

**Universidad de las Ciencias Informáticas
Facultad 15**



“Sistema de Gestión de No Conformidades.”

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

Autor(es): Yoanger Sotolongo Cruz.

Roberto Cardero Rodríguez.

Tutor(es): Ing. Belkis Yadira Torres González.

Ing. Leonardo Soto Borroto.

Ciudad Habana

Sistema de Gestión de No Conformidades

Declaración de Autoría

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

Para que así conste firmamos la presente a los ____ días del mes de del año 2010.

Yoanger Sotolongo Cruz

Roberto Cardero Rodríguez

Ing. Leonardo Soto Borroto

Ing. Belkis Yadira Torres González.

Sistema de Gestión de No Conformidades

DEDICATORIA

Soto:

Este trabajo de diploma se lo quiero dedicar primeramente a mis padres que dieron todo su apoyo para que me graduará, también a mi abuela que aunque hace muchos años murió yo sé que me hubiera brindado el mismo apoyo que mis padres, también quiero dedicar este trabajo de diploma a Noemy, que no pudo vivir este momento pero fue una de esas persona que me dió fuerzas para levantarme en momentos difíciles y que siempre creyó en todos nosotros, a mis segundas madres Mima y Martha que supieron y saben hacerme feliz.

Roberto:

Ante todo le dedico este Trabajo a mis abuelitos Chaguito y Estrella, Que en Paz Descansen. A mis otros abuelos Walfrido y Cachita, a mis padres que fueron los principales impulsores de que yo estuviera hoy en día aquí , que día a día me brindan su amor desinteresado y que nunca dudaron de mi; a mi tía Elsa, que Dios la tenga en la gloria.

AGRADECIMIENTOS

Soto:

A todas las personas que de una forma u otra me ayudaron en algún momento de mi carrera, también a los profesores que durante 5 años y específicamente los de primero y segundo me ayudaron y tuvieron mucha paciencia, a mis eternos hermanos que nunca los olvidaré de primero y segundo de aquel apto 9104 que me tendieron la mano y tuvieron paciencia con todas las cosas que le preguntaba, al primer grupo en general porque nos ayudábamos y me sentí como en familia, a mi novia Lili que ha sido la única que me aguantó bastante tiempo, a mis eternas hermanas de la UCI Carmen y Yeny, a mis amistades de tercer año en especial Yenier que nos ayudo siempre a todo el edificio familia el 110 a todos en especial a Carlos el Denis, a mis amistades de la tierra los que se encuentran y los que no se encuentran y a mi hermano Ernesto que la vida profesional.

Roberto:

Les agradezco principalmente a mis padres por su amor infinito; a mis dos hermanos Caco e Israel; a mis tíos Humberto y Carlitos a pesar de sus defectos; a mi familia en general por su apoyo incondicional y por darme todo el aliento necesario para poder realizar este sueño. A mis amigos: David y Soto, por sus consejos válidos y por todo lo que sufrimos juntos. A Javier por ser leal todo el tiempo y confiar siempre en mí. Al Dash y Alain por la ayuda brindada; a todas aquellas personas que han compartido buenos y malos momentos conmigo durante todos estos años. Al piquete de “Los Mamuts”; a mi gente del 110 por pasar agradables momentos de furia y discusiones. Agradecerles especialmente a mis padres de Mayarí: Eulalia y Macías, y toda su familia por acogerme sin ningún interés. A mi chiquitica y su familia, a la “Trituradora” y a todos mis amigos de Mayarí, que siempre los tendré presente. En fin: a todos.



"La calidad nunca es un accidente; siempre es el resultado de un esfuerzo de la inteligencia."

John Ruskin

Resumen

La investigación está asociada al mejoramiento en la Gestión de las No Conformidades de los proyectos productivos en la Universidad de las Ciencias Informáticas. Para el logro de este objetivo se desarrollará una herramienta informática Web que automatice el proceso de identificación de las No Conformidades, capaz de tramitar todos los elementos de las mismas. Se definirá un grupo de roles para determinar el nivel de acceso a la aplicación. El intercambio de información a través de la red, será un aspecto importante, ya que la misma estará almacenada en una base de datos centralizada. Para garantizar la portabilidad de la aplicación se desarrollará bajo ambientes multiplataforma, fundamentalmente con herramientas de software libre.

Sistema de Gestión de No Conformidades

Índice

Introducción	1
CAPÍTULO I FUNDAMENTACIÓN TEÓRICA	4
1.1. Introducción.....	4
1.2. Sistemas de Gestión.	4
1.3. Sistemas actuales de Gestión de No Conformidades.....	4
1.4. Lenguajes de Programación y Frameworks.....	5
1.4.1. Introducción.....	5
1.4.2. Sistemas Gestores de Base de Datos (SGBD)	7
1.5. Metodologías de desarrollo de software	8
1.5.1. RUP (Racional Unified Process).....	8
1.5.2. XP (Extreme Programming).....	9
1.5.3. Metodologías utilizadas.....	9
1.6. Lenguaje de Modelado	9
1.7. Servidores Web.....	10
1.8. Arquitectura.....	11
1.8.1. Arquitectura en capas.....	11
1.8.2. Arquitectura Orienta a Servicios (SOA).....	11
1.9 Patrones de Diseño	12
1.10. Herramientas de Modelado.....	13
1.10.1. Visual Paradigm.....	13
1.10.2. Rational Rose Enterprise.....	13
1.11. NetBeans (IDE de desarrollo).....	14
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA.....	15
2.1. Introducción.....	15
2.1.1. Modelo del negocio.....	15
2.1.2. Responsables del Negocio	15
2.1.3. Diagrama de Procesos del Negocio	17
2.2. Proceso de Gestión de No Conformidades.....	18
2.2.1. Clasificación de las No conformidades	19

Sistema de Gestión de No Conformidades

2.3. Especificación de los requisitos de software	19
2.3.1. Requerimientos funcionales del sistema	19
2.3.2. Requerimientos no funcionales del sistema	22
2.4. Modelo de casos de usos del sistema	23
2.4.1. Actores del sistema.	23
2.4.2. Listado de casos de uso.	23
2.4.4. Especificación textual del Caso de Uso “Generar Reportes”	26
<i>CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA</i>	30
3.1. Introducción.....	30
3.2.1. Diagramas de clases del análisis.....	30
3.2.2. Diagramas de colaboración del Análisis	31
3.3.3. Diagrama de clases persistentes	34
<i>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA</i>	36
4.1. Introducción.....	36
4.2.1. Diagrama de Modelo de datos.....	36
4.2.2. Modelo de despliegue	37
4.2.3. Diagrama de componentes	38
4.2.4. Diagrama de componentes	38
4.2.5. Diagrama de componentes	39
4.3. Prueba	40
4.3.1. Diseños de caso de prueba	41
4.3.2 Pruebas realizadas.....	42
<i>CONCLUSIONES DEL TRABAJO</i>	43
<i>RECOMENDACIONES</i>	43
<i>REFERENCIAS BIBLIOGRÁFICAS</i>	44
<i>ANEXOS</i>	45
Anexos del Capítulo1	45
Anexos del Capítulo 2.....	47
Anexos del Capítulo 3.....	63
Anexos del Capítulo 4.....	68

GLOSARIO DE TÉRMINOS 71

Sistema de Gestión de No Conformidades

Introducción.

El actual desarrollo de la información en la sociedad se caracteriza por el uso intenso de la misma en cada una de sus esferas, así como por la necesidad de identificar y utilizar el conocimiento y la información existentes en las entidades, para ponerlas en función de sus objetivos y desarrollo en general.

En la Universidad de las Ciencias Informáticas a la calidad del software se le dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software con la mejor calidad posible, que cumpla, y que pueda superar las expectativas de los usuarios. El control de la calidad en el ciclo de vida de un proyecto se ha convertido en un elemento crucial para los negocios, con un impacto directo en la minimización de riesgos y costos. Además no se ha transformado únicamente en uno de los requisitos esenciales del producto sino que en la actualidad es un factor estratégico clave del que dependen la mayor parte de las organizaciones, no solo para mantener su posición en el mercado sino incluso para asegurar su supervivencia. Debido al vertiginoso desarrollo que ha experimentado la universidad en lo que a producción se refiere, se ha hecho indispensable realizar una serie de investigaciones en función de la situación actual de las No Conformidades (NC) a la hora de interactuar con toda la información recogida en el Expediente de Proyecto (EP) la cual tiene gran importancia en los procesos de calidad.

Las NC representan alguna insatisfacción del cliente o del equipo revisor, las cuales deben ser corregidas en el menor tiempo posible. Esto representa pérdida de tiempo y en muchas ocasiones la entrega tardía del producto final. Por tanto; el registro de NC es una parte esencial de cualquier sistema de calidad. El primer paso es identificar lo que no se está haciendo bien, y utilizar esta información para detectar y eliminar las causas que provocan el problema. Esto repercutirá en la calidad de todas las actividades que son vitales y obligatorias en todos aquellos proyectos productivos que requiera garantizar un buen funcionamiento. Mediante una investigación se llegó a la conclusión que los sistemas de gestión de la calidad tienen gran importancia ya que garantizan la organización del proyecto y de los grupos de trabajos.

En la actualidad en los proyectos de la Universidad de las Ciencias Informáticas se cometen errores lo que conlleva al registro de No Conformidades. Este control debe permitir que no solo se pueda llevar de forma ágil y organizada sino que incurra en una buena organización para un posterior avance, ya que de esto depende en gran medida la calidad del proyecto, sin embargo solo se enmarca en repartir las tareas por grupos de trabajo y de gestionar reportes lo cual no le es de gran ayuda en general a los líderes de

Sistema de Gestión de No Conformidades

proyectos, atravesando por un proceso mediante el cual un responsable brinda una solución a la misma en un tiempo predefinido, siendo eliminadas posteriormente. Esto ha provocado que, de un proyecto a otro, no se tengan identificadas las NC detectadas anteriormente, generando que se identifiquen en los nuevos proyectos NC ya corregidas, ejecutándose un ciclo continuo de detección y corrección.

A partir de la situación problemática antes expuesta se formula el siguiente **Problema a resolver** ¿Cómo disminuir el carácter repetitivo en la detección de las No Conformidades de un proyecto a otro y al mismo tiempo mejorar la gestión de las mismas dentro de la UCI?

La investigación tiene como **Objeto de Estudio** el proceso de Gestión de las No Conformidades en proyectos productivos de la Universidad de las Ciencias Informáticas y como **Campo de Acción** la gestión de las No Conformidades en la Universidad de las Ciencias Informáticas.

El **Objetivo General** que persigue el trabajo es desarrollar un sistema que permita gestionar y clasificar las NC de los proyectos.

Para cumplir con lo antes planteado se derivan los siguientes **Objetivos específicos**:

- Determinar cuales parámetros de clasificación de las NC no se tienen en cuenta en los proyectos productivos que influyen en la calidad del proceso de desarrollo.
- Analizar las herramientas que permitan obtener una solución óptima.
- Realizar el Análisis, Diseño e Implementación del Sistema de Gestión de No Conformidades.
- Definir la estructura para almacenar las No Conformidades detectadas en los proyectos.

Para cumplir los objetivos trazados, se definieron las siguientes **tareas**:

- Realización de entrevistas a los líderes de proyecto de la facultad.
- Estudio de los elementos de clasificación de No Conformidades.
- Definición de los requerimientos del sistema.
- Definición de las plantillas que utilizarán los probadores para el proceso de detección de No Conformidades.
- Estudio y selección de las herramientas a utilizar para el desarrollo de la aplicación.
- Estudio de las pruebas que se le deben realizar al sistema para comprobar su seguridad.
- Documentación de la situación actual del proceso de gestión de No Conformidad de los proyectos productivos.

A continuación un breve resumen de los capítulos que conforman la investigación.

Capítulo 1 Fundamentación Teórica: contiene algunos sistemas actuales de No Conformidades existentes así como lenguajes de programación y tecnologías tanto del lado del cliente como del lado del servidor. Conjuntamente las metodologías de desarrollo de software utilizadas, sistemas gestores de base de datos, arquitecturas, herramientas manejadas y servidores webs.

Capítulo 2 Características del Sistema: está conformado por el análisis de los procesos, los trabajadores y personas que desarrollan dichos procesos. Se define la propuesta del sistema, el proceso de automatización y la información manejada. Se identifican los casos de uso con sus descripciones, el modelo de objeto, los requisitos tanto funcionales como no funcionales, y los diagramas como: Procesos del Negocio, Casos de Uso.

Capítulo 3 Análisis y Diseño del Sistema: lo conforman los diagramas de colaboración del análisis, para diseñar las clases que se implementarán también se analizan los casos de uso del sistema, el diagrama de las clases diseñadas con sus relaciones, el diagrama de clases persistentes, el diagrama entidad relación y la descripción de las tablas de la base de datos.

Capítulo 4 Implementación y Prueba: está constituido por los diagramas de despliegue, los de componentes y el modelo de prueba correspondiente a la aplicación desarrollada, donde se implementará la propuesta realizada en el análisis y diseño.

CAPÍTULO I FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórico conceptual para el desarrollo de este trabajo, se estudiarán las herramientas a utilizar para el desarrollo del sistema propuesto, se mostrará la arquitectura, se analizará la metodología de desarrollo de software escogida así como lenguajes de programación del lado del cliente y del servidor.

1.2. Sistemas de Gestión.

Las instituciones actuales desarrolladoras de software afrontan desafíos significativos, entre ellos: crecimiento y tecnología, competitividad, velocidad de cambios. La implementación de un sistema de gestión estable puede ayudar a:

- ✚ Ampliar la complacencia de los clientes.
- ✚ Disminuir costos.
- ✚ Lograr mejoras en la organización de la institución.

Además respetan factores de calidad del software los cuales se agrupan en tres aspectos: los cambios, su adaptabilidad a nuevos entornos, sus características operativas.

1.3. Sistemas actuales de Gestión de No Conformidades.

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. Estos sistemas son los encargados de facilitarle al líder del proyecto y a diferentes trabajadores de determinado grupo de trabajo del proyecto el trabajo y la organización, además esas herramientas humanizan el trabajo de los probadores y diferentes integrantes del grupo de trabajo de calidad. En la actualidad en el mundo se han vuelto necesarios dichos sistemas y algunos se adaptan al campo de acción entre los que se debe

Citar: **IGQuality (Integrated Global Quality)** que es una herramienta para la gestión de las actividades de mejora de la empresa, consecución de objetivos, dominio de procesos, disminución de No Conformidades, control de documentación, y gestión de personal. Este sistema avala el cumplimiento de los distintos sistemas certificables reconocidos como ISO 9001, Medioambiente ISO14000, Gestión de proyectos UNE 16601, Gestión de Higiene y Seguridad OSHA 18001. IGQuality aplica metodologías para el registro, la acción inmediata, la resolución e investigación de la causa de la NC para su posterior análisis de mejora. También posee un software(A-Gestion) para la gestión de normas: UNE-EN ISO 9001, UNE-EN ISO 14001, BRC-IFS. Este permite la gestión de las No Conformidades y Acciones Correctivas y

Preventivas. Asimismo se puede señalar el **KMKey Quality**, software de gestión de calidad ideal para la implantación y mantenimiento de un sistema de gestión de calidad (SGC) de cualquier tipo. Incorpora un Gestor Documental donde almacenar las distintas revisiones de los documentos propios de un sistema de calidad, tales como el Manual de Calidad, la Política y los Objetivos de Calidad permitiendo siempre un acceso directo a la revisión vigente. Igualmente **REDMINE** es un sistema que gestiona las tareas de diferentes grupos de trabajo por proyectos. Este sistema se le integrarán todos los proyectos productivos de la UCI ya sean Aduanas, Sigep Venezuela, Sigep Cuba, Banco. Brinda distintos servicios importantes para el desarrollo de la producción, por él los diferentes líderes de proyecto les pueden asignar a los Responsables de los grupos de calidad tareas a realizar y por el mismo pueden revisar si las mismas ya han sido cumplidas en su totalidad. En nuestra universidad con respecto a la gestión de las NC se han realizado varios trabajos investigativos y tesis dentro de las que cabe destacar **Sistema de Gestión de No Conformidades** realizada por los Ingenieros **Martha Nieves Borrero, Asnier Góngora Rodríguez**. Los sistemas antes planteados se limitan al hecho de adicionar, aceptar, cerrar y eliminar las NC detectadas, tratando de forma inefectiva la clasificación y gestión de los datos de estas (algunos obvian el proceso de clasificación), lo que conlleva a que no se tenga un estricto control de ¿cuales se repiten de un proyecto a otro? ó ¿cuales tienen mayor incidencia de un proyecto a otro? las NC al ser clasificadas permiten una mayor organización, lo que ayudará a un mayor encapsulamiento de los errores, además una vez identificadas, ayudarán a generar reportes que ayuden a realizar estudios para mejorar la continua repetición de las mismas y de esta manera conseguir que disminuyan paulatinamente.

1.4. Lenguajes de Programación y Frameworks.

1.4.1. Introducción.

Los lenguajes de programación son importantes pues permite crear programas y software que sirven para el desarrollo en general de la mayoría de las industrias, además la humanidad se ha vuelto cada vez más dependiente de lo que ellos desarrollan para subsistir.

Los Frameworks (marco de trabajo), son soluciones completas que contemplan herramientas de apoyo a la construcción. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

1.4.1.1. Java Script

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Es el lenguaje que permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. Es el lenguaje de programación del lado del cliente más utilizado, siendo compatible con la mayoría de los navegadores modernos. El navegador del cliente es el encargado de interpretar las instrucciones Java script y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador. Este lenguaje permite validar y ejecutar acciones básicas del lado del cliente haciendo que la aplicación no sobrecargue el servidor; ya que se encarga de realizar acciones, como pueden ser pedir datos, mostrar mensajes, etc.

1.4.1.2. Html

HTML, es el lenguaje de marcado para la construcción de páginas web. Este lenguaje se basa en instrucciones que le dicen al texto como deben mostrarse (tags), y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores.

1.4.1.3. PHP

Es un lenguaje de programación del lado del servidor, que permite acceder a los recursos que tenga el servidor y el resultado enviarlo al navegador. Está creado para el desarrollo de páginas Web dinámicas. Puede ser incluido con facilidad dentro del código HTML.

PHP es gratuito, puede descargarse y utilizarse en cualquier aplicación, personal o profesional cuando se desee sin costo alguno. Dispone de una conexión propia a todos los sistemas de base de datos. PHP está disponible para una gran cantidad de sistemas operativos diferentes. Al tratarse de código abierto posibilita el acceso al código fuente de PHP, a diferencia de los productos comerciales y de código cerrado.

1.4.1.4. JAVA

Es un lenguaje de programación orientado a objetos que se refiere a un método de programación y al diseño del lenguaje. También es multiplataforma, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Proporciona la posibilidad de desarrollar pequeñas aplicaciones y luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Estas mini-aplicaciones se

ejecutan en una JVM que el navegador tiene configurada como extensión (*plug-in*) en un contexto de seguridad restringido configurable para impedir la ejecución local de código potencialmente malicioso. Contiene Servlets y JSP (Java Server Pages) las cuales posibilitan disminuir diferentes inconvenientes como: lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica.

1.4.1.5. Ext. JS

Es un framework implementado en java script, es una librería de código abierto para la creación de interfaces usuarios en el desarrollo de aplicaciones Web. Ofrece una serie de librerías para integrar en el sistema. Entre las posibilidades que ofrece se encuentra la creación de formularios, combos, grids o menús. A parte ayuda a la comunicación entre el cliente y el servidor. Ext JS es compatible con varios navegadores como Internet Explorer 6+, Firefox 1.5+. La versión utilizada de Ext JS fue la 3.1.

