitos se emplea la métrica global Q_v que incluye tanto los requisitos funcionales como los no funcionales.

$$Q_v = nc / (nc + nnv)$$

donde nc es el número de requisitos que se han validado como correctos y nnv el número de requisitos que no se han validado todavía. (Pressman, 2005)

Para la evaluación de la métrica del grado de validación de los requisitos se hizo una revisión de los mismos justo antes de comenzar el diseño del sistema y se comprobó que todos los requisitos funcionales y no funcionales fueron validados.

Luego, la fórmula queda así:

$$Q_v = 57 / (57 + 0)$$

$$Q_v = 1.$$

3.1.3 Resultado del análisis.

Tras analizar los resultados alcanzados en las distintas revisiones realizadas para la especificación de requisitos, se puede llegar a la conclusión de que los requisitos para el desarrollo de la herramienta SeReq carecen de ambigüedad prácticamente en su totalidad, ya que el valor de Q_e se encuentra muy cercano a 1. En cuanto a la métrica para evaluar el grado de validación, la revisión realizada arrojó que los requisitos han sido validados en su totalidad.

3.2 Métricas para el Modelo de diseño orientado a objetos.

Las métricas para el diseño proporcionan una mejor visión interna, permiten estimar la efectividad del proceso y ayudan a que el diseño evolucione a un nivel superior de calidad.

A medida que los modelos de diseño orientado a objetos van creciendo en tamaño y complejidad, es aconsejable y beneficioso tener una visión más objetiva de las características del diseño. Esto se puede lograr a través del uso de las métricas orientadas a objetos.

3.2.1 Métricas orientadas a Clases.

Una clase es la unidad principal de todo sistema orientado a objetos, por lo que las medidas y métricas para una clase individual, la jerarquía de clases y las colaboraciones de clases resultan sumamente valiosas para un ingeniero de software que necesite estimar la calidad de un diseño. (Pressman, 2005)

Las características mencionadas anteriormente han sido utilizadas como bases para el conjunto de métricas Chidamber y Kemerer (CK).

3.2.2 Métricas propuestas por Chidamber y Kemerer.

Las métricas de Chidamber y Kemerer consisten originalmente en seis métricas calculadas por cada clase:

- Métodos ponderados por clases (MPC)
- Árbol de profundidad de herencia (APH)
- Número de descendientes (NDD)
- Acoplamiento entre clases objeto (ACO)
- Respuesta para una clase (RPC)
- Carencia de cohesión de los métodos (CCM). (Pressman, 2005)

En el presente trabajo se hace uso de algunas de ellas para la validación de las clases del diseño.

Árbol de profundidad de herencia

El objetivo de esta métrica es medir la distancia desde una clase a la clase raíz del árbol de herencia. Una clase con un número de herencia muy alto tiende a incrementar los errores, por lo que se recomienda un APH de cinco o menos clases heredadas. (Pressman, 2005)

Número de descendientes

Esta métrica cuenta el número de hijos inmediatos del árbol de herencia. Un alto grado de NDD indica una alta reutilización de los métodos, ya que la herencia es una forma de reutilización; no obstante, la abstracción representada por la clase predecesora puede diluirse. Esto significa que existe una posibilidad de que algunos descendientes no sean miembros, realmente apropiados, de la clase predecesora. Una clase con muchos hijos requiere además, de mayor tiempo para las pruebas. (Pressman, 2005)

Carencia de cohesión de los métodos

Cada método dentro de una clase, accede a uno o más atributos (también llamados variables de instancia). CCM es el número de métodos que accede a uno o más de los mismos atributos. Si no existen métodos que accedan a los mismos atributos, entonces $CCM = 0$. (Pressman, 2005)

3.2.2.1 Aplicación de las métricas.

Métricas CK	Resultado del análisis
Árbol de profundidad de herencia	La longitud máxima desde la raíz tiene un valor de 1, por lo que el diseño de las clases se puede clasificar como sencillo y fácil de implementar.
Número de descendientes	El número mayor de clases hijas es 1, lo cual refleja un buen manejo de la abstracción en clases predecesoras.
Carencia de cohesión de los métodos	Luego de revisar detenidamente cada diagrama de diseño se pudo constatar que en las clases, a lo sumo existen tres métodos que acceden a un mismo atributo. Esto arroja un valor $CCM = 3$ en los casos extremos, en el resto de los casos el CCM es menor. Dado que este valor es pequeño, se llega a la conclusión de que las clases del diseño poseen un alto grado de cohesión.

Tabla 3 Aplicación de métricas CK al modelo de diseño.

Al nivel orientado a clases, la colección de métricas CK puede complementarse con las métricas propuestas por Lorenz y Kidd. (Pressman, 2005)

3.2.3 Métricas propuestas por Lorenz y Kidd.

Lorenz y Kidd separan las métricas basadas en clases en cuatro amplias categorías: tamaño, herencia, valores internos y valores externos. (Pressman, 2005)

- Tamaño de clase (TC).
- Número de operaciones redefinidas para una subclase (NOR).
- Número de operaciones añadidas por una subclase (NOA).
- Índice de especialización (IES).

En el presente trabajo se hace uso de algunas de ellas para la validación de las clases del diseño.

Número de operaciones redefinidas para una subclase.

Existen casos en que una subclase reemplaza una operación heredada de su superclase por una versión especializada para su propio uso, a lo cual se le llama redefinición. Los valores grandes para el NOR generalmente indican un problema de diseño. (Pressman, 2005)

Índice de especialización

El índice de especialización proporciona una indicación aproximada del grado de especialización para cada una de las subclases en un sistema orientado a objetos. La especialización se puede alcanzar añadiendo o eliminando operaciones, pero también redefiniendo.

$$IE = [NOR * nivel] / M_{total}$$

donde nivel corresponde al nivel en la jerarquía de clases en que reside la clase, y M_{total} es el número total de métodos de la clase. Cuanto más elevado sea el valor de IE, más probable será que la jerarquía de clases tenga clases que no se ajusten a la abstracción de la superclase. (Pressman, 2005)

3.2.3.1 Aplicación de las métricas.

Métricas Lorenz y Kidd	Resultado del análisis
Número de operaciones redefinidas para una subclase	Las clases del diseño no redefinen los métodos que heredan de las superclases. Por lo que ha sido manejada adecuadamente la abstracción representada por las superclases.
Índice de especialización	$IE = [0 * 1] / 6 = 0 \quad (1)$ $IE = [0 * 1] / 10 = 0 \quad (2)$ <p>El índice de especialización ha sido evaluado en el diseño de Atributo, tomado como ejemplo en el capítulo anterior.</p> <p>En (1) se utilizó la clase AtributoController y en (2) la clase AtributoModel.</p> <p>En el resto de las clases del diseño este valor también es cero lo cual prueba que las clases están ajustadas a la abstracción de las superclases.</p>

Tabla 4 Aplicación de métricas Lorenz y Kidd al modelo de diseño.

Conclusiones parciales

En este capítulo se han aplicado métricas a los artefactos obtenidos a través del desarrollo de la solución propuesta. El resultado es la verificación de estos artefactos, para lo cual se aplicaron métricas para comprobar la calidad de la especificación de los requisitos y de los diagramas de clases del diseño. Adicionalmente se elaboraron los prototipos de interfaz gráfica como técnica de validación de requisitos con el fin de obtener una especificación realmente completa, clara y consistente.

