



**Universidad de las Ciencias Informáticas**  
**Facultad 7**

**Título: Diseño de una aplicación para el Seguimiento de  
Errores de los productos software  
de la Facultad 7.**

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

**Autores:** Yanersy Díaz Colomé.  
Yenisel Molina Hernández.

**Tutora:** Ing. Lourdes Escalona Peral.

**Asesora:** Ing. Yunaysy Ortiz Batista.

**Consultante:** Ms. Marta Rosa Abreu Bosch.

Ciudad de La Habana  
Junio de 2007  
“Año 49 de la Revolución”

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_29\_ días del mes de \_junio\_ del año \_2007\_.

\_\_\_\_\_  
Yanersy Díaz Colomé

\_\_\_\_\_  
Yenisel Molina Hernández

\_\_\_\_\_  
Lourdes Escalona Peral

***"Lo que está de seguro en el juicio, de seguro será bien  
puesto en los labios."***

***José Martí***

## **AGRADECIMIENTOS**

A Daniel, por las brillantes ideas aportadas al trabajo y todas las horas que nos ha dedicado.

A Marta, por su positiva influencia en nuestra formación profesional y toda la ayuda brindada.

A Alain, porque siempre estuvo ahí, para responder nuestras dudas.

Al piquete de “ingenieras”, por haber dedicado al menos cinco minutos de su tiempo a este proyecto: Nadiezka, Keila, Yanelis y Kenia.

A Rudel, por toda su colaboración, a pesar de la distancia.

A Lázaro por su contribución y apoyo brindado.

A todos, los que de una forma u otra han contribuido, aunque sea con un gesto o una palabra, muchas gracias.

## **DEDICATORIA**

### **De Yenisel:**

A mis padres, mi abuela y mi hermano que tanto confiaron en mí y me han brindado siempre su apoyo, cariño y dedicación.

Al Comandante en Jefe Fidel Castro, por darme la oportunidad de estudiar en esta maravillosa Universidad.

A los Cabrera – Celada, que también son parte de mi familia y tanto han contribuido a mi formación y crianza.

A mis amigas de hoy y de siempre, que han estado ahí, compartiendo los buenos y malos momentos: Suanly, Keila, Nadiezka, Yanelis y Kenia.

A mis amigos Alain, Javier y Alejandro por lo mucho que me han apoyado.

A mi compañera de tesis, porque a pesar de todo, nos hemos soportado; logrando trabajar en equipo y gracias a eso, ha sido posible realizar este sueño.

### **De Yanersy:**

A mis padres, por haber sido guía y ejemplo a lo largo de toda mi vida. Por haberme apoyado siempre y por darme tanto amor.

A mi tía Zoraida, a quien debo gran parte de lo que soy y que siempre soñó con este día.

A Tita, por haber estado a mi lado en todo momento, por todo su apoyo y confianza, por ser una persona tan importante para mí.

A mi hermano Carlitos, que siempre ha seguido mis pasos, gracias por tanto cariño.

A mi hermana, por ser una personita tan especial.

A mi abuela.

A Daniel, por entregarme tanto amor a diario y ser un ejemplo para mí.

A Mayelín, por haber confiado tanto en que este momento llegaría y regalarme una amistad tan bonita.

A mis amigas de siempre: Yeni y Yanaisy, por tantos buenos y malos momentos compartidos.

Al Comandante, por ser guía y ejemplo de todos los cubanos.

A mi compañera de tesis Yenisel, por haberme soportado por más de seis meses.

Al colectivo de profesores de la Facultad Regional de Artemisa, con el que compartí tantos buenos momentos y a mis alumnos de allá.

## **RESUMEN.**

Desde los inicios, la Universidad de las Ciencias Informáticas, ha sido llamada a convertirse en una industria para la producción de software de alto prestigio, cuyos productos puedan insertarse de forma satisfactoria en el mercado mundial. Para lograr dicho propósito, se desarrolla un proceso de Control de la Calidad, que intenta ser cada vez más eficiente. El Grupo de Calidad de la Facultad 7, trabaja para controlar que los productos desarrollados en ella cumplan con los requisitos establecidos aunque persisten problemas, principalmente, en el proceso de seguimiento de errores.

El objetivo del presente trabajo de diploma, es diseñar un sistema informático para la automatización del seguimiento de errores, que favorezca el proceso de control de calidad de los productos de software desarrollados en la Facultad 7.

El sistema está concebido bajo la metodología RUP, por ser la más utilizada a nivel mundial; basada a su vez, en UML (Lenguaje Unificado de Modelado). Para documentar el software, se utilizó la herramienta Rational Rose Enterprise Edition 2003.

A partir del uso de la aplicación, se facilitará el trabajo a los miembros del Grupo de Calidad de la Facultad 7, en cuanto a la detección y el seguimiento de errores. Además, para los desarrolladores, será más sencillo corregir los errores detectados. La aplicación debe propiciar la disminución del período de pruebas de un producto y el aumento de su eficiencia. El sistema facilitará además, la elaboración de los documentos legales que se elaboran en el proceso de revisión de un producto.

# TABLA DE CONTENIDOS.

<b>Introducción:</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica</b> .....	<b>5</b>
1.1 Conceptos básicos asociados al negocio .....	5
1.2 Sistemas automatizados existentes .....	5
1.3 Tecnologías actuales a considerar .....	10
1.3.1 ¿Qué es Internet? .....	10
1.3.2 Aplicaciones Web. Ventajas y Carencias .....	10
1.3.3 Servidores Web.....	13
1.3.4 Lenguajes de Programación Web .....	14
1.3.5 Tecnología .NET vinculada con el paquete de Office .....	24
1.3.6 Navegadores o browsers .....	24
1.3.7 Sistemas Gestores de Bases de Datos .....	28
1.3.8 Entornos Distribuidos. Modelo Cliente-Servidor. Ventajas .....	31
1.3.9 Metodologías de desarrollo de software, lenguajes y herramientas para el modelado .....	35
1.4 Tecnologías y herramientas a utilizar.....	40
<b>Capítulo 2: Características del Sistema.</b> .....	<b>42</b>
2.1 Objeto de estudio .....	42
2.1.1 Flujo actual de los procesos involucrados en el campo de acción. ....	42
2.1.2 Análisis crítico de la ejecución de los procesos.....	43
2.1.3 Información que se maneja. ....	43
2.1.4 Objeto de automatización.....	44
2.3 Modelo de negocio. ....	45
2.3.1 Actores del Negocio:.....	45
2.3.2 Trabajadores del Negocio. ....	46
2.3.3 Diagrama de Casos de Uso del Negocio. ....	46
2.3.4 Descripción Textual de los Casos de Uso del Negocio. ....	47
2.3.5 Diagramas de Actividades.....	49
2.3.6 Diagrama de clases del modelo de objetos. ....	49
2.4 Especificación de los requisitos de software. ....	49
2.4.1 Requisitos Funcionales. ....	49
2.4.2 Requisitos No funcionales: .....	51
2.5 Definición de los casos de uso. ....	54

2.5.1 Definición de los actores .....	54
2.5.2 Listado de casos de uso .....	55
2.5.3 Diagrama de casos de uso.....	58
2.5.4 Casos de uso expandidos. ....	59
<b>Capítulo 3: Análisis y Diseño del Sistema. ....</b>	<b>65</b>
3.1 Análisis.....	65
3.1.1 Diagramas de clases del análisis.....	65
3.1.2 Diagramas de interacción (Colaboración).....	67
3.2 Diseño.....	71
3.2.1 Fundamentación del uso de patrones. ....	71
3.2.2 Definición de elementos de diseño. ....	71
3.2.3 Diagramas de clases del diseño. ....	73
3.2.4 Diagramas de interacción. (Secuencia).....	76
3.2.5 Descripción de las clases.....	84
3.2.6 Diseño de la Base de Datos. ....	91
3.2.6.1 Descripción de las tablas. ....	94
3.2.7 Tratamiento de errores. ....	100
3.2.8 Interfaz. ....	100
3.2.9 Concepción de la ayuda.....	101
3.3 Diagrama de despliegue. ....	101
<b>Conclusiones. ....</b>	<b>103</b>
<b>Recomendaciones. ....</b>	<b>104</b>
<b>Referencias Bibliográficas. ....</b>	<b>105</b>
<b>Bibliografía. ....</b>	<b>108</b>
<b>Anexos. ....</b>	<b>109</b>
<b>Glosario de Términos. ....</b>	<b>114</b>



## LISTA DE TABLAS

Tabla 2.1 – Descripción de los actores del negocio. ....	45
Tabla 2.2 – Descripción de los trabajadores del negocio. ....	46
Tabla 2.3 – Descripción del Caso de Uso del Negocio: Revisar producto. ....	49
Tabla 2.4 – Descripción de los actores del sistema. ....	54
Tabla 2.5 – Caso de Uso Actualizar Estado de No Conformidades. ....	55
Tabla 2.6 – Caso de Uso Mostrar No Conformidades Aceptadas. ....	55
Tabla 2.7 – Caso de Uso Corregir No Conformidades de la documentación. ....	56
Tabla 2.8 – Caso de Uso Obtener Estadísticas ....	56
Tabla 2.9 – Caso de Uso Gestionar No Conformidades. ....	56
Tabla 2.10 – Caso de Uso Gestionar Acta de Liberación Parcial. ....	57
Tabla 2.11 – Caso de Uso Gestionar Certificado de Liberación ....	57
Tabla 2.12 – Caso de Uso Subir documentación. ....	57
Tabla 2.13 – Descripción Expandida del CU Gestionar No Conformidades. ....	63
Tabla 3.1 – Descripción CD: UI_ListarNC ....	84
Tabla 3.2 – Descripción CD: UI_Listar_NCSw ....	84
Tabla 3.3 – Descripción CD: UI_ActE_NCSW ....	85
Tabla 3.4 – Descripción CD: UI_ActE_NCDoc. ....	85
Tabla 3.5 – Descripción CD: UI_GestionarNC. ....	85
Tabla 3.6 – Descripción CD: UI_GestionarNCsw. ....	86
Tabla 3.7 – Descripción CD: UI_GestionarNCdoc ....	87
Tabla 3.8 – Descripción CD: UI_Mostrar NC Accept ....	87
Tabla 3.9 – Descripción CD: UI_VerNCDoc ....	88
Tabla 3.10 – Descripción CD: UI_VerNCSW ....	88
Tabla 3.11 – Descripción CD: UI_Det_NCSW ....	88
Tabla 3.12 – Descripción CD: CC_GestionarNC. ....	90
Tabla 3.13 – Descripción CD: CC_Controlador_Archivos. ....	90

Tabla 3.14 – Descripción CD: CE_ No Conformidad .....	91
Tabla 3.15 – Descripción CD: CE_NCSW .....	91
Tabla 3.16 – Descripción CD: CE_ CDoc .....	91
Tabla 3.17 – Descripción T_ CE_ NoConformidad. ....	94
Tabla 3.18 – Descripción T_ CE_ NCSw. ....	95
Tabla 3.19 – Descripción T_ CE_ NCdoc.....	95
Tabla 3.20 – Descripción T_ CE_ Cumplimiento. ....	95
Tabla 3.21 – Descripción T_ CE_ Clasificación. ....	96
Tabla 3.22 – Descripción T_ CE_ Estado. ....	96
Tabla 3.23 – Descripción T_ CE_ Certificado de Liberación. ....	97
Tabla 3.24 – Descripción T_ CE_Acta de Liberación Parcial. ....	97
Tabla 3.25 – Descripción T_ CE_ NCondición de Trabajo. ....	97
Tabla 3.26 – Descripción T_ CE_ Condición de Trabajo .....	97
Tabla 3.27 – Descripción T_ CE_ Entrega del laboratorio. ....	98
Tabla 3.28 – Descripción T_ CE_Elementos No Revisados. ....	98
Tabla 3.29 – Descripción T_ CE_ Elementos Revisados. ....	98
Tabla 3.30 – Descripción T_ CE_ Documentos. ....	99
Tabla 3.31 – Descripción T_ CE_ Documento de Prueba. ....	99
Tabla 3.32 – Descripción T_ CE_ Documento del producto. ....	99

## LISTA DE FIGURAS

Fig. 2.1 – Diagrama de Casos de Uso del Negocio. ....	47
Fig. 2.2 – Diagrama de Casos de Uso. ....	58
Fig. 3.1 – DCA_ CU Gestionar No Conformidades. ....	66
Fig. 3.2 – DCA_ CU Actualizar Estado de No Conformidades. ....	66
Fig. 3.3 – DCA_ CU Mostrar No Conformidades Aceptadas.....	67
Fig. 3.4 – DC_ CU Gestionar No Conformidades (Escenario_ Listar NC). ....	67
Fig. 3.5 – DC_ CU Gestionar No Conformidades (Escenario_ Registrar NC Doc). ....	68
Fig. 3.6 – DC_ CU Gestionar No Conformidades (Escenario_ Registrar NC Sw). ....	68
Fig. 3.7 – DC_ CU Gestionar No Conformidades (Escenario_ Modificar NC Doc). ....	68
Fig. 3.8 – DC_ CU Gestionar No Conformidades (Escenario_ Modificar NC Sw). ....	69
Fig. 3.9 – DC_ CU Gestionar No Conformidades (Escenario_ Eliminar NC Doc). ....	69
Fig. 3.10 – DC_ CU Gestionar No Conformidades (Escenario_ Eliminar NC Doc). ....	69
Fig. 3.11 – DC_ CU Actualizar Estado de No Conformidades ....	70
Fig. 3.12 – DC_ CU Mostrar No Conformidades Aceptada. ....	70
Fig. 3.13 – Diagrama de subsistemas del diseño. ....	72
Fig. 3.14 – Vista general de la arquitectura. ....	73
Fig. 3.15 – DCD_ CU Gestionar No Conformidades. ....	74
Fig. 3.16 – DCD_ CU Actualizar Estado de No Conformidades. ....	75
Fig. 3.17 – DCD_ CU Mostrar No Conformidades Aceptadas.....	76
Fig. 3.18 – DS_ CU Gestionar No Conformidades (Escenario_ Buscar No Conformidades). ....	77
Fig. 3.19 – DS_ CU Gestionar No Conformidades (Escenario_ Insertar No Conformidades de Doc). ....	78
Fig. 3.20 – DS_ CU Gestionar No Conformidades (Escenario_ Insertar No Conformidades de Sw). ....	79
Fig. 3.21 – DS_ CU Gestionar No Conformidades (Escenario_ Modificar No Conformidades de Doc). ....	80
Fig. 3.22 – DS_ CU Gestionar No Conformidades (Escenario_ Modificar No Conformidades de Sw). ....	81
Fig. 3.23 – DS_ CU Gestionar No Conformidades (Escenario_ Eliminar No Conformidades de Doc). ....	82
Fig. 3.24 – DS_ CU Gestionar No Conformidades (Escenario_ Eliminar No Conformidades de Sw). ....	83

Fig. 3.25 –.Diagrama de clases persistentes .....	92
Fig. 3.26 –. Diagrama de modelo de datos. ....	93
Fig. 3.27 –. Diagrama de Despliegue. ....	101
Fig. A2.1 –. Diagrama de Actividades.....	112
Fig. A3.1 –. Modelo de Objetos .....	113

## ***Introducción:***

La Universidad de las Ciencias Informáticas (UCI), es una institución surgida al calor de la Batalla de Ideas en el año 2002. Tiene dentro de sus principales objetivos convertirse en una Industria de Software de alto prestigio a nivel nacional e internacional.

En sus inicios esta idea parecía una utopía, pero con el devenir de los años se ha ido materializando poco a poco, obteniendo algunos resultados prometedores. Aunque, es un proceso lento que requiere de mucho esfuerzo tanto por parte de los estudiantes como profesores. Actualmente la Universidad cuenta con más de 200 proyectos productivos, que en su mayoría aportan considerables ingresos a la economía del país. De forma general, se contribuye al cumplimiento de una de las líneas directrices de la Batalla de Ideas, la informatización de la sociedad.

Para garantizar que los proyectos obtengan, como resultado de un proceso de desarrollo de software, productos de altísima calidad, la UCI se trazó como estrategia la formación de un Grupo de Calidad en cada Facultad, dirigidos por el Centro Nacional de Calidad de Software (CaliSoft).

Según Roger S. Pressman, la Calidad del Software se define como: “La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se esperan de todo software desarrollado profesionalmente”. (PRESSMAN 2005a)

La calidad de los sistemas de software se torna cada día más importante debido a que las inversiones en sistemas continúan creciendo.

Una empresa que se caracterice por productos de baja calidad pierde prestigio en el mercado y no suele llegar muy lejos. Esta idea se concreta en la siguiente frase citada en el libro, Un enfoque práctico: “La gente olvida cómo de rápido hiciste un trabajo, pero siempre recuerda cómo de bien lo hiciste” (Howard Newton).

Desde el momento en que comenzaron a funcionar los Grupos de Calidad, los miembros y directivos de los mismos fueron detectando, poco a poco, problemas que tienen durante la Gestión de la Calidad, y no han dado tregua hasta determinar o proponer estrategias para mejorar este proceso, principalmente la parte concerniente al seguimiento de errores.

No importa cuanto se prueben los programas y el cuidado que se tenga en su desarrollo, siempre salen errores inesperados. Por ello se hace necesario llevar un control de los errores detectados durante el proceso de revisión del producto, sin dejar pasar por alto ninguno, de esta manera se logrará verificar su corrección a fin de liberar un producto con la calidad requerida.

No existe en la Universidad ningún sistema que controle o gestione los errores detectados en los productos desarrollados en los diferentes proyectos productivos durante la revisión de los mismos. Este proceso se ha realizado mediante la elaboración de documentos digitales. Tampoco existe una investigación anterior con vistas a obtener una herramienta de este tipo, sin embargo, a nivel internacional muchas son las aplicaciones de gestión de errores utilizadas, el trabajo con alguna de ellas, es muy útil y efectivo.

Un desarrollador se beneficia al tener un sistema de control y seguimiento de errores, ya que le llegarán partes de los errores, con toda la información necesaria para arreglarlo. También se beneficia un probador, ya que puede ponerles prioridad a los errores y ver los que se han arreglado. Gestionar los errores beneficia a todos y principalmente al software resultante.

El seguimiento del error nos permite comparar el trabajo actual con esfuerzos pasados y proporciona una indicación cuantitativa de la calidad del trabajo que estamos realizando. (PRESSMAN 2005b)

En la Universidad de las Ciencias Informáticas, incluyendo la Facultad 7, este proceso no está definido concretamente, lo cual trae consigo que sea ineficiente todo el mecanismo de control y gestión de la calidad. Una vez detectados los errores por parte del equipo de calidad estos se le hacen llegar a los desarrolladores por medio de documentos (plantillas de no conformidades) para su corrección, pero el posterior seguimiento de los aspectos señalados y otros que puedan surgir, se hace engorroso al ser este un trabajo no automatizado.

Las técnicas empleadas para la revisión de la documentación no son factibles, se caracterizan por entorpecer el trabajo debido a su lentitud de utilización. Todo esto puede provocar que se escapen errores sin corregir o no detectados, repercutiendo en la calidad del producto y el prestigio de la institución.

Cuando al fin se consideran corregidos todos los errores del software el Grupo de Calidad emite un Acta de Liberación Parcial que envía a CaliSoft y si este aprueba el Acta de Liberación Parcial entonces envía un Certificado de Liberación al Grupo de Calidad. Este documento avala la Calidad del producto, luego una copia del Certificado de Liberación y del Acta de Liberación Parcial del Producto se les hace llegar a los

desarrolladores por parte del Grupo de Calidad. Este proceso tampoco se hace mediante un sistema automatizado para este fin.

Luego del análisis de cómo se realiza el proceso de seguimiento de errores, basado en la experiencia del Grupos de Calidad de la facultad 7, se identificó el siguiente **problema científico**: La no existencia de un sistema informático para el seguimiento de errores, que favorezca el proceso de control de la calidad de los productos de software desarrollados en la Facultad 7.

Este problema se enmarca en el **objeto de estudio**: Proceso de control de la calidad del software.

El **campo de acción** abarcado es: Seguimiento de errores en el proceso de control de la calidad de los productos de software desarrollados en la Facultad 7.

**Objetivo general:**

Diseñar un sistema informático para la automatización del seguimiento de errores, mediante el cual se favorezca el proceso de control de calidad de los productos de software desarrollados en la Facultad 7.

**Objetivos específicos:**

- Valorar las nuevas tendencias de las tecnologías de la información relacionadas con las aplicaciones Web, así como los sistemas existentes para el seguimiento de errores.
- Determinar las características del proceso de seguimiento de errores durante el control de la calidad de los productos software desarrollados en la Facultad 7.
- Realizar el diseño de una aplicación Web para el seguimiento de errores durante el proceso de control de la calidad de los productos software desarrollados en la Facultad 7.

Las **tareas** que se llevarán a cabo para darle cumplimiento a los objetivos trazados son:

- Analizar las herramientas utilizadas actualmente a nivel mundial para llevar a cabo el proceso de seguimiento de errores, valorar sus tendencias y tomar posición al respecto.
- Seleccionar la metodología y herramientas a utilizar para el desarrollo del sistema informático.
- Estudiar detalladamente el proceso de seguimiento de errores y evaluar la información obtenida con vistas a determinar los aspectos de interés.

- Analizar las necesidades de funcionamiento del sistema informático.
- Elaborar la documentación correspondiente al análisis y diseño del sistema a desarrollar.

El presente trabajo consta de Introducción, tres Capítulos, Recomendaciones, Bibliografía, Anexos y Glosario de Términos.

En el Capítulo 1 **Fundamentación Teórica** se muestra un estudio detallado sobre la situación de los sistemas automatizados utilizados en la actualidad a nivel mundial, así como la situación de las tecnologías actuales, cuyo resultado permitirá la toma de decisión sobre qué herramientas utilizar para el diseño e implementación del sistema.

En el Capítulo 2 **Características del Sistema** se abordan los aspectos relacionados con el objeto de estudio tales como, el flujo actual de los procesos, un análisis crítico de cómo se ejecutan los procesos, información que se maneja, así como Objeto de automatización. También se llevan a cabo los flujos de trabajo de Modelado del Negocio y Especificación de los Requisitos de software. En el primero se exponen los principales artefactos resultantes del Modelo de negocio, constituyendo estos la base para la obtención del modelo del sistema. En el segundo se muestran los requisitos funcionales y no funcionales, además de la descripción de los actores y casos de uso resultantes del flujo de trabajo de Requisitos.

En el Capítulo 3 **Análisis y Diseño del sistema** se hace referencia al Modelo de Análisis, al Modelo de Diseño, que incluye, los diagramas de clases de diseño, diagramas de interacción, descripción de las clases de diseño, así como al diseño de la Base de Datos.



## **Capítulo 1: Fundamentación Teórica**

En el presente capítulo se hace alusión a los conceptos fundamentales asociados al negocio. Se aborda, además, el estado del arte de los sistemas automatizados existentes para el seguimiento de errores. Por último se realiza un estudio de las tendencias, tecnologías y metodologías más usadas en la actualidad y se concluye con la selección de las herramientas a utilizar durante el desarrollo de la aplicación.

### **1.1 Conceptos básicos asociados al negocio**

**No Conformidades (NC):** Según la definición de ISO 9000: 2000 (3.6.2), una no conformidad es el “incumplimiento de un requisito”. (Grupo de Prácticas de auditoría ISO 9001 Guía sobre: Documentación de una no conformidad 2005) o un error detectado en la documentación durante la revisión.

**Registro de No Conformidades:** Documento donde se registran todas las no conformidades que se le detectan al producto.

**Calisoft:** Organización de Calidad perteneciente al Ministerio de la Informática y las Comunicaciones, radica en la UCI y se encarga de certificar la calidad de los productos que en dicho centro se desarrollan.

**Acta de Liberación Parcial (ALP):** Documento que envía el Grupo de Calidad a Calisoft solicitando la certificación de un producto software y donde se especifican algunas características del mismo.

**Certificado de Liberación (CL):** Documento emitido por Calisoft que certifica la calidad de un producto software.

### **1.2 Sistemas automatizados existentes**

En la actualidad se han introducidos cambios sustanciales al proceso de seguimiento de errores en los proyectos de desarrollo de software, muchos son los sistemas automatizados que gestionan este proceso a nivel mundial, estas herramientas son muy recomendadas pues su uso facilita el manejo de los errores detectados durante el proceso de calidad del software, y a su vez el trabajo del equipo encargado de llevar a cabo el mismo.

Sin dudas uno de los sistemas más conocidos es el llamado **Bugzilla** creado por Mozilla, nos permite tener un seguimiento de errores y cambios, comunicación entre los miembros del proyecto, enviar y

revisar parches y manejar la garantía de calidad. Además, entre otras características, nos permite llevar un control de tiempo, distintas formas de autenticación, gráficos y multilinguaje. Está diseñado de tal forma que se puede aplicar a prácticamente cualquier desarrollo de software. Bugzilla es un Software Libre, y por ello puede usarse de forma gratuita por quien quiera. Además, a diferencia del software gratuito (freeware), tiene la ventaja de que al estar disponible el código fuente, se puede modificar el programa para adaptarlo a las necesidades específicas que alguien pueda tener.

Actualmente Bugzila:

- Está en un período desarrollo activo.
- Está siendo puesto a prueba constantemente por la Mozilla Foundation.
- Es soportado por un equipo excelente de desarrollo.
- Tiene incontables características de las que la mayoría de las soluciones costosas carecen.

Al estar basada en web y ser open source (código abierto) se ha convertido en la herramienta de seguimiento de errores elegida por muchos proyectos, tanto en código abierto como no libres. Las muchas compañías que la utilizan se han percatado que éste sistema ayuda a reducir el tiempo muerto, aumenta la productividad, eleva la satisfacción del cliente, y mejora la comunicación. Bugzilla puede ser usado para cualquier organización, sin importar su tamaño.

Bugzilla utiliza un servidor HTTP (como puede ser Apache) y una base de datos (normalmente MySQL) para llevar a cabo su trabajo. Los errores pueden ser enviados por cualquiera, y pueden ser asignados a un desarrollador en particular. Cada error puede estar en diferentes estados, así como ir acompañado de notas del usuario o ejemplos de código que ayuden a corregir el error.

Otros sistemas conocidos son:

**BUGS - the Bug Genie.** Desarrollado por Ruffdogs, fácil de usar (por ejemplo un error puede reportarse en tres o cuatro pasos), opensource, seguro, con historial sobre los errores, permite adjuntar ficheros a los reportes de errores. Además posee la capacidad de hacer estos reportes según diversos criterios de búsqueda del error: por producto al que pertenece, fecha en que se reportó, estado en que se encuentra, desarrollador al que se le asignó corregirlo y cadena de texto relacionada con el mismo. Este sistema además tiene servicio de mensajería privada, temas y multilinguaje, entre otras características. Está

construido con tecnologías open source. Requiere un servidor web (se recomienda Apache), con PHP 4.0 ó superior instalado y MySQL.

**GNATS.** Proyecto GNU, soporta distintas bases de datos, además existe una comunidad que desarrolla nuevas extensiones. Propone herramientas para el seguimiento de los errores reportados por los usuarios a un sitio central. GNATS almacena toda la información sobre reportes de errores en sus bases de datos y proporciona las herramientas para consultar, editar, y dar mantenimiento a las mismas. Gracias a su arquitectura, GNATS no está limitado a una sola interfaz de usuario - puede ser utilizado vía la línea de comando, E-mail o Emacs, generalmente es usada con una interfaz Web. Básicamente, si las herramientas de los GNATS no proporcionan todo usted necesita, usted puede agregar sus propias utilidades adicionales usando las herramientas estándares del GNU.

**CodeTrack.** Es un sistema para el seguimiento de errores cuya característica principal es que no necesita ni base de datos ni servidor de correo (algo que puede ser una ventaja dependiendo de los recursos que podemos obtener en un proyecto). La información la guarda en XMLs y tan solo es necesario tener instalado Apache y PHP. De fácil instalación, funciona en navegadores antiguos, seguro, puede realizar el seguimiento de errores de varios proyectos a la vez, es muy rápido (busca cientos de errores en segundos), guarda el historial de cada error, en los reportes de errores se pueden incluir capturas de pantalla u otros adjuntos. Incluye una autenticación robusta, y permite el acceso personalizado a proyectos individuales. El sistema corre de igual manera en Linux y Windows. Estas y otras características hacen que sea una aplicación sencilla y útil.