1.4.1.6. CodeIgniter

CodeIgniter (CI) es framework PHP que contiene un conjunto de herramientas para construir aplicaciones web. Minimiza la cantidad de líneas de código en el desarrollo de una aplicación, lo que permite que cargue más rápido y que consuma menos espacio en memoria. Es perfectamente compatible con todas las versiones de PHP. El sistema requiere solo unas bibliotecas muy pequeñas. CodeIgniter se basa en el modelo de desarrollo Modelo-Vista-Controlador (MVC) que permite una buena separación entre la lógica del negocio y la presentación. CodeIgniter permite acceder a cualquier base de datos, permite el envío de correo electrónico, validación de los datos de un formulario, mantenimiento de sesiones. También permite poner el URL del sitio en un archivo de la configuración pudiendo acceder al resto del sitio. Cuando se desee mover el sitio, solo se necesita cambiar el archivo de la configuración, y todos sus hipervínculos se ponen al día automáticamente. Se utilizó el CodeIgniter versión 1.7.1.

1.4.2. Sistemas Gestores de Base de Datos (SGBD)

Un Sistema Gestor de base de datos es importante en toda aplicación. Por tanto debe permitir:

- ❖ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ❖ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- ❖ Manipular la base de datos: realizar consultas, actualizarla, generar informes.

1.4.2.1. PostgreSQL

Es un Sistema Gestor de Base de Datos (SGBD) relacional orientado a objetos de software libre. Como su nombre lo indica, utiliza el lenguaje SQL para llevar a cabo sus búsquedas de información. Es multiplataforma, ya que puede operar en varios Sistemas Operativos.

Está caracterizado por su alta concurrencia, pues mediante su sistema denominado Acceso Concurrente Multi Versión (MVCC) permitiendo el acceso a las tablas que están bajo un proceso que se está ejecutando en ellas sin necesidad de bloqueos.

1.4.2.2. Microsoft SQL Server

Es un sistema para la gestión de bases de datos basado en el modelo relacional. Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información. Además permite administrar información de otros servidores de datos. Soporta procedimientos almacenados. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

1.4.2.3. ¿Por qué utilizar PostgreSQL?

Contiene una alta concurrencia ya que mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. Además es un sistema gestor de base de datos no privativo y de código abierto.

1.5. Metodologías de desarrollo de software

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

1.5.1. RUP (Racional Unified Process)

La metodología RUP, divide en 4 fases el desarrollo del software, representando en ciclos el desarrollo de un producto de software. Posee tres características fundamentales, y es que está dirigido por casos de usos, va a ser iterativo e incremental y centrado en la arquitectura.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos. Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos. RUP utiliza UML (herramienta potente en la creación de diagramas que brinda una mejor perspectiva de lo que se quiere). Además es efectiva, pues está conformada por Flujos de trabajos y Fases que ayuda mucho a agilizar el trabajo.

1.5.2. XP (Extreme Programming)

Es una de las metodologías de desarrollo de software ligera que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Tiene dentro de sus principales características la realización de pruebas a sus principales procesos, buscando posibles fallas que pudieran ocurrir. Una de las características más importantes es la reutilización de código, para lo cual se crean patrones o modelos estándares. Tiene como particularidad la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Lo fundamental en este tipo de metodología es la comunicación, entre los usuarios y los desarrolladores. Teniendo como objetivo primordial la satisfacción del cliente.

1.5.3. Metodologías utilizadas

Una vez estudiada las metodologías anteriores, se decide atendiendo a sus características utilizar elementos de ambas metodologías (XP, RUP) que permitan desarrollar de forma ágil y consistente. RUP utiliza el Lenguaje Unificado de Modelado (UML), un lenguaje potente que facilita el entendimiento entre el cliente y los desarrolladores. Además es efectiva pues está conformada por flujos de trabajo y fases que ayuda al equipo de desarrollo a agilizar y organizar el trabajo. XP posee diferentes características que permiten desarrollar en menor tiempo como la programación en pares. Tiene como objetivo fundamental dar al cliente el software que necesita y potenciar al máximo el trabajo en grupo.

1.6. Lenguaje de Modelado

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje. UML no es una guía para realizar el

análisis y diseño orientado a objeto, es decir no es un proceso. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objeto.

1.7. Servidores Web.

Son programas que están diseñados para transferir hipertextos, páginas web o páginas HTML : textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP que pertenece a la capa de aplicación del modelo OSI.

1.7.1. Apache Tomcat

Es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

1.7.2 Servidores Web

Existen distintos tipos de servidores diseñados para la web. Se pueden citar entre ellos Internet Information Server y Apache. Para el desarrollo del sistema se utilizará Apache.

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características destacan:

Multiplataforma

Es un servidor web conforme al protocolo HTTP.

Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.

Se desarrolla de forma abierta.

Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.8. Arquitectura

La arquitectura de software es el diseño de más alto nivel de la estructura del sistema, que tiene como responsabilidad ensamblar los elementos del diseño, dígase módulos, clases y la interacción entre estos. Define la responsabilidad de los elementos del diseño y el control del flujo de datos en el mismo. Según IEEE Std 1471-2000: “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”.

1.8.1. Arquitectura en capas

El modelo n- capas ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma.

La arquitectura tres capas es una de las tres propuestas de arquitecturas de capas para sistemas de información la cual dispone de tres tipos de nodos:

Clientes: interactúan con los usuarios.

Servidores de aplicación: procesan los datos.

Servidores de base de datos: almacenan todos los datos.

Presenta como características fundamentales la separación de la lógica de negocios de la lógica de diseño, permite construir y probar el modelo independientemente de la representación visual, no existe dependencia directa del modelo con respecto a la vista y posee una mayor adaptabilidad al cambio.

1.8.2. Arquitectura Orienta a Servicios (SOA)

La arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y recombinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento de una organización. El resultado final es que las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva.

¿Por qué utilizar Arquitectura en 3- capas?

Para el desarrollo de la aplicación se seleccionó la arquitectura en 3-capas. La ventaja principal de esta es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido, es decir permite implementar los componentes por separado; la conexión entre el Modelo y sus Vistas es ágil. Admite refinamientos, optimizaciones y amplia reutilización de código.

1.9 Patrones de Diseño

El diseño propuesto fue creado con los siguientes patrones, que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, permitiendo llevar a cabo la implementación del sistema informático bajo estos como el Modelo Vista Controlador (MVC) y los Patrones Generales de Software para Asignación de Responsabilidades (GRASP).

- ✚ **MVC:** El patrón MVC realiza un diseño que desacopla la vista del modelo, con la finalidad de mejorar la reusabilidad, permitiendo esto que las modificaciones realizadas en las vistas influyan en menor medida en la lógica de negocio o de datos.
- ✚ **Creador:** El patrón Creador se encarga de identificar la clase responsable de la creación de nuevos objetos. Presente en todas las clases controladoras cuando crean diferentes objetos del modelo según los datos que necesiten.
- ✚ **Alta cohesión:** El patrón Alta Cohesión indica que la información que almacena una clase debe ser coherente, de manera que todos sus métodos tengan un comportamiento bien definido. Es utilizado para la construcción del sistema informático debido a que cada clase implementa las funcionalidades que le son correspondidas.
- ✚ **Bajo acoplamiento:** El patrón Bajo Acoplamiento, su objetivo en el sistema informático es tratar de mantener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas se tenga la mínima repercusión posible en el resto, potenciando la reutilización y disminuyendo sus dependencias.
- ✚ **Experto:** El patrón Experto es el encargado de asignar una responsabilidad al experto en información. La clase controladora delega la responsabilidad a las clases modelos de realizar la acción, por ejemplo al insertar, las modelos son las encargadas de realizar esta función porque

conocen toda la información.

- ✚ **Controlador:** El patrón Controlador se encarga de gestionar los eventos generados en capas anteriores. Las clases controladoras a partir de dichos eventos toman las decisiones apropiadas, invocando funcionalidades contenidas en capas más profundas como el acceso a datos. Su función es de mediador o intermediario, del controlador del negocio asociado que en este caso son las clases modelos.

1.10. Herramientas de Modelado

1.10.1. Visual Paradigm

Su mayor éxito consiste en la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. Visual Paradigm utiliza UML como lenguaje de modelado ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido, bien y más barato. Es fácil de usar y presenta un ambiente gráfico agradable para el usuario. Su notación es parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo.

1.10.2. Rational Rose Enterprise

Es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a Unified Modeling Language (UML), pero no son compatibles con las mismas tecnologías de implementación. Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software.

¿Por qué utilizar Visual Paradigm?

Es una herramienta CASE, que posee características que ayudan en gran medida a quien la utiliza, dentro de las que cabe destacar que se ejecuta sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma. También ofrece soluciones de software que permiten a las organizaciones desarrollar las aplicaciones con un alto nivel de calidad, en el menor tiempo posible,

generando costos bajos. Contiene un alto grado de configuración ayudando a los usuarios a desarrollar de forma ágil y organizada. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario.

1.11. NetBeans (IDE de desarrollo)

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

Conclusiones

El presente capítulo recogió el resultado del análisis de los procesos fundamentales relacionados con el trabajo. Se hace una descripción de la tecnología que se utiliza para el desarrollo del sistema, se concluye que las metodologías utilizadas para el desarrollo del sistema sean en conjunto RUP y XP; UML como lenguaje de modelado; Visual Paradigm como herramienta case; PHP y JavaScript como lenguajes de programación con el apoyo de los Frameworks CodeIgniter y Ext JS además como servidor web se utilizará WAMP y PostgreSQL como gestor de base de datos.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En el siguiente capítulo se modela el Negocio, definiéndose las características del mismo. Se presenta una descripción detallada de los procesos, se capturan los requisitos funcionales y no funcionales. Se definen y describen los casos de uso del sistema y sus actores.

2.1.1. Modelo del negocio

El objetivo del modelo del negocio es comprender la estructura del negocio así como obtener una visión de la organización. Además especificar los procesos, roles y responsabilidades. Propiciando que los usuarios finales y los desarrolladores tengan un entendimiento común de la organización del sistema.

2.1.2. Responsables del Negocio

Un responsable del negocio es una persona (o grupo de personas) que actúa en el negocio realizando uno o varios procesos, interactuando con otros responsables del negocio y manipulando entidades del negocio.

A continuación se presenta la descripción de los Responsables del Negocio

Nombre	Descripción
Líder de proyecto	Persona que le indica al Responsable de Calidad el comienzo de la realización de pruebas al software. Es quien inicia las operaciones que dan comienzo al proceso de negocio.
Responsable de Calidad	Persona que determina las pruebas a realizar y las No Conformidades que proceden o no. Además realiza la asignación de las mismas a sus respectivos grupos de trabajo.
Probador	Persona que realiza las pruebas establecidas y registra las NC de la aplicación vigente.
Equipo de Desarrollo	Grupo de personas que realizan las acciones correctivas a las NC

Tabla 2.1 Justificación de los trabajadores del Negocio.

A continuación se definen y describen las actividades del negocio.

Descripción de las actividades del negocio	
Actividad	Descripción

Sistema de Gestión de No Conformidades

Orientar Prueba	Proceso mediante el cual el Líder del proyecto orienta que se deben realizar pruebas, puntualizando que no debe tardar más de una semana obtener los resultados de las mismas.
Definir Pruebas	Proceso mediante el cual el Responsable de Calidad basándose en sus conocimientos y estudios realizados define las pruebas que se le deben realizar al software, buscando como objetivo la mejora de la organización.
Realizar Pruebas	Proceso mediante el cual el probador para realizar las pruebas indicadas consulta el documento redactado por el responsable de calidad y realiza las mismas de forma ágil y organizada. Si no detecta NC termina la actividad sino pasa a otra actividad.
Registrar NC	Proceso mediante el cual el probador registra las NC detectadas en las pruebas realizadas a la aplicación; creando así un depósito donde se recogen las mismas, el cual precisará datos de interés para proyectos posteriores.
Aprobar NC	Proceso mediante el cual el Responsable de calidad una vez recibidas las NC detectadas por el probador basándose en sus conocimientos básicos determina si se deben o no de aprobar las NC. Proceso importante pues es prácticamente el filtro de todas las NC del proyecto.
Reenviar NC	Proceso mediante el cual el Responsable de Calidad después de haber o no aprobado las NC, se las reenvía al probador si no las aprueba o se dispone a enviárselas al equipo de desarrollo y termina así esta actividad.
Corregir NC	Proceso mediante el cual el equipo de desarrollo una vez recibidas las NC aprobadas por el Responsable de Calidad, realiza la corrección de las mismas y determinan si se pueden o no hacer las acciones correctivas desarrollando sus conocimientos básicos requeridos.
Modificar Documento	Proceso mediante el cual el Equipo de Desarrollo se dispone a modificar el documento creado por el probador con las actualizaciones y cambios realizados con respecto a las NC el cual posteriormente debe ser

Sistema de Gestión de No Conformidades

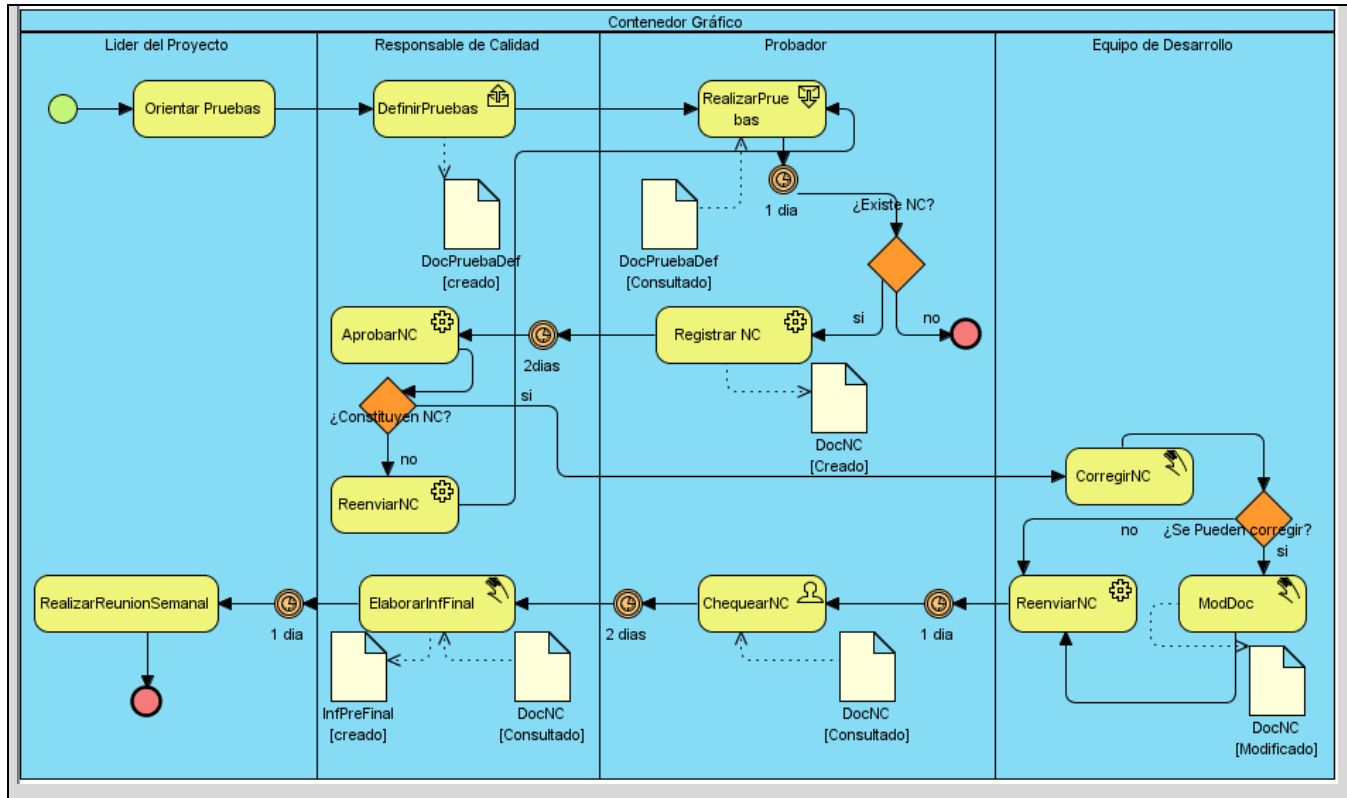
	chequeado.
Enviar NC	Proceso mediante el cual el Equipo de Desarrollo realiza el envío de las NC si no se pueden corregir por un u otro problema.
Chequear NC	Proceso mediante el cual el probador realiza un chequeo de las NC enviadas por el Equipo de Desarrollo donde revisa si las mismas ya han sido resueltas y crea un documento donde se recogen detalles específicos de las mismas.
Elaborar Informe Final	Proceso mediante el cual responsable de calidad consulta el documento creado por el probador y resume lo esencial con respecto a las NC. Es de suma importancia para la realización de la reunión semanal pues sirve de guía al Líder de Proyecto.
Realizar Reunión Semanal	Proceso mediante el cual el Líder del Proyecto una vez recibido el documento elaborado por el Responsable de Calidad que resume todas las características y datos esenciales de las NC tanto corregidas o no, realiza la reunión semanal en la cual el tema principal es el seguimiento de las NC y que medidas van a realizar para eliminarlas.

Tabla 2.2 Descripción de los Procesos del Negocio.

2.1.3. Diagrama de Procesos del Negocio

Diagrama de Procesos del Negocio

Sistema de Gestión de No Conformidades



2.2. Proceso de Gestión de No Conformidades

El proceso de gestión de No Conformidades tiene como objetivo principal promover la organización de los diferentes grupos de trabajo al realizar el proceso de detección y eliminación de las NC, una vez realizadas las pruebas, el proceso antes mencionado comienza cuando el Líder del proyecto le indica al Responsable de Calidad que se deben realizar pruebas y que los resultados se le deben mostrar semanalmente, posteriormente el Responsable de Calidad aplicando los conocimientos requeridos precisa las pruebas que se le deben realizar al software proceso del cual se hará cargo el probador, el mismo tendrá que consultar los detalles que agregó el Responsable de calidad, a continuación dicho Responsable una vez agregada las NC detectadas por el probador, será el encargado de aprobar o no las NC antes detectadas, lo cual determinará la eliminación de las mismas o la asignación al grupo de trabajo encargado de exterminarlas. Todos los detalles y resultados obtenidos se recogerán en un documento que posteriormente se le mostrará al Líder de Proyecto que servirá para la realización de la reunión semanal de análisis de proyecto.

2.2.1. Clasificación de las No conformidades

Las clasificaciones de las NC varían de la documentación a la aplicación aunque existen similitudes en los defectos categorizados como **Ortografía**, en esta se recoge todo lo referente a los errores ortográficos, gramaticales y de concordancia, estas son siempre de tipo significativa; **Validación** se refiere a los mensajes que deben ser mostrados en la aplicación y no aparecen recogidos en los diseños de casos de prueba(DCP) y viceversa, incluye también los campos que no están correctamente validados según la entrada de datos que deben tener y son de tipo significativa; **Funcionalidad** en la documentación recoge los errores en los pasos del flujo central que guían lo que debe hacer el usuario y aparecen en la sección Requisitos a probar del DCP, en la aplicación concierne a los errores que no permiten que una funcionalidad determinada se ejecute, son de tipo significativa; **No correspondencia con los casos de uso o los requisitos** trata los errores cuando algún detalle no coincide entre DCU o Diseño de requerimientos (DR), aplicación y DCP, estas son de tipo significativa; la clasificación **Otros** recoge los errores relacionados con estándares de diseño o plantillas y son de tipo no significativas a excepción de los errores idiomáticos que son la mezcla de dos o más idiomas y se clasifican de tipo significativa. Propias de la aplicación son **Opciones inhabilitadas** que agrupan los errores de componentes, opciones o funciones que aparecen sin desarrollar ninguna operación y son de tipo significativa. **Excepciones** por su parte, recoge los errores de excepciones controladas o no controladas por el sistema que no se corresponden con los errores cometidos y son significativa. De tipo **Interfaz** son los errores de incumplimiento con los estándares para el diseño de interfaces y son de tipo significativa. El probador efectúa las pruebas, detecta diferentes NC y realiza el proceso de clasificación de las mismas, llenando los campos del formulario Adicionar NC y elaborando una descripción donde plasma en que consiste la NC.

