

**Universidad de las Ciencias Informáticas
Facultad 5**



Título: “Herramienta para la obtención de requisitos en el desarrollo de software a distancia”.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autor(es):

Kenia González Vivero
Joan Gabriel Rodríguez Hidalgo

Tutor(es):

Ing. Dayany Díaz Corona

**Ciudad de la Habana, mayo 2009
Año del 50 Aniversario**

DATOS DE CONTACTO

Ing. Dayany Díaz Corona (ddiazc@uci.cu).

Graduada de Ingeniera en Ciencias Informática en la UCI, en el año 2007. Actualmente profesora Instructora de la Universidad de las Ciencias Informáticas, Asesora del Departamento Central de Ingeniería y Gestión de Software. Experiencia en el desempeño del rol de Analista, en proyectos del Polo de Hardware y Automática de la Facultad 5. Se encuentra cursando el Diplomado de Calidad e Ingeniería de Software.

Dedicatoria

A Nuestros Padres.

AGRADECIMIENTOS

Agradecemos:

A nuestro Comandante Fidel y Raúl y a la Universidad de Ciencias Informáticas por hacernos partícipes de este proyecto de la Revolución.

A nuestros amigos y compañeros por los momentos compartidos.

A nuestra tutora de tesis Dayany por sus consejos, su apoyo y dedicación.

De Kenia:

A mis padres por encaminarme siempre hacia los estudios, y por su amor, cariño y dedicación. A mis hermanos Maykel, y Reinier, a los que admiro y quiero mucho aunque no se los diga siempre. A mi abuela Elvira, por ser tan cariñosa y preocupada conmigo. A toda mi familia por su apoyo. A mi mejor amiga en la universidad Yanet, a Yulie y el resto de mis amigos y compañeros de la UCI. A mi amiga Chelita. A mi compañero de tesis y novio por su comprensión, su amor y entrega incondicional.

De Joan:

A mis padres por brindarme el apoyo, el amor y las fuerzas para que yo estudiara. A mis abuelos en especial a mi abuela Elida y mi abuelo René que me han enseñado mucho. A mis hermanos Rodney, Eliseo y Brayan, a todos mis tíos y mis tías y a toda mi familia por estar siempre a mi lado. A mis amigos Iriam, Yanet, Yulie y todos los de mi grupo y amigos que compartieron conmigo los mejores momentos de mi vida estudiantil. A Kenia mi compañera de tesis, de la vida y del amor, la persona que se convirtió en lo más importante de mi vida, la que me ha enseñado a pensar y a vivir la vida de otra manera, la que ha llenado de amor y ternura toda mi existencia que fue la principal inspiración para la realización de este trabajo.

RESUMEN

Luego de realizar un estudio del proceso de desarrollo de software en la Universidad de las Ciencias Informáticas, se muestra que generalmente existen problemas de comunicación entre clientes y analistas en el momento de realizar la captura de requisitos del proyecto.

El objetivo de la investigación se centra en el diseño e implementación de una aplicación web para facilitar la obtención de requisitos en el desarrollo de software a distancia para los proyectos de la Universidad de las Ciencias Informáticas, que genera un documento final o reporte con una descripción detallada de los requisitos funcionales y no funcionales, casos de uso y glosario de términos.

Este documento recoge los resultados de todo el trabajo realizado. Se lleva a cabo un análisis de las tendencias y tecnologías existentes seleccionándose las más apropiadas para desarrollar la aplicación. Además, se muestran los resultados del análisis y diseño de la propuesta del sistema, y al final se concluyen y recomiendan algunos aspectos para el mejoramiento futuro del mismo.

PALABRAS CLAVE

Herramienta para la obtención de requisitos, Reporte, Requisitos, Offshore, Ingeniería de Requisitos.

Índice

INTRODUCCIÓN	9
Capitulo 1. Fundamentación Teórica.	13
1.1 Introducción.....	13
1.2 Características de offshore	13
1.3 Ingeniería de requisitos.....	16
1.3.2 Clasificación de los requisitos.	19
1.4 Técnicas utilizadas para la captura de requisitos	22
1.5 Estudio de las aplicaciones para la gestión de requisitos existentes en el mundo.	25
1.5.1 Gatherspace.....	25
1.5.2 Raquest	26
1.5.3 Tiger Pro.....	27
1.5.4 Open Source Requeriments Management Tool (Herramienta de código abierto de administración de requerimientos).	28
1.5.5 Rational Requisite Pro	29
1.5.6 Caliber-RM	29
1.5.7 REM.....	30
1.6 Metodología utilizada.....	31
1.6.1 UML	32
1.6.2 Rational Rose	32
1.7 Arquitectura Cliente/Servidor.....	32
1.8 Servidor web: Apache	33
1.9 Análisis de los Gestores de Base de Datos	34
1.9.1 Oracle	34
1.9.2 SQL Server.....	35
1.9.3 MySQL.....	36
1.9.4 PostgreSQL.....	36
1.10 Gestor de base de datos elegido: MySQL	38

INDICE

1.11 Análisis sobre los lenguajes de programación.....	38
1.11.1 HTML.....	38
1.11.2 Java Script.....	39
1.11.3 CSS.....	39
1.11.4 ASP.NET.....	40
1.11.5 PHP.....	42
1.11.6 Perl.....	43
1.11.7 Ajax.....	43
1.12 Lenguaje seleccionado: PHP.....	44
1.13 Integración PHP, MySQL y Apache.....	44
1.14 Herramientas Utilizadas.....	44
Capítulo 2. Características del sistema.....	46
2.1 Introducción.....	46
2.2 Descripción de los procesos del negocio propuestos.....	46
2.3 Modelo del dominio.....	47
2.3.1 Descripción del modelo del dominio.....	48
2.4 Solución propuesta.....	49
2.5 Requisitos Funcionales.....	50
2.6 Requisitos no Funcionales.....	54
2.7 Modelo de casos de uso del sistema.....	55
2.7.1 Definición de los actores del sistema.....	56
2.7.2 Definición de los casos de usos del sistema.....	57
2.7.3 Descripción de los casos de uso del sistema.....	61
2.7.4 Expansión de los Casos de Uso.....	67
Capitulo 3. Desarrollo de la solución propuesta.....	115
3.1 Introducción.....	115
3.2 Patrones de diseño.....	115
3.3 Patrón Modelo Vista Controlador.....	115

INDICE

3.4 Diagrama de clases con extensiones Web	117
3.5. Diseño de la BD.....	128
3.5.1. Diagrama de clases persistentes	128
3.5.2. Modelo de Datos.....	129
3.6 Seguridad	130
3.7 Principios de diseño.	131
3.7.1 Estándares en la interfaz de la aplicación.....	131
3.7.2 Formato de los reportes.....	132
3.7.3 Tratamiento de excepciones.....	132
3.7.4 Estándares de codificación.....	133
3.8 Diagrama de despliegue.....	133
CONCLUSIONES	135
RECOMENDACIONES	136
REFERENCIAS BIBLIOGRÁFICAS.....	137
BIBLIOGRAFÍA	139
ANEXOS.....	140

INTRODUCCIÓN

En nuestros días el desarrollo del software se encuentra en un nivel alto en correspondencia con el avance tecnológico, la gran mayoría de los sistemas realizados en el mundo se acercan bastante a las necesidades de los usuarios; aunque, en muchos casos, los clientes no quedan del todo satisfechos con el producto final, debido a la falta de comunicación entre desarrolladores y clientes cuando hay que determinar las características que va a tener el producto, lo que trae consigo que exista un desnivel entre la reputación de los desarrolladores y el sistema que realicen.

La mayoría de los grupos de programadores siguen metodologías para la implementación de sus sistemas, dichas metodologías contienen diferentes fases que ayudan a organizar el trabajo. La fase de Requerimientos se destaca porque en ella se conocen las características que debe tener el software a desarrollar.

Actualmente existen herramientas que permiten la gestión de requisitos en los sistemas, generalmente son realizadas con el fin de brindar apoyo y ayudar al entendimiento entre usuarios y desarrolladores.

Muchas veces el trabajo realizado utilizando desarrollo a distancia es rechazado por inconformidad del cliente o porque no se hace un trabajo eficiente cuando se realiza la captura de requisitos; esto se debe a los problemas de comunicación que existen entre los desarrolladores, clientes y usuarios finales del producto de software. Otros factores como falta de participación del usuario, requisitos incompletos y el cambio de los requisitos influyen en gran medida en los fracasos del desarrollo de software. Según informes del SEI (Software Engineering Institute, Instituto de Ingeniería del Software), alrededor de un 50% de los problemas que sufren nuestros desarrollos están originados en una deficiente etapa de gestión y definición de requisitos; sobre el 45% del esfuerzo total de un proyecto se debe a los costes que implican los requisitos defectuosos; por último, se destaca que un problema en la fase de requisitos, si no es encontrado a tiempo, puede costar entre 5 y 200 veces más esfuerzo solucionarlo que si se hubiese detectado a tiempo.

La Universidad de las Ciencias Informáticas (UCI), es un centro que viene ganando en experiencia desde el punto de vista de desarrollo de aplicaciones, lo que viene dado por el trabajo de los diferentes proyectos, donde prevalece la organización y el nivel de conocimiento del equipo.

En la UCI se desarrollan diferentes proyectos, donde los clientes en la mayoría de los casos se encuentran en el extranjero o fuera de la provincia, y en muchas ocasiones gestionan los requisitos a través de correos electrónicos, chat, entre otros. Esto implica que la comunicación entre los clientes y los desarrolladores no sea la mejor, además puede provocar atrasos en los cronogramas y conjuntamente dilación en la terminación del producto, en resumen, las expectativas de los clientes pueden no cumplirse. Definir procedimientos que establezcan los pasos y los análisis que se realizarán antes de aceptar los cambios propuestos, cambiar los atributos de los requisitos afectados, son actividades que están presentes en la gestión de requisitos que desarrollan los grupos de proyectos en la elaboración de un sistema. Cuando se realiza software a distancia (offshore) la definición y gestión de Requisitos es mucho más compleja que si se realizara de forma profesional, porque generalmente los usuarios se muestran inconformes con el producto final, debido a que las expectativas y necesidades de los mismos no fueron captadas totalmente.

De ahí el surgimiento de nuestro trabajo como necesidad de dar respuesta a las situaciones antes expuestas, por lo que el **problema a resolver** queda formulado de la siguiente forma: **¿Cómo lograr que la determinación de los requisitos en el desarrollo de software a distancia se realice de forma eficiente?**

Para ello el **objeto de estudio** lo constituye la obtención de requisitos en sistemas realizados a distancia y el **campo de acción** las herramientas para realizar la obtención de requisitos en sistemas realizados a distancia en la Universidad de las Ciencias informáticas.

Esta investigación tiene como propósito **defender la idea** de que si se implementa una herramienta que permita realizar la obtención de requisitos en los proyectos de la UCI, que desarrollan software a distancia, se facilitaría el proceso de levantamiento de requisitos de los analistas que trabajan en estos proyectos, disminuyendo el número de inconformidades en el producto final por parte de los clientes y a su vez, haciendo un software que cumpla con sus expectativas.

Para dar respuesta al problema de la investigación se definió que el **Objetivo General** consistiera en: Desarrollar la implementación de una herramienta para realizar la obtención de requisitos de sistemas desarrollados a distancia.

Para cumplir con el objetivo se plantearon las siguientes tareas investigativas:

- Investigar sobre el estado del arte de herramientas para la Obtención de Requisitos que sean de código abierto.
- Investigar sobre el estado del arte de las herramientas que se utilicen para realizar la obtención de requisitos en el desarrollo a distancia.
- Analizar y caracterizar cómo se encuentran en el ámbito internacional las tecnologías que se utilizan para llevar a cabo herramientas como la que se pretende desarrollar.
- Seleccionar la Metodología de Análisis y Diseño de Sistemas Informáticos, que facilite la creación y garantice la calidad del sistema.
- Diseñar una base de datos que soporte las funcionalidades del sistema.
- Implementar una aplicación Web.
- Iniciar el proceso de desarrollo hasta que el producto se encuentre con una capacidad operativa inicial.

La culminación de esta herramienta, permitirá a los diferentes proyectos de la Universidad de las Ciencias Informáticas realizar la captura de requisitos sin limitaciones de distancia ni de comunicación.

Para el desarrollo de este trabajo la investigación quedará estructurada de la siguiente manera:

Capítulo 1: Describe los conceptos básicos en los que se basa el trabajo que permiten un mejor entendimiento del mismo, también se abordará sobre la existencia y características de aplicaciones que permitan la captura de requisitos a distancia, además se realizará un estudio de las herramientas y procedimientos que serán definidos para la realización de este trabajo de diploma.

Capítulo 2: Describe el negocio a través de un modelo de Dominio, que sirve de base para determinar qué es lo que se va a desarrollar. Además, la descripción de la aplicación que se propone, así como de todos los requisitos funcionales y no funcionales que esta debe cumplir.

Capítulo 3: Describe el desarrollo de la solución propuesta, se modelan los diagramas de clases de diseño, los diagramas de interacción, el diagrama de clases persistentes, se plantea el modelo de datos, se especifican los patrones de diseño utilizados, así como los principios para el diseño gráfico y el diagrama de despliegue.

Capítulo 1. Fundamentación Teórica.

1.1 Introducción

En este capítulo se abordarán los principales conceptos que servirán para un mejor entendimiento del tema de esta investigación. Además se realiza un estudio exhaustivo de las principales herramientas utilizadas en el mundo para la gestión de requisitos, con el objetivo de resumir las funcionalidades y características que estas poseen. Por último se realiza una caracterización de las tecnologías, herramientas y lenguajes para definir y seleccionar las que se utilizarán en el desarrollo de este trabajo de diploma.

1.2 Características de offshore

Offshore es un término del idioma inglés que significa "en el mar, alejado de la costa". En informática, el término se aplica al recorte de gastos mediante la externalización del desarrollo de software, empleando a bajo coste a programadores de países extranjeros con menor renta.

Se conoce como offshore la práctica de subcontratar servicios a empresas localizadas en países cuya mano de obra es más barata que el país en el que se sitúa la compañía que externaliza dichos servicios.

Gracias a las mejoras en telecomunicaciones, la externalización a otros países está afectando tanto a la industria como a los servicios. Las tecnologías de la información y la comunicación son especialmente fáciles de deslocalizar puesto que el producto o servicio puede transmitirse instantáneamente por las redes. De ahí que surja la necesidad de crear herramientas para la gestión del desarrollo de las aplicaciones offshore, o sea, que se realizan a distancia como la que se pretende desarrollar en este trabajo de diploma.

La externalización de servicios informáticos permite variar los costes fijos destinados al desarrollo y mantenimiento de los sistemas informáticos, liberar recursos para desarrollar iniciativas estratégicas y

acceder a especialistas de los que no se dispone internamente. Sin embargo, la relativa novedad del mercado del offshore y la falta de experiencia propia de los directores de operaciones y responsables de informática pone de manifiesto cierta inseguridad a la hora de definir la estrategia más adecuada para la empresa.

Los temores principales como, por ejemplo, la pérdida de control se pueden minimizar estableciendo un proceso de selección detallado del proveedor de servicios externo, poniendo mucho cuidado en la negociación final a la hora de redactar el contrato y los niveles de servicio esperados y formando un equipo especializado en gestionar a los proveedores y monitorizar el cumplimiento de los niveles de servicio contratados. [1]

Ventajas del offshore:

- Nivel de servicios y calidad igual o superior. Acceso a recursos inmediatamente disponibles y altamente cualificados.
- Disminución real de los costes, especialmente en actividades de trabajo intenso.
- Oportunidad para que las empresas se centren en competencias claves traspasando las actividades de menor importancia a terceros.
- Desplazamiento de capacidades y aptitudes.
- Escalabilidad.
- Turnos de noche y velocidad en la entrega.
- Necesidades de despliegue global y desarrollo de negocio.
- Mitigación del riesgo geográfico.

Riesgos que tienden a reducirse [2]:

- Riesgo geopolítico.

- Riesgos legales.
 - Carencia de protección de la propiedad intelectual.
 - Aspectos jurisdiccionales: dificultad para hacer cumplir los contratos en otros países.
 - Riesgos sobre la seguridad de la información
- Capacidades técnicas vs. Conocimiento del negocio.
- Riesgo cultural e idiomático.

A continuación se indican algunos factores claves que los responsables de la toma de decisiones en las empresas deberán analizar cuando se planteen la posibilidad de abrazar el modelo offshore:

- **Coste.** Considere el coste total del proyecto, incluyendo el personal, la infraestructura física y de sistemas, las telecomunicaciones, las licencias de software, los desplazamientos, los costes efectivos del riesgo y las diferencias de productividad, y los gastos indirectos de gestión.
- **Calidad.** Dese cuenta que la calidad de todo el proyecto –no sólo la del código- es clave.
- **Capacidad del proceso.** Gestione la totalidad de los procesos por todas las ubicaciones e intégreolos con la mayor perfección posible.
- **Protección del capital intelectual y los conocimientos de la empresa.** Proteja los conocimientos propios que puedan ser valiosos, la capacidad del proceso en curso y la experiencia empresarial. Asegúrese –mediante procesos formales y planificados– de que los conocimientos importantes y la experiencia propia permanecen en su empresa (o vuelven a ella, por ejemplo, utilizando portales de gestión del conocimiento).
- **Disponibilidad de talento.** Evalúe el número de trabajadores disponibles, la capacidad del sistema educativo “del país huésped”, al que se externalizará el proyecto, para producir trabajadores cualificados y la magnitud de las aptitudes disponibles.

- **Comprensión de los requisitos y cultura de las empresas locales.** Plantéese sus propias afinidades y diferencias con el país huésped, porque algunas culturas requieren un mayor grado de gestión de las relaciones que otras.
- **Riesgo.** Un plan de toma de decisiones bien equilibrado debería comparar el ahorro en los costes con, por ejemplo, un aumento potencial de los riesgos para llegar a un cálculo de ahorro de costes ajustado al riesgo capaz de evaluar el rendimiento real.
- **Malas relaciones con los empleados.** Prevea que los empleados verán el movimiento de puestos de trabajo fuera del país como una amenaza tangible para su bienestar, y diseñe planes para tratar con antelación esta percepción.

Pese a sus incuestionables ventajas, la externalización offshore no es siempre la mejor opción. El proceso de toma de decisiones al respecto se hará aún más complejo cuando el concepto de offshore se convierta en la realización global de proyectos globales, una transformación que ya está en camino. De hecho, uno de los aspectos más problemáticos y delicados que afrontan los responsables de las tecnologías de la información de muchas empresas es si externalizar determinados proyectos recurriendo al offshore o realizarlos en su propio país. Actualmente usar el offshore es abrirse al mundo y saber aprovechar lo mejor de cada país.

1.3 Ingeniería de requisitos

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir.

Muchos autores han expuesto su criterio al respecto, por lo que en la literatura se pueden encontrar múltiples definiciones. A continuación se citan algunas de ellas.

Según Christel y Kang, la Ingeniería de Requisitos se define como:

a) Aplicación disciplinada de principios científicos y técnicas para desarrollar, comunicar y gestionar requisitos. [3]

b) El proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del conocimiento obtenido. [4]

Amador Durán afirma que: La ingeniería de Requisitos puede considerarse como un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades. [4]

Según Presman: "... un proceso de descubrimiento, refinamiento, modelado y especificación". 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. El proceso de ingeniería de requisitos puede ser descrito en 5 pasos distintos: Identificación de Requisitos, Análisis de Requisitos y Negociación, Especificación de Requisitos, Modelado del Sistema, Validación de Requisitos y Gestión de Requisitos.

Para describir completamente el sistema en cuanto a qué funciones va a realizar, se desarrollan un conjunto de artefactos, sugiere los siguientes como fuentes de información importantes: el documento visión, el modelo de casos de uso, los casos de uso, los requisitos del sistema y especificaciones suplementarias, mostrando todos los involucrados, incluyendo clientes y usuarios potenciales.

Amador Durán asegura que el aspecto más importante en este proceso es la *comunicación*. Esta característica es la que hace la gestión de requisitos una disciplina especialmente compleja ya que el *factor humano* juega un importante y decisivo papel, por lo que este proceso no sólo tendrá aspectos técnicos sino sociales y culturales. [4] Es por ello que es recomendable utilizar determinadas técnicas para enfrentarse a las diferentes situaciones que se pueden encontrar. En el epígrafe 1.7 se aborda el tema.

1.3.1 Características de los requisitos

Las características de un requerimiento son sus propiedades principales. A continuación se muestran las más relevantes.

a) Correcto: Una especificación de requisitos es correcta si, y sólo si, cada requisito declarado se encuentra en el software. No hay ninguna herramienta o procedimiento que aseguran la exactitud. Alternativamente el cliente o el usuario pueden determinar si el SRS refleja las necesidades reales correctamente. Identificando los requerimientos hace este procedimiento más fácil y hay menos probabilidad al error.

b) Inequívoco: Un SRS es inequívoco si, y sólo si, cada requisito declarado tiene sólo una interpretación. Como un mínimo, se requiere que cada característica de la última versión del producto se describa usando un único término. En casos dónde un término en un contexto particular tenga significados múltiples, el término debe ser incluido en un glosario dónde su significado es hecho más específico.

c) Completo: Un SRS está completo si, y sólo si, incluye los elementos siguientes:

a) Los requisitos están relacionados a la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas. En particular debe reconocerse cualquier requisito externo impuesto por una especificación del sistema y debe tratarse.

b) La definición de las respuestas del software a todos los posibles datos de la entrada del sistema y a toda clase de situaciones. Una nota que es importante especificar son las contestaciones a las entradas válidas e inválidas a ciertos valores.

c) Tener todas las etiquetas llenas y referencias a todas las figuras, tablas, diagramas en el SRS y definición de todas las condiciones y unidades de medida.

d) Consistente: La consistencia se refiere a la consistencia interior. Si un SRS no está de acuerdo con algún documento del superior-nivel, como una especificación de requisitos de sistema, entonces no es correcto.

e) Comprobable: Un SRS es comprobable si, y sólo si, cada requisito declarado es comprobable. Un requisito es comprobable si, y sólo si, allí existe algún proceso rentable finito con que una persona o la máquina puede verificar que el producto del software reúne el requisito. En general cualquier requisito ambiguo no es comprobable.

f) Modificable: Un SRS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requisitos fácilmente, completamente y de forma consistente mientras conserva la estructura y estilo.

g) Identificable: Un SRS es identificable si el origen de cada uno de sus requisitos está claro y si facilita las referencias de cada requisito en el desarrollo futuro o documentación del mismo. [5]

1.3.2 Clasificación de los requisitos.

Durante el proceso de obtención de requisitos de un proyecto de software surgen muchas ideas por parte de los clientes, usuarios y equipo de desarrollo acerca de lo que debe de hacer el sistema, estas ideas son analizadas como candidatas a requisitos. Los requisitos a su vez son clasificados según su naturaleza en dos grandes categorías: Requisitos Funcionales y Requisitos No Funcionales.

Amador Durán plantea que es conveniente separar dentro de los requisitos funcionales, los requisitos de datos o de almacenamiento de información, que establecen que la información debe almacenar el sistema por ser relevante para las necesidades y objetivos de los clientes y usuarios.

Requisitos Funcionales

Los Requisitos funcionales especifican acciones que el sistema debe ser capaz de cumplir, sin tomar en cuenta algún tipo de restricción física. Por lo general se describen a través de casos de uso, obteniéndose las actividades que serán objeto de automatización. Estas actividades son el punto de partida para identificar qué debe hacer el sistema.

Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta.

- **Requerimientos de Software**
Debe mencionarse el software del que se debe disponer, por ejemplo: Sistema Operativo Windows 95 o Superior; Máquina Virtual de Java versión 1.3 o Superior; etc.
- **Requerimientos de Hardware**
Al igual que en la sección anterior enunciar aquí los elementos de hardware de que se disponen, por ejemplo: se requiere disponer de un MODEM estándar o una tarjeta digitalizadora de video, etc.
- **Restricciones en el diseño y la implementación**
Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Ejemplos de ellas son:
 - Estándares requeridos.
 - Lenguajes de programación a ser usados para la implementación.
 - Uso obligatorio de ciertas herramientas de desarrollo.
 - Restricciones en la arquitectura o el diseño.
 - Bibliotecas de clases.
- **Requerimientos de apariencia o interfaz externa.**
Este tipo de requerimiento describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto.

Los requerimientos de apariencia se vuelven más importantes a medida que los productos de software se mueven hacia áreas más orientadas al consumidor. Productos muy sofisticados como las cámaras digitales, cámaras de video u organizadores personales les dan la impresión a los clientes de ser muy fáciles de operar, sin embargo contienen un gran nivel de funcionalidad.

Requerimientos de apariencia también pueden ser necesidades de cumplir con normas estándares como por ejemplo las interfaces de sistemas tipo Windows/Apple, o con los estándares de la empresa para la cual se esté desarrollando el software.

- **Requerimientos de Seguridad**

Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:

- **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
- **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
- **Disponibilidad:** Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

- **Requerimientos de Usabilidad**

Estos requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello debe revisarse la especificaciones de los perfiles de usuarios y echarle un vistazo a las clasificaciones de sus niveles de experiencia.

Debe conocerse:

- ¿Qué tipo de personas son?
 - ¿Qué tipo de producto necesitan para realizar su trabajo?
 - ¿Qué tipo de requerimiento de usabilidad haría el producto adecuado para ellos?
 - Los requerimientos de usabilidad se derivan de una combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo, estos elementos deben tenerse claro antes de escribirlos.
- **Requerimientos de Soporte**

Abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este así como lograr su mejoramiento progresivo y evolución en el tiempo. Pueden incluir: Pruebas, Extensibilidad, Adaptabilidad, Mantenimiento, Compatibilidad, Configuración, Servicios, Instalación, Internacionalización y Requerimientos de Portabilidad. [5]

1.4 Técnicas utilizadas para la captura de requisitos

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

A continuación se presentan un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software:

Entrevistas: Resulta una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización.

Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista).

A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar. Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Aquí desempeña un papel fundamental la preparación de la entrevista.

JAD (Joint Application Development/Desarrollo conjunto de aplicaciones): Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación.

Durante la sesión se discute en grupo sobre estos temas, llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema.

Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ahorrando tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

Brainstorming (Tormenta de ideas): Es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador.

Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como éste.

Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

Concept Mapping (Mapas Conceptuales): Los mapas conceptuales son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el

usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy subjetivos y pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual.

Sketches y Storyboards: Esta técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno web. La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard).

Casos de Uso: Aunque inicialmente se desarrollaron como técnica para la definición de requisitos, algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los *actores* del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades.

Cuestionarios y Checklists(Listas de control): Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

Comparación de terminología: Uno de los problemas que surge durante la captura de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada

en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos.

Existen más técnicas para la captura de requisitos (análisis de otros sistemas, estudio de la documentación, etc.), incluso también es común encontrar alternativas que combinen varias de estas técnicas. Sin embargo, las presentadas ofrecen un conjunto representativo de las más utilizadas y resultan suficientes para el objetivo de este trabajo. [6]

1.5 Estudio de las aplicaciones para la gestión de requisitos existentes en el mundo.

La captura de requisitos en el desarrollo de los proyectos de nuestra universidad, es un área de proceso clave para asegurar el éxito de los proyectos, y la mayoría de las veces se realiza este trabajo a distancia, lo que dificulta el buen funcionamiento de esta actividad. Cuanto mejor se comunique y administre los requisitos, mejor será la oportunidad que tendrán los proyectos para brindar la solución correcta a tiempo. En la actualidad, existen aplicaciones que se utilizan para realizar la captura de requisitos a distancia. A continuación se realiza un análisis del estado en que se encuentra el desarrollo de las principales herramientas de este tipo, en el mundo.

1.5.1 Gatherspace

Es una herramienta que gestiona toda la fase de análisis de un proyecto mediante una aplicación Web. Se accede a ella a través de un navegador. Es una herramienta usada por empresas como Siemens o AT&T y ofrece gran cantidad de documentos de prueba y tutoriales.

Ventajas de Gatherspace

- Al ser una herramienta Web, no necesita instalación y es portable a cualquier sistema operativo.
- Ofrece un sistema para indicar los actores del proyecto y modelar los casos de uso.
- Genera informes automáticamente con los requerimientos especificados.

- Al ser una herramienta Web permite que varias personas trabajen en un mismo proyecto a la vez, sin que existan problemas de concurrencia, ya que se puede establecer una jerarquía de usuarios del proyecto.
- Toda la información con la que trabaja se puede exportar a un disco local en cualquier momento en forma de fichero (con restricciones, al ser una versión de prueba). Esto facilita la documentación.

Desventajas de Gatherspace

- Es una herramienta comercial.
- Su versión de prueba está muy limitada en cuanto a opciones. Por ejemplo, los cambios que se hacen en los requerimientos a lo largo del proyecto solo se pueden visualizar en los informes, y no en la pantalla de captura de datos. Lo mismo sucede con los casos de uso.
- Obliga a estructurar los requisitos en paquetes, que son los contenedores que usa la herramienta.

Aunque se trata de una herramienta muy completa, sus limitaciones en la versión de prueba hacen bastante tedioso el trabajar con ella. Por otra parte es fácil de manejar y se encarga de generar informes de manera automática en varios formatos (entre ellos .pdf) que pueden ser descargados al disco. Su gran potencial reside en ser usada en equipos de desarrollo distribuidos geográficamente, y esto se debe precisamente a que es una aplicación Web.

1.5.2 Raquest

Se trata de la herramienta de gestión de requisitos de Enterprise Architect. Es una herramienta comercial que ofrece servicio de prueba de 30 días. Solo está disponible para la plataforma Windows, y además requiere tener instalado Enterprise Architect para poder usarse.

Ventajas de Raquest:

- Es una herramienta comercial muy completa, pensada para usarse en proyectos grandes. Por ejemplo, se puede conectar un archivo de especificación de requerimientos con el repositorio de la empresa.
- Es muy fácil de manejar, y posee unos menús muy intuitivos.
- Los proyectos se pueden exportar a un gran número de tipos de fichero, entre ellos HTML, formato Word (.doc) y Excel (.xls). También permite conectarse a una impresora virtual para crear informes de requerimientos en este formato.
- La herramienta es muy adaptable, por ejemplo, se pueden visualizar los requisitos en función de su tipo, en forma de árbol, en forma de matriz.

Desventajas de Raquest:

- Solamente se centra en la fase de requisitos, sin embargo, Enterprise Architect ofrece otras aplicaciones para el modelado otras partes del proyecto, por ejemplo, para modelado de casos de uso.
- La versión de prueba esta algo limitada, por ejemplo, en definir nuestros propios atributos, o a la hora de crear relaciones entre requisitos.
- Necesita tener instalado Enterprise Architect.

1.5.3 Tiger Pro

Tiger Pro es una herramienta para la definición de requisitos con licencia Shareware. Solamente está disponible para la plataforma Windows. En su página Web se presenta como una herramienta para ayudar a escribir de forma correcta requerimientos, ayudar a definirlos y detectar posibles fallos al capturarlos, especialmente para proyectos basados en el paradigma orientado a objetos.

Ventajas de Tiger Pro

- Soporta gestión de riesgos.
- Incluye ejemplos de proyectos para cargar en el programa, que funciona con ficheros .txt.
- Genera gráficos para ayudar a comprender las características del proyecto con el que se está trabajando.

Desventajas de Tiger Pro

- A diferencia de las otras aplicaciones estudiadas, detrás de esta no se encuentra una “gran” empresa de desarrollo, sino que parece una aplicación creada por un particular. Esto no es un gran problema, sin embargo, al no contar con ese respaldo no se garantiza el mantenimiento del producto, cuya última versión data del 2006.
- No genera informes .pdf, sino .txt.

Aunque se trata de una herramienta bastante completa (versión 2.0), no es muy manejable y aprender a manejarla costaría mucho tiempo.

1.5.4 Open Source Requirements Management Tool (Herramienta de código abierto de administración de requerimientos).

Aparte de herramienta para la gestión de requisitos, cubre otras fases del desarrollo de proyectos informáticos como la implementación, el diseño o las pruebas. Incluye control de versiones, riesgos comunes, control de requisitos de código.

Ventajas de Open Source Requirements Management Tool

- Es un proyecto código abierto, por lo tanto una herramienta libre. Se publica bajo la licencia GPL.

- Esta escrita en java, por lo que se puede usar en cualquier sistema operativo que cuente con una implementación de la máquina virtual de java. Es multiplataforma.

Desventajas de Open Source Requirements Management Tool

- La versión 1.4, ocupa 116 MB. Es mucho comparado con las otras herramientas del estudio.
- Es bastante difícil de instalar. La herramienta se distribuye en un formato (.jar) y requiere java 1.5 para poder ejecutarse, conectarse con su propio server y configurar las rutas.

1.5.5 Rational Requisite Pro

La herramienta IBM Rational Requisite Web, brinda a los desarrolladores una interfaz web que permite que los requisitos sean accesibles para todos los integrantes del equipo, especialmente en lugares remotos o en entornos de plataformas múltiples.

Ventajas del Rational Requisite Pro

- Facilita crear, visualizar y modificar requisitos y documentos de requisitos.

Desventajas del Rational Requisite Pro

- Esta herramienta es propietaria.

1.5.6 Caliber-RM

Caliber-RM es una herramienta que resulta una solución sencilla para la gestión de requerimientos altamente colaborativa preparada para accesos a través de Internet, permitiendo a los equipos de desarrollo definir, gestionar y comunicar requisitos cambiantes a lo largo del ciclo de vida.

Ventajas de Caliber-RM

- Tiene un entorno multiusuario basado en web manejando un versionado automático de requerimientos con sus respectivos responsables, estados, tipos, etc.
- Maneja usuarios y almacena las discusiones colaborativas como un requerimiento.

Desventajas de Caliber-RM

- Esta herramienta es propietaria.

1.5.7 REM

REM (Requirements Manager) es una aplicación de gestión de requisitos desarrollada por la Universidad de Sevilla. Se trata de un proyecto libre, no comercial, y con menos opciones que las otras herramientas del estudio. Sin embargo, estas limitaciones no son problema a la hora de definir requisitos de un proyecto pequeño / mediano.

Ventajas de REM:

- Es una herramienta libre, de carácter docente y cuenta con un manual básico de manejo.
- Está íntegramente desarrollada en español.
- Tiene un manejo muy intuitivo y simple. Su interfaz no está sobrecargada de opciones y tiene una buena organización en menús, submenús y barras de herramientas.

Desventajas de REM:

- Es una herramienta destinada a proyectos pequeños, debido a sus limitaciones.

- No permite el modelado de diagramas de casos de uso, solo trabaja con texto plano, es decir, genera el informe de requisitos tras indicar datos del proyecto.
- Ofrece su propia clasificación de requisitos, definiendo requisitos de almacenamiento de información, de restricción, funcional o no funcional. Esto limita la libertad del usuario a la hora de adaptar la aplicación a sus propios requisitos (que no tienen porque coincidir con los establecidos).
- Aunque se genera un fichero HTML como informe, resulta bastante poco flexible y muy estricto en cuanto a su forma. Se usan tablas para separar cada elemento del informe. Es cuestión del usuario organizar la disposición de las entradas.