Tras el análisis de estos resultados se puede llegar a la conclusión de que los artefactos desarrollados en el presente trabajo de diploma han sido revisados y poseen la calidad requerida para continuar su uso en la implementación de la herramienta SeReq.

CONCLUSIONES GENERALES

La evaluación y validación de la especificación de requisitos y el modelo de diseño, demostró la validez técnica y funcional del modelado, de modo que este permita, al resto del equipo, culminar el desarrollo de la herramienta SeReq.

Siguiendo el modelo de desarrollo para la especificación de requisitos y modelado del diseño, propuesto por el proyecto ERP Cuba, es posible reutilizar componentes desarrollados previamente.

RECOMENDACIONES

Se recomienda continuar con el desarrollo de la herramienta SeReq, específicamente en los flujos de trabajo de Implementación y Pruebas, para obtener un producto concreto que resuelva las necesidades reales de los analistas del proyecto ERP Cuba.

Se recomienda integrar a la solución el componente reporteador dinámico para ampliar el conjunto de consultas posibles a los elementos de seguimiento.

Una vez terminada, se recomienda realizar una prueba piloto de la herramienta en un por ciento de las líneas de trabajo del proyecto ERP Cuba para comprobar su funcionamiento en el escenario para el cual fue concebida.

BIBLIOGRAFÍA REFERENCIADA

Anaya, Víctor and Letelier, Patricio. *SmarTTrace: Una Herramienta para Trazabilidad de Requisitos en Proyectos basados en UML.* Valencia : s.n.

—. *Trazabilidad de Requisitos Adaptada a las Necesidades del Proyecto: Un Caso de Estudio Usando Alternativamente RUP y XP.* Valencia : s.n.

Betancourt Cruz, Aracelis Reina and Vega Miniet, Yanet. 2007. *Procedimiento para desarrollar la Ingeniería de Requisitos en el proyecto Sistema de Gestión Penitenciaria.* La Habana : s.n., 2007. pp. 4-18.

Carrascoso, Yoan Arlet and Chaviano, Enrique. 2008. *Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano.* Ciudad de la Habana : s.n., 2008.

Connor, Catherine. 2005. *What's New in CaliberRM.* s.l. : Borland, 2005.

Díaz Peña, Omar Antonio. 2009. *Documento de referencia al componente de excepciones.* 2009. Especificaciones de la Arquitectura del proyecto ERP Cuba..

Energy, U. S. Department of. Abril 2000. *Guideline for Requirement Management.* abril 2000.

Gómez Baryolo, Oiner and Mena, Grette Leydi. 2008. *Arquitectura de seguridad del software ERP Cuba.* 2008. Especificaciones de la Arquitectura del proyecto ERP Cuba..

Hernán, Schenone Marcelo. 2004. *Diseño de una Metodología Ágil de Desarrollo de Software.* Buenos Aires : s.n., 2004.

Hernández, Jesús. *Ciclo de vida del Software Gestión de requisitos.*

IBM. 2009. IBM Rational Rose Enterprise. [Online] IBM, 2009. http://www-142.ibm.com/software/dre/ecatalog/Detail.wss?locale=es_ES&synkey=M221280M46834Z27.

IEEE. 1998. *IEEE Guide to Software Requirements Specifications.* 1998.

—. 1984. *IEEE Guide to Software Requirements Specifications.* 1984.

INCOSE. August 1997. *Interfacing Requirements Management Tools In The Requirements Management Process - A First Look Published in Proceedings of the Seventh International Symposium of the INCOSE.* August 1997. Vol. Volume II, A Requirements Working Group Information Report.

Ingeniería de Requisitos y Orientación a Objetos: un enfoque práctico con IRqA. SIMO. 2002. s.l. : SIMO 2002: Sesión técnica de calidad del software, 2002. Grupo de Calidad del Software: Asociación de Técnicos de Informática.

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. 2004. *El proceso unificado de desarrollo de software.* La Habana : Editorial Félix Varela, 2004. Volumen I.

Larman, Craig. 2006. *UML y Patrones.* Ciudad de la Habana : Félix Varela, 2006. pp. 25-87.

Malumbres, Ana Belén. 2008. *Visure lanza IRQA 4, la Solución Corporativa de Ingeniería de Requisitos.* Madrid : Visure Solutions, 2008.

McDonald Landazuri, Barbara A. Septiembre 2005. *Definición de Perfiles en Herramientas de Gestión de Requisitos.* Madrid : s.n., septiembre 2005.

Morales, Jose Raúl Perera. 2007. *Arquitectura de software para sistema gestion de inventarios.* Ciudad de la Habana : s.n., 2007.

Pinheiro, Francisco. Julio 2000. *Formal and Informal Aspects of Requirements Tracing.* Rio de Janeiro : s.n., julio 2000. III Workshop de Engenharia de Requisitos WER '2000.

Pressman, Roger S. 2005. *Ingenieria del Software un enfoque practico.* La Habana : Felix Varela, 2005. pp. 171-187.

—. **2005.** *Ingeniería del Software Un enfoque práctico.* Quinta edición. s.l. : Felix Varela, 2005.

Rational Software, Corporation. 2003. *Rational Unified Extended Help.* 2003. Rational Unified Process: Overview.

Reifer, Donald. 1994. *Encyclopedia of Software Engineering.* [ed.] J.J Marciniack. s.l. : Wiley, 1994. pp. 1043-1054.

Schwaber, Ken and Beedle, Mike. 2002. *Agile Software Development with Scrum.* s.l. : Prentice Hall, 2002.

Sommerville, I. and Sawyer, P. 1997. *Requirements Engineering.* s.l. : Wiley, 1997.

Thayer, R.H. and Dorfman, M. 1997. *Software Requirements Engineering.* 2da. s.l. : IEEE Computer Society Press, 1997.

UCI. 2006-2007. *Conferencia 1: FT Análisis y Diseño. Modelo de diseño.* Ciudad de La Habana : s.n., 2006-2007.

— **2007-2008.** *Conferencia 2: Fase de Inicio. Modelo del Negocio.* 2007-2008. Asignatura: Ingeniería de Software I.

Vega Miniet, Yanet. 2008. *Plantilla DCS - Especificación de Requisitos 3.0.* 2008. Equipo de Analistas Principales del proyecto ERP Cuba.

BIBLIOGRAFÍA CONSULTADA

Antonelli, Leandro and Oliveros, Alejandro. *Traceability en la Etapa de Elicitación de Requisitos.* pp. 2-5.

Davis, Alan M. 1993. *Software Requirements. Objects, Functions and States.* New Jersey : s.n., 1993.

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. 2004. *El proceso unificado de desarrollo de software.* La Habana : Editorial Félix Varela, 2004. Volumen I.

Pereiro, Daniel. *Maximize the business value of software.* s.l. : Borland. Borland Developer Studio.

Wieggers, K.E. 1999. *Software Requirements.* Redmond : s.n., 1999.

ANEXOS.

Anexo 1: Plantilla para la especificación de requisitos.