2.3. Especificación de los requisitos de software

Los requerimientos del software definen el producto de software que se va a construir, son características que poseerá el producto. Se dividen en dos grupos, requerimientos funcionales y no funcionales.

2.3.1. Requerimientos funcionales del sistema

Los requerimientos funcionales son las características básicas del sistema que definen el comportamiento interno del mismo, en ellos se recogen las diferentes operaciones que debe realizar el producto.

Identificador	RF1 Autenticar Usuarios
---------------	-------------------------

Sistema de Gestión de No Conformidades

Requisito funcional que debe permitir la autenticación del usuario en el sistema.

Identificador	RF2 Gestionar Líderes de proyecto
---------------	--

Requisito funcional que debe permitir a los administradores generales, la gestión de los Líderes de proyecto, brinda y distribuye la cadena de mando dentro del proyecto y proporciona una mayor organización y estructuración del mismo.

RF2.1 Adicionar Líderes de proyecto.

RF2.2 Modificar Líderes de proyecto.

RF2.3 Eliminar Líderes de proyecto.

Identificador	RF3. Gestionar Usuario de Proyecto
---------------	---

Este requisito que debe permitir a los líderes de proyecto poder crear diferentes actividades como Adicionar usuarios los cuales tendrán diferentes roles, también modificarlos en caso de algún contratiempo o eliminarlos así como mostrarlos.

RF3.1 Adicionar usuarios de proyectos

RF3.2 Modificar usuarios de proyectos

RF3.3 Eliminar usuarios de proyectos

Identificador	RF4. Gestionar proyecto
---------------	--------------------------------

Este requisito que debe permitir a los administradores realizar diferentes actividades con respecto a los proyectos que participan, como mostrarlos en la página principal, editarlos por si alguno ya no es de interés y diferentes acciones comunes como adicionarlos y eliminarlos

RF4.1 Adicionar proyectos

RF4.2 Modificar proyectos

RF4.3 Eliminar proyectos

Identificador	RF5. Gestionar elemento de prueba
---------------	--

Requisito funcional esencial para los procesos de prueba en el cual debe permitir determinar acciones importantes para el probador, proporcionar las pruebas que se le deben realizar al software y a la vez que sean eliminadas.

RF5.1 Adicionar prueba

RF5.2 Modificar prueba

RF5.3 Eliminar prueba

Sistema de Gestión de No Conformidades

identificador	RF6. Adicionar No conformidades.
En este requisito funcional que debe permitir al Probador adicionar la NC a la aplicación la cual debe almacenar y proporcionar información importante pues servirá para posteriores estudios, además con esta adición el Responsable del equipo de Calidad ofrece su aprobación para que posteriormente el equipo de desarrollo la corrija o la modifique.	
identificador	RF7. Eliminar No Conformidades
Requisito funcional que debe permitir al probador la facilidad de eliminar una NC añadida ya sea porque el mismo la identificó y eliminó o el Responsable de Calidad dedujo que estaba incorrecta y se la reenvió al probador para que la arreglara o eliminara.	
identificador	RF8. Modificar No Conformidades
Requisito funcional que debe permitir al equipo de desarrollo, al responsable de calidad como al probador; de modificar las NC según el papel que juega su rol.	
identificador	RF9. Asignar No Conformidades
Requisito funcional que debe permitir al Responsable de Calidad distribuir el trabajo entre los diferentes integrantes del grupo de trabajo de calidad.	
identificador	RF10. Generar reportes
Requisito funcional que debe permitir informar el estado de las NC de los proyectos a sus respectivos líderes y clasificar las mismas por parámetros predefinidos.	
RF10.1 Mostrar reportes de Cantidad de No Conformidades por Fases en un determinado proyecto.	
RF10.2 Mostrar reportes de estado de las No Conformidades (resueltas, no resueltas).	
RF10.3 Mostrar reportes de cantidad de No Conformidades por usuario asignadas.	
RF10.4 Mostrar reportes de cantidad de No Conformidades por Flujo de trabajos en un determinado proyecto.	
RF10.5 Mostrar reportes de No Conformidades por prioridades del proyecto (alta, media, baja).	
Identificador	RF 11. Generar reportes comparativos entre proyectos (concluidos o en desarrollo) de las No Conformidades detectadas.
Requisito funcional que debe permitir después de un estudio detallado de diferentes proyectos	

determinar la media (cantidad numérica de No Conformidades que se detectan) por proyecto de acuerdo a parámetros predefinidos o no.

RF11.1 Mostrar media de No Conformidades por proyectos.

RF11.2 Mostrar media de No Conformidades según clasificadores.

2.3.2. Requerimientos no funcionales del sistema

Los requerimientos no funcionales son aspectos del sistema perceptibles para el usuario, que no se relacionan directamente con el comportamiento funcional del software.

Apariencia o interfaz externa:

- Diseño para la resolución 800x600, aunque podrá verse en otras resoluciones.
- El sistema informático debe poseer una interfaz web amigable y sencilla, fácil para la interacción del usuario, sin saturación de colores ni imágenes.

Usabilidad

- El sistema podrá ser usado por cualquier persona, que posea conocimientos básicos en el manejo de la computadora.

Soporte:

- Versión de PHP 5.0 o superior.
- Se requiere la instalación del navegador Mozilla Firefox 3.5.
- Se requiere la instalación del Gestor de Base de Datos PostgreSQL en su versión 8.3 y el Servidor Web Apache.

Portabilidad

- El software podrá ser usado bajo los sistemas operativos Windows y Linux.

Seguridad

- Establecer permisos para garantizar que solo se acceda a la información quien tenga permiso.
- Garantizar la protección ante acciones no autorizadas.
 - Verificación sobre acciones irreversibles (eliminación, modificación).

Confiabilidad

- Garantizar un tratamiento adecuado de excepciones y validaciones de las entradas de los usuarios.
- Garantizar la recuperación del sistema ante fallos y errores.

2.4. Modelo de casos de usos del sistema

Se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso, los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

2.4.1. Actores del sistema.

Un actor representa un rol que juega una o varias personas, un equipo o un sistema automatizado, no son parte del sistema y pueden ser un recipiente pasivo de información.

Actores	Justificación
Responsable de Calidad	Máximo responsable, es el encargado de definir las pruebas a realizar, aprobar las NC y de elaborar el informe de las NC que se le presenta al Líder de Proyecto. Puede mostrar las NC detectadas y modificarlas.
Probador	Encargado de realizar las pruebas y registrar las NC detectadas. Puede Modificar las NC.
Líder de proyecto	Define los integrantes y sus roles en el proyecto. Puede generar reportes para revisar el estado de su proyecto así como realizar comparaciones con otros.
Equipo de Desarrollo	Encargado de corregir y modificar las NC encontradas por el probador
Administrador	Máximo responsable de gestionar a los administradores del sistema que van a poder realizar diferentes actividades en cuanto a su proyecto además gestiona los proyectos a los cuales los líderes le podrán realizar diferentes acciones.

Tabla2.3 Actores del Sistema

2.4.2. Listado de casos de uso.

CU-1	Autenticar Usuario
Actor	Usuarios del sistema
Descripción	Se autentican los usuarios del sistema en la aplicación.
Referencia	RF1
CU-2	Gestionar Líder de Proyecto

Sistema de Gestión de No Conformidades

Actor	Administrador
Descripción	Se gestionan los datos de los diferentes Líderes de proyecto.
Referencia	RF2
CU-3	Gestionar Usuarios del Proyecto
Actor	Líder del Proyecto
Descripción	Se gestionan los usuarios que desempeñarán diferentes roles dentro de la aplicación.
Referencia	RF3
CU-4	Gestionar Proyecto
Actor	Administrador
Descripción	Se gestionan los proyectos que formarán parte del entorno de la aplicación.
Referencia	RF4
CU-5	Gestionar Elementos de prueba
Actor	Responsable de Calidad
Descripción	Se gestionan las pruebas que se realizarán las cuales ayudarán a detectar las NC en el proyecto.
Referencia	RF5
CU-6	Adicionar No Conformidad
Actor	Probador
Descripción	El probador una vez realizadas las pruebas a la aplicación adiciona la NC detectadas,
Referencia	RF6
CU-7	Eliminar No Conformidad
Actor	Probador
Descripción	El probador Elimina la NC, ya sea porque el Responsable de Calidad se la reasignó para que la reevaluara o el mismo dedujeran que no representaba una.
Referencia	RF7

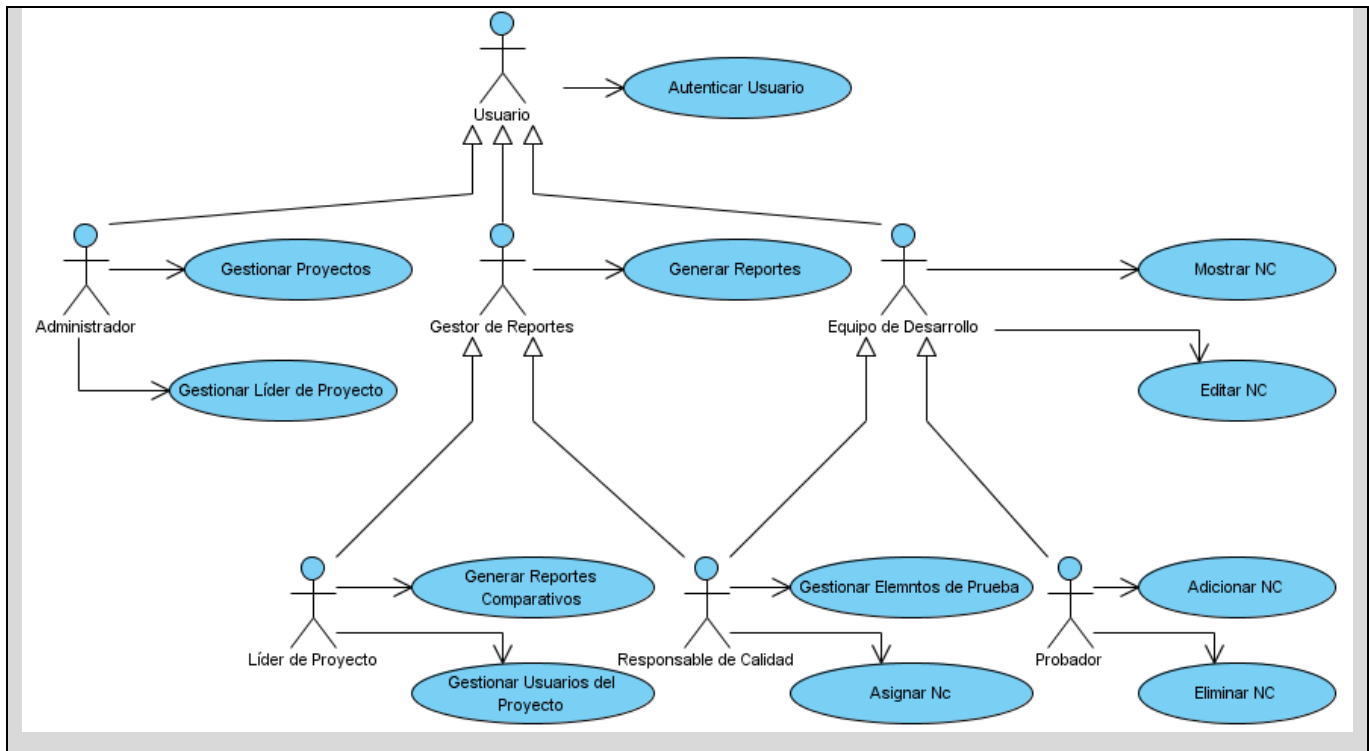
Sistema de Gestión de No Conformidades

CU-8	Modificar No Conformidad
Actor	Probador, Responsable de Calidad, Equipo de Desarrollo.
Descripción	El probador Modifica la NC, ya sea porque el Responsable de Calidad se la reasignó para que la reevaluara o el mismo dedujeran que requería una modificación.
Referencia	RF8
CU-9	Asignar NC
Actor	Responsable de Calidad
Descripción	Una vez detectadas y registradas las NC, el Responsable de Calidad es el encargado de asignar las mismas, ya sea para que el probador las vuelva a revisar o para que el equipo de desarrollo las elimine.
Referencia	RF9
CU-10	Generar reportes
Actor	Líder de proyecto
Descripción	Se generan todos los reportes de NC con respecto al proyecto.
Referencia	RF10
CU-11	Generar reportes comparativos entre proyectos de las No Conformidades detectadas
Actor	Líder de proyecto
Descripción	Se generan reportes de otros proyectos que se basan en informaciones que tienen que ver con datos que persisten los cuales brindarán al líder una perspectiva de en que nivel está su proyecto en la actualidad.
Referencia	RF11

2.4.3 Diagrama de casos de uso del sistema.

Diagrama de Casos de Uso

Sistema de Gestión de No Conformidades



Una vez definidos los casos de uso del sistema se describe textualmente los mismos, se explica de forma detallada el funcionamiento y el desarrollo de los casos de uso, así como sus interrelaciones entre ellos. Por la existencia de doce casos de uso, se incluye en este capítulo sólo la descripción textual del caso de uso “Generar Reportes” pues trata de forma central el problema a resolver que contiene el trabajo de diploma, el resto de las descripciones textuales se encuentran en la sección de anexos del documento, desde Anexos del Capítulo 2 “Descripción textual de los casos de uso del sistema” hasta Anexos del Capítulo 3 “Diagramas de Clases del Análisis”.

2.4.4. Especificación textual del Caso de Uso “Generar Reportes”

Caso de Uso	Generar Reportes
Actores	Gestor de Reportes
Propósito	Permitir al Gestor de Reportes mostrar diferentes reportes de NC que permitan tener una visión del estado del proyecto en cuanto a las No Conformidades.
Resumen	El Gestor de Reportes muestra diferentes reportes.
CU asociados	
Precondiciones	El sistema contiene NC insertadas, modificadas y revisadas.

Sistema de Gestión de No Conformidades

Poscondiciones	El sistema muestra los reportes generados.	
Referencias	RF11	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona del menú la opción Reportes.	2. El sistema muestra interfaz con la opción de los tipos de Reportes a Generar.	
3. El usuario selecciona la opción NC por fases. -Si el actor selecciona la opción NC por Estado (Ver Sección I) -Si el actor selecciona la opción NC por Usuarios. (Ver Sección II) -Si el actor selecciona la opción NC por flujos de Trabajo. (Ver Sección III) -Si el actor selecciona la opción NC por Prioridad. (Ver Sección IV)	4. El sistema chequea la existencia de NC según el reporte seleccionado.	
	5. El sistema muestra interfaz del reporte seleccionado.	
	6. El usuario presiona el botón Aceptar.	
	7. Va la línea 2.	
Flujos Alternos		
3.1 El usuario presiona el botón Salir.	3.2 Se muestra interfaz inicial del sistema.	
5.2 El usuario presiona el botón Aceptar.	5.1 El sistema muestra un mensaje "No existe elementos suficientes para generar este reporte."	
	5.3 Va la línea 2.	
Sección I: NC por Estado		
Acción del Actor	Respuesta del sistema	
1. El usuario selecciona la opción NC por Estado.	2. El sistema muestra un formulario con los datos a ingresar.	
3. El usuario selecciona el estado.	5. El sistema chequea la existencia de NC según el reporte seleccionado.	
	6. El sistema muestra el reporte seleccionado.	

Sistema de Gestión de No Conformidades

4. El usuario presiona el botón Aceptar .	7. Va la línea 1.
Flujos Alternos	
1.1. El usuario presiona el botón Aceptar .	1.2 Se muestra un mensaje "Debe seleccionar un estado".
1.1.1. El usuario presiona el botón Cancelar .	1.1.2. El sistema muestra la pagina principal.
	6.1 El sistema no muestra el reporte.
	6.2 Va a la línea 1.
Sección II: NC por Usuarios	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción NC por Usuarios .	2. El sistema muestra un formulario con los datos a ingresar. 5. El sistema chequea la existencia de NC según el reporte seleccionado.
3. El usuario introduce los datos.	6. El sistema muestra una interfaz con el reporte seleccionado.
4. El usuario presiona el botón Aceptar .	7. Va a la línea 1.
Flujos Alternos	
1.1: El actor presiona el botón Aceptar .	1.2 El sistema muestra un mensaje" Este campo es obligatorio".
1.1.1. El usuario presiona el botón Cancelar .	1.1.2. El sistema muestra la pagina principal.
	6.1 El sistema no muestra el reporte.
	6.2 Va a la línea 1.
Sección III: Mostrar Reportes de Cantidad de NC por flujos de trabajo de un determinado Proyecto	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Mostrar Reportes de Cantidad de NC por flujos de trabajo de un determinado Proyecto .	2. El sistema chequea la existencia de NC según el reporte seleccionado.

Sistema de Gestión de No Conformidades

	3. El sistema muestra una interfaz con el reporte seleccionado.
4. El actor da clic en el botón salir .	5. El sistema cierra la interfaz.
Flujos Alternos	
1.1: El actor presiona el botón Salir .	1.2 El sistema cierra la interfaz.
	3.1 El sistema muestra un mensaje "No existe este tipo reporte."
3.2 El actor presiona el botón Aceptar .	3.3 Va la línea 2.
Sección IV: Mostrar Reportes de NC por Prioridades de un Proyecto	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Mostrar Reportes de NC por Prioridades de un Proyecto .	2. El sistema chequea la existencia de NC según el reporte seleccionado.
	3. El sistema muestra una interfaz con el reporte seleccionado.
4. El actor da clic en el botón salir .	5. El sistema cierra la interfaz.
Flujos Alternos	
1.1: El actor presiona el botón Salir .	1.2 El sistema cierra la interfaz.
3.2 El actor presiona el botón Aceptar .	3.1 El sistema muestra un mensaje "No existe este tipo reporte."
	3.3 Va la línea 2.

Conclusiones.

En este capítulo se inició el desarrollo de la solución propuesta para resolver la situación problemática que fue presentada en el capítulo anterior. A partir del análisis del negocio se lograron definir las funcionalidades que debe presentar el sistema, se precisaron los actores con los cuales se relacionan. Fueron descritas las actividades que se realizan en el negocio y las acciones que se ejecutarán en el sistema que permitirá automatizar los procesos, por otro lado se definieron los casos de uso del sistema y la descripción textual de los mismos.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En este capítulo se representa el análisis y el diseño que proporcionará la comprensión del sistema que se implementará, también se recogen los diagramas correspondientes que contribuirán a la posterior implementación del proyecto.

3.2. Análisis

El modelo de análisis ayuda a refinar y a estructurar los requisitos, permite entender mejor los aspectos internos del Sistema, ofrece un mayor poder expresivo y una mayor formalización a través de sus diagramas.