Más que una herramienta para la gestión de requisitos, es una aplicación para crear informes automáticamente con un formato a partir de los requisitos que se han detectado en un proyecto. No dibuja diagramas de casos de uso, lo que limita su funcionalidad.

1.6 Metodología utilizada

La metodología de desarrollo utilizada en la realización de este Trabajo de Diploma es el Proceso Unificado de Desarrollo (RUP), que sirve de guía para realizar el análisis y diseño de la aplicación, siendo una metodología que ha probado su efectividad durante muchos años, dado el gran número de proyectos que la han utilizado para desarrollar su software. Además, tiene una cantidad considerable de documentos publicados que se pueden consultar para esclarecer dudas. También porque siguiendo sus pasos propuestos se obtiene una buena documentación de la tesis. A continuación se muestran las características que más influyeron en la selección de esta metodología:

- Guiado por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo del sistema.
- Centrado en arquitectura: La arquitectura muestra la visión común del sistema completo.

- Iterativo e Incremental: RUP divide el proyecto en fases de desarrollo, propone además que cada una de ellas se desarrolle en iteraciones, las cuales aportan un incremento en el proceso de desarrollo y terminan con el cumplimiento del punto de control trazado en la fase.
- Utilización de un único lenguaje de modelado: UML.

1.6.1 UML

Para modelar el análisis y diseño de nuestro trabajo se escogió el Lenguaje Unificado de Modelado (*UML*), debido a las potencialidades descriptivas que posee. UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

1.6.2 Rational Rose

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

1.7 Arquitectura Cliente/Servidor

Definición: Sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

Entre los objetivos que persigue la arquitectura Cliente/Servidor se encuentran: la localización transparente, los recursos compartidos y la escalabilidad dada por la migración a otras plataformas

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- 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 implican pocos o ningún cambio en el cliente.
- Entorno de aplicaciones incremental.

-Añadir un nuevo servidor.

-Añadir un nuevo cliente.

-Modificar un cliente para usar un nuevo servidor.

1.8 Servidor web: Apache

Es el servidor web hecho por excelencia, su fácil configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache es uno de los mejores servidores de Webs utilizados en la red internet desde hace mucho tiempo,

únicamente le hace competencia un servidor de Microsoft, el IIS. Por lo que éste servidor es uno de los mayores triunfos del software libre, que tanto gusta a los usuarios de LINUX. [7]

Razones de su popularidad

Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.

Apache es una tecnología gratuita de código abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber.

Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. [8]

Es un servidor de web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Implementa los últimos protocolos, aunque se base en el HTTP / 1.1 Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos.

1.9 Análisis de los Gestores de Base de Datos

Para seleccionar la herramienta a utilizar se debe realizar un análisis de las características principales de los distintos gestores de base de datos más utilizados en la actualidad, para este análisis y selección vamos a tener en cuenta los siguientes gestores: Oracle, SQL Server, MySQL y PostgreSQL.

1.9.1 Oracle

Oracle es un sistema de gestión de base de datos relacional. Se considera como uno de los sistemas de bases de datos más completos, destacando su:

Soporte de transacciones, que no es más que una interacción con una estructura de datos que, aún siendo compleja y estar compuesta por varios procesos que se han de aplicar uno después del otro, se pretende que sea equivalente a una interacción. Es decir, que se realice de una sola vez.

Estabilidad, lo que significa que su nivel de fallo disminuye, en dependencia de la estabilidad que se requiera.

Escalabilidad, es decir, su posibilidad de estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos, siendo capaz de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes. Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias).

Oracle resulta muy difícil de instalar.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros con licencia libre como PostgreSQL y MySQL.

1.9.2 SQL Server

Es un sistema de gestión de base de datos relacionales, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Este es un lenguaje de programación interactivo y estandarizado, declarativo de acceso. Agrupa características del álgebra y el cálculo relacional, permitiendo la realización de consultas para recuperar de forma sencilla la información de una base de datos. Teniendo unas ventajas que a continuación se pueden describir:

De transacciones.

Escalabilidad, estabilidad y seguridad.

Soporta procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

Además permite administrar información de otros servidores de datos.

Microsoft SQL Server, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft (Windows). La licencia es de alto costo.

1.9.3 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario que gestiona bases de datos relacionales poniendo las tablas en ficheros diferenciados, con más de seis millones de instalaciones, es uno de los gestores de base de datos más populares desarrollados bajo la filosofía de código abierto. (Open Source).

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. [9]

Tiene como principales características:

Facilidad de configuración e instalación.

Su gran velocidad y su precio reducido. Es uno de los servidores de bases de datos más rápido y el de menor precio.

MySQL es muy utilizado en aplicaciones PHP o Perl en servidores Linux. En general, si no necesita características como transacciones, procedimientos almacenados, triggers o sentencias SQL complejas, MySQL cumplirá la misma función que otras bases de datos más potentes, pero de forma más rápida y con un coste menor.

Para aplicaciones Windows, MySQL es una alternativa económica a SQL Server, pues su coste es menor y además se puede aprovechar todo el espacio para datos, mientras que SQL Server necesita una parte del espacio para el log de transacciones. [10]

MySQL es gratis incluso para su uso comercial mientras trabaje como servidor de web, pero si se desea trabajar con otras aplicaciones será entonces necesario obtener una licencia.

1.9.4 PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos (PostgreSQL) es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia

multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario). PostgreSQL se diseñó como una base de datos orientada a objetos. Esto significa, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto. Mediante un sistema denominado MVCC (Acceso concurrente multiversión). PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.

Es apropiada para aplicaciones más complejas, en las que se utilizan las normas ANSI-SQL, vistas (views), métodos (stored procedures) y disparadores (triggers). [11]

Características del PostgreSQL:

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

Es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix.

Para aplicaciones Windows, PostgreSQL es una alternativa económica a SQL Server, pues su coste es menor y tiene similares prestaciones. Esta diferencia económica es especialmente sustancial si se necesita un Servidor Dedicado de bases de datos.

En principio, tiene un sistema de bases de datos de mayor nivel a MySQL, a la altura de Oracle, Sybase.

Limitaciones:

El soporte a orientación a objetos es una simple extensión que ofrece prestaciones como la herencia, no un soporte completo.

La mayor limitación de PostgreSQL viene dada por su velocidad: es el sistema de bases de datos más lento, sobrecarga el sistema bastante más que MySQL.

1.10 Gestor de base de datos elegido: MySQL

Luego de haber realizado este análisis de las características de los diferentes gestores que tienen más repercusión en el mundo, se muestra el siguiente resumen comparativo para seleccionar la herramienta de trabajo a utilizar en el proceso de desarrollo de base de datos:

Teniendo en cuenta que Oracle es muy costoso, que SQL Server no es multiplataforma, y ambos son software propietario se puede deducir que ninguno de ellos son de utilidad a este trabajo, aunque sean gestores muy potentes.

Por lo tanto sólo quedaría la comparación entre el PostgreSQL y MySQL, que son software libre, se tendrá en cuenta a MySQL ya que este mientras se le dé un uso particular o comercial(en el caso de que trabaje como servidor web), es gratis. Por otro lado, en el campo de empleo de la velocidad se puede decir que el PostgreSQL es lento y esa es una de las características que más dificulta su uso. Por tanto se determina a MySQL como Gestor de base de datos para este trabajo.

1.11 Análisis sobre los lenguajes de programación

En este epígrafe se realizará un estudio de los diferentes lenguajes de programación, que pueden llevar a la solución de este trabajo.

1.11.1 HTML

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). HTML permite crear documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos. [12]

HTML es el lenguaje utilizado para la creación de páginas Web. Más concretamente, HTML significa “HyperText Mark-Up Language”, que viene a ser, en nuestro idioma, “Lenguaje para el Formato de Documentos de Hipertexto”. Es decir, los documentos HTML no son documentos de texto normal, sino documentos de hipertexto ya que en el propio documento aparecen enlaces a otros documentos.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro (World Wide Web Consortium). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. [13]

1.11.2 Java Script

La característica principal de Javascript, es la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar y teniendo en cuenta la relación que se establece entre el mecanismo cliente-servidor. [14]

En Javascript, la seguridad es casi total y sólo en su primera versión el CIAC (Computer Incident Advisory Committee) señaló problemas de leve entidad, entre ellos la lectura de la caché y de los sitios visitados, de la dirección e-mail y de los archivos presentes en el disco. Sin embargo, estos fallos se corrigieron ya en las versiones de Netscape sucesivas a la 2.0; el código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido. JavaScript comparte muchos elementos con otros lenguajes de alto nivel. Hay que tener en cuenta que este lenguaje es muy semejante a otros como C, Java o PHP, tanto en su formato como en su sintaxis, aunque por supuesto tiene sus propias características definitorias.

1.11.3 CSS

Hojas de Estilo en Cascada (CSS), es una alternativa que permite controlar cómo se va a mostrar un documento en la web.

La utilización de este mecanismo se ha hecho popular en los desarrolladores web, a continuación mostramos sus principales características:

Estilo enriquecido. CSS permite la creación de documentos visualmente mucho más ricos que lo que HTML nunca permitirá. No en vano CSS está pensado única y exclusivamente para asistir al diseñador a la hora de dar estilo a un documento estructurado.

Fácil de utilizar. La utilización de hojas de estilo CSS hace que el diseñador pueda reducir sustancialmente su carga de trabajo al diseñar todo un sitio. Esto se debe a que CSS es capaz de centralizar ciertos efectos visuales que plasmemos en diversas secciones del sitio, en lugar de tenerlos diseminados por páginas y páginas del sitio.

Reutilización en múltiples páginas. Una hoja de estilo que recoja aspectos visuales comunes a varias páginas puede ser reutilizada en cualquier sección del sitio aprovechando dichos efectos ya definidos. De esta manera es sencillo generar un estilo general del web y mantenerlo así consistente para todas las páginas. Así, si deseamos modificar un estilo que es común a todo el sitio, sólo necesitaríamos modificar una línea de nuestro fichero CSS (con la aproximación clásica que ofrece HTML, deberíamos modificar todas y cada una de las páginas).

1.11.4 ASP.NET

ASP es una tecnología desarrollada por Microsoft para crear páginas Web de contenido dinámico apoyándose en scripts ejecutados en el servidor. [15]

Principales características de ASP.NET

Soporte de Lenguajes.

Esta es una de las novedades más importantes que vienen de la mano de ASP.NET. La posibilidad de escribir código en diferentes lenguajes es un alivio para los desarrolladores que en numerosas ocasiones, veían acotadas sus aplicaciones web, al estar obligados a trabajar con VBScript o JScript. ASP.NET soporta la programación en lenguajes potentes como, VisualBasic.Net (VB) y C#, el nuevo lenguaje

creado por Microsoft con la intención de aprovechar la potencia del C++ y combinarlo con las facilidades que brinda a la programación en Internet un lenguaje como Java.

Contenido y Código, por separado.

Muchos desarrolladores de sitios web han tenido que lidiar con el inconveniente de tener que crear la interfaz de usuario y el código ASP todo junto. Esta mezcla de imágenes, botones y tablas en código HTML con pedazos de código en VBScript o Jscript llegaba a ser algo muy molesto para el desarrollador. ASP.NET viene a solucionar este problema, utilizando un criterio similar al que utiliza Visual Basic, es decir, separar la interfaz de usuario con el código.

Compatibilidad con Navegadores ASP.NET permite crear una página web que funcionará correctamente en todos los navegadores. Esta mejora está dada especialmente por los controles de servidor incluidos en ASP.NET. Cuando un control es procesado, este automáticamente chequea el tipo de navegador que lo está ejecutando, generando una página adecuada para ese navegador.

Código Compilado.

ASP.NET ya no interpreta el código como la hace la versión anterior de ASP. Dentro del entorno NGWS (New Generation Windows Services, Servicios de Nueva Generación de Windows) el código es compilado en tiempo real, logrando un enorme aumento en el rendimiento, a través de soporte nativo y servicios de caché. Controles de Servidor, uno de los aspectos más importantes dentro del .NET Framework es su librería de clases.

Esta librería es común en toda la plataforma .NET, lo que le brinda al programador una herramienta ideal para crear aplicaciones multiplataforma, con un considerable ahorro de líneas de código. Los controles de servidor están divididos en dos categorías: Controles Web y Controles HTML.

Posiblemente sean los Controles Web, lo que atractivos para el desarrollador, ya que permiten crear automáticamente controles que realicen tareas importantes en el servidor como validar la entrada de formularios, verificar las capacidades de los navegadores o implementar un sistema de banners rotativos.

1.11.5 PHP

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft, embebido en páginas HTML y ejecutado en el servidor. [16]

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML o XML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

A diferencia de Java o JavaScript que se ejecutan en el navegador PHP, se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en .pdf (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

1.11.6 Perl

PERL (Practical Extraction and Report Language) es un lenguaje creado a principio de los noventa por Larry Wall, el cual lo realizó casi como una obra altruista, por eso, su distribución es gratuita. Perl es un lenguaje pensado para la manipulación de cadenas de caracteres, archivos y procesos. [17]

El lenguaje Perl no es precompilado, pero aún así es más rápido que la mayoría de lenguajes interpretados. Esto se debe a que los programas en Perl son analizados, interpretados y compilados por el interprete *perl* antes de su ejecución.

Estas características hacen que el mantenimiento y la depuración de un programa en Perl sean mucho más sencillos que el mismo programa escrito en C.

Por todo esto, Perl es un lenguaje muy utilizado en los dos campos siguientes:

- **La administración de sistemas operativos.** Debido a sus características Perl es muy potente en la creación de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas, etc. Además, aunque Perl nació en un entorno Unix, hay versiones para casi todas las plataformas existentes.
- **La creación de formularios en la Web.** Es decir, se utilizan para la creación de scripts CGI (*Common Gateway Interface*). Estos scripts realizan el intercambio de información entre aplicaciones externas y servicios de información, es decir, se encargan de tratar y hacer llegar la información que el cliente WWW manda al servidor a través de un formulario.

1.11.7 Ajax

Ajax, es una de las novedades más radicales surgidas de la red en los últimos años, significa Asynchronous Java Script and XML y es una mezcla de tecnologías que agilizan el uso de internet e impulsan la llegada de la Web 2.0, un escenario en el que la mayoría de herramientas que ahora utilizamos (procesador de textos, hojas de cálculo, etcétera) estarán en línea y se utilizarán a partir del navegador y no del sistema operativo. [18]

Entre las características de AJAX se incluyen bibliotecas de scripts de cliente, que incorporan las tecnologías JavaScript y HTML dinámico (DHTML) para varios exploradores, e integración con la plataforma de desarrollo para servidores de ASP.NET. Gracias a las características de AJAX puede mejorar la experiencia del usuario y la eficacia de sus aplicaciones web.

1.12 Lenguaje seleccionado: PHP

Luego de un análisis concreto de los diferentes lenguajes antes expuestos, seleccionamos PHP para el desarrollo de nuestra herramienta debido a que es un lenguaje muy eficaz en el trabajo con los servidores y además es compatible con varios gestores de base de datos, entre los que se encuentra MySQL (Gestor de Base de Datos seleccionado en esta investigación), además es código abierto, por lo que se puede contar con la ayuda de un gran grupo de programadores. Una de las características más adecuada para este trabajo de diploma es la posibilidad de realizar el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Su integración con bibliotecas externas, facilita el trabajo con documentos como XML y PDF.

1.13 Integración PHP, MySQL y Apache

Después de realizar un estudio exhaustivo de los lenguajes de programación web, los gestores de base de datos y los servidores, se llegó a la conclusión de que la integración de PHP, MySQL y Apache es la mejor opción, para la construcción de la aplicación que se pretende desarrollar. La gran seguridad que brinda Apache y la potencia y aseguramiento que tiene implementada MySQL son características que motivan a seleccionar estas herramientas, teniendo en cuenta también que son libres y gozan de gran popularidad en el ámbito de la informática.

1.14 Herramientas Utilizadas

En el desarrollo de esta investigación se trabajará con el Rational Rose Enterprise Edition 2003, UML para realizar el modelo completo que representa el dominio del problema del sistema de software a desarrollar, debido a que es una herramienta muy integral que brinda potencialidades adecuadas para el desarrollo de software.

Para el diseño de la página Web se utilizará el Dreamweaver 8, herramienta muy eficaz para el diseño de interfaces y el trabajo vinculado con las Hojas de Estilo, para el diseño de las imágenes se empleará el Adobe Photoshop CS3, considerado el más adecuado para la edición digital de las imágenes, debido a que brinda muchas funcionalidades; y para la edición del código PHP el ZendEstudio versión 5.1, que contiene un completamiento de código muy útil para ganar en fluidez del trabajo.

El cliente que se tendrá en cuenta para manipular la base de datos es MySQL, herramienta muy adecuada para el trabajo con vistas y procedimientos, y eficaz para la manipulación de las tablas existentes en la base de datos. El servidor web será Apache, versión 2.2.

Además se utilizará el framework Ext el cual contiene utilidades para trabajar plantillas o DOM y suporta el uso de DOM o fragmentos de HTML de forma transparente.

Capítulo 2. Características del sistema.

2.1 Introducción.

En este capítulo se describirán las principales características que poseerá la herramienta a desarrollar, teniendo en cuenta la situación problemática anteriormente planteada en la introducción de la investigación. Además, se abordará lo referente al negocio a estudiar mostrando en detalles los procesos que serán objeto de automatización del mismo, se obtiene el Modelo del Dominio y las descripciones de los casos de uso del sistema. Se definen los requerimientos funcionales y los no funcionales, así como se identifican los actores y casos de uso que intervienen en el Sistema.

2.2 Descripción de los procesos del negocio propuestos.

En la Universidad de las Ciencias Informáticas se desarrollan disímiles proyectos productivos, donde los clientes se encuentran en su mayoría fuera de Ciudad de la Habana, por lo que en algunos casos el proceso de captura de requisitos se realiza a través de correos, chat, llamadas telefónicas, etc. Debido a la utilización de esos métodos de comunicación se presentan problemas en la terminación de cada proyecto.

Para establecer un mejor intercambio entre analistas y clientes, y además poder crear un estándar en el proceso de captura de requisitos en los proyectos de la Universidad de las Ciencias Informáticas, se decidió la construcción y montaje de una aplicación web que permita emitir como resultado final un documento con la descripción de los requisitos funcionales como principal objetivo.

La aplicación brindará a los analistas la forma de realizar la descripción de requisitos, además de poder cambiarlos en caso de existir algún desacuerdo con el cliente. Para hacer más factible la comunicación entre analistas y clientes la aplicación pondrá a disposición de los usuarios encuestas, así como un foro de discusión. Por otro lado los clientes podrán tener acceso a los requerimientos y dar su opinión acerca de ellos de forma que se pueda llegar a un consenso común entre ambas partes.

En general se debe manejar un gran número de información referente a usuarios y analistas, además la información de los proyectos que se creen para realizar los sistemas que se aprueben y de los requisitos que se describen en el desarrollo de cada proyecto.

2.3 Modelo del dominio.

El proceso de modelar permite obtener una visión de la organización, permitiendo definir los procesos y roles relacionado con la obtención de requerimientos y del análisis-diseño. Teniendo en cuenta el bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en tecnologías informáticas, se ha llegado a la conclusión de que el negocio no necesita un modelado completo del mismo, por lo que no es necesario realizar un Modelo de Negocio para comprender la problemática que ha de resolverse, se propone un Modelo del Dominio, ya que de manera visual permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Para un mejor entendimiento todos los conceptos que se emplearán en el diagrama se utilizará un glosario de términos que se definen a continuación:

Analista: el encargado de crear requerimientos, casos de uso y encuestas.

Cliente: el encargado de enviar una descripción del sistema a desarrollar.

Gestión Requisitos: proceso que realiza la captura de Requisitos y la especificación de Casos de Uso.

Herramienta: vía que utiliza los clientes para enviar la información.

Chat: herramienta utilizada por los clientes.

Correo: herramienta utilizada por los clientes.

Fax: herramienta utilizada por los clientes.

Información: recopilación de datos que envía el cliente.

Datos: descripción de lo que desea implementar el cliente y sus datos personales.

Grupo analistas: Equipo de analistas designados para realizar un proyecto determinado.

Analista: Persona(s) que conforma un grupo para realizar la captura de requisitos de un proyecto determinado.

Requisito: Conjunto de requisitos que integrarán el documento final.

Caso Uso: Conjunto de casos que integrarán el documento final.

Información final: Documento con la recopilación de los requisitos funcionales y no funcionales, además de los casos de uso.

2.3.1 Descripción del modelo del dominio.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

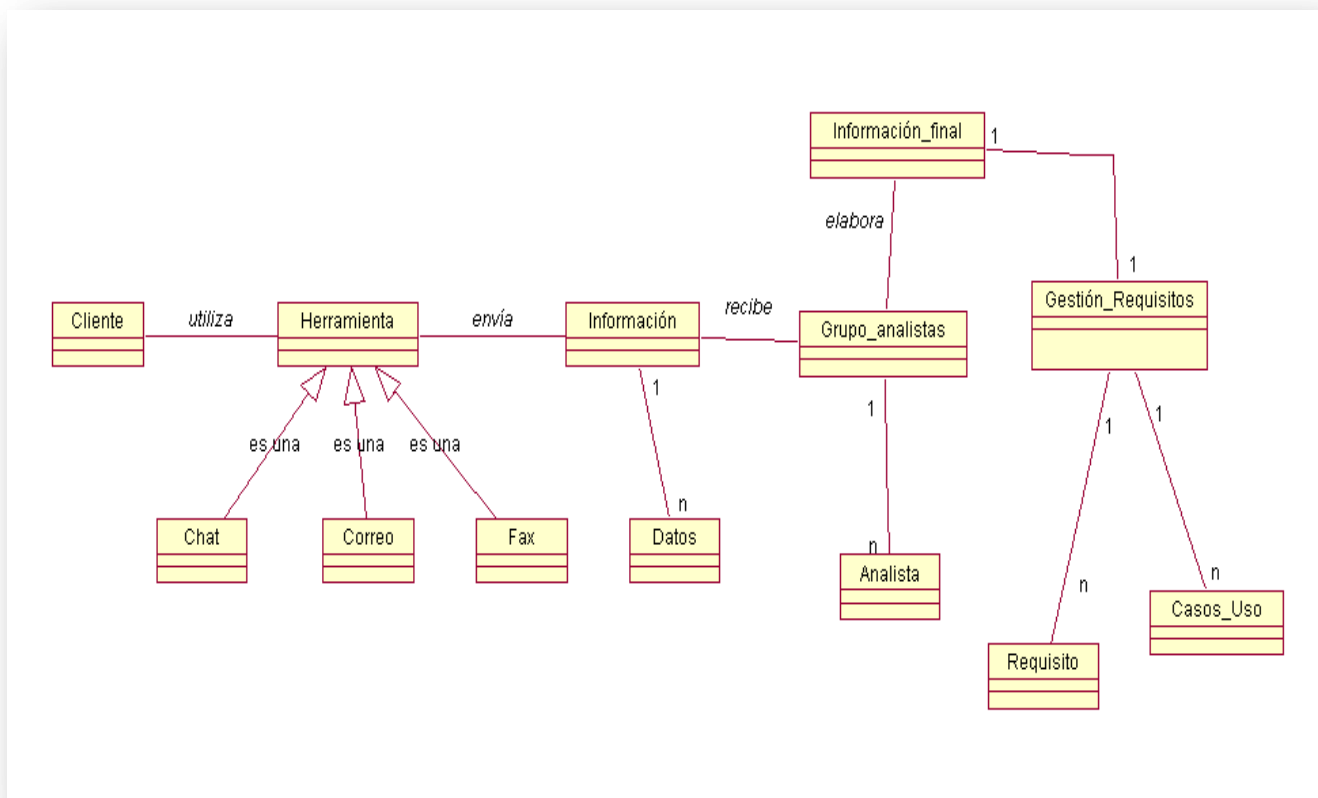


Fig. 2.1 Diagrama Modelo del Dominio

2.4 Solución propuesta

La Obtención de Requisitos cumple un papel primordial en el proceso de producción de software, ya que se define lo que se desea producir, generando las especificaciones correctas que describen con claridad, sin ambigüedades, de forma consistente y compacta, el comportamiento del sistema.

Cuando se realiza software a distancia, generalmente las expectativas del cliente no son satisfechas en su totalidad y el producto no se entrega en el plazo de tiempo acordado. La comunicación a distancia entre clientes y desarrolladores se reduce a llamadas telefónicas, fax y mensajes de correo electrónico donde el tiempo y el costo constituyen las principales limitantes. Aunque existen numerosas herramientas que pueden ser utilizadas para la Gestión de Requisitos, no ocurre así para la captura de estos, pues en el mundo las empresas que desarrollan software a distancia, no utilizan una herramienta para facilitar la comunicación entre cliente-analista y en muchos casos no se realiza una adecuada captura de requisitos, y en otros se resume a que el cliente mande un documento explicando detalladamente qué es lo que quiere.

Por ello para facilitar la comunicación entre clientes-analistas en el desarrollo offshore, y lograr que el proceso de captura de requisitos sea más eficiente, se decidió desarrollar una herramienta que permita realizar la captura de requisitos a distancia, proporcionando como producto final la descripción detallada de los Requerimientos funcionales y no funcionales, además de los casos de uso y la trazabilidad.

La propuesta para dar solución consiste en estructurar la arquitectura del sistema de modo que cuente con cuatro módulos:

El módulo de propuesta ofrece la oportunidad a los usuarios no registrados de interactuar con la aplicación inicialmente.

El módulo de usuario del sistema puede ser utilizado por todos los roles, y en él se brinda la información relacionada con las distintas posibilidades que tienen los usuarios registrados de la aplicación.

El módulo de usuario de intercambio ofrece una serie de servicios que facilitan el trabajo y la comunicación entre clientes y analistas, destacándose dentro de estos el foro. Además detalla las funcionalidades que van a tener cada uno en específico.

El módulo de administración controla todo lo relacionado con la gestión de los usuarios que acceden al sistema, así como los proyectos que existan. A este módulo puede acceder sólo el administrador, el que a su vez, tampoco puede cumplir las funciones de los demás roles.

2.5 Requisitos Funcionales

Una vez conocidos los conceptos que rodean al objeto de estudio, se puede comenzar a analizar ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello se enumerarán a través de requerimientos funcionales las funciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

RF1. Autenticar Usuario.

RF1.1. La aplicación debe ser capaz de permitir al usuario introducir su nombre de usuario y contraseña (no debe exceder de 8 caracteres) para ingresar al sistema.

RF1.2. La aplicación debe validar que el usuario tenga acceso al sistema.

RF1.3. La aplicación de acuerdo al usuario autenticado debe permitir el acceso al módulo correspondiente a los privilegios otorgados.

RF2. Finalizar sesión de usuario.

RF2.1. La aplicación debe permitir que el usuario registrado cierre su sesión de trabajo desde cualquier lugar del sistema.

RF3. Cambiar contraseña.

RF3.1. La aplicación debe permitir a los usuarios registrados poder cambiar su contraseña de acceso al sistema.

RF4. Visualizar requisitos.

RF4.1.La aplicación debe permitir a los usuarios intercambio (clientes/analistas) visualizar los requisitos existentes, así como los comentarios que se hagan acerca de ellos.

RF5. Comentar requisito.

RF5.1.La aplicación debe permitir a los usuarios intercambio realizar comentarios acerca de los requisitos expuestos en la aplicación.

RF6. Ver comentarios del foro.

RF6.1.La aplicación debe permitir a los usuario intercambio (clientes/analistas) visualizar los temas que se discuten el foro.

RF7. Enviar mensajes en el foro

RF7.1.La aplicación debe permitir a los usuarios intercambio (clientes/analistas) enviar mensajes al foro sobre los diferentes temas que existan en el mismo.

RF8. Mostrar encuesta.

RF8.1. La aplicación debe permitir a los usuarios intercambio (clientes/analistas) visualizar las encuestas publicadas, así como las respuestas que realicen los clientes de estas.

RF9.Responder encuesta.

RF9.1. La aplicación debe permitir a los clientes responder las encuestas que se encuentren publicadas.

RF10. Mostrar información del proyecto.

RF10.1. La aplicación debe permitir a los clientes visualizar en su sesión la información del proyecto al que está asociado.

RF11. Gestionar foro.

RF11.1 La aplicación debe permitir al analista insertar nuevos temas en el foro.

RF11.2 La aplicación debe permitir al analista eliminar temas existentes en el foro.

RF12.Gestionar encuesta.

RF12.1. La aplicación debe permitir al analista crear encuestas.

RF12.2. La aplicación debe permitir al analista modificar encuestas existentes.

RF12.3. La aplicación debe permitir al analista eliminar encuestas existentes.

RF13.Gestionar requisitos

RF13.1. La aplicación debe permitir al analista insertar requisitos.

RF13.2. La aplicación debe permitir al analista modificar los datos de los requisitos existentes.

RF13.3. La aplicación debe permitir al analista eliminar requisitos existentes.

RF13.4. La aplicación debe permitir al analista establecer y modificar la relación entre los requisitos que vaya determinando y los casos de uso existentes.

RF14.Gestionar casos de uso.

RF14.1. La aplicación debe permitir al analista insertar un caso de uso.

RF14.2. La aplicación debe permitir al analista modificar los datos de un caso de uso.

RF14.3. La aplicación debe permitir al analista eliminar un caso de uso.

R15. Mostrar la matriz de trazabilidad.

R15.1. La aplicación debe permitir al analista visualizar la matriz de trazabilidad para facilitar el control de los cambios, es decir, si algún requerimiento sufre una modificación, es fácil determinar qué partes del sistema se ven afectados.

R16. Gestionar Glosario de Términos.

RF16.1. La aplicación debe permitir al analista la insertar nuevos términos al glosario de términos.

RF16.2. La aplicación debe permitir al analista modificar los términos existentes en el glosario de términos.

RF16.3. La aplicación debe permitir al analista eliminar los términos existentes en el glosario de términos.

R17. Generar reporte.

R17.1. La aplicación debe permitir al analista generar un reporte (.pdf) con la especificación de los requisitos y los casos de uso a partir de la información recopilada.

RF18. Gestionar proyecto

RF18.1. La aplicación debe permitir al administrador crear un proyecto.

RF18.2. La aplicación debe permitir al administrador modificar los datos de un proyecto.

RF18.3. La aplicación debe permitir al administrador eliminar un proyecto.

RF18.4. La aplicación debe permitir al administrador visualizar un listado de los proyectos existentes para mantener el control de las actualizaciones que se realicen.

RF19. Gestionar grupo de analistas.

RF19.1. La aplicación debe permitir al administrador asignar analista(s) a un proyecto existente.

RF19.2. La aplicación debe permitir al administrador quitar analista(s) de un proyecto existente.

RF20.Gestionar Usuario.

R20.1.La aplicación debe permitir al administrador crear nuevos usuarios del sistema con los datos (Nombre, Apellido, Email, Empresa, Rol, País y Sexo).

R.20.La aplicación debe permitir al administrador eliminar usuarios del sistema.

R21. Enviar email de notificación.

RF21.1.La aplicación debe permitir que el administrador envíe un email al cliente con usuario, contraseña y link de la aplicación, en el caso de que haya sido aceptada su propuesta de proyecto. En caso contrario se le notificará la negativa.

R22. Enviar email con propuesta.

RF22.1.La aplicación debe permitir que el cliente que aún no está registrado envíe un email con su datos personales (nombre y apellidos, edad, país, dirección de email, empresa), además de una descripción del producto que quiere.

2.6 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Usabilidad: El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.

Rendimiento: Para un buen funcionamiento de la aplicación se seguirán las diferentes técnicas de trabajo sobre plataforma Web, que faciliten el rápido acceso a sus páginas.

La herramienta propuesta Web podrá ser usada bajo cualquier sistema operativo. El sistema almacenará todos los datos en una base de datos utilizando MySQL estándar, a la cual se pueda tener acceso por otros programas.

Software: En la computadora que haga función de servidor, independientemente del sistema operativo, se necesita el lenguaje de programación PHP, MySQL y servidor Apache. En las computadoras de los usuarios solo se requiere de navegador para Internet Explorer.

Hardware: Se requiere de un servidor con mínimo de 256 MB de RAM y 200 MB de espacio libre en disco duro, todas las computadoras implicadas, tanto para la administración como las de los usuarios, deben estar conectadas a una.

Portabilidad: El sistema correrá no solo sobre Windows sino también sobre Linux o Unix, de forma tal que no haya dificultad y se pueda llevar a cabo esta acción sin necesidad de efectuar cambios significativos.

Interfaz: La herramienta propuesta será usada por personas que no necesariamente tienen habilidades en el uso de la web, por lo que la interfaz será de navegación fácil, permitiendo que no se necesite de mucho entrenamiento para utilizarla.

Requerimiento de ayuda y documentación en línea: La herramienta no posee ayuda solo cuenta con ciertas aclaraciones sobre algunas opciones del sistema que garantizaran el buen desempeño de los usuarios a la hora de interactuar con el mismo.

2.7 Modelo de casos de uso del sistema.

En este epígrafe se hace referencia a los actores del sistema, y se da una breve descripción de los principales casos de uso.

El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, es decir, representan terceros fuera del sistema que se relacionan con él. Estos suelen corresponderse con trabajadores o actores del negocio.

2.7.1 Definición de los actores del sistema.

Esta aplicación se construye sobre la base de las nuevas tendencias relacionadas con la Web, teniendo en cuenta las características del usuario. A continuación se muestran los actores del sistema:

Tabla 1. Actores del sistema

Actores	Descripción
Usuario no registrado	Usuario que es un cliente probable. Inicialmente tendrá como única opción la de enviar un email al administrador con sus datos personales y una descripción de lo que quiere que le hagan.
Usuario Registrado	Usuario que tendrá nombre de usuario y contraseña para acceder al sistema, así como podrá visualizar en cualquier interfaz la opción de cambiar contraseña y/o cerrar sesión.
Usuario Intercambio	Usuario que podrá visualizar el foro de la aplicación, así como comentar sobre los temas que existan en el mismo. Además podrá ver las encuestas, los requisitos y comentar sobre ellos.
Administrador	Usuario encargado de la creación de los grupos de analistas, así como la gestión de los mismos (adicionar o eliminar analistas de los grupos). También deberá aceptar o no la creación un proyecto determinado, notificando al cliente en cualquier caso. Además es el responsable de la gestión de los usuarios del sistema teniendo en cuenta los diferentes niveles de accesibilidad.
Analista	Usuario que tendrá como funcionalidad la gestión de

	los requisitos y casos de uso del proyecto a realizar, teniendo en cuenta la información enviada por los clientes, además de que podrá crear foros y encuestas para lograr un mejor intercambio.
Cliente	Usuario registrado en el sistema que podrá realizar comentarios de los requisitos del proyecto al que pertenezca, además de intercambiar con los analistas de su proyecto en cuestión, a través de foros y encuestas.

2.7.2 Definición de los casos de usos del sistema

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Por lo general el nombre de un caso de uso comienza con un verbo en infinitivo. Un caso de uso describe un proceso de principio a fin, es decir, una secuencia de eventos, las acciones y las transacciones que se requieren para realizarlo.