Conceptos tratados	Conceptos [Se listan los conceptos del mundo asociados al requisito].	Atributos [Se listan los atributos de los conceptos listados en la celda anterior].
Precondiciones	Precondiciones [Describe las condiciones en que deben estar los objetos de dominio antes de comenzar la realización del requisito]. [En caso de que no existan Precondiciones se escribirá “No procede.”].	Pre-requisito [Se listarán los requisitos que deben haberse ejecutado antes de comenzar a ejecutar el presente. En caso de que no existan Pre-requisitos se escribirá “No procede.”].
Descripción	[Se realizará una breve descripción del requisito en términos de su funcionalidad].	
Validaciones	[Se especificará de qué modo se validan los datos].	
Post-condiciones	[Describe las condiciones en que deben quedar los objetos de dominio al concluir la realización del requisito]. [En caso de que no existan Postcondiciones se escribirá “No procede.”].	

Post-requisito	[Se listarán los requisitos que seguidamente tendrán que ser ejecutados. En caso de que no existan Post-requisitos se escribirá “No procede.”]
-----------------------	--

Anexo 2: Prototipos de interfaz gráfica de usuario.

Este prototipo de interfaz gráfica de usuario, titulado "Gestionar tipo de elemento de seguimiento", presenta una barra de menú superior con los botones "Adicionar", "Modificar" y "Eliminar". Debajo de esta barra, se encuentran dos campos de texto etiquetados "Código:" y "Nombre:", seguidos de un botón "Buscar". El elemento central es una tabla con tres columnas: "Código", "Nombre" y "Descripción". La tabla contiene dos filas de datos: la primera con "ER" en "Código" y "Especificación de requisitos" en "Nombre"; la segunda con "CP" en "Código" y "Caso de prueba" en "Nombre". El resto de la tabla está vacío. En la parte inferior derecha del formulario, se encuentran los botones "Aceptar" y "Cancelar".

Código	Nombre	Descripción
ER	Especificación de requisitos	
CP	Caso de prueba	

Figura 11 Prototipo de interfaz Caso de Uso del Sistema Gestionar tipo de elemento de seguimiento.

Este prototipo de interfaz gráfica de usuario, titulado "Adicionar tipo de elemento de seguimiento", incluye un campo de texto "Código:" y un campo de texto "Nombre:" a la izquierda, y un campo de texto "Descripción:" a la derecha. Debajo de los campos "Código:" y "Nombre:" se encuentra un botón "Asociar/Desasociar tipo de atributo". En la parte inferior derecha del formulario, se encuentran los botones "Aceptar" y "Cancelar".

Figura 12 Prototipo de interfaz Requisito Adicionar tipo de elemento de seguimiento.

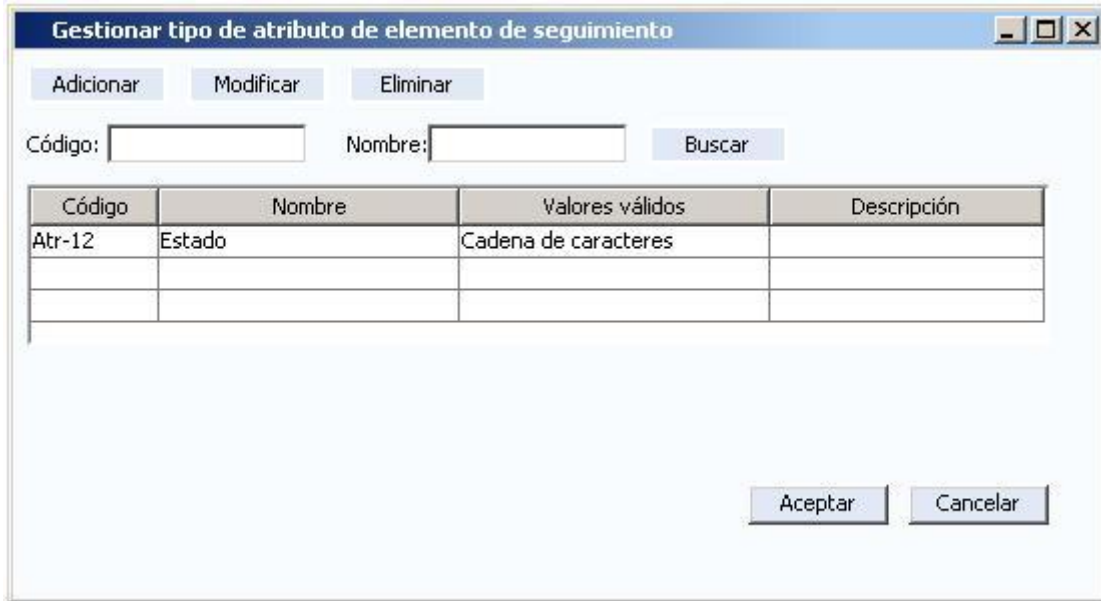


Figura 13 Prototipo de interfaz Caso de Uso del Sistema del Sistema Gestionar tipo de atributo.

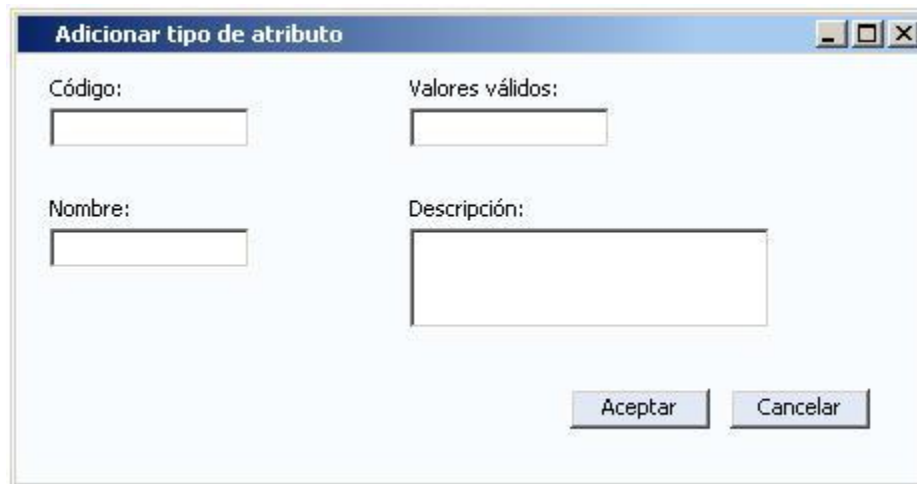


Figura 14 Prototipo de interfaz Requisito Adicionar tipo de atributo.

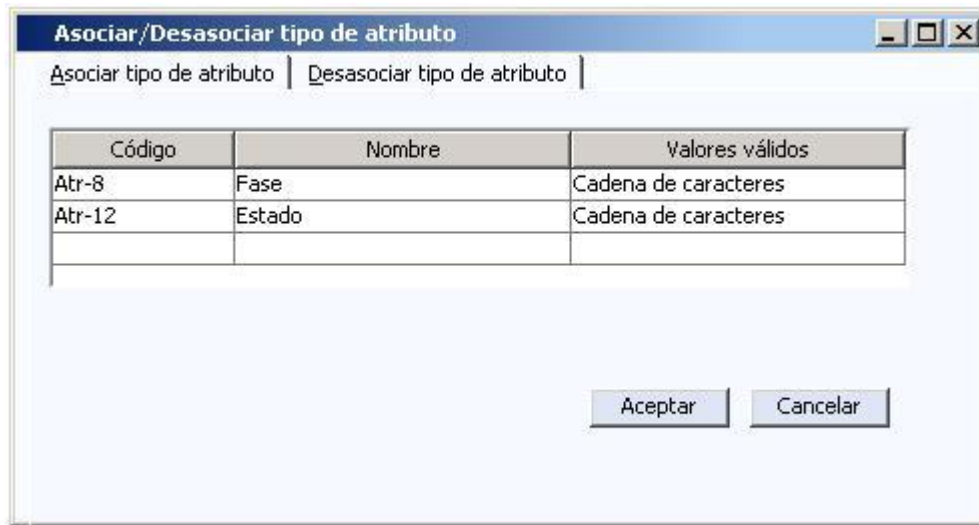


Figura 15 Prototipo de interfaz Asociar/Desasociar tipo de atributo.