3.2.1. Diagramas de clases del análisis

Los diagramas de clases están compuestos por diferentes clases: interfaz, controladoras y entidad, mediante estos diagramas se conoce como queda el flujo de la información, así como la distribución de las interfaces y la relación entre ellas. Igual que el capítulo anterior se adjuntará sólo el diagrama de clases del análisis del caso de uso “Generar Reportes”, el resto de los diagramas se encuentran en los Anexos del trabajo desde Anexos del Capítulo 3 “Diagrama de clases del análisis;” hasta Anexos del Capítulo 3 “Diagramas de Clases del Diseño.”

Diagrama de Clases del Análisis

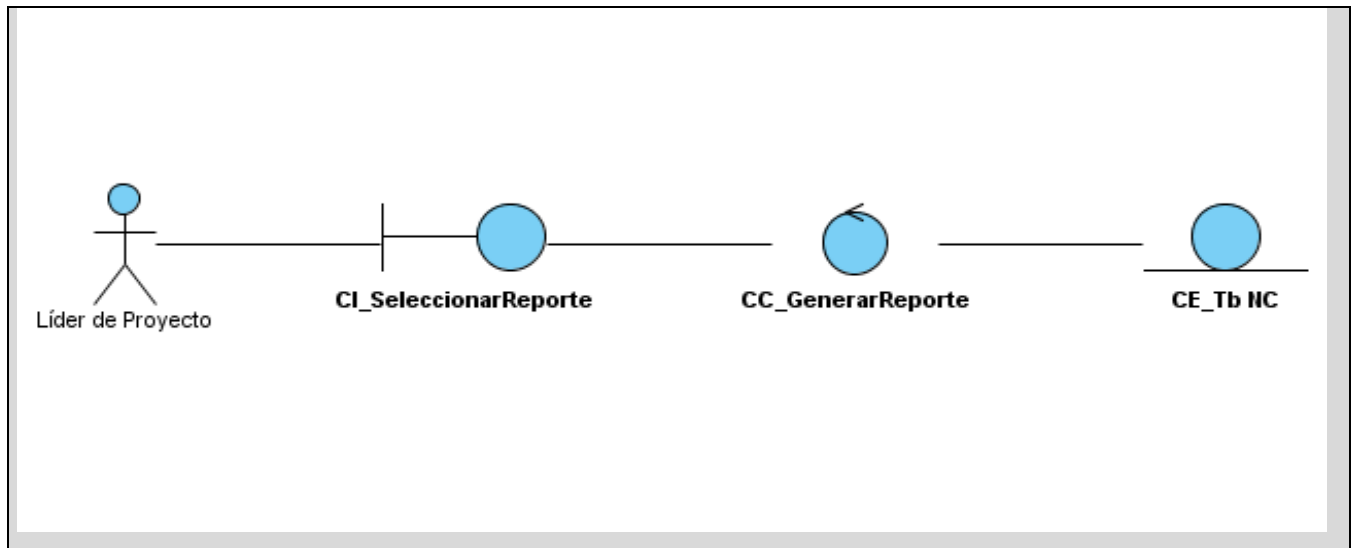


Figura 3.1 Diagrama de clases del análisis “Generar Reportes”

3.2.2. Diagramas de colaboración del Análisis

Existen dos tipos de diagramas de interacción los de secuencia y los diagramas de colaboración los cuales muestran interacciones organizadas alrededor de los roles. A continuación se presentan los diagramas de colaboración de los casos de uso que se identificaron en el sistema. Continuando con el ejemplo ilustrativo de los epígrafes anteriores se adjunta el diagrama de colaboración del caso de uso “Generar Reportes”, se adjuntaron solamente los Adicionar de todos los CU del sistema por la extensión de los mismos, en caso del CU ilustrativo que no posee una función Adicionar se adjunta el generar reportes. El resto de los diagramas de este tipo se encuentran en los Anexos del trabajo desde Anexos del Capítulo 3 “Diagramas de colaboración “hasta Anexos del Capítulo 4 “Tablas de Descripción de las clases persistentes.”

“Diagrama de colaboración.”

Diagrama de Colaboración del Análisis

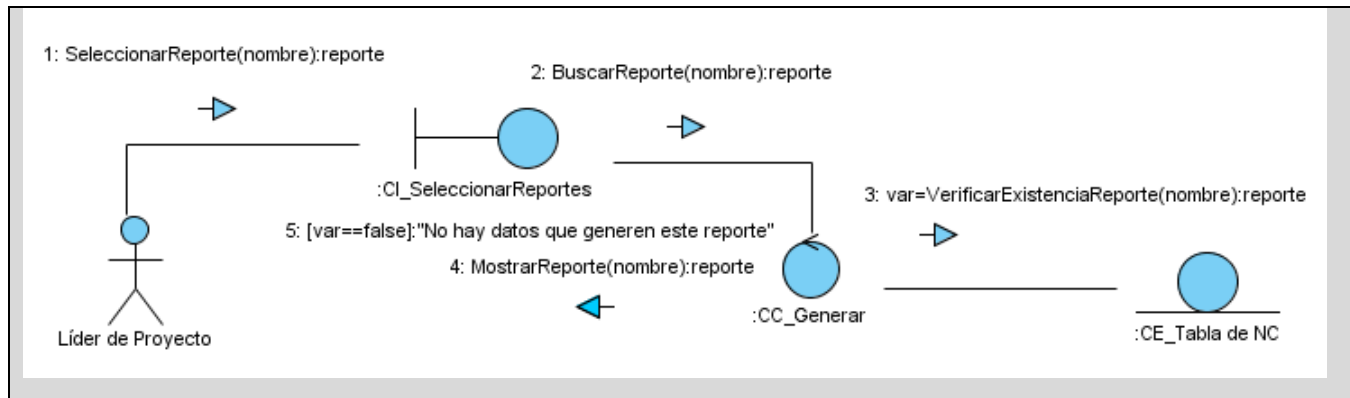


Figura 3.2 Diagrama de colaboración del Análisis “Generar Reportes”

3.3. Diseño

El modelo de diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución, preparando la implementación y prueba del sistema. Pretende crear un plano del modelo de implementación, por lo que el grueso del esfuerzo está en las últimas iteraciones de elaboración y las primeras de construcción.

3.3.1. Estructura Propuesta

Durante la etapa de diseño del sistema se definieron tres módulos para darle respuesta a los requerimientos del sistema.

- ❖ **Administración:** en el siguiente módulo existe un administrador general el cual gestiona los proyectos que estarán presentes en la aplicación; además se gestionan los administradores de sistema que representa a los líderes de proyecto, el cual será el encargado de gestionar los usuarios que formarán parte del mismo. Cada trabajador por cuestiones de seguridad se le asignó por su respectivo rol solo las funcionalidades que debe realizar.
- ❖ **Reportes:** en el siguiente módulo los líderes de proyecto y Responsables de calidad generan diferentes reportes sobre funcionalidades internas del proyecto así como de datos estadísticos que brinden comparaciones con otros proyectos. Cada reporte representa crear mejorías que ayuden tanto al perfeccionamiento del proyecto como de sus trabajadores.
- ❖ **No Conformidades:** en el siguiente módulo el Responsable de calidad, los probadores y el equipo de desarrollo realizan procesos que conllevan a la detección y corrección de las NC que brindarán la información necesaria para la construcción de los reportes.



Figura 3.4 Módulos Existentes

3.3.2. Diagrama de clases del diseño web

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas. En las aplicaciones Web, éste representa la colaboración entre las páginas donde cada página lógica puede ser representada como una clase. En esta sección se adjunta el diagrama de clases del diseño web del caso de uso “Generar Reportes”, el resto de los diagramas de este tipo se encuentran en los Anexos del trabajo desde Anexos del Capítulo 3 “Diagrama de clases del diseño web;” hasta Anexos del Capítulo 3 “Diagramas de colaboración.”

Diagrama de clases del diseño web

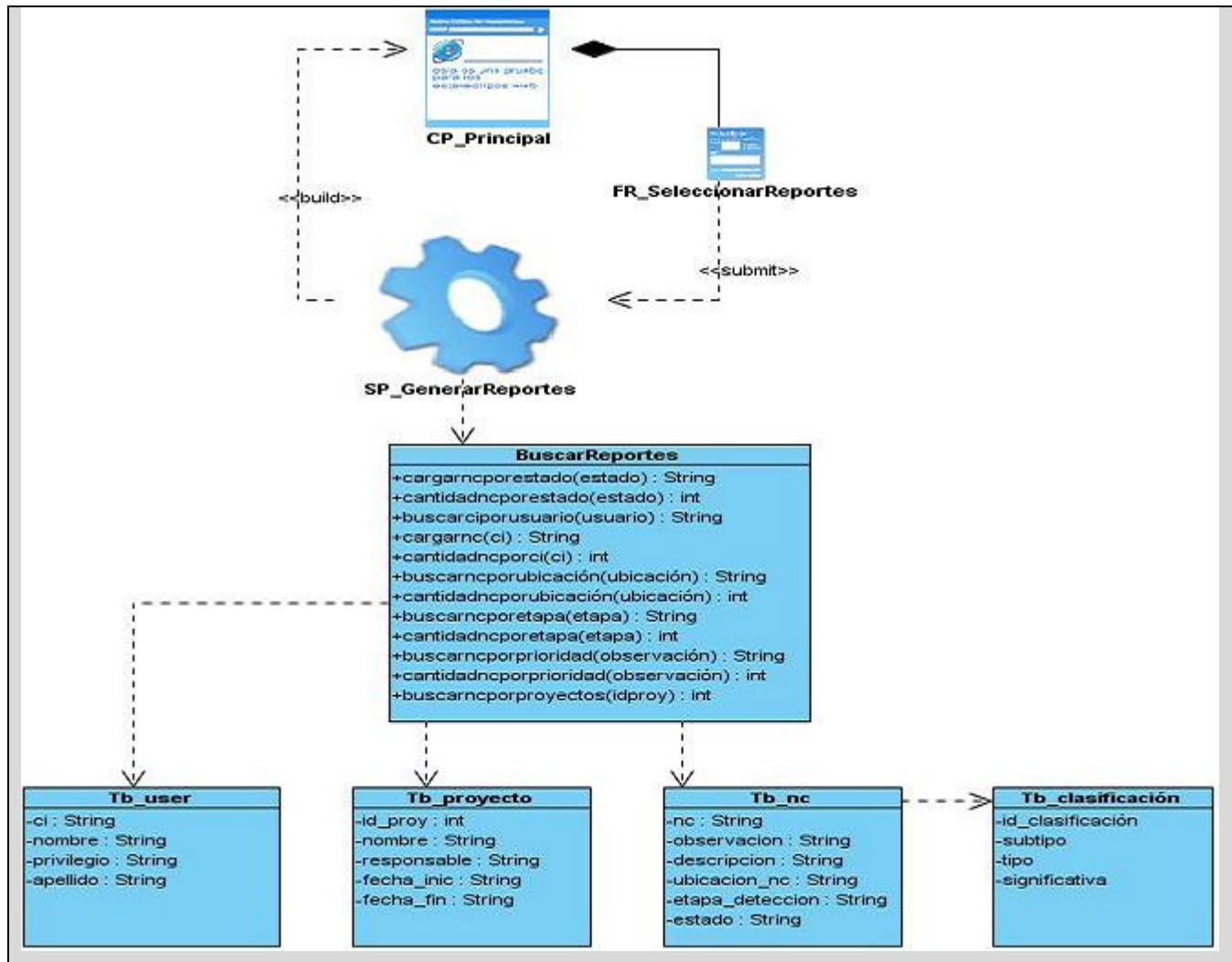


Figura 3.3 Diagrama de clases del Diseño “Generar Reportes”

3.3.3. Diagrama de clases persistentes

Los diagramas de clases contienen las clases que tendrá la base de datos del sistema, los atributos de cada clase y sus relaciones así como la cardinalidad que existe entre ellas.

Diagrama de Clases Persistentes

Sistema de Gestión de No Conformidades

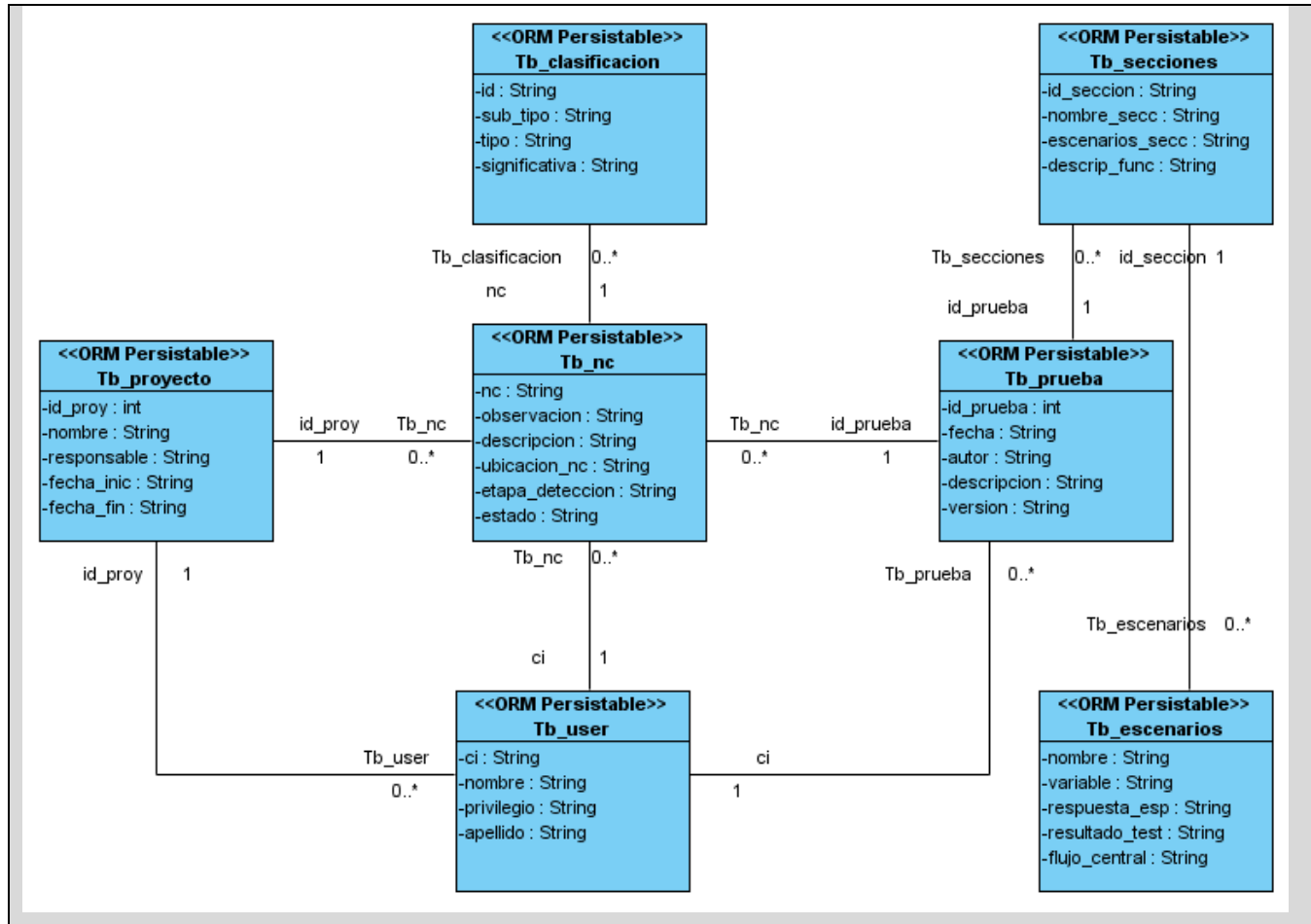


Figura 3.3 Diagrama de Clases Persistentes

3.3.4. Descripción de las tablas de la base de datos.

Las diferentes tablas que describen las clases persistentes se encuentran en los Anexos del Capítulo 4 “Tablas de Descripción de las clases persistentes.”

Nombre: Tb_nc		
Descripción: Se almacenan los datos correspondientes a las NC detectadas.		
id_nc	String	Es la llave primaria y almacena el identificador de la nc.
observación	String	Describe de forma detallada la prioridad de la nc
descripción	String	Describe de forma sencilla en que consiste la nc.
ubicación_nc	String	Describe detalladamente el flujo de trabajo en el que fue detectada la nc.

Sistema de Gestión de No Conformidades

etapa_detección	String	Describe detalladamente en que fase fue detectada dicha nc.
estado	String	Describe en que estado se encuentra la nc.
id_proy	int	Es la llave foránea y almacena el identificador del proyecto en que se detectó la nc.
ci	String	Es la llave foránea y almacena el carnet de identidad del usuario que detectó la nc.
id_prueba	int	Es la llave foránea y almacena el id de la prueba que detectó la nc.

Tabla3.1. Descripción de las tablas de la base de datos “Tb_ nc“

Conclusiones

Durante el desarrollo de este capítulo se realizó el proceso de Modelamiento de las fases de Análisis y Diseño, se generaron los Diagramas de clases del análisis, los Diagramas de colaboración y el Diagrama de clases persistentes. Una vez elaborados estos artefactos, se concluye el capítulo con un enfoque más detallado de lo que será el sistema que se llevará a cabo en la implementación.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

Una vez desarrollada la fase de Análisis y Diseño le sigue a continuación la de Implementación del sistema. Dentro de esta existen artefactos que toman elementos de entrada desarrollados en la fase anterior, siguiendo el objetivo de conformar un sistema completo que cumplan estrictamente los requerimientos definidos.

4.2. Implementación

La implementación se comienza con el resultado del análisis y diseño, y se implementa el sistema en términos de componentes, es decir: ficheros de código fuente, scripts, ficheros de códigos binarios y ejecutables. Dentro de los principales objetivos se encuentran: Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue e implementar las clases y subsistemas encontrados en el diseño, entre otros.