Para modelar los casos de usos del sistema se ha decidido dividir estos en cuatro paquetes atendiendo a las características de los diferentes módulos:

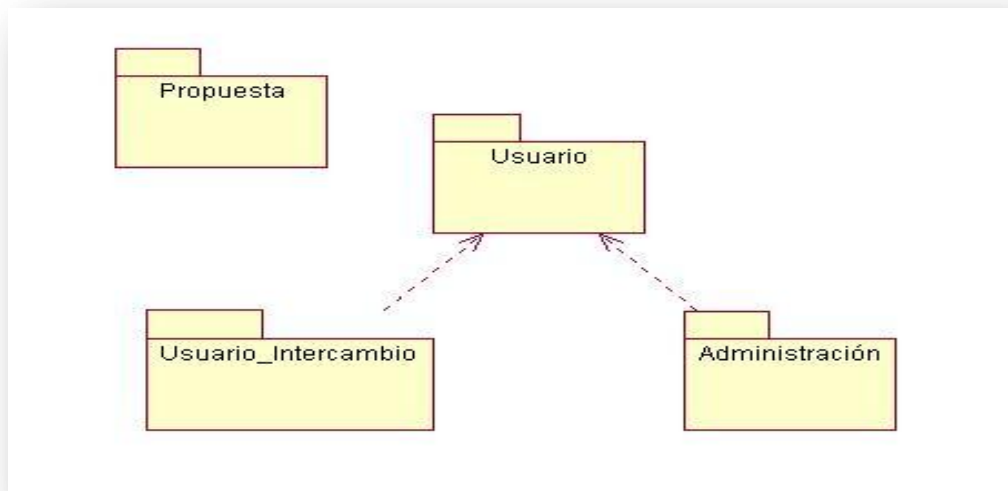


Fig.2.2 Paquete del Modelo de Casos de Uso del Sistema.

Módulo de Usuario.

Los Casos de Uso que a continuación se presentan reflejan las funcionalidades que tienen los usuarios registrados en el sistema para el manejo de sus cuentas como miembros del grupo que son. Para poder utilizarlos se necesita una previa autenticación.

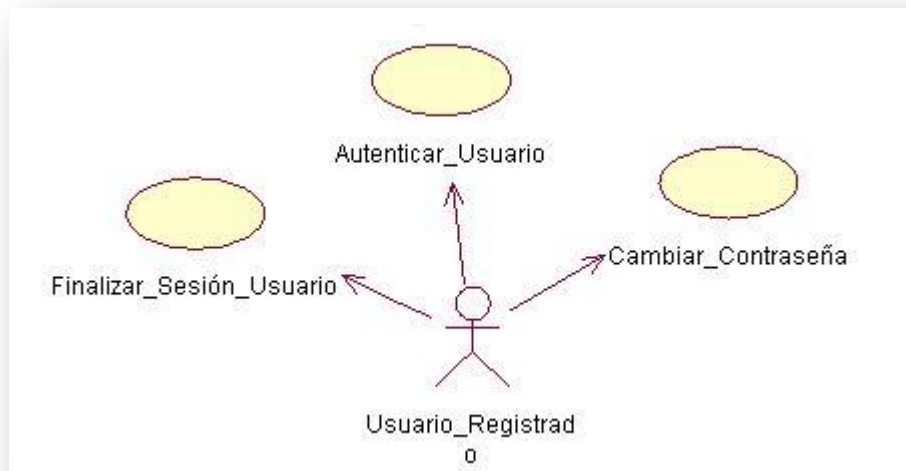


Fig.2.3 Casos de Uso asociados al Paquete Usuario.

Módulo de Usuario Intercambio.

Los Casos de Uso que se muestran en este paquete pertenecen a todas las opciones que tienen los usuarios registrados, exceptuando administradores, con el fin de lograr una correcta comunicación para alcanzar los objetivos propuestos. Además de detallan las funcionalidades de cada uno en particular.

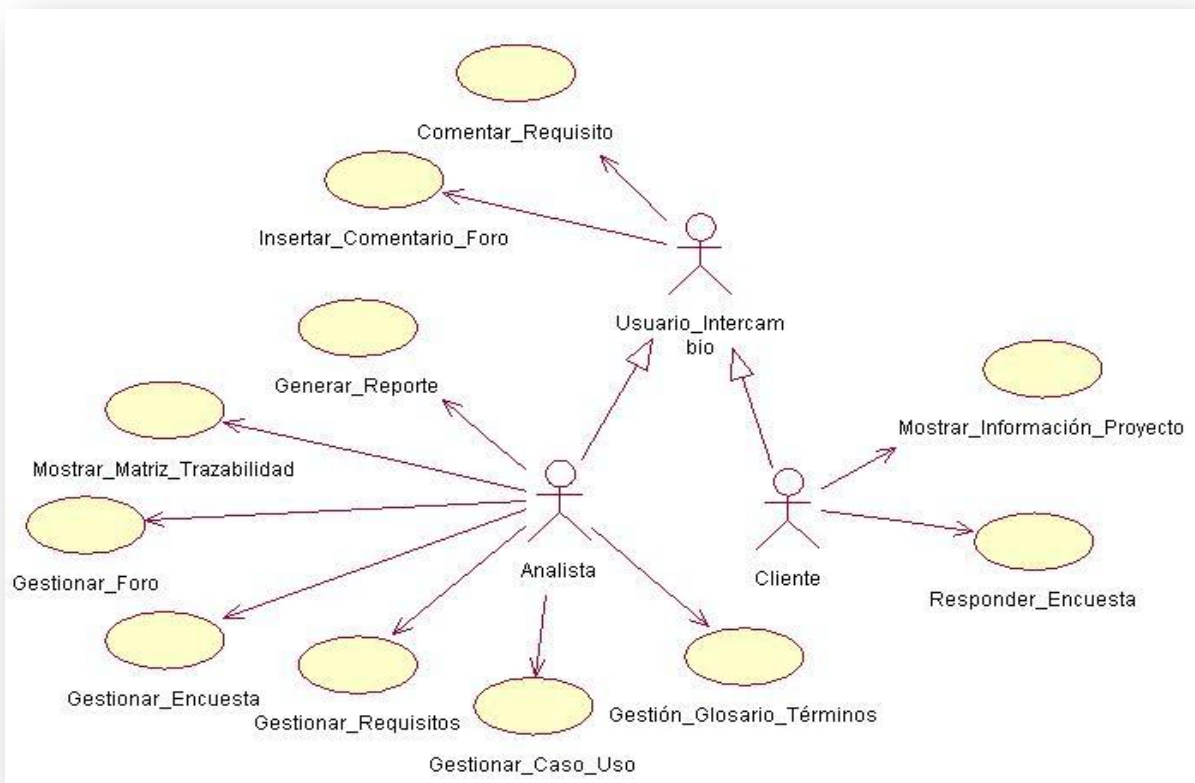


Fig.2.4 Casos de Uso asociados al Paquete Intercambio.

Módulo de Administración.

En este paquete es donde se muestra al Administrador y su relación con los Casos de Uso que pertenecen a la gestión de los usuarios, proyectos y grupos de proyectos rigen el funcionamiento del sistema.

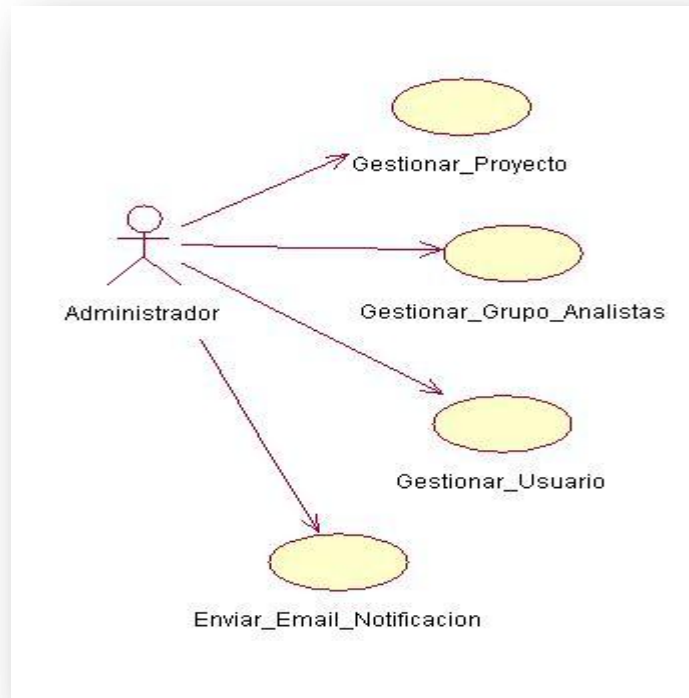


Fig.2.5 Casos de Uso asociados al Paquete de Administración.

Módulo Propuesta.

En el siguiente paquete se muestran las oportunidades que tienen los usuarios no registrados de interactuar con el sistema.

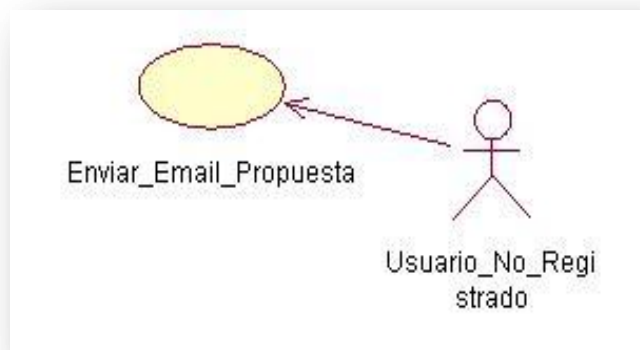


Fig.2.6 Casos de Uso asociados al Paquete Propuesta.

2.7.3 Descripción de los casos de uso del sistema

Módulo Usuario.

Tabla 1. CU Autenticar Usuario

CU-1	Autenticar Usuario
Actor	Usuario Registrado
Descripción	El caso de uso se inicia cuando el actor introduce su usuario y contraseña para acceder a la aplicación, estos datos se verifican y finaliza direccionándolo a su sesión con los permisos que tenga de acuerdo con el rol que desempeñe.
Referencia	RF1

Tabla 2. CU Finalizar Sesión Usuario

CU-2	Finalizar Sesión Usuario
Actor	Usuario Registrado
Descripción	El caso de uso se inicia cuando el actor, previamente

	autenticado, decide salir del sistema. La aplicación cierra su sesión y lo envía a la página inicial.
Referencia	RF2

Tabla 3. CU Cambiar Contraseña

CU-3	Cambiar contraseña
Actor	Usuario registrado
Descripción	El caso de uso se inicia cuando el actor decide cambiar la contraseña.
Referencia	RF3

Módulo Usuario Intercambio.

Tabla 4. CU Comentar Requisito

CU-4	Comentar Requisito
Actor	Usuario intercambio
Descripción	El caso de uso se inicia cuando el actor decide acotar un comentario acerca de algún requisito existente que no esté del todo claro, con el fin de que se encuentre lo más cercano posible a la realidad.
Referencia	RF5

Tabla 5. CU Insertar Comentario Foro

CU-5	Insertar Comentario Foro
Actor	Usuario intercambio
Descripción	Opción que permite al actor enviar mensajes a los diferentes foros a los que tengan acceso.
Referencia	RF6, RF7

Tabla 6. CU Responder Encuesta

CU-6	Responder Encuesta
Actor	Cliente
Descripción	El caso de uso se inicia cuando el cliente introduce las respuestas a determinadas preguntas que existan en la encuesta.
Referencia	RF9

Tabla 7. CU Mostrar Información Proyecto

CU-7	Mostrar Información Proyecto
Actor	Cliente
Descripción	El caso de uso se inicia después de haberse autenticado el cliente accediendo a su sesión donde puede visualizar los datos de su proyecto en cuestión.
Referencia	RF10

Tabla 8. CU Gestionar Foro

CU-8	Gestionar Foro
Actor	Analista
Descripción	El analista inserta al foro nuevos temas para alcanzar una mejor comunicación con los clientes. Conjuntamente a esto se podrán modificar y/o eliminar los temas ya existentes.
Referencia	RF11

Tabla 9. CU Gestionar Encuesta

CU-9	Gestionar Encuesta
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista adiciona pregunta(s) a la encuesta de un determinado

proyecto, con el fin de lograr un mejor intercambio con los clientes. Paralelamente se podrán eliminar las preguntas ya existentes.

Referencia	RF12
-------------------	------

Tabla 10. CU Gestionar Requisito

CU-10	Gestionar Requisito
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista inserta un requisito. Conjuntamente a esto se pueden modificar y/o eliminar los requisitos ya existentes.
Referencia	RF13

Tabla 11. CU Gestionar Caso Uso

CU-11	Gestionar Caso Uso
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista inserta un nuevo caso de uso. Conjuntamente a esto se pueden modificar o eliminar los casos de uso existentes.
Referencia	RF14

Tabla 12. CU Mostrar Matriz Trazabilidad

CU-12	Mostrar Matriz Trazabilidad
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista accede a visualizar la matriz de trazabilidad creada a partir de las relaciones establecidas entre los requisitos y los casos de uso que se hayan determinado.
Referencia	RF15

Tabla 13. CU Gestionar Glosario Términos

CU-13	Gestionar Glosario Términos
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista inserta nuevos términos, organizándolos con sus respectivos significados. Conjuntamente a esto se pueden modificar o eliminar los términos existentes.
Referencia	RF16

Tabla 14. CU Generar Reporte

CU-14	Generar Reporte
Actor	Analista
Descripción	El caso de uso se inicia cuando el analista desea generar un documento a partir de la información obtenida con la recopilación de los requisitos funcionales, no funcionales y los casos de uso.
Referencia	RF17

Módulo Administración.

Tabla 15. CU Gestionar Proyecto

CU-15	Gestionar Proyecto
Actor	Administrador
Descripción	El caso de uso se inicia cuando al administrador introduce los datos de un proyecto para crearlo. Así como también puede modificarlos y/o eliminarlos.
Referencia	RF18

Tabla 16. CU Gestionar Grupo Analistas

CU-16	Gestionar Grupo Analistas
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador decide insertar analista(s) al equipo de un proyecto creado previamente.
Referencia	RF19

Tabla 17. CU Gestionar Usuario

CU-17	Gestionar Usuario
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador introduce los datos de un cliente/analista para registrarle y termina cuando el administrador le envía vía email al usuario su login y contraseña. Conjuntamente a esto se podrán modificar los datos y/o eliminar usuarios ya existentes
Referencia	RF20

Tabla 18. CU Enviar Email Notificación

CU-18	Enviar Email Notificación
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador envía al cliente un email informándole si fue o no aprobada su propuesta. En caso positivo le envía también su cuenta de usuario y contraseña, así como el link a la página de la aplicación.
Referencia	RF21

Módulo Propuesta.

Tabla 19. CU Enviar Email Propuesta

CU-19 Enviar Email Propuesta	
Actor	Usuario no registrado
Descripción	El caso de uso se inicia cuando el actor introduce los datos personales y una descripción del lo que quieren que le hagan y es enviado al administrador para posterior análisis de la propuesta.
Referencia	RF22

2.7.4 Expansión de los Casos de Uso

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. A continuación se describen en formato expandido los casos de uso.

Caso de uso	
CU1	Autenticar Usuario.
Propósito	Proporcionar al actor la entrada al sistema y garantizar que en todo momento se conozca quién está accediendo al mismo.
Actores: Usuario Registrado(Inicia)	
Resumen: El caso de uso se inicia cuando el actor introduce su usuario y contraseña para acceder a la aplicación, estos datos se verifican y finaliza direccionándolo a su sesión con los permisos que tenga de acuerdo con el rol que desempeñe.	
Referencias	RF 1
Precondiciones	Existencia de Usuario y Contraseña válidos para el sistema.
Poscondiciones	El actor es autenticado correctamente.



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Usuario Registrado necesita entrar al sistema.	2. El sistema muestra la interfaz inicial con las opciones de autenticarse y de enviar email (ver Pantalla 1. Entrar (A) y Enviar Email (B)).
3. El Usuario Registrado presiona el botón de Entrar.	4. El sistema muestra la interfaz correspondiente para que el usuario registrado introduzca sus datos (ver Pantalla 2. Usuario (A) y Contraseña (B)).



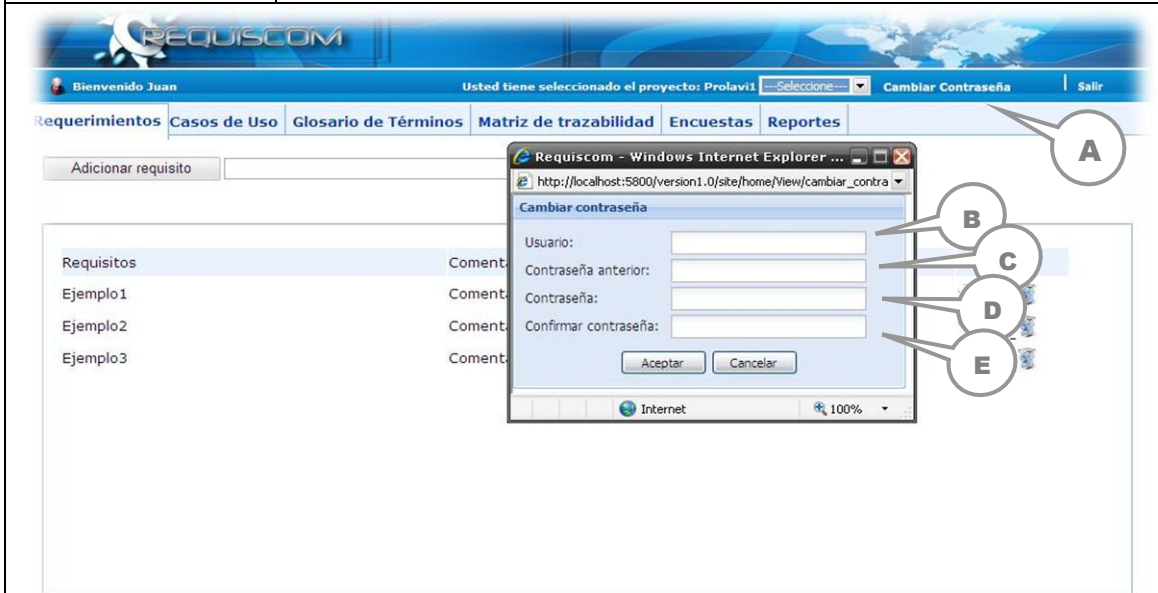
Pantalla 2

<p>5. El Usuario registrado introduce los datos requeridos y presiona el botón Aceptar (C) para esperar respuesta del sistema.</p>	<p>6. El sistema valida la identidad del actor y muestra un mensaje de autenticación confirmada.</p>
	<p>7. El Sistema lo envía al actor directamente a las principales opciones a las que tiene acceso según el rol que desempeñe.</p>
<p>Flujo alternativo</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>6. En caso que la autenticación se realice de forma incorrecta se muestra un mensaje de error y retorna a la acción 3.</p>
<p>Puntos de extensión.</p>	
<p>Prioridad</p>	<p>Crítico</p>

Caso de uso	
CU2	Finalizar Sesión Usuario
Propósito	Cerrar la sesión del usuario que se encuentre autenticado.
Actores: Usuario Registrado(Inicia)	
Resumen: El caso de uso se inicia cuando el actor, previamente autenticado, decide salir del sistema. La aplicación cierra su sesión y lo envía a la página inicial.	
Referencias	RF 2
Precondiciones	El actor tiene que estar autenticado satisfactoriamente en el sistema.
Poscondiciones	El sistema cierra la sesión de usuario correctamente.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona la opción "Salir".	2. El sistema cierra la sesión del usuario y lo envía a la página inicial.
Prioridad	Secundario

Caso de uso	
CU3	Cambiar Contraseña
Propósito	Brindar al administrador la posibilidad de cambiar su contraseña para acceder al sistema.
Actores: Usuario Registrado(Inicia)	
Resumen: El caso de uso se inicia cuando el actor decide cambiar la contraseña.	
Referencias	RF 3
Precondiciones	El actor tiene que estar autenticado satisfactoriamente por el sistema.

Poscondiciones	El actor cambia la contraseña satisfactoriamente.
-----------------------	---



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Usuario Registrado accede a la opción Cambiar Contraseña(A). (ver Pantalla 1).	2. El Sistema muestra una interfaz donde solicita los datos necesarios para cambiar la contraseña.
3. El Usuario Registrado introduce los datos:(Usuario (B), Contraseña Anterior(C), Nueva ContraseñaD), Confirmar Contraseña (E)) y presiona el botón Aceptar.	4. El sistema valida la identidad del actor y que los datos entrados estén correctos.
	5. El Sistema sustituye la antigua contraseña del Usuario Registrado por la nueva y la guarda en la Base de Datos.

Flujo alternativo	
Acción del actor	Respuesta del sistema

		5. El Sistema muestra un mensaje de error debido a que los datos no han sido entrados correctamente y retorna al paso 3.
Prioridad	Secundario	

Caso de uso	
CU4	Comentar Requisito
Propósito	Hacer un comentario sobre un requisito debido a alguna duda planteada.
Actores: Usuario Intercambio(Inicia)	
Resumen: El caso de uso se inicia cuando el actor decide acotar un comentario acerca de algún requisito existente que no esté del todo claro, con el fin de que se encuentre lo más cercano posible a la realidad.	
Referencias	RF 5

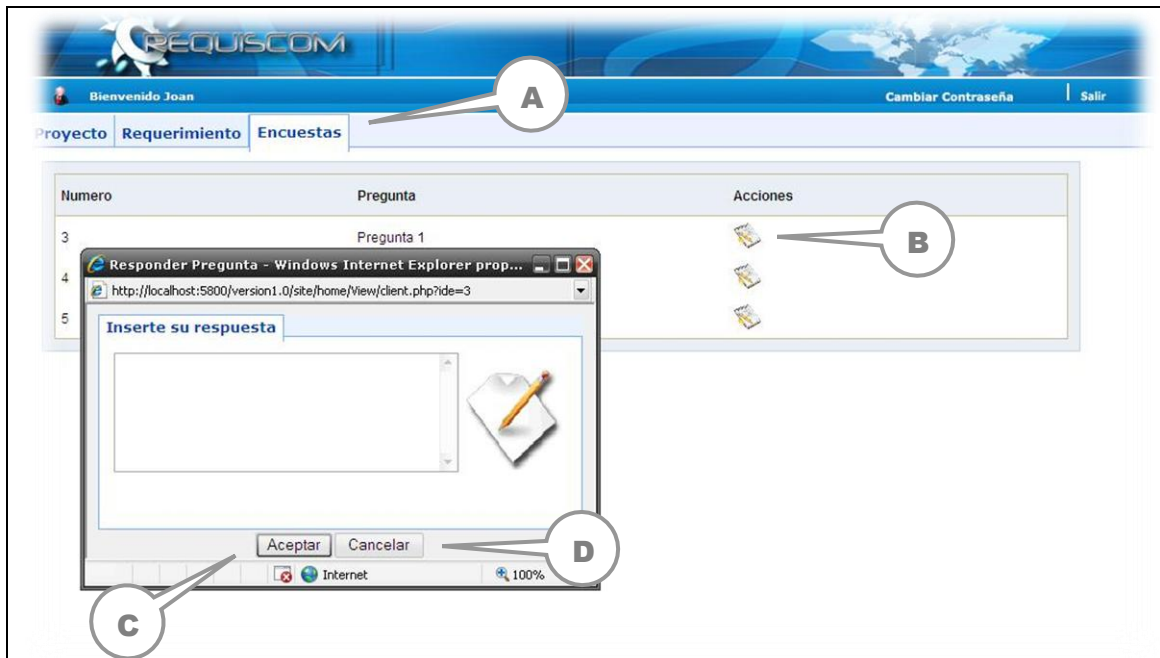
Pantalla 1

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El Usuario Intercambio accede a la pestaña "Requerimientos".(A, ver Pantalla 1)	2. El Sistema muestra la interfaz con el listado de Requerimientos existentes.
3. El Usuario Intercambio selecciona la opción Comentar (B) del requisito al que se quiere hacer el comentario.	4. El Sistema muestra una pequeña interfaz con un cuadro de texto para escribir el comentario.
5. El Usuario Intercambio escribe el comentario y presiona el botón "Aceptar"(C).	6. El Sistema cierra la interfaz, registra el comentario y lo visualiza en la parte derecha del requisito.
Flujo alternativo	
Acción del actor	Respuesta del sistema
5. El Usuario Intercambio presiona el botón "Cancelar" (D).	6. El Sistema retorna a la interfaz anterior.
Prioridad	Crítico

Caso de uso	
CU5	Insertar Comentario Foro
Propósito	Permitir a los usuarios intercambiar información y opiniones acerca de sus respectivos proyectos para lograr un intercambio favorable.
Actores: Usuario Intercambio(Inicia)	
Resumen: Opción que permite al actor enviar mensajes a los diferentes foros a los que tengan acceso.	
Referencias	RF6, RF7
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El Usuario Intercambio accede a la pestaña "Foro".	2. El Sistema le muestra una tabla con los foros (temas) a los que tiene acceso de acuerdo al tipo de usuario (Si es Cliente tendrá acceso al foro del proyecto al que

	está asociado; si es Analista tendrá acceso a los foros de todos los proyectos a los que esté asociado).
3. El Usuario Intercambio selecciona un tema.	4. El Sistema le muestra los mensajes que han sido enviados incluyendo los nombres de las personas que los han emitido. Además, ofrece la oportunidad de responder cualquiera de los mensajes escritos hasta el momento.
5. El Usuario Intercambio selecciona responder un mensaje deseado.	6. El Sistema le muestra un cuadro de texto para escribir el mensaje.
7. EL Usuario Intercambio escribe el mensaje y presiona el botón Aceptar.	8. EL Sistema guarda el mensaje en la Base de Datos y lo muestra a modo de respuesta del mensaje anterior.
Prioridad	Secundario

Caso de uso	
CU6	Responder Encuesta
Propósito	Dar respuesta a las preguntas planteadas por el analista en la encuesta.
Actores: Cliente(Inicia)	
Resumen: El caso de uso se inicia cuando el cliente introduce las respuestas a determinadas preguntas que existan en la encuesta.	
Referencias	RF92
Precondiciones	Deben existir pregunta(s) en la encuesta previamente elaboradas por el analista.



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Cliente selecciona la pestaña “Encuesta”(A, ver Pantalla1).	2. El Sistema muestra una interfaz con el listado preguntas existentes.
3. El Cliente presiona el icono “Responder”(B) correspondiente a la pregunta deseada.	4. El Sistema muestra una pequeña interfaz con un cuadro de texto
5. El Cliente escribe la respuesta de la pregunta y presiona el botón “Aceptar”(C).	6. El Sistema guarda la información en la Base de Datos y la muestra a continuación de la pregunta.

Flujo alternativo

Acción del actor	Respuesta del sistema
5. El Cliente presiona el botón “Cancelar”(D).	6. El Sistema retorna a la interfaz anterior.

Puntos de extensión.

Prioridad	Crítico
------------------	---------

Caso de uso	
CU7	Mostrar Información Proyecto
Propósito	Permitir visualizar la información concerniente con el proyecto al que se encuentre vinculada la persona en cuestión.
Actores: Cliente(Inicia)	
Resumen: El caso de uso se inicia después de haberse autenticado el cliente accediendo a su sesión donde puede visualizar los datos de su proyecto en cuestión.	
Referencias	RF 10

Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Cliente accede a la pestaña "Proyecto"(A, ver Pantalla 1).	2. El Sistema le muestra una interfaz con un listado de los analistas que se encuentran vinculados a su proyecto (B),

	así como las características del mismo(C): Nombre, Perfil, Descripción.
Prioridad	Crítico

Caso de uso	
CU8	Gestionar Foro
Propósito	Organizar los foros de discusión por temas, permitiendo un correcto intercambio entre clientes y analistas.
Actores: Analista(Inicia)	
Resumen: El analista inserta al foro nuevos temas para alcanzar una mejor comunicación con los clientes. Conjuntamente a esto se podrán modificar y/o eliminar los temas ya existentes.	
Referencias	RF 11
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El Analista accede a la pestaña "Foro".	<p>2. El Sistema le muestra una tabla con los foros (temas) a los que tiene acceso teniendo en cuenta el proyecto que haya seleccionado, según los proyectos a los que esté asociado). Además de las opciones correspondientes que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso de que el Analista desee insertar un nuevo tema ir a Sección: Insertar Tema. • En caso de que el Analista desee eliminar un tema existente ir a Sección: Eliminar Tema.
Flujo Normal de Eventos	
Sección: "Insertar Tema"	

1. Accede a crear un nuevo tema en foro del sistema, seleccionando la pestaña Insertar.	2. El Sistema genera la interfaz correspondiente para insertar un nuevo tema, mostrando los campos que deben de ser completados.
3. El Analista llena el campo tema y mensaje inicial y pulsa el botón Aceptar para que se registren los datos en la Base de Datos.	4. El sistema verifica que el tema no exista previamente.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5. El Sistema muestra un mensaje de error debido a que el tema ya existía en la Base de Datos y retorna a la interfaz anterior.
Flujo Normal de Eventos	
Sección: "Eliminar Tema."	
Acción del actor	Respuesta del sistema
1. Accede a eliminar un tema previamente creado.	2. El Sistema genera la interfaz correspondiente para eliminar un tema.
3. El Analista selecciona el tema que desea eliminar y pulsa posteriormente el botón Eliminar (B).	4. El Sistema elimina los datos correctamente en la Base de Datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6. El Sistema muestra un mensaje de

	error debido que no se seleccionó correctamente, y retorna a la acción 3.
Prioridad	Secundario

Caso de uso	
CU9	Gestionar Encuesta
Propósito	Proporcionar a los analistas la posibilidad de agregar preguntas a la encuesta, así como eliminarlas.
Actores: Analista(Inicia)	
Resumen: El caso de uso se inicia cuando el analista adiciona pregunta(s) a la encuesta de un determinado proyecto, con el fin de lograr un mejor intercambio con los clientes. Paralelamente se podrán eliminar las preguntas ya existentes.	
Referencias	RF12

Pantalla 1

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El Analista accede a la pestaña “Encuestas”(A, ver Pantalla1).	<p>2. El sistema muestra la interfaz con el listado de las preguntas existentes, dando la oportunidad de visualizar las respuestas si se pasa el mouse por encima de la pregunta. Además de las opciones correspondientes que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso de que el Analista desee insertar una nueva pregunta ir a Sección: Insertar Pregunta. • En caso de que el Analista desee eliminar una pregunta existente ir a Sección: Eliminar Pregunta.
Flujo Normal de Eventos	
Sección: “Insertar Pregunta”	
1. El Analista escribe en el cuadro texto de la parte superior la pregunta deseada y presiona el botón “Adicionar Pregunta”(B).	4. El Sistema valida que la pregunta no exista.
	5. El Sistema guarda la pregunta en la Base de Datos y la muestra en el listado de preguntas.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5. El Sistema muestra un mensaje de error debido a que la pregunta ya existía en la Base de Datos y retorna a la interfaz

	anterior.
Precondiciones:	Debe existir al menos una pregunta previamente insertada.
Flujo Normal de Eventos	
Sección: "Eliminar Pregunta"	
Acción del actor	Respuesta del sistema
1. El Analista presiona el icono "Eliminar" (C) correspondiente a la pregunta que desea quitar de la encuesta.	2. El Sistema muestra un mensaje preguntando al Analista si está seguro de eliminar la pregunta.
3. El Analista presiona el botón "Sí".	4. El Sistema elimina la pregunta de la Base de Datos y actualiza el listado de preguntas.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón "No".	4. El Sistema cierra la ventana y vuelve al paso 1.
Prioridad	Crítico

Caso de uso	
CU10	Gestionar Requisito
Propósito	Proporcionar a los analistas la posibilidad de agregar requisitos, así como modificar la información referente a ellos y/o eliminarlos.
Actores: Analista(Inicia)	

Resumen: El caso de uso se inicia cuando el analista inserta un requisito.
 Conjuntamente a esto se pueden modificar y/o eliminar los requisitos ya existentes.

Referencias RF 13



Pantalla 1

Flujo Normal de Eventos

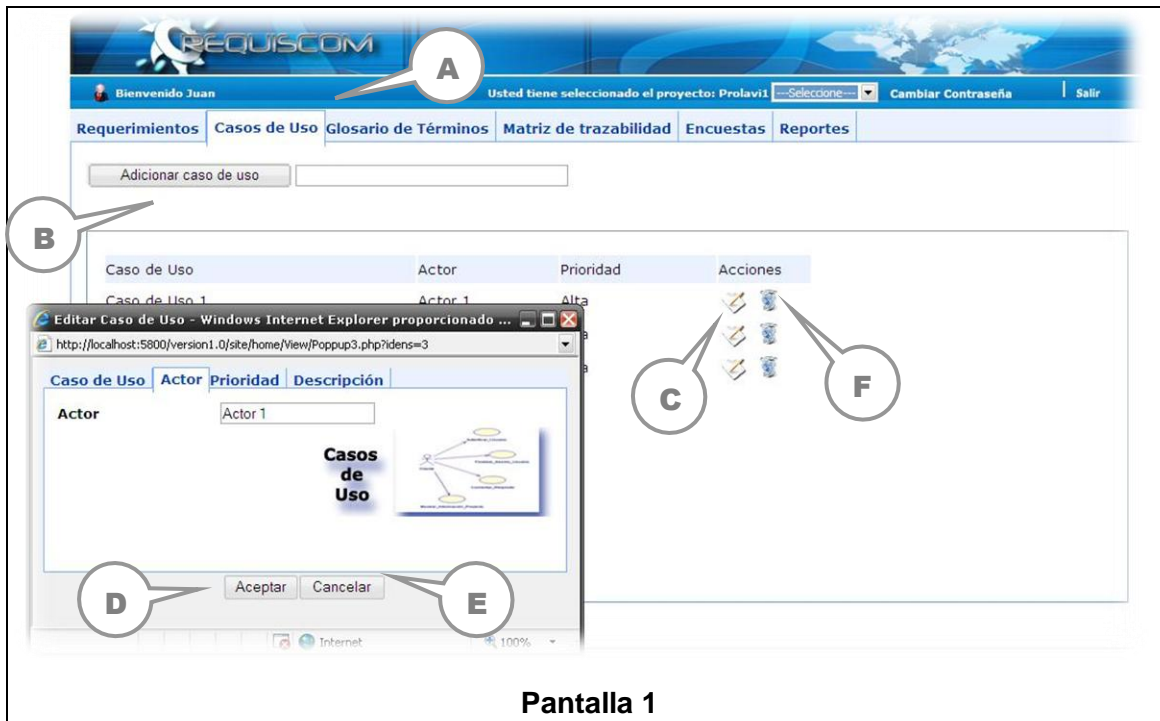
Acción del actor	Respuesta del sistema
1. El Analista accede a la pestaña "Requerimientos". (A, ver Pantalla 1).	El sistema muestra la interfaz con el listado de los requerimientos existentes y las opciones correspondientes que se pueden realizar. <ul style="list-style-type: none"> En caso de que el Analista desee insertar un nuevo requerimiento ir a Sección: Insertar Requerimiento.