Figura 16 Prototipo de interfaz Caso de Uso del Sistema Gestionar tipo de relación entre elementos de seguimiento.

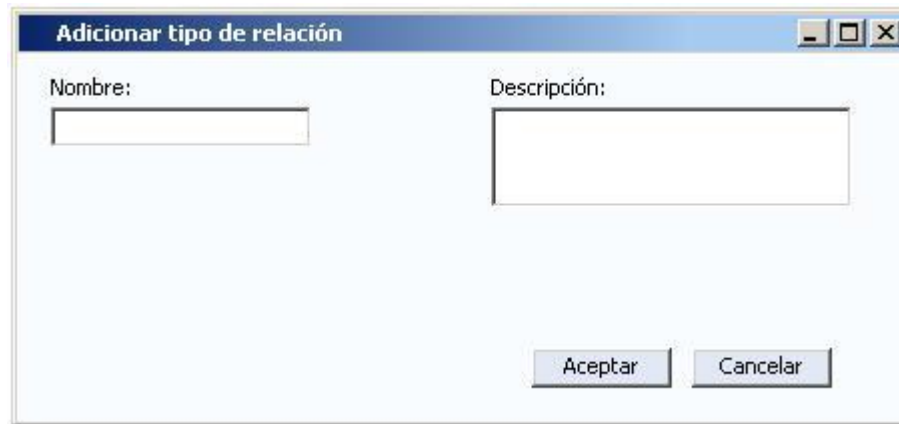


Figura 17 Prototipo de interfaz Requisito Adicionar tipo de relación entre elementos de seguimiento.

Gestionar elemento de seguimiento

Código: Nombre:
 Autor: Fecha: Tipo:

Código	Nombre	Tipo	Autor	Fecha	Descripción
ER-18	ER-Gestionar caja	Especificación de ...	Juana González	20090218	

Atributos asociados:

Código: Valor: Tipo:

Código	Tipo	Valor	Descripción
a-15	Estado	incorporado	

Figura 18 Prototipo de interfaz Caso de Uso del Sistema Gestionar elemento de seguimiento.

Adicionar elemento de seguimiento

Código:

Nombre:

Tipo de elemento:

Descripción:

Conformar equipo de trabajo

Aceptar Cancelar

Figura 19 Prototipo de interfaz Requisito Adicionar elemento de seguimiento.

Gestionar persona

Adicionar Modificar Eliminar

Nombre:

Categoría: Rol de proyecto: Buscar

Nombre	Rol de proyecto	Categoría
Rolando Suárez	Programador	Estudiante
Diana Ruiz	Arquitecto	Profesor
Jesús Morales	Analista	Estudiante

Aceptar Cancelar

Figura 20 Prototipo de interfaz Caso de Uso del Sistema Gestionar persona.

Adicionar persona

Nombre:

Rol de proyecto:

Categoría:

Aceptar Cancelar

Figura 21 Prototipo de interfaz Requisito Adicionar persona.

Conformar equipo de trabajo

Incluidos | No incluidos

Nombre	Rol de proyecto	Categoría
Rolando Suárez	Programador	Estudiante
Diana Ruiz	Arquitecto	Profesor
Jesús Morales	Analista	Estudiante

Quitar persona Cerrar

Figura 22 Prototipo de interfaz Caso de Uso del Sistema Conformar equipo de trabajo asociado a un requisito (1).

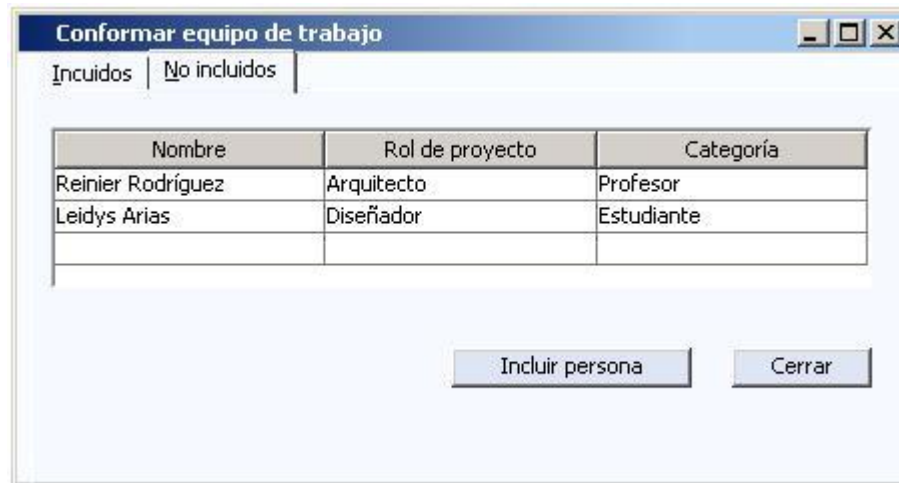


Figura 23 Prototipo de interfaz Caso de Uso del Sistema Conformar equipo de trabajo asociado a un requisito (2).

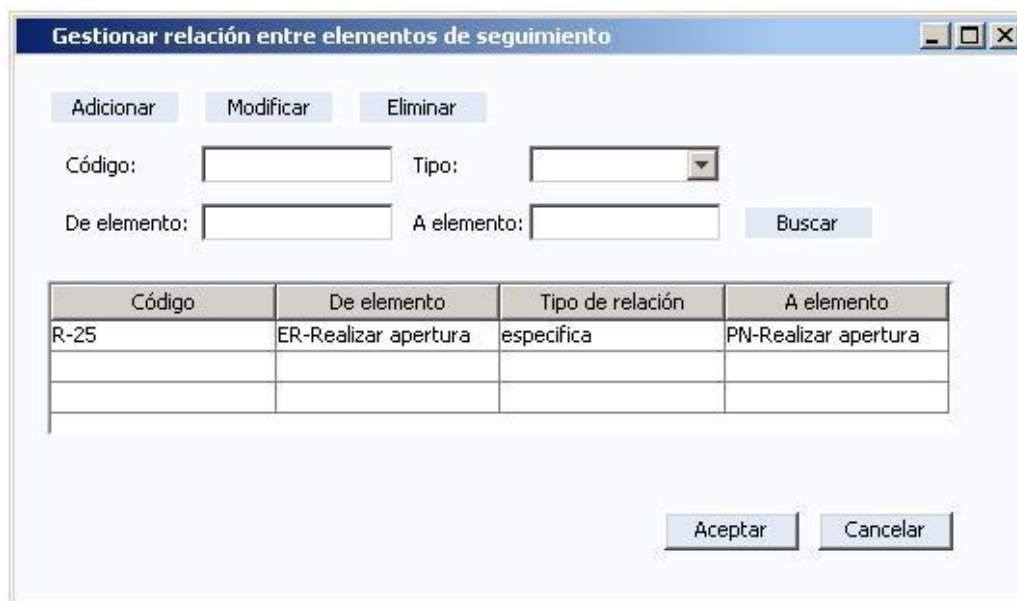


Figura 24 Prototipo de interfaz Caso de Uso del Sistema Gestionar relación entre elementos de seguimiento.