4.2.1. Diagrama de Modelo de datos

Modelo de datos

Sistema de Gestión de No Conformidades

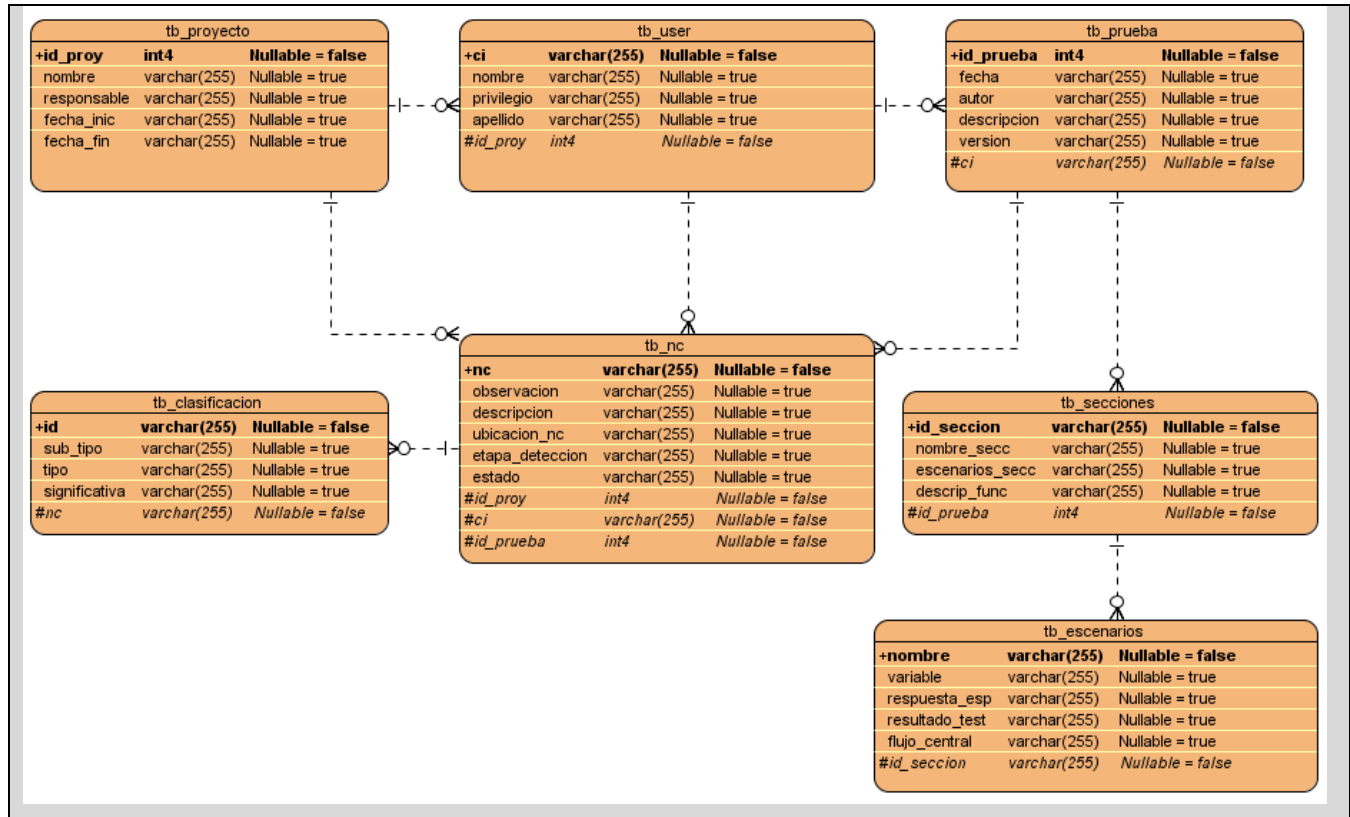


Figura 4.1 Diagrama de Modelo de Datos

4.2.2. Modelo de despliegue

El diagrama de despliegue representa los elementos físicos con que contará el sistema, los componentes de este diagrama son los nodos y las relaciones entre ellos, estas relaciones pueden ser de diferentes tipos dependiendo de lo que represente el nodo.

Modelo de despliegue

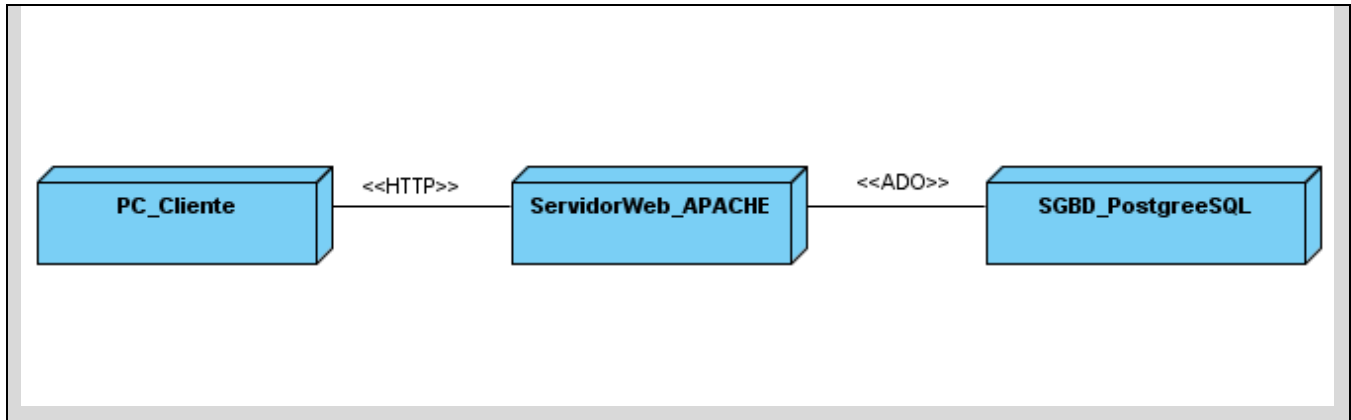
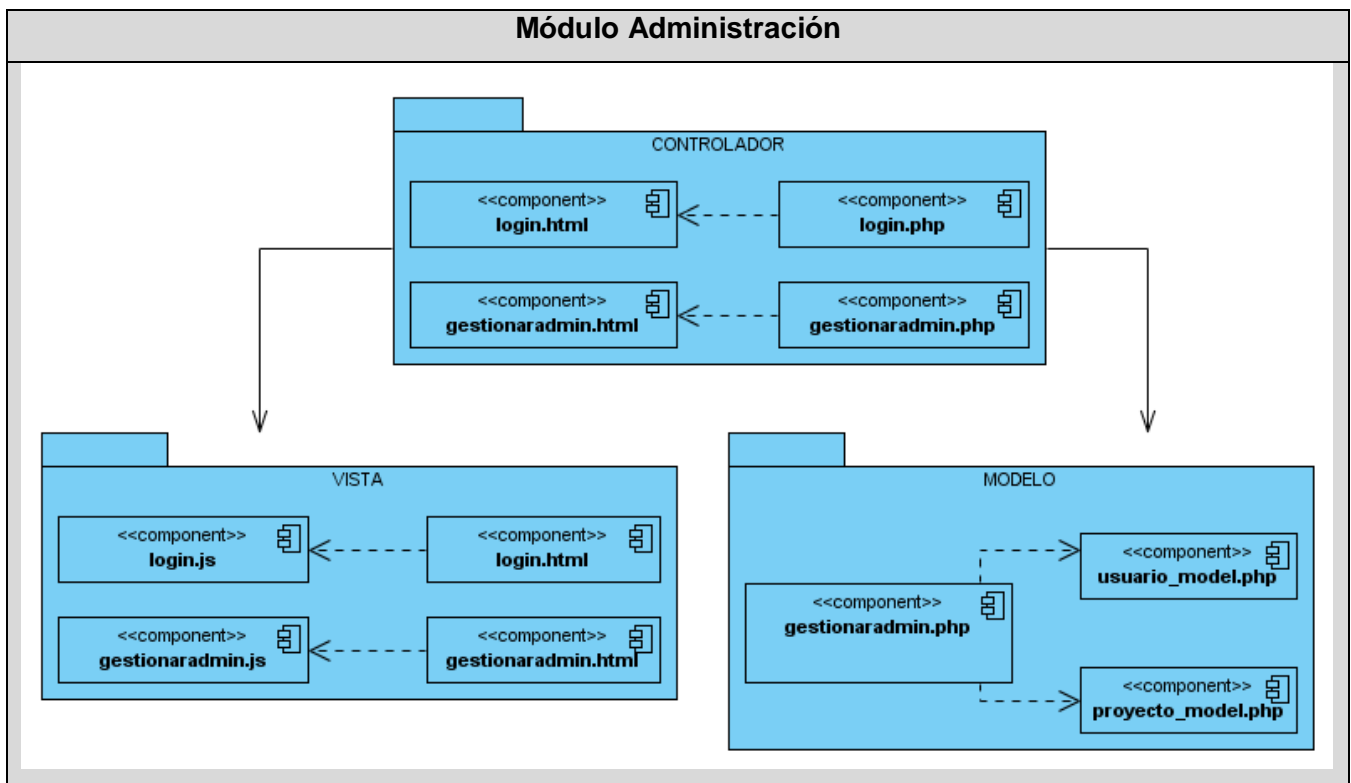


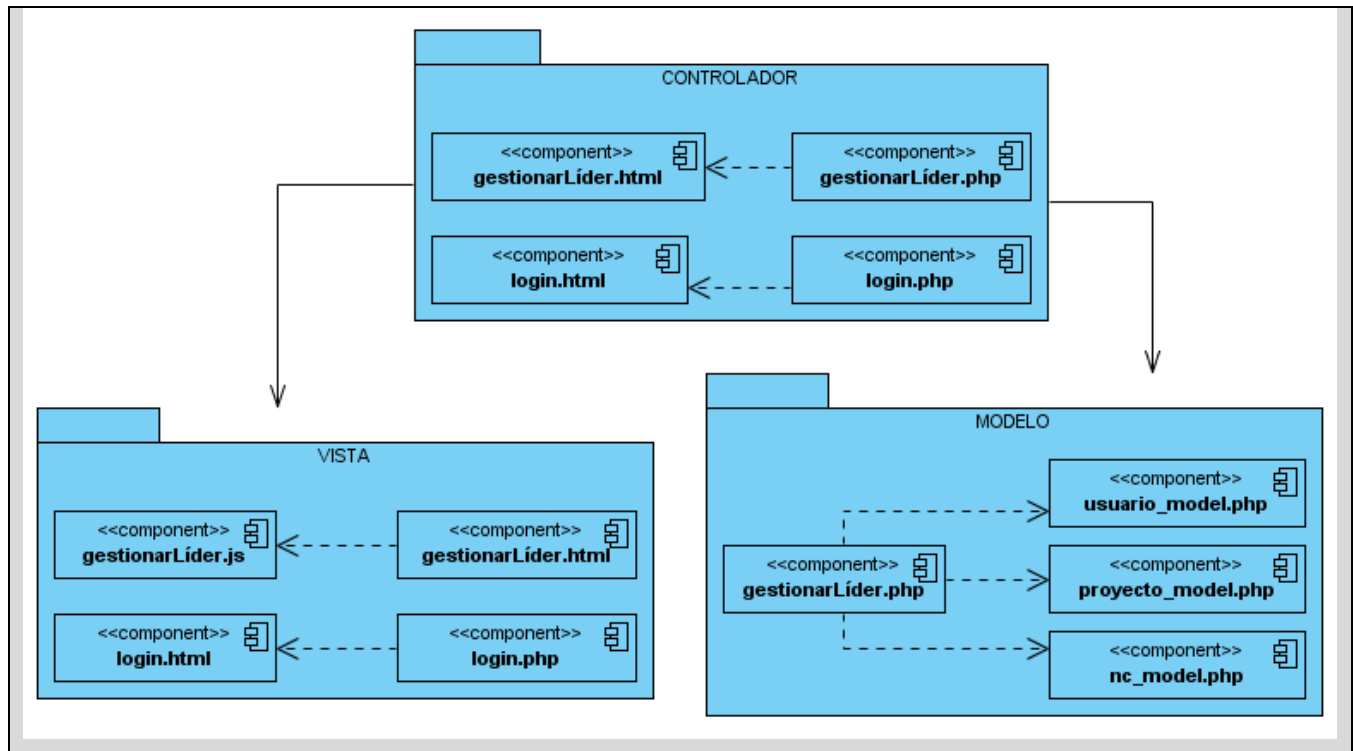
Figura 4.2 Diagrama de Modelo de Despliegue

4.2.3. Diagrama de componentes



4.2.4. Diagrama de componentes

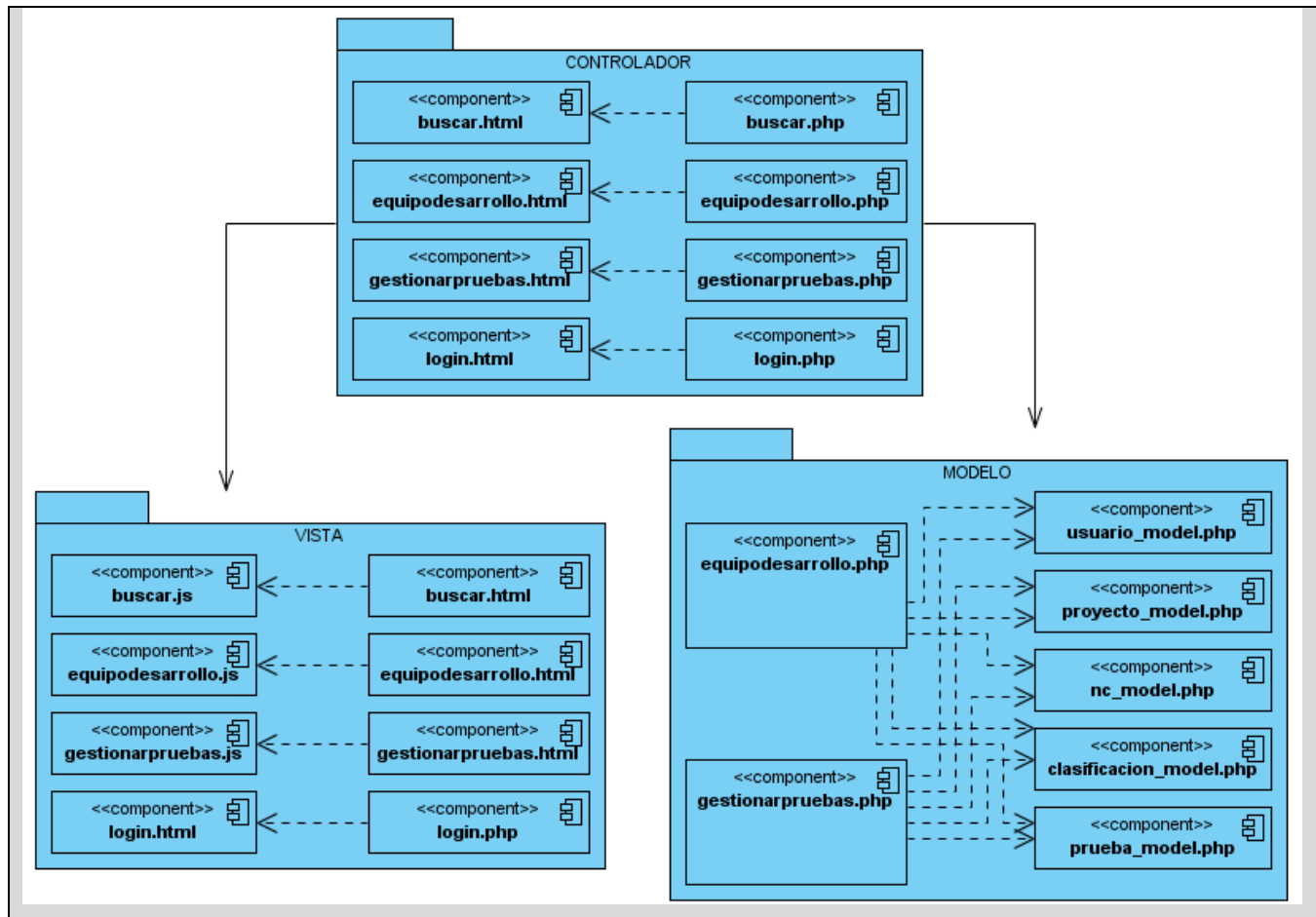




4.2.5. Diagrama de componentes

Módulo NC

Sistema de Gestión de No Conformidades



4.3. Prueba

Existen actualmente diferentes tipos de pruebas, su función es encontrar la mayor cantidad de defectos para ofrecer un producto con la mayor calidad posible. Entre los principales métodos de prueba se encuentran, pruebas de **Caja Blanca** y de **Caja Negra**, las primeras se centran en la revisión del código fuente del software y requieren conocimiento del funcionamiento interno del sistema. Las últimas por su parte se refieren a las pruebas que se realizan a la interfaz del sistema, son conocidas también como pruebas de comportamiento y se basan en los requerimientos funcionales del sistema.

Para comprobar el **Sistema de Gestión de No conformidades** se efectuaron las siguientes pruebas, verificando todos los requisitos funcionales del sistema:

- ✓ **Pruebas funcionales:** Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Los analistas de pruebas, no enfocan su atención a

como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida.

- ✓ **Pruebas de validación:** Las pruebas de validación en la ingeniería de software son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido. La validación es el proceso de comprobar lo que se ha especificado es lo que el usuario realmente quería. Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.
- ✓ **Pruebas de integración:** Las pruebas de integración (algunas veces llamadas integración y testeo) es la fase del testeo de software en la cual módulos individuales de software son combinados y testeados como un grupo. Son las pruebas posteriores a las pruebas unitarias y preceden el testeo del sistema. Son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

4.3.1. Diseños de caso de prueba

Los diseños de casos de prueba es un método de Caja Negra para probar el sistema, se elaboran por CU o por requisitos su objetivo principal es proveer una guía de cómo realizar la prueba, mostrando los pasos a seguir y los datos a introducir con la finalidad de obtener el resultado esperado. Se adjunta una breve descripción del DCP utilizado durante las pruebas de aceptación.

Diseño de Pruebas

Pruebas de funcionalidad.	
Pre-requisitos	La aplicación debe tener NC añadidas y funcionando correctamente.
Propósito	Comprobar que la funcionalidad responde correctamente según se diseñó.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar la funcionalidad. Se debe seleccionar la acción a ejecutar insertándole valores que permitan evaluar su correcto funcionamiento.
Artefacto Generado	Informe de la evaluación de la funcionalidad.
Resultado esperado	El sistema responda correctamente a los elementos y las funcionalidades

Sistema de Gestión de No Conformidades

	ejecutadas.
Resultado de la prueba	Será evaluada en satisfactorio o no satisfactorio; a determinar por el Responsable de Calidad del proyecto.

Tabla4.1.Diseño de Caso de Pruebas.

4.3.2 Pruebas realizadas

#Iteración 1					
Tipo de Prueba	Positivas	Negativas	Total	%Pruebas Negativas	Fecha
Pruebas funcionales	4	13	17	%76	20/5/2010
Pruebas de validación	2	10	12	%77	21/5/2010
Pruebas de integración	0	4	4	%100	21/5/2010
Observaciones: La iteración presentó problemas con respecto a la introducción de datos puesto que se definió una base de datos con campos indebidos que crearon problemas al gestionar los usuarios.					

Tabla4.2.Descripción de las pruebas realizadas en la primera iteración.

#Iteración 2					
Tipo de Prueba	Positivas	Negativas	Total	%Pruebas Negativas	Fecha
Pruebas funcionales	17	3	20	%85	12/6/2010
Pruebas de validación	9	0	9	0	13/6/2010
Pruebas de integración	3	0	3	0	13/6/2010
Observaciones: La iteración presentó problemas en la validación de algunos campos de la aplicación creando problemas al insertar diferentes datos incorrectos en la base de datos.					

Tabla4.3.Descripción de las pruebas realizadas en la segunda iteración.

Conclusiones

Una vez realizado el capítulo se desarrollaron las fases de Implementación y Prueba con la realización de los artefactos correspondientes de cada una, definiéndose el ambiente de implantación para la aplicación y se efectuándose los DCP a aplicar en las pruebas piloto al sistema, las cuales resultaron placenteras, teniendo buena aceptación por los usuarios finales.

CONCLUSIONES DEL TRABAJO

Una vez realizado el estudio del estado del arte referente a los diferentes sistemas de gestión de No Conformidades existentes en Cuba y en el mundo, se llevó a cabo la implementación de una aplicación web la cual tiene como objetivo la disminución de la repetición de las mismas No Conformidades. Para ello se realizó también:

- Un análisis de las principales tecnologías para el desarrollo de aplicaciones web que permitió seleccionar las herramientas y metodologías para desarrollar el sistema.
- Un estudio de los procesos existentes en diferentes proyectos productivos que propició amplio conocimiento del negocio y de los procesos a informatizar.
- Una investigación de los sistemas actuales y en específico las clasificaciones que obvian y que son parte importante del proceso de gestión de las NC.
- El desarrollo del análisis, el diseño y la implementación de una aplicación web que permite la gestión de las No conformidades que se detectan en las diferentes fases de elaboración de un proyecto.
- El desarrollo de una estructura lo suficientemente segura donde almacenar las NC registradas de forma que perduren para que en el futuro se puedan realizar estudios de las mismas.
- La aprobación de la solución propuesta por parte del cliente, proceso en el cual surgieron una serie de recomendaciones para propiciar el perfeccionamiento futuro del sistema.