	<ul style="list-style-type: none"> • En caso de que el Analista desee modificar un requerimiento existente ir a Sección: Modificar Requerimiento. • En caso de que el Administrador desee eliminar un requerimiento ir a Sección: Eliminar Requerimiento.
Flujo Normal de Eventos	
Sección: “Insertar Requerimiento”	
1. El Analista escribe en el cuadro texto de la parte inferior el nombre del requerimiento y presiona el botón “Adicionar Requerimiento”(B).	2. El Sistema valida que el requerimiento no exista.
	3. El Sistema guarda el requerimiento en la Base de Datos y lo muestra en el listado de requerimientos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3. El Sistema muestra un mensaje de error debido a que el requerimiento ya existía en la Base de Datos y retorna a la interfaz anterior.
Precondiciones:	Debe existir al menos un requerimiento previamente insertado.
Flujo Normal de Eventos	

Sección: “Modificar Requerimiento”	
Acción del actor	Respuesta del sistema
1. El Analista presiona el icono “Editar”(C) correspondiente al requisito deseado.	2. El Sistema muestra una pequeña interfaz (D) con las pestañas Requisito, que incluye los campos Nombre, Prioridad y tipo; la pestaña Caso de Uso asociado, que incluye la opción de seleccionar el caso de uso al que se encuentra asociado y la pestaña Descripción, que incluye el campo Descripción.
3. El Analista llena los campos y presiona el botón “Aceptar”.	4. El Sistema cierra la interfaz y registra la información en la Base de Datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón “Cancelar”.	4. El Sistema cierra la interfaz de edición.
Precondiciones:	Debe existir al menos un requisito previamente insertado.
Flujo Normal de Eventos	
Sección: “Eliminar Requisito”	
Acción del actor	Respuesta del sistema

1. El Analista presiona el icono “Eliminar” (E).	2. El Sistema muestra un mensaje preguntando al Analista si está seguro de eliminar el requerimiento.
3. El Analista presiona el botón “Sí”.	4. El Sistema elimina el requerimiento y actualiza el listado.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón “No”.	4. El Sistema cierra la ventana y vuelve al paso 1.
Prioridad:	Crítico

Caso de uso	
CU11	Gestionar Caso Uso
Propósito	Proporcionar a los analistas la posibilidad de agregar casos de uso, así como modificar la información referente a ellos y/o eliminarlos.
Actores: Analista(Inicia)	
Resumen: El caso de uso se inicia cuando el analista inserta un nuevo caso de uso. Conjuntamente a esto se pueden modificar o eliminar los casos de uso existentes.	
Referencias	RF 14



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
<p>1. El Analista accede a la pestaña “Casos de Uso”. (A, ver Pantalla 1).</p>	<p>El sistema muestra la interfaz con el listado de los casos de uso existentes y las opciones correspondientes que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso de que el Analista desee insertar un nuevo caso de uso ir a Sección: Insertar Caso de Uso. • En caso de que el Analista desee modificar un caso de uso existente ir a Sección: Modificar Caso de Uso. • En caso de que el Administrador desee eliminar un caso de uso ir a

		Sección: Eliminar Caso de Uso.	
Flujo Normal de Eventos			
Sección: “Insertar Caso de Uso”			
1. El Analista escribe en el cuadro texto de la parte inferior el nombre del caso de uso y presiona el botón “Adicionar Caso de Uso”(B).		2. El Sistema valida que el caso de uso no exista.	
		3. El Sistema guarda el caso de uso en la Base de Datos y lo muestra en el listado de casos de uso.	
Flujos Alternos			
Acción del Actor		Respuesta del Sistema	
		3. El Sistema muestra un mensaje de error debido a que el caso de uso ya existía en la Base de Datos y retorna a la interfaz anterior.	
Precondiciones:	Debe existir al menos un caso de uso previamente insertado.		
Flujo Normal de Eventos			
Sección: “Modificar Caso de Uso”			
Acción del actor		Respuesta del sistema	

<p>1. El Analista presiona el icono “Editar”(C) correspondiente al caso de uso que se desea modificar.</p>	<p>2. El Sistema muestra una pequeña interfaz con las pestañas Caso de Uso, que incluye el campo Nombre; Actor, que incluye el campo nombre del Actor; Prioridad, que incluye una lista desplegable para seleccionar la prioridad y Descripción que incluye un campo para introducir la descripción del caso de uso. (Ver Pantalla 2).</p>
<p>3. El Analista llena los campos en las diferentes pestañas y presiona el botón “Aceptar”(D)</p>	<p>4. El Sistema cierra la interfaz y registra la información en la Base de Datos.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>3. El Analista presiona el botón “Cancelar” (E).</p>	<p>4. El Sistema cierra la interfaz de edición.</p>
<p>Precondiciones:</p>	<p>Debe existir al menos un caso de uso previamente insertado.</p>
<p style="text-align: center;">Flujo Normal de Eventos</p>	
<p style="text-align: center;">Sección: “Eliminar Caso de Uso”</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. El Analista presiona el icono “Eliminar” (F) correspondiente al caso de uso deseado.</p>	<p>2. El Sistema muestra un mensaje preguntándole al Analista si está segura de eliminar el caso de uso.</p>

3. El Analista presiona el botón “Sí”.	4. El Sistema elimina el caso de uso y actualiza el listado.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón “No”.	4. El Sistema cierra la ventana y vuelve al paso 1.
Prioridad	Crítico

Caso de uso	
CU12	Mostrar Matriz Trazabilidad
Propósito	Permite definir las relaciones entre requisitos y casos de uso del proyecto en cuestión, además de facilitar el control de los cambios contribuyendo a reducir el esfuerzo y lograr mayor agilidad en el desarrollo.
Actores: Analista(Inicia)	
Resumen: El caso de uso se inicia cuando el analista accede a visualizar la matriz de trazabilidad creada a partir de las relaciones establecidas entre los requisitos y los casos de uso que se hayan determinado.	
Referencias	RF15
Precondiciones	Deben existir requisitos y casos de uso previamente listados por parte del analista.

Matriz de Trazabilidad

	Caso de Uso 1	Caso de Uso 2	Caso de Uso 3
Ejemplo1	X		
Ejemplo2		X	
Ejemplo3			X

Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Analista accede a la pestaña “Matriz de Trazabilidad”(A, ver Pantalla 1)	2. El Sistema muestra una interfaz con una matriz que muestra horizontalmente los requerimientos y verticalmente los casos de uso del proyecto en cuestión, señalando con una “X” en caso de estar relacionados.
Prioridad:	Crítico

Caso de uso	
CU13	Gestionar Glosario Términos

Propósito	Permite definir y explicar los términos que son utilizados en los requerimientos del proyecto.
Actores:	Analista(Inicia)
Resumen:	El caso de uso se inicia cuando el analista inserta nuevos términos, organizándolos con sus respectivos significados. Conjuntamente a esto se pueden modificar o eliminar los términos existentes.
Referencias	RF16
Precondiciones	Deben existir requisitos listados por parte del analista.



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Analista accede a la pestaña “Glosario	El sistema muestra la interfaz con el listado de los términos existentes y las

de Términos”. (A, ver Pantalla 1).	<p>opciones correspondientes que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso de que el Analista desee insertar un nuevo término ir a Sección: Insertar Término. • En caso de que el Analista desee modificar un término existente ir a Sección: Modificar Término. • En caso de que el Administrador desee eliminar un término ir a Sección: Eliminar Término.
Flujo Normal de Eventos	
Sección: “Insertar Término”	
1. El Analista escribe en el cuadro texto de la parte inferior el nombre del término y presiona el botón “Adicionar Término”(B).	2. El Sistema valida que el término no exista.
	3. El Sistema guarda el término en la Base de Datos y lo muestra en el listado de términos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3. El Sistema muestra un mensaje de error debido a que el término ya existía en la Base de Datos y retorna a la interfaz

	anterior.
Precondiciones:	Debe existir al menos un término previamente insertado.
Flujo Normal de Eventos	
Sección: “Modificar Término”	
Acción del actor	Respuesta del sistema
1. El Analista selecciona el icono “Editar” (C) que se encuentra en la parte derecha de cada término.	2. El Sistema muestra una pequeña interfaz con las pestañas Término (para dar la posibilidad de cambiar el nombre del término) y Descripción (con un cuadro de texto para escribir la descripción del término.)
3. El Analista llena los campos y presiona el botón “Aceptar”(D).	4. El Sistema cierra la interfaz, registra la información y lo visualiza en la parte derecha del término.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón “Cancelar” (E).	4. El Sistema cierra la interfaz y regresa a la anterior.
Precondiciones:	Debe existir al menos un término previamente insertado.
Flujo Normal de Eventos	
Sección: “Eliminar Término”	
Acción del actor	Respuesta del sistema

1. El Analista presiona el icono “Eliminar” (F) que se encuentra en la parte derecha de cada término.	2. El Sistema muestra un mensaje preguntándole al Analista si está seguro de eliminar el término.
3. El Analista presiona el botón “Sí”.	4. El Sistema elimina el término y actualiza el listado.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Analista presiona el botón “No”.	4. El Sistema cierra la ventana y vuelve al paso 1.
Prioridad:	Crítico

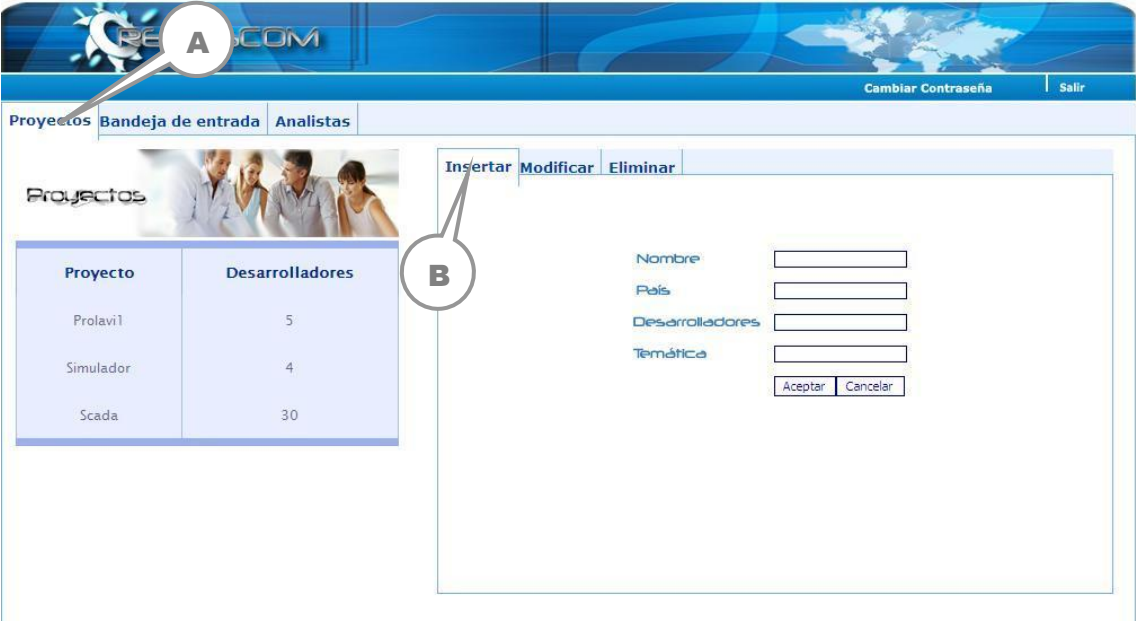
Caso de uso	
CU14	Generar Reporte
Propósito	Mostrar toda la información que se ha podido generar a partir de todo el intercambio realizado entre analistas y clientes en la aplicación.
Actores: Analista(Inicia)	
Resumen: El caso de uso se inicia cuando el analista desea generar un documento a partir de la información obtenida con la recopilación de los requisitos, los casos de uso y glosario de términos.	
Referencias	RF17



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Analista selecciona la pestaña Reportes(A, ver Pantalla 1).	2. El Sistema le muestra una interfaz con la opción de generar el reporte del “Documento de Requerimientos” (B).
3. El Analista selecciona la opción deseada.	4. El Sistema le muestra un documento en formato .pdf con la información el Analista seleccionó previamente.
Prioridad:	Secundario

Caso de uso	
CU15	Gestionar Proyecto
Propósito	Brindar la posibilidad al administrador de actualizar la información referente a los proyectos existentes.
Actores: Administrador(Inicia)	
Resumen: El caso de uso se inicia cuando al administrador introduce los datos de un proyecto para crearlo. Así como también puede modificarlos y/o eliminarlos.	
Referencias	RF 18
Precondiciones	El administrador debe estar autenticado
 <p>The screenshot displays a web application interface for managing projects. At the top, there is a navigation bar with 'Proyectos', 'Bandeja de entrada', and 'Analistas' tabs. Below this, a table lists existing projects with columns for 'Proyecto' and 'Desarrolladores'. To the right, there is a form with fields for 'Nombre', 'País', 'Desarrolladores', and 'Temática', along with 'Aceptar' and 'Cancelar' buttons. A callout 'A' points to the 'Proyectos' tab, and a callout 'B' points to the 'Insertar' button in the form's header.</p>	
Pantalla 1	
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la pestaña Proyectos(A, ver Pantalla 1).	2. El sistema muestra un listado con los proyectos existentes y además la interfaz

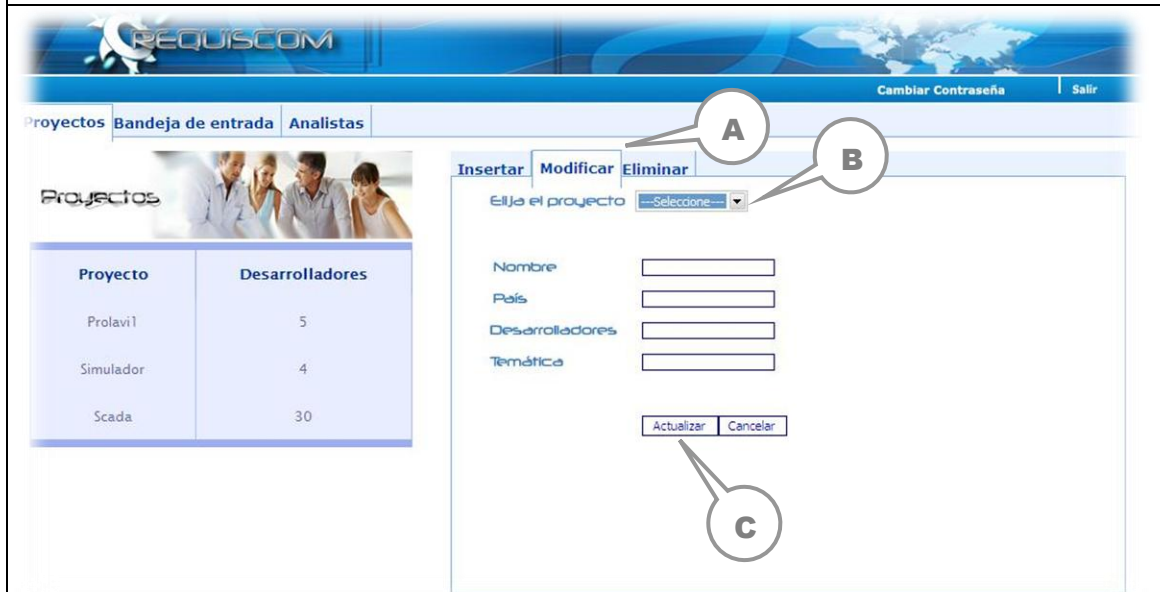
	<p>correspondiente a las actividades que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso de que el Administrador desee crear un nuevo proyecto ir a Sección: Insertar Proyecto. • En caso de que el Administrador desee modificar la información referente a un proyecto ir a Sección: Actualizar Proyecto. • En caso de que el Administrador desee eliminar un proyecto existente ir a Sección: Eliminar Proyecto.
Flujo Normal de Eventos	
Sección: “Insertar Proyecto”	
<p>1. Accede a crear un nuevo proyecto, seleccionando la pestaña Insertar Proyecto (B).</p>	<p>2. El Sistema genera la interfaz correspondiente para crear un nuevo proyecto, mostrando los campos que deben de ser completados para crear el proyecto.</p>
<p>3. El Administrador completa cada campo: Nombre, País, Cantidad de Desarrolladores, Perfil, Descripción; y pulsa el botón Aceptar para que se registren los datos en la Base de Datos.</p>	<p>4. El sistema verifica que el nombre del proyecto no exista previamente.</p>
	<p>5. El Sistema almacena los datos correctamente en la Base de Datos.</p>
Flujos Alternos	

Acción del Actor	Respuesta del Sistema
	5. El Sistema muestra un mensaje de error debido a que el nombre del proyecto ya existe en la Base de Datos, y retorna a la acción 3.

Precondiciones:	El proyecto a modificar se debe haber creado con anterioridad.
------------------------	--

Flujo Normal de Eventos

Sección: “Actualizar Proyecto”

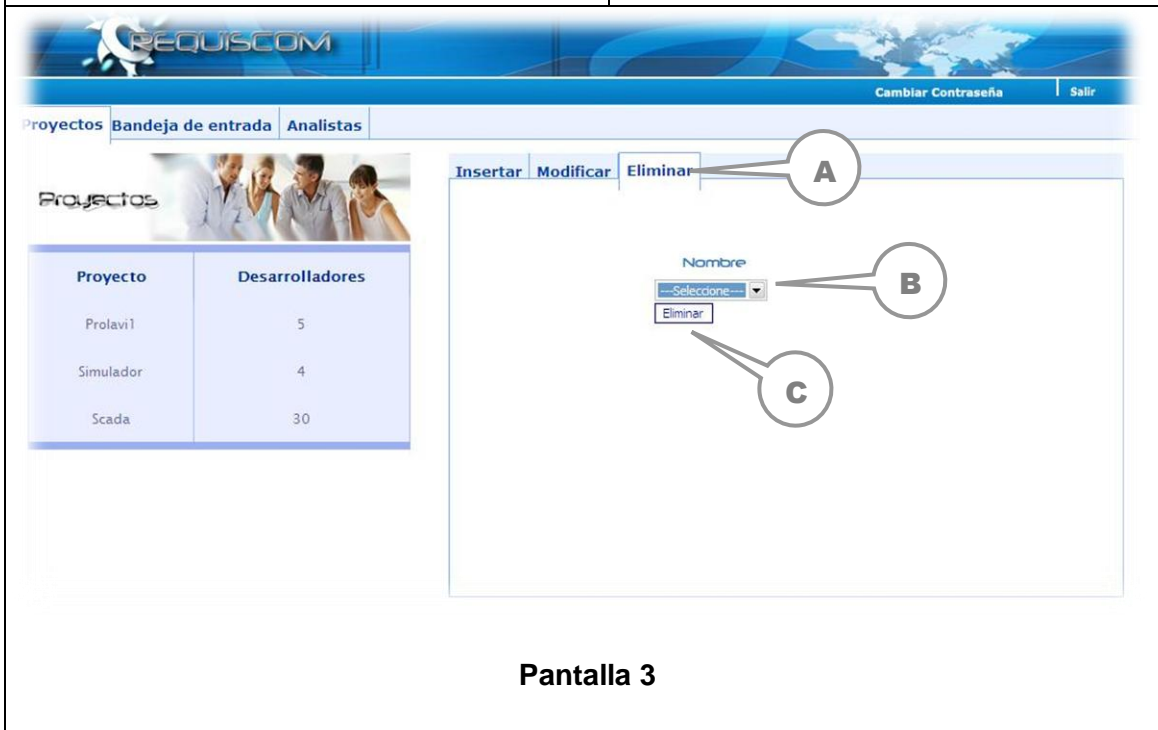


Pantalla 2

Acción del actor	Respuesta del sistema
1. Accede a modificar un proyecto previamente creado(A, ver Pantalla 2).	2. El Sistema genera una interfaz que muestra una lista con los proyectos existentes.

3. El administrador selecciona el proyecto(B) al que se le realizarán las modificaciones.	4. El Sistema muestra los campos que pueden ser modificados para actualizar la información del proyecto.
5. El Administrador completa cada campo: Nombre, País, Cantidad de Desarrolladores, Perfil, Descripción; para que se registren los datos en la Base de Datos con la información correspondiente para actualizar los datos del proyecto, pulsando posteriormente el botón "Actualizar"(C).	6. El sistema verifica que los campos obligatorios estén llenados correctamente.
	7. El Sistema almacena los datos correctamente en la Base de Datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6. El Sistema muestra un mensaje de error debido a que ha existido algún error con la entrada de los datos, y retorna a la acción 5.
Precondiciones:	El proyecto a eliminar se debe haber creado con anterioridad.
Flujo Normal de Eventos	
Sección: "Eliminar Proyecto"	
Acción del actor	Respuesta del sistema

1. Accede a eliminar un proyecto previamente creado.(A, ver Pantalla 3)	2. El Sistema genera la interfaz correspondiente para eliminar un usuario del sistema.
3. El Administrador selecciona el proyecto (B) que desea eliminar y pulsa el botón “Eliminar”(C).	4. El Sistema elimina los datos correctamente, actualizando la Base de Datos.



Prioridad	Crítico
------------------	---------

Caso de uso	
CU16	Gestionar Grupo Analistas
Propósito	Organizar los equipos de analistas en los distintos proyectos que se realicen.

Actores: Administrador(Inicia)	
Resumen: El caso de uso se inicia cuando el administrador decide insertar analista(s) al equipo de un proyecto creado previamente.	
Referencias	RF 19
Precondiciones	Deben existir proyectos creados previamente, además de analistas registrados en el sistema.
Poscondiciones	Se conforman los equipos satisfactoriamente.

Pantalla 1



Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
------------------	-----------------------

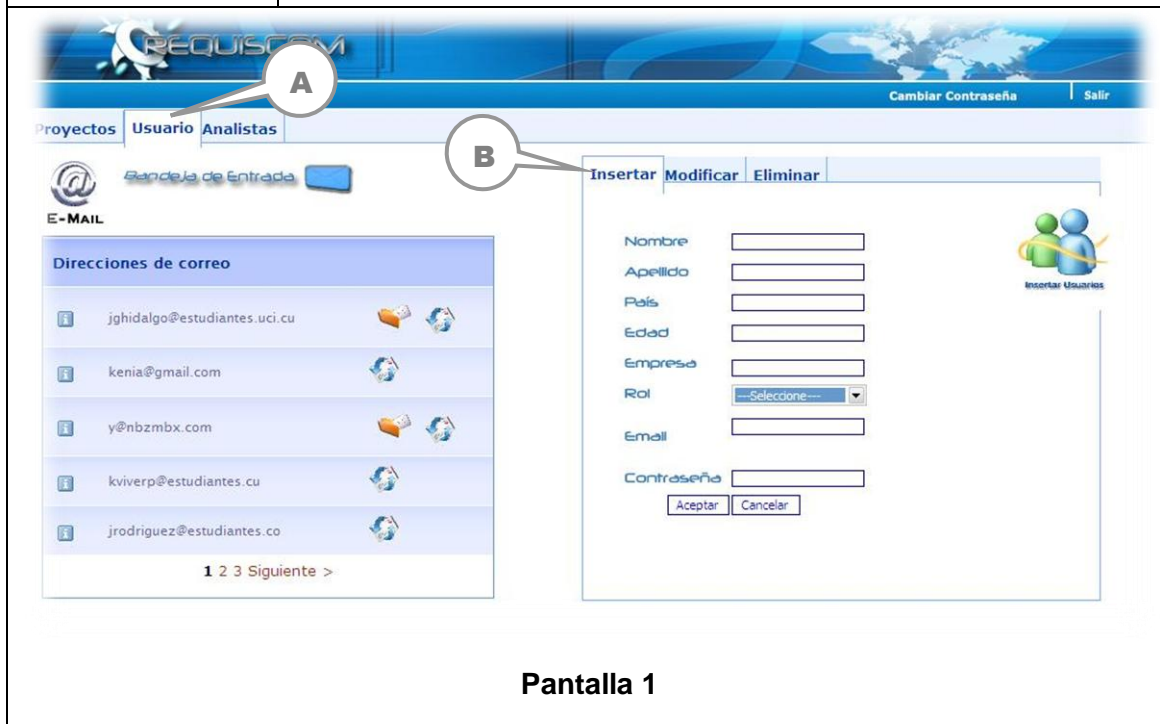
<p>1. El administrador accede a la pestaña Analistas. (A, ver Pantalla 1).</p>	<p>2. El Sistema muestra una interfaz para la gestión de los analistas: El menú de la izquierda muestra los analistas registrados en el Sistema (B); el menú de la derecha tiene la opción de seleccionar el proyecto al que se le quiere agregar un analista(C), y debajo un listado de los analistas que pertenecen a ese proyecto (D).</p>
	<p>3. El Sistema ofrece la oportunidad de realizar las actividades correspondientes:</p> <ul style="list-style-type: none"> • En caso que el Administrador desee agregar un analista en un determinado proyecto ir a Sección: Agregar Analista. • En caso que el Administrador desee quitar un analista de un determinado proyecto ir a Sección: Eliminar Analista.
<p>Flujo Normal de Eventos</p>	
<p>Sección: “Agregar Analista”</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. El Administrador selecciona del menú “Grupo de analistas” el proyecto al que desea agregar un analista en el equipo.</p>	<p>2. El Sistema muestra un listado con los analistas que actualmente posee el proyecto.</p>

<p>1. El Administrador selecciona del menú “Analistas” el que desea agregar, y presiona el botón “=>”.</p>	<p>2. El Sistema valida que se hayan seleccionado el analista y el proyecto al que se desea agregar, correctamente. Además, comprueba que el proyecto al que se quiere agregar el analista no tenga el número máximo requerido de analistas.</p>
	<p>4. El Sistema integra al nuevo analista al equipo del proyecto en cuestión, correctamente.</p>
	<p>5. El Sistema muestra una tabla en la parte derecha con el listado actualizado de analistas del proyecto seleccionado.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>4. El Sistema muestra un mensaje de error debido a que no se ha realizado correctamente la selección del analista y/o el proyecto.</p>
<p>Precondiciones:</p>	
<p>Flujo Normal de Eventos</p>	
<p>Sección: “Eliminar Analista”</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>

6. El Administrador selecciona del menú "Grupo de analistas" el proyecto al que desea realizar cambios en su equipo.	2. El Sistema muestra un listado con los analistas que actualmente posee el proyecto.
3. El Administrador selecciona de la lista el analista que desea quitar del equipo, y presiona el botón "<=".	5. El Sistema valida que se haya seleccionado el analista correctamente.
	6. El Sistema sustrae al analista del equipo del proyecto en cuestión, correctamente y actualiza la Base de Datos.
	7. El Sistema muestra una tabla en la parte derecha con el listado actualizado de analistas del proyecto seleccionado.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6. El Sistema muestra un mensaje de error debido a que no se ha realizado correctamente la selección del analista.
Prioridad	Crítico

Caso de uso	
CU17	Gestionar Usuario

Propósito	Proporcionar al administrador la posibilidad de crear nuevos usuarios en el sistema, así como modificar y eliminar los ya existentes.
Actores:	Administrador(Inicia)
Resumen:	El caso de uso se inicia cuando el administrador introduce los datos de un cliente/analista para registrarle y termina cuando el administrador le envía al usuario un email su nombre de usuario y contraseña. Conjuntamente a esto se podrán modificar los datos y/o eliminar usuarios ya existentes.
Referencias	RF 20
Precondiciones	El administrador debe estar autenticado



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
-------------------------	------------------------------

<p>1. El administrador selecciona la pestaña Usuario(A, ver Pantalla 1).</p>	<p>2. El sistema muestra en la parte derecha la interfaz correspondiente a las actividades que se pueden realizar.</p> <ul style="list-style-type: none"> • En caso que el Administrador desee crear un nuevo usuario ir a Sección: Crear Usuario del Sistema. • En caso que el Administrador desee modificar un usuario ir a Sección: Modificar Usuario del Sistema. • En caso que el Administrador desee eliminar un usuario ir a Sección: Eliminar Usuario del Sistema.
<p>Flujo Normal de Eventos</p>	
<p>Sección: “Crear Usuario del Sistema”</p>	
<p>1. Accede a crear un nuevo usuario del sistema, seleccionando la pestaña Insertar (B).</p>	<p>2. El Sistema genera la interfaz correspondiente para insertar un nuevo usuario, mostrando los campos que deben de ser completados para crear el proyecto.</p>
<p>3. El Administrador completa cada campo: Nombre, Apellidos, Email, País, Edad, Empresa, Rol, Contraseña pulsa el botón Aceptar para que se registren los datos en la Base de Datos.</p>	<p>4. El sistema verifica que el nombre del usuario no exista previamente.</p>

	5. El Sistema almacena los datos correctamente en la Base de Datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5. El Sistema muestra un mensaje de error debido a que el nombre del usuario ya existe en la Base de Datos, y retorna a la acción 3.
Precondiciones:	El usuario al que se necesitan modificar sus datos se debe haber creado con anterioridad.

Pantalla 2

Flujo Normal de Eventos	
Sección: “Modificar Usuario del Sistema.”	
Acción del actor	Respuesta del sistema
1. Accede a modificar los datos de un usuario previamente creado.(A, ver Pantalla 2)	2. El Sistema genera una interfaz que da la opción de seleccionar de una lista el usuario registrado que se desee.
3. El administrador selecciona el usuario al que se le realizarán las modificaciones (B).	4. El Sistema genera la interfaz correspondiente para actualizar los datos del usuario, mostrando los campos que pueden ser modificados para actualizar la información del personal del usuario.
5. El Administrador completa cada campo: Nombre, Apellidos, Email, País, Edad, Empresa, Rol, Contraseña y Confirmar Contraseña; y pulsa el botón Actualizar para que se registren los datos en la Base de Datos.	6. El sistema verifica que los campos obligatorios estén llenados correctamente.
	7. El Sistema almacena los datos correctamente en la Base de Datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6. El Sistema muestra un mensaje de error debido a que todos los campos no, y retorna a la acción 5.

Precondiciones: El proyecto a eliminar se debe haber creado con anterioridad.



Pantalla 3

Flujo Normal de Eventos

Sección: “Eliminar Usuario del Sistema.”

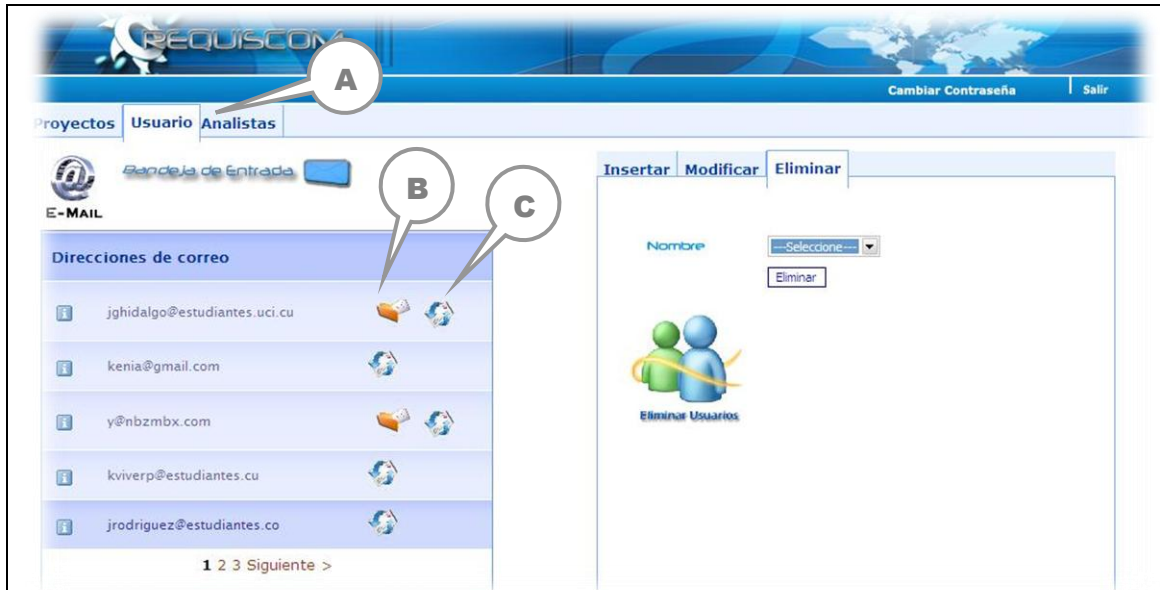
Acción del actor	Respuesta del sistema
1. Accede a eliminar la información de un usuario previamente creado.(A, ver Pantalla 3)	2. El Sistema genera la interfaz correspondiente para eliminar un usuario del sistema.
3. El Administrador selecciona el proyecto que desea eliminar y pulsa posteriormente el botón eliminar (B).	4. El Sistema elimina los datos correctamente en la Base de Datos.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

	6. El Sistema muestra un mensaje de error debido a que todos los campos no fueron llenados correctamente, y retorna a la acción 5.
Prioridad	Crítico

Caso de uso	
CU18	Enviar Email Notificación
Propósito	Notificar al cliente si su propuesta de proyecto fue aprobada o no.
Actores: Administrador(Inicia)	
Resumen: El caso de uso se inicia cuando el administrador envía al cliente un email informándole si fue o no aprobada su propuesta. En caso positivo le envía también su cuenta de usuario y contraseña, así como el link a la página de la aplicación.	
Referencias	RF 21
Precondiciones	El administrador debe haber recibido previamente un determinado correo del cliente con la propuesta del sistema que este quiere que le desarrollen.