Adicionar relación entre elementos de seguimiento

Código:

De elemento: Tipo de Relación: A elemento:

Descripción:

Aceptar Cancelar

Figura 25 Prototipo de interfaz Requisito Adicionar relación entre elementos de seguimiento.

Gestionar métrica

Adicionar Modificar Eliminar

Código: Nombre: Buscar

Código	Nombre	Tipo de elemento	Descripción
M-10	Estabilidad	Especificación de requisitos	

Aceptar Cancelar

Figura 26 Prototipo de interfaz Caso de Uso del Sistema Gestionar métrica.

Adicionar métrica

Código: Tipo de elemento:

Nombre: Descripción:

Fórmula:

Valores disponibles:

Tipo de elemento	Tipo de atributo	Valor
Especificación de...	Fase	Incorporado
Especificación de...	Fase	Modificado
Especificación de...	Fase	Sin cambios
Especificación de...	Estado	Identificado
Especificación de...	Estado	Descritos

Figura 27 Prototipo de interfaz Requisito Adicionar métrica.

Realizar consulta

Seleccione el criterio por el que desea relalizar la consulta:

Según tipo de elemento de seguimiento

Tipo de elemento de seguimiento:

Tipo de atributo:

Según métrica

Nombre de la métrica:

Sobre dos elementos de seguimiento

Tipo de elemento a mostrar en las filas:

Tipo de elemento a mostrar en las columnas:

Figura 28 Prototipo de interfaz Caso de Uso del Sistema Realizar consulta.

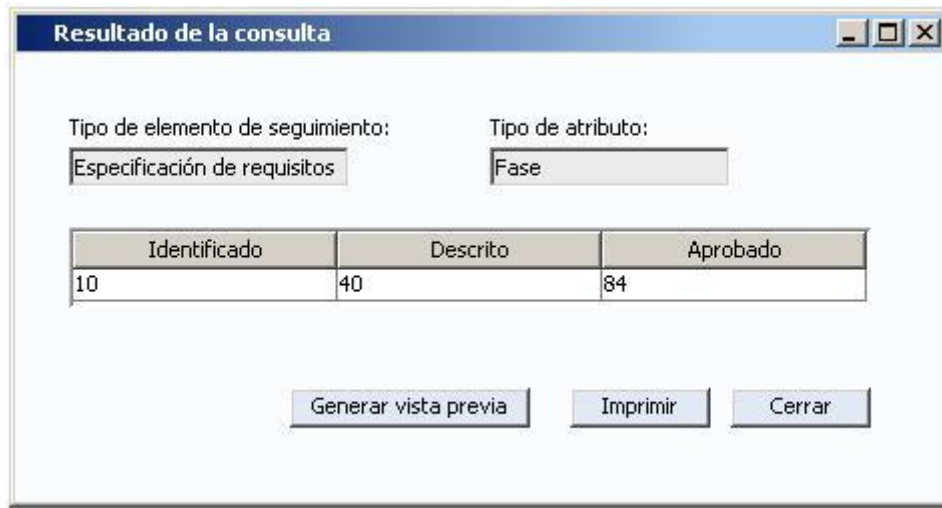


Figura 29 Prototipo de interfaz Resultado de la consulta según tipo de elemento de seguimiento.

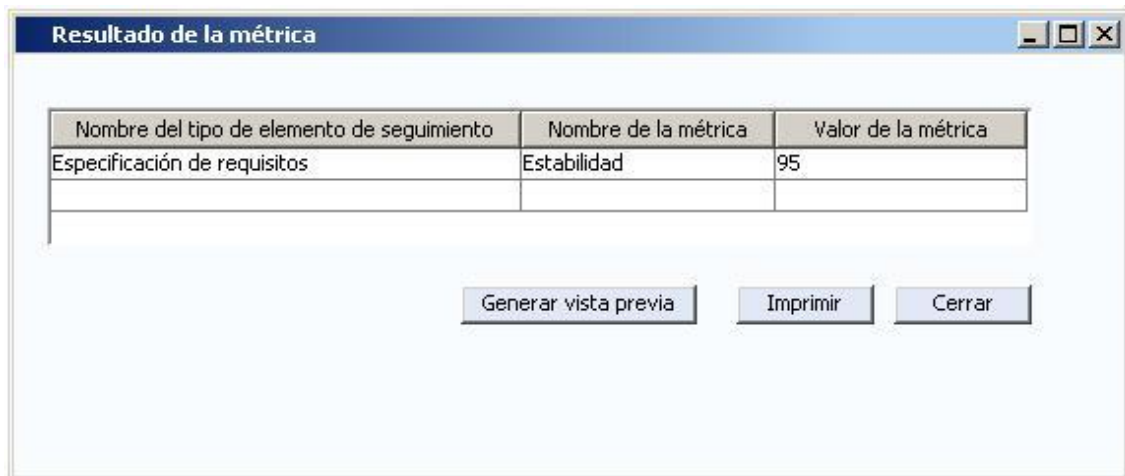


Figura 30 Prototipo de interfaz Resultado de la consulta según métrica.

The image shows a software window titled "Matriz de seguimiento". Inside the window, there is a table with the following data:

Matriz de seguimiento	ER-22	ER-23	ER-30
CP-2	X	X	
CP-4		X	
CP-8	X	X	X

Below the table, there are three buttons: "Generar vista previa", "Imprimir", and "Cerrar".

Figura 31 Prototipo de interfaz Resultado de la consulta sobre dos tipos de elemento de seguimiento (Matriz de seguimiento).

Anexo 3: Diagramas de clases de diseño.