De acuerdo a las consideraciones expuestas anteriormente es posible afirmar que se alcanzó de manera satisfactoria el objetivo propuesto: informatizar los procesos asociados a la gestión de No Conformidades y generación de reportes.

RECOMENDACIONES

A modo de recomendaciones se añaden una serie de observaciones a tener en cuenta para el futuro perfeccionamiento del sistema:

- Integrar elementos que permitan evaluar las NC con Estándares de Calidad.
- Integrar funciones que permitan evaluar automáticamente Métricas.
- Integrar a la aplicación el diseño de las pruebas a realizar, para lograr una mayor independencia de factores externos.
- Exportar los reportes generados a archivos pdf, doc, txt, etc.

Sistema de Gestión de No Conformidades

- Agregar comparaciones que permitan evaluar según las clasificaciones de las NC de un mismo proyecto.
- Permitir gestionar equipos de calidad rotativos que puedan trabajar en diferentes proyectos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Modelo Cliente Servidor [Disponible en: <<http://www.deepinphp.com/2007/06/17/el-modelo-cliente-servidor>> [Fecha de consulta: 15 enero 2009]
- [2] Definición arquitectura cliente servidor [Disponible en: <<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>> [Fecha de consulta: 15 enero 2009]
- [3] Lenguajes de lado servidor [Disponible en: <<http://www.desarrolloweb.com/articulos/243.php>> [Fecha de consulta: 18 enero 2009]
- [4] Programación Web: PHP + Base de Datos [Disponible en: <http://www.ecicomputacion.com.ar/php___bases_de_datos.html> [Fecha de consulta: 20 enero 2009]
- [5] Ingeniería de Software I y II [Disponible en: Universidad de las Ciencias Informáticas. Entorno Virtual de Aprendizaje. eva.uci.cu]. [Fecha de consulta: 21 enero 2010]
- [6] Internet Information Services [Disponible en: <<http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>> [Fecha de consulta: 26 enero 2010]
- [7] Estudio de diseño web y diseño gráfico [Disponible en: <http://www.ideocentro.com/recursos.php?id=13&estudio_diseno_web_ideocentro=MySQL> [Fecha de consulta: 30 enero 2010]
- [8] PHP [Disponible en: <<http://www.softwarelibre.venezuela.net.ve/index.php>> [Fecha de consulta: 18 marzo 2009]
- [9] Sección de paradigma visual [Disponible en: <http://www.freedownloadmanager.org/es/downloads/paradigma_visual_libre/> [Fecha de consulta: 21 marzo 2010]

Bibliografías Consultadas

1. [En línea] [Citado el: 12 de febrero de 2010.]<http://www.calidaddelsoftware.com/>

2. [En línea] [Citado el: 14 de febrero de 2010.] <http://www.sergiovillagra.com/Contenidos/recursos/Calidad%20en%20el%20Desarrollo%20de%20Software-CMMI.PDF>
3. [En línea] [Citado el: 16 de febrero de 2010.] www.kinexo.com/qms
4. [En línea] [Citado el: 18 de febrero de 2010.] http://www.tcpsi.com/servicios/ciclo_vida.htm
5. [En línea] [Citado el: 22 de febrero de 2010.] <http://www.eumed.net/libros/2007c/306/METODOS%20GENERALES%20DE%20LA%20INVESTIGACION%20CIENTIFICA.htm>
6. [En línea] [Citado el: 24 de febrero de 2010.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>
7. [En línea] [Citado el: 24 de febrero de 2010.] <http://service-desktop.softonic.com/>
8. KMKey. [En línea] <http://www.kmkey.com/>.
9. [En línea] [Citado el: 25 de febrero de 2010.] http://10.128.50.74/login?back_url=http
10. [En línea] [Citado el: 27 de febrero de 2010.] <http://www.hotfrog.es/Empresas/IGQuality>
11. [En línea] [Citado el: 28 de febrero de 2010.] <http://10.36.16.150:5800/projects/show/sigep>
12. [En línea] [Citado el: 5 de marzo de 2010.] http://www.kmkey.com/productos/software_gestion_calidad
13. [En línea] [Citado el: 7 de marzo de 2010.] www.oc.ccn.cni.es/PO-002-noconformidades_es.html.
14. [En línea] [Citado el: 10 de marzo de 2010.] [http:// http://extjs.com/](http://http://extjs.com/).

ANEXOS

Anexos del Capítulo 1

Anexo 1.1

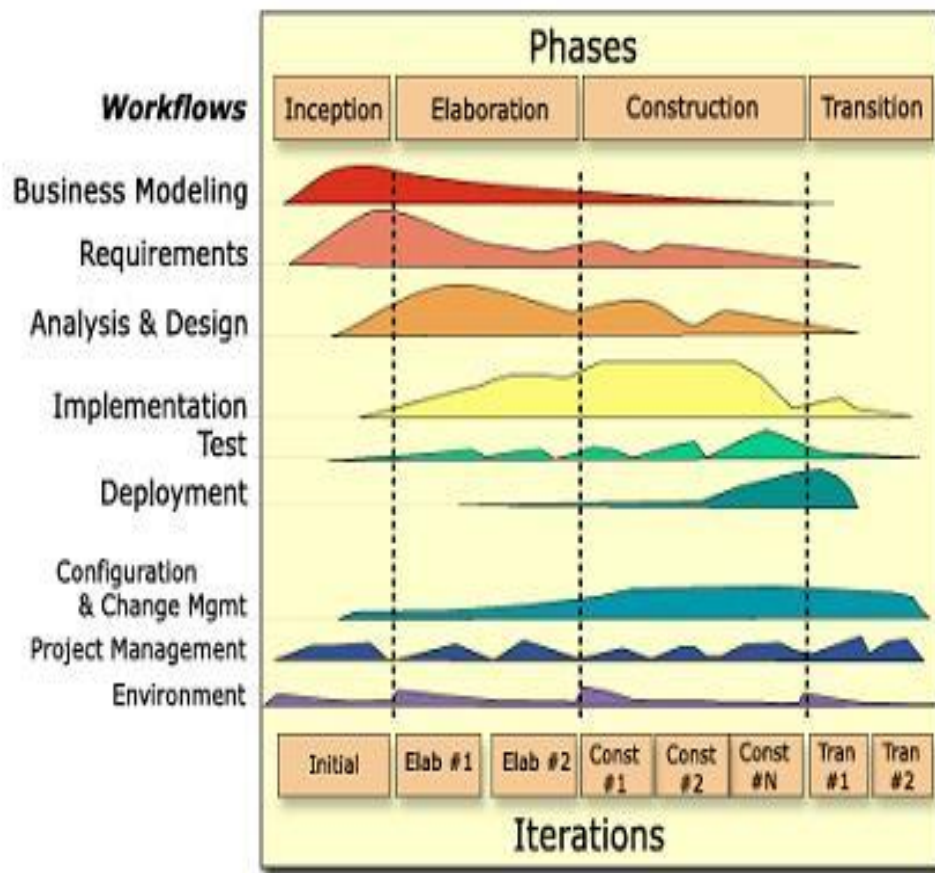


Figura1: RUP
Anexo 1.2

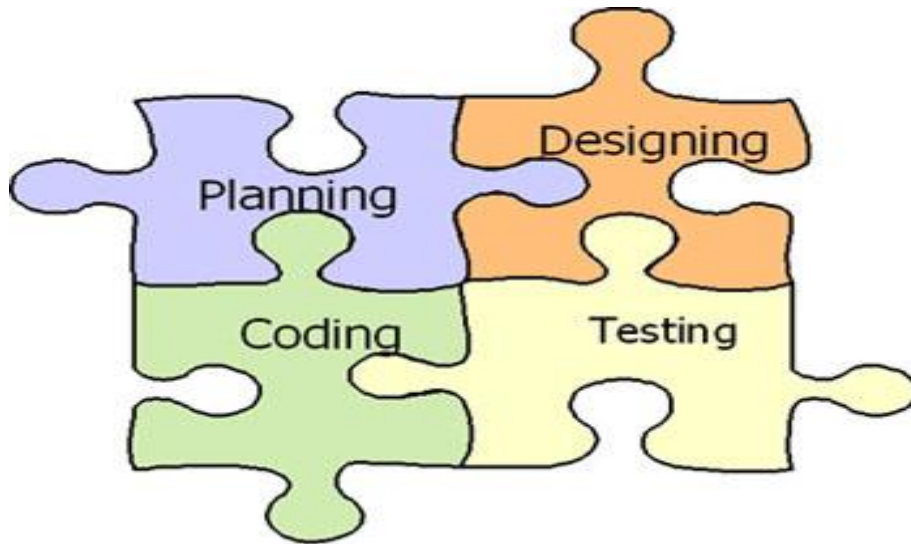


Figura 2: Metodología Extreme Programming

Anexo 1.3



Figura 3: Metodología MSF

Anexos del Capítulo 2 “Especificación textual de los CU”

Anexo2.1 Especificación textual del CU “Autenticar”

Caso de Uso	Autenticar Usuario
Actor	Usuario del Sistema
Propósito	Chequea el acceso a la aplicación.
Resumen	El usuario se autentifica en la aplicación.
CU asociados	
Precondiciones	

Sistema de Gestión de No Conformidades

Poscondiciones	Tener acceso al sistema según su rol.	
Referencias	RF1	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema le muestra una interfaz con un formulario para que se autentifique.	
2. El actor ingresa los datos en el formulario (usuario y contraseña) y presiona el botón Aceptar .	3. El sistema verifica que los campos del formulario no estén vacíos.	4. Verifica la existencia del usuario en el sistema.
	5. El sistema le da acceso al usuario a interactuar con la aplicación según su rol.	
Flujos Alternos		
2.1. El actor presiona el botón Cancelar .	2.2. El sistema muestra un mensaje "Por favor introduzca bien los datos".	
3.2. Va al paso 2.	3.1. Muestra mensaje de error "Faltan campos por llenar".	5.1. Muestra un mensaje "Usted no pertenece al Sistema".
	El sistema va al paso 2.	

Anexo2.2 Especificación textual del CU "Gestionar Proyecto"

Caso de Uso	Gestionar Proyecto
Actor	Administrador del Sistema
Propósito	Permitir al administrador del sistema insertar, modificar o eliminar un proyecto los cuales formarán parte de un grupo de proyectos del sistema.
Resumen	El Administrador del Sistema inserta, elimina o modifica un proyecto.
CU asociados	
Precondiciones	El usuario debe ser Administrador del Sistema.

Sistema de Gestión de No Conformidades

Poscondiciones	El sistema contiene proyectos insertados.
Referencias	RF4.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Adicionar Proyecto . -Si el actor selecciona la opción Modificar Proyecto . (Ver Sección I) -Si el actor selecciona la opción Eliminar Proyecto . (Ver Sección II)	2. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo proyecto.
3. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	4. El sistema verifica que no existan campos vacíos. 6. El sistema verifica que el proyecto no exista.
	7. El sistema muestra un mensaje indicando que el proyecto ha sido insertado correctamente.
	8. El sistema actualiza el listado de los proyectos.
Flujos Alternos	
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra la ventana.
	4.1. El Sistema muestra un mensaje de error "Verificar datos de entrada." 4.2. El sistema va al paso 4. 6.1. Muestra un mensaje de error "El proyecto ya existe." 6.2. El sistema va al paso 6.
Sección I: Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic derecho sobre el proyecto que	2. El sistema muestra una interfaz con los datos

Sistema de Gestión de No Conformidades

desea modificar y selecciona el botón Modificar .	que posee el proyecto.
3. El actor modifica los datos del proyecto y presiona el botón Aceptar .	4. El sistema verifica que no existan campos vacíos.
	5. El sistema actualiza la lista de proyectos.
Flujos Alternos	
3.1. El usuario presiona el botón Cancelar .	3.2. El sistema cierra la interfaz.
	4.1. El sistema muestra un mensaje "Verificar datos de entrada."
	4.2. El sistema va al paso 4.
Sección II: Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el proyecto presiona clic derecho sobre el proyecto que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje "Está seguro que desea eliminar el Proyecto".
3. El actor presiona el botón Si .	4. El sistema muestra un mensaje "Se eliminó correctamente el Proyecto."
	5. El sistema actualiza la lista de proyecto.
Flujos Alternos	
3.1. El actor presiona el botón No .	3.2. El sistema cierra la ventana.

Anexo2.3 Especificación textual del CU "Gestionar Usuario de Proyecto"

Caso de Uso	Gestionar Usuario de Proyecto
Actores	Líder del Proyecto
Propósito	Permitir al Líder del Proyecto insertar, modificar, eliminar y mostrar un usuario a los cuales se le asignaran diferentes roles dentro del proyecto.
Resumen	El Líder del Proyecto inserta, elimina, modifica y muestra un usuario.
CU asociados	
Precondiciones	El Líder de Proyecto debe haberse autenticado.
Poscondiciones	El sistema contiene usuarios insertados.

Sistema de Gestión de No Conformidades

Referencias	RF3.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el botón Adicionar usuario . -Si el actor selecciona la opción Modificar usuario . (Ver Sección I) -Si el actor selecciona la opción Eliminar usuario . (Ver Sección II)	2. El sistema muestra un formulario con los campos necesarios para adicionar un nuevo usuario.
3. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	4. El sistema verifica que no existan campos vacíos.
	5. El sistema verifica que los datos introducidos están correctamente.
	6. El sistema verifica que el usuario no exista.
	7. El sistema muestra un mensaje indicando que el usuario ha sido insertado correctamente.
8. El actor presiona el botón Ok .	9. El sistema actualiza el listado de los usuarios del proyecto.
Flujos Alternos	
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra el formulario.
	4.1. El Sistema muestra un mensaje de error "Verificar los datos de entrada."
4.2. El actor presiona el botón Ok .	4.3. El sistema va al paso 4.
	5.1. Muestra un mensaje "Verificar los datos de entrada."
5.2. El actor presiona el botón Ok .	5.3. El sistema va al paso 5.
	6.1. Muestra un mensaje de error "El usuario ya ha sido adicionado".

Sistema de Gestión de No Conformidades

6.2. El actor presiona el botón Ok .	6.3. El sistema va al paso 6.
Sección I: Modificar usuario	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el usuario que desea modificar y selecciona el botón Modificar usuario .	2. El sistema muestra un formulario con los datos que posee el usuario.
3. El actor modifica los datos del usuario y presiona el botón Aceptar .	4. El sistema chequea que los datos introducidos estén correctos.
	5. El sistema muestra un mensaje "Seguro que desea actualizar el usuario."
6. El actor presiona el botón Si .	7. El sistema muestra un mensaje de confirmación.
	8. El sistema actualiza la lista de usuarios.
Flujos Alternos	
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra el formulario.
6.1. El actor presiona el botón No .	6.2. El sistema cierra el formulario.
Sección II: Eliminar usuario	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el usuario que desea eliminar y presiona el botón Eliminar usuario .	2. El sistema muestra un mensaje "Seguro que desea eliminar el usuario".
3. El actor presiona el botón Si del mensaje de confirmación.	4. El sistema muestra un mensaje "Se eliminó correctamente el usuario."
	5. El sistema actualiza la lista de usuarios del proyecto.
Flujos Alternos	
3.1. El actor da clic sobre el botón No .	3.2. El sistema cierra la ventana de confirmación.

Anexo2.4 Especificación textual del CU "Gestionar Líderes de Proyecto"

Caso de Uso	Gestionar Líderes de Proyecto
--------------------	-------------------------------

Sistema de Gestión de No Conformidades

Actores	Administradores	
Propósito	Permitir a los Administradores Adicionar, Eliminar, Editar y Mostrar Líderes de Proyecto los cuales serán los que realicen diferentes actividades dentro del proyecto.	
Resumen	Permitir a los Administradores Adicionar, Eliminar, Editar y Mostrar Líderes de Proyecto.	
CU asociados		
Precondiciones	El Administrador debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	El sistema contiene Líderes de Proyecto añadidos.	
Referencias	RF3	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona la opción Adicionar usuario . -Si el actor selecciona la opción Modificar usuario . (Ver Sección I) -Si el actor selecciona la opción Eliminar usuario . (Ver Sección II)	2. El sistema muestra un formulario con los campos necesarios para adicionar un nuevo administrador del sistema.	
	3. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	
	4. El sistema verifica que no existan campos vacíos.	
	5. El sistema verifica que los datos introducidos están correctamente.	
	6. El sistema verifica que el usuario no exista.	
	7. El sistema muestra un mensaje indicando que el usuario ha sido insertado correctamente.	
	8. El sistema actualiza el listado de los usuarios.	
Flujos Alternos		

Sistema de Gestión de No Conformidades

3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra la ventana.
	4.1. El Sistema muestra un mensaje de error "Verificar los datos de entrada."
	4.2. El sistema va al paso 4.
	5.1. Muestra un mensaje "Verificar los datos de entrada."
	5.2. El sistema va al paso 5.
	6.1. Muestra un mensaje de error "No se pudo insertar el usuario."
	6.2. El sistema va al paso 6.
Sección I: Modificar usuario	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el usuario que desee modificar y selecciona el botón Modificar .	2. El sistema muestra un formulario con los datos que posee el usuario.
3. El actor modifica los datos del usuario y presiona el botón Aceptar .	4. El sistema chequea que los datos introducidos estén correctos.
	5. El sistema actualiza la lista de proyectos.
Flujos Alternos	
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra el formulario.
	4.1. El sistema muestra un mensaje "No pueden quedar campos vacíos."
4.2. El actor presiona el botón Aceptar .	4.3. El sistema va al paso 4.
Sección II: Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el usuario que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje "Seguro que desea eliminar el usuario".
3. El actor presiona el botón Aceptar del mensaje de confirmación.	4. El sistema muestra un mensaje "Se eliminó correctamente el usuario".
	5. El sistema actualiza la lista de usuarios.

Sistema de Gestión de No Conformidades

Flujos Alternos

3.1. El actor presiona el botón Cancelar	3.2. El sistema cierra el formulario.
---	---------------------------------------



Anexo2.5 Especificación textual del CU “Asignar NC “

Caso de Uso	Asignar NC
Actores	Responsable de Calidad
Propósito	Permitir al Responsable de Calidad asignar las NC halladas para su posterior corrección.
Resumen	Responsable de Calidad asigna las NC.
CU asociados	
Precondiciones	El Responsable de Calidad debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad. Debe haber NC añadidas.
Poscondiciones	El equipo de desarrollo contiene las NC halladas.
Referencias	RF9

Descripción

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El actor selecciona la NC a enviar y oprime el botón Asignar NC .	2. El sistema muestra la interfaz con los usuarios necesarios para Asignar una No Conformidad.
3. El actor selecciona al usuario que se le va a asignar la NC y oprime el botón Asignar.	4. El sistema envía los datos de la no conformidad y muestra un mensaje de confirmación.
5. El actor da clic en el botón Ok del mensaje.	6. La ventana se cierra.