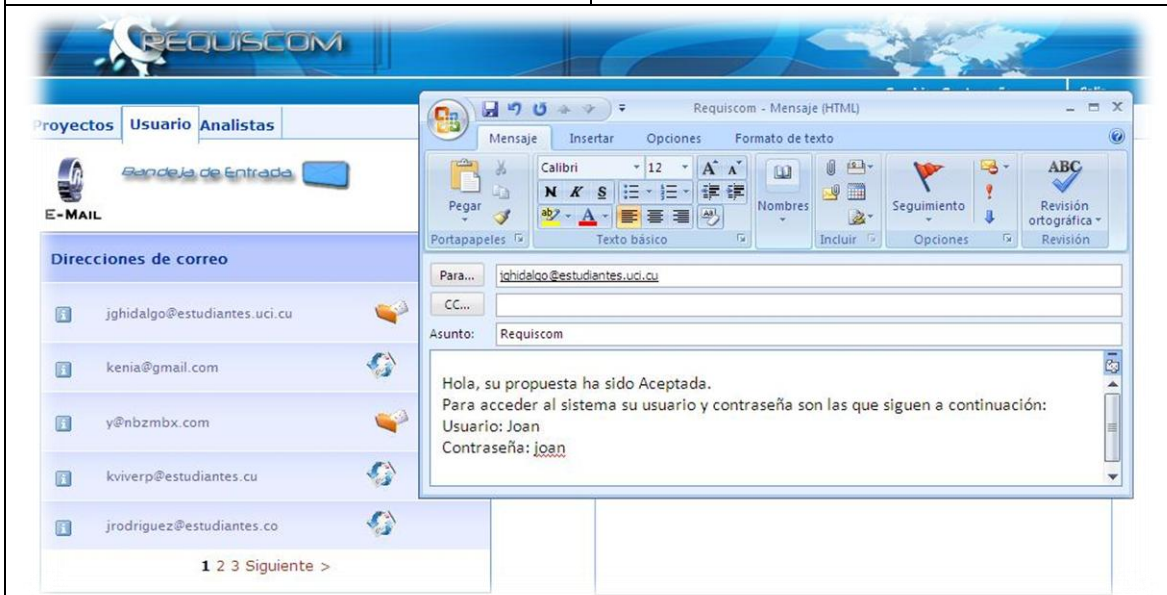


Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El administrador selecciona la pestaña Usuario(A, ver Pantalla 1).	2. El Sistema genera una interfaz que muestra en la parte izquierda las direcciones electrónicas de los clientes que han enviado correos con las propuestas a desarrollar y las opciones de Notificar Aprobar (B) y Notificar Denegar (C).
3. El administrador selecciona un correo.	4. El sistema muestra un cuadro de texto con el contenido del correo enviado por el cliente.
5. El administrador presiona la opción deseada, de acuerdo con la decisión	6. El Sistema muestra una interfaz(ver Pantalla 2) con los campos Destinatario, Asunto y el Cuerpo del Correo, en el que

<p>tomada.</p>	<p>se le especifica si su propuesta de proyecto ha sido aprobada o no.(En caso de ser aprobada se enviará el usuario y contraseña de acceso al sistema)</p>
----------------	---



Pantalla 2

<p>7. El administrador presiona el botón “Enviar”.</p>	<p>8. El Sistema envía el correo a la dirección electrónica del cliente seleccionado.</p>
--	---

<p>Prioridad</p>	<p>Crítico</p>
-------------------------	----------------

<p>Caso de uso</p>	
<p>CU19</p>	<p>Enviar Email Propuesta</p>
<p>Propósito</p>	<p>Enviar al administrador un email con la propuesta de proyecto además de los datos personales del cliente.</p>

Actores: Usuario no registrado(Inicia)
Resumen: El caso de uso se inicia cuando el actor introduce los datos personales y una descripción del lo que quieren que le hagan y es enviado al administrador para posterior análisis de la propuesta.

Referencias	RF 22
--------------------	-------



Pantalla 1

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El Usuario no registrado necesita enviar un email de propuesta.	2. El Sistema muestra la interfaz inicial con las opciones de autenticarse y de enviar email (ver Pantalla 1. Entrar (A) y Enviar Email (B)).
3. El Usuario no registrado presiona el botón de “Enviar Email”.	6. El Sistema muestra una interfaz (ver Pantalla 2) con los campos Destinatario, Asunto y el Cuerpo del Correo, en donde

	se deben llenar los datos personales del cliente.
7. El Usuario no registrado escribe la información necesaria y presiona el botón “Enviar”(A).	8. El Sistema envía el correo al administrador.



Pantalla 2

Flujo alternativo	
Acción del actor	Respuesta del sistema
7. El Usuario no registrado presiona el botón “Cancelar” (B).	8. El Sistema retorna a la interfaz inicial.
Prioridad	Crítico

Capítulo 3. Desarrollo de la solución propuesta

3.1 Introducción.

En este capítulo se muestran elementos relacionados con el diseño del sistema, tales como los diagramas de clases de diseño con extensiones web con sus respectivos diagramas de secuencia. Se añade al mismo tiempo, una descripción del diseño de la base de datos mediante el diagrama de clases persistentes y el modelo de datos. También se abordarán brevemente los patrones de diseño y arquitectura que se utilizarán y finalizando se presentará el modelo de implementación mediante el diagrama de despliegue que resultó del diseño realizado de cada uno de los casos de uso del sistema.

3.2 Patrones de diseño.

Los patrones son parejas de problema/solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Asignar correctamente las responsabilidades es muy importante en el diseño orientado a objetos. (LARMAN)

En el diseño de la aplicación se tuvieron en cuenta los patrones GRASP (General Responsibility Assignment Software Patterns) principalmente los patrones Experto y Controlador. El primero plantea que siempre se debe asignar una responsabilidad al experto en información, o sea, la clase con toda la información necesaria para llevarla a cabo. El segundo expresa asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora.

3.3 Patrón Modelo Vista Controlador.

La arquitectura Modelo Vista Controlador (MVC) divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

- **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Se han desarrollado a lo largo de los años, desde la presentación de este patrón a la comunidad científica 3 variantes fundamentales, que se presentan brevemente a continuación.

Variante I: Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista y esta última recibe los datos a mostrar a través del Controlador.

Variante II: (Figura 3.1)

Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos la busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades de este último.

Variante III:

Variante en la cual se diversifica las funcionalidades del Modelo teniendo en cuenta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

Para la realización de la arquitectura del software se empleó la Variante II del patrón MVC.

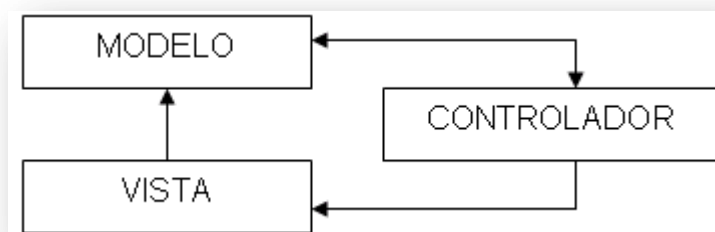


Fig.3.1 Variante Intermedia del Patrón MVC.

3.4 Diagrama de clases con extensiones Web.

Cuando se habla de modelar aplicaciones Web con UML se puede apreciar que difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio, que los detalles de presentación.

Desde el punto de vista del impacto que tienen en el sistema, el diagrama de diseño Web se centra en los requerimientos funcionales y no funcionales, y los analiza conjuntamente con otras restricciones relacionadas con el entorno de implementación. Es precisamente este diagrama el que permite mostrar las clases con sus atributos y métodos, e incluso, representar la colaboración que tiene lugar entre ellas. Se modelan además las páginas con los enlaces que las interrelacionan.

De acuerdo a la forma en que se ha organizado el contenido del trabajo, se considera que se debe presentar los modelos organizados por subsistemas (que coinciden con los paquetes en los que se definieron anteriormente los casos de uso), de forma que se pueda entender mejor la lógica del sistema.

1. **Subsistema 1.** Usuario:

Casos de Uso: Autenticar Usuario, Cambiar Contraseña, Finalizar_Sesión.

2. **Subsistema 2.** Intercambio:

Casos de Uso: Comentar Requisito, Insertar Comentario Foro, Generar Reporte, Gestionar Requisito, Gestionar Caso Uso, Gestionar Foro, Gestionar Glosario Términos, Gestionar Encuesta, Mostrar Matriz Trazabilidad, Responder Encuesta, Mostrar Información Proyecto.

3. **Subsistema 3.** Administración.

Casos de Uso: Gestionar Proyecto, Gestionar Usuario, Gestionar Grupo Analistas, Enviar Email Notificación.

4. **Subsistema 4.** Propuesta.

Caso de Uso: Enviar Email Propuesta.

A continuación se presentan los diagramas de clases del diseño de los subsistemas que representan los procesos y diagramas de interacción para la realización de los casos de usos críticos descritos en el capítulo anterior.

Subsistema 1. Usuario.

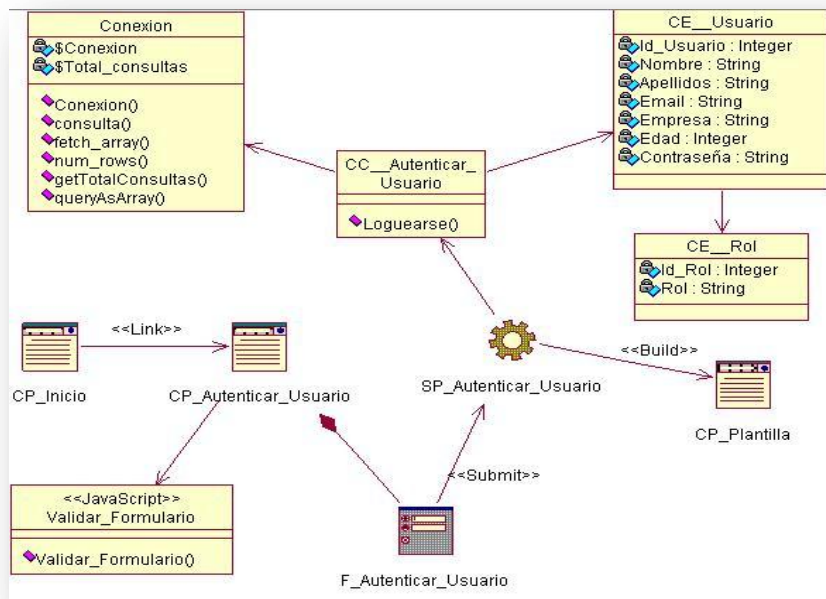


Fig. 3.2 Diagrama de Clases del Diseño “Autenticar Usuario.”

Diagrama de secuencia del caso de uso “Autenticar Usuario”. [Ver Anexo 1]

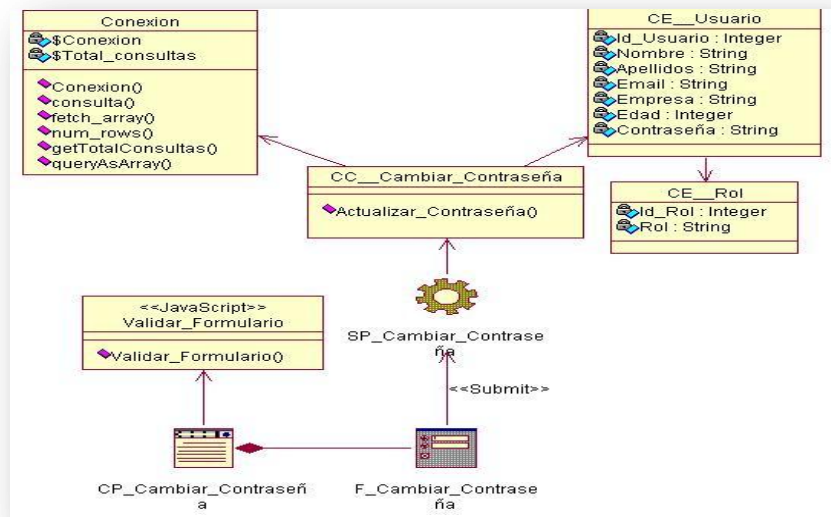


Fig. 3.3 Diagrama de Clases del Diseño “Cambiar Contraseña.”

Diagrama de secuencia del caso de uso “Cambiar Contraseña”. [Ver Anexo 2]

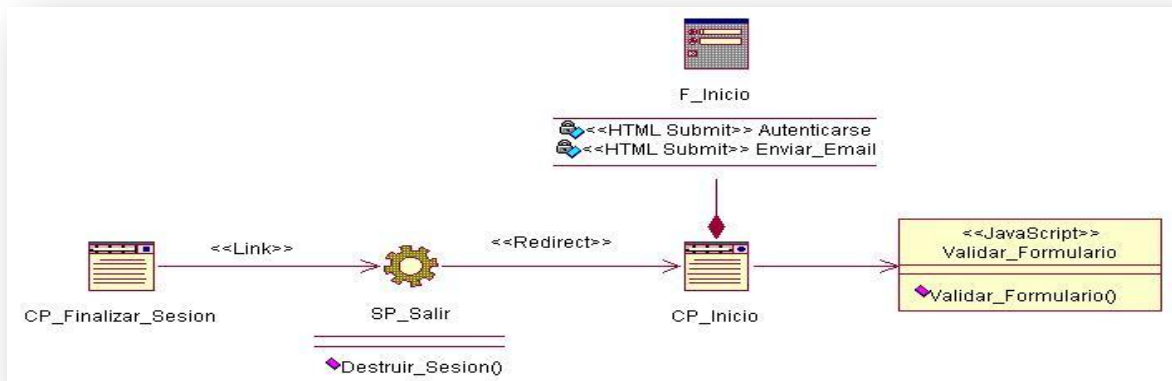


Fig. 3.4 Diagrama de Clases del Diseño “Finalizar Sesión Usuario.”

Diagrama de secuencia del caso de uso “Finalizar Sesión Usuario”. [Ver Anexo 3]

Subsistema 2. Intercambio.

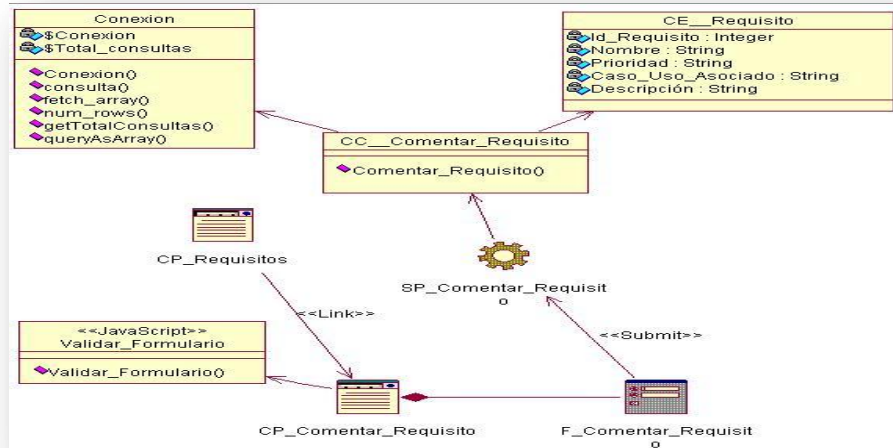


Fig. 3.5 Diagrama de Clases del Diseño “Comentar Requisito.”

Diagrama de secuencia del caso de uso “Comentar Requisito”. [Ver Anexo 4]

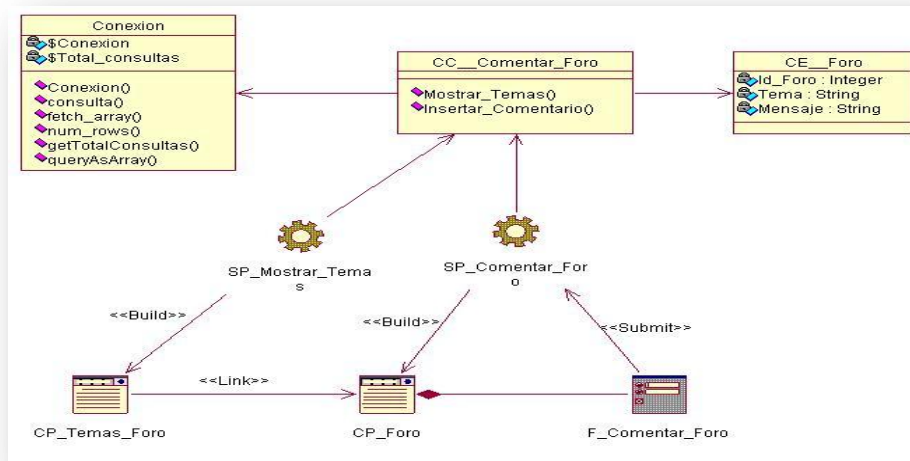


Fig. 3.6 Diagrama de Clases del Diseño “Insertar Comentario Foro.”

Diagrama de secuencia del caso de uso “Insertar Comentario Foro”. [Ver Anexo 5]

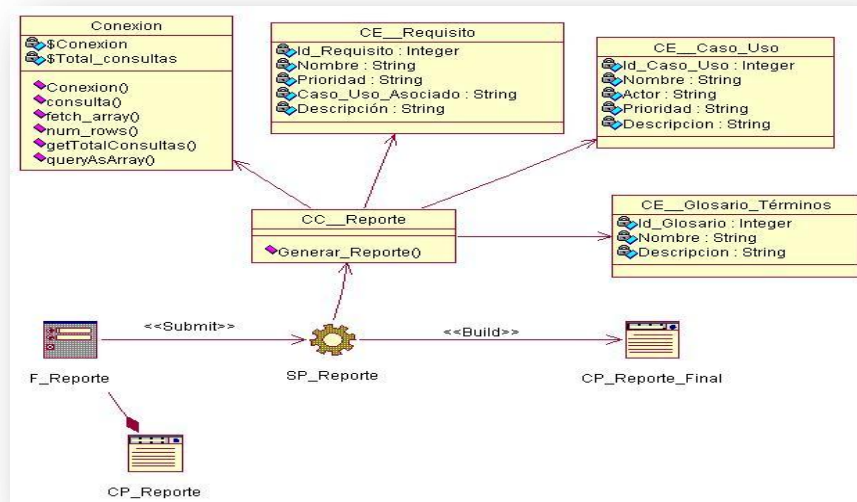


Fig. 3.7 Diagrama de Clases del Diseño “Generar Reporte”.

Diagrama de secuencia del caso de uso “Generar Reporte”. [Ver Anexo 6]

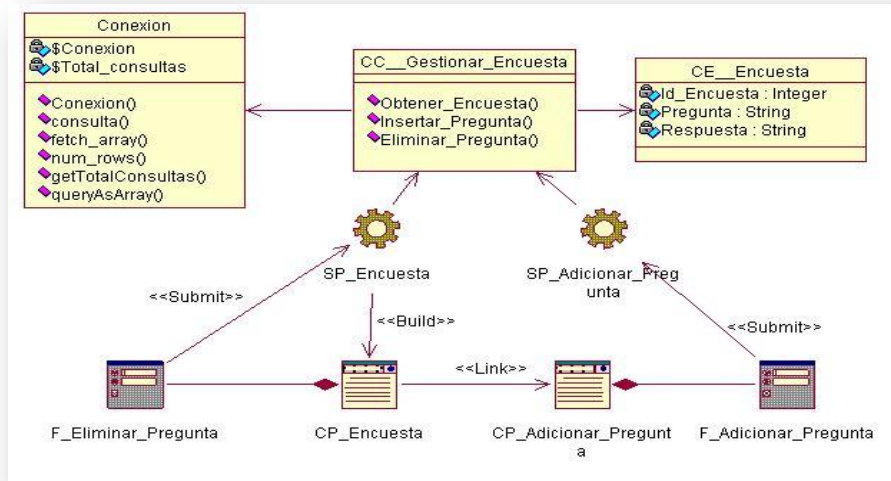


Fig. 3.8 Diagrama de Clases del Diseño “Gestionar Encuesta”.

Diagrama de secuencia del caso de uso “Gestionar Encuesta”. [Ver Anexo 7]

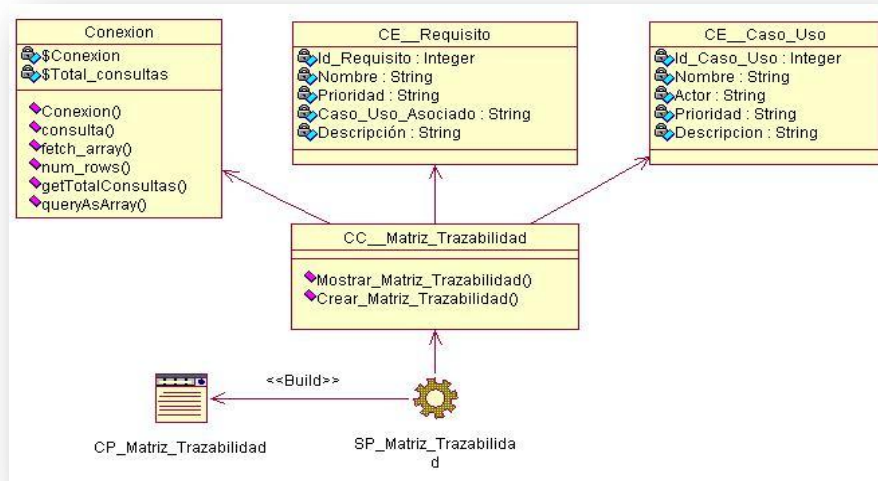


Fig. 3.9 Diagrama de Clases del Diseño “Mostrar Matriz Trazabilidad.”

Diagrama de secuencia del caso de uso “Mostrar Matriz Trazabilidad”. [Ver Anexo 8]

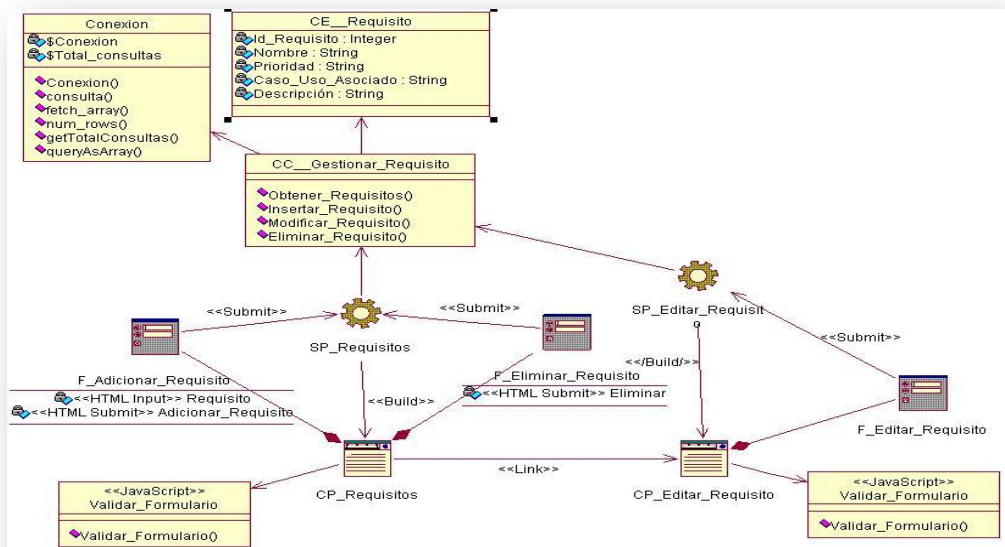


Fig. 3.10 Diagrama de Clases del Diseño “Gestionar Requisito.”

Diagrama de secuencia del caso de uso “Gestionar Requisito”. [Ver Anexo 9]

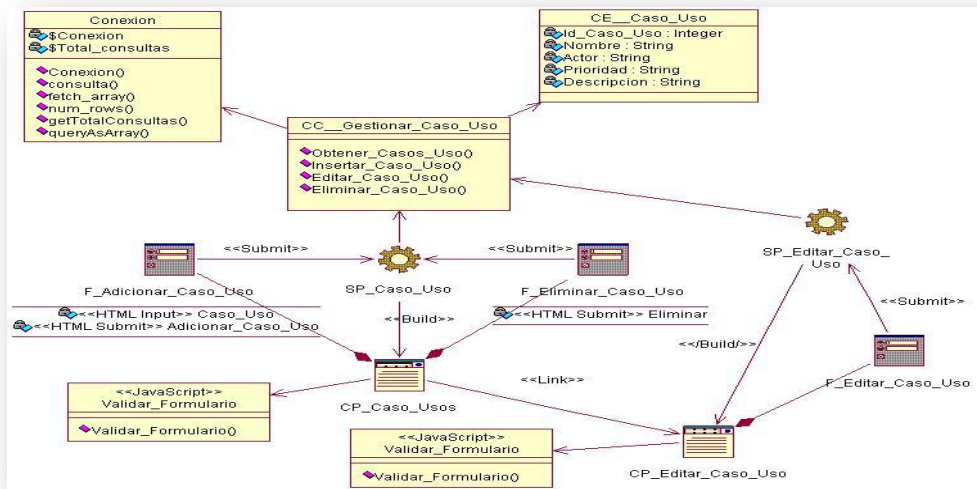


Fig. 3.11 Diagrama de Clases del Diseño “Gestionar Caso Uso.”

Diagrama de secuencia del caso de uso “Gestionar Caso Uso”. [Ver Anexo 10]

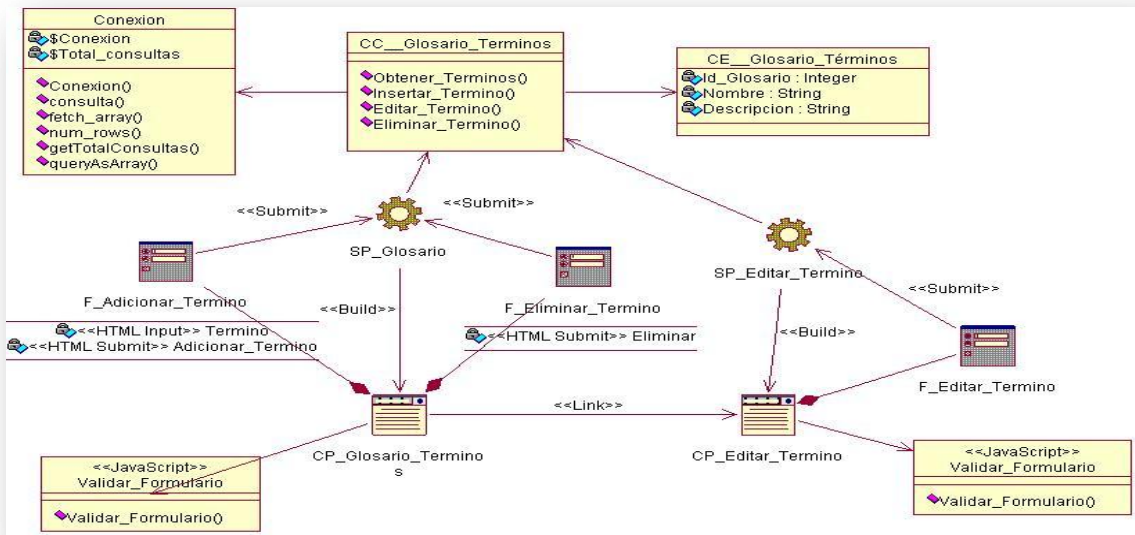


Fig. 3.12 Diagrama de Clases del Diseño “Gestionar Glosario Términos.”

Diagrama de secuencia del caso de uso “Gestionar Glosario Términos”. [Ver Anexo 11]

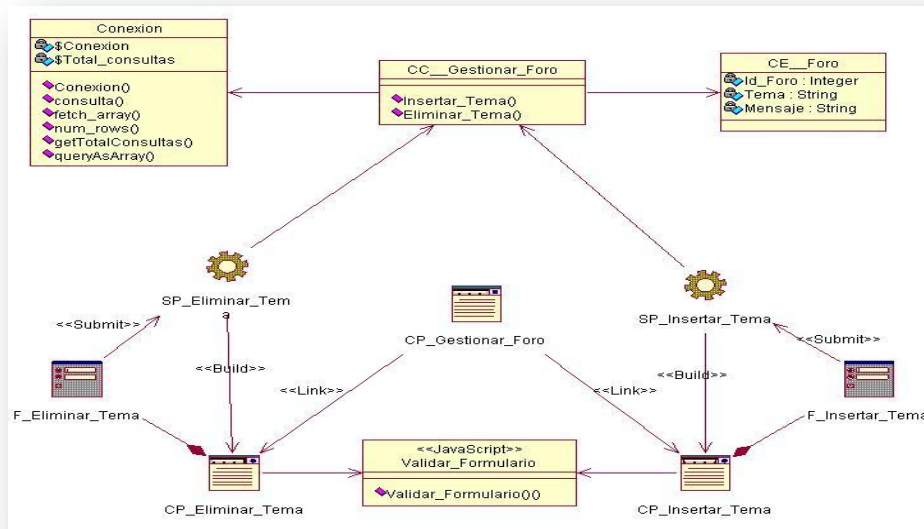


Fig. 3.13 Diagrama de Clases del Diseño “Gestionar Foro.”

Diagrama de secuencia del caso de uso “Gestionar Foro”. [Ver Anexo 12]

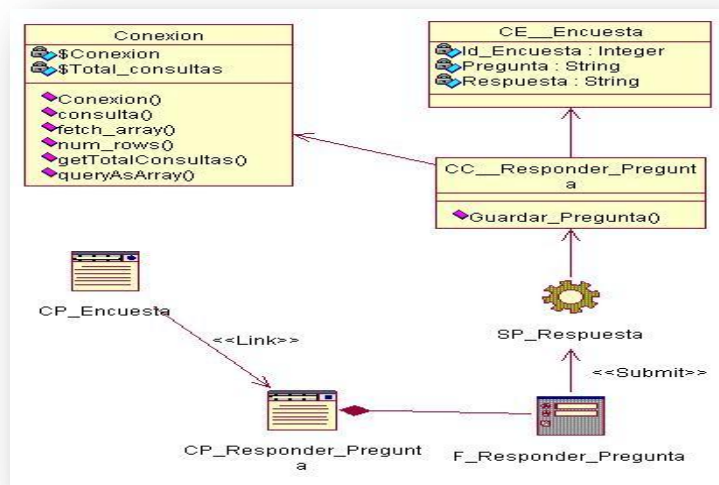


Fig. 3.14 Diagrama de Clases del Diseño “Responder Encuesta.”

Diagrama de secuencia del caso de uso “Responder Encuesta”. [Ver Anexo 13]

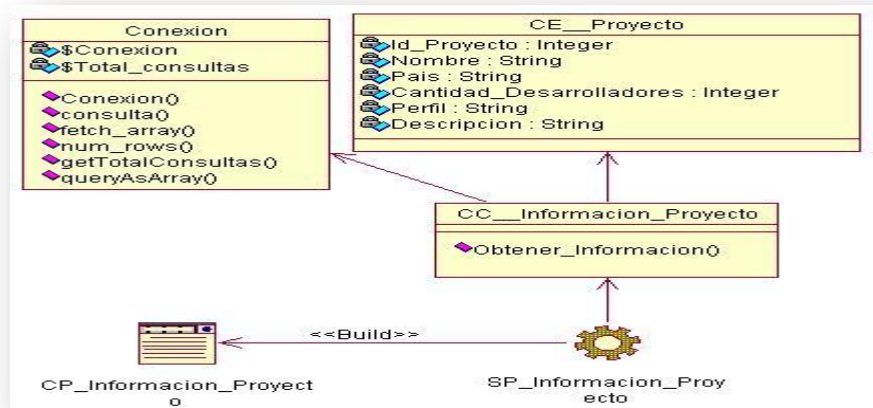


Fig. 3.15 Diagrama de Clases del Diseño “Mostrar Información Proyecto.”

Diagrama de secuencia del caso de uso “Mostrar Información Proyecto”. [Ver Anexo 14]

Subsistema 3. Administración.

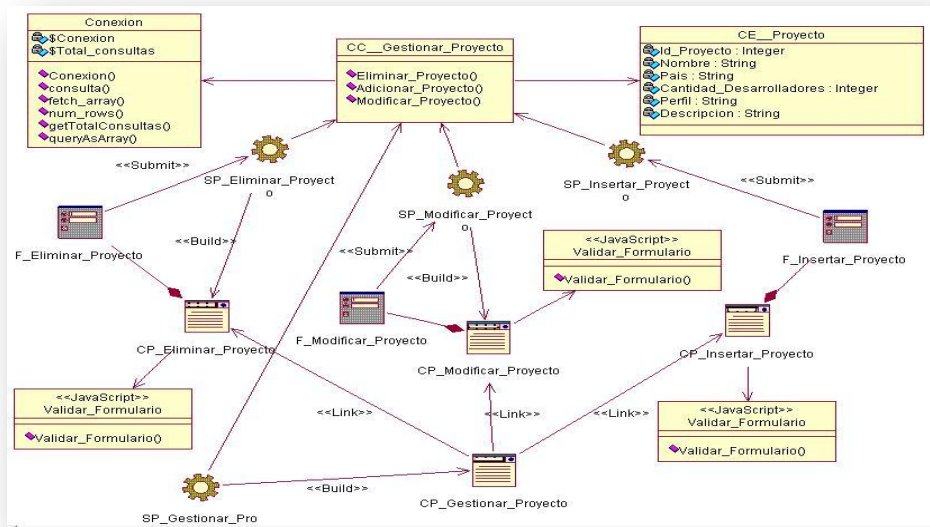


Fig. 3.16 Diagrama de Clases del Diseño “Gestionar Proyecto.”

Diagrama de secuencia del caso de uso “Gestionar Proyecto”. [Ver Anexo 15]

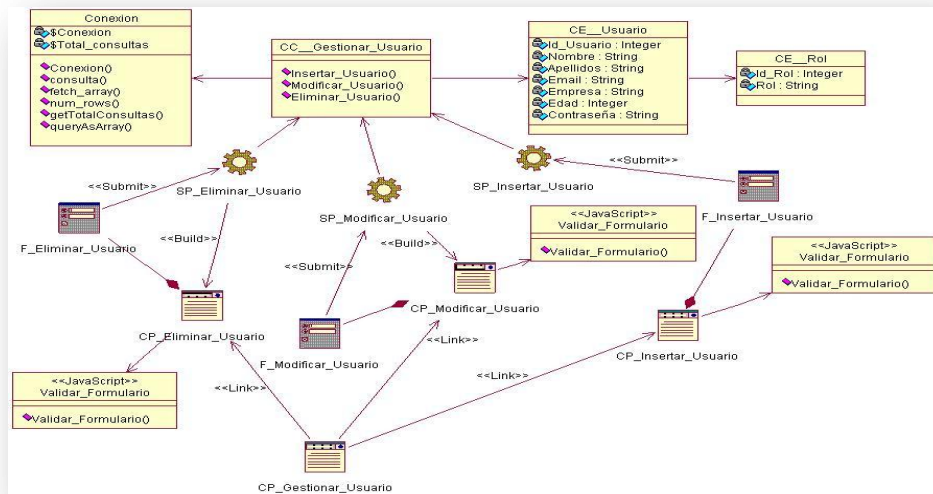


Fig. 3.17 Diagrama de Clases del Diseño “Gestionar Usuario.”

Diagrama de secuencia del caso de uso “Gestionar Usuario”. [Ver Anexo 16]

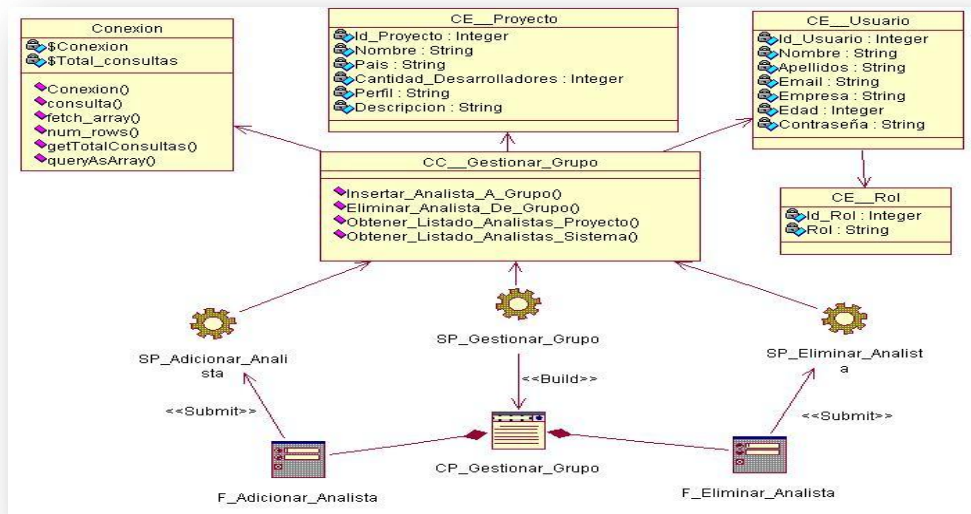


Fig. 3.18 Diagrama de Clases del Diseño “Gestionar Grupo Analistas.”