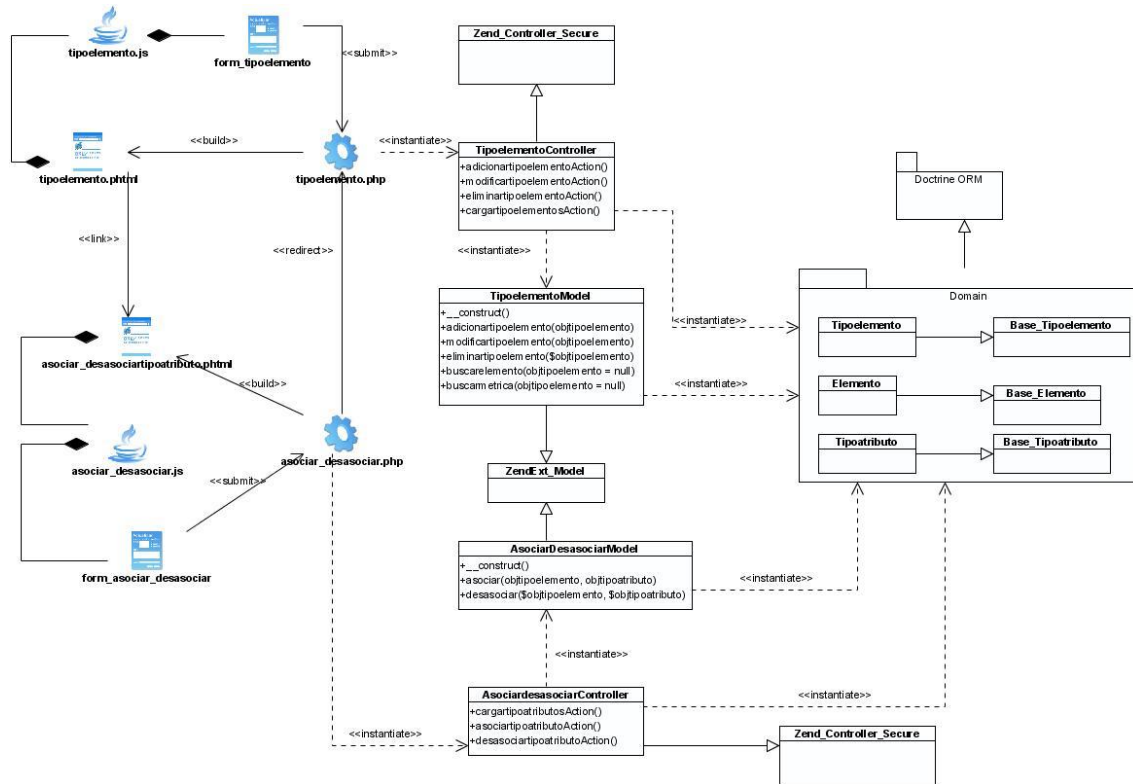


Figura 32 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar tipo de elemento de seguimiento.

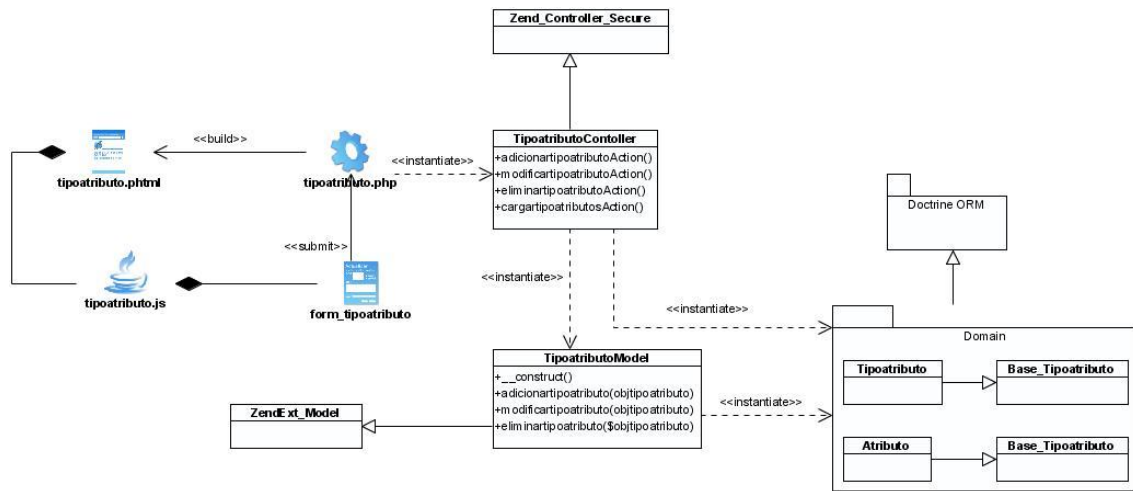


Figura 33 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar tipo de atributo.

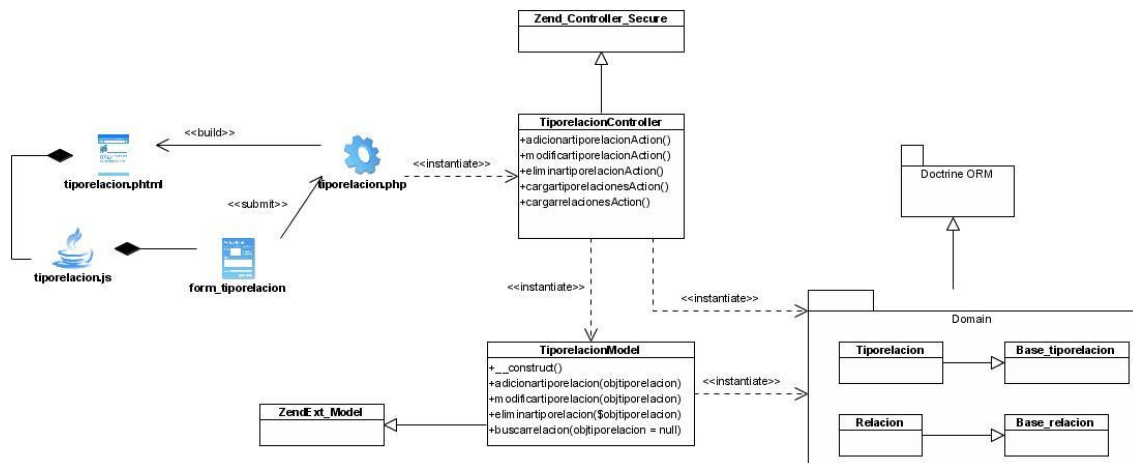


Figura 34 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar tipo de relación.

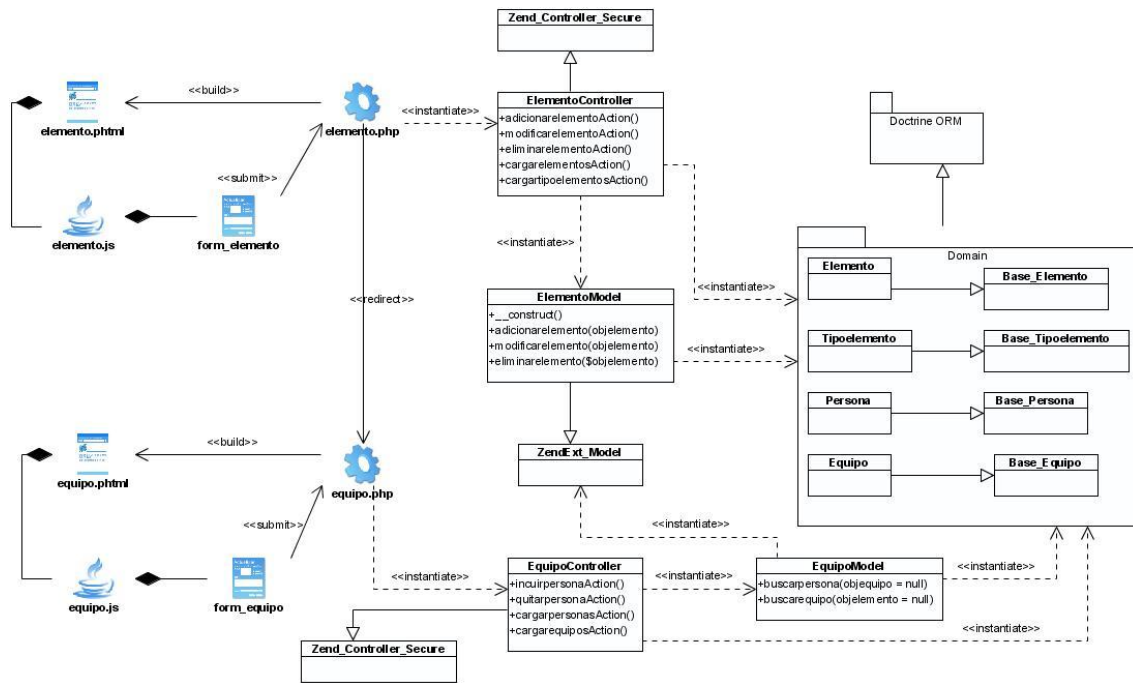


Figura 35 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar elemento.