**Mantis.** Sistema de seguimiento de errores que tiene entre sus características el ser fácil de instalar, permitir varios proyectos, varios lenguajes, emails, búsquedas simples y por filtros. Este sistema corre sobre runs on Windows, MacOS, OS/2, y una variedad de sistemas operativos de Unix. El Mantis requiere la instalación de los siguientes software: PHP 4.0.6 ó superior, servidor de base de datos (MySQL 3.23.2 ó superior o Postgree 7.0 ó superior) y servidor web (puede ser Apache o Internet Information Server, preferible la versión 1.3 ó superior y la 6.0 ó superior respectivamente, que son las que están probadas). En el caso de usar MySQL es imprescindible tener conocimientos de Administración Básica de MySQL, dígame que deben dominarse los siguientes aspectos: Autenticación, Crear nuevos usuarios y Dar permisos a los usuarios. Mantis es un software libre. Está escrito en PHP. Casi cualquier navegador puede funcionar como un cliente. Fue lanzado bajo términos de la licencia GPL.

**Bug-a-Boo.** Una Web basada en un sistema de reportes de errores, pero solo para Linux, liberado bajo los términos de GNU General Public License (GPL). No requiere de ningún servidor de base de datos, pero solo brinda acceso a sus tablas locales a través de tdbengine. Soporta varios lenguajes, tiene 3 niveles de usuario, ofrece notificaciones de email, y perfiles individuales de exhibición para cada usuario. Prácticamente no tiene limitaciones en cuanto al número de errores que puede gestionar.

**PR-Tracker.** Tiene todas las funciones prácticas de un sistema de seguimiento de errores. Diseñado para su utilidad. Es intuitivo y simple. Ofrece clasificación de errores, posibilita asignarlos; ordenarlos, buscarlos y reportarlos, también mantiene un control de accesos, permisos de usuarios, notificación del correo electrónico, se puede hacer el seguimiento de los errores de varios proyectos a la vez. Puede ser configurado para guardar datos en Acces 2000 ó en SQL Server. Crea y guarda consultas de gran alcance utilizando los operadores AND, OR y NOT. Permite la copia de uno o de un grupo de errores detectados de un proyecto a otro. Permite adjuntar ficheros a los reportes de errores. (No es gratis)

**Bug Traker.** Es un sistema de seguimiento de errores que permite registrar y gestionar los errores detectados durante el desarrollo del software. Permite asignar, organizar y llevar el control de los errores detectados y los que ya han sido resueltos. Este sistema permite el acceso simultáneo por varios usuarios. Su uso y mantenimiento son sencillos. Ofrece un ambiente de trabajo flexible. La búsqueda rápida le permite encontrar sus reportes fácilmente. Posee notificación automática de correo. La asignación automática o manual de las tareas le da más flexibilidad. Permite adjuntar ficheros a los reportes de errores. Posee una potente búsqueda con filtros que hace uso de los operadores lógicos AND, OR y NOT. Soporta un ilimitado número de reporte de errores.

En la Universidad de las Ciencias Informáticas no se utiliza ningún sistema que automatice los procesos de calidad, mucho menos el seguimiento de errores. En Cuba es muy poca la información relacionada con la existencia de herramientas de este tipo, aunque se cuenta con algunas que, entre otras funcionalidades, notifican errores, ya sea de forma automatizada o informatizada, estas constituyen incipientes aproximaciones al tipo de sistema que constituye el centro del presente trabajo de diploma.

**WGestNefro:** No es un sistema desarrollado propiamente para el seguimiento de errores. Es una aplicación elaborado por la Empresa Softel, para el Instituto Nacional de Nefrología con la finalidad de encontrar el receptor idóneo para un órgano (riñón) dado. Esta aplicación incluye una opción de notificación de errores a su desarrollador, de forma automática, siempre que el sistema da error, se

envía sin mediar usuario, un correo al desarrollador, conteniendo como texto el error y el lugar en que se encontraba el sistema ejecutando (la traza o camino).

Los otros sistemas que incluyen notificación de errores no lo hacen de forma automática, sino que son portales de servicios o Call Center, donde se reportan problemas (no necesariamente errores de software) por parte de los clientes y se organiza la atención a los mismos. Las empresas que poseen estos portales son:

- GET (Empresa del MINTUR, encargada de la Informática y las Comunicaciones)
- Copextel
- Correos de Cuba
- ETECSA
- Cubacel
- Desoft, etc

La Universidad de las Ciencias Informáticas como máximo exponente del desarrollo tecnológico del país, ha tenido entre sus prioridades la de convertirse en una gran industria del software y para ello, como antes se ha mencionado, es preciso el desarrollo de productos software con una alta calidad. Al no existir ninguna herramienta encargada de automatizar dicho proceso, surge la necesidad de desarrollar un sistema informático que, aprovechando las tecnologías recientes, logre automatizar el proceso de seguimiento de errores.

No se optó por la selección de ninguna de las herramientas estudiadas anteriormente para la automatización de dicho proceso, pues ninguno de estos sistemas responden a las características particulares del proceso que se lleva a cabo en la Facultad o sea, no solucionan las necesidades del proceso de seguimiento de errores en la Facultad 7, ya que éste tiene características particulares que requieren la elaboración de un sistema informático especial para su automatización, teniendo en cuenta que se prevé además que dicho sistema automatice algunos aspectos fuera del proceso, pero que pertenecen también a la Gestión de la Calidad.

### **1.3 Tecnologías actuales a considerar**

#### **1.3.1 ¿Qué es Internet?**

Algunos definen Internet como "La Red de Redes", y otros como "La Autopista de la Información".

Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores. Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información.

Es una red de alcance mundial que une una gran cantidad de redes de ordenadores. Se ha establecido que en Internet, toda la información ha de ser transmitida mediante el Protocolo TCP/IP.

Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras.

Los servicios más usados en Internet son: Correo Electrónico, World Wide Web, FTP, Grupos de Noticias, IRC y Servicios de Telefonía. (MOREA 1997)

Internet es el mayor conjunto que existe de información, personas, ordenadores y software funcionando de forma cooperativa, publicando y organizando información, e interactuando a nivel global. (*¿Qué es Internet?* 2005)

#### **1.3.2 Aplicaciones Web. Ventajas y Carencias**

En ingeniería de software una aplicación Web es aquella que los usuarios usan accediendo a un servidor Web a través de Internet o de una Intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. (*Aplicación Web* 2006)

Ventajas:

- Compatibilidad multiplataforma. Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías

incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.

- **Actualización.** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).
- **Inmediatez de acceso.** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria.** Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs.** Las aplicaciones basadas en Web deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en Web deberían tener mucho menos bugs que el software de escritorio descargable tradicional.
- **Precio.** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Los datos también van online.** Por supuesto con el desplazamiento de las aplicaciones locales a aquellas basadas en web también los datos que se crean y a los que se acceden van a necesitar experimentar profundos cambios. A nadie le gusta no poder acceder a su propio e-mail cuando está de

viaje, o poder recuperar un documento particular cuando se conecta desde un cyber café a 15.000 kilómetros de su oficina "Los clientes no deberían almacenar datos; deberían ser como teléfono. De hecho pueden convertirse en teléfonos, o viceversa. Y a medida que los clientes se hagan más pequeños, usted tiene otra razón para no tener sus datos en ellos: algo que usted lleva consigo pueden perderse o ser robado" (Source: Paul Graham - The Road Ahead, 2001)

- Múltiples usuarios concurrentes. Las aplicaciones basadas en Web pueden realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. Las compañías de conferencia Web y colaboración online están involucradas en algunas transformaciones claves y los usuarios necesitan explorar que significa realmente trabajar efectivamente y co-editar documentos juntos.
- Los datos son más seguros. Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se haga cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que provee aplicaciones basadas en Web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. Usted puede imaginar que si una compañía comercial pierde los datos de la gente será puesta de rodillas (financieramente) en cuestión de días.
- Desarrollar aplicaciones en el lenguaje que usted quiera. Una vez que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos, pueden también ser escritas en prácticamente cualquier lenguaje de programación. Ya que las aplicaciones Web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación que esté por ahí. Mientras que para software escritorio usted está limitado a usar el mismo lenguaje que el sistema operativo subyacente este no es el caso cuando la aplicación de software es independiente del sistema operativo. (CORSI 2006)



Las carencias:

1. Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
2. La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.
3. Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
4. Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones. (*Ventajas e inconvenientes de las aplicaciones web* 2005)

### **1.3.3 Servidores Web**

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. (VEGAS 2002)

Los servidores Web más difundidos a nivel mundial son: Apache e Internet Information Server (IIS)

#### **Apache**

Apache, sustancialmente, es un proyecto nacido para crear un servidor de web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código ya existente y de una serie de patch para mejorar su fiabilidad y sus características; de ahí su nombre. (*¿Qué es Apache?* 2006)

Apache funciona en casi todas las plataformas actuales. Debido a esto se puede escoger la plataforma que más se adapte a las características de cada cual, y también podemos cambiar de plataforma si en un momento determinado una plataforma ofrece más ventajas que la que se está utilizando.

Apache permite la autenticación de usuarios en varias formas. Además, permite el uso de bases de datos DBM para la autenticación de usuarios. De esta forma se puede restringir el acceso a determinadas páginas de un sitio Web de una forma sencilla y de fácil mantenimiento.

Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

Apache se distribuye bajo una licencia especial Apache Software License. Los binarios y el código fuente de Apache se pueden usar y distribuir de forma libre y en las condiciones mencionadas en la licencia anterior.

Según estudios realizados por diferentes empresas el servidor Web más utilizado en Internet es Apache. (*Apache* 2006).

## **IIS**

Servidor conectado a Internet que puede contener páginas web, servicio de correo electrónico etc. (*Diccionario Casadomo* 2006)

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un computador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web). (*Internet Information Services* 2007)

### **1.3.4 Lenguajes de Programación Web**

Existen numerosos lenguajes de programación utilizados para el desarrollo de Aplicaciones Web. Podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el

desarrollo de páginas dinámicas son el ASP, PHP y PERL. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento. (*Lenguaje de lado servidor* 2006) (*Lenguajes de lado servidor o cliente* 2006)

## **PHP**

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es un acrónimo recurrente que significa "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado, usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). La versión más reciente de PHP es la 5.2.0 (Noviembre 2006).

Los principales usos del PHP son los siguientes: Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión. Programación en consola, al estilo de Perl o Shell scripting. Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

Ventajas de PHP:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel. (*PHP* 2006)

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y php.
- La orientación a objetos es aún muy deficiente para aplicaciones grandes. (*Ventajas y desventajas de PHP 2006*)

## ASP

**Active Server Pages (ASP)** es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS).

ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP clásico.

En el último ASP clásico, ASP 3.0, hay seis objetos integrados disponibles para el programador, Application, ASPError, Request, Response, Server y Session. Cada objeto corresponde a un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas web dinámicas.

Las páginas pueden ser generadas mezclando código de scripts del lado del servidor (incluyendo acceso a base de datos) con HTML y código del lado del servidor.

Desde 2002, el ASP clásico está siendo reemplazado por ASP.NET, que, entre otras cosas, reemplaza los lenguajes interpretados como VBScript o JScript por lenguajes compilados a código intermedio (llamado MSIL o Microsoft Intermediate Language) como Visual Basic, C#, o cualquier otro lenguaje que soporte la plataforma .NET. (*ASP 2006*).

ASP.net trae diversas mejoras entre las cuales se destacan:

- Rendimiento: la aplicación se compila en una sola vez al lenguaje nativo, y luego, en cada petición tiene una compilación Just In Time, es decir se compila desde el código nativo, lo que permite mucho mejor rendimiento. También permite el almacenamiento del caché en el servidor.

- Rapidez en programación: mediante diversos controles, podemos con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.
- Servicios Web: trae herramientas para compartir datos e información entre distintos sitios.
- Seguridad: tiene diversas herramientas que garantizan la seguridad de nuestras aplicaciones. (FAHNLE 2006)

**ASP.NET** introduce el concepto del code-behind, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el de código. Con ello se facilita la programación de aplicaciones en múltiples capas, lo que en definitiva se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado. Por tanto, cualquier cambio drástico de especificaciones minimiza los cambios en la aplicación y maximiza la facilidad de mantenimiento. Asimismo, ASP.NET nos sirve tanto para Webs sencillas como para grandes aplicaciones.

Ventajas:

- Mayor velocidad, mayor potencia, mayor seguridad, mayor facilidad de mantenimiento y herramientas de trabajo.
- Caché: se puede almacenar en la caché del servidor tanto páginas enteras, como controles personalizados o simples variables. En páginas críticas con mucha carga de base de datos nos es muy útil almacenar datos de la base de datos en la caché, reduciendo enormemente el consumo de recursos.
- Carpetas especializadas, como por ejemplo app\_code que compila automáticamente las clases que se alojan en él, o la carpeta app\_theme que alojan ficheros que marcan los temas de estilos de la Web.
- Los archivos de configuración Web.config y Machine.config permiten realizar operación de configuración en ficheros que hasta ahora había que realizar en el servidor.
- La adaptación automática del código devuelto a los dispositivos que le acceden. Una misma página puede servirnos para el Internet Explorer, para el Pocket Internet Explorer desde una PDA o para un navegador de un móvil cualquiera.
- La eliminación total de la necesidad de frames con la introducción de las masterpages.
- La extraordinaria compatibilidad con XML y los servicios Web.

- La multitud de controles Web que permiten mucha funcionalidad con poco código. Desde enlace con las bases de datos o enseñar fácilmente todos los datos, hasta simples etiquetas, hiperenlaces o generadores de imágenes.
- Se puede utilizar hasta cuarenta lenguajes distintos para el desarrollo en ASP.NET, aunque en el 95% de las aplicaciones se usa C#, VB.NET o J#. (*ASP.NET 2006*)

### **C#**

C# es un lenguaje de propósito general orientado a objetos creado por Microsoft para su plataforma .NET. C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi. De hecho, su creador Anders Heljsberg fue también el creador de muchos otros lenguajes y entornos como Turbo Pascal, Delphi o Visual J++. La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic, y que además la migración a este lenguaje por los programadores de C/C++/Java sea lo más inmediata posible. (*Qué es C#. 2005*)

### **Visual Basic .NET**

Visual Basic .NET, la última generación del lenguaje Visual Basic, es un modo rápido y fácil de crear aplicaciones .NET, como servicios Web XML y aplicaciones Web.

Visual Basic.NET ofrece numerosas características nuevas y mejoradas, como herencia, interfaces y sobrecarga, que lo convierten en un eficaz lenguaje de programación orientado a objetos. Otras características nuevas del lenguaje son el subprocesamiento libre y el control de excepciones estructurado. Visual Basic.NET integra completamente el entorno .NET Framework y Common Language Runtime, que proporcionan interoperabilidad entre lenguajes, recolección de elementos no utilizados, seguridad mejorada y mayor compatibilidad entre versiones. (*Lenguaje Visual Basic 2007*).

### **Visual J# .NET**

Microsoft Visual J# .NET es una herramienta de desarrollo para programadores de Java que deseen generar aplicaciones y servicios en Microsoft .NET Framework.

Visual J# .NET proporciona la transición más fácil para desarrolladores de Java al mundo de los servicios Web XML y mejora enormemente la interoperabilidad de los programas en Java con el software existente escrito en otros lenguajes de programación

Visual J# .NET incluye tecnología que permite a los clientes migrar inversiones en Java a .NET Framework. Las aplicaciones existentes desarrolladas con Visual J++ pueden modificarse fácilmente para ejecutarse en .NET Framework, interoperar con otras aplicaciones y lenguajes para .NET e incorporar nueva funcionalidad .NET como Microsoft ASP.NET, Microsoft AD.NET y Microsoft Windows Forms. Además, los desarrolladores pueden utilizar Visual J# .NET para crear aplicaciones .NET completamente nuevas. (IVAR JACOBSON 2004).

## **PERL**

**Perl, Lenguaje Práctico para la Extracción e Informe** (Practical Extraction and Report Language) es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script. La estructura completa de Perl deriva ampliamente del lenguaje C. Perl es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas.

Perl es software libre y está licenciado bajo la GNU General Public License. Existen distribuciones disponibles para la mayoría de sistemas operativos. Está especialmente extendido en Unix y en sistemas similares, pero ha sido portado a las plataformas más modernas (y otras más obsoletas). (*Perl* 2006)

Algunas de las ventajas de Perl:

- Es un buen lenguaje "pegamento". Se pueden juntar varios programas de una forma sencilla para alcanzar una meta determinada. Los usuarios de Windows agradecerán esta propiedad ya que normalmente adolecen de un buen lenguaje tipo "script".
- Es relativamente rápido para un lenguaje tipo "script".



- Está disponible en múltiples plataformas y sistemas operativos. De hecho funciona bajo diferentes sabores de UNIX, Linux y todo tipo de Windows. Un programa que se escriba teniendo en cuenta la compatibilidad puede ser escrito en una plataforma y ejecutado en otra.
- El desarrollo de aplicaciones es muy rápido.
- Hay una colección enorme de módulos que pueden ser incorporados a cualquier "script" de Perl. Están disponibles en el CPAN ("Comprehensive Perl Archive Network"). En particular existe una extensión para cálculo numérico denominada PDL.
- Perl es gratuito. Mucho más que eso, es "Software Libre". Esto quiere decir que el código fuente está disponible para que cualquiera lo pueda ver o modificar, y lo que es más importante, siempre lo estará. Aunque nunca se pretenda cambiar el código, es importante disponer de la posibilidad de hacerlo, ya que siempre se podrá contratar a una tercera persona para que lo modifique en el caso de que haya un error, y debería ser posible solucionarlo.
- Le otorga al programador mucha libertad para que haga el programa como quiera. Tal como dice el eslogan de Perl "Hay más de una forma de hacerlo".

Algunas de las desventajas de Perl son:

- Es lento para algunas aplicaciones, como programación a bajo nivel, escribiendo un "driver" para una aplicación o corriendo modelos numéricos de cálculo intensivo. Si bien se pueden insertar subrutinas FORTRAN o C en Perl, teniendo lo mejor de los dos mundos, pero con algo más de complejidad.
  - La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible. Si no se escribe con cuidado puede llegar a ser difícil de leer.
  - No se pueden compilar programas Perl. Aunque actualmente se está desarrollando un compilador que realice esta tarea y hay uno comercial disponible para la plataforma Windows. Este tema no es tan crítico como suena, los programas Perl no correrán mucho más rápidos cuando se compilen, la única ventaja está en la desaparición de la fase inicial de compilación al correr la aplicación.
  - Utiliza muchos recursos de la máquina. Esto significa que no es tan ligero como un programa en C, pero en la práctica es ligero comparado con la potencia de computación de los ordenadores actuales.
- (Introducción a Perl)*

## JSP

**Java Pages Server (JSP)**, en el campo de la Informática, es una tecnología para crear aplicaciones Web. Es un desarrollo de la compañía Sun Microsystems, y su funcionamiento se basa en scripts, que utilizan una variante del lenguaje java.

La **JSP**, es una tecnología Java que permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML, o de otro tipo. Las JSP's permite al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento Web.

En las **JSP**, se escribe el texto que va a ser devuelto en la salida (normalmente código HTML) incluyendo código java dentro de él para poder modificar o generar contenido dinámicamente. El código java se incluye dentro de las marcas de etiqueta `<% y %>`, a esto se le denomina scriptlet.

La principal ventaja de **JSP** frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

Sin embargo **JSP** no se puede considerar un script al 100% ya que antes de ejecutarse el servidor Web compila el script y genera un servlet, por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

Debido a esto la tecnología JSP, así como Java está teniendo mucho peso en el desarrollo web profesional (sobre todo en intranets). (*JavaServer Pages Technology* 2006)

## HTML

El HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

## Java Script

Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Los navegadores más populares interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la the European Computer Manufacturers' Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen a ambas versiones con frecuencia incompatibles.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera versión 7, y Mozilla desde su primera versión. (*JavaScript 2006*).

### 1.3.5 Tecnología .NET vinculada con el paquete de Office

El Microsoft .NET Framework y sus bibliotecas de clases, tecnologías y lenguajes asociados solucionan muchas de las preocupaciones que los desarrolladores profesionales tenían con respecto al desarrollo de aplicaciones relacionadas con el Office. Actualmente el desarrollo de soluciones de este tipo se puede llevar a cabo con el Visual Studio 2005, el cuál constituye un rico ambiente de programación. Los desarrolladores pueden utilizar lenguajes .NET tales como Visual Basic .NET o C#. Además, las PIAs permiten al código .NET comunicarse con el modelo de objetos no administrado que cada una de las aplicaciones Office expone.

¿Qué son las PIAs o Primary Interop Assemblies?

Cuando el desarrollador necesita comunicarse desde .NET con un modelo de objetos Office, lo hace a través de una tecnología .NET llamada COM interop. Los modelos de objetos del Office están escritos en código no administrado (C y C++) y exponen interfaces COM. El código administrado (C# o Visual Basic) para interoperar con estas interfaces se comunica a través de un componente. Este componente no es más que un conjunto de clases .NET compiladas dentro de un ensamblado llamado Primary Interop Assembly o PIA.

¿Por qué Visual Studio Tools for Office?

El paquete Visual Studio Tools for Office (VSTO) añade al framework .NET soporte para programar utilizando Word, Excel, Outlook, e InfoPath en Visual Studio. VSTO convierte los documentos de Word o Excel en programables utilizando clases del framework, repletas de soporte para enlace de datos y controles que pueden ser usados en las aplicaciones desarrolladas. Permite integrar de forma sencilla código .NET con Outlook. Permite a los desarrolladores poner código de .NET detrás de las formas de InfoPath. Los desarrolladores pueden incluso programar contra los objetos dominantes de Office sin tener que atravesar el modelo del objeto de Office.

### 1.3.6 Navegadores o browsers

Un navegador Web, hojeador o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW) o Telaraña

Mundial. Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

La funcionalidad básica de un navegador Web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que este conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web). Tales documentos, comúnmente denominados páginas Web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación; que es de donde se origina el nombre de navegador. Por otro lado, hojeador es una traducción literal del original en inglés, browser, aunque su uso es minoritario. Otra denominación es explorador Web inspirada en uno de los navegadores más populares el Internet Explorer. (*Funciones del Navegador* 2006)

Dentro de los navegadores más utilizados se encuentran: Opera, Mozilla Firefox e Internet Explorer.

## **Opera**

Opera Browser o simplemente Opera es un navegador de Internet creado por la empresa noruega Opera Software en 1994. Es gratuito desde su versión 8.50.

Opera Browser es conocido por su velocidad, seguridad, soporte de estándares (especialmente CSS), tamaño reducido, internacionalidad y constante innovación. Fue uno de los primeros navegadores en implementar "pestañas" para la navegación de sitios web, siendo esta su principal característica en sus primeras versiones.

Sus mayores distribuciones son para Windows, Apple Macintosh y Linux, pero se encuentra disponible también para una variedad de otras plataformas, incluyendo OS/2, Solaris y FreeBSD.

Principales características de las últimas versiones de Opera (estas las poseen de la 8.5 en adelante):

- Cliente de correo integrado (llamado M2): Destaca por ser muy pequeño (ocupa 40 KB en el disco duro) y completo.

- Identificador (administrador de usuarios/contraseñas): Ofrece la posibilidad de recordar datos de acceso de cualquier sitio Web que así lo requiera, ya sea recordar datos sólo para la página actual, todo el servidor, no para la página actual o no para todo el servidor.
- Administrador de descargas: Permite, entre otras cosas, pausar cualquier descarga y continuarla cuando así se requiera (especialmente útil con archivos grandes), o realizar múltiples descargas simultáneamente llevando una mejor monitorización, así como poder llevar un historial de las mismas.
- Cliente de chat IRC.
- Navegación por sesiones: Permite guardar la sesión de navegación actual y volver a retomarla. Así, el usuario puede cerrar el navegador, y abrirlo más tarde teniendo exactamente las mismas páginas abiertas que cuando lo cerró.
- Administrador de cookies (huellas): Permite explorar/buscar las cookies por el sitio o usuarios del identificador, además de mostrar en detalle la información contenida en las mismas.
- Notas: Permite almacenar pequeños textos asociados a la página que se esté visitando.
- Soporte de estándares. Soporta CSS 2.1, XHTML 1.1, HTML 4.01, WML 2.0, ECMAScript, DOM 2 and SVG 1.1 tiny.
- Eliminar información privada. Permite eliminar con sólo un clic cookies, datos y páginas protegidos por contraseña, caché, historial de páginas visitadas, historial de direcciones escritas, historial de enlaces visitados, historial de archivos descargados y otros tipos de información privada.
- Navegación por voz: Sólo en inglés y para Windows 2000/XP.
- Navegación por medio de tabs (pestañas).
- Zoom en las páginas: Permite ver el contenido de cualquier página a distintas escalas; esto resulta útil para ver imágenes que sobrepasan las dimensiones del monitor, o páginas que son más anchas que la pantalla, por lo que resultan difíciles de leer en otros navegadores.
- Bloqueo de popups (ventanas emergentes). (*Opera 2006*)

## **Mozilla Firefox**

Mozilla Firefox es un navegador de código abierto, multiplataforma, con interfaz gráfica de usuario desarrollado por la Mozilla Corporation y cientos de voluntarios.

Firefox incorpora bloqueo de ventanas emergentes, navegación por pestañas, marcadores dinámicos, soporte para estándares abiertos, y un mecanismo para añadir funcionalidades mediante extensiones. Aunque otros navegadores también incluyen estas características, Firefox fue el primero en incluir algunas de ellas y conseguir una amplia difusión.

Firefox cuenta también con un Gestor de Descargas y con Herramientas de Desarrollo Web (un inspector DOM y una consola JavaScript.). Incluye 22 idiomas europeos, que permiten que los usuarios lo utilicen en su lengua materna. (esto no es textual)

Mozilla Firefox es un navegador multiplataforma que proporciona soporte a varias versiones de Microsoft Windows, incluyendo 98, 98SE, Me, NT 4.0, 2000, XP, y Server 2003. También funciona en Mac OS X y sistemas operativos basados en GNU/Linux que usen el sistema de ventanas X Window. Aunque no está oficialmente liberado para ciertos sistemas operativos, el código libre disponible, funciona en muchos otros sistemas, incluyendo pero no restrictivamente a: FreeBSD, OS/2, Solaris, SkyOS, BeOS y más recientemente a, Windows XP Professional x64 Edition. (*Firefox 2006*)

## **Internet Explorer**

Windows Internet Explorer (también conocido antes como Internet Explorer, IE o MSIE) es un navegador de Internet producido por Microsoft para su plataforma Windows y más tarde para Apple Macintosh. (*Internet Explorer 2006*)

Internet Explorer fue creado en el año 1995. Actualmente es uno de los navegadores que gozan de más popularidad en el mundo, constituyendo la razón principal de ello el hecho de que de hallarse preinstalado y configurado como navegador predeterminado en el sistema operativo Windows (incluido de fábrica en gran parte de computadores); buena parte de los usuarios no avanzados ni siquiera saben que haya alternativas.

El talón de Aquiles de este navegador es en la débil seguridad que lo caracteriza. La misma, en conjunto con su profunda integración en el sistema operativo ha comprometido en numerosas ocasiones la seguridad integral de la plataforma Windows.

### 1.3.7 Sistemas Gestores de Bases de Datos

Los Sistema de gestión de base de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y el usuario, las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. (*Sistema de Gestión de Base de Datos 2006*)

Existen distintos objetivos que deben cumplir los SGBD:

- Abstracción de la información
- Independencia del tratamiento de información
- Redundancia mínima
- Consistencia
- Seguridad
- Integridad
- Respaldo y recuperación de la información
- Control de la concurrencia
- Tiempo de respuesta mínimo

Entre los SGBD más utilizados actualmente se encuentran: SQLServer, MySQL, PostgreSQL y Oracle.

#### **SQLServer**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones
- Gran estabilidad
- Gran seguridad



- Escalabilidad
- Soporta procedimientos almacenados
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información
- Además permite administrar información de otros servidores de datos. (*Microsoft SQL Server 2006*)

La principal desventaja de SQLServer lo constituye el hecho de no ser multiplataforma, ya que solo corre sobre sistemas operativos de Microsoft.

## **MySQL**

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

Está desarrollado en su mayor parte en ANSI C.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL. Cada lenguaje utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. Es decir, MySQL puede combinarse con múltiples lenguajes para el desarrollo de aplicaciones.

MySQL es muy utilizado en aplicaciones Web, sobre todo en combinación con PHP. Una de sus principales características es la velocidad, ya que fue desarrollado teniendo en cuenta la misma como objetivo principal.

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.

- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto. (*MySQL 2006*)

Su principal desventaja la constituye el hecho de que hay que pagar licencia comercial si se pretende desarrollar con él software propietario.

### **PostgreSQL**

PostgreSQL es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD. (*PostgreSQL 2006*)

PostgreSQL tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos. Además corre en la mayoría de los Sistemas Operativos más utilizados incluyendo, Linux, varias versiones de UNIX, y Windows. Cumple la prueba ACID (Atomicidad, Consistencia, Integridad, Durabilidad) y tiene soporte completo para llaves foráneas, joins, vistas, subconsultas (incluyendo subconsultas en la cláusula FROM), triggers, y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos de los estándares SQL92 y SQL99. También soporta almacenamiento de objetos grandes (imágenes, sonido y video). Así, como sus propias interfaces de programación para C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, entre otros, y una documentación muy completa (aunque la documentación se encuentra en inglés hay un proyecto de traducción gestándose). (*¿Qué es Postgree? 2005*)

Aunque la velocidad de respuesta que ofrece este gestor con BD relativamente pequeñas puede parecer un poco deficiente, esta misma velocidad la mantiene al gestionar BDs realmente grandes, cosa que resulta loable.