Diagrama de secuencia del caso de uso “Gestionar Grupo Analistas”. [Ver Anexo 17]

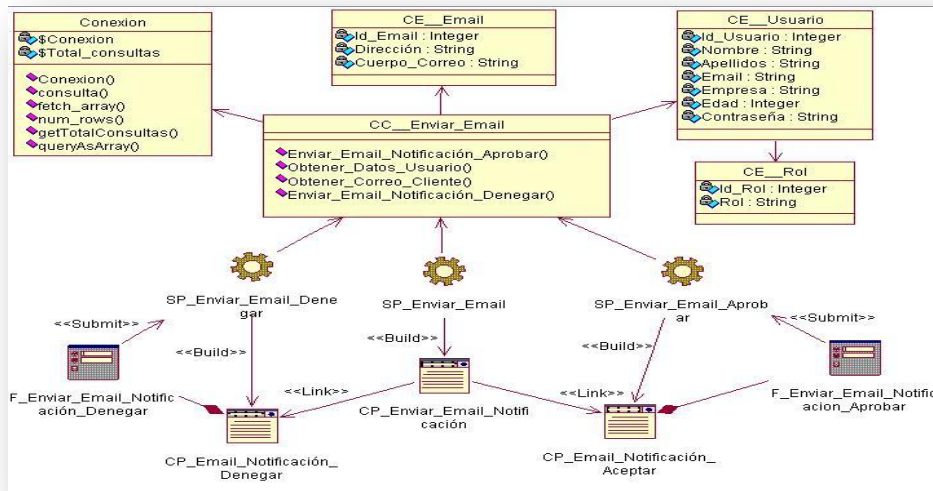


Fig. 3.19 Diagrama de Clases del Diseño “Enviar Email Notificación.”

Diagrama de secuencia del caso de uso “Enviar Email Notificación”. [Ver Anexo 18]

Subsistema 4. Propuesta.

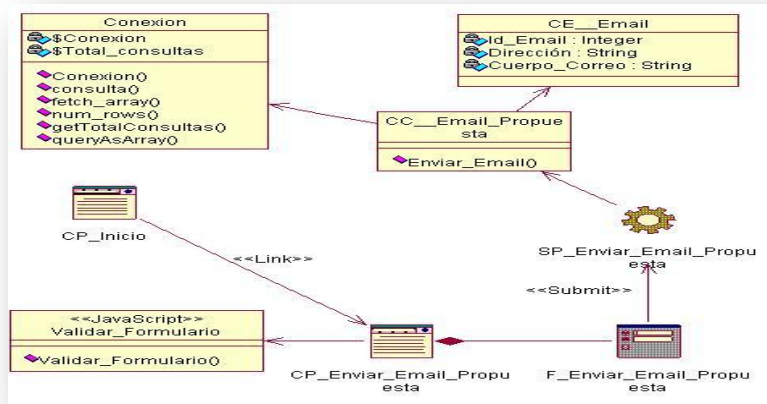


Fig. 3.20 Diagrama de Clases del Diseño “Enviar Email Propuesta.”

Diagrama de secuencia del caso de uso “Enviar Email Propuesta”. [Ver Anexo 19]

3.5. Diseño de la BD.

3.5.1. Diagrama de clases persistentes

El diagrama de clases persistentes permite modelar aspectos relacionados con el almacenado de datos del sistema. De forma general, las clases persistentes y sus atributos tienen correlación con las entidades lógicas y sus atributos. La figura muestra el diagrama de clases persistentes del sistema que se propone.

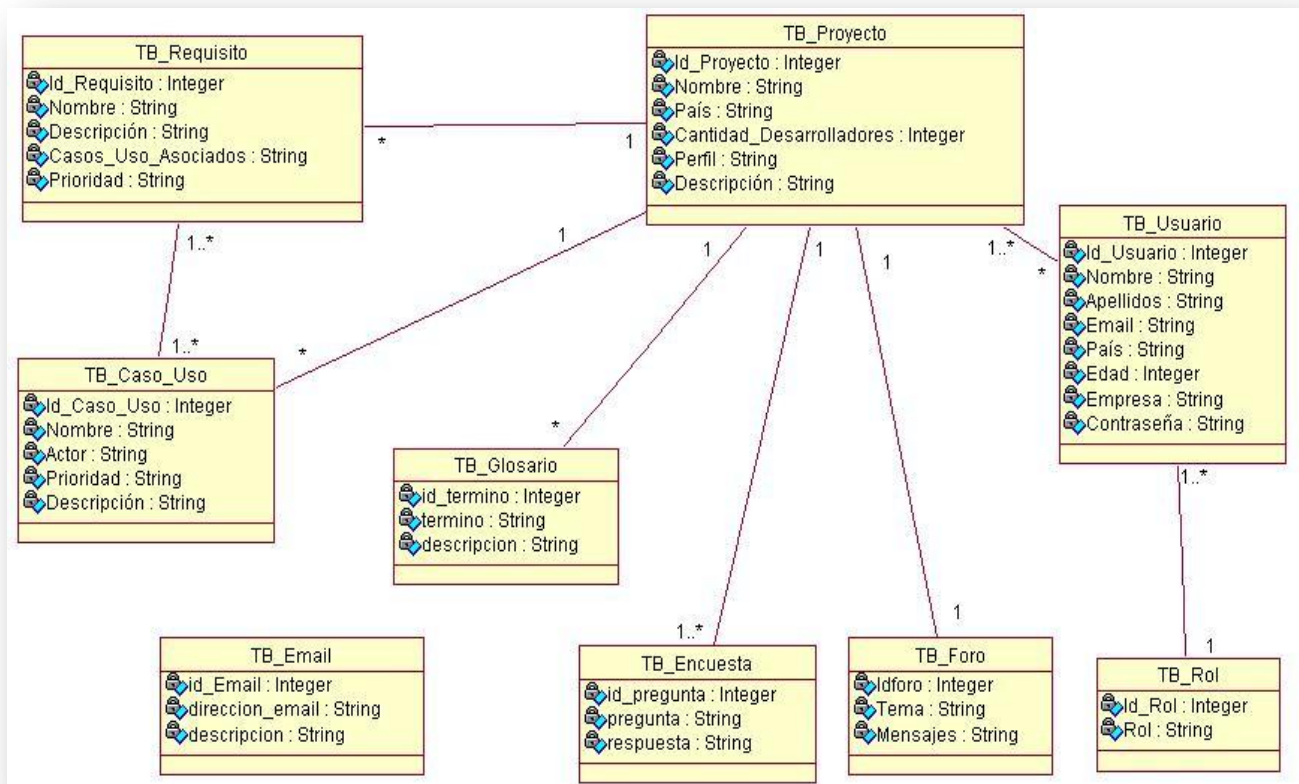


Fig. 3.21 Diagrama de clases persistentes.

3.5.2. Modelo de Datos

El modelo de datos es un mecanismo muy útil cuando se desea implementar un nivel de abstracción adecuado sobre el almacenamiento físico de información que la mayoría de usuarios no necesita conocer. En pocas palabras un modelo de datos es un conjunto de conceptos que permiten describir la estructura de una base de datos. Estos conceptos no son otros sino: los datos, las relaciones entre ellos y las restricciones que deben cumplirse sobre los mismos.

El modelo de datos puede ser teniendo en cuenta los tipos de conceptos que brindan para la descripción de la estructura de la base de datos. Los modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios perciben los datos, mientras que los modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en la computadora, por lo que están dirigidos al personal informático, no a los usuarios finales. La figura muestra el modelo físico de la base de datos.

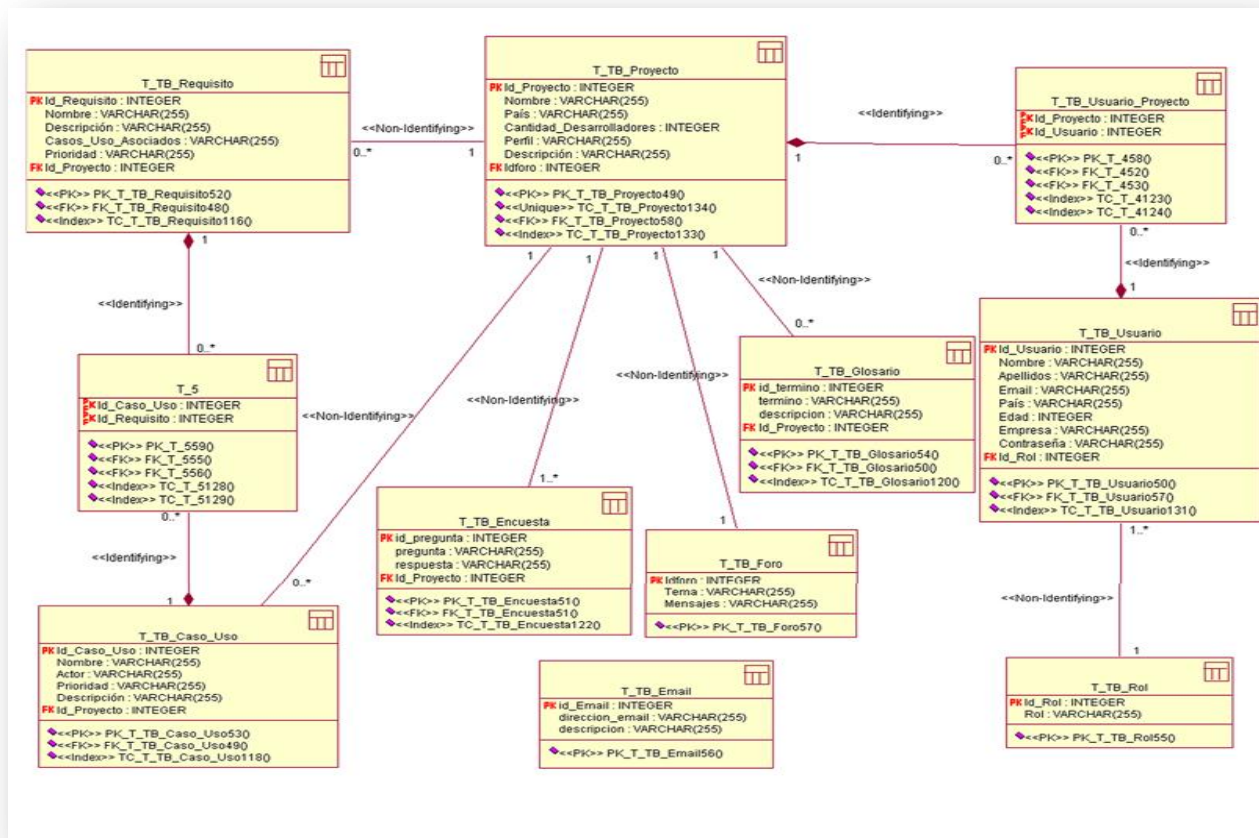


Fig.3.22 Modelo de Datos.

3.6 Seguridad

Debido a la importancia de los datos que se manejarán en la aplicación es necesario garantizar la protección y seguridad de la información para un mejor funcionamiento del sistema, con el objetivo de conseguir este propósito se utilizó el mecanismo de diseño de Seguridad. El sistema realiza el control de la seguridad utilizando un servicio Web encargado del control de los accesos, autenticación de los usuarios en el sistema. Para garantizar que determinada información presente en la aplicación sólo se muestre a los usuarios registrados y con autorización previa, se trabaja con variables de sesión, de forma

tal que siempre se sabe qué usuario intenta visitar dichas páginas y estas se muestran sólo para aquellos que pueden tener acceso a ellas.

3.7 Principios de diseño.

El diseño, sea cual sea el objeto del mismo, tiene que basarse en el usuario, debido a que, generalmente muchos de ellos no presentan una preparación en las cuestiones de la informática. Para ello, cada sistema debe utilizar ciertos principios generales que garanticen la usabilidad en los diseños para aplicaciones Web.

3.7.1 Estándares en la interfaz de la aplicación.

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Es por ello que para lograr un diseño consistente de la interfaz de la aplicación se utiliza en todas las páginas el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en la parte izquierda superior a modo de logo. En el navegador se incluyen los enlaces, en forma de pestañas, a las distintas secciones. En el área del contenido se muestran los formularios de entrada, las informaciones que se deben mostrar, los reportes, etc.

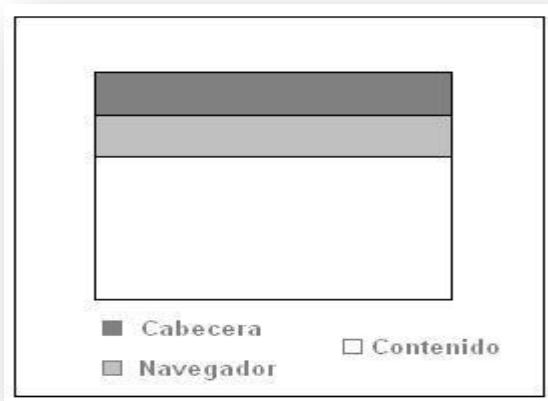


Fig. 3.23 Esquema de página.

Se utilizan para el diseño las tablas y plantillas, dado que son cien por ciento compatibles con todos los navegadores, también se emplea hojas de estilos para guardar la configuración del diseño de todas las páginas. Estas hojas de estilos establece el tipo, tamaño de fuente de los distintos elementos de cada página, el color de los vínculos, el color de fondo, el formato de los controles de formulario y las tablas, entre otros. En la aplicación se utilizó el color azul en tonalidades claras y oscuras, porque es un color neutro y pasivo, sugiere seriedad y está muy asociado a las aplicaciones informáticas.

3.7.2 Formato de los reportes.

Los reportes se muestran en formato de documento .pdf. Exponen la información organizada en un “Documento de Requerimientos”, que incluye presentación, índice y disponen los requerimientos, casos de uso y glosario de términos con sus datos asociados, en forma de tablas, teniendo en cuenta el proyecto que se seleccione.

[Ver Anexo 20].

3.7.3 Tratamiento de excepciones.

Para reducir la ejecución de errores se utilizó javascript, evitando así que el servidor Web procese la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones.

También, se utilizan errores en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

[Ver Anexo 21].

3.7.4 Estándares de codificación.

Para lograr un mejor entendimiento del código en la implementación de la aplicación es necesario establecer un estándar de codificación a usar. En la política seguida al respecto todas las variables y nombres de funciones a utilizar se definieron en idioma español. Los inicios ({} y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen. Se usa una línea propia para {. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Los nombres de las variables utilizadas comienzan en minúscula y son cortos, claros y describen su propósito. Los comentarios se definen comenzando con los caracteres /* y terminando con */ para los comentarios de varias líneas, y comenzando con los caracteres // para los de una sola línea.

3.8 Diagrama de despliegue.

El modelo de despliegue representa un modelo de objetos que muestra cómo se distribuye físicamente el sistema, teniendo en cuenta la funcionalidad entre cada nodo de cómputo. Este modelo es utilizado como base para la realización de las actividades de diseño e implementación.

Se puede observar lo siguiente sobre el modelo de despliegue:

- Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar.
- Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como Internet, Intranet, bus y similares.
- El modelo de despliegue puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación.

- La funcionalidad (los procesos) de un nodo se define por los componentes que se distribuyen sobre ese nodo.
- El modelo de despliegue en sí mismo representa una correspondencia entre la arquitectura de software y la arquitectura del sistema (hardware).

La figura muestra el diagrama de despliegue del sistema que se propone.

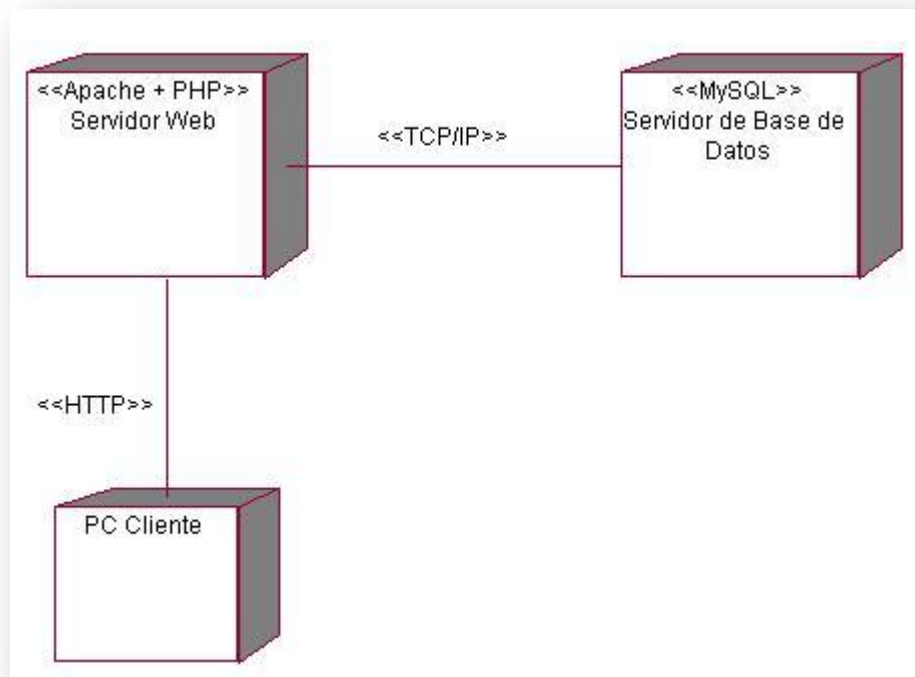


Fig.3.23 Modelo de Despliegue.

CONCLUSIONES

A partir de la investigación realizada para la elaboración de este sistema utilizando el Proceso Unificado de Desarrollo de Software (RUP) como metodología idónea para lograr una mejor comprensión de los requisitos de la aplicación y formalización de los mismos, se arriba a las siguientes conclusiones:

- A través del estudio que se llevó a cabo, se detectaron aspectos que incidían en el proceso de desarrollo de software en los proyectos productivos de la UCI, en el momento de realizar la captura de requisitos.
- Como respuesta al problema originado, se llegó a la propuesta de realizar una aplicación web para dar soporte al proceso de captura de requisitos en el desarrollo de software a distancia para los proyectos productivos de la universidad, facilitando la comunicación entre clientes y desarrolladores.
- El sistema desarrollado incluye funcionalidades que, independientemente de permitir la captura de los requisitos y determinación de los casos de uso, generan un intercambio favorable entre las personas involucradas para poder alcanzar la información más cercana a la realidad.
- La utilización de esta aplicación contribuye al fortalecimiento del proceso de captura de requisitos y su puesta en práctica significará una mejora considerable, pues ante el pedido de realización de proyectos por parte de clientes de diversos territorios, no habrá ningún inconveniente, y se maximizarán las posibilidades de alcanzar software de calidad.

Con la propuesta y el estudio realizado se materializó el objetivo general planteado al inicio de esta investigación. Además, se corroboró la idea de que la implementación de esta herramienta facilitaría el proceso de levantamiento de requisitos de los analistas que se encuentran integrados a proyectos productivos en la UCI, que realizan software a distancia.

RECOMENDACIONES

- Continuar el estudio con el objetivo de encontrar nuevas funcionalidades para refinar e implementar una herramienta más completa y general.
- Implementar el foro diseñado para la interacción cliente-analista.
- Incorporar reportes de métricas que permitan poder hacer un estudio posterior del estado de los proyectos.
- Integrar una herramienta case a la aplicación, para poder obtener el diagrama de Casos de Uso del Sistema.
- Realizar la migración de la aplicación hacia el framework Ext 3.0 y lograr así una interfaz más intuitiva para la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

[1] Cerf, S. Introducción al offshore: algunos datos. [En línea] 28/03/2005

Disponible en: <http://www.baquia.com/noticias.php?id=9457>

[2] **Stephane, Cerf.** ¿Qué es la informática offshore? [En línea] 28 de noviembre del 2006.

Disponible en: http://www.idg.es/cio/_Que_es_la_informatica_offshore_/art180016-management.htm

[3] **Christel, M. G., Kang, K. C.** "Issues in Requirements Elicitation". Technical Report CMU/SEI-92-TR-12 (ESC-TR-92-012). Software Engineering Institute. Carnegie Mellon University, Pittsburgh, PA 15213 (USA). September 1992.

Disponible en: <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr12.92.pdf>

[4] **Durán Toro, A.** "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información". Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla. Septiembre de 2000

Disponible en: <http://www.lsi.us.es/~amador/publicaciones/tesis.pdf.zip>

[5] **IEEE.** IEEE Recommended Practice for Software Requirements Specifications. [En línea] 1998

Disponible en http://arantxa.ii.uam.es/~sacuna/is1/normas/IEEE_Std_830_1998.pdf

[6] **Escalona, M.E., Koch N.** Ingeniería de Requisitos en Aplicaciones para la Web Un estudio comparativo. [En línea] 2002.

Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>

[7] **Apache.org.** [En línea] 2005.

Disponible en: <http://www.facilnet.net/matriz/web2/apache.html>

[8] **Una introducción a apache.**

Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/

[9] **PEREZ, J. M.** ¿Qué es MYSQL? [En línea] 10 de octubre de 2007.

Disponible en: <http://www.espestudio.com/articulo/desarrollo-web/base-de-datos-mysql/Que-es-MySQL.htm>

[10] ¿Qué es MySQL? Historia de MySQL Características principales.[En línea] 4 de noviembre del 2007.

Disponible en: <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>

[11] **E, Quiñones, A.** Introducción a PostgreSQL.[En línea] 2 de noviembre de 2007.

Disponible en: http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf

[12] **Programación en castellano.** HTML en castellano.

Disponible en: <http://www.programacion.com/html/>

[13] **Guillermo Prado Ajona** .Universidad de la Rioja, Página creada para la asignatura de Sistemas Informáticos.

Disponible en: <https://belenus.unirioja.es/~guprado/pagweb/carachtml.html>

[14] **Ilario Valdelli.** [En línea] 2006.

Disponible en: http://www.htmlpoint.com/javascript/corso/js_02.htm

[15] **Gassmann A.** Gamarod Condigos Java Scripts. [En línea] 20 de abril del 2003.

Disponible en: http://www.gamarod.com.ar/articulos/introduccion_a_aspnet.asp

[16] **Rodas, R.** Usuarios y Grupos en Linux.

Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

[17] **Pérez, A.F.** Universidad de Costa Rica, Escuela de Ciencias de la Computación e Informática San José, Costa Rica. [En línea] 2007.

Disponible en: <http://www.di-mare.com/adolfo/cursos/2007-1/pp-Perl.pdf>

[18] **Barnet, A.** Corporación la prensa. [En línea] 2006.

Disponible en: <http://www.prensa.com/Actualidad/Tecnologia/2006/06/10/index.htm>

BIBLIOGRAFÍA

Agencia Tributaria. 2006. Captura de Requisitos en el desarrollo de Proyectos Informáticos. [En línea] 3 de Marzo de 2006. <http://www.actica.org/actica/colabora/capturarequisitos.pdf>.

FERNANDEZ, S. P. 2007. Comparativa Servidores Web. [En línea] 2007. <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=363>.

GILFILLAN, I. 2006. La Biblia MySQL. [En línea] 30 de 11 de 2006. <http://bibliodoc.uci.cu/pdf/reg02847.pdf>.

HAREM. 2007. Lighttpd Vs. Apache, PhpMagazine. [En línea] 2007. http://www.phpmagazine.net/2007/02/lighttpd_vs_apache.html.

JACOBSON, I y BOOCH, G. *El proceso unificado de desarrollo de software*. s.l. : Addison-Wesley. 84-7829-036-2.

MySQL, MySQL AB. 2007. The world's most popular open source database. [En línea] September de 2007. <http://dev.mysql.com/doc/refman/5.0/es/index.html>.

Pazos Arias, José Juan. 2000. Ingeniería de Requisitos. [En línea] 02 de 2000. <http://tvdi.det.uvigo.es/~jose/doctorado/re/>.

Pressman, R., *Ingeniería de Software. Un enfoque práctico*, McGraw-Hill, 2000.

S.B.A LTD AR. 2003. cuentabancariasuiza. [En línea] 2003. http://cuentabancariasuiza.com.ar/que_es_offshore.htm.

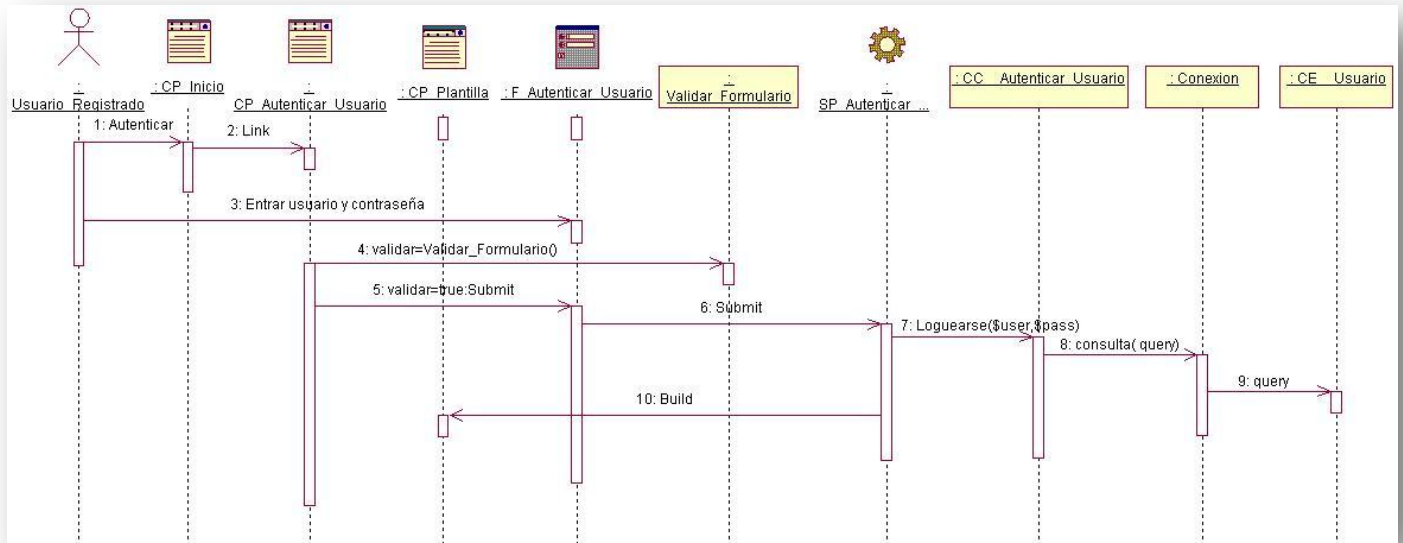
SCHWARTZ, JONATHAN. 2008. Ya es oficial: MySQL forma parte de Sun. [En línea] 2008. http://blogs.sun.com/jonathan_es/.

Sun Microsystem. Why MySQL? [En línea] <http://www.mysql.com/why-mysql/>.

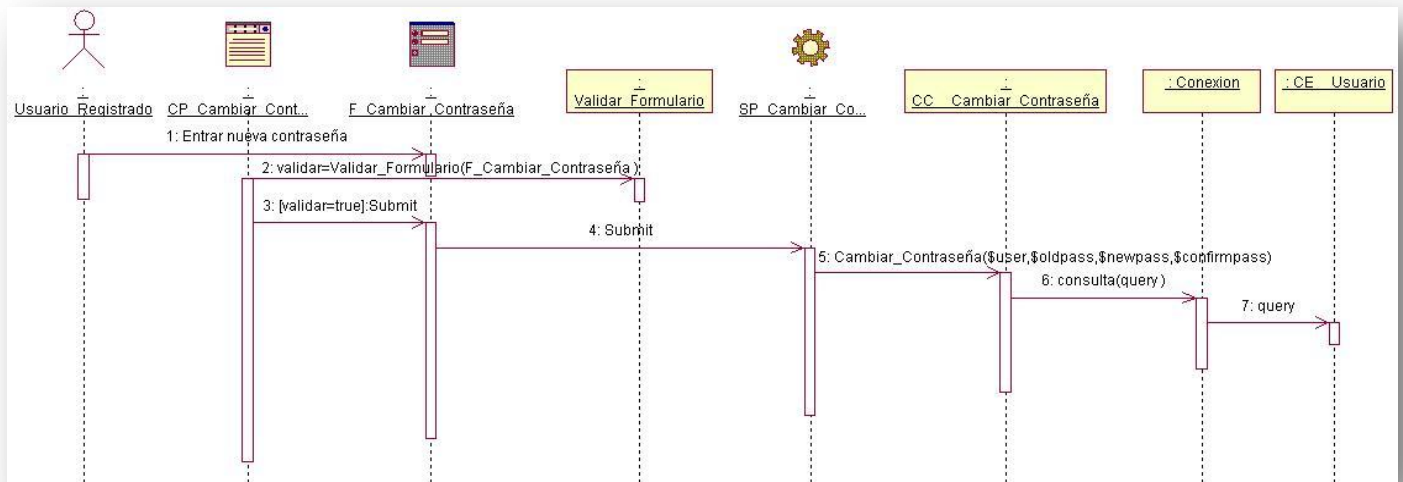
WELLING, L. y THOMSON, L. 2006. Desarrollo Web con PHP y MySQL. [En línea] 5 de 11 de 2006. <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.

ANEXOS

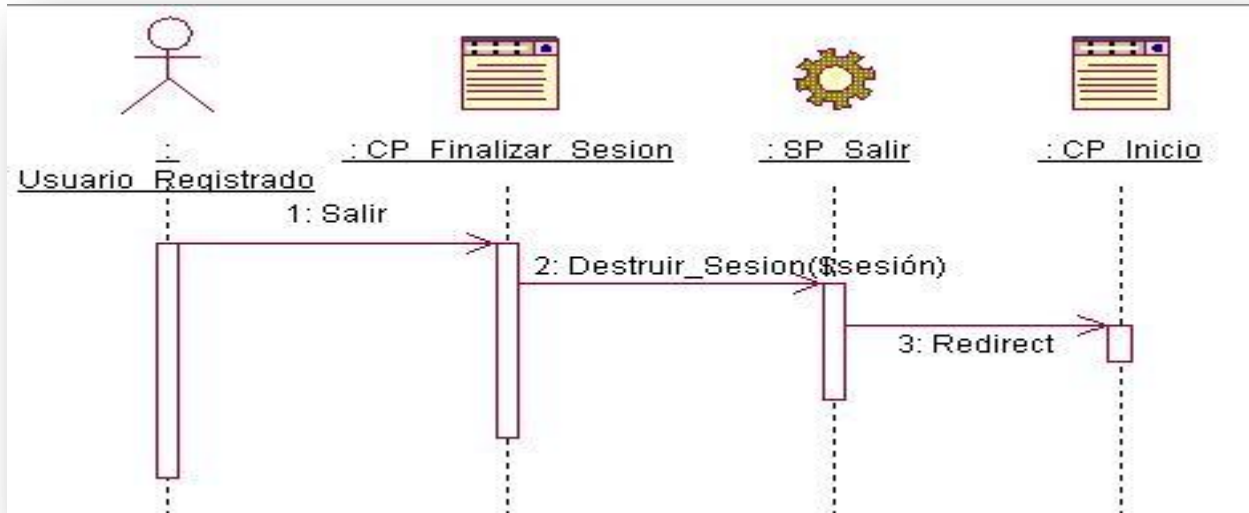
Anexo 1. Diagrama de secuencia con extensiones web del caso de uso “Autenticar Usuario”.



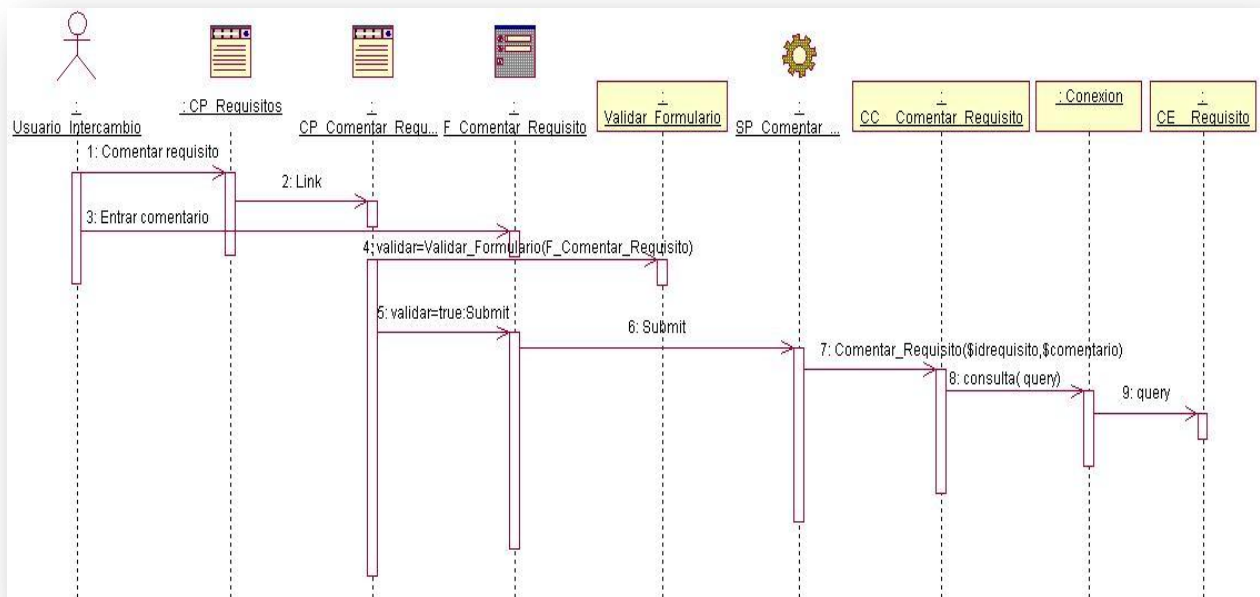
Anexo 2. Diagrama de secuencia con extensiones web del caso de uso “Cambiar Contraseña”.



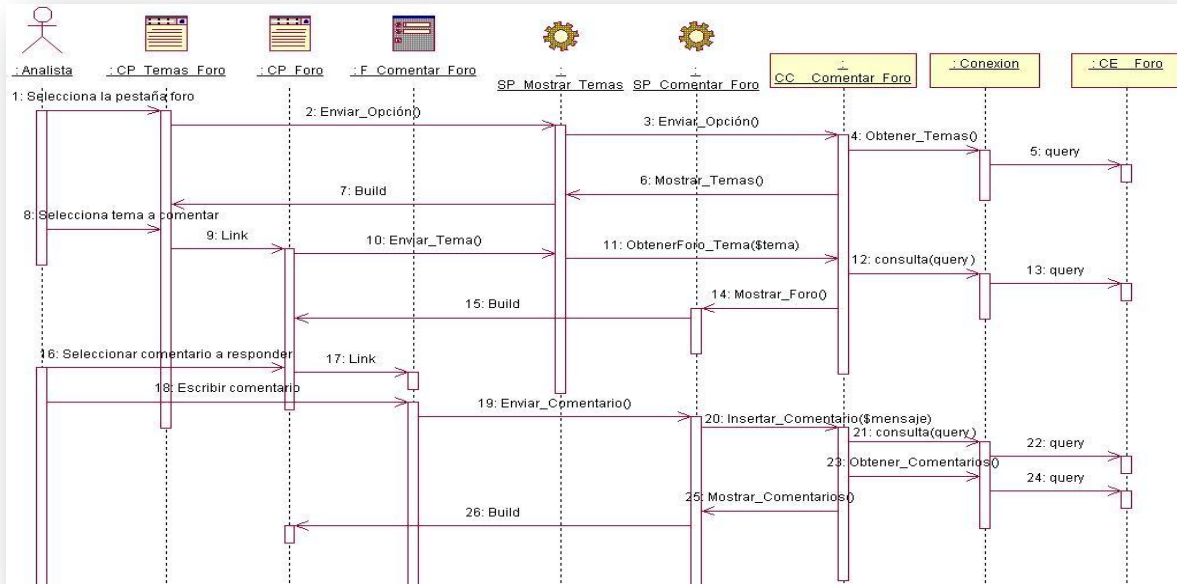
Anexo 3. Diagrama de secuencia del caso de uso “Finalizar Sesión Usuario”.