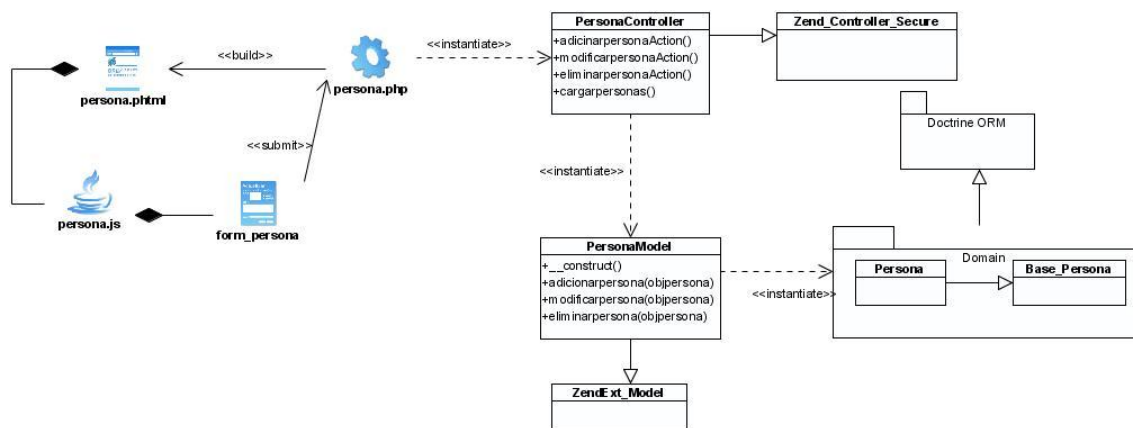


Figura 36 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar persona.

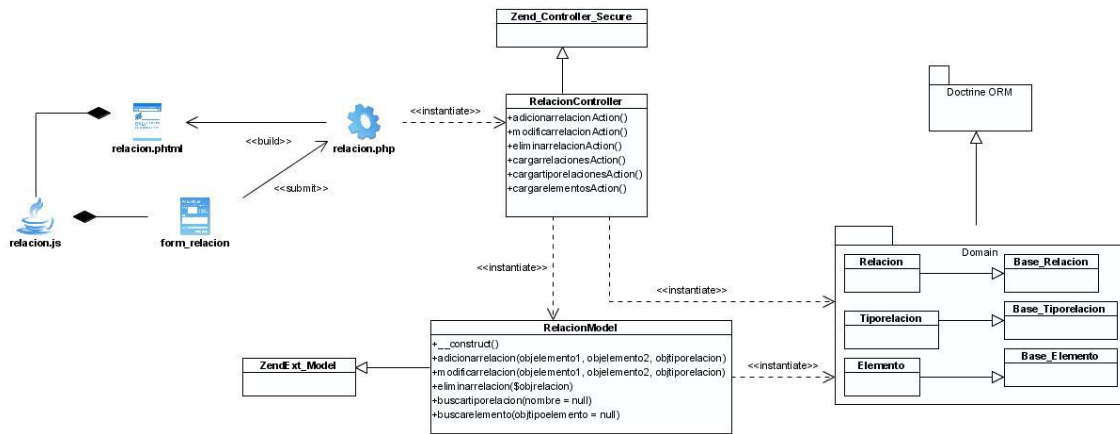


Figura 37 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar relación.

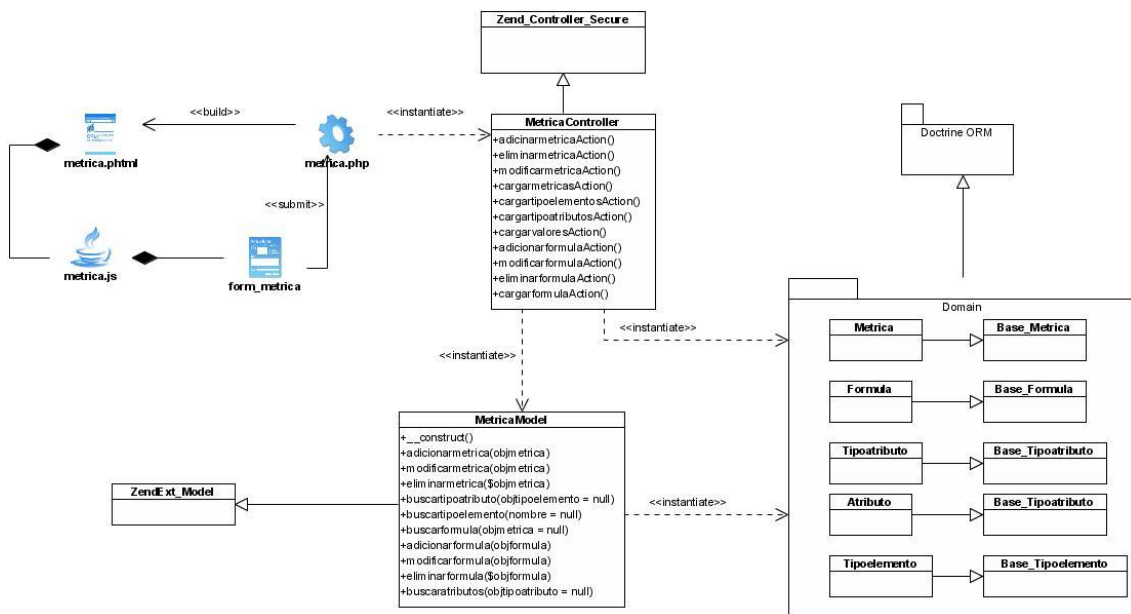


Figura 38 Diagrama de clases de diseño Caso de Uso del Sistema Gestionar métrica.

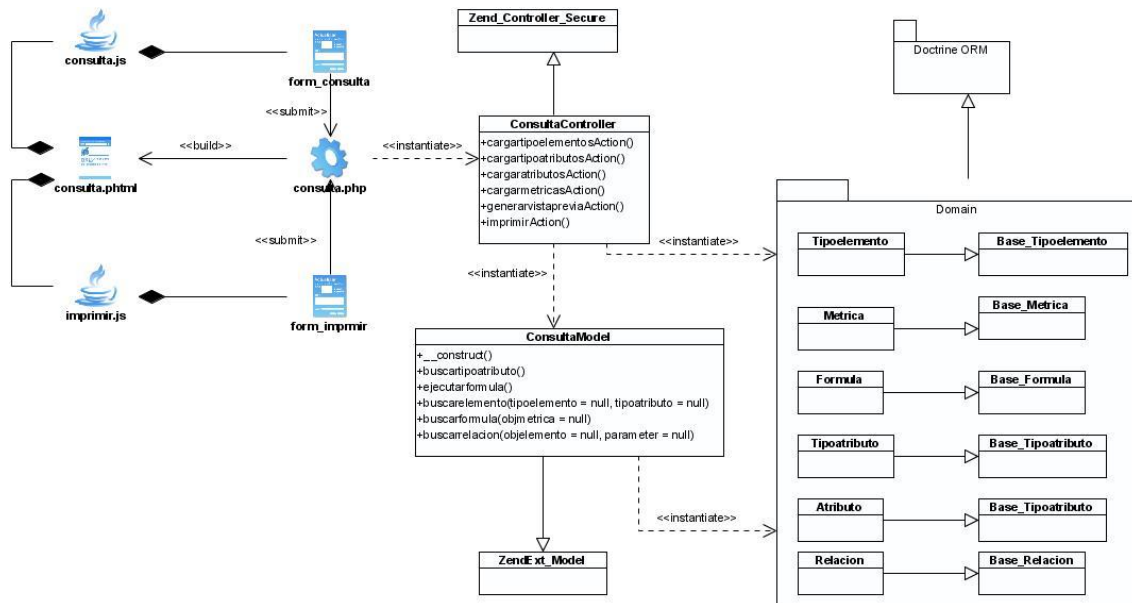


Figura 39 Diagrama de clases de diseño Caso de Uso del Sistema Realizar consulta.