La principal desventaja de Postgre es que tiene un consumo de recursos y una carga del sistema muy elevados. A cambio, debido a su arquitectura escala muy bien al aumentar el número de CPUs y la cantidad de RAM. (*Mercado tradicional desarrollado a través de Internet*)

## Oracle

Oracle es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Es multiplataforma.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, otros sistemas con licencia libre como PostgreSQL y MySQL. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). (Oracle 2006)

### 1.3.8 Entornos Distribuidos. Modelo Cliente-Servidor. Ventajas

La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificarlas.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En la funcionalidad de un programa distribuido se pueden distinguir 3 capas o niveles:

1. Manejador de Base de Datos (Nivel de almacenamiento),
2. Procesador de aplicaciones o reglas del negocio (Nivel lógico)
3. Interfaz del usuario (Nivel de presentación)

En una arquitectura monolítica no hay distribución; los tres niveles tienen lugar en el mismo equipo.

Ventajas de la arquitectura cliente-servidor:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión. (*Cliente-servidor 2006*)

### **El modelo cliente-servidor de dos capas**

En este modelo la lógica de las aplicaciones debe estar en el cliente o en el servidor, unida a la capa de presentación o a la de datos respectivamente. La comunicación con el servidor es transparente para el usuario.

Limitaciones:

- No es escalable
- No es manejable
- Bajo rendimiento
- Este modelo suele ser costoso de mantener y pesado de depurar.
- La lógica de la aplicación no puede ser reutilizada ya que está ligada o a la interfaz de usuario o al manejo de persistencia de datos.
- Incremento de la carga de la red en el caso de que la lógica de la aplicación esté ligada a la interfaz de usuario en el cliente: dado que el procesamiento de los datos se realiza en él, gran cantidad de información debe ser transmitida desde el servidor.
- Implica un proceso de distribución complicado en caso de que la lógica de la aplicación esté ligada a la interfaz de usuario en el cliente. Como regla general esto es mala idea ya que fuerza a que cada cambio en la aplicación lleve a ir usuario por usuario cambiándole la aplicación.

### **El modelo cliente-servidor de tres capas**

En esta arquitectura la lógica de la aplicación ocupa una capa intermedia; está separada tanto de los datos como de la interfaz de usuario. En teoría, los sistemas en 3 capas son de más fácil ampliación y

más robustos y flexibles. Además, pueden integrar datos de múltiples fuentes. Es importante notar que los límites entre las capas son lógicos. Cada capa está formada por componentes.

#### Responsabilidades de las capas

##### Capa de presentación

- Es responsable de la presentación de los datos, recibiendo los eventos de los usuarios y controlando la interfaz de usuario.

##### Capa de lógica de negocios

- Esta capa es nueva, es decir, no está presente en la arquitectura en 2 capas en forma explícita.
- Los objetos de negocios que implementan las reglas de negocios “viven” aquí, y están disponibles para la capa de presentación.
- Esta capa es la clave para resolver los problemas de la arquitectura en 2 capas.
- Protege del acceso directo a la información desde la capa de presentación.

##### Capa de persistencia

- Es responsable del almacenamiento de los datos.
- Es común reutilizar sistemas existentes de bases de datos en esta capa.
- Actualmente se usan manejadores relacionales: son avanzados, permiten el uso de triggers y paquetes. Existen manejadores Orientados a Objetos.

#### Ventajas:

- Separación clara de la interfaz de usuario de la lógica de la aplicación. Esta separación permite tener diferentes presentaciones accediendo a la misma lógica.
- La redefinición del almacenamiento de información no tiene influencia sobre la presentación.
- En contraste con una arquitectura en 2 capas, donde solamente datos están accesibles al público, los objetos de negocios pueden brindar servicios (lógica de la aplicación) por la red. (PEROVICH)

- Los componentes que se crean se pueden reutilizar en posteriores aplicaciones por lo que el tiempo en su desarrollo se aprovecha.
- Como es posible que los componentes de la aplicación se puedan ejecutar en diferentes máquinas, podemos distribuirlos según las necesidades en máquinas más potentes por lo que aumenta el rendimiento general de estas aplicaciones.
- Se puede subdividir una aplicación compleja en partes mucho más sencillas de entender y gestionar.
- Cuando existen necesidades de mantenimiento, las modificaciones a realizar se hacen solamente sobre determinados componentes, permaneciendo los restantes intactos por lo que el mantenimiento es más fácil. (GARCÍA)

Clasificación de los sistemas Cliente-Servidor de tres capas:

1. Representación distribuida.- La interacción con el usuario se realiza en el servidor, el cliente hace de pasarela entre el usuario y el servidor.
2. Representación Remota.-La lógica de la aplicación y la base de datos se encuentran en el servidor. El cliente recibe y formatea los datos para interactuar con el usuario.
3. Lógica Distribuida.- El cliente se encarga de la interacción con el usuario y de algunas funciones triviales de la aplicación. Por ejemplo controles de rango de campos, campos obligatorios, etc. Mientras que el resto de la aplicación, junto con la base de datos, están en el servidor.
4. Gestión Remota de Datos.- El cliente realiza la interacción con el usuario y ejecuta la aplicación y el servidor es quien maneja los datos.
5. Base de Datos Distribuidas.- El cliente realiza la interacción con el usuario, ejecuta la aplicación, debe conocer la topología de la red, así como la disposición y ubicación de los datos. Se delega parte de la gestión de la base de datos al cliente.
6. Cliente servidor a tres niveles.- El cliente se encarga de la interacción con el usuario, el servidor de la lógica de aplicación y la base de datos pueden estar cada uno en otro servidor. (JARA)

### 1.3.9 Metodologías de desarrollo de software, lenguajes y herramientas para el modelado

#### RUP

El Proceso Unificado Rational (Rational Unified Process) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos basado íntegramente en Lenguaje Unificado de Modelado UML como soporte a la metodología.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. (*Rational Unified Process(RUP)* 2006).

RUP define cuatro fases: Inicio, Elaboración, Construcción y Transición. Y nueve flujos de trabajo, seis de Ingeniería (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y tres de apoyo (Gestión de la Configuración, Gestión de Proyecto y Ambiente).

RUP pretende implementar las mejores prácticas actuales en Ingeniería de Software:

- Desarrollo iterativo del Software
- Administración de requerimientos
- Uso de arquitecturas basadas en componentes
- Modelación visual del software
- Verificación de la calidad del software
- Control de cambios. (GUERRERO)

#### Metodologías ágiles

Los procesos ágiles de desarrollo de software, conocidos también como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. (*Desarrollo ágil de software* 2007)

#### Programación Extrema

La programación extrema o Extreme Programming (XP) es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los

defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y facilidad de mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo. (*Programación Extrema 2007*)



- Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- 40 horas por semana: Se debe trabajar un máximo de 40 horas por semana. No se trabaja horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- Estándares de programación: XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. (JOSÉ H. CANÓS)

### **Scrum**

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos: 1) El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. 2) El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

### **Crystal Methodologies**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

### **Dynamic Systems Development Method7 (DSDM)**

Define el marco para desarrollar un proceso de producción de software. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone

cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

### **Adaptive Software Development (ASD)**

Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

### **Feature -Driven Development9 (FDD)**

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

### **Lean Development10 (LD)**

Utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios. (JOSÉ H. CANÓS)

### **UML**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar. (*Unified Modeling Language 2006*)

Ventajas de UML:

1. Nuestro sistema de software profesional es diseñado y documentado antes de que sea codificado. Sabremos exactamente los que conseguiremos, por adelantado.
2. Ya que el sistema del sistema está antes de crear la primera parte del código, es fácil descubrir el código reutilizable y tratarlo para una mejor eficacia.
3. Los lógicos 'agujeros' en el diseño del sistema podrán ser descubiertos antes sobre los diagramas del mismo. El software se comportará de la forma esperada y surgirán menos sorpresas.
4. El diseño total del sistema dicta el modo en que se desarrollará el software. Las decisiones finales se harán antes de que nos encontremos con código mal escrito. Con esto ahorraremos tiempo en nuestro desarrollo.
5. Cuando necesitemos hacer modificaciones en el sistema, es mucho más fácil hacerlo sobre la documentación UML. Hay que recurrir mucho menos a rehacer un nuevo estudio. Volvemos a ahorrar tiempo en nuestro desarrollo.
6. Si se incorporan nuevos desarrolladores al proyecto, los diagramas UML les permitirá hacerse rápidamente una idea del sistema.
7. La comunicación con nuestros desarrolladores, y con desarrolladores externos, es mucho más eficiente. (*UML-Análisis y diseño de Software*)

Una desventaja de UML es que no se presta con facilidad al diseño de sistemas distribuidos. En tales sistemas cobran importancia factores como transmisión, serialización, persistencia, etc. UML no cuenta con maneras de describir tales factores. (*Unified Modeling Language 2006*)

### **Rational Rose**

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo,

construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. (*Rational Rose: Procedimientos básicos para desarrollar un proyecto con UML*)

#### **1.4 Tecnologías y herramientas a utilizar**

El sistema deberá estar concebido bajo la metodología RUP que constituye la metodología estándar más utilizada para el diseño, implementación y documentación de sistemas y que se basa a su vez en UML (Lenguaje Unificado de Modelado). Además, es altamente configurable, ya que permite construir solamente los artefactos que se necesiten para el desarrollo de un producto software. Para documentar el desarrollo del software se utilizará la herramienta Rational Rose Enterprise Edition 2003 ya que esta herramienta es muy completa y ofrece amplias potencialidades.

Para la implementación, teniendo en cuenta las particularidades del sistema a desarrollar, específicamente la comunicación que debe tener con el Microsoft Word, se utilizará el Visual Studio .NET 2005, tecnología ASP.Net (dado que se trata de una aplicación web) con lenguaje de programación C#, incluyendo la herramienta Visual Studio Tools for Office Second Edition, la cual facilita altamente la comunicación con el paquete de Office.

Como servidor Web: Apache 2.0, por ser multiplataforma y garantizar que la aplicación a su vez lo sea; debe instalarse el módulo mod\_mono para asegurar que las paginas ASP.Net se ejecuten en dicho servidor web.

Sistema Gestor de Base de Datos: MySQL 5.0, por su rapidez cuando se trabaja con una base de datos pequeña, característica que se ajusta al sistema propuesto, además, consume pocos recursos.

Navegador: Internet Explorer 5.5 o superior. Mozilla Firefox 2.0 y Lenguaje del lado del cliente: JavaScript,

básicamente para validación de datos. La aplicación se sustentará en una Arquitectura Cliente – Servidor de tres capas.

Como resultado del análisis realizado durante el presente capítulo, se pudo concluir que: ninguno de los sistemas estudiados, relacionados con el seguimiento de errores, responden a las características de este proceso en la Facultad, por lo que se necesita diseñar un nuevo sistema que automatice el proceso mencionado. Se realizó un estudio de las principales tecnologías y herramientas existentes y se seleccionaron las más adecuadas para el desarrollo de la aplicación.

## **Capítulo 2: Características del Sistema.**

En el presente capítulo se hace alusión a los aspectos fundamentales relacionados con el objeto de estudio, se describe el flujo actual de procesos y se realiza un análisis crítico de la ejecución de estos, incluyendo toda la información que se maneja, o sea aquellos documentos que se manipulan durante el mismo, además, se plantea el objeto de automatización.

Se aborda el tema relacionado con la modelación del negocio, siendo este el primer flujo de trabajo durante el proceso de desarrollo de un software, el cual tiene entre sus objetivos comprender los procesos de negocio de la organización, identificar las mejoras potenciales y derivar los requerimientos del sistema que va a soportar la organización.

Se muestran los requisitos funcionales y no funcionales, además de la descripción de los actores y casos de uso resultantes del flujo de trabajo de Requisitos.

### **2.1 Objeto de estudio**

#### **2.1.1 Flujo actual de los procesos involucrados en el campo de acción.**

Durante el proceso de Control de la Calidad de los productos software desarrollados por los diferentes grupos de desarrolladores, la revisión del producto, que incluye el registro y seguimiento de los errores detectados a los diferentes elementos de configuración, es un aspecto esencial y de vital importancia.

Dicho proceso permite que los diferentes Grupos de Desarrollo de la Facultad 7 puedan certificar la calidad de sus productos. Tiene como punto de partida la entrega de un producto por parte de un Grupo de Desarrollo al Grupo de Calidad de la Facultad. Los miembros de este último entonces revisan el producto y registran las no conformidades encontradas. Luego entregan el Registro de No Conformidades al Grupo de Desarrollo, el cual corrige el software y la documentación y envía el Registro de No Conformidades (especificando cuáles fueron aceptadas) y el producto ahora corregido al Grupo de Calidad.

Este procedimiento se repite hasta que el grupo de Calidad valora que todas las NC han sido corregidas y elabora el Acta de Liberación Parcial, así como el documento Expediente del Producto y los envía a Calisoft, junto a toda la documentación referente al software y a todo el proceso de pruebas a que fue sometido recogida en el Expediente del Proyecto, si Calisoft acepta el Acta de Liberación Parcial y emite

una valoración positiva del proceso de revisión realizado, entonces elabora el Certificado de Liberación y se le hace llegar una copia al Grupo de Calidad y este a su vez otra al Grupo de Desarrollo.

En caso de que Calisoft rechace el Acta de Liberación Parcial, el Grupo de Calidad tiene que proceder a repetir todo el proceso de pruebas y de revisión de la documentación, luego realizar las actividades que se suceden, según la descripción anteriormente expuesta, este proceso termina una vez que el Acta de Liberación Parcial sea aceptada y se emitan los documentos antes especificados, significando esto que el producto esta liberado por tener la calidad requerida.

### **2.1.2 Análisis crítico de la ejecución de los procesos.**

Actualmente el registro y seguimiento de errores o No Conformidades detectados a un producto software por parte del Grupo de Calidad durante el proceso descrito con anterioridad, se realiza mediante documentos digitales, manipulados actualmente por el Microsoft Word del paquete de Office, lo cual provoca que en muchas ocasiones dicho proceso sea engorroso, haciendo que se consuma más tiempo y esfuerzo por parte de los miembros del Grupo de Calidad.

Además el proceso puede no resultar del todo satisfactorio, dado que se puede perder información, en caso de que se escapen errores sin corregir o no detectados. En el caso específico de las técnicas empleadas para la revisión de la documentación, estas no son factibles, se caracterizan por entorpecer el trabajo debido a su lentitud de utilización. A la larga, todos estos aspectos negativos pueden repercutir de forma notoria en la duración del período de prueba de un producto.

A pesar del uso de las herramientas para el control de versiones (fundamentalmente Subversion) en el Grupo de Calidad de la Facultad 7 las problemáticas antes planteadas subsisten, ya que estas herramientas no proveen los mecanismos para darles solución, solo se gana en cuanto a organización.

### **2.1.3 Información que se maneja.**

Los documentos fundamentales procesados son:

El Registro de No Conformidades, el cual incluye una lista de las No Conformidades detectadas durante las pruebas y describe las circunstancias en las que fueron generadas las mismas. Contiene aspectos que

permiten una mejor identificación y descripción de las No Conformidades, tales como el elemento de configuración a revisar, una breve descripción de la No Conformidad, una clasificación que permite dar una prioridad, un estado, que se debe determinar por parte del grupo de desarrollo, en dependencia de si es aceptada o no la No Conformidad; se especifica el cumplimiento, para el caso de si fue resuelta o no, y las observaciones, que no son más que especificaciones que ayuden a encontrar más rápido la no conformidad por parte de los desarrolladores, se recomienda en este punto incluir lugar donde fue encontrada, anexos que generalmente son imágenes del software con el error señalado.

El Acta de Liberación Parcial, se elabora por parte del grupo de calidad para dar una liberación parcial del producto. En esta se incluyen: los elementos revisados y resultados finales, los elementos que no se pudieron revisar y las causas, entregas del laboratorio a los desarrolladores, condiciones de Trabajo existente, No Conformidades aún presentes en el producto. Además, incluye Recomendaciones Importantes para los Desarrolladores, así como la firma de los involucrados en el proceso.

El Certificado de Liberación es el documento legal que avala la calidad de un producto software, es emitido por Calisoft. En este documento se registran los siguientes datos: a quién se le otorga el certificado, las evaluaciones a las cuales se sometió el producto, el nombre del producto que es certificado, la fecha de inicio y de fin del período de evaluación y el número del expediente en el cuál constan estos datos, el nombre del Plan de Pruebas que se siguió para la evaluación, así como los métodos de pruebas que se utilizaron, la fecha en que se expide el certificado y la firma de quien lo expide (Director del Centro Nacional de Calidad de Software).

#### **2.1.4 Objeto de automatización.**

El sistema propuesto formará parte del Portal de Grupo de Calidad de la Facultad 7, constituyendo un módulo del mismo. Este surge para: permitir a los proyectos de la Facultad conocer la documentación a entregar para solicitar un análisis de calidad del software, así como el estado del proceso una vez que se haya iniciado. Dicho portal agilizará la entrega y liberación de los productos y además contribuirá a complementar la formación de los miembros del Grupo de Calidad de la Facultad, debido fundamentalmente, a la abundante documentación online con la que podrán contar.



La razón de ser del presente trabajo de diploma, es el diseño de un módulo del Portal que debe automatizar el registro de No Conformidades detectadas a los productos durante el Proceso de pruebas.

Debe permitir a los probadores del Grupo de Calidad registrar las No Conformidades detectadas, tanto del software como de la documentación, esta última deberá ser revisada desde el propio sistema. Además de permitir a los probadores almacenar toda la documentación correspondiente al proceso de pruebas, entiéndase plan de pruebas, listas de chequeo, diseños de casos de prueba, por mencionar algunos.

Posibilitará al líder de un proyecto acceder al registro de No Conformidades de cada uno de sus productos y visualizarlas, de esta manera podrá decidir cuáles de ellas acepta o no. También debe permitir a cualquier miembro de su proyecto visualizar las No conformidades que fueron aceptadas de un producto en específico y corregir la documentación, accediendo a la misma desde el propio sistema, donde deberá aparecer una lista de los errores que le fueron detectados.

Mostrará reportes estadísticos acerca del comportamiento de las No Conformidades a lo largo de todo el proceso, para uso fundamentalmente de los líderes de calidad y directivos de la Facultad y se informatizará la elaboración del Acta de Liberación del Producto y del Certificado de Liberación.

## 2.3 Modelo de negocio.

### 2.3.1 Actores del Negocio:

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actores del negocio	Justificación
Cliente	Agente externo al negocio, que interactúa con él y se beneficia del mismo. Está interesado en que su proyecto sea revisado con vistas a ser certificado. (líderes y miembros en general de los proyectos)

Tabla 2.1 -Descripción de los actores del negocio.

### 2.3.2 Trabajadores del Negocio.

Un trabajador del negocio representa a personas o sistemas dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso.

Estos trabajadores están dentro de la frontera del negocio, son los que en un futuro se convertirán en usuarios del sistema que se quiere construir.

Trabajadores del negocio	Justificación
Probador	Es quién se encarga de todo el proceso de detección, registro y control de las no conformidades del producto. (Miembros del grupo de calidad).
Calisoft	Es la organización encargada de emitir o no el certificado de Liberación que avala legalmente la calidad de un producto

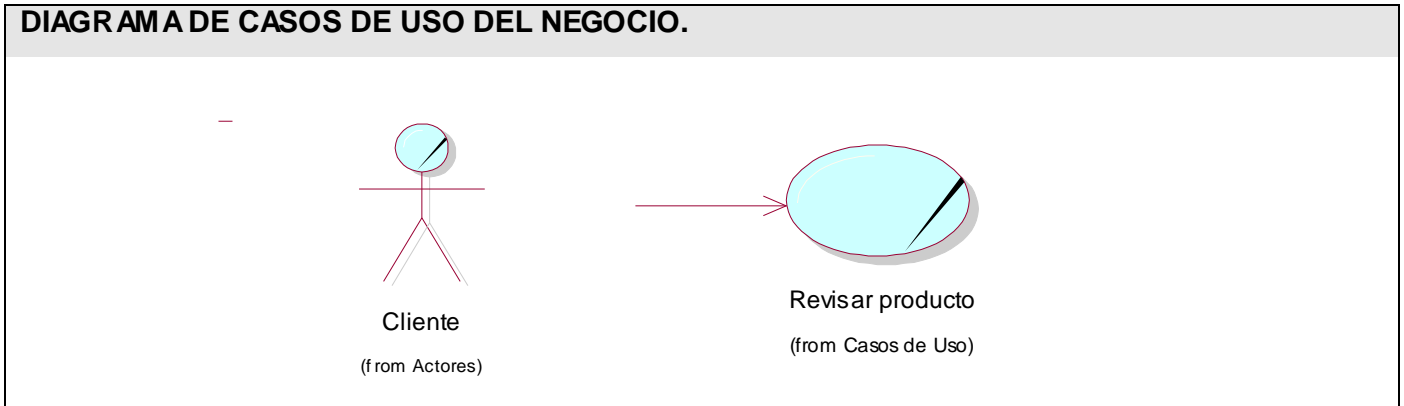
Tabla 2.2 -Descripción de los trabajadores del negocio.

### 2.3.3 Diagrama de Casos de Uso del Negocio.

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

Para el caso específico del presente trabajo, se define como proceso observable, la revisión del producto que produce como resultados más relevantes el Registro de las No Conformidades, así como la elaboración del Acta de Liberación Parcial y el Certificado de Liberación.

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.



**Fig. 2.1 -Diagrama de Casos de Uso del Negocio.**

**2.3.4 Descripción Textual de los Casos de Uso del Negocio.**

<b>Caso de Uso:</b>	<b>Revisar Producto</b>
<b>Actores:</b>	Cliente (inicia)
<b>Trabajadores:</b>	Probador y Calisoft
<p><b>Resumen:</b> El caso de uso se inicia cuando el cliente le entrega su producto al equipo de calidad para que lo revise, este detecta las no conformidades, el cliente las acepta o no para luego corregir las aceptadas, luego se verifica si estas fueron corregidas y se repite el ciclo hasta que Calisoft emite un Certificado de Liberación que avala legalmente la calidad del producto.</p>	
<b>Precondiciones:</b>	El cliente debe entregar el sistema y la documentación.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Cliente entrega el sistema y la documentación.	1.1. El Probador diseña los casos de prueba. 1.2. El Probador revisa el software y la documentación. 1.3. El Probador registra y clasifica las no conformidades detectadas en el Registro de No Conformidades. 1.4. El probador envía al cliente el Registro de No Conformidades.
2. El cliente revisa el Registro de no	

Conformidades para aceptarlas o no. Acepta alguna(s) no conformidad(es)	
3. El Cliente corrige el software y la documentación.	
4. El cliente envía registro de NC actualizado y producto corregido.	<p>4.1. El Probador verifica que todas las No Conformidades detectadas fueron corregidas y que no existen nuevas. Actualiza el Registro de No Conformidades.</p> <p>4.2. El probador elabora el Acta de Liberación Parcial del producto y el documento Expediente del Producto.</p> <p>4.3. El Probador envía el Acta de Liberación Parcial del Producto, el Expediente del Producto y el Expediente del Proyecto (toda la documentación relativa al producto) a Calisoft.</p> <p>4.4. Calisoft revisa el Acta de Liberación Parcial del Producto y la documentación necesaria. Acepta el Acta de Liberación Parcial del Producto.</p> <p>4.5. Calisoft emite un Certificado de Liberación.</p> <p>4.6. Calisoft envía copia del Certificado de Liberación al Probador.</p> <p>4.7. El Probador recibe copia del Certificado de Liberación.</p> <p>4.8. El probador envía una segunda copia del Certificado de Liberación al cliente.</p>
5. El cliente recibe el Certificado de Liberación.	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
Actividad 2. Si el Cliente no acepta alguna no conformidad entonces envía el Registro de No Conformidades al probador y se pasa al paso 4.2	

	4.1. Si aún hay No Conformidades sin corregir entonces se envía el Registro de No Conformidades actualizado al cliente y se pasa al paso 2.
	Actividad 4.4. Si Calisoft no acepta el Acta de Liberación Parcial del producto entonces envía el Acta con señalamientos al Probador, este la recibe y pasa al paso 1.1.
<b>Poscondiciones</b>	El cliente obtiene el Certificado de Liberación.

Tabla 2.3 -.Descripción del Caso de Uso del Negocio: Revisar producto.

### 2.3.5 Diagramas de Actividades.

Los casos de uso del negocio consisten en secuencias de actividades que, en conjunto, producen algo para el actor del negocio. El proceso (workflow) consiste de un flujo básico de una o más alternativas de flujos. La estructura del flujo se describe gráficamente con la ayuda de un diagrama de actividad. (Ver Anexo 2)

### 2.3.6 Diagrama de clases del modelo de objetos.

Muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. (Ver Anexo 3).

## 2.4 Especificación de los requisitos de software.

### 2.4.1 Requisitos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, especifican acciones que el sistema debe ser capaz de realizar. Deben ser comprensibles por clientes, usuarios y desarrolladores, deben tener una sola interpretación y estar definidos en forma medible y verificable. A continuación se muestran los requisitos que debe cumplir el sistema:

R1- Gestionar Registro de No Conformidades.

R1.1- Registrar NC detectadas.

R1.2- Listar NC por por elementos de configuración de producto determinado.

R1.2.1- Listar NC aceptadas.

R1.2.2-Listar NC por clasificación.

R1.3- Mostrar detalles de las NC.

R1.4- Modificar una NC.

R1.5- Eliminar una NC.

R1.6- Seleccionar NC que proceden (aceptadas).

R1.7- Corregir NC de la documentación.

R2- Gestionar Acta de Liberación Parcial.

R2.1- Crear Acta de Liberación Parcial.

R2.2- Modificar Acta de Liberación Parcial.

R2.3- Imprimir Acta de Liberación Parcial.

R2.4- Buscar Acta de Liberación Parcial.

R3- Gestionar Certificado de Liberación.

R3.1- Crear Certificado de Liberación.

R3.2- Modificar Certificado de Liberación.

R3.3- Imprimir Certificado de Liberación.

R3.4- Buscar Certificado de Liberación.

R4- Gestionar Estadísticas de NC.

R4.1- Graficar Estadísticas

R4.2- Mostrar Estadísticas.

R4.3- Imprimir Estadísticas.

R5- Subir documentación.

## **2.4.2 Requisitos No funcionales:**

### **Interfaz Externa:**

- El sistema tendrá una interfaz de fácil navegación.
- El sistema tendrá una interfaz sencilla, ya que no estará cargado de imágenes ni colores, se usarán colores oscuros para los textos y colores tenues para los fondos de páginas, dado que es un sistema con fines de trabajo.
- Estará diseñado para una resolución de 1024 x 768 píxeles.

### **Usabilidad:**

- El sistema podrá ser utilizado por personas que tengan los conocimientos básicos de computación, lo cuál es suficiente, ya que los usuarios finales del mismo son estudiantes o profesionales de la informática (el sistema esta destinado principalmente a directivos, grupo de calidad y proyectos productivos de la Facultad 7, aunque puede ser utilizado de forma mas restringida por cualquier otro miembro de la comunidad universitaria).
- El sistema contará con una ayuda online que siempre estará disponible para facilitar el trabajo de los usuarios.
- Estará disponible 24 h al día por lo que podrá ser utilizado en cualquier momento.

### **Rendimiento:**

- Las páginas del sistema estarán poco cargadas de imágenes para garantizar que el tiempo de ejecución de los hipervínculos, las búsquedas, las actualizaciones, las adiciones y eliminaciones no excedan de los 5 segundos.

**Soporte:**

- El sistema tendrá una ayuda online.
- El sistema estará bien documentado, de forma tal que se puedan realizar futuros mantenimientos del mismo.

**Portabilidad:**

- El sistema podrá ejecutarse sobre Linux o Windows.

**Seguridad:**

- El uso de un sistema manejador de contenido (CMS) garantiza la seguridad del Portal.
- El acceso a la información será limitado. Teniendo en cuenta que a cada usuario se le permitirá hacer solo las operaciones que le corresponden según su rol, que es el que define su nivel de acceso.

**Políticos-culturales:**

- El sistema sólo podrá ser utilizado dentro de la UCI.
- No debe contener palabras en otros idiomas.

**Software:**

Para el cliente:

- Navegador Mozilla Firefox 2.0 ó Internet Explorer 5.5 o superior.
- Sistema operativo Linux o Windows 98 ó Superior.