Cursos Alternos

--

Sistema de Gestión de No Conformidades

Anexo2.6 Especificación textual del CU “Gestionar Elementos de Prueba “

Caso de Uso	Gestionar Elementos de Prueba	
Actores	Responsable de Calidad	
Propósito	Permitir al Responsable de Calidad adicionar, Eliminar, Modificar y Mostrar los elementos de prueba para realizar las pruebas de las cuales se sacaran las NC.	
Resumen	El Responsable de Calidad adicionará, eliminará, modificará y mostrará los elementos de prueba.	
CU asociados		
Precondiciones	El Responsable de Calidad debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	El sistema contiene los elementos de prueba añadidos.	
Referencias	RF5	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el botón Adicionar Elementos de prueba . -Si el actor selecciona la opción Modificar Elementos de prueba . (Ver Sección I) -Si el actor selecciona la opción Eliminar Elementos de prueba . (Ver Sección II)	2. El sistema muestra un formulario con los campos necesarios para adicionar un nuevo elemento de prueba.	
3. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	4. El sistema verifica que no existan campos vacíos.	
	5. El sistema muestra un mensaje indicando que el elemento de prueba ha sido insertado correctamente.	
6. El actor presiona el botón Aceptar .	7. El sistema actualiza el listado de elementos de prueba.	
Cursos Alternos		

Sistema de Gestión de No Conformidades

2.1. El actor presiona el botón Cancelar.	2.2. El sistema cierra el formulario.
	4.1. El Sistema muestra un mensaje de error “La prueba no se ha podido adicionar satisfactoriamente.”
6.1. El actor presiona el botón Cancelar .	6.2. El sistema va al paso 4.
Sección I: Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el elemento de prueba que desee modificar y selecciona el botón Modificar .	2. El sistema muestra una interfaz con los datos que posee el elemento de prueba.
3. El actor modifica los datos del elemento de prueba y presiona el botón Guardar .	4. El sistema chequea que no existan campos vacíos.
	5. El sistema chequea que los datos introducidos estén correctos.
	6. El sistema actualiza la lista de elementos de prueba.
Cursos Alternos	
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra la interfaz.
	4.1. El sistema muestra un mensaje “No pueden quedar campos vacíos.”
4.2. El actor presiona el botón Aceptar .	4.3. El sistema va al paso 4.
	5.1. El sistema muestra un mensaje” Introduzca correctamente los datos.”
5.2. El actor presiona el botón Aceptar .	5.3. El sistema va al paso 5.
4.2.1. El actor presiona el botón Cancelar .	4.2.2. El sistema cierra la interfaz.
5.2.1. El actor presiona el botón Cancelar .	5.2.2. El sistema cierra la interfaz.
Sección II: Eliminar	
Acción del Actor	Respuesta del Sistema

Sistema de Gestión de No Conformidades

1. El actor da clic sobre el elemento de prueba que desee eliminar y selecciona el botón Eliminar .	2. El sistema muestra un mensaje donde indica si está seguro que desea eliminar dicho el elemento de prueba.
3. El actor da clic sobre el botón Aceptar .	4. El sistema actualiza la lista de elementos de prueba.

Cursos Alternos

3.1 El actor da clic sobre el botón Cancelar.



Anexo2.7 Especificación textual del CU “Adicionar NC “

Caso de Uso	Adicionar NC	
Actor	Probador	
Propósito	Permitir al probador Adicionar la NC que se tratará de corregir.	
Resumen	El Probador Adiciona la NC.	
CU asociados		
Precondiciones	El Probador debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	NC añadidas.	
Referencias	RF6	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el botón Adicionar NC . -Si el actor selecciona la opción Modificar NC .(Ver Sección I) -Si el actor selecciona la opción Eliminar NC . (Ver Sección II)	2. El sistema muestra una interfaz con los campos necesarios para adicionar una No Conformidad.	
3. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	4. El sistema verifica que no existan campos vacíos.	

Sistema de Gestión de No Conformidades

	5. El sistema muestra un mensaje indicando que la NC ha sido insertada correctamente y cierra el formulario.
6. El actor presiona el botón ok .	7. El sistema actualiza el listado de las No Conformidades.
Cursos Alternos	
	4.1. El Sistema muestra un mensaje de error "Verificar datos de entrada."
4.2. El actor presiona el botón Ok .	4.3. El sistema va al paso 4.

Anexo2.8 Especificación textual del CU "Eliminar NC "

Caso de Uso	Eliminar NC	
Actor	Probador	
Propósito	Permitir al probador Eliminar la NC.	
Resumen	El Probador Elimina la NC.	
CU asociados		
Precondiciones	El Probador debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	NC eliminadas.	
Referencias	RF7	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona la NC a eliminar la opción Eliminar NC.	2. El sistema muestra una ventana de confirmación preguntando si está seguro que desea eliminar la NC.	
3. El actor presiona el botón Yes .	4. El sistema elimina la NC.	
Flujos Alternos		
3.1. El Probador da clic en el botón No.	3.2. El sistema cierra la ventana de confirmación.	

Sistema de Gestión de No Conformidades



Anexo2.9 Especificación textual del CU “Modificar NC “

Caso de Uso	Modificar NC	
Actor	Equipo de Desarrollo	
Propósito	Permitir al Equipo de Desarrollo modificar la NC.	
Resumen	El Equipo de Desarrollo modifica la NC.	
CU asociados		
Precondiciones	El integrante del Equipo de Desarrollo debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	NC modificadas.	
Referencias	RF8	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor da clic sobre la NC que desee modificar y selecciona el botón Modificar .	2. El sistema muestra un formulario con los datos que posee la NC.	
3. El actor modifica los datos de la NC y presiona el botón Aceptar .	4. El sistema chequea que no existan campos vacíos.	
	5. El sistema muestra una ventana de confirmación de modificación de la NC.	
6. El actor presiona el botón yes .	7. El sistema cierra el formulario y actualiza la lista de No Conformidades.	
Flujos Alternos		
3.1. El actor presiona el botón Cancelar .	3.2. El sistema cierra el formulario.	
6.2: El actor presiona el botón No .	6.3. El sistema va al paso 3.	



Anexo2.10 Especificación textual del CU “Generar reportes comparativos entre proyectos (concluidos o en desarrollo) de las No Conformidades detectadas. “

Sistema de Gestión de No Conformidades

Caso de Uso	Generar reportes comparativos entre proyectos (concluidos o en desarrollo) de las No Conformidades detectadas	
Actor	Líder de Proyecto	
Propósito	Permitir al Líder de Proyecto generar reportes con respecto a la información que se ha guardado en el tiempo importante para los proyectos de nuevo inicio.	
Resumen	El Líder de Proyecto genera reportes de información almacenada en el tiempo.	
CU asociados		
Precondiciones	El actor debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	Reportes generados.	
Referencias	RF12	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona la opción reportes históricos.	1. El sistema muestra una interfaz con diferentes tipos de reportes a escoger.	
2. El actor selecciona la opción Mostrar media de No Conformidades por estado. -Si el actor selecciona la opción Mostrar media de No Conformidades por fases. (Ver Sección I) -Si el actor selecciona la opción Mostrar media de No Conformidades por flujo de trabajo. (Ver Sección II) -Si el actor selecciona la opción Mostrar media de No Conformidades por prioridad. (Ver Sección III)	3. El sistema muestra un formulario con los detalles del reporte por llenar.	
4. El actor llena los campos y presiona el botón Aceptar.	5. El sistema muestra una ventana de confirmación con el supuesto cálculo.	

Sistema de Gestión de No Conformidades

6. El actor presiona el botón ok .	7. El sistema cierra la ventana.
Cursos Alternos	
5.1 El actor presiona el botón Cancelar .	5.2 El sistema cierra la ventana de confirmación.
Sección I: Mostrar media de No Conformidades por fase	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Mostrar media de No Conformidades por fase.	2. El sistema muestra un formulario con los detalles del reporte por llenar.
3. El actor llena los campos y presiona el botón Aceptar.	4. El sistema muestra una ventana de confirmación con el supuesto cálculo.
5. El actor presiona el botón ok .	6. El sistema cierra la ventana.
Cursos Alternos	
5.1 El actor presiona el botón Cancelar .	5.2 El sistema cierra la ventana de confirmación.
Sección II: Mostrar media de No Conformidades por flujo de trabajo	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Mostrar media de No Conformidades por fase.	2. El sistema muestra un formulario con los detalles del reporte por llenar.
3. El actor llena los campos y presiona el botón Aceptar.	4. El sistema muestra una ventana de confirmación con el supuesto cálculo.
5. El actor presiona el botón ok .	6. El sistema cierra la ventana.
Flujos Alternos	
5.1 El actor presiona el botón Cancelar .	5.2 El sistema cierra la ventana de confirmación.
Sección III: Mostrar media de No Conformidades por prioridad	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Mostrar media de No Conformidades por fase.	2. El sistema muestra un formulario con los detalles del reporte por llenar.
1. El actor llena los campos y presiona el botón Aceptar.	2. El sistema muestra una ventana de confirmación con el supuesto cálculo.
3. El actor presiona el botón ok .	4. El sistema cierra la ventana.

Sistema de Gestión de No Conformidades

Flujos Alternos

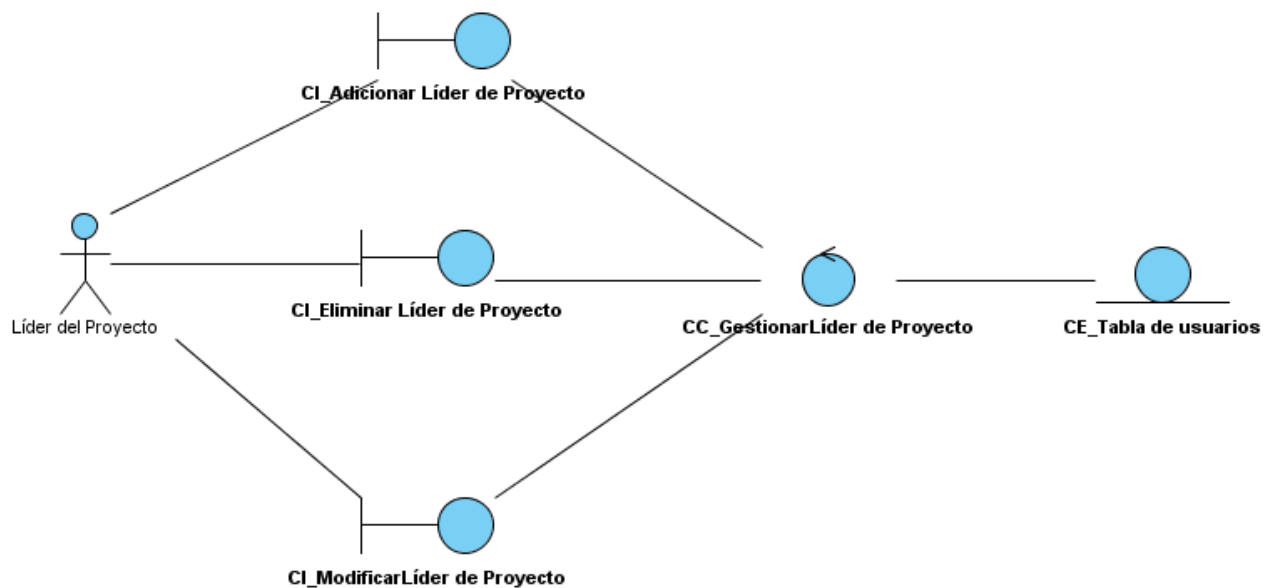
5.1 El actor presiona el botón Cancelar.

5.2 El sistema cierra la ventana de confirmación.

Anexos del Capítulo 3 “Diagramas de Clases del Análisis”

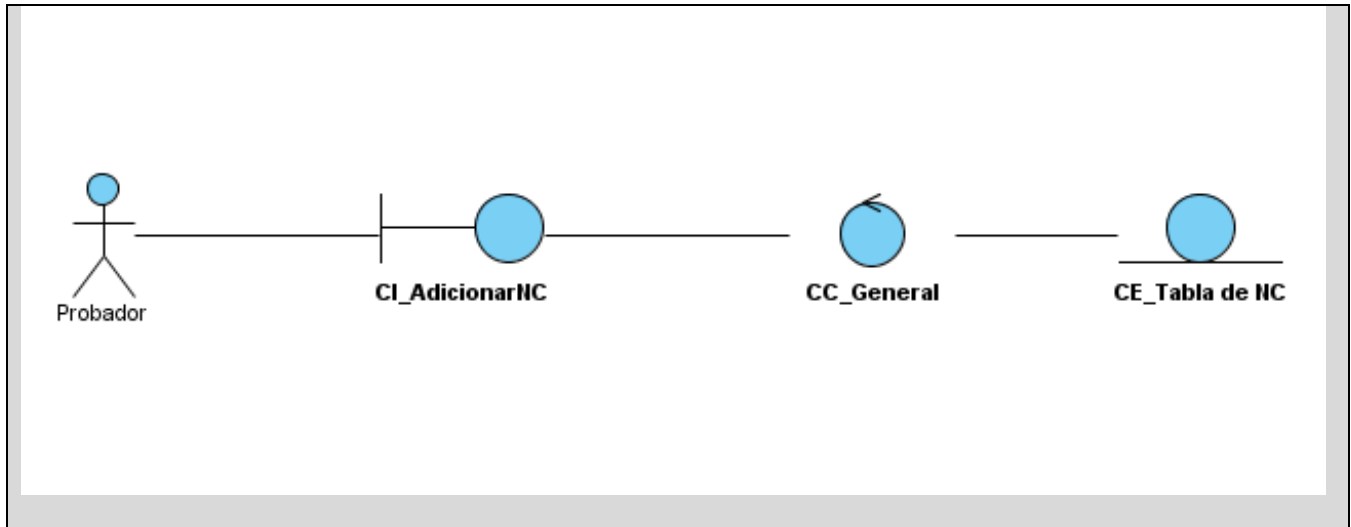
Anexo3.1 Diagramas de Clases del Análisis “Gestionar Administradores de Sistema”

Diagrama de Clases del Análisis



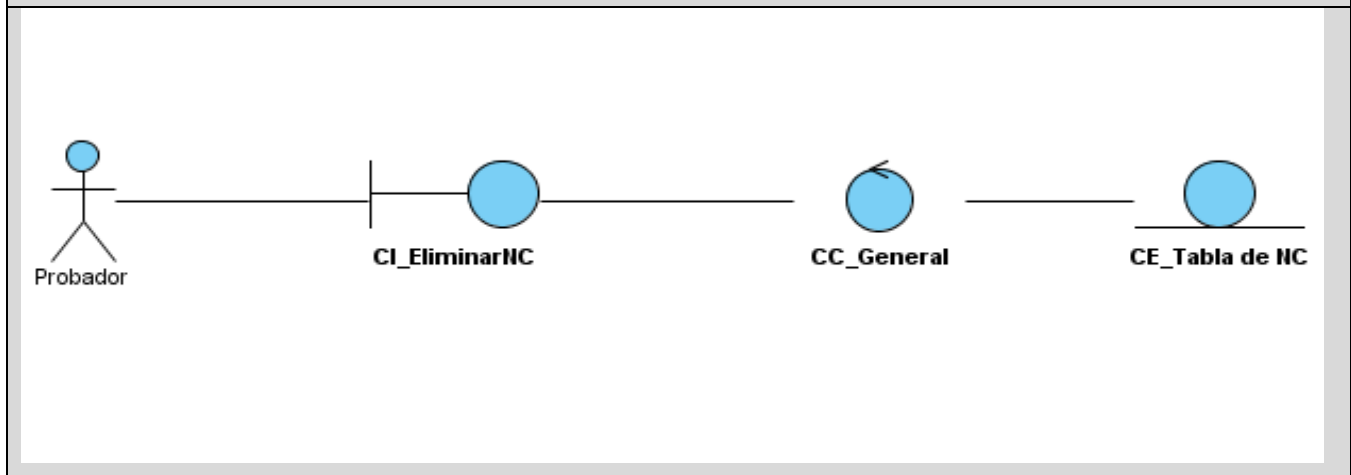
Anexo3.2 Diagramas de Clases del Análisis “Adicionar NC”

Diagrama de Clases del Análisis



Anexo3.3 Diagramas de Clases del Análisis “Eliminar NC”

Diagrama de Clases del Análisis

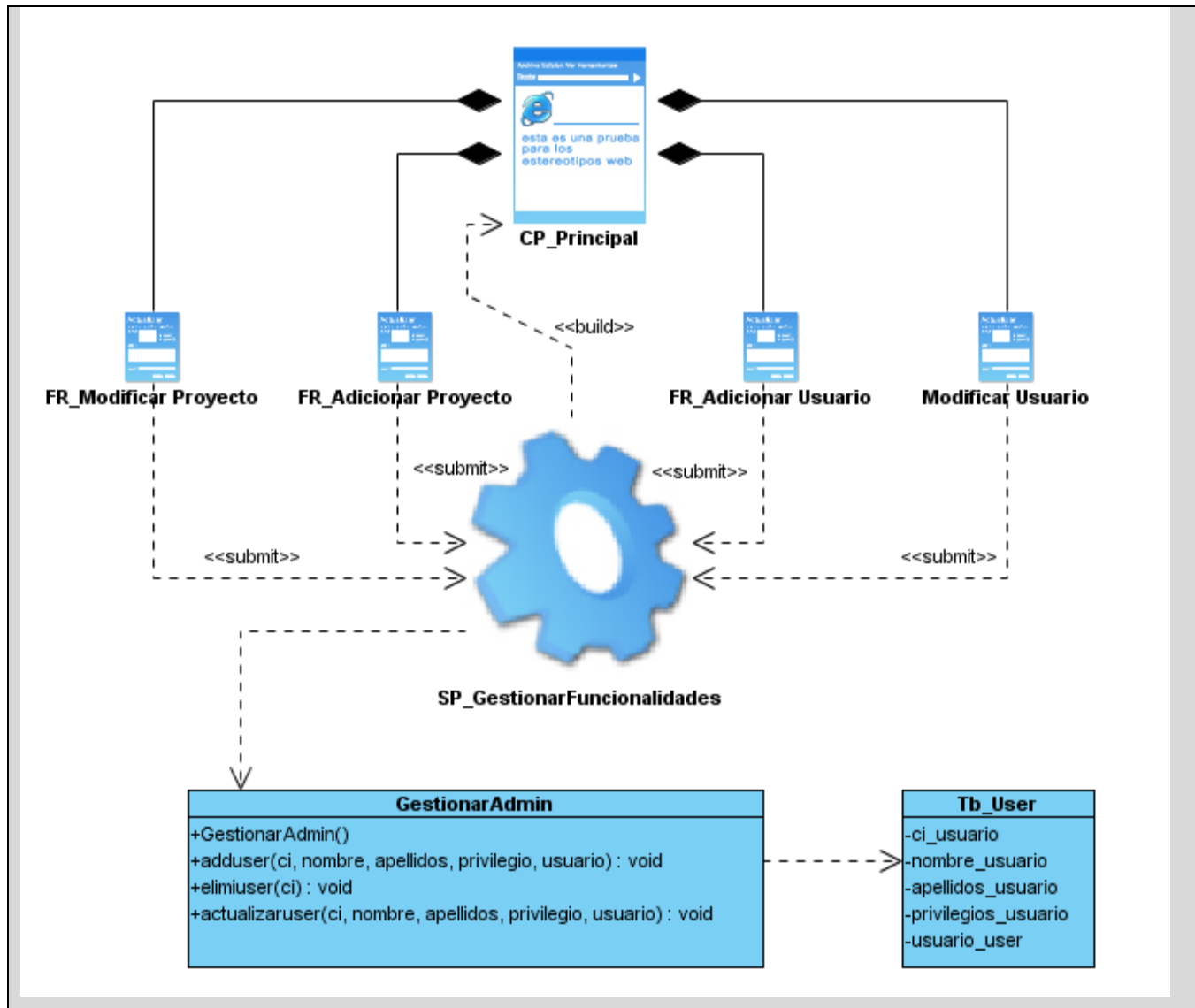


Anexos del Capítulo 3 “Diagramas de Clases del Diseño.”