Anexo 4: Descripción de las tablas del modelo de datos.

Nombre: nom_tipoatributo		
Descripción: Tabla que guarda la información relacionada con los tipos de atributos existentes en el sistema.		
Atributo	Tipo	Descripción
codigo	Numérico.	Contiene un número único que identifica el objeto.
nombre	Cadena de caracteres.	Contiene el nombre del objeto.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.
valores_validos	Cadena de caracteres.	Contiene los valores válidos que puede tomar el objeto.
idtipoatributo	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 5 Descripción de la tabla nom_tipoatributo.

Nombre: dat_atributo		
Descripción: Tabla que guarda la información relacionada con los atributos existentes en el sistema.		
Atributo	Tipo	Descripción
codigo	Numérico.	Contiene un número único que identifica el objeto.

valor	Cadena de caracteres.	Contiene el valor del atributo.
descripcion	Cadena de caracteres.	Es una columna que admite valores vacíos donde se almacena una breve descripción del objeto.
idelemento	Numérico.	Es la llave foránea que refiere a la tabla dat_elemento.
idtipoatributo	Numérico.	Es la llave foránea que refiere a la tabla dat_tipoatributo.
idatributo	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 6 Descripción de la tabla dat_atributo.

Nombre: nom_tipoatributo_nom_tipoelemento		
Descripción: Tabla que contiene la información de la relación entre las tablas nom_tipoatributo y nom_tipoelemento.		
Atributo	Tipo	Descripción
idtipoatributo	Numérico.	Es la llave foránea que refiere a la tabla dat_tipoatributo, pero a su vez es llave primaria de la tabla.
idtipoelemento	Numérico.	Es la llave foránea que refiere a la tabla dat_tipoelemento, pero a su vez es llave primaria de la tabla.

Tabla 7 Descripción de la tabla nom_tipoatributo_nom_tipoelemento.

Nombre: nom_tipoelemento		
Descripción: Tabla que guarda la información relacionada con los tipos de elementos de seguimiento existentes en el sistema.		
Atributo	Tipo	Descripción
idtipoelemento	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.
codigo	Numérico.	Contiene un número único que identifica el objeto.
nombre	Cadena de caracteres.	Contiene el nombre del objeto.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.

Tabla 8 Descripción de la tabla nom_tipoelemento.

Nombre: dat_elemento		
Descripción: Tabla que guarda la información relacionada con los elementos de seguimiento existentes en el sistema.		
Atributo	Tipo	Descripción
idelemento	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.
codigo	Numérico.	Contiene un número único que identifica el objeto.

nombre	Cadena de caracteres.	Contiene el nombre del objeto.
fecha	Fecha.	Contiene la fecha en que el elemento de seguimiento es registrado en el sistema.
autor	Cadena de caracteres.	Contiene el nombre de la persona que registra el elemento de seguimiento en el sistema. Generalmente es el autor del mismo.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.
idtipoelemento	Numérico.	Es la llave foránea que refiere a la tabla dat_tipoelemento.

Tabla 9 Descripción de la tabla dat_elemento.

Nombre: nom_tiporelacion		
Descripción: Tabla que guarda la información relacionada con los tipos de relaciones entre elementos de seguimiento existentes en el sistema.		
Atributo	Tipo	Descripción
nombre	Cadena de caracteres.	Contiene el nombre del objeto.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.
idtiporelacion	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 10 Descripción de la tabla nom_tiporelacion.

Nombre: dat_relacion		
Descripción: Tabla que guarda la información relacionada con las relaciones entre elementos de seguimiento existentes en el sistema.		
Atributo	Tipo	Descripción
idelementoa	Numérico.	Es la llave foránea que refiere a la tabla dat_elemento definida por el primer elemento de seguimiento presente en la relación.
idelementob	Numérico.	Es la llave foránea que refiere a la tabla dat_elemento definida por el segundo elemento de seguimiento presente en la relación.
idtiporelacion	Numérico.	Es la llave foránea que refiere a la tabla dat_tiporelacion.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.
idrelacion	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 11 Descripción de la tabla dat_relacion.

Nombre: dat_metrica		
Descripción: Tabla que guarda la información relacionada con las métricas existentes en el sistema.		
Atributo	Tipo	Descripción
codigo	Numérico.	Contiene un número único que identifica el

		objeto.
nombre	Cadena de caracteres.	Contiene el nombre del objeto.
descripcion	Cadena de caracteres.	Contiene una breve descripción del objeto. Admite valores nulos.
idformula	Numérico.	Es la llave foránea que refiere a la tabla dat_formula.
idmetrica	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 12 Descripción de la tabla dat_metrica.

Nombre: dat_formula		
Descripción: Tabla que guarda la información relacionada con las fórmulas existentes en el sistema.		
Atributo	Tipo	Descripción
operacion	Cadena de caracteres.	Contiene un el símbolo que defina la operación a realizar.
argumento	Numérico.	Contiene el valor de cada argumento (factor) de la fórmula.
idformula	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.
idtipoelemento	Numérico.	Es la llave foránea que refiere a la tabla nom_tipoelemento.

Tabla 13 Descripción de la tabla dat_formula.

Nombre: dat_persona		
Descripción: Tabla que guarda la información relacionada con las personas que integran los equipos de trabajo asociados a los requisitos de la herramienta existentes en el sistema.		
Atributo	Tipo	Descripción
nombre	Cadena de caracteres.	Contiene el nombre del objeto.
rol	Cadena de caracteres.	Contiene el rol en que la persona se desempeña en el proyecto.
categoria	Cadena de caracteres.	Contiene la categoría de la persona.
idpersona	Numérico.	Es la llave primaria de la tabla, es un número único que identifica el objeto.

Tabla 14 Descripción de la tabla dat_persona.

Nombre: dat_equipo		
Descripción: Tabla que guarda la información relacionada con los equipos de desarrollo asociados a un requisito existentes en el sistema.		
Atributo	Tipo	Descripción
idequipo	Cadena de caracteres.	Es la llave primaria de la tabla, es un número único que identifica el objeto.
idelemento	Cadena de caracteres.	Es la llave foránea que refiere a la tabla nom_elemento.

Tabla 15 Descripción de la tabla dat_equipo.

Nombre: dat_equipo_dat_persona		
Descripción: Tabla que contiene la información de la relación entre las tablas dat_equipo y dat_persona.		
Atributo	Tipo	Descripción
idequipo	Cadena de caracteres.	Es la llave primaria de la tabla, es un número único que identifica el objeto.
idelemento	Cadena de caracteres.	Es la llave foránea que refiere a la tabla nom_elemento.

Tabla 16 Descripción de la tabla dat_equipo_dat_persona.