Para el servidor:

- Sistema operativo Linux o Windows Advancer Server 2000 ó superior.
- Servidor Web Apache 1.13.19 o con módulo mod\_mono y PHP 4.2.x o superior instalados
- Microsoft .NET Framework 2.0 (si el sistema operativo es Windows) o Mono (si el sistema operativo es Linux).
- Servidor de Base de Datos MySQL 3.23.x o superior



### **Hardware:**

Requerimientos mínimos:

Para el cliente:

- Procesador Pentium III a 1GHz de velocidad de procesamiento y 128 Mb de memoria RAM.
- Monitor VGA o superior.
- Tarjeta de red.

Para el servidor:

- Procesador Intel Pentium Dual Core a 3.0GHz de velocidad de procesamiento y 2Gb de memoria RAM.
- 40Gb de espacio libre en disco.
- Tarjeta de red.

### **Confiabilidad:**

- Se hará una transformación correcta del diseño a un lenguaje de programación, garantizando así que el usuario final pueda contar con las funcionalidades que espera del sistema.
- Deben montarse sistemas de respaldo eléctrico en los locales donde se encuentren los servidores para garantizar que los servicios estén disponibles en todo momento.

### **Restricciones para el diseño y la implementación:**

- Utilizar los estándares de diseño establecidos.
- Para la implementación del sistema se deberá utilizar PHP 5.0 y para el módulo Gestión de No Conformidades, tecnología ASP.Net con lenguaje de programación C#. Las validaciones del lado del cliente se harán a través de código Java Script.
- El SGBD a utilizar deberá ser MySQL 5.0.

## 2.5 Definición de los casos de uso.

### 2.5.1 Definición de los actores

Actores	Justificación
Miembro de proyecto	A través del sistema visualiza de forma detallada las no conformidades que le fueron detectadas a su producto y fueron aceptadas por su Líder de proyecto (lo cuál le facilita su corrección). Corrige de forma automatizada los errores de la documentación en el mismo sistema.
Líder de proyecto	Tiene la posibilidad aceptar o rechazar las No Conformidades que se le hayan detectado a su producto.
Probador	Es el encargado de registrar a través del sistema todos los errores que se le detectan al producto que se encuentra en revisión, así como de mantener el control acerca del estado de los mismos. Obtiene las estadísticas que necesita sobre los errores detectados y puede gestionar la creación y modificación del Acta de Liberación Parcial de un producto dado.
Calisoft	Es el encargado de gestionar a través del sistema la creación o modificación del Certificado de Liberación de un producto dado. Puede también de forma automatizada hacer señalamientos a un Acta de Liberación Parcial del Producto que sea rechazada.

Tabla 2.4 –. Descripción de los actores del sistema.

**2.5.2 Listado de casos de uso**

CU-1	Actualizar Estado de No Conformidades
<b>Actor</b>	Líder de proyecto
<b>Descripción</b>	Este caso de uso se inicia cuando el líder de proyecto selecciona la opción para ver el Registro de No Conformidades de uno de sus productos y lleva a cabo la selección de las No Conformidades que proceden, actualizando el Registro de No Conformidades mediante esta acción.
<b>Referencia</b>	R1.2, R1.2.1, R1.2.4, R1.3, R1.6
<b>Prioridad</b>	Crítico

**Tabla 2.5 –Caso de Uso Actualizar Estado de No Conformidades.**

CU-2	Mostrar No Conformidades Aceptadas
<b>Actor</b>	Miembro de proyecto
<b>Descripción</b>	Este caso de uso se inicia cuando el miembro de proyecto selecciona la opción para Mostrar las No Conformidades Aceptadas de un producto con la finalidad de corregir los errores, por tanto en este caso de uso se incluye que se puedan visualizar los detalles de las No Conformidades.
<b>Referencia</b>	R1.2.2, R1.3
<b>Prioridad</b>	Crítico

**Tabla 2.6 –Caso de Uso Mostrar No Conformidades Aceptadas.**

CU-3	Corregir No Conformidades de la documentación
<b>Actor</b>	Miembro de proyecto
<b>Descripción</b>	Este caso de uso se inicia opcionalmente durante la ejecución del caso de uso Mostrar No Conformidades Aceptadas si el miembro de proyecto selecciona ver los detalles de una No Conformidad de la documentación y la corrige. Por otra parte el líder de proyecto mientras selecciona las No Conformidades que proceden puede además corregirlas.
<b>Referencia</b>	R1.7, CU-1(extend), CU-2(extend),
<b>Prioridad</b>	Crítico

**Tabla 2.7 –.Caso de Uso Corregir No Conformidades de la documentación.**

CU-4	Obtener Estadísticas
<b>Actor</b>	Probador
<b>Descripción</b>	Este caso de uso se inicia cuando el probador selecciona la opción Estadísticas, tiene la posibilidad de verlas en forma gráfica o de tablas, según su decisión y también puede imprimirlas.
<b>Referencia</b>	R4
<b>Prioridad</b>	Secundario

**Tabla 2.8 –.Caso de Uso Obtener Estadísticas**

CU-5	Gestionar No Conformidades
<b>Actor</b>	Probador
<b>Descripción</b>	El caso de uso se inicia cuando el probador accede al sistema para registrar alguna No Conformidad, modificarla o eliminarla, así como para listarlas según diversos criterios.
<b>Referencia</b>	R1.1, R1.2, R1.3, R1.4, R 1.5
<b>Prioridad</b>	Crítico

**Tabla 2.9 –.Caso de Uso Gestionar No Conformidades.**

CU-6	Gestionar Acta de Liberación Parcial
<b>Actor</b>	Probador
<b>Descripción</b>	Este caso de uso se inicia cuando el probador va a solicitar a Calisoft la certificación de un producto a través del sistema, para ello crea o modifica un Acta de Liberación Parcial.
<b>Referencia</b>	R2.1, R2.2, R2.3, R2.4
<b>Prioridad</b>	Crítico

Tabla 2.10 –.Caso de Uso Gestionar Acta de Liberación Parcial.

CU-7	Gestionar Certificado de Liberación
<b>Actor</b>	Calisoft
<b>Descripción</b>	El caso de uso se inicia cuando Calisoft se dispone a dar respuesta a un Acta de Liberación Parcial aceptada, para ello crea un Certificado de Liberación, puede modificarlo en caso de ser necesario.
<b>Referencia</b>	R3.1, R3.2, R3.3, R3.4
<b>Prioridad</b>	Crítico

Tabla 2.11 –.Caso de Uso Gestionar Certificado de Liberación

CU-8	Subir documentación
<b>Actor</b>	Probador
<b>Descripción</b>	El caso de uso se inicia cuando el probador selecciona la opción Subir Documentación para subir al servidor algún documento.
<b>Referencia</b>	R5
<b>Prioridad</b>	Crítico

Tabla 2.12 –.Caso de Uso Subir documentación.

2.5.3 Diagrama de casos de uso.

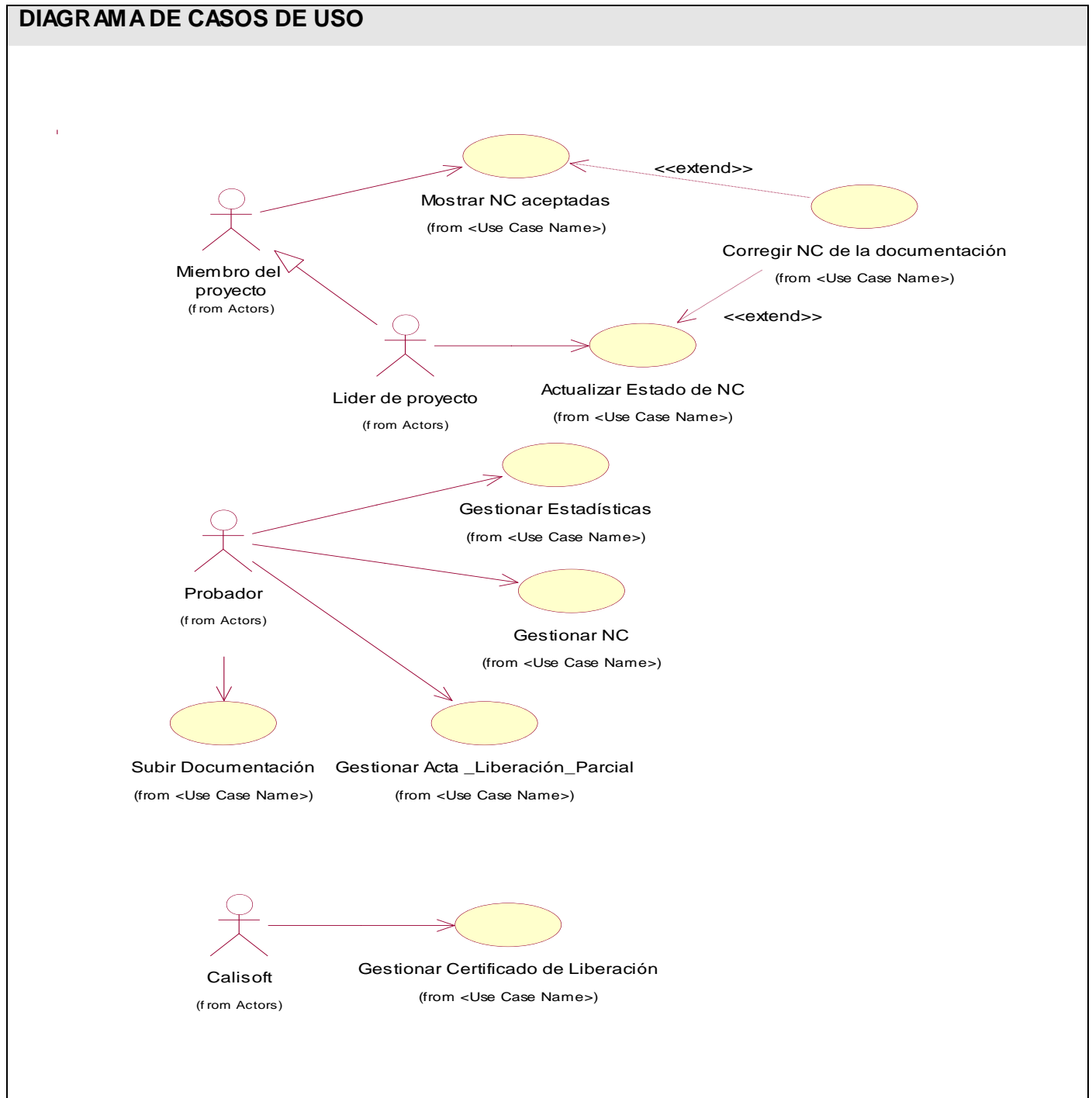


Fig. 2.2 -Diagrama de Casos de Uso.

### 2.5.4 Casos de uso expandidos.

Se presentará la descripción con formato expandido de uno de los casos de uso principales. Las otras descripciones se podrán encontrar en el Anexo 4.

<b>Caso de uso</b>	
CU-5	Gestionar No Conformidades.
<b>Propósito</b>	Gestionar el Registro de No Conformidades de un producto dado.
<b>Actores</b>	<b>Probador</b>
<b>Resumen:</b> El caso de uso se inicia cuando el probador accede al sistema para registrar alguna No Conformidad, modificarla o eliminarla, así como para listarlas según diversos criterios.	
<b>Referencias</b>	R1.1, R1.2, R1.3, R1.4, R1.5
<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El usuario selecciona el proyecto.	1.1- El sistema muestra en un dropdownlist la lista de productos correspondientes al proyecto seleccionado
2- El usuario selecciona el producto que desea ver sus No Conformidades.	2.1-El sistema muestra una lista de los ECS que correspondan al producto seleccionado.
3- El usuario selecciona el ECS que desea ver.	3.1- El sistema ejecuta una de las siguientes acciones. a) Si decide seleccionar un ECS correspondiente a la documentación ir a la sección “No Conformidades de la Documentación”. b) Si decide seleccionar un ECS correspondiente al software ir a la sección “No Conformidades del Software”.
<b>Sección “No Conformidades de la Documentación”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

<p>2- El usuario puede registrar o seleccionar una No Conformidad para modificarla o eliminarla.</p>	<p>1.1-El sistema muestra en una página el documento seleccionado, un formulario con los datos para registrar las no conformidades detectadas en el mismo, así como una lista con las no conformidades encontradas en revisiones anteriores en caso de tener algunas.</p> <p>2.1- El sistema ejecuta una de las siguientes acciones:</p> <p>a) Si decide Registrar una nueva No Conformidad ir a la sección “Registrar No Conformidad de la Documentación ”</p> <p>b) Si decide seleccionar una de las No Conformidades listadas ir a la sección “Seleccionar No Conformidades de la Documentación”.</p>
<p><b>Sección “Registrar No Conformidad de la Documentación”.</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
<p>1-El usuario selecciona en el documento el error y llena los datos correspondientes al mismo en el formulario y presiona el botón “Registrar”</p>	<p>1.1 – El sistema verifica que los datos estén correctos.</p> <p>1.2 –El sistema verifica la no existencia de la No Conformidad</p> <p>1.3 – El sistema almacena los datos de la No Conformidad</p> <p>1.4- El sistema guarda los cambios en el documento.</p>
<p><b>Flujo alternativo</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>1.1- El sistema muestra un mensaje de error porque los datos están incorrectos.</p> <p>1.2- El sistema muestra un mensaje de error porque esa No conformidad ya estaba registrada.</p>



<b>Sección “Seleccionar No Conformidades de la Documentación”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El usuario selecciona una de las No Conformidades listadas.	1.1- El sistema muestra los datos de la No Conformidad y la localiza (marcada) en el documento.
2- El usuario puede modificar o eliminar la No Conformidad	2.1 El sistema ejecuta las siguientes acciones: a) Si decide Modificar ir a la sección “Modificar No Conformidades de la Documentación”. b) Si decide Eliminar ir a la sección “Eliminar No Conformidades de la Documentación”.
<b>Sección “Modificar No Conformidades de la Documentación”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El usuario modifica los datos de la No Conformidad y presiona el botón “Modificar”.	1.1- El sistema verifica que los datos estén correctos. 1.2- El sistema actualiza los datos de la No Conformidad. 1.3- El sistema guarda los cambios del documento
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1- El sistema muestra un mensaje de error porque los datos están incorrectos.
<b>Sección “Eliminar No Conformidades de la Documentación”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El usuario presiona el botón “Eliminar”.	1.1- El sistema envía un mensaje de confirmación.  2.1- El sistema elimina la No Conformidad.
2- El usuario presiona “Aceptar”	
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
2- El usuario presiona “Cancelar”.	2.1- El sistema no elimina la No Conformidad.

<b>Sección “No Conformidades del Software”</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
2- El usuario puede registrar o seleccionar una No Conformidad para modificarla o eliminarla.	<p>1.1-El sistema muestra en una página un formulario con los datos para registrar las no conformidades detectadas al software, así como una lista con las no conformidades encontradas en revisiones anteriores en caso de tener algunas.</p> <p>2.1- El sistema ejecuta una de las siguientes acciones:</p> <p>a) Si decide Registrar una nueva No Conformidad ir a la sección “Registrar No Conformidad del software ”</p> <p>b) Si decide seleccionar una de las No Conformidades listadas ir a la sección “Seleccionar No Conformidad del software”.</p>
<b>Sección “Registrar No Conformidades del Software”</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1- El actor llena los datos de la No Conformidad	<p>1.1- El sistema verifica que los datos estén correctos.</p> <p>1.2- El sistema almacena los datos de la No Conformidad.</p>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
	1.1- El sistema envía un mensaje de error porque los datos están incorrectos.
<b>Sección “Seleccionar No Conformidades del Software”</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>

1- El usuario selecciona una de las No Conformidades que fueron listadas anteriormente.	1.1- El sistema muestra los datos de la No Conformidad seleccionada.
2- El usuario puede modificar o eliminar la No Conformidad	2.1- El sistema ejecuta una de las siguientes acciones: a) Si desea Modificar ir a la sección “Modificar No Conformidad del software”. b) Si decide Eliminar ir a la sección “Eliminar No Conformidades de la Documentación”.
<b>Sección “Modificar No Conformidades del Software”</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1- El usuario modifica los datos de la No Conformidad y presiona el botón “Modificar”	1.1- El sistema verifica que los datos estén correctos. 1.2- El sistema actualiza los datos de la No Conformidad.
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
	1.1-El sistema emite un mensaje de error porque los datos están incorrectos.
<b>Sección “Eliminar No Conformidades del Software”</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1- El usuario presiona el botón “Eliminar”.	1.1- El sistema emite un mensaje de confirmación.
2- El usuario selecciona “Aceptar”.	2.1- El sistema elimina la No Conformidad.
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
2-El usuario selecciona “Cancelar”	2.1- El sistema no elimina la No Conformidad.

Tabla 2.13 –.Descripción Expandida del CU Gestionar No Conformidades.

En este capítulo se han abordado aspectos relacionados con el proceso de control de la calidad, con el objetivo de profundizar en el conocimiento del proceso seguimiento de errores y fundamentar la necesidad de dar una solución informática a la situación existente.

Se realizó la modelación del negocio, a fin de lograr una mejor comprensión de los procesos del mismo. Logrando que los usuarios finales y desarrolladores tengan un entendimiento común. Se obtuvo el diagrama de casos de uso del negocio, así como la descripción de cada uno de ellos en formato expandido y los diagramas de actividades, donde se sombreadon las actividades que deben automatizarse en esta primera versión del sistema. Se identificaron y describieron los actores y trabajadores del negocio, Además, se obtuvo el modelo de objetos del negocio.

Se identificaron los requerimientos del sistema, obteniéndose así las funcionalidades que este debe tener, además se obtuvo un listado de requerimientos no funcionales a tener en cuenta para el desarrollo de la aplicación.

Se definieron los casos de uso del sistema, mostrándose el diagrama de casos de uso. Además, se describió cada uno de estos, así como los actores del sistema que se van a relacionar con los mismos, dando de esta forma una vista global de la concepción y funcionamiento del sistema.

## **Capítulo 3: Análisis y Diseño del Sistema.**

Como resultado del flujo de trabajo de requisitos se obtiene una vista externa del sistema, que en el lenguaje del cliente, describe lo que se espera de él a través del Diagrama de casos de uso. A partir de aquí se debe profundizar en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos.

Este proceso se desarrolla fundamentalmente dentro de la fase de elaboración y se corresponde principalmente con el flujo de trabajo de análisis y diseño. En el presente capítulo se abordaran los aspectos relacionados con este flujo de trabajo, incluyendo, los diagramas de clases tanto del análisis como del diseño, diagramas de interacción, descripción de las clases de diseño, así como al diseño de la Base de Datos.

El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos.

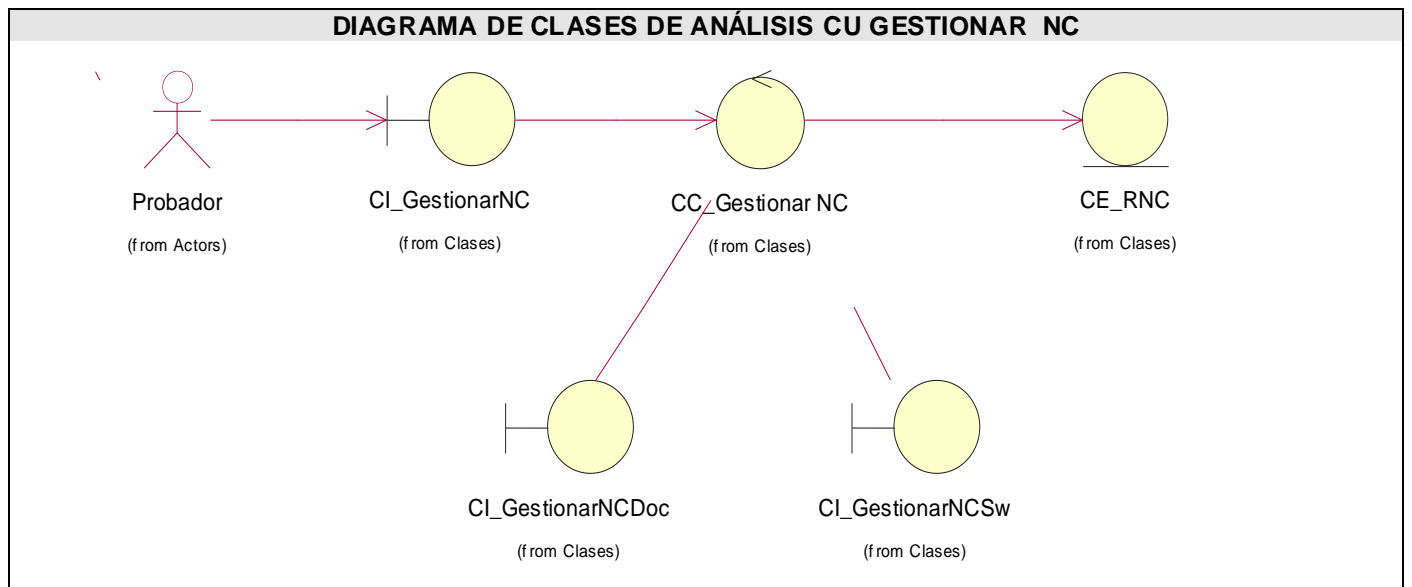
### **3.1 Análisis.**

#### **3.1.1 Diagramas de clases del análisis.**

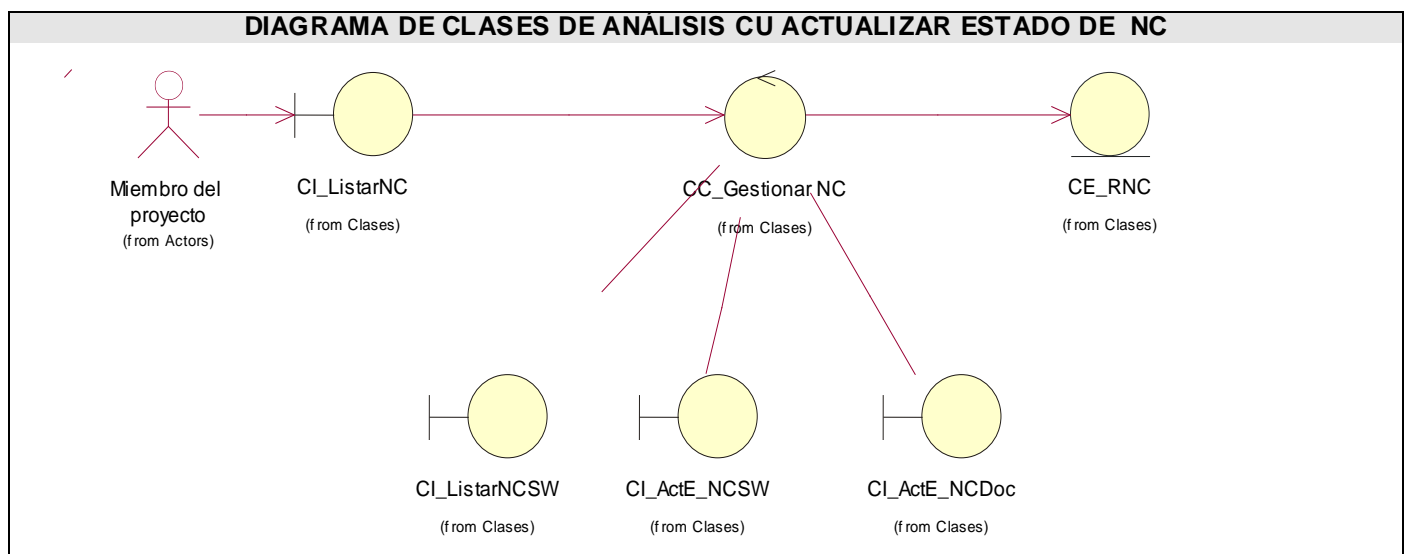
Los diagramas de clases del análisis representan la relación entre las clases, las cuales se centran en los requisitos funcionales, tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar a las clases en: interfaz, controladora y entidad.

Las clases interfaz Modelan la interacción entre el sistema y sus actores. Las controladoras coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso. Las clases Entidad modelan información que posee larga vida y que es a menudo persistente.

Se presentarán los diagramas de clases del análisis de los casos de uso Gestionar No Conformidades, Actualizar Estado de las No Conformidades y Mostrar No Conformidades aceptadas. El resto de los Diagramas de clases se podrán encontrar en el Anexo 5.



**Fig. 3.1 –. DCA\_ CU Gestionar No Conformidades.**



**Fig. 3.2 –. DCA\_ CU Actualizar Estado de No Conformidades.**

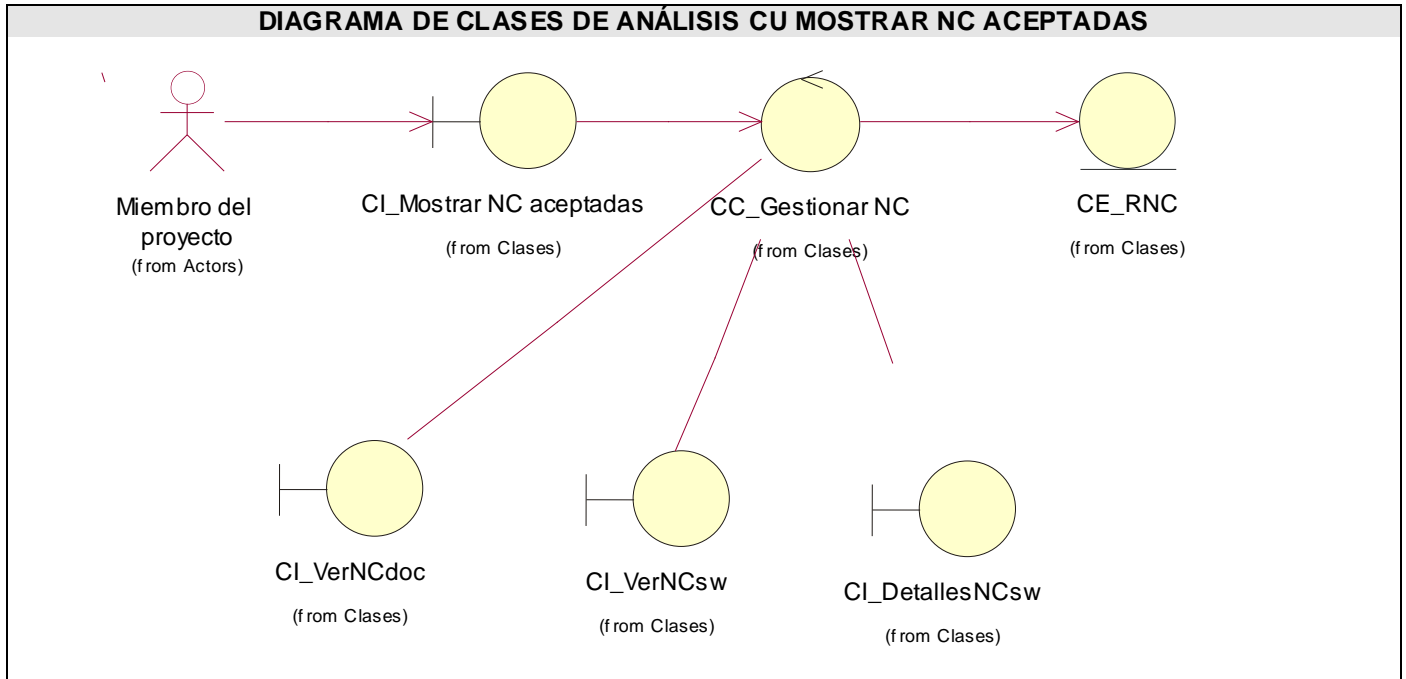


Fig. 3.3 –. DCA\_CU Mostrar No Conformidades Aceptadas.

### 3.1.2 Diagramas de interacción (Colaboración).

Se presentarán los diagramas de interacción de los casos de uso anteriormente mencionados, divididos por escenarios, lo cual permitirá una mejor comprensión de los mismos. El resto de los diagramas de interacción se podrán encontrar en el Anexo 6.