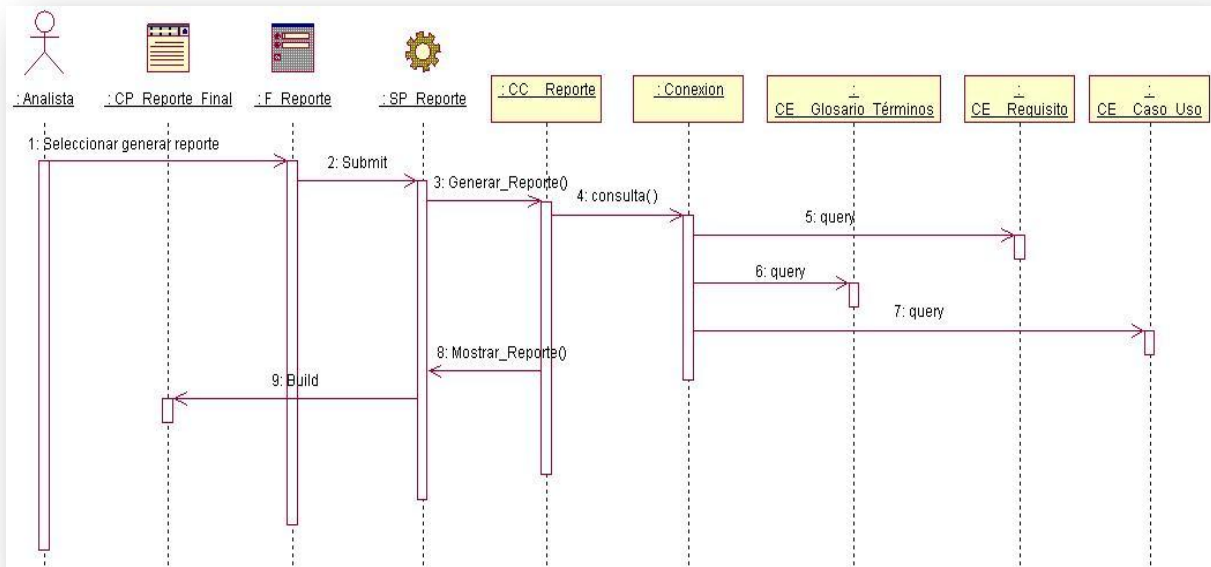
Anexo 4. Diagrama de secuencia del caso de uso “Comentar Requisito”.



Anexo 5. Diagrama de secuencia del caso de uso "Insertar Comentario Foro".



Anexo 6. Diagrama de secuencia del caso de uso "Generar Reporte".



Anexo 7. Diagrama de secuencia del caso de uso “Gestionar Encuesta”. Sección: Adicionar Pregunta.

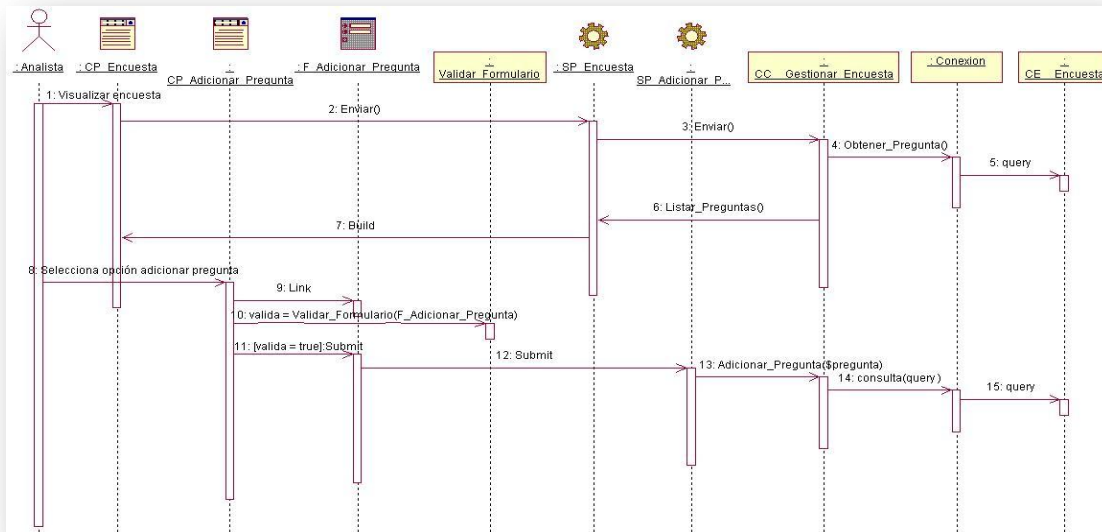
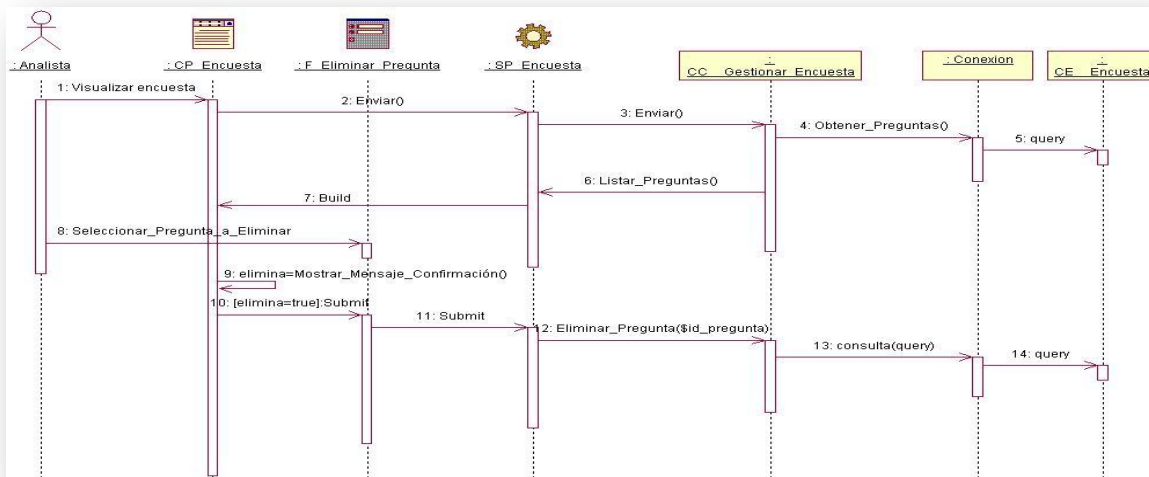
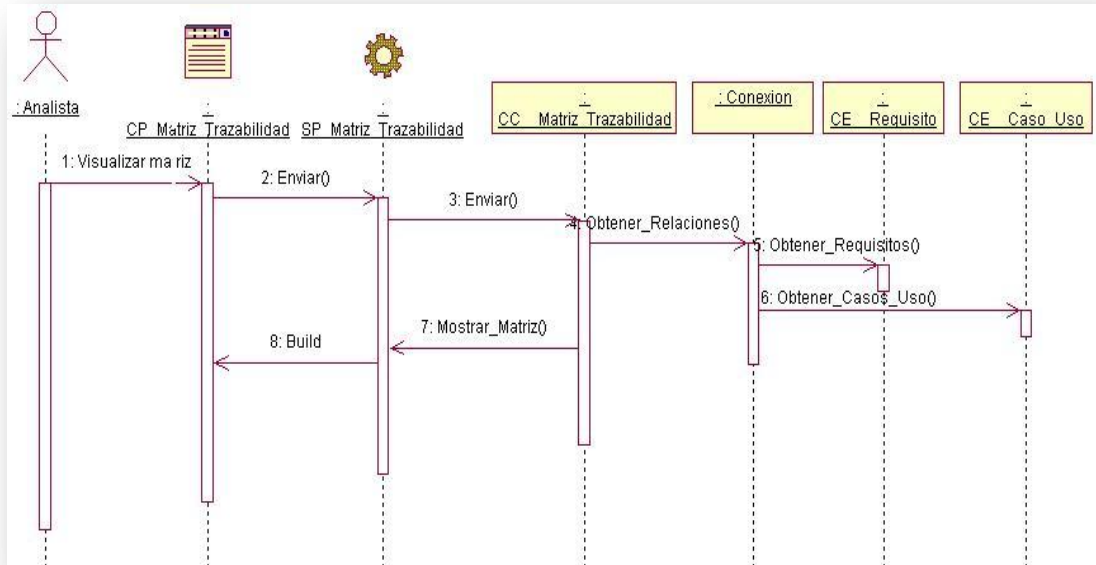


Diagrama de secuencia del caso de uso “Gestionar Encuesta”. Sección: Eliminar Pregunta.



Anexo 8. Diagrama de secuencia del caso de uso “Mostrar Matriz Trazabilidad”.



Anexo 9. Diagrama de secuencia del caso de uso “Gestionar Requisito”. Sección “Eliminar Requisito”.

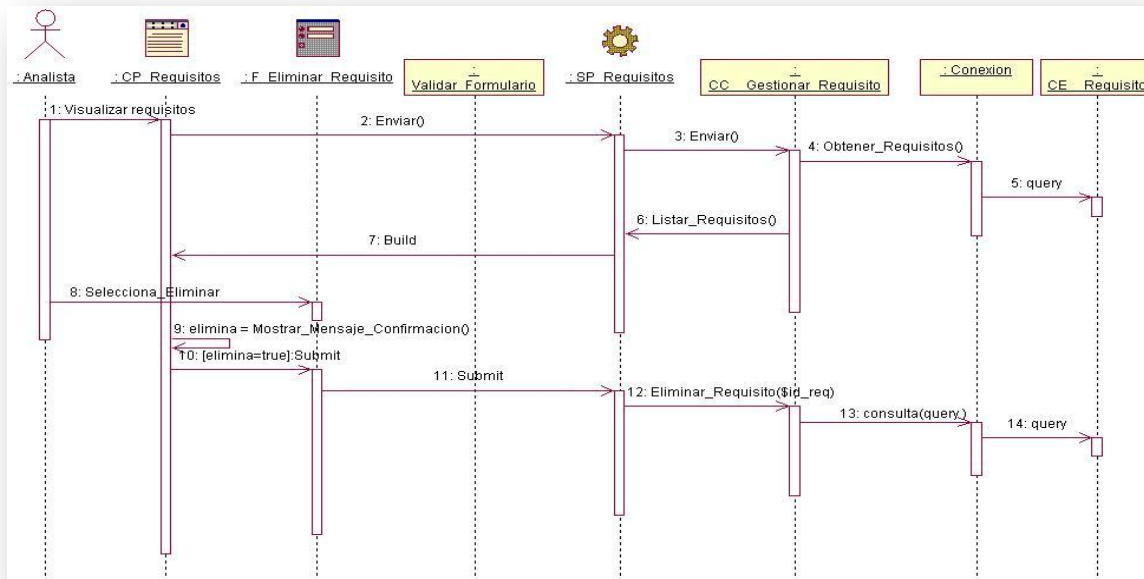


Diagrama de secuencia del caso de uso “Gestionar Requisito”. Sección “Adicionar Requisito”.

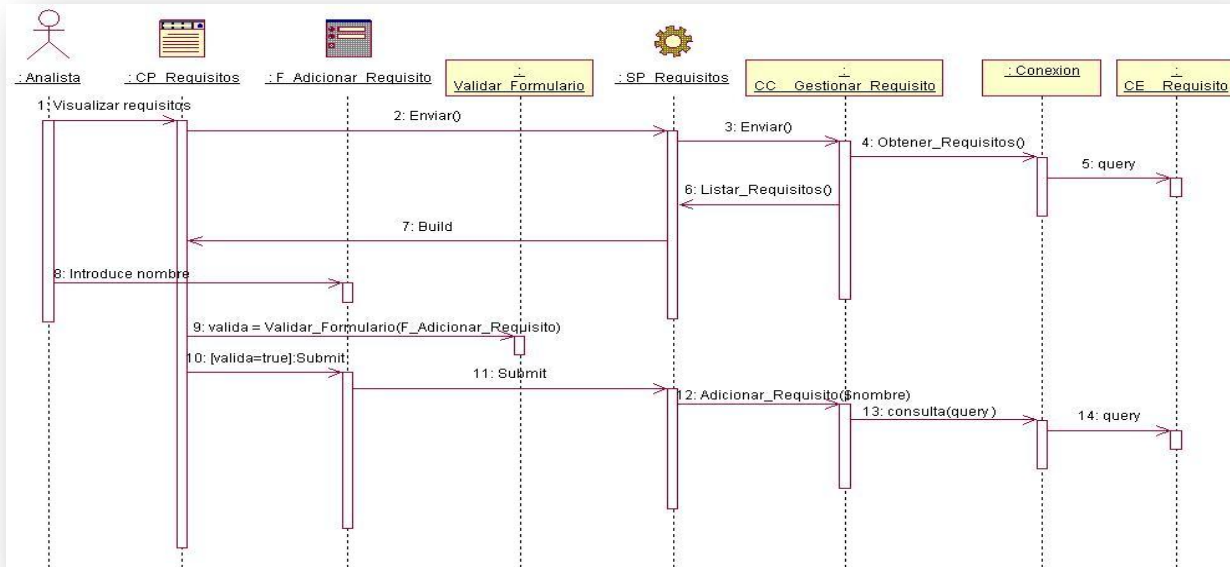
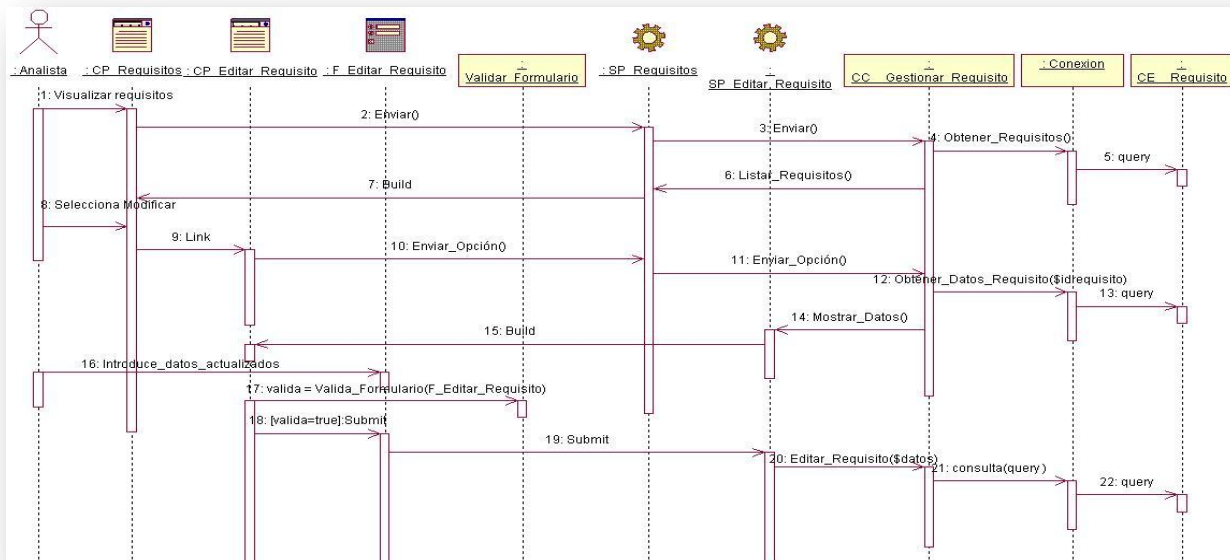


Diagrama de secuencia del caso de uso “Gestionar Requisito”. Sección “Modificar Requisito”.



Anexo 10. Diagrama de secuencia del caso de uso “Gestionar Caso Uso”. Sección “Eliminar Caso de Uso”.

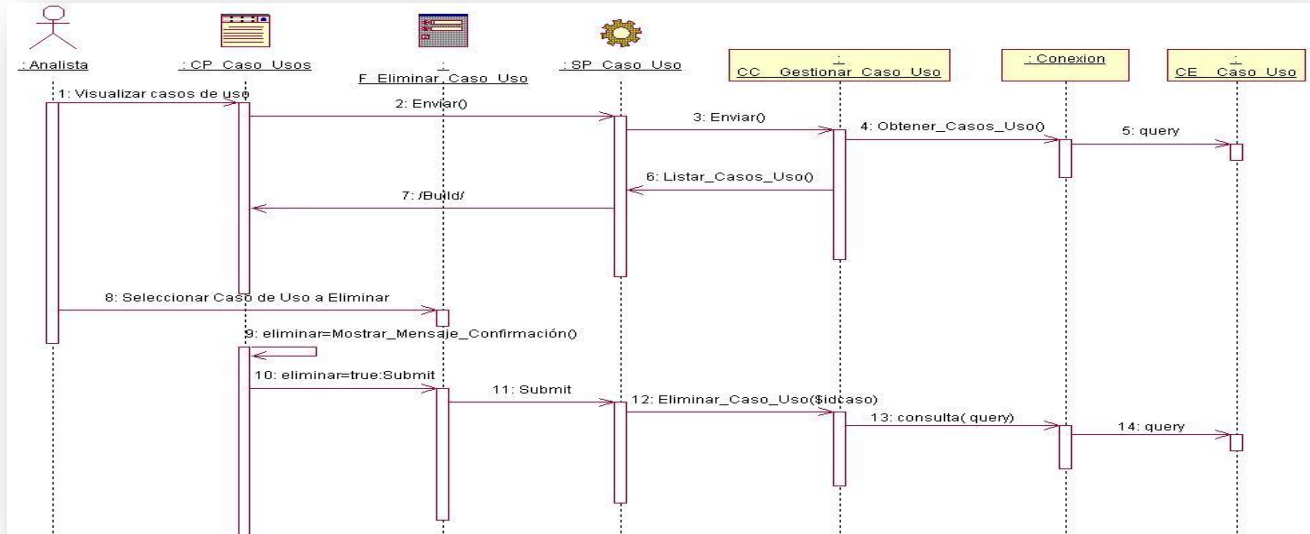


Diagrama de secuencia del caso de uso “Gestionar Caso Uso”. Sección “Insertar Caso de Uso”.

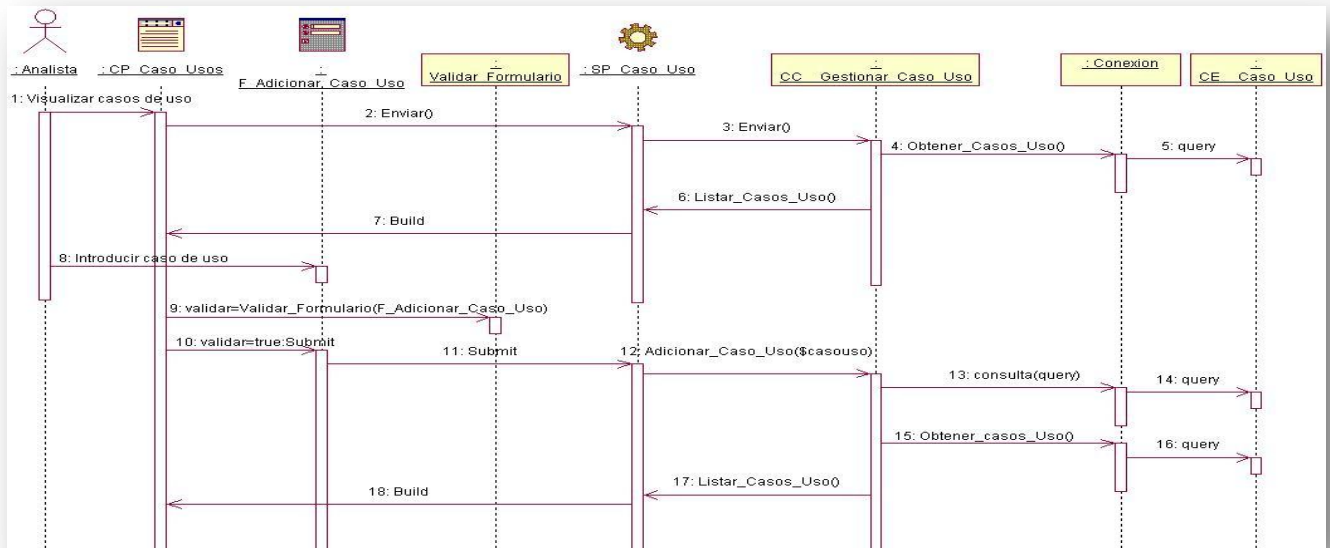
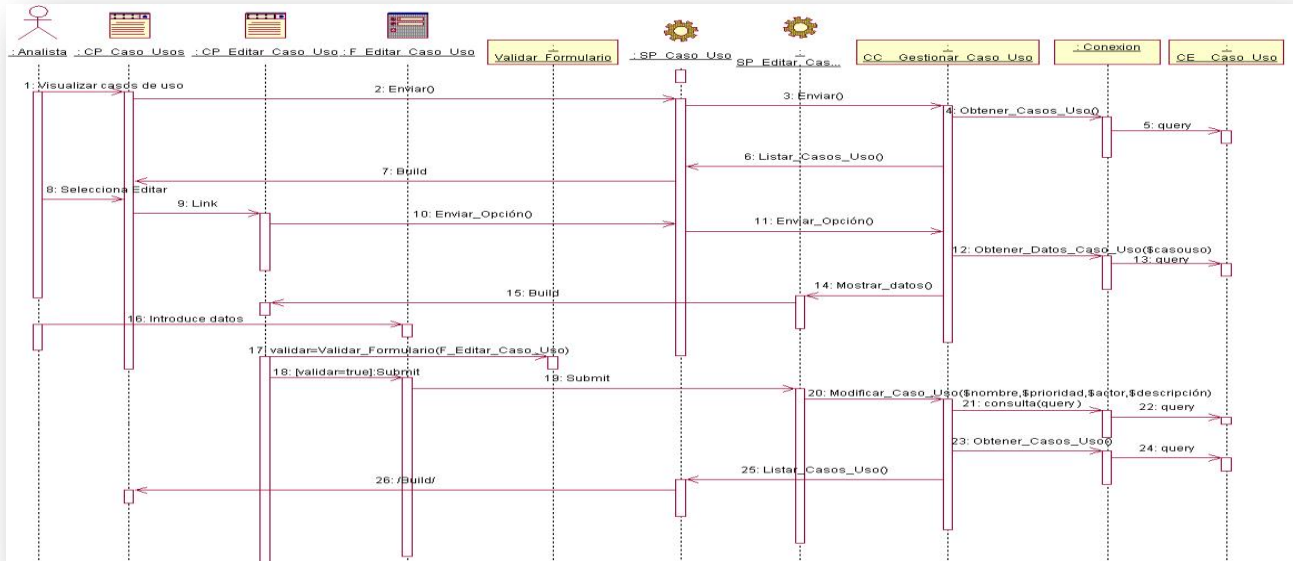


Diagrama de secuencia del caso de uso “Gestionar Caso Uso”. Sección “Modificar Caso de Uso”.



Anexo 11. Diagrama de secuencia del caso de uso “Gestionar Glosario Términos”. Sección “Eliminar Término”.

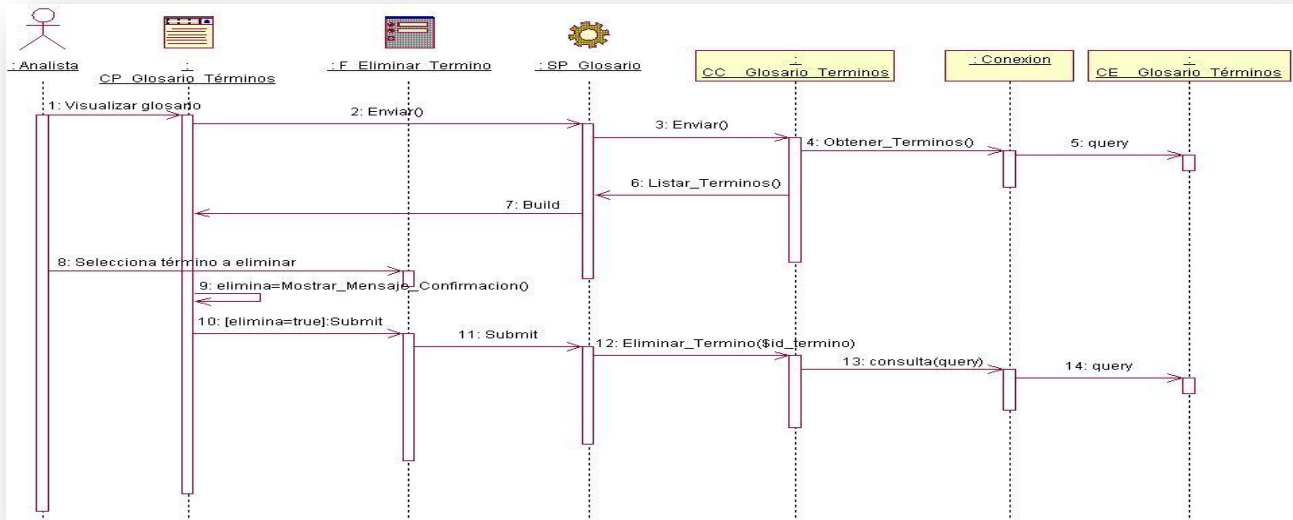


Diagrama de secuencia del caso de uso “Gestionar Glosario Términos”. Sección “Adicionar Término”.

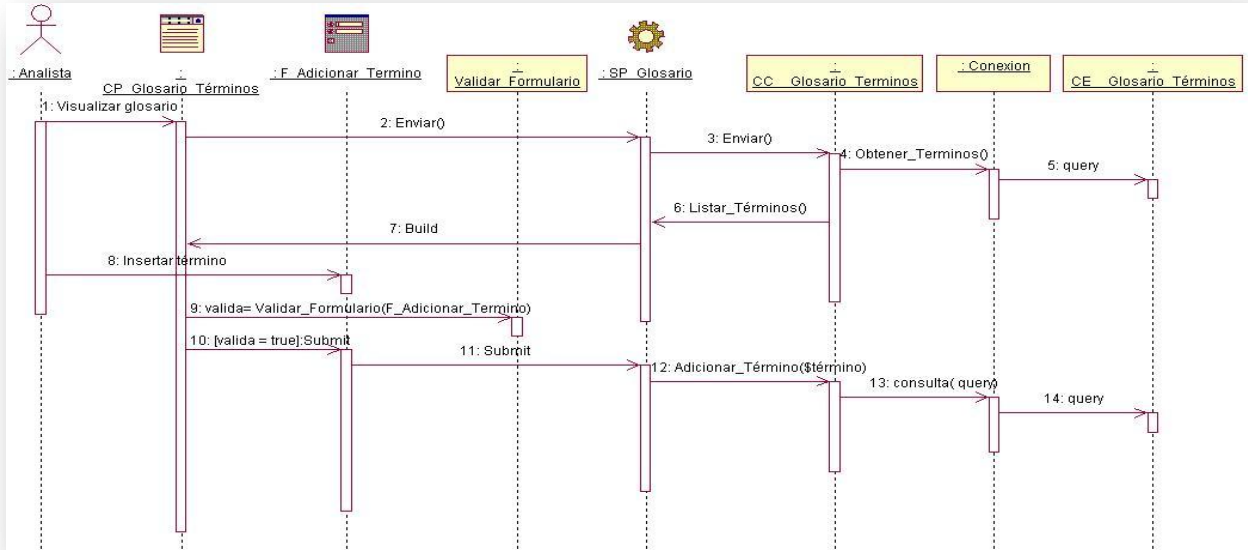
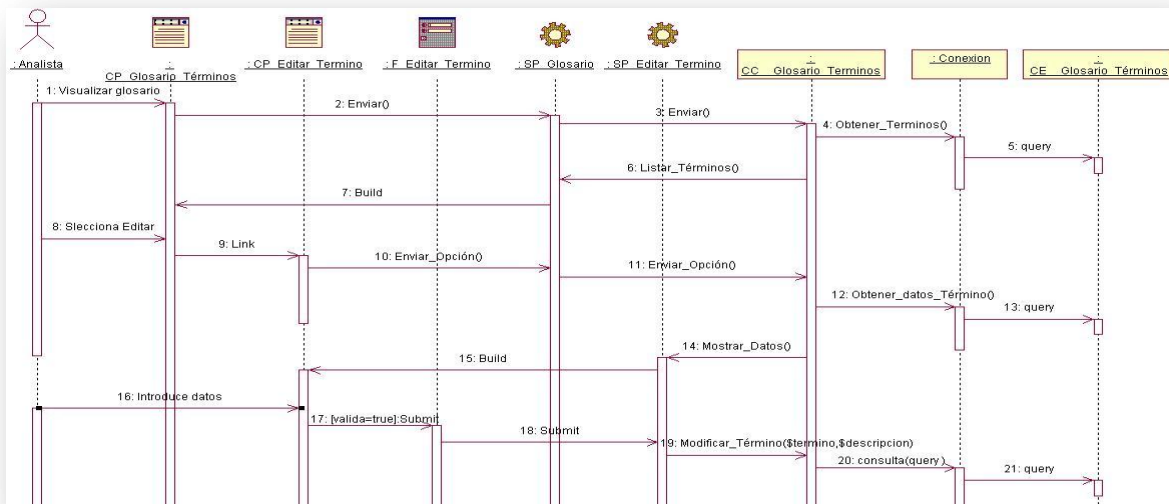


Diagrama de secuencia del caso de uso “Gestionar Glosario Términos”. Sección “Modificar Término”.



Anexo 12. Diagrama de secuencia del caso de uso “Gestionar Foro”. Sección “Insertar Tema”.

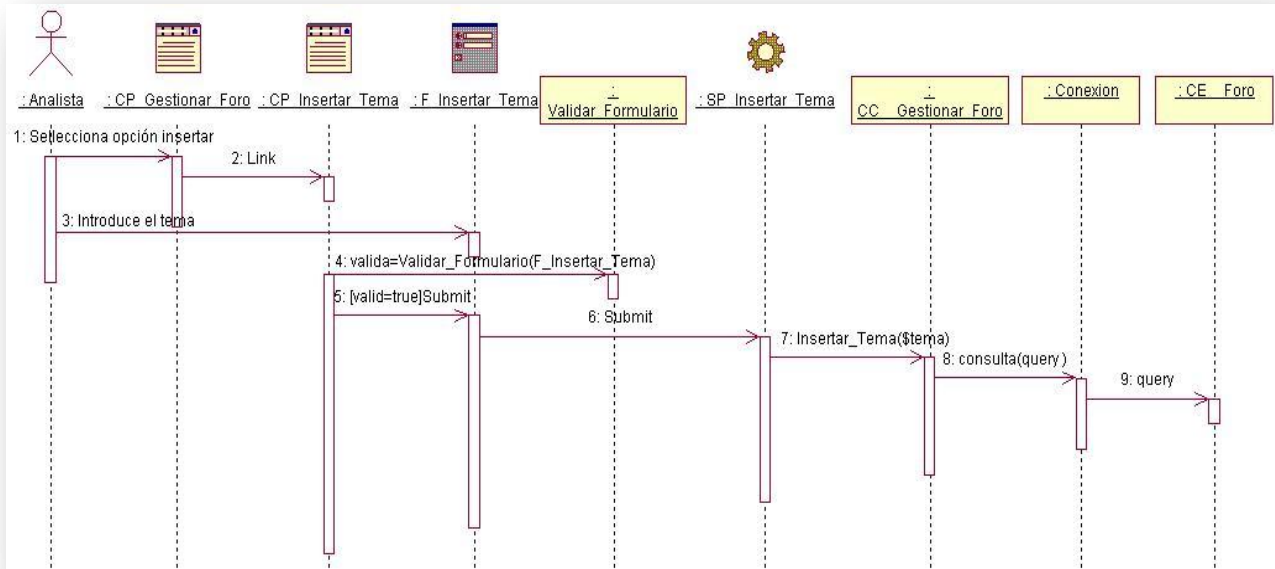
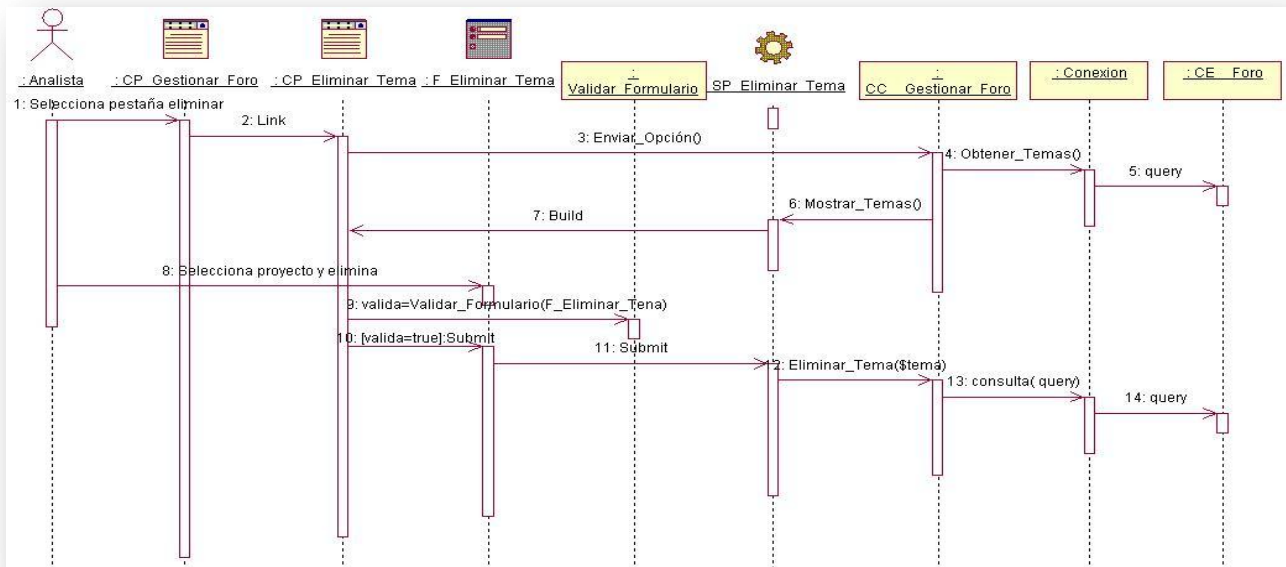
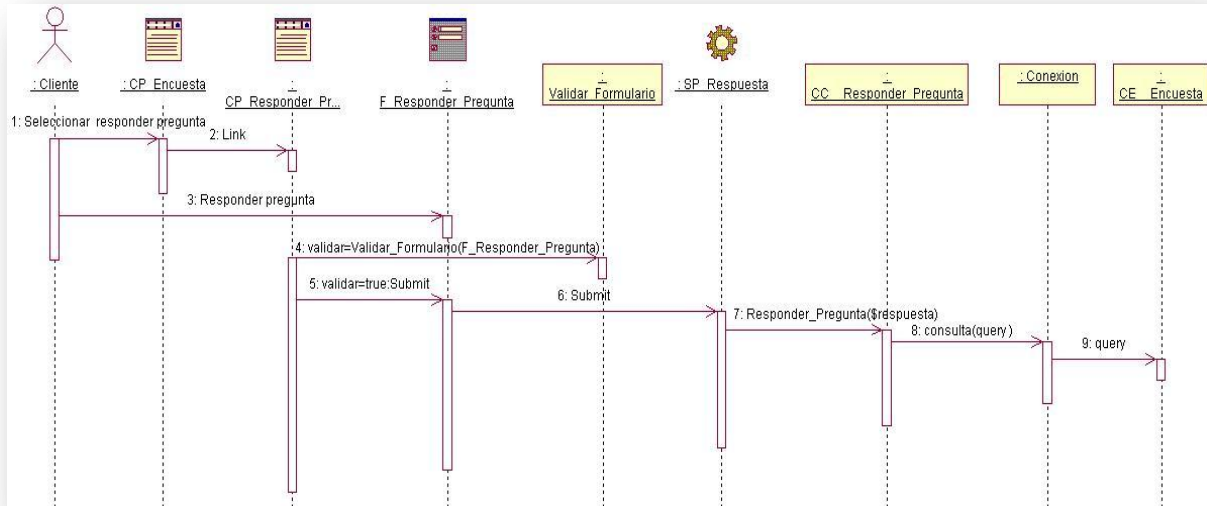


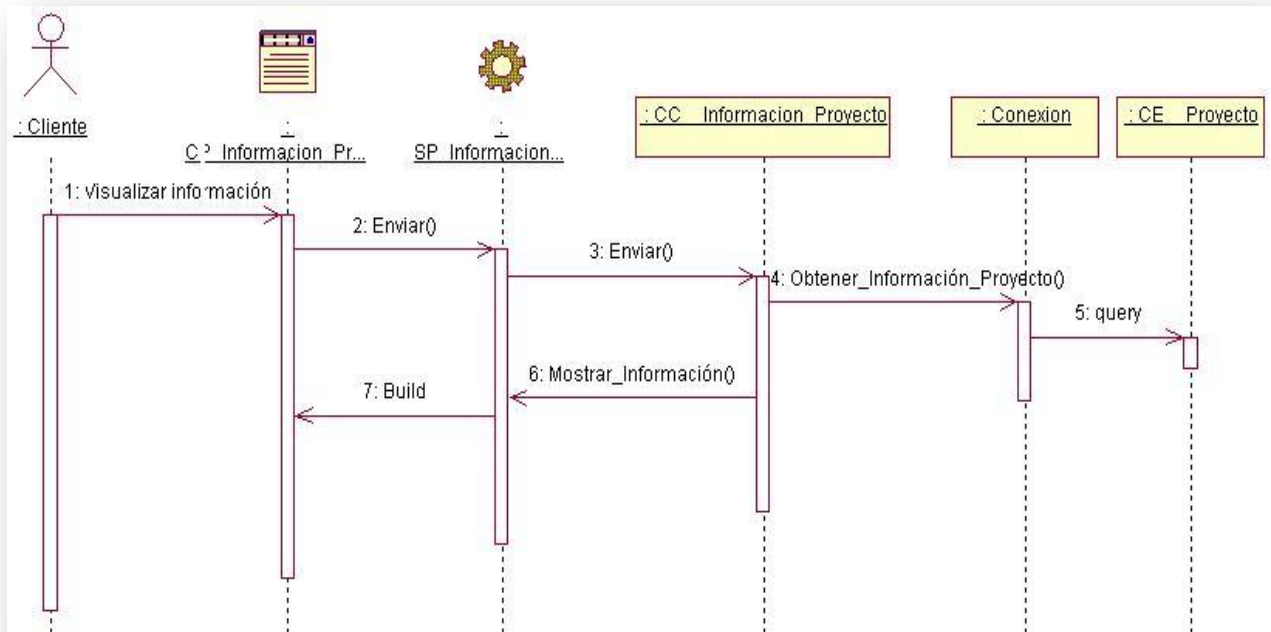
Diagrama de secuencia del caso de uso “Gestionar Foro”. Sección “Eliminar Tema”.