Anexo3.2.1 Diagramas de Clases del Diseño “Gestionar Administradores de Sistema”

Diagramas de Clases del Diseño

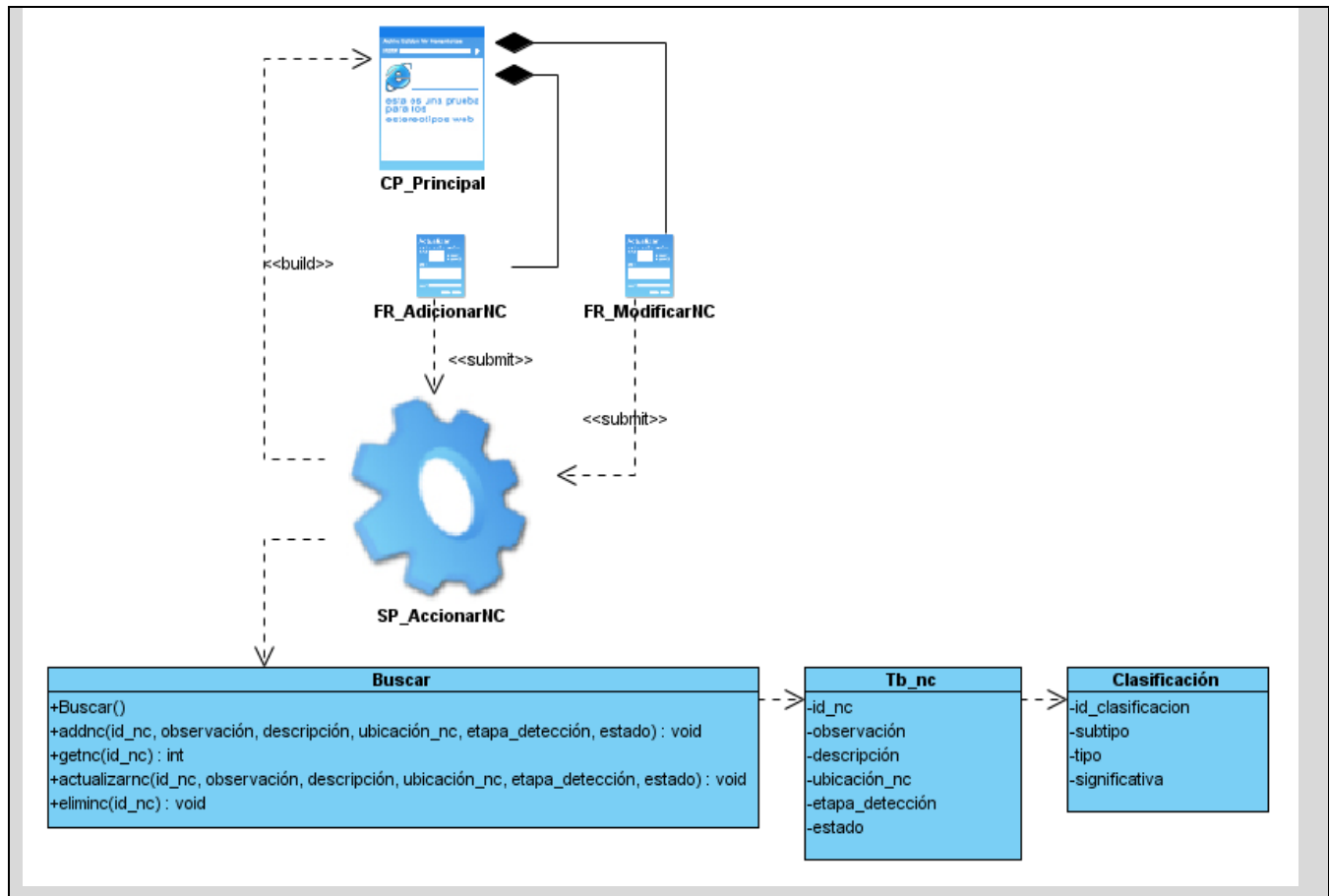
Sistema de Gestión de No Conformidades



Anexo3.2.2 Diagramas de Clases del Diseño “Añadir y Eliminar NC”

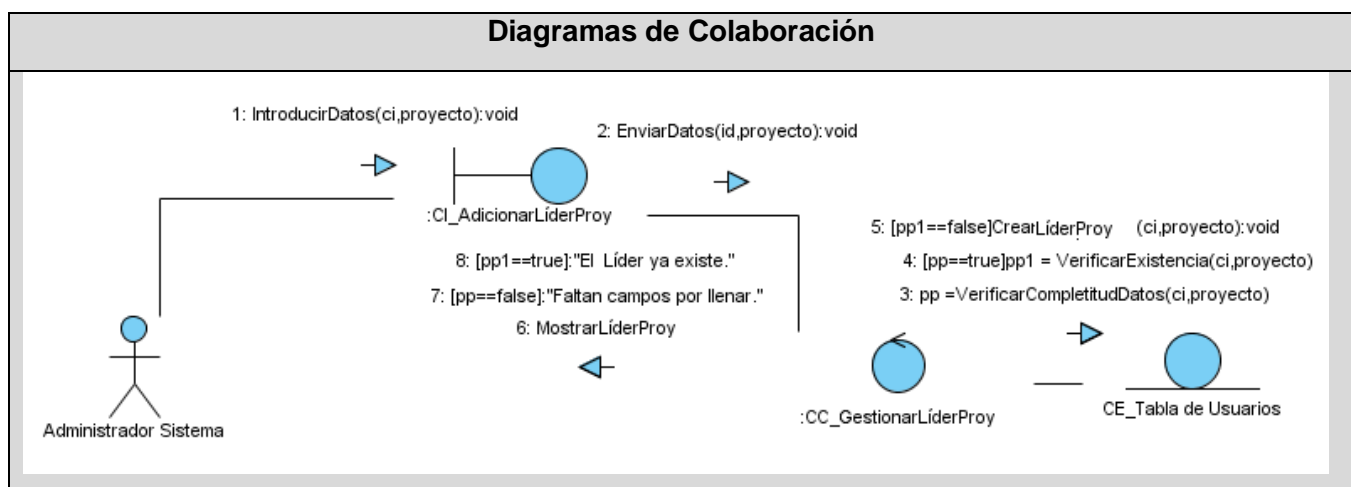
Diagramas de Clases del Diseño

Sistema de Gestión de No Conformidades

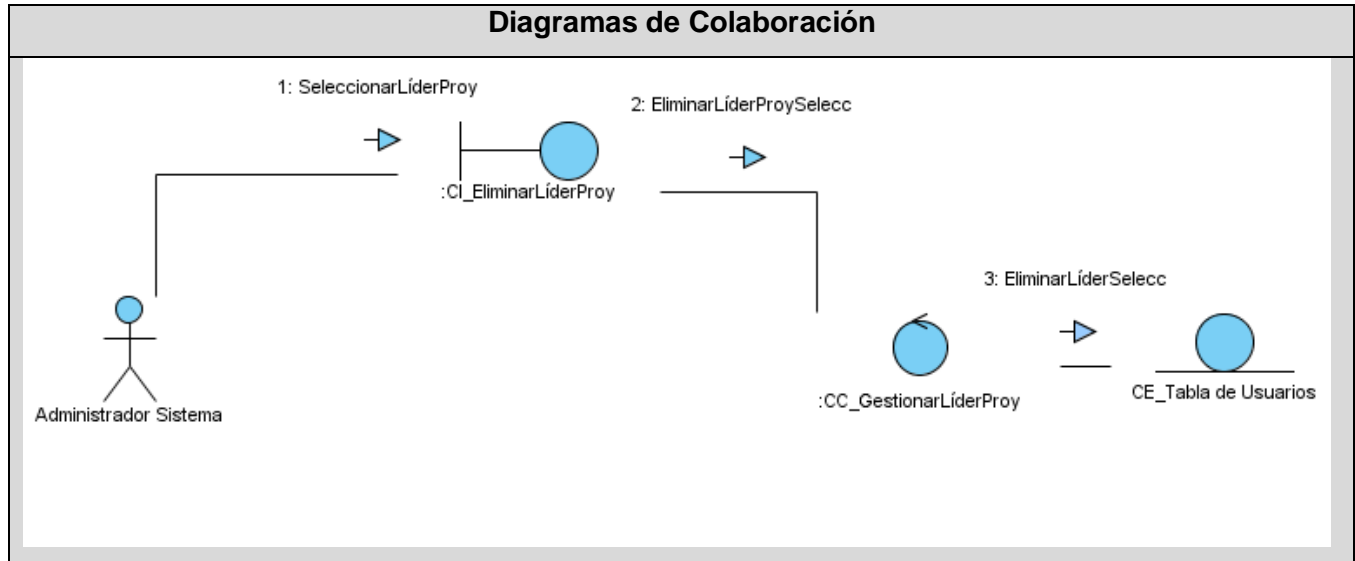


Anexos del Capítulo 3 “Diagramas de colaboración del análisis.”

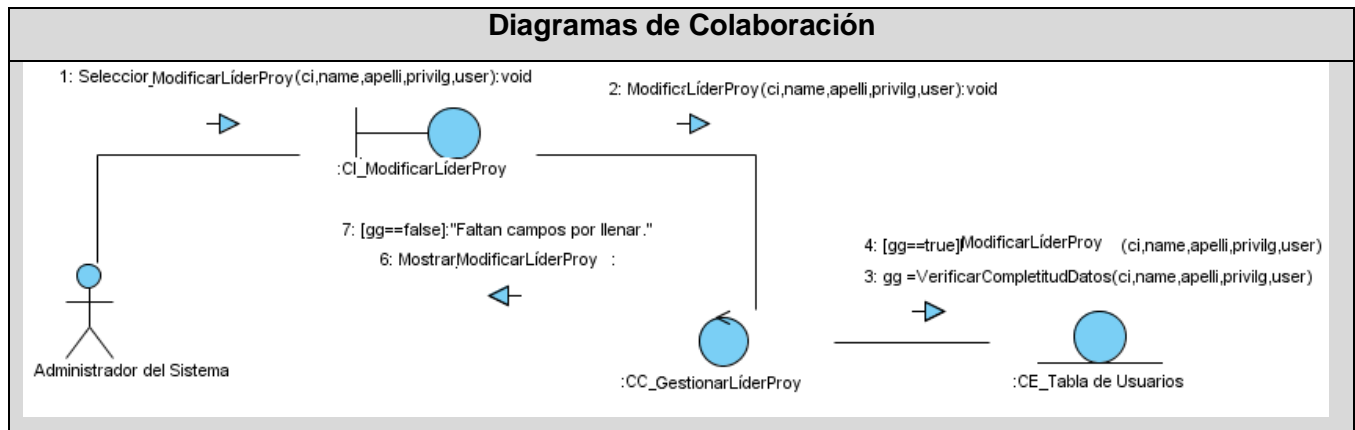
Anexo3.3.1 Diagramas de Colaboración del Análisis “Adicionar Líder de Proyecto”



Anexo3.3.2 Diagramas de Colaboración del Análisis “Eliminar Líder del Proyecto”



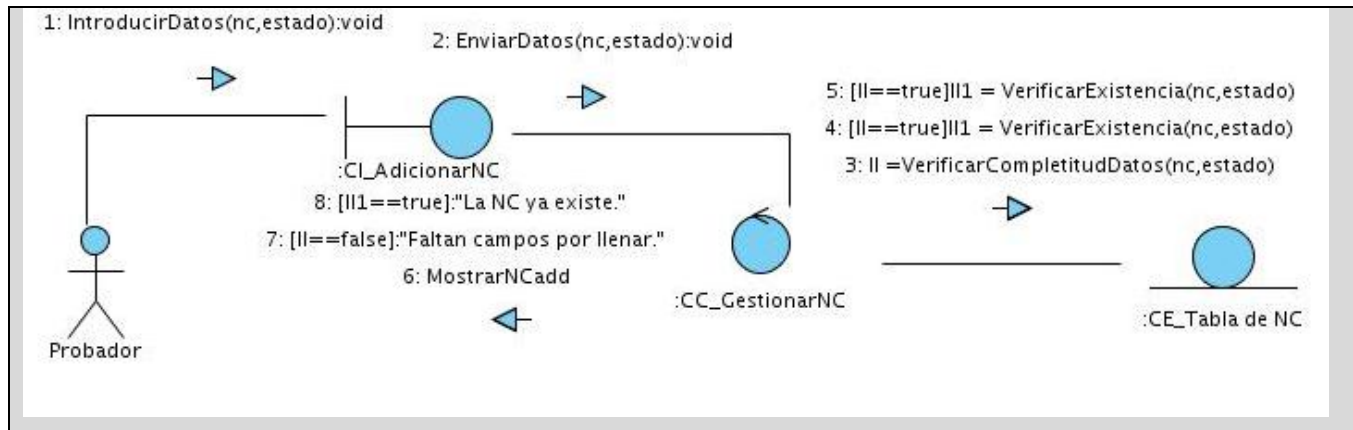
Anexo3.3.3 Diagramas de Colaboración del Análisis “Modificar Líder del Proyecto”



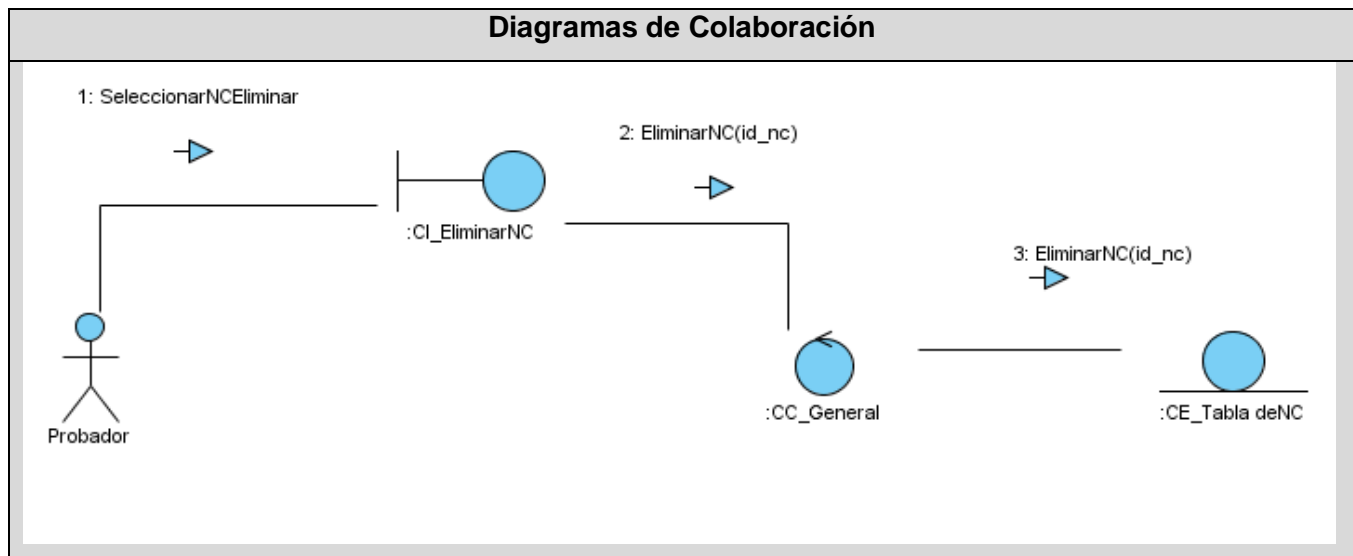
Anexo3.3.4 Diagramas de Colaboración del Análisis “Adicionar NC”



Sistema de Gestión de No Conformidades



Anexo3.3.5 Diagramas de Colaboración del Análisis “Eliminar NC“



Anexos del Capítulo 4 “Tablas de Descripción de las clases persistentes.”

Anexo4.1 Tabla “Tb_ proyecto“

Nombre: Tb_ proyecto		
Descripción: Se almacenan los datos correspondientes a los proyectos con los que se van a trabajar.		
Atributo	Tipo	Descripción
id_proy	int	Es la llave primaria y almacena el identificador de un proyecto.
nombre	String	Almacena el nombre que lleva el proyecto.
responsable	String	Almacena el nombre del responsable al frente del proyecto.

Sistema de Gestión de No Conformidades

fecha_inic	String	Almacena el tiempo en que se comienza realizar el proyecto.
fecha_fin	String	Almacena el tiempo en que se termina la realización del proyecto.

Anexo4.2 Tabla “Tb_ user“

Nombre: Tb_ user		
Descripción: Se almacenan los datos correspondientes a los usuarios con los que se van a trabajar.		
Atributo	Tipo	Descripción
ci	String	Es la llave primaria y almacena el carnet de identidad del usuario.
nombre	String	Almacena el nombre que lleva el usuario.
apellidos	String	Almacena los apellidos del usuario.
privilegio	String	Almacena con respecto a su función el privilegio del usuario.
usuario	String	Almacena el usuario del sistema del usuario
id_proy	int	Es la llave foránea y almacena el identificador de un proyecto.

Anexo4.3 Tabla “Tb prueba“

Nombre: Tb prueba		
Descripción: Se almacenan los datos correspondientes a las pruebas que se van a realizar.		
Atributo	Tipo	Descripción
id_prueba	int	Es la llave primaria y almacena el identificador de la prueba.
fecha	String	Almacena la fecha del día que se realizó.
autor	String	Almacena el nombre que lleva el autor.
descripción	String	Contiene una descripción detallada de la prueba a realizar.
versión	float	Almacena la versión de la prueba
ci	String	Es la llave foránea y almacena el carnet de identidad del usuario que realizó dicha prueba.
id_proy	int	Es la llave foránea y almacena el id del proyecto que realizó dicha prueba.

Anexo4.4 Tabla “Tb_ clasificación“

Sistema de Gestión de No Conformidades

Nombre: Tb_ clasificación		
Descripción: Se almacenan los datos correspondientes a la clasificación de la no conformidad.		
Atributo	Tipo	Descripción
id_clasificación	String	Es la llave primaria y almacena el identificador de dicha NC.
sub_tipo	String	Describe el sub_tipo de no conformidad existente.
tipo	String	Describe el tipo de no conformidad existente.
significativa	String	Describe si es significativa o no.

Anexo4.5 Tabla “Tb_ secciones”

Nombre: Tb_ secciones		
Descripción: Se almacenan los datos correspondientes a las secciones de las pruebas realizadas.		
Atributo	Tipo	Descripción
id_sección	int	Es la llave primaria y almacena el identificador de la sección.
nombre_secc	String	Almacena el nombre de la sección.
escenarios_secc	String	Almacena los escenarios de cada sección.
descrip_func	String	Almacena la descripción de las funciones de la sección.

Anexo4.6 Tabla “Tb_ escenarios”

Nombre: Tb_ escenarios		
Descripción: Almacena los valores esperados de cada sección una vez realizadas las pruebas.		
Atributo	Tipo	Descripción
nombre	String	Almacena el nombre del escenario.
variable	String	Almacena la variable determinada.
respuesta_esp	String	Almacena el resultado que se espera al realizar la prueba.
resultado_test	String	Almacena el resultado que se obtiene al realizar la prueba.
flujo_central	String	Pasos a desarrollar para probar la Funcionalidad que se indicó.
id_sección	String	Es la llave foránea y almacena el identificador de la sección.
Id_escenario	int	Almacena el id del escenario.

GLOSARIO DE TÉRMINOS

VP: Visual Paradigm.

NC: No Conformidades.

RC: Responsable de Calidad.

J2EE: Java 2, Enterprise Edition.

DCP: Diseño de Caso de prueba.

DCU: Diseño de Caso de uso.

DR: Diseño de requerimientos.

HTTP: HyperText Transfer Protocol.

GNU GPL (GNU General Public License)

UCI: Universidad de las Ciencias Informáticas.