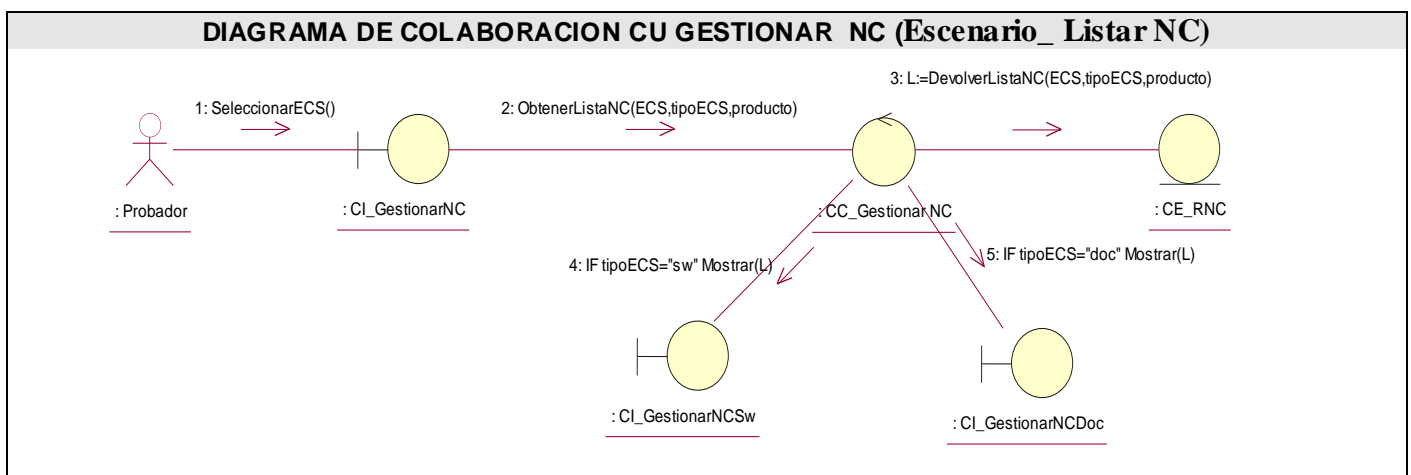
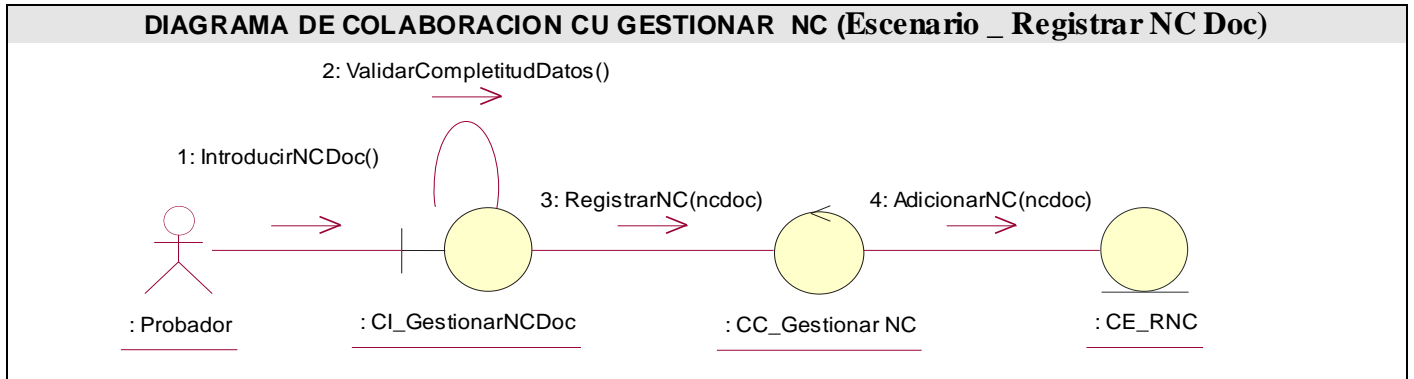
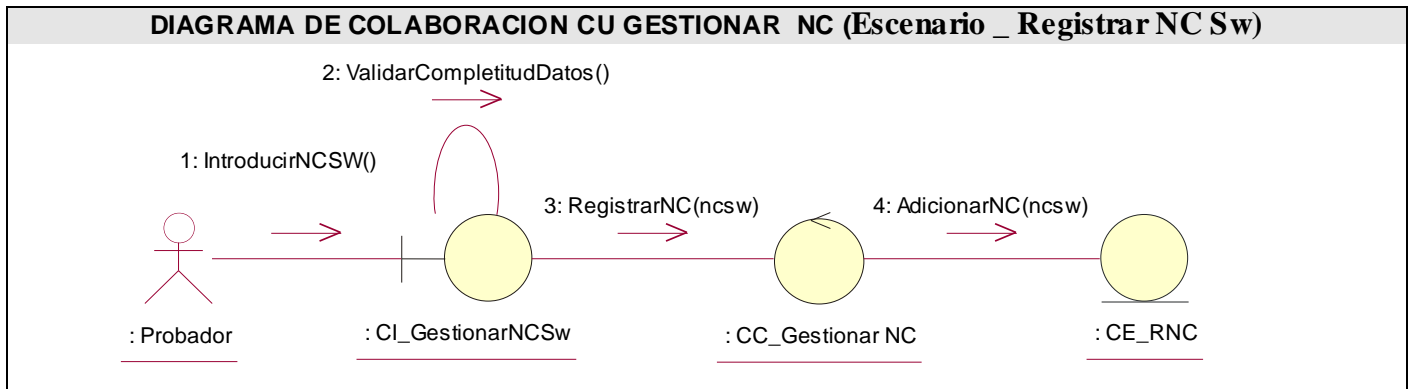


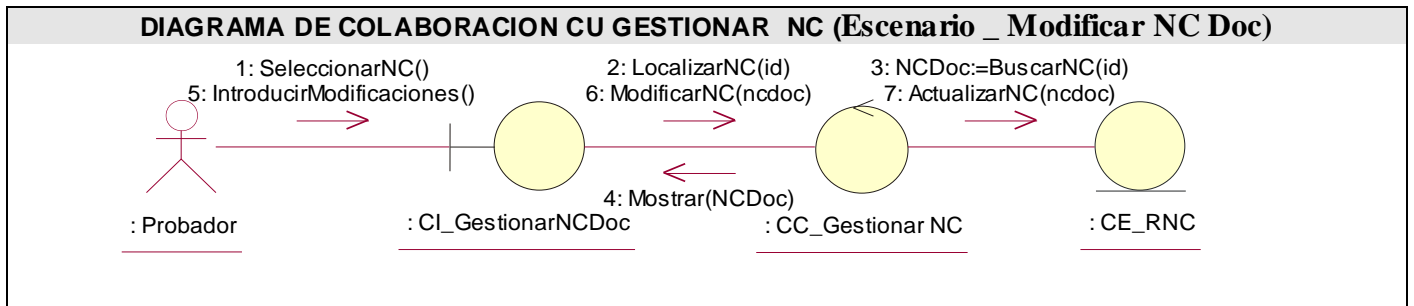
Fig. 3.4 –. DC\_CU Gestionar No Conformidades (Escenario\_Listar NC).



**Fig. 3.5 –. DC\_ CU Gestionar No Conformidades (Escenario\_ Registrar NC Doc).**

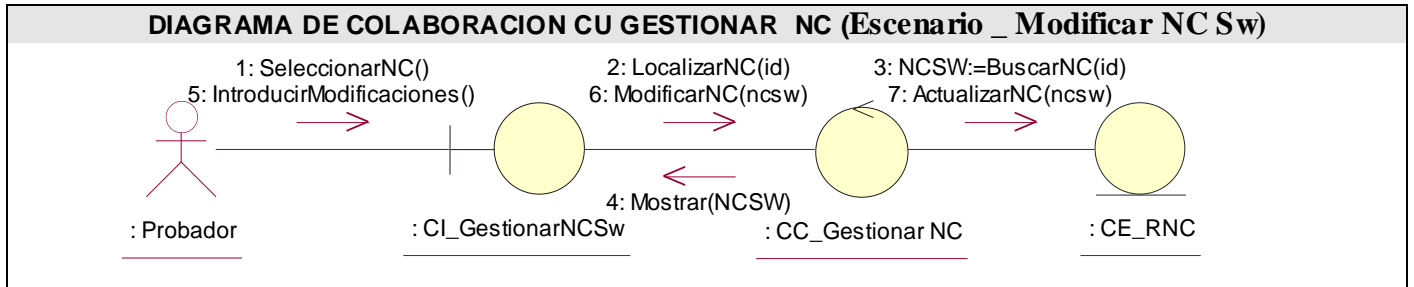


**Fig. 3.6 –. DC\_ CU Gestionar No Conformidades (Escenario\_ Registrar NC Sw).**

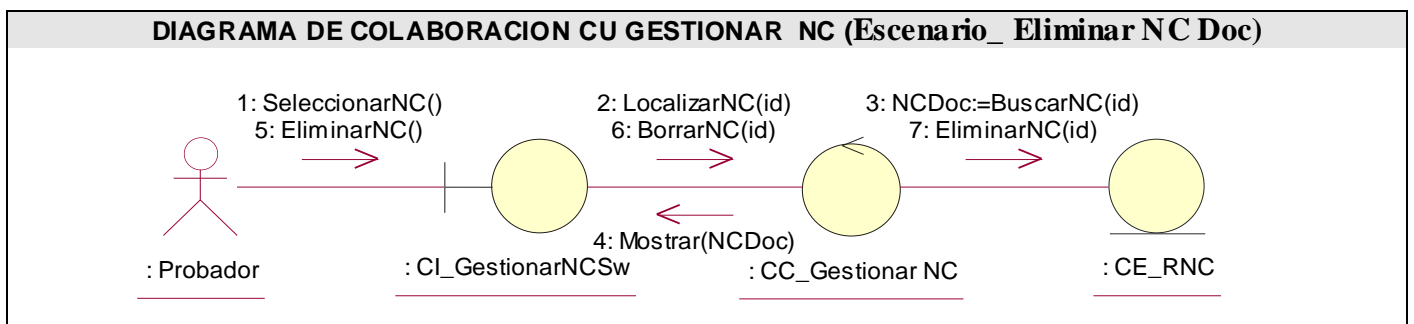


**Fig. 3.7–. DC\_ CU Gestionar No Conformidades (Escenario\_ Modificar NC Doc).**

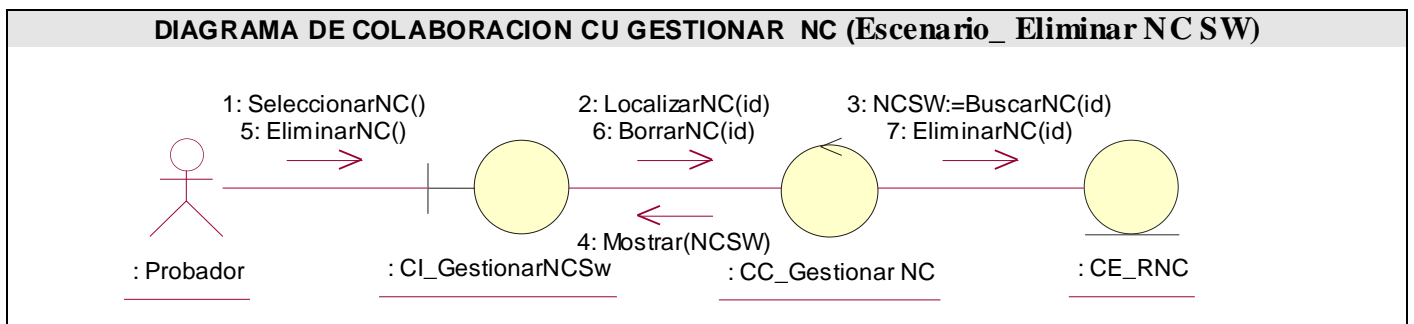




**Fig. 3.8 –. DC\_ CU Gestionar No Conformidades (Escenario\_ Modificar NC Sw).**



**Fig. 3.9 –. DC\_ CU Gestionar No Conformidades (Escenario\_ Eliminar NC Doc).**



**Fig. 3.10 –. DC\_ CU Gestionar No Conformidades (Escenario\_ Eliminar NC Doc).**

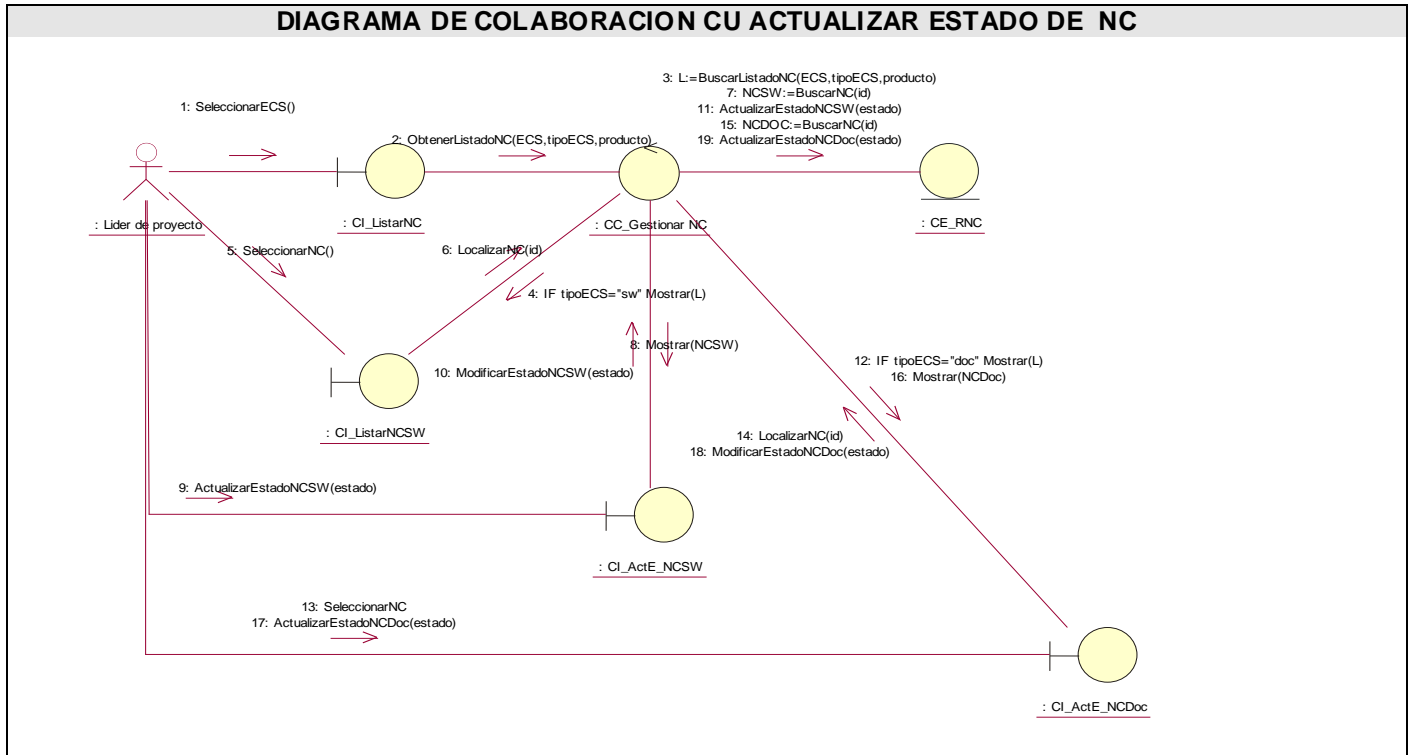


Fig. 3.11 -. DC\_ CU Actualizar Estado de No Conformidades

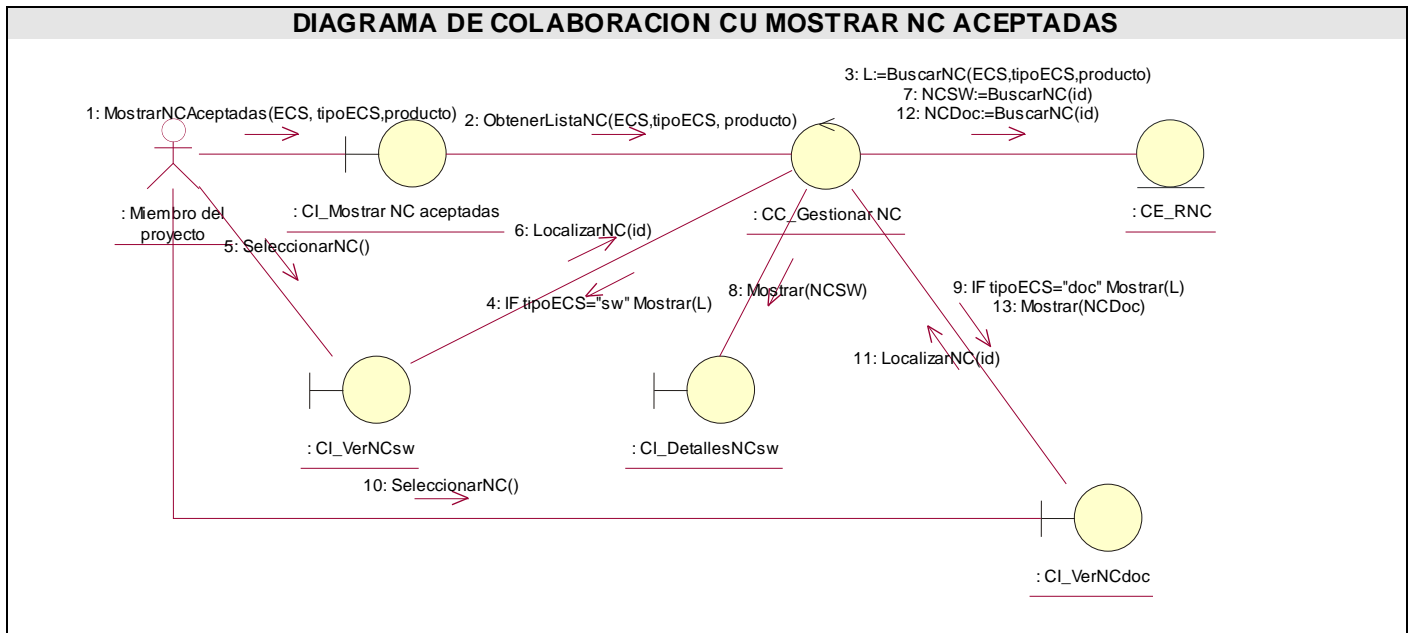


Fig. 3.12 -. DC\_ CU Mostrar No Conformidades Aceptada.

## **3.2 Diseño.**

### **3.2.1 Fundamentación del uso de patrones.**

Para llevar a cabo un buen diseño se han definido una serie de patrones.

Los patrones de diseño de software constituyen un conjunto de principios generales y expresiones que ayudan a desarrollar software. (ERAZO 2006)

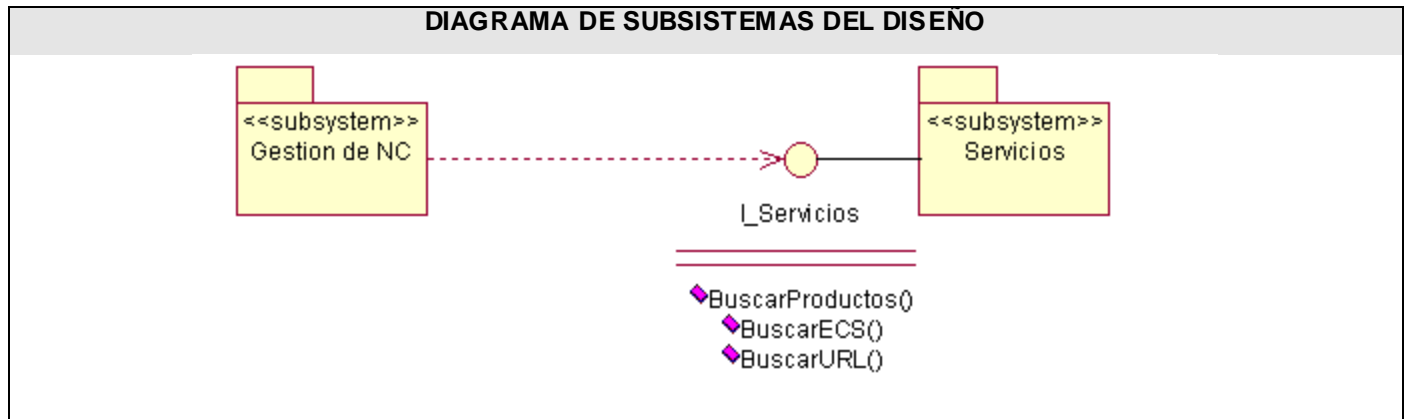
Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. (SAAVEDRA 2007)

Dentro de este grupo se identifican 5 patrones fundamentales experto, creador, alta cohesión, bajo acoplamiento y el controlador. En los diagramas de clases elaborados se aplican dichos patrones, se utilizan a fin de distribuir responsabilidades en las mismas, y establecer sus relaciones, tratando de que no estén muy sobrecargadas de funcionalidades ni exista mucha dependencia entre ellas.

### **3.2.2 Definición de elementos de diseño.**

Se ha definido un subsistema de diseño que agrupa aquellos casos de uso arquitectónicamente significativos para la Gestión de No Conformidades (NC) dentro del cual se encuentran los diagramas de clases del diseño con sus respectivos diagramas de interacción, en este caso, de secuencia.

Dada la necesidad que surge, al estar estrechamente vinculados, se utiliza un subsistema para la gestión de servicios que, a pesar de no ser desarrollado en el presente trabajo, sí se debe mostrar de alguna manera la relación que existe entre ambos. Dicho subsistema posibilita, a través de una interface Servicios, la búsqueda de productos de cada proyecto, así como la búsqueda de los Elementos de Configuración por producto y la URL de los documentos almacenados.



**Fig. 3. 13–. Diagrama de subsistemas del diseño.**

El sistema se encuentra dividido en tres capas bien definidas, que son la presentación, la lógica del negocio y la capa de acceso a datos.

**Presentación:** Incluye las clases interfaces, a través de las cuales se establece la comunicación con los usuarios, y para dar cumplimiento a los pedidos de estos se invocarán los métodos de la lógica del negocio.

**Lógica del negocio:** Incluye las clases controladoras, así como las entidades del negocio, que representan objetos que van a ser manejados o consumidos por toda la aplicación.

**Acceso a datos:** Contiene clases que interactúan con la base de datos y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

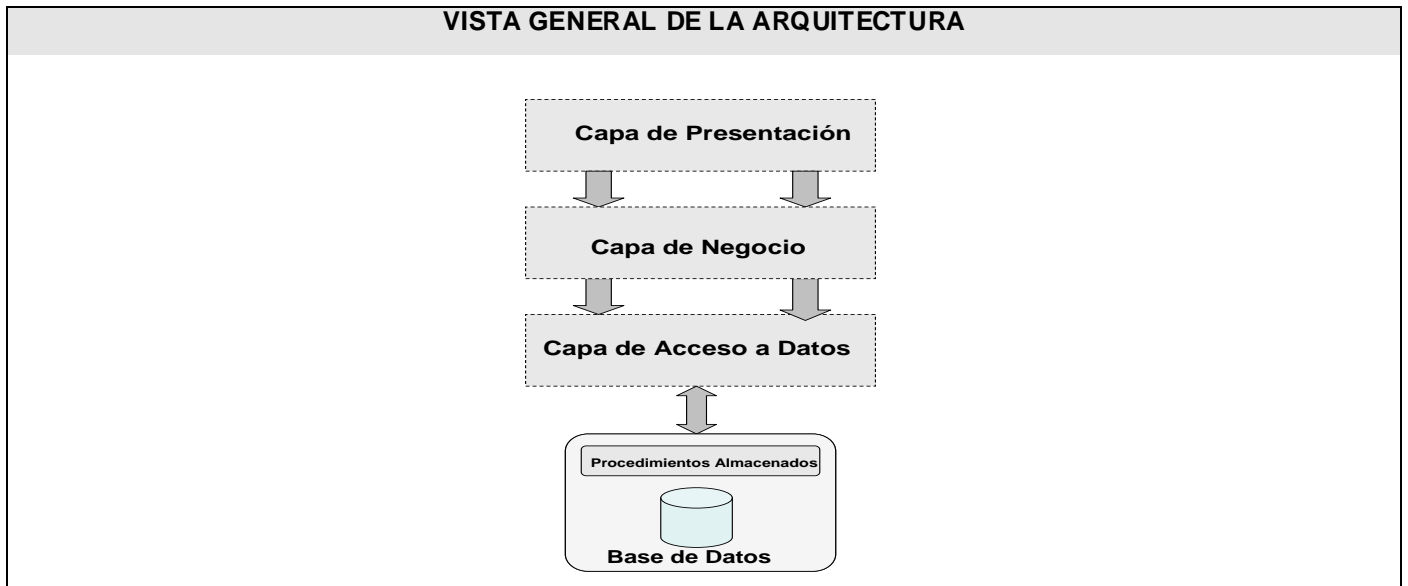


Fig. 3. 14-. Vista general de la arquitectura.

### 3.2.3 Diagramas de clases del diseño.

En el diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema.

Se presentarán los diagramas de clases del diseño de los casos de uso Gestionar No Conformidades, Actualizar Estado de las No Conformidades y Mostrar No Conformidades aceptadas. El resto de los diagramas de clases del diseño se podrán encontrar en el Anexo 7.

DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR NC

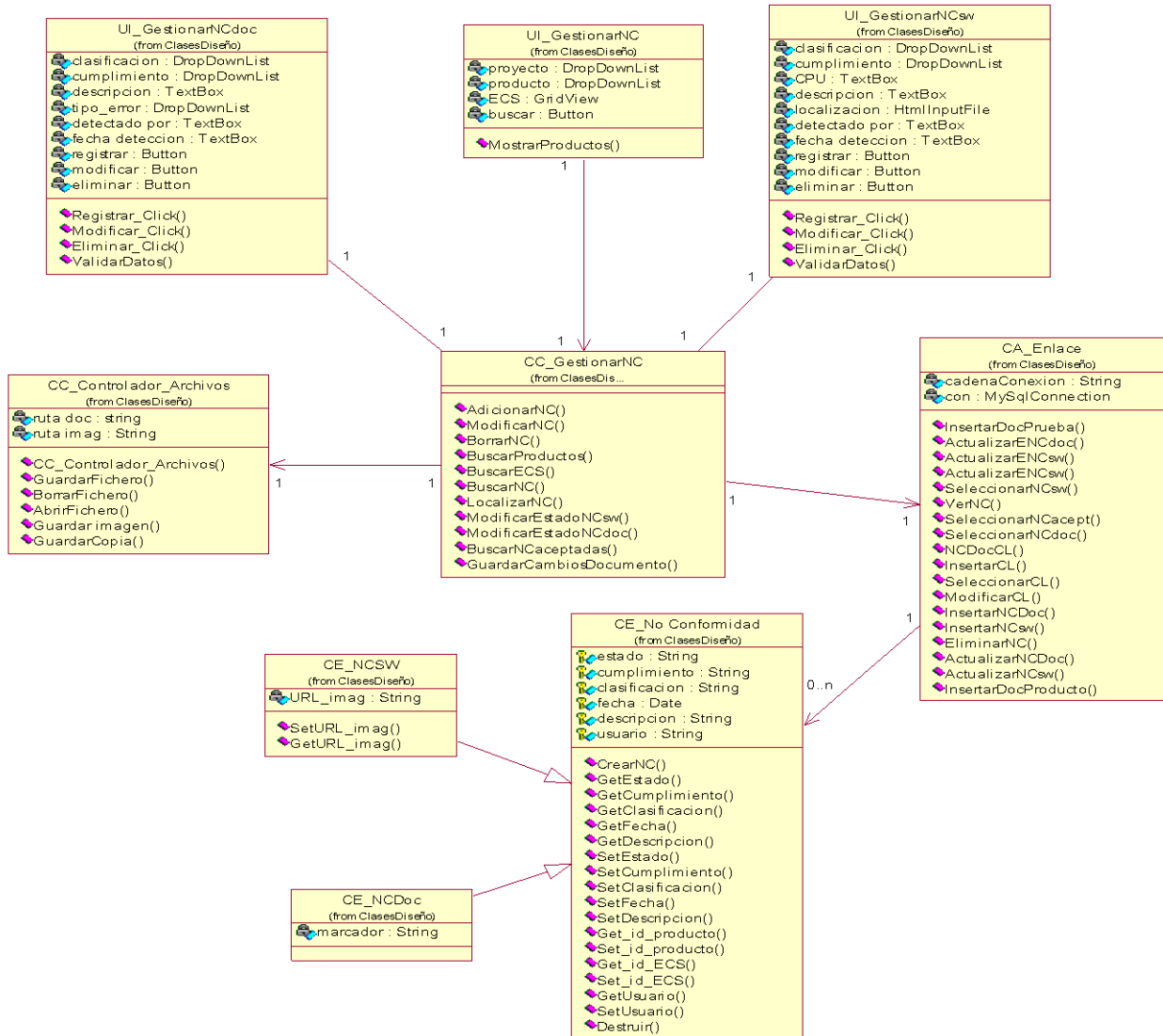


Fig. 3.15-. DCD\_ CU Gestionar No Conformidades.