Anexo 13. Diagrama de secuencia del caso de uso “Responder Encuesta”.



Anexo 14. Diagrama de secuencia del caso de uso “Mostrar Información Proyecto”.



Anexo 15. Diagrama de secuencia del caso de uso “Gestionar Proyecto”. Sección “Modificar Proyecto”.

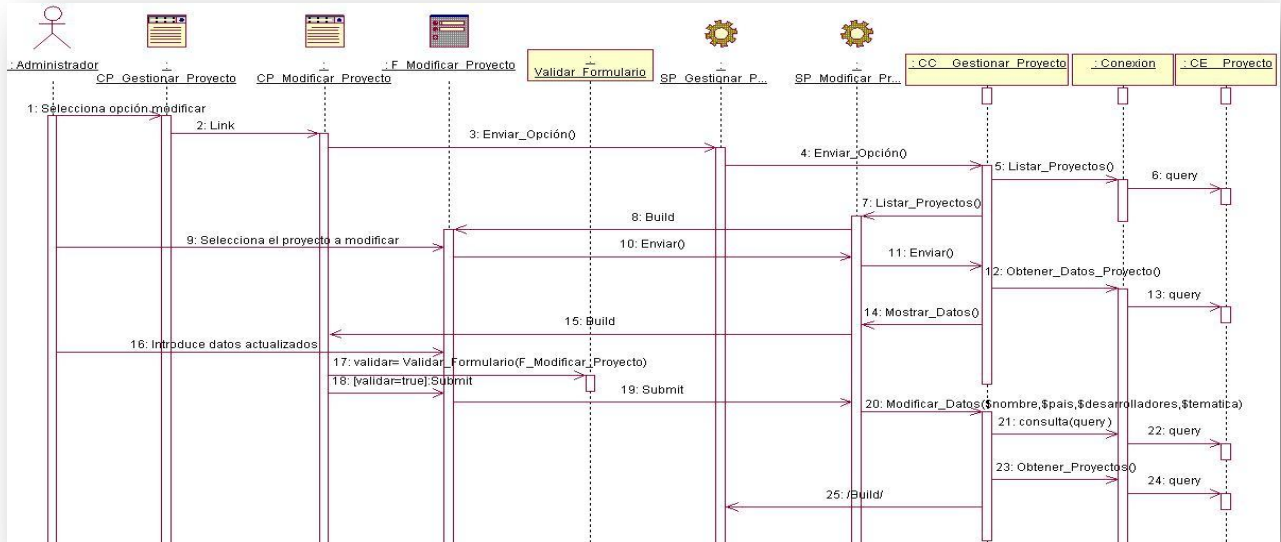


Diagrama de secuencia del caso de uso “Gestionar Proyecto”. Sección “Insertar Proyecto”.

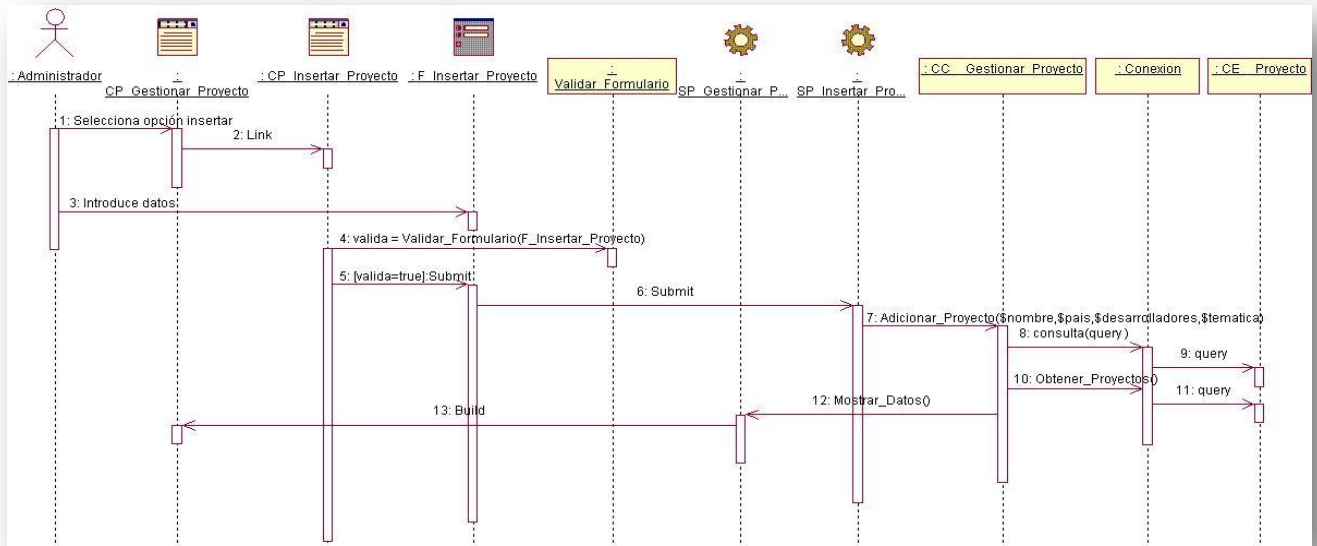
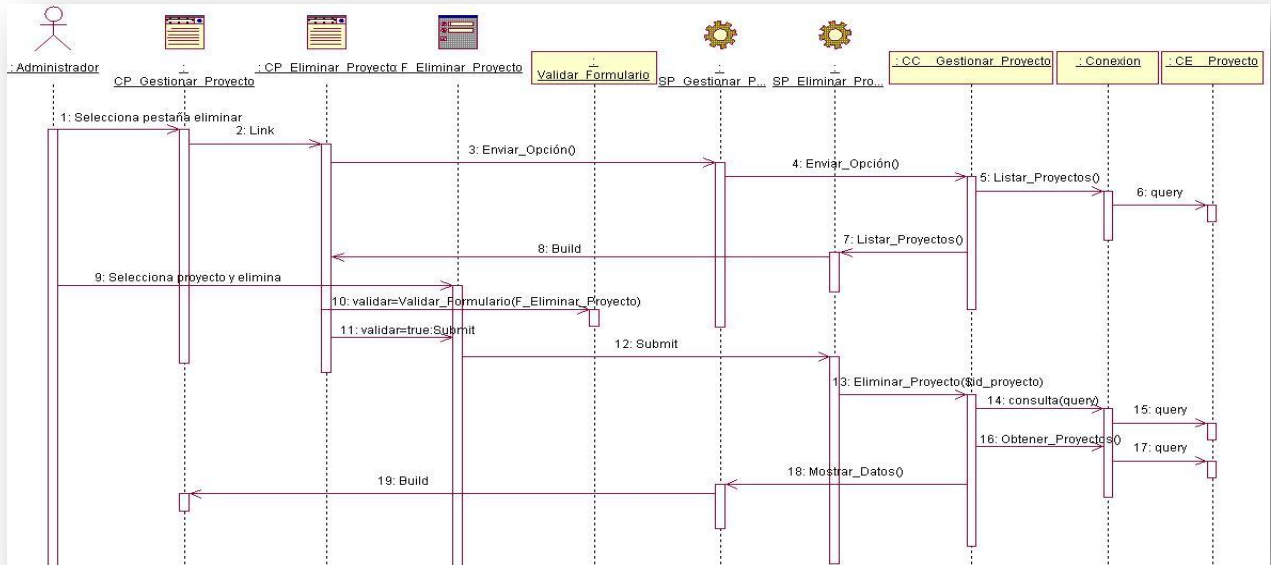


Diagrama de secuencia del caso de uso "Gestionar Proyecto". Sección "Eliminar Proyecto".



Anexo 16. Diagrama de secuencia del caso de uso "Gestionar Usuario". Sección "Adicionar Usuario".

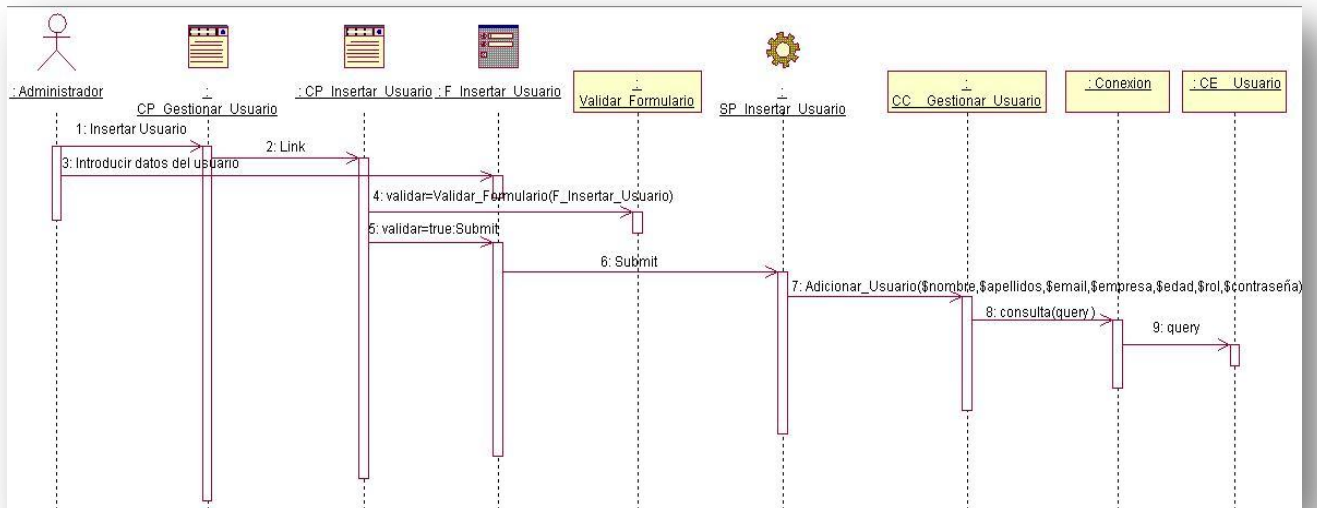


Diagrama de secuencia del caso de uso "Gestionar Usuario". Sección "Eliminar Usuario".

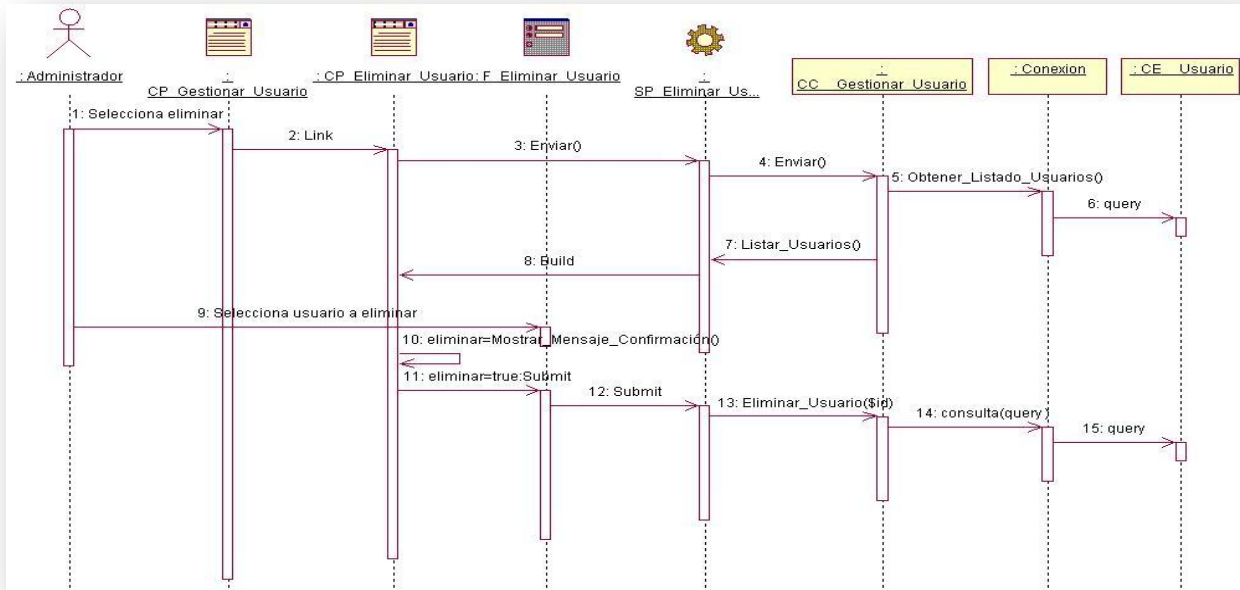
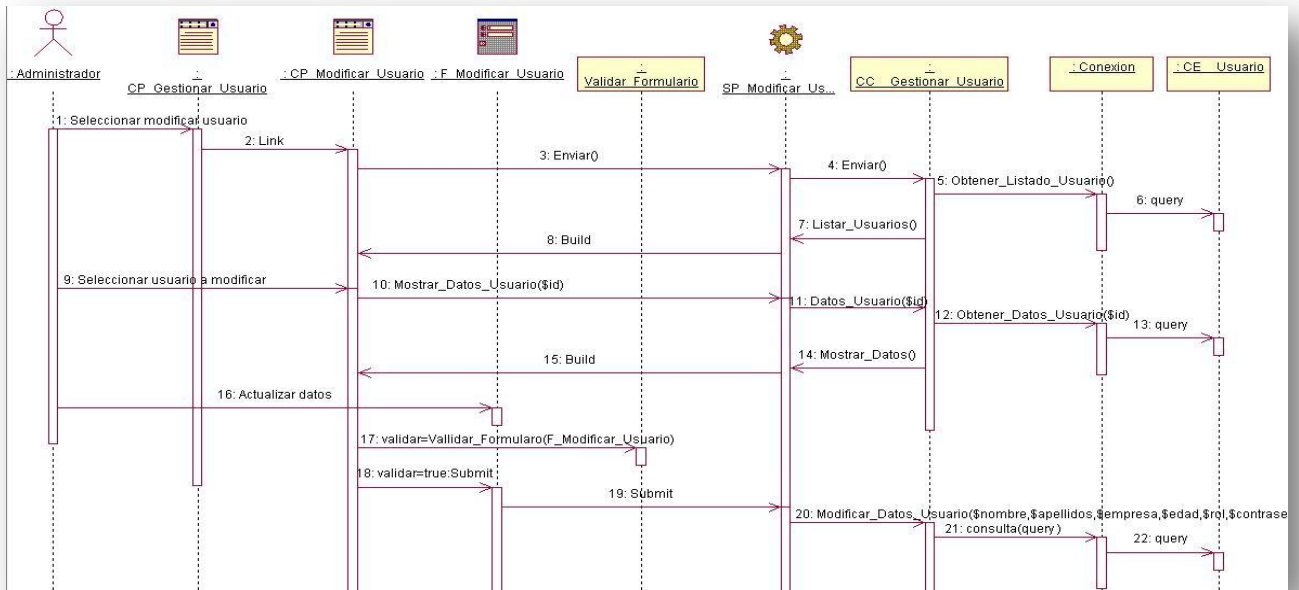


Diagrama de secuencia del caso de uso "Gestionar Usuario". Sección "Modificar Usuario".



Anexo 17. Diagrama de secuencia del caso de uso “Gestionar Grupo Analistas”. Sección “Adicionar Analista al Grupo”.

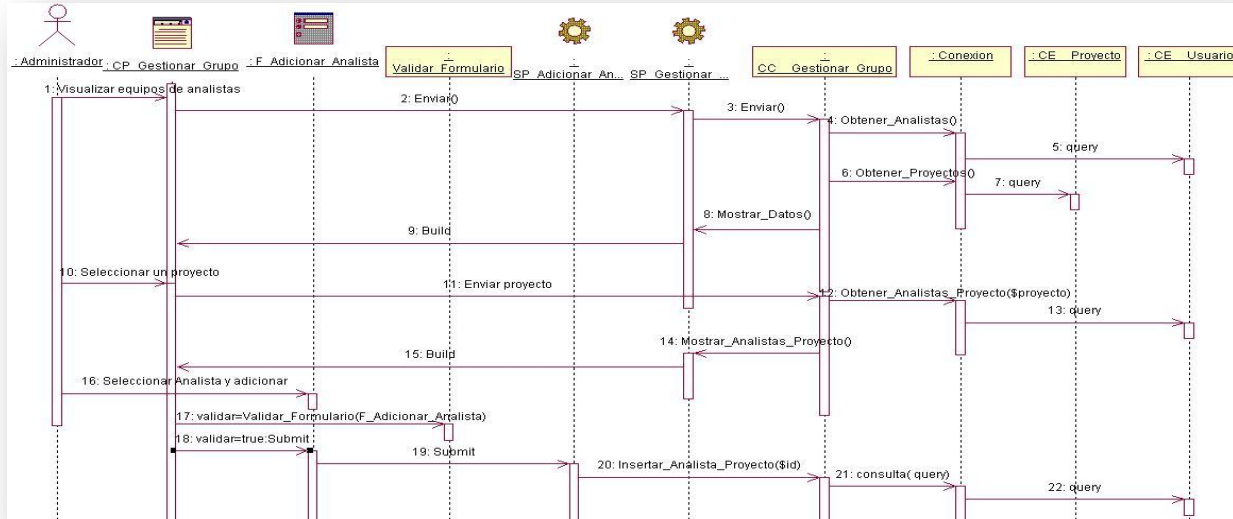
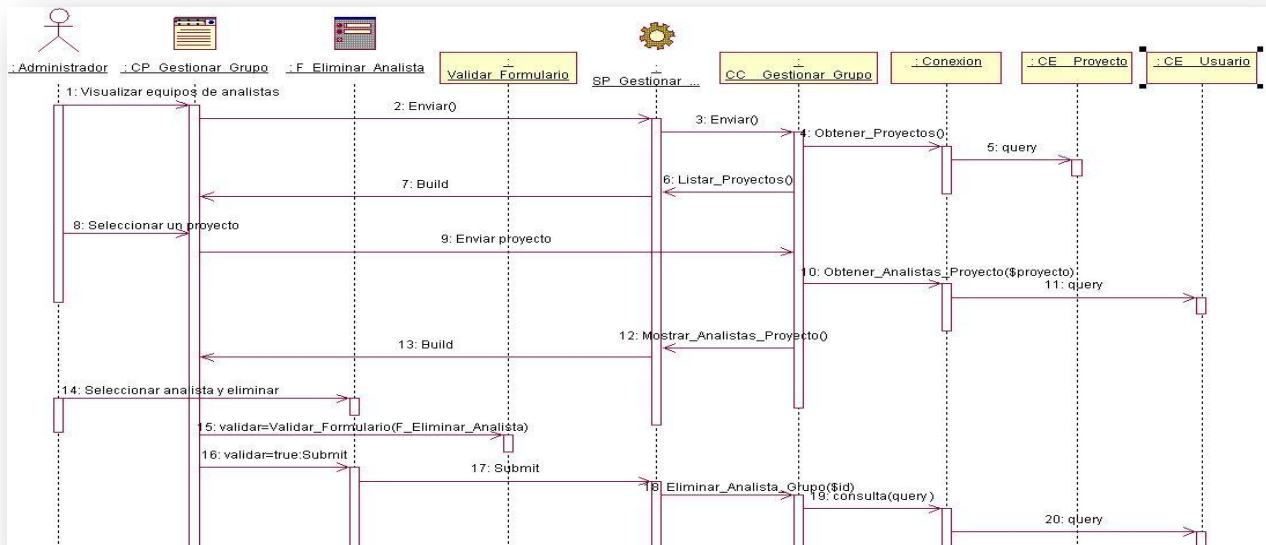


Diagrama de secuencia del caso de uso “Gestionar Grupo Analistas”. Sección “Eliminar Analista del Grupo”.



Anexo 18. Diagrama de secuencia del caso de uso “Enviar Email Notificación”. Sección “Aprobar”.

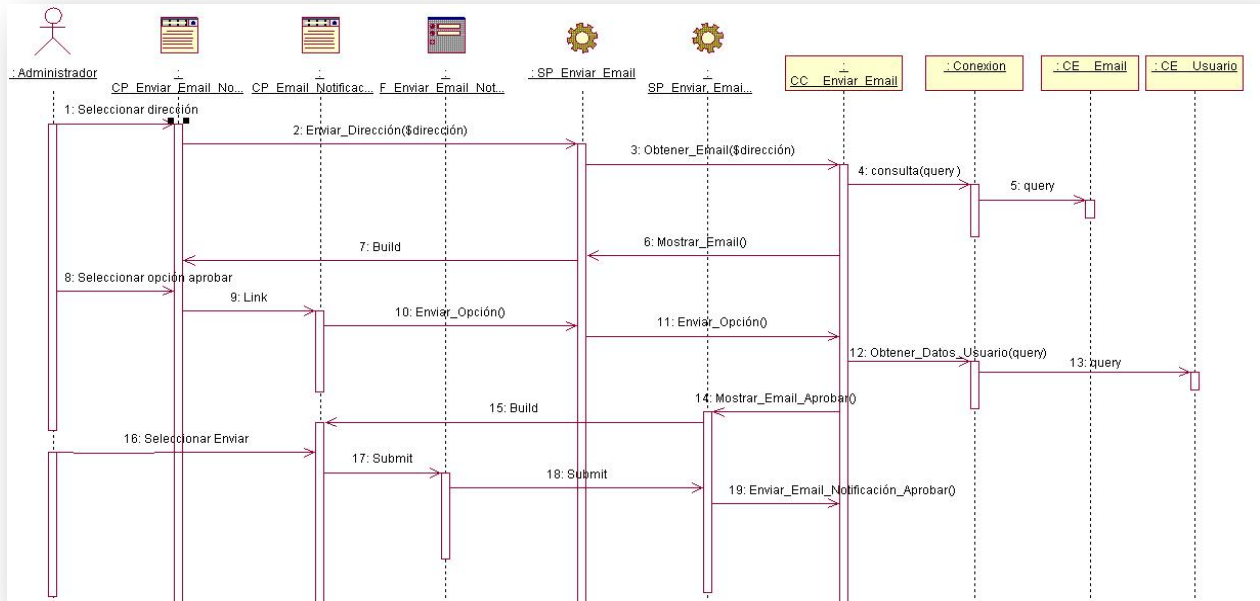
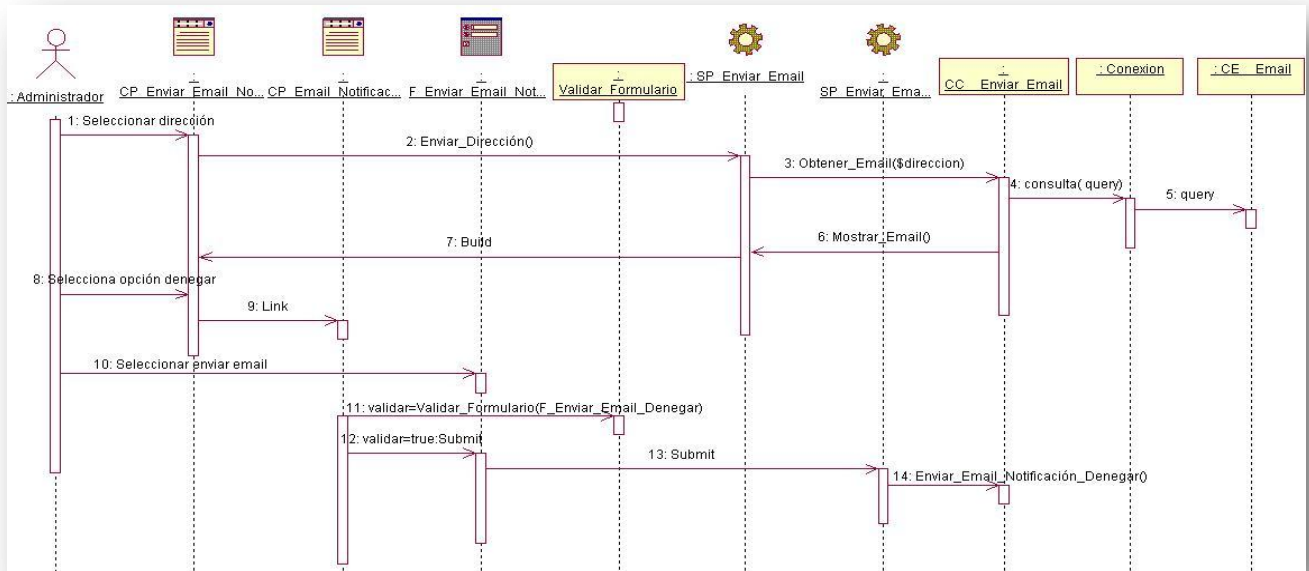
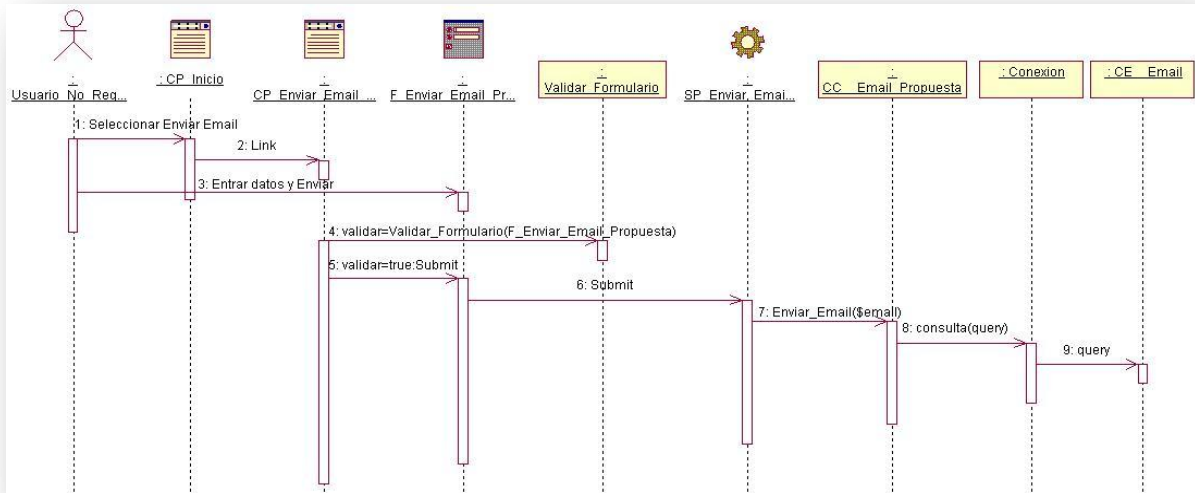


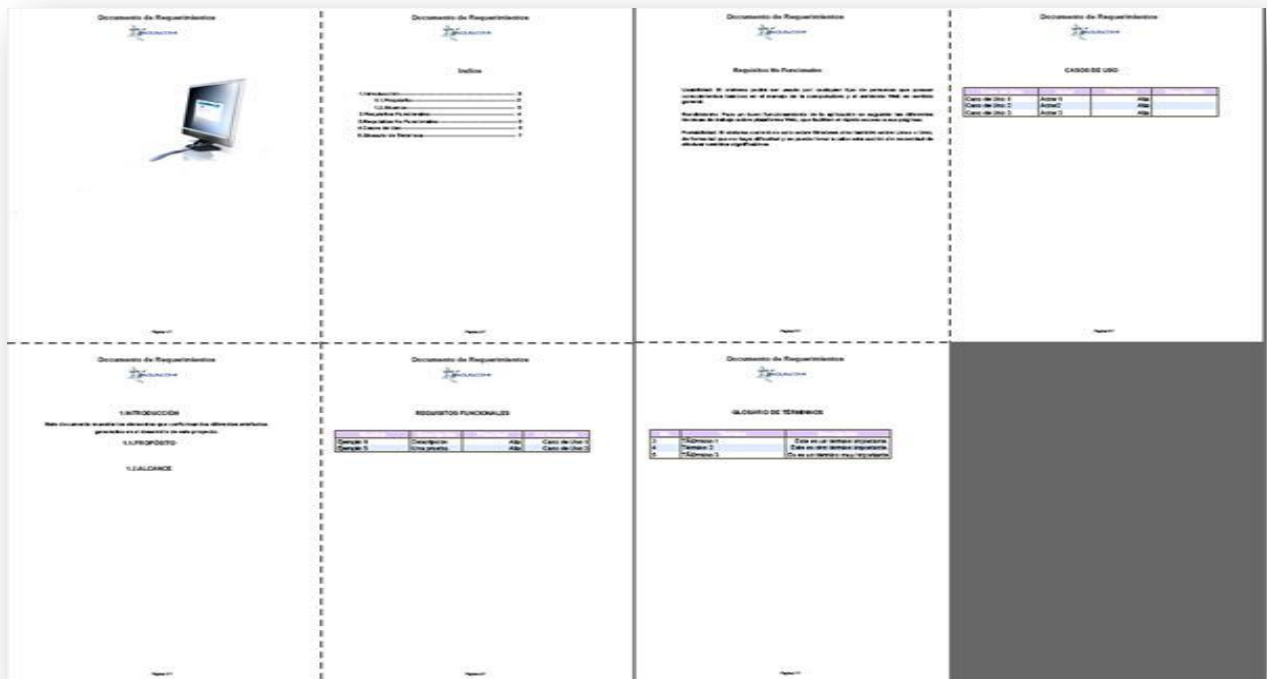
Diagrama de secuencia del caso de uso “Enviar Email Notificación”. Sección “Denegar”.



Anexo 19. Diagrama de secuencia del caso de uso “Enviar Email Propuesta”.



Anexo 20. Ejemplo de reporte.

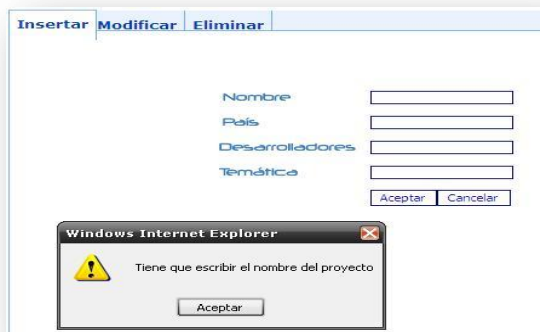


Anexo 21. Tratamiento de excepciones.

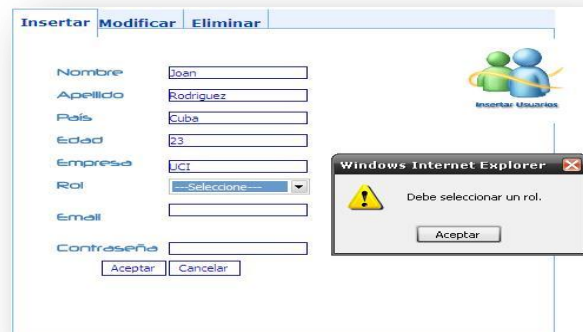
Mensajes de error en la autenticación.



Mensaje de error en la gestión de proyectos.



Mensaje de error en la gestión de usuarios.



Mensaje de error en la gestión de casos de uso.



Error en la gestión de grupos de analistas.