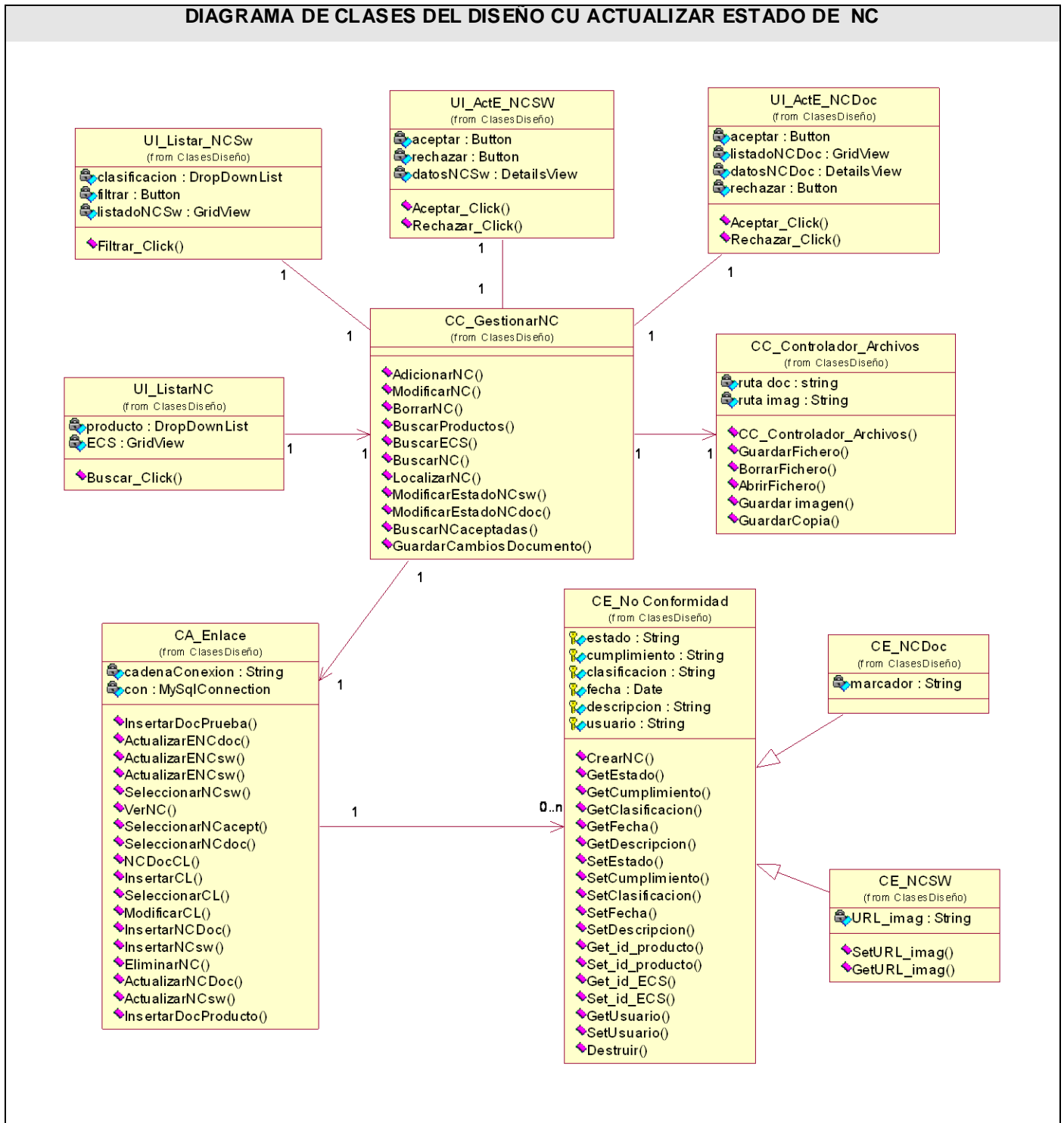


Fig. 3.16 -. DCD\_ CU Actualizar Estado de No Conformidades.

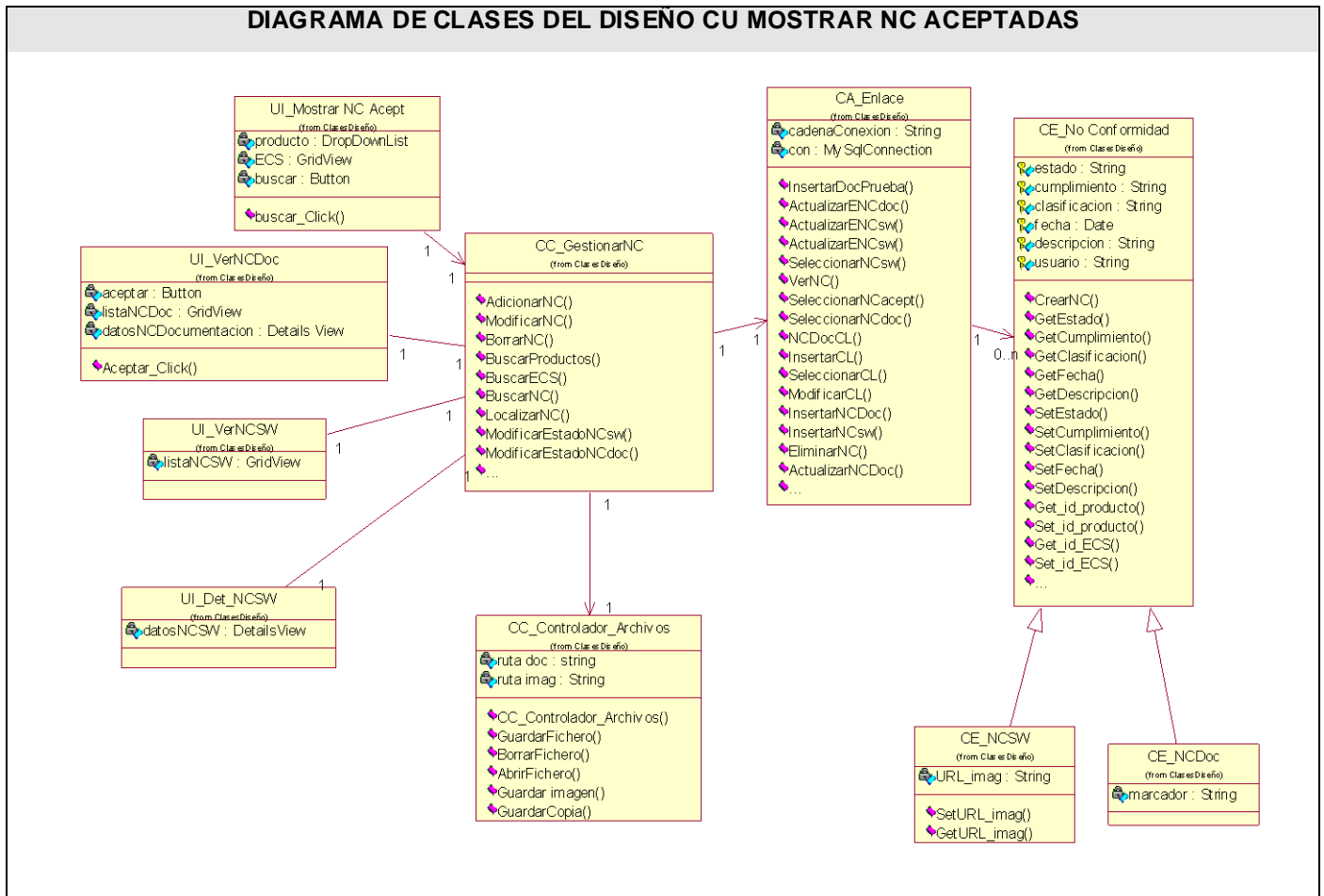


Fig. 3.17 -. DCD\_ CU Mostrar No Conformidades Aceptadas.

### 3.2.4 Diagramas de interacción. (Secuencia)

Se presentarán los diagramas de interacción del caso de uso Gestionar No Conformidades, dividido por escenarios, para lograr una mejor comprensión del mismo. El resto de los diagramas de interacción se podrán encontrar en el Anexo 8.



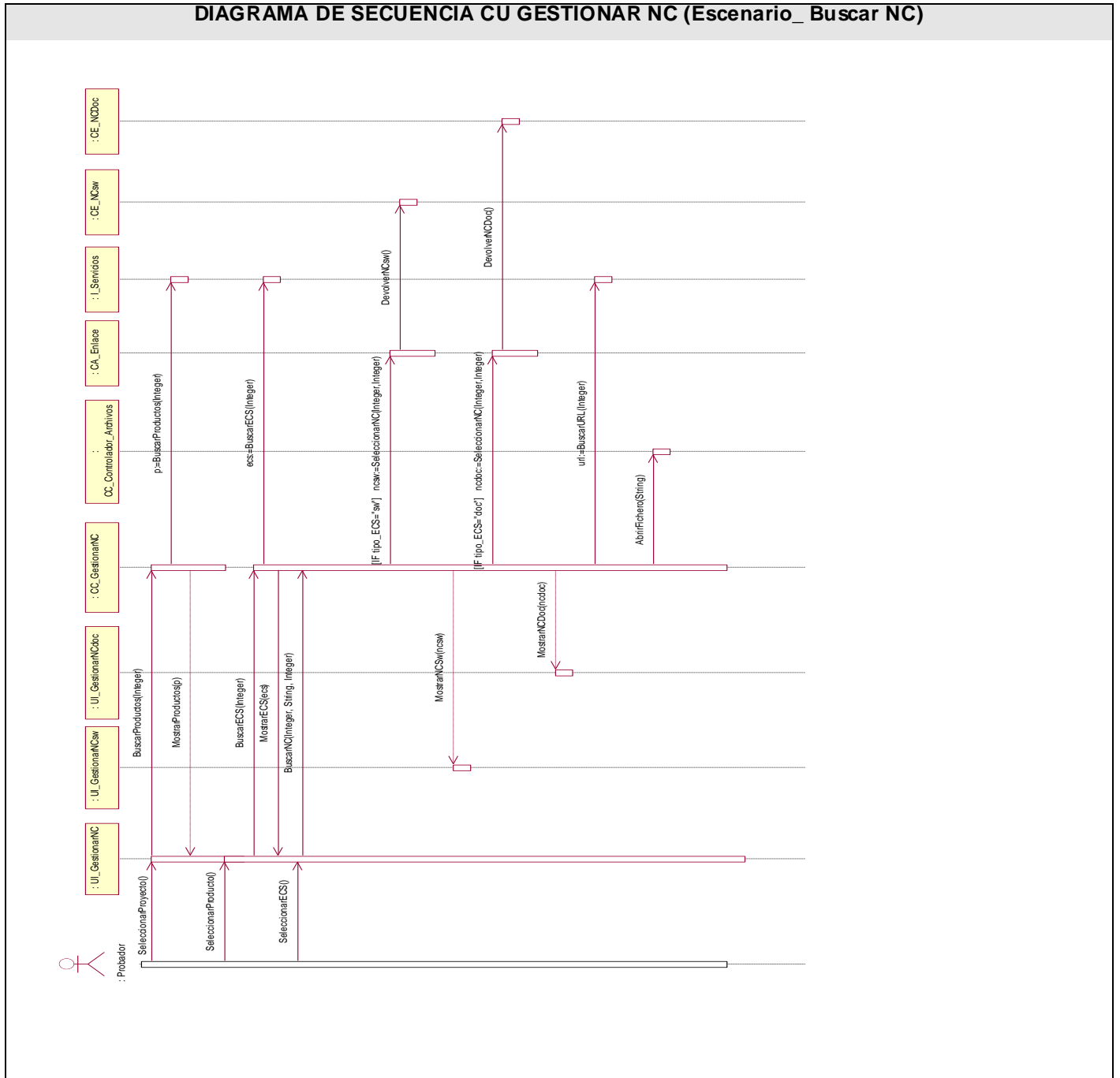


Fig. 3.18 -. DS\_ CU Gestionar No Conformidades (Escenario\_ Buscar No Conformidades).

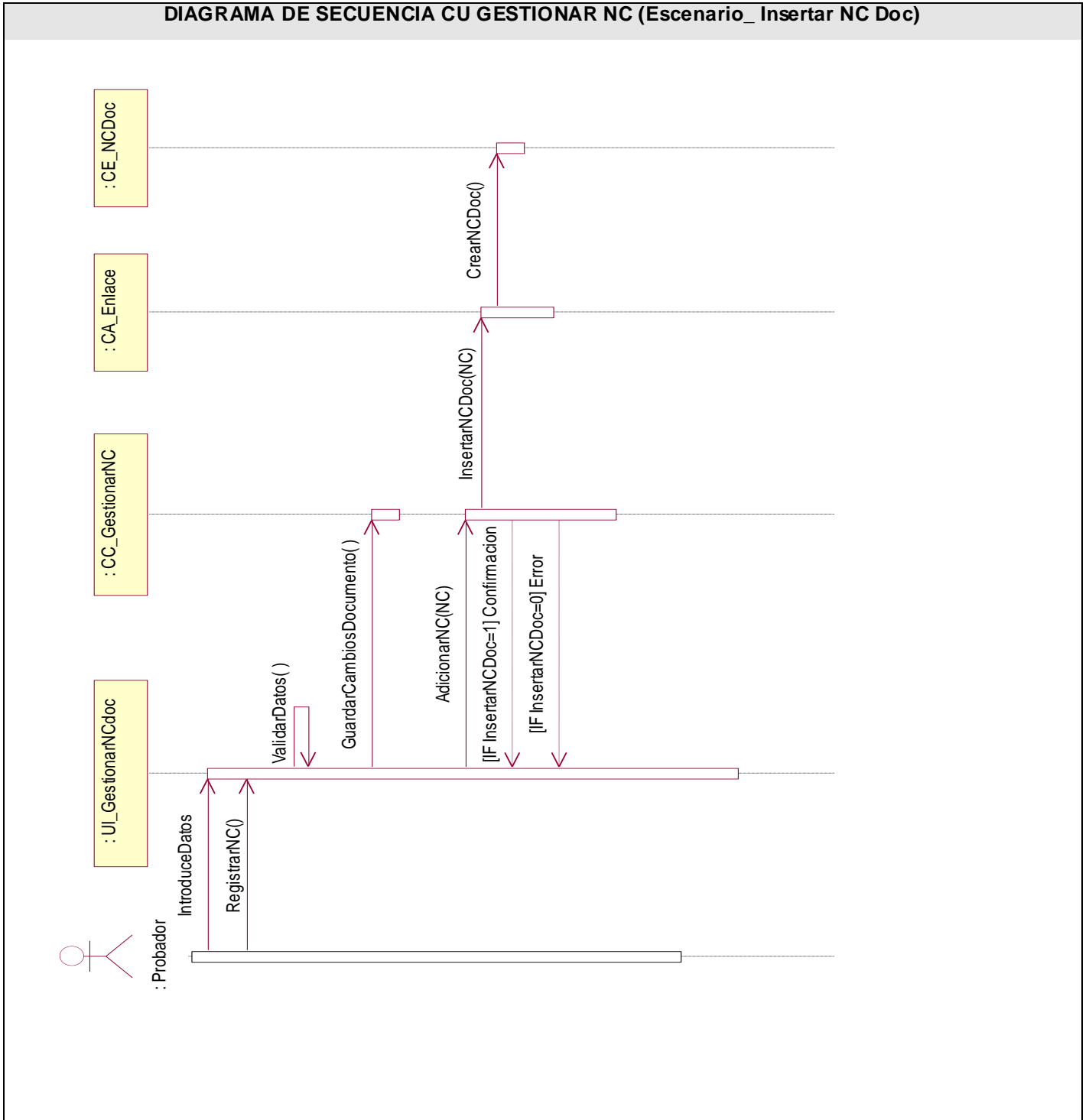


Fig. 3.19 –DS\_ CU Gestionar No Conformidades (Escenario\_ Insertar No Conformidades de Doc).

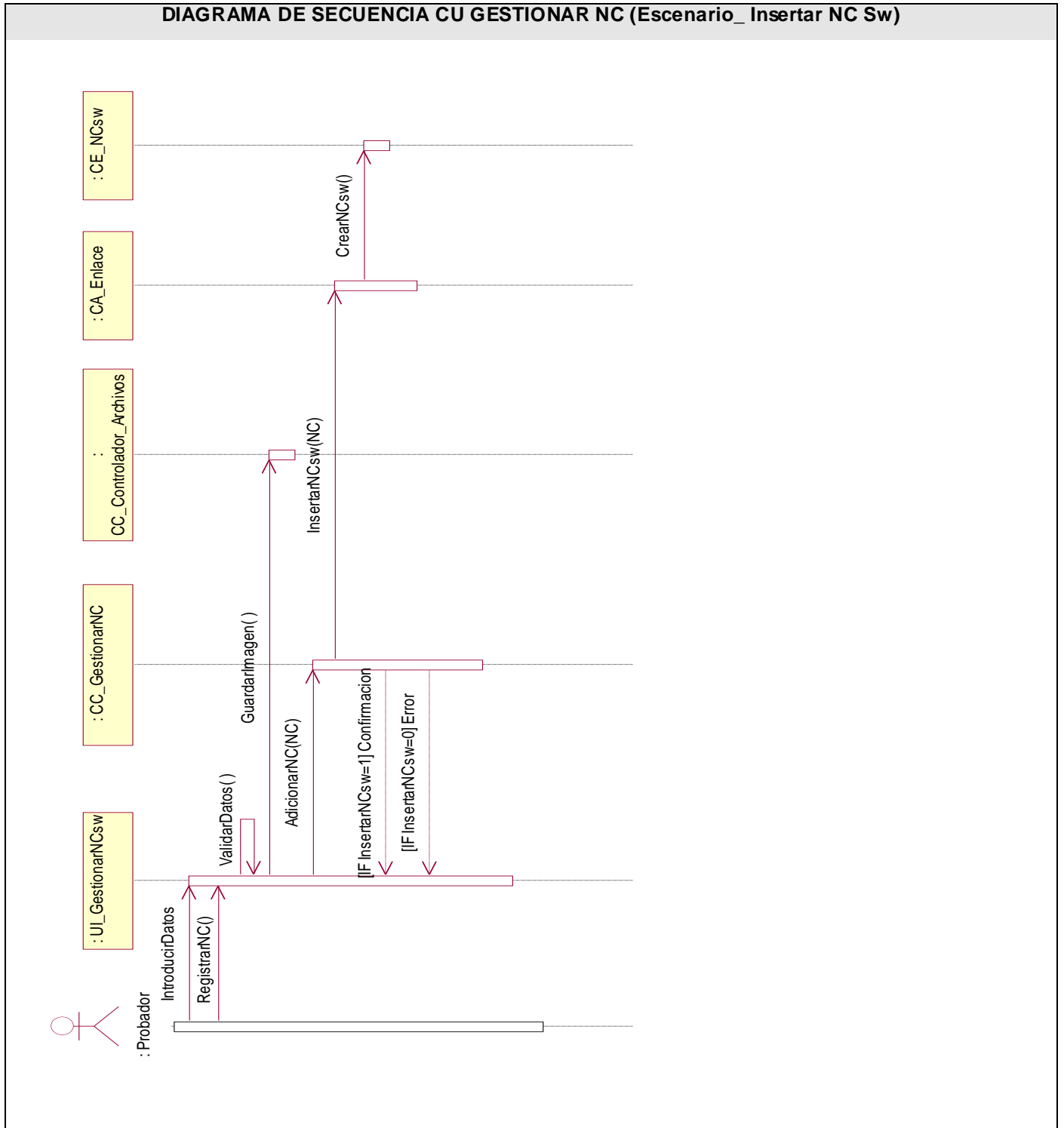


Fig. 3.20 –DS\_ CU Gestionar No Conformidades (Escenario\_ Insertar No Conformidades de Sw).

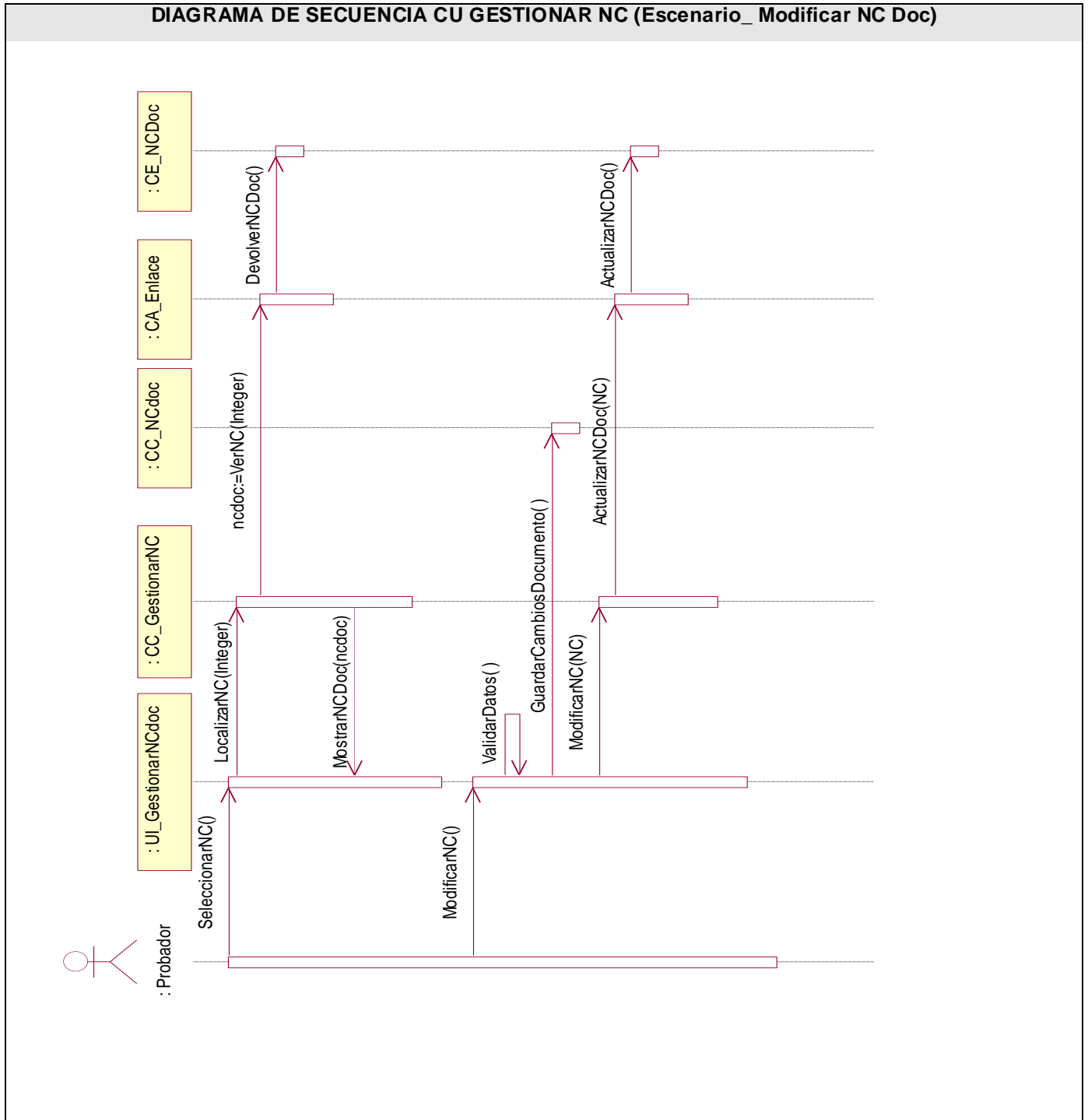


Fig. 3.21 –DS\_ CU Gestionar No Conformidades (Escenario\_ Modificar No Conformidades de Doc).

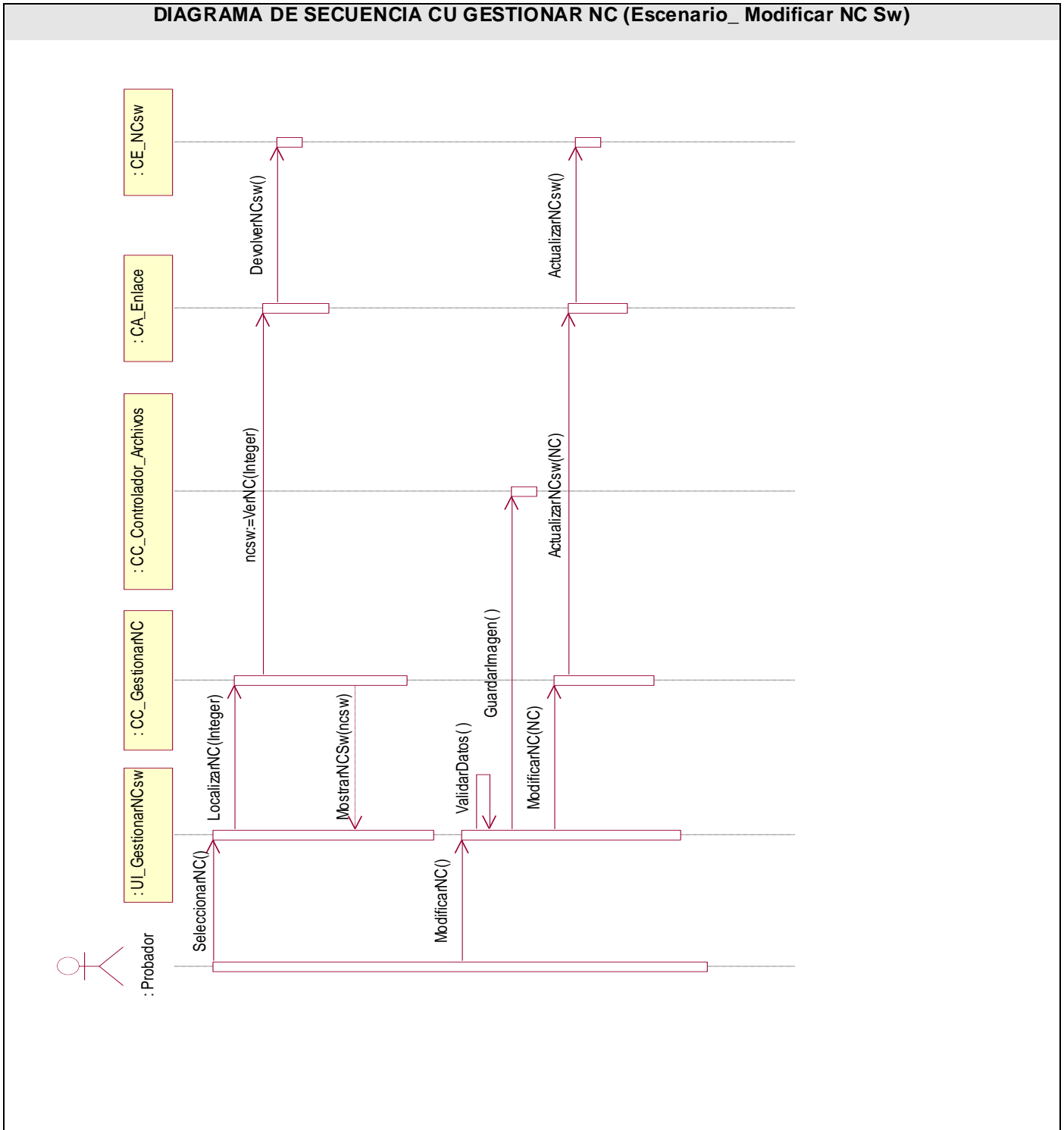


Fig. 3.22 –.DS\_ CU Gestionar No Conformidades (Escenario\_ Modificar No Conformidades de Sw).

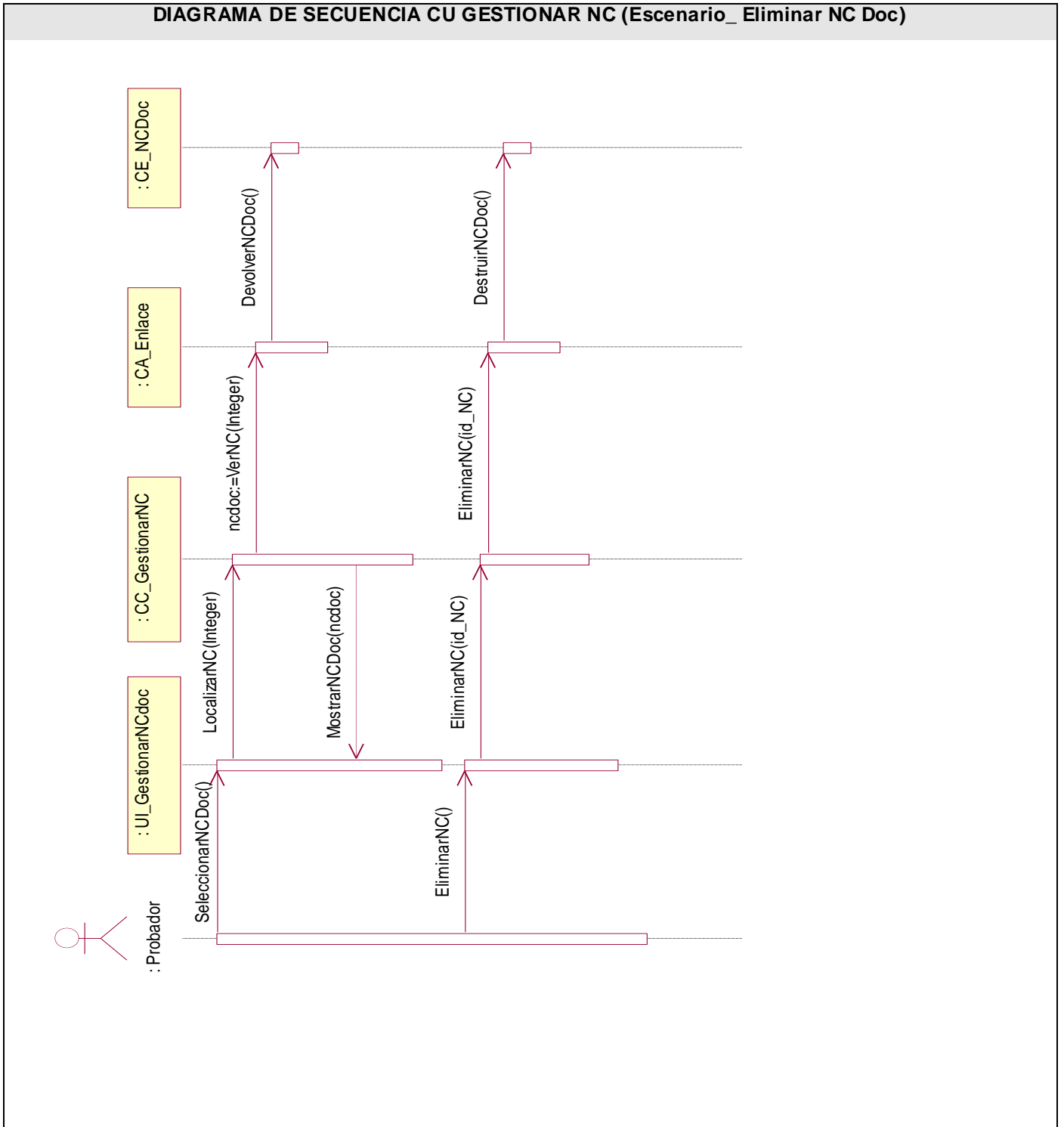


Fig. 3.23 –.DS\_ CU Gestionar No Conformidades (Escenario\_ Eliminar No Conformidades de Doc).

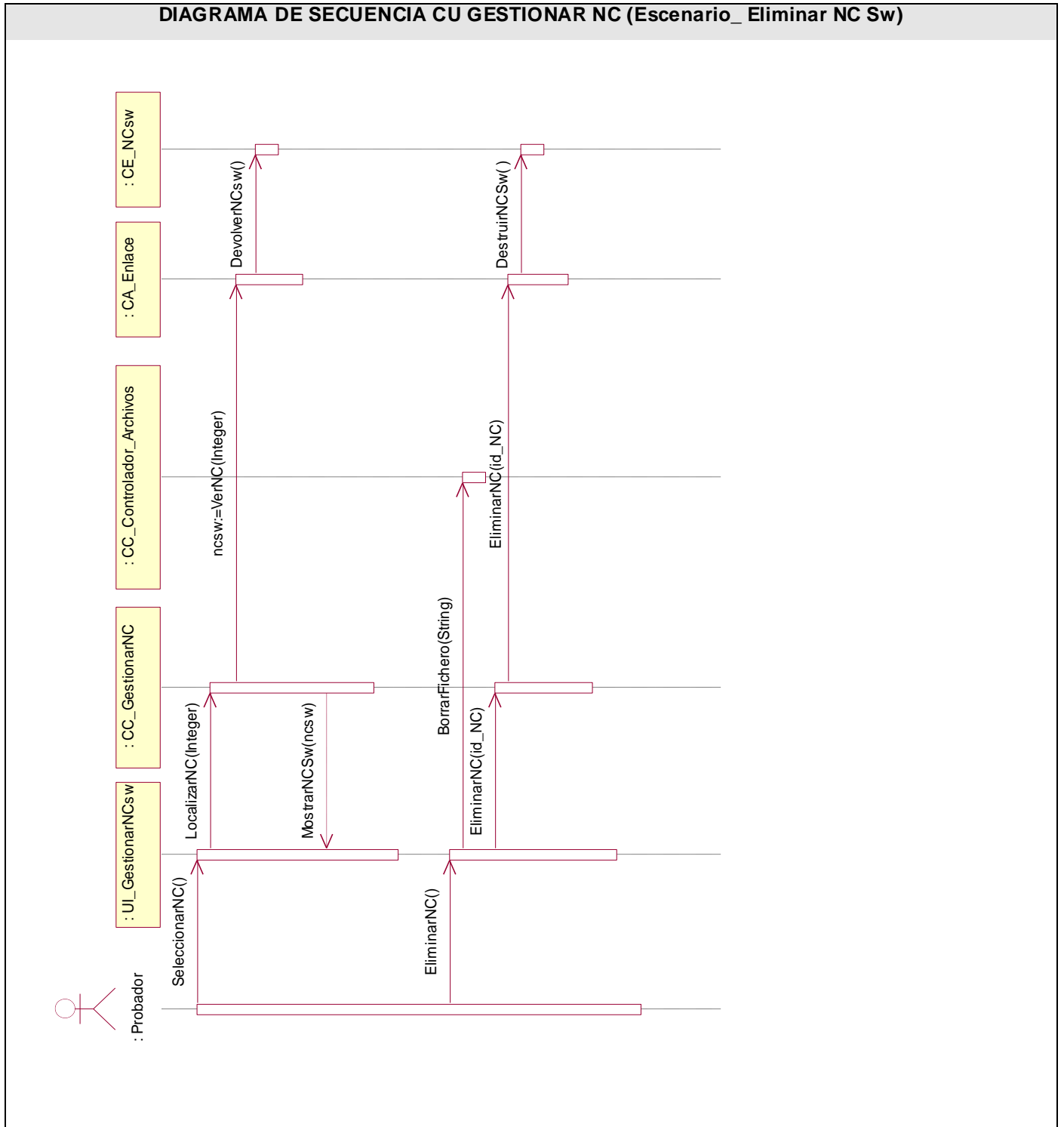


Fig. 3.24 –DS\_ CU Gestionar No Conformidades (Escenario\_ Eliminar No Conformidades de Sw).

### 3.2.5 Descripción de las clases.

Se presentarán las descripciones de las clases de los casos de uso de los cuales se expuso el diagrama. El resto de las descripciones de clases se podrán encontrar en el Anexo 9.

<b>Nombre:</b> UI_ListarNC	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
producto	DropDownList
ECS	GridView
Buscar	Button
<b>Para cada responsabilidad:</b>	
Nombre:	void Buscar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Buscar.

Tabla 3.1 –.Descripción CD: UI\_ListarNC

<b>Nombre:</b> UI_Listar_NCSw	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
clasificacion	DropDownList
Filtrar	Button
listadoNCSw	GridView
<b>Para cada responsabilidad:</b>	
Nombre:	void Filtrar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Filtrar.

Tabla 3.2 –.Descripción CD: UI\_Listar\_NCSw

<b>Nombre:</b> UI_ActE_NCSW	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
Aceptar	Button
Rechazar	Button



datosNCSw		DetailsView
<b>Para cada responsabilidad:</b>		
Nombre:	void Aceptar_Click(Object sender, EventArgs e)	
Descripción:	Este método se ejecuta cuando se presiona el botón Aceptar.	
Nombre:	void Rechazar_Click(Object sender, EventArgs e)	
Descripción:	Este método se ejecuta cuando se presiona el botón Rechazar.	

**Tabla 3.3 –.Descripción CD: UI\_ActE\_NCSW**

<b>Nombre:</b> UI_ActE_NCDoc		
<b>Tipo de clase:</b> interfaz		
<b>Atributo</b>	<b>Tipo</b>	
Aceptar	Button	
listadoNCDoc	GridView	
datosNCDoc	DetailsView	
<b>Para cada responsabilidad:</b>		
Nombre:	void Aceptar_Click(Object sender, EventArgs e)	
Descripción:	Este método se ejecuta cuando se presiona el botón Aceptar.	

**Tabla 3.4 –.Descripción CD: UI\_ActE\_NCDoc.**

<b>Nombre:</b> UI_GestionarNC		
<b>Tipo de clase:</b> interfaz		
<b>Atributo</b>	<b>Tipo</b>	
proyecto	DropDownList	
producto	DropDownList	
ecs	GridView	
Buscar	Button	
<b>Para cada responsabilidad:</b>		
Nombre:	void Buscar_Click(Object sender, EventArgs e)	
Descripción:	Este método se ejecuta cuando se presiona el botón Buscar.	

**Tabla 3.5 –.Descripción CD: UI\_GestionarNC.**

<b>Nombre:</b> UI_GestionarNCsw	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
clasificacion	DropDownList
cumplimiento	DropDownList
cpu	TextBox
descripcion	TextBox
localizacion	HtmlInputFile
detectado por	TextBox
fecha deteccion	TextBox
registrar	Button
modificar	Button
eliminar	Button
<b>Para cada responsabilidad:</b>	
Nombre:	void Registrar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Registrar.
Nombre:	void Modificar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Modificar.
Nombre:	void Eliminar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Eliminar.
Nombre:	void ValidarDatos()
Descripción:	Este método se encarga de verificar que el usuario haya llenado todos los campos obligatorios y que los datos entrados sean válidos.

Tabla 3.6 –.Descripción CD: UI\_GestionarNCsw.

<b>Nombre:</b> UI_GestionarNCdoc	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
clasificacion	DropDownList
cumplimiento	DropDownList

descripcion	TextBox
tipo_error	DropDownList
detectado por	TextBox
fecha deteccion	TextBox
registrar	Button
modificar	Button
eliminar	Button
<b>Para cada responsabilidad:</b>	
Nombre:	void Registrar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Registrar.
Nombre:	void Modificar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Modificar.
Nombre:	void Eliminar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Eliminar.
Nombre:	void ValidarDatos()
Descripción:	Este método se encarga de verificar que el usuario haya llenado todos los campos obligatorios y que los datos entrados sean válidos.

**Tabla 3.7 –.Descripción CD: UI\_GestionarNCdoc**

<b>Nombre:</b> UI_ Mostrar NC Acept	
<b>Tipo de clase:</b> interfaz	
<b>Atributo</b>	<b>Tipo</b>
producto	DropDownList
ECS	GridView
Buscar	Button
<b>Para cada responsabilidad:</b>	
Nombre:	void Buscar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Buscar

**Tabla 3.8 –.Descripción CD: UI\_ Mostrar NC Acept**

<b>Nombre:</b> UI_ VerNCDoc	
<b>Tipo de clase:</b> interfaz	
<b>Atributo</b>	<b>Tipo</b>
Aceptar	Button
listaNCDoc	GridView
datosNCDoc	DetailsView
<b>Para cada responsabilidad:</b>	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Aceptar.

Tabla 3.9 –.Descripción CD: UI\_ VerNCDoc

<b>Nombre:</b> UI_ VerNCSW	
<b>Tipo de clase:</b> interfaz	
<b>Atributo</b>	<b>Tipo</b>
listaNCSW	GridView
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

Tabla 3.10 –.Descripción CD: UI\_ VerNCSW

<b>Nombre:</b> UI_ Det_ NCSW	
<b>Tipo de clase:</b> interfaz	
<b>Atributo</b>	<b>Tipo</b>
datosNCSW	DetailsView
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

Tabla 3.11 –.Descripción CD: UI\_ Det\_ NCSW

<b>Nombre:</b> CC_GestionarNC	
<b>Tipo de clase:</b> Controladora	

Atributo	Tipo
<b>Para cada responsabilidad:</b>	
Nombre:	void AdicionarNC(NC nc)
Descripción:	Este método es el encargado de adicionar una nueva No Conformidad al listado de No Conformidades ya existente y validar que se haya adicionado correctamente.
Nombre:	void ModoficarNC(NC nc)
Descripción:	Este método modifica una No Conformidad determinada y validar que se haya modificado correctamente.
Nombre:	void BorrarNC(NC nc)
Descripción:	Este método es el encargado de eliminar una No Conformidad determinada del Listado de No Conformidades y validar que se haya eliminado correctamente.
Nombre:	ArrayList BuscarProductos(int id_proyecto)
Descripción:	Este método gestiona la búsqueda de los productos pertenecientes a un proyecto determinado.
Nombre:	ArrayList BuscarECS(int id_producto)
Descripción:	Este método gestiona la búsqueda de los elementos de configuración pertenecientes a un producto determinado.
Nombre:	ArrayList BuscarNC(int id_ECS, string tipo_ECS, int id_producto)
Descripción:	Este método busca las todas las No Conformidades que le fueron detectadas a un elemento de configuración determinado de un producto dado.
Nombre:	NC LocalizarNC(int id_NC)
Descripción:	Este método busca una No Conformidad determinada.
Nombre:	void ModificarEstadoNCsw (int id_NC, string estado)
Descripción:	Este método actualiza el estado de una No Conformidad de software determinada.
Nombre:	void ModificarEstadoNCdoc(boolean arrayEstados)
Descripción:	Este método actualiza el estado de un grupo de No Conformidades de la documentación.

Nombre:	ArrayList BuscarNCAceptadas(int id_proyecto)
Descripción:	Este método busca las todas las No Conformidades aceptadas de un elemento de configuración determinado de un producto dado.

**Tabla 3.12 –.Descripción CD: CC\_ GestionarNC.**

<b>Nombre:</b> CC_Controlador_Archivos	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
ruta doc	string
ruta imag	string
<b>Para cada responsabilidad:</b>	
Nombre:	Void CC_Controlador _Archivo(string proyecto, string producto, string ECS, stringURL)
Descripción:	Este método es el constructor de la clase.
Nombre:	Void GuardarFichero(string proyecto, string producto, HtmlInputFile file)
Descripción:	Este metodo guarda un fichero determinado en el servidor.
Nombre:	Void BorrarFichero(string URL)
Descripción:	Este metodo elimina un fichero determinado del servidor.
Nombre:	Void AbrirFichero(string URL)
Descripción:	Este método abre un fichero determinado.
Nombre:	Void GuardarImagen (string producto, string proyecto, HtmlInputFile file)
Descripción:	Este método guarda una imagen determinada en el servidor.

**Tabla 3.13 –.Descripción CD: CC\_ Controlador\_Archivos.**

<b>Nombre:</b> CE_No Conformidad	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
estado	string
cumplimiento	string
clasificación	string
fecha	date

descripción	string
usuario	string
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

Tabla 3.14 –.Descripción CD: CE\_ No Conformidad.

<b>Nombre:</b> CE_NCSW	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
URL_imag	string
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

Tabla 3.15 –.Descripción CD: CE\_NCSW

<b>Nombre:</b> CE_NCDoc	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
marcador	string
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

Tabla 3.16 –.Descripción CD: CE\_ CDoc

### 3.2.6 Diseño de la Base de Datos.

Es en este epígrafe donde se representa el diseño de la base de datos a través del diagrama de clases persistentes y el modelo de datos.

Para el diseño de la BD se contemplan las clases persistentes del Subsistema de Gestión de No Conformidades, así como las del Subsistema de Servicios. Para una mejor comprensión, se diferencian en los diagramas, las tablas que fueron definidas en el subsistema que concierne al presente trabajo.

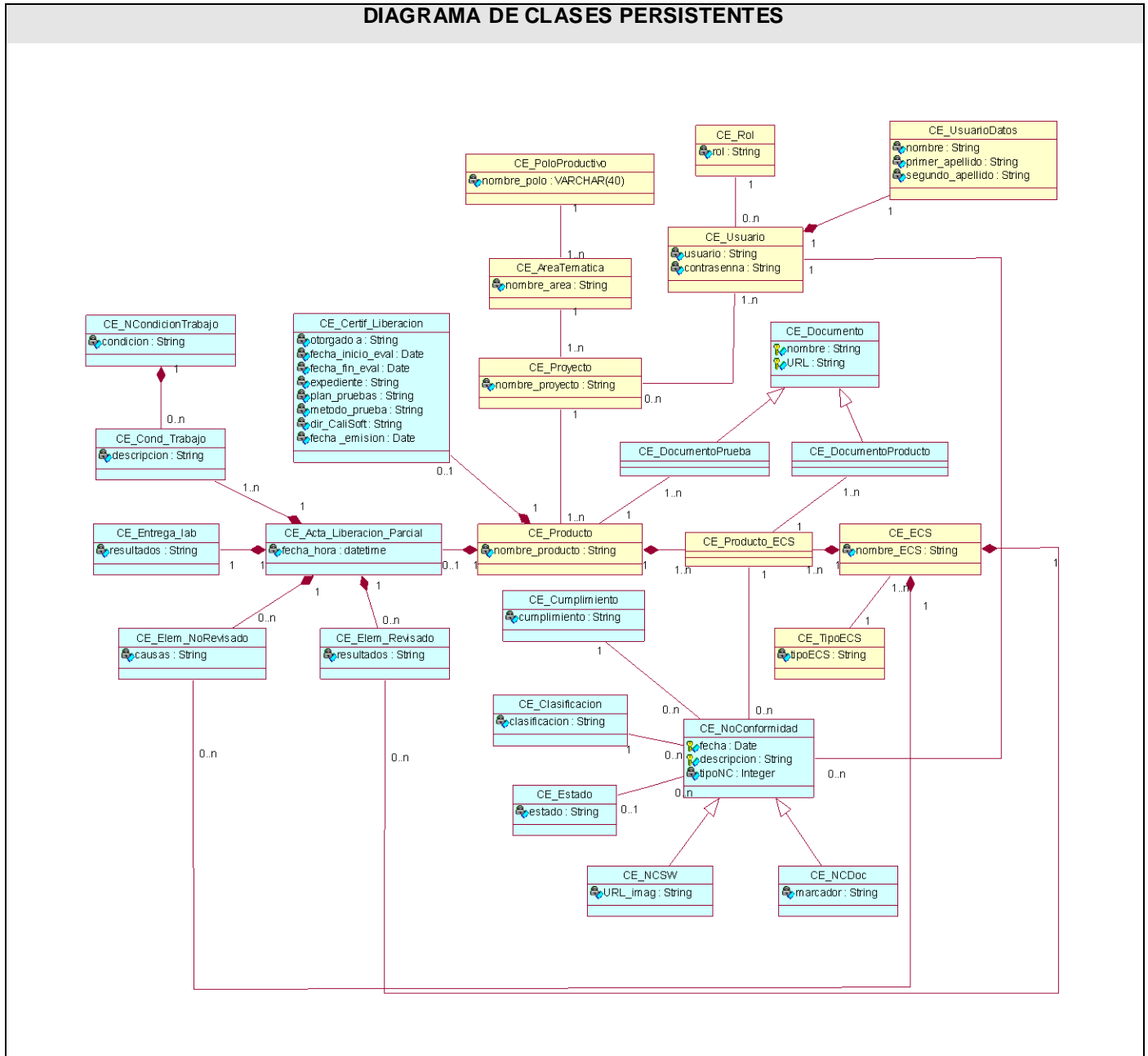


Fig. 3.25--Diagrama de clases persistentes.



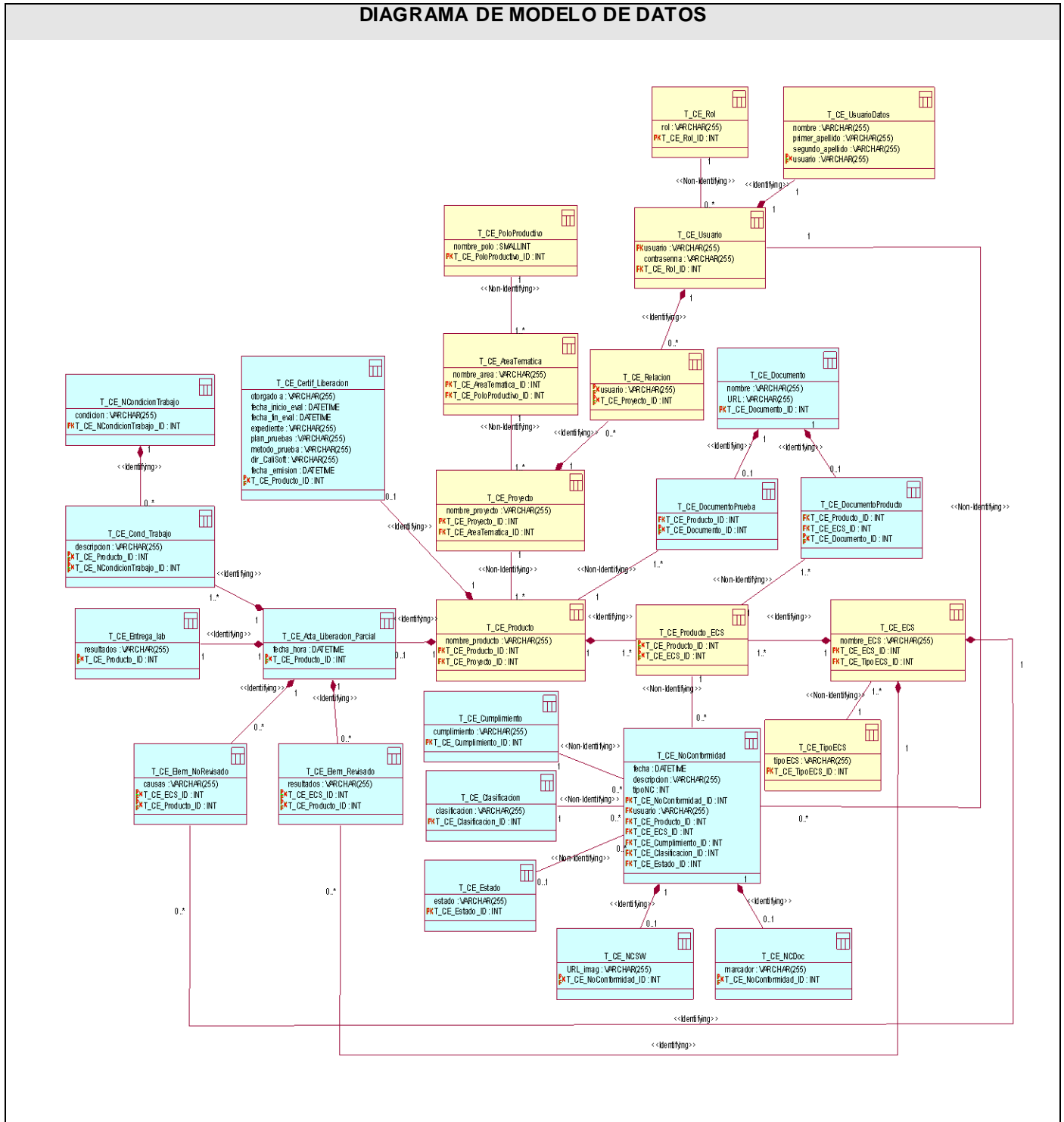


Fig. 3.26 –.Diagrama de modelo de datos.

3.2.6.1 Descripción de las tablas.

<b>Nombre:</b> T_CE_NoConformidad		
<b>Descripción:</b> Esta tabla almacena los datos que son generales de cualquier No Conformidad.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fecha	DATETIME	Fecha en la que se detectó la No Conformidad.
descripcion	VARCHAR	Descripción de la No Conformidad.
tipoNC	INT	Valor que indica el tipo de la No Conformidad, si es 0 es una No Conformidad de software y si es 1 es una No Conformidad de la documentación.
T_CE_NoConformidad_ID	INT	Identificador de la No Conformidad.
T_CE_Producto_ID	INT	Identificador del producto al que se le detectó la No Conformidad.
T_CE_ECS_ID	INT	Identificador del ECS al que se le detectó la No Conformidad.
T_CE_Cumplimiento_ID	INT	Identificador del cumplimiento que tiene la No Conformidad.
T_CE_Clasificacion_ID	INT	Identificador de la clasificación que tiene la No Conformidad
T_CE_Estado_ID	INT	Identificador del estado que tiene la No Conformidad.

Tabla 3.17 -. Descripción T\_ CE\_ NoConformidad.

<b>Nombre:</b> T_CE_NCSW		
<b>Descripción:</b> Esta tabla almacena los datos que son específicos de las No Conformidades de Software.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
URL_imag	VARCHAR	Dirección URL donde se encuentra guardada la imagen que describe gráficamente la No

		Conformidad de Software.(cuando los probadores detectan una NC pueden tomar una foto del sistema en fallo ).
T_CE_NoConformidad_ID	INT	Identificador de la No Conformidad.

Tabla 3.18 –. Descripción T\_ CE\_ NCSw.

<b>Nombre:</b> T_CE_NCDoc		
<b>Descripción:</b> Esta tabla almacena los datos que son específicos de las No Conformidades de la documentación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
marcador	VARCHAR	Nombre del bookmark que se encuentra asociado a la región del documento donde se detectó la No Conformidad.
T_CE_NoConformidad_ID	INT	Identificador de la No Conformidad.

Tabla 3.19 –. Descripción T\_ CE\_ NCdoc.

<b>Nombre:</b> T_CE_Cumplimiento		
<b>Descripción:</b> Esta tabla es nomencladora y almacena todos los cumplimientos que puede tener una No Conformidad.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
cumplimiento	VARCHAR	Cumplimiento que puede tener una No Conformidad, actualmente una No Conformidad puede ser: Cumplida o No Cumplida.
T_CE_cumplimiento_ID	INT	Identificador del cumplimiento.

Tabla 3.20–. Descripción T\_ CE\_ Cumplimiento.

<b>Nombre:</b> T_CE_clasificacion		
<b>Descripción:</b> Esta tabla es nomencladora y almacena todas las clasificaciones que puede tener una No Conformidad.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
clasificacion	VARCHAR	Clasificación que puede tener una No

		Conformidad, actualmente una No Conformidad puede clasificarse como: Leve, Moderada o Crítica.
T_CE_clasificacion_ID	INT	Identificador de la clasificación.

Tabla 3.21 –. Descripción T\_ CE\_ Clasificación.

<b>Nombre:</b> T_CE_estado		
<b>Descripción:</b> Esta tabla es nomencladora y almacena todos los estados que puede tener una No Conformidad.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
estado	VARCHAR	Estado que puede tener una No Conformidad, actualmente el estado de una No Conformidad puede ser Aceptada o Rechazada.
T_CE_estado	INT	Identificador del estado.

Tabla 3.22 –. Descripción T\_ CE\_ Estado.

<b>Nombre:</b> T_CE_Certif_Liberacion		
<b>Descripción:</b> Esta tabla almacena los datos de los Certificados de Liberación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
otorgado a	VARCHAR	Área de la Universidad a la cuál se le otorgó el Certificado.
fecha_inicio_eval	DATETIME	Fecha en la que se inició el proceso de evaluación del producto.
fecha_fin_eval	DATETIME	Fecha en la que se culminó el proceso de evaluación del producto.
expediente	VARCHAR	Expediente en el cuál constan los datos de las evaluaciones realizadas.
plan_pruebas	VARCHAR	Plan de pruebas que se siguió.
metodo_prueba	VARCHAR	Método de prueba que se utilizó.
dir_Calisoft	VARCHAR	Nombre del director de Calisoft.
fecha_emision	DATETIME	Fecha en que se emitió el certificado.

T_CE_Producto_ID	INT	Identificador del producto que se certificó.
------------------	-----	--

**Tabla 3.23 –. Descripción T\_ CE\_ Certificado de Liberación.**

<b>Nombre:</b> T_CE_Acta_Liberacion_Parcial		
<b>Descripción:</b> Esta tabla almacena los datos básicos de las Actas de Liberación Parcial		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fecha_hora	DATETIME	Fecha y hora en que se emitió el Acta de Liberación Parcial.
T_CE_Producto	INT	Identificador del producto que se liberó parcialmente.

**Tabla 3.24 –. Descripción T\_ CE\_Acta de Liberación Parcial.**

<b>Nombre:</b> T_CE_NCondicionTrabajo		
<b>Descripción:</b> Esta tabla es nomencladora y almacena todas las condiciones de trabajo.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
condicion	VARCHAR	Condición de trabajo.
T_CE_NCondicionTrabajo	INT	Identificador de la condición de trabajo.

**Tabla 3.25 –. Descripción T\_ CE\_ NCondición de Trabajo.**

<b>Nombre:</b> T_CE_Cond_Trabajo		
<b>Descripción:</b> Esta tabla almacena los datos referentes a las condiciones de trabajo de los probadores durante la revisión de cada producto que es liberado parcialmente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
descripcion	VARCHAR	Descripción de la condición de trabajo.
T_CE_Producto	INT	Identificador del producto que se revisó bajo esa condición de trabajo.
T_CE_NCondicionTrabajo	INT	Identificador de la condición de trabajo.

**Tabla 3.26 –. Descripción T\_ CE\_ Condición de Trabajo**

<b>Nombre:</b> T_CE_Entrega_lab		
<b>Descripción:</b> Esta tabla almacena los datos referentes a los resultados finales de la revisión de cada producto que es liberado parcialmente.		

Atributo	Tipo	Descripción
resultados	VARCHAR	Resultados finales de la revisión del producto que es liberado parcialmente.
T_CE_Producto	INT	Identificador del producto que es liberado parcialmente.

**Tabla 3.27 –. Descripción T\_ CE\_ Entrega del laboratorio.**

<b>Nombre:</b> T_CE_Elem_NoRevisado		
<b>Descripción:</b> Esta tabla almacena los datos referentes a los elementos de configuración de cada producto que no fueron revisados.		
Atributo	Tipo	Descripción
causas	VARCHAR	Causas por las cuáles no se revisó el elemento de configuración.
T_CE_ECS_ID	INT	Identificador del elemento de configuración que no se revisó.
T_CE_Producto	INT	Identificador del producto al cuál pertenece el elemento que no se revisó.

**Tabla 3.28 –. Descripción T\_ CE\_ Elementos No Revisados.**

<b>Nombre:</b> T_CE_Elem_Revisado		
<b>Descripción:</b> Esta tabla almacena los datos referentes a los elementos de configuración de cada producto que fueron revisados.		
Atributo	Tipo	Descripción
resultados	VARCHAR	Resultados de la revisión del elemento de configuración.
T_CE_ECS_ID	INT	Identificador del elemento de configuración que se revisó.
T_CE_Producto	INT	Identificador del producto al cuál pertenece el elemento que se revisó.

**Tabla 3.29 –. Descripción T\_ CE\_ Elementos Revisados.**

<b>Nombre:</b> T_CE_Documento		
<b>Descripción:</b> Esta tabla almacena los datos generales referentes a los documentos que gestiona el sistema.		
Atributo	Tipo	Descripción
nombre	VARCHAR	Nombre del documento.
URL	VARCHAR	Dirección URL donde se encuentra guardado el documento físicamente.
T_CE_Documento	INT	Identificador del documento.

Tabla 3.30 –. Descripción T\_ CE\_ Documentos.

<b>Nombre:</b> T_CE_DocumentoPrueba		
<b>Descripción:</b> Esta tabla almacena los datos específicos de la documentación referente a las pruebas.		
Atributo	Tipo	Descripción
T_CE_Producto_ID	INT	Identificador del producto al que le elaboró el documento de prueba.
T_CE_Documento	INT	Identificador del documento.

Tabla 3.31 –. Descripción T\_ CE\_ Documento de Prueba.

<b>Nombre:</b> T_CE_DocumentoProducto		
<b>Descripción:</b> Esta tabla almacena los datos específicos de la documentación de cada producto.		
Atributo	Tipo	Descripción
T_CE_Producto_ID	INT	Identificador del producto al que pertenece el documento.
T_CE_ECS_ID	INT	Identificador del ECS con el cuál se corresponde el documento. (Cada documento es un ECS).
T_CE_Documento	INT	Identificador del documento.

Tabla 3.32 –. Descripción T\_ CE\_ Documento del producto.

### **3.2.7 Tratamiento de errores.**

La validación de datos se realiza del lado del cliente, a través de código JavaScript, con el objetivo de disminuir el tráfico innecesario de datos erróneos entre el cliente y el servidor.

Cuando se produce un error en tiempo de ejecución se lanza una excepción interrumpiendo el flujo normal de las sentencias. De esta manera se consigue un diseño mucho más estructurado, legible, robusto y fácil de mantener.

Cuando se introduce un valor incorrecto u ocurre un error en tiempo de ejecución, se informa a los usuarios por medio de mensajes, para que los mismos conozcan y solucionen el problema.

### **3.2.8 Interfaz.**

El sistema propuesto constituirá un módulo del Portal de Calidad de la Facultad 7 por lo que deberá integrarse al mismo.

Existe una pantalla principal desde la cual se accederá, a través del menú, a las diferentes utilidades que brinda el sistema.

El menú estará situado en una barra vertical derecha. Se han definido colores básicos y la utilización de un logotipo que identificará el portal, se usará una tonalidad de colores azul y blanco.

Todo el diseño está determinado fundamentalmente por el principio de usabilidad, teniendo en cuenta que no se trata de un sitio Web, sino de una aplicación de trabajo donde el diseño tiene como principal propósito facilitar su uso, comprensión y navegación, por encima de ornamentos inútiles, aunque manteniendo pautas estéticas y agradables.

La resolución óptima para la cual está diseñada la aplicación es de 1024 x 768 píxeles.

La tipografía será siempre Tahoma, por su amplia legibilidad y por las facilidades conocidas que brinda para la lectura digital.



### 3.2.9 Concepción de la ayuda.

La ayuda debe estar accesible como parte del menú en todas las páginas de la aplicación, con el objetivo de que el usuario vea la información que necesita en ese momento. Ésta cuenta con informaciones generales de la aplicación, explicaciones de la funcionalidad del sistema. Cada página muestra cómo realizar aquellas operaciones que estén relacionadas con la posición donde se encuentre el usuario en dicho momento.

### 3.3 Diagrama de despliegue.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. En general un nodo será una unidad de computación de algún tipo. Muestra la disposición física de los componentes hardware, formando la topología del sistema.

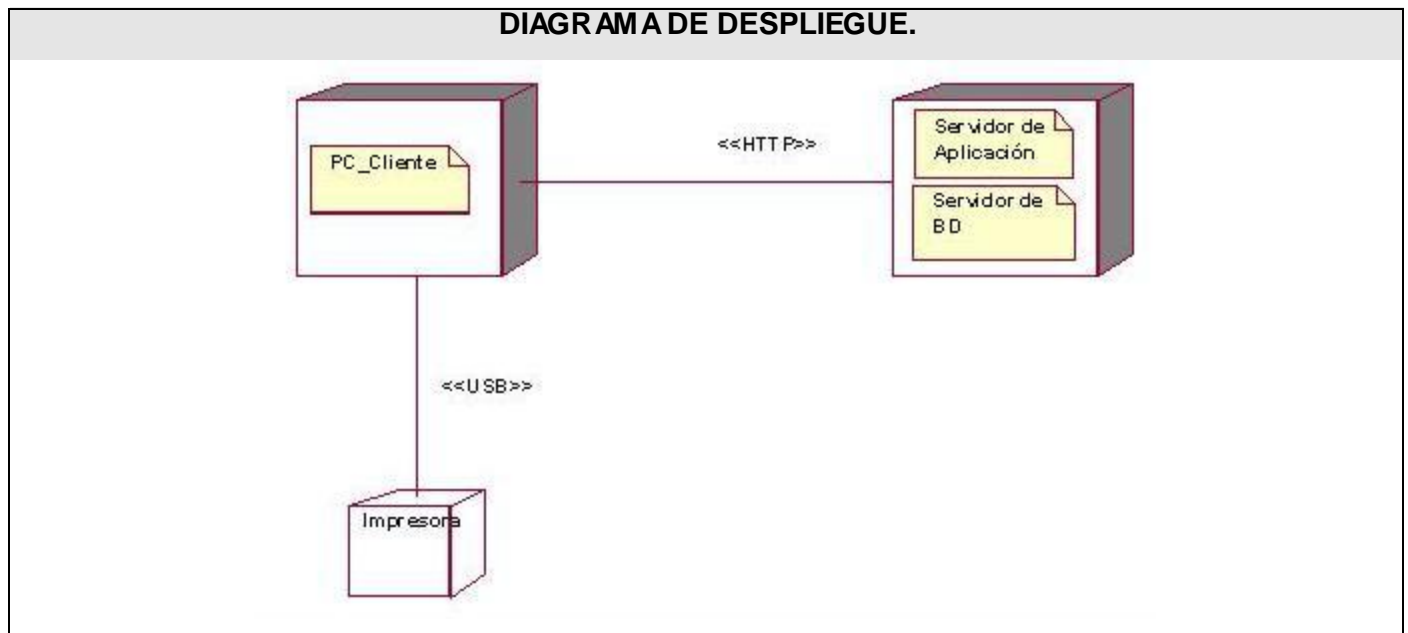


Fig. 3.27 –Diagrama de Despliegue.

Una vez realizado el análisis y diseño del sistema, se arriba a las conclusiones del capítulo:

Se han presentado los diagramas de clases del análisis y del diseño de los casos de uso principales del sistema. Se hace referencia al modelo de datos con el diagrama de clases persistentes que lo origina.

De esta manera, se han logrado modelar todos los procesos que han sido objeto de estudio durante el transcurso del trabajo investigativo, proporcionando una idea completa de lo que realmente es el software.

Se materializan con precisión los requerimientos del cliente.

Los diagramas y especificaciones de diseño que se proponen constituyen una guía que puede ser fácilmente leída y comprendida por aquellos que construirán el código, lo probarán y le darán mantenimiento; lo único que resta es implementar la aplicación que se ha diseñado.

## ***Conclusiones.***

Una vez concluida la investigación, se dieron cumplimiento a los objetivos planteados, obteniéndose los siguientes resultados:

- Se logró seleccionar la metodología y herramientas adecuadas para el desarrollo del sistema informático, según las características particulares del mismo.
- Se realizó un estudio del proceso de seguimiento de errores efectuado durante el control de la calidad de los productos de software desarrollados en la Facultad 7.
- Se realizó el diseño del sistema informático para el seguimiento de errores durante el proceso de control de la calidad de los productos de software desarrollados en la Facultad 7.

## ***Recomendaciones.***

Luego de la presentación del estudio realizado que culmina con el diseño del Sistema para el seguimiento de errores con que contará el Portal de Calidad de la Facultad 7, se listan a continuación una serie de recomendaciones para la ampliación, modificación, mejora y construcción de nuevas versiones de este sistema:

- Implementar el sistema, utilizando el paradigma de Programación Orientada a Objeto.
- Modelar en próximas iteraciones la Gestión de Estadísticas referentes a las No Conformidades, ya que en el presente trabajo solo se realizaron los casos de uso arquitectónicamente significativos para el desarrollo del proyecto.
- Modelar el sistema en iteraciones posteriores de forma tal que, pueda ser extendido a todos los Grupos de Calidad de la Universidad.

## **Referencias Bibliográficas.**

*Apache*. 2006. [2007]. Disponible en: [http://www.augcyl.org/glol/old/N\\_1/apache.html](http://www.augcyl.org/glol/old/N_1/apache.html)

*Aplicación Web*. 2006. [2006]. Disponible en: [http://es.wikipedia.org/wiki/Aplicación\\_web](http://es.wikipedia.org/wiki/Aplicación_web)

*ASP*. 2006. [2006]. Disponible en: <http://www.asptutor.com/>

*ASP.NET*. 2006. [2007]. Disponible en: <http://www.subgurim.net/Articulos/asp-net-general-articulo3.aspx>.

*Cliente-servidor*. 2006. [2006]. Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>

CORSI, M. *Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo"*, 2006. [2006]. Disponible en:

[http://www.masternewmedia.org/es/aplicaciones\\_web/temas\\_de\\_aplicaciones\\_web/Beneficios\\_De\\_Las\\_Aplicaciones\\_Basadas\\_En%20Web\\_Y\\_El\\_Anuncio\\_De\\_Microsoft\\_De\\_La\\_Era\\_En\\_Vivo.htm](http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm)

*Desarrollo ágil de software*. 2007. [2007]. Disponible en:

[http://es.wikipedia.org/wiki/Metodolog%C3%ADas\\_%C3%A1giles](http://es.wikipedia.org/wiki/Metodolog%C3%ADas_%C3%A1giles)

*Diccionario Casadomo*. 2006. [2007]. Disponible en:

<http://www.casadomo.com/diccionario.aspx?id=3521&idm=126&pat=126>

ERAZO, J. *Un gran inicio: Patrones GRASP*, 2006. [2007]. Disponible en:

<http://jorgeerazo.blogspot.com/2006/08/patrones-grasp.html>

FAHNLE, P. *¿Qué es ASP.net?*, 2006. [2007]. Disponible en:

[http://www.programacion.net/asp/articulo/aspnet\\_quees/](http://www.programacion.net/asp/articulo/aspnet_quees/)

*Firefox*. 2006. [2006]. Disponible en: <http://www.mozilla.com/en-US/firefox/>

*Funciones del Navegador*. 2006. [2006]. Disponible en: [http://www.network-press.org/?navegador\\_concepto](http://www.network-press.org/?navegador_concepto)

GARCÍA, A. S. *Introducción a la arquitectura Cliente/Servidor*, 2006]. Disponible en:

<http://www.mygnet.com/articulos/vb/82/>

Grupo de Prácticas de auditoria ISO 9001 Guía sobre: Documentación de una no conformidad, 2005.

GUERRERO, L. A. *Rational Unified Process*, 2006]. Disponible en:

[www.dcc.uchile.cl/~luguerre/cc61j/recursos/clase2.ppt](http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/clase2.ppt)

*Internet Explorer*. 2006. [2006]. Disponible en:

<http://www.microsoft.com/spain/windows/products/winfamily/ie/default.mspx>

- Internet Information Services*. 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/IIS>
- Introducción a Perl*. 2006]. Disponible en: <http://es.tldp.org/Presentaciones/200002hispalinux/conf-14/14-htm/Introduccion.html>
- IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software*. La Habana, 2004. p.
- JARA, O. H. *Sistemas Distribuidos*, 2006]. Disponible en: <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>
- JavaScript*. 2006. [2006]. Disponible en: <http://es.wikipedia.org/wiki/JavaScript>
- JavaServer Pages Technology*. 2006. [2006]. Disponible en: <http://java.sun.com/products/jsp/>
- JOSÉ H. CANÓS, P. L. Y. C. P. *Metologías ágiles en el desarrollo de Software*, 2007]. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- Lenguaje de lado servidor*. 2006. [2006]. Disponible en: <http://www.desarrolloweb.com/articulos/243.php>
- Lenguaje Visual Basic*. 2007. [2007]. Disponible en: <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcn7/html/vaconProgrammingWithVB.asp>
- Lenguajes de lado servidor o cliente*. 2006. [2006]. Disponible en: <http://www.desarrolloweb.com/articulos/239.php>
- Mercado tradicional desarrollado a través de Internet*. Universidad de Oviedo, 2006]. Disponible en: <http://pasm.iespana.es/doc/02-Introduccion.pdf>
- Microsoft SQL Server*. 2006. [2006]. Disponible en: [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)
- MOREA, L. *¿Qué es Internet?*, 1997. [Disponible en: [www.monografias.com/trabajos5/queint/queint.shtml](http://www.monografias.com/trabajos5/queint/queint.shtml)
- MySQL*. 2006. [2006]. Disponible en: <http://www.mysql.com/>
- Opera*. 2006. [2006]. Disponible en: <http://www.opera.com/>
- Oracle*. 2006. [2006]. Disponible en: <http://www.oracle.com/index.html>
- Perl*. 2006. [2006]. Disponible en: <http://www.desarrolloweb.com/articulos/541.php>
- PEROVICH, A. V. Y. D. *Arquitecturas y tecnologías para el desarrollo de Aplicaciones Web*, 2006]. Disponible en: <http://www.fing.edu.uy/inco/grupos/coal/investigacion/publicaciones/vp01.pdf>
- PHP*. 2006. [2006]. Disponible en: <http://www.php.net/>
- PostgreSQL*. 2006. [2006]. Disponible en: <http://www.postgresql.org/>

- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque Práctico*. Quinta. La Habana, Editorial Félix Varela, 2005a. 131 p.
- . *Ingeniería de Software. Un enfoque Práctico*. Quinta. La Habana, Editorial Félix Varela, 2005b. 129 p.
- Programación Extrema*. 2007. [2007]. Disponible en: <http://deigote.blogspot.com/2006/03/extreme-programming.html>
- ¿Qué es Apache?* , 2006. [2007]. Disponible en: <http://www.htmlpoint.com/faq/apache/01.htm>
- Qué es C#* . 2005. [Disponible en: <http://www.desarrolloweb.com/articulos/561.php>
- ¿Qué es Internet?* , 2005. [2006]. Disponible en: <http://www.microsiervos.com/archivo/internet/que-es-internet.html>]
- ¿Qué es Postgree?* , 2005. [2006]. Disponible en:  
<http://www.openecuador.org/modules/news/article.php?storyid=31>
- Rational Rose: Procedimientos básicos para desarrollar un proyecto con UML* 2006]. Disponible en:  
<http://www.vico.org/TallerRationalRose.pdf>
- Rational Unified Process(RUP)*. 2006. [2006]. Disponible en:  
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducción%20a%20RUP.doc>
- SAAVEDRA, J. *Patrones GRASP (Patrones de Software para la asignación General de Responsabilidad).Parte II*, 2007. [2007]. Disponible en:  
<http://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>
- Sistema de Gestión de Base de Datos*. 2006. [2006]. Disponible en:  
[http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_base\\_de\\_datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos)
- UML-Análisis y diseño de Software*. 2006]. Disponible en: <http://cfrela.en.eresmas.com/uml/uml analisis.htm>
- Unified Modeling Language*. 2006. [2006]. Disponible en: <http://www.uml.org/>
- VEGAS, J. *El Servidor Web* 2002. [2006]. Disponible en:  
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>
- Ventajas e inconvenientes de las aplicaciones web*. 2005. [2007]. Disponible en:  
<http://www.avidos.net/blogold/aplicaciones-web/>
- Ventajas y desventajas de PHP*. 2006. [2006]. Disponible en: <http://ascii.eii.us.es/docs/2002-03/php/php4.html>

## ***Bibliografía.***

- Bug-A-Boo - A web based bug tracking system.* 2005. [2006]. Disponible en: <http://www.bug-a-boo.org/>
- Bug Tracking, Free Defect Tracking, Issue Tracking, Free download, Free to use Bug Tracker.* 2005. [2006]. Disponible en: <http://www.vaxtech.com/bug-tracking.htm>
- Bugzilla.* 2007. [2007]. Disponible en: <http://www.bugzilla.org/>
- ERIC CARTER, E. L. *Visual Studio Tools for Office: Using C# with Excel, Word, Outlook, and InfoPath.* 2005. p.
- GNATS.* 2005. [2006]. Disponible en: <http://www.gnu.org/software/gnats/>
- IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software.* La Habana, 2004. p.
- Mantis Bug Tracker.* 2007. [2007]. Disponible en: <http://www.mantisbugtracker.com/>
- PR-Tracker Bug Tracking, Defect Tracking, Problem Tracking Software.* 2007. [2007]. Disponible en: <http://www.prtracker.com/>
- WHITE, K. *CodeTrack: Bug Reporting and Tracking for the Rest of Us!*, 2006. [2007]. Disponible en: <http://kenwhite.sourceforge.net/codetrack/>



## Anexos.

### Anexo 1: Documentos oficiales asociados al proceso.

#### Registro de No Conformidades

##### Registro de Revisiones

Nombre de la Facultad o Empresa: Facultad 7	
Proyecto: Calidad	
Fecha de la Revisión: 8/11/06	Tipo de Revisión: Pruebas Exploratorias

##### De compilación:

[Para los datos del Equipo de Revisión, Objetivos de la Revisión, Actividades Ejecutadas, y Cuestionario de puntos verificados, remitirse a la **Planificación de la Revisión.**]

##### No conformidades detectadas:

[Descripción de las circunstancias en las que fueron generadas las no-conformidades]

CPU	Elementos de configuración	# No Conformidad	Descripción	Clasificación	Estado	Cumplimiento	Detectado por:	Observaciones
Aquí se pone el nombre del Caso de Uso al que pertenece el Caso de Prueba.	Elemento a revisar.ej: Manual de usuario, lista de requerimientos, aplicación.	Especificar el número de la no conformidad. (1,2...)	Breve descripción de la no conformidad	Leve (no implica errores de funcionamiento) <b>Moderada</b> (implica errores de funcionamiento, pero el software continúa funcionando normalmente) <b>Crítica</b> (implica el cese del funcionamiento normal del software)	<b>Aceptada</b> (se acepta la no conformidad) <b>Rechazada</b> (se rechaza la no conformidad)	<b>Si</b> (ya fue resuelta la no conformidad) <b>No</b> (aún no ha sido resuelta la no conformidad)	Nombre y apellidos de la persona que detectó la no conformidad	Especificaciones que ayuden a encontrar más rápido la no conformidad por parte de los desarrolladores, por ejemplo imágenes, etc....

**Acta de Liberación Parcial**

Grupo de Investigación y Control de la Calidad Facultad 7 (UCV)



**Pruebas de Liberación  
Resultado Final  
19 de diciembre del 2008  
07:00 PM**

Proyecto:

**Introducción**

**Elementos revisados y resultados Finales**

Basados en los Resultados Parciales y las no conformidades finales de libeación en esta liberación se llegan a las conclusiones siguientes:



Elemento revisado	Resultado obtenido

**Elementos que no se pudieron revisar y las causas**

A continuación se muestran los elementos que no fueron cubiertos por las pruebas realizadas y sus causas:

Elementos que no se han podido revisar	Causas

**Entregas del laboratorio a los desarrolladores**

Una vez iniciadas las pruebas se notifica el estado a los desarrolladores.

Entrega de resultados de las pruebas

**Condiciones de Trabajo existente.**


(Continuación)

Grupo de Investigación y Control de la Calidad Facultad 7 (UC)



No conformidades aún presentes en el producto.

Elemento de configuración	# No Conformidades	Descripción	Clasificación

Recomendaciones importantes para los Desarrolladores.

Siendo las \_\_\_\_\_ del día \_\_\_\_\_ hacemos entrega del producto \_\_\_\_\_ con el fin de que se ~~evalúen~~ las dificultades planteadas para posteriores implementaciones del producto.

Nota: A partir de este momento de entrega, los desarrolladores tienen la ~~posibilidad~~ de hacer las modificaciones que estimen pertinentes en su producto, pero el grupo de Investigación y Control de la Calidad de la Facultad 7 no se hace responsable de los errores que se puedan originar.

Firman la presente:

Ing. Lourdes Escalona Peral  
Representante de Calidad

Ug. Eduardo Solís Céspedes  
~~Ug. de~~ de producción

~~Reinjer~~ Alonso González  
Lider de Proyecto



## Anexo 3: Modelo de Objetos

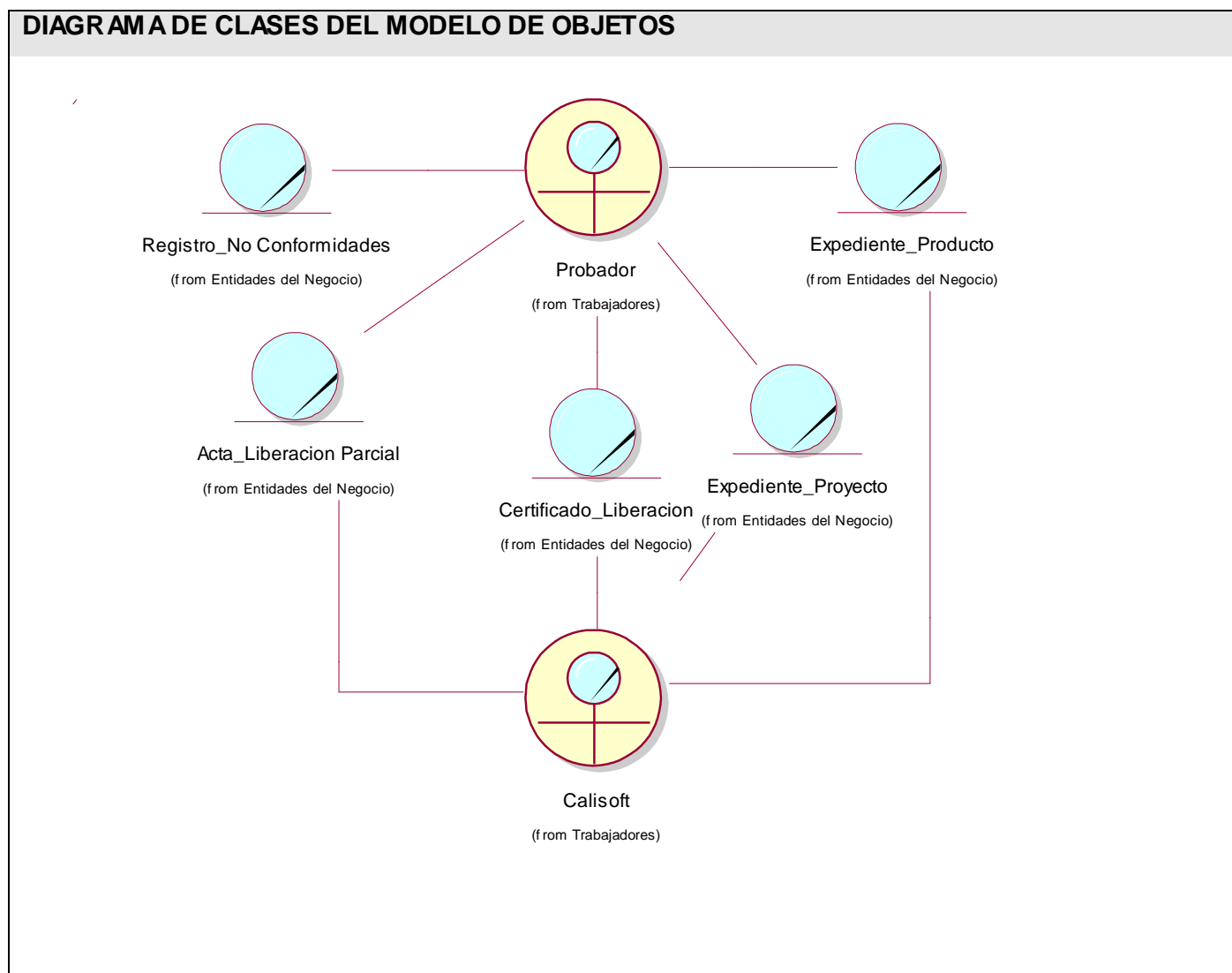


Fig. A3.1 -. Modelo de Objetos

## ***Glosario de Términos.***

**Acciones pro-activas:** Acciones preventivas, antes de que algo suceda.

**Ajax:** JavaScript y XML asíncronos, es una tecnología de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

**API (Application Programming Interface):** Interfaz de Programación de Aplicaciones, es el conjunto métodos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

**Bugs:** Errores.

**Caso de prueba:** Especificación de un caso para probar el sistema, incluyendo qué probar, con qué entradas y resultados y bajo qué condiciones.

**Cookie:** Es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

**DDL (Data Definition Language):** Lenguaje de definición de datos.

**DML (Data Manipulation Language):** Lenguaje de manipulación de datos.

**DOM (Document Object Model):** El Modelo de Objetos de Documento, es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML) como objetos que tienen sus propios métodos y propiedades. En efecto, el DOM es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como Javascript.

**Driver:** Es un controlador de dispositivo, un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz -posiblemente estandarizada- para usarlo.

**ECS (Elemento de configuración del software):** Información creada como parte del proceso de ingeniería de software.

**Emacs:** Es un editor de texto con una gran cantidad de funciones, muy popular entre programadores y usuarios técnicos.

**Frame:** Es la división de la pantalla del navegador en dos o más cuadros que muestran contenidos de dos o más sitios simultáneamente. (También se conocen como marcos).

**Framework:** Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**FTP (File Transfer Protocol):** Protocolo de transferencia de ficheros. Permite la transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

**Hipervínculo:** Es un elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.

**HTTP (HyperText Transfer Protocol):** Protocolo de transferencia de hipertexto. Es el protocolo usado en cada transacción de la Web (WWW).

**HTTPS:** Es la versión segura del protocolo HTTP.

**IRC:** Internet Relay Chat, es un protocolo de comunicación en tiempo real basado en texto, que permite debates en grupo o entre dos personas y que está clasificado dentro de la Mensajería instantánea.

**NNTP (Network News Transport Protocol):** Protocolo para la transferencia de noticias en red. Protocolo usado para la lectura y publicación de artículos de noticias en Usenet (Red de usuarios).

**Parche:** Es una actualización de un programa usado para solucionar problemas que se le hayan detectado o añadirle alguna funcionalidad.

**PDA:** Del inglés Personal Digital Assistant, (Ayudante personal digital) es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura. Hoy día se puede usar como una

computadora doméstica (ver películas, crear documentos, juegos, correo electrónico, navegar por Internet, etc.).

**Plan de pruebas:** Plan que describe las estrategias, recursos y programación de las pruebas.

**RAM (Random Access Memory):** Memoria de Acceso Aleatorio, es donde el computador guarda los datos que está utilizando en el momento presente. El almacenamiento es considerado temporal por que los datos y programas permanecen en ella mientras que la computadora esté encendida o no sea reiniciada.

**Refactorización:** Es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo, puede entenderse como optimizar el código y hacerlo más claro.

**Script:** Un script es un tipo de programa que consiste de una serie de instrucciones que serán utilizadas por otra aplicación, generalmente se almacena en un archivo de texto plano.

**Servlet:** La palabra deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

**SMTP (Simple Mail Transfer Protocol):** Protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos (teléfonos móviles, etc).

**TCP/IP:** Familia de protocolos de Internet, su nombre se debe a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

**Tdbengine:** Es un sistema de administración de base de datos con un lenguaje de programación integrado. Está diseñado para manipular bases de datos en la web.

**Trigger:** Un trigger o un disparador en una Base de datos es un evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).