

Universidad de las Ciencias Informáticas

Facultad 7



**Herramienta para la revisión y seguimiento de
errores de la documentación generada en los
Proyectos de la Facultad 7**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Lester Delgado Pérez

José Rolando Pérez Sandó

Tutora: Ing. Yenisel Molina Hernández

Ciudad de la Habana, Junio 2009

“Año del 50 aniversario del triunfo 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 firmamos la presente a los 15 días del mes de junio del año 2009.

Lesther Delgado Pérez

José Rolando Pérez Sandó

Ing. Yenisel Molina Hernández

“Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación.”

Fidel Castro Ruz

DEDICATORIA

De Lesther:

...a mami y papi

...a mima y abú

...a mi Reina

De José Rolando:

*A mi abuela Ramona, que aunque no me verá graduado puso todo su empeño
y sacrificio en mi crianza.*

A mis abuelos, Mirtha y Antonio, por dedicarme todo el tiempo cada vez que es necesario.

*A mi madre, Elena, por haberme inculcado los deseos de seguirme superando
y dar siempre lo mejor.*

*A mi padre, José Antonio, porque me enseñó que todo el mundo tiene algo que aprender
y que brindar a los demás.*

*A mi hermano Julio Antonio, que luego de todos los dolores de cabeza que me ha dado espero
se convierta en un buen médico, él sabe que es mi súper hermano.*

*A mi hermana Yunieska, que a pesar del poco tiempo la quiero como si la conociera de toda
la vida.*

*A mi tía Reina, por haberme acogido a mí y a mi grupo de amigos durante estos 5 años
como sus hijos.*

*A toda mi familia, que siempre me ha apoyado y ha dedicado su esfuerzo en hacer realidad
mis deseos.*

*A mis amigos de toda la vida, los muchachos del Clan de Moa: Albin, Yoe, Rafa, Keiler,
Yoel, Lucas, Felipe, Arturo, Yasser.*

*A las niñas del Clan de Moa, que tantos dolores de cabeza y alegrías nos han dado: Yadira,
Anabel, Liusmila, Liudmila, Yaima, Milenis, Lienys, Bauza, Linet.*

AGRADECIMIENTOS

De Lesther:

A las personas más importantes de mi vida: mami, papi, mima y abú. Sin los cuales no estaría donde estoy ni sería hoy quien soy.

A Daysel (mi Reina) por su confianza, apoyo y amor incondicional. . .

A mis tías: Mari, Rosi, y Melbi. A mis tíos Evis, Hubert y Joel. Por quererme y querer a los que quiero.

A mis primos Mahelys, Maidelis, Ernestico, Karelito, Mailén, Ana Carla y Evito, porque siempre me han adorado y tenido como ejemplo.

A mis papa-suegros Dago y Elsi, por quererme tanto, tener una hija tan excepcional y llevarme el desayuno a la cama. A Dagel (El Cuña) por su amistad, preocupación y cariño.

A mis amigos de Camaguey, en especial a: Viti, Jose, Wilfre, Yanier y Jiu.

A mis amigos de la UCI en especial a: Juan Pablo, Robert, Alberto, Segui, Dariel, Yosmany, Félix, Maiquel, Osvaldo, Karel (El Punisher), Santi, Dunieski, Aleko.

A mi Bro Karel, por estar ahí siempre, y llamarme cuando necesita que esté yo para él. A sus padres Carlitín y Doriña, a su hermanita Laurent y a su novia Daylenis (la Cuñi).

A José Rolando (El Partner), por ser el mejor compañero de tesis.

A Yenisel por su tutoría ejemplar.

Al tribunal de esta Tesis por señalarnos con paciencia nuestros errores.

A todas las personas que me han ayudado o han ayudado a los que quiero, directa o indirectamente.

A mi Cuba. . .

De José Rolando:

A los familiares de los muchachos y muchachas del Clan de Moa por formar una familia que se quiere y apoya en las buenas y en las malas.

A mi compañero en este trabajo, Lesther, por el esfuerzo realizado en la obtención de los resultados.

A todos los integrantes del Proyecto Calidad de la Facultad 7, en especial para los ingenieros de esta graduación.

A mi tutora, Yenisel, por haber puesto su empeño en este trabajo y contribuir de forma excepcional en la realización del mismo.

A los profes del Proyecto Calidad por la ayuda en la realización de este Trabajo de Diploma y en mi formación.

A los muchachos y muchachas de la Rueda de Casino de la Facultad 7 (Somos Cubanos) y de la Rueda UCI (Movimiento Futuro).

A todos los que han compartido conmigo durante estos 5 años en la Universidad y de una forma u otra me han ayudado a convertirme en Ingeniero.

A los amigos del barrio (Javier, Tico, EL Rubio, Elieski), a toda la gente que compartió conmigo desde la Secundaria y hasta el Pre Universitario y aún me acompañan (sin nombres que se me queda gente y no quiero celos).

RESUMEN

En la Universidad de las Ciencias Informáticas se ha puesto en práctica una estrategia para el Aseguramiento de la Calidad, orientada a perfeccionar el proceso de desarrollo de los productos de software. El Proyecto Calidad de la Facultad 7 ha organizado su estrategia de trabajo con el fin de controlar que los productos de esta facultad cumplan con los requisitos de calidad exigidos. Actualmente existen algunos problemas que influyen en el logro de los resultados deseados, uno de ellos es el seguimiento de los errores detectados en el Proceso de Pruebas a la Documentación.

El objetivo del presente Trabajo de Diploma es obtener un sistema informático para la automatización del seguimiento de errores en el Proceso de Pruebas a la Documentación, que favorezca el proceso de control y aseguramiento de la calidad de los productos de software desarrollados en la Facultad 7.

El uso de esta aplicación facilitará el trabajo en el Proyecto Calidad de la Facultad 7, durante el Proceso de Pruebas a la Documentación, en cuanto a la detección y seguimiento de errores. Además, los miembros de los equipos de desarrollo contarán con una herramienta que hará más ágil la corrección de los errores detectados. Esta herramienta disminuirá el período del Proceso de Pruebas a la Documentación y facilitará la toma de decisiones mediante la generación de reportes actualizados.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1. Conceptos básicos asociados al Negocio	5
1.2. Sistemas automatizados existentes	5
1.2.1. Ámbito Internacional	5
1.2.2. Ámbito Nacional.....	11
1.3. Acercamiento a las tendencias tecnológicas actuales.....	12
1.3.1. Sobre internet.....	12
1.3.2. Aplicaciones WEB. Ventajas y Desventajas.....	13
1.3.3. Servidores WEB	16
1.3.4. Lenguajes de Programación WEB	18
1.3.5. Sistemas Gestores de Base de Datos (SGBD)	23
1.3.6. Arquitectura	30
1.3.7. Metodologías de Desarrollo de Software	33
1.3.8. Herramientas CASE (Computer – Aided Software Engineering)	36
1.3.9. Lenguaje Unificado de Modelado (UML)	39
1.4. Tecnologías y Herramientas a utilizar	40
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	42
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.....	44
2.1.4. Objeto de automatización	45
2.2. Modelo de Negocio	45
2.2.1. Actores del Negocio.....	45
2.2.2. Trabajadores del Negocio	46

2.2.3.	Diagrama de Casos de Uso del Negocio	47
2.2.4.	Descripción Textual de los Casos de Uso del Negocio	48
2.2.5.	Diagrama de Actividades	50
2.2.6.	Diagrama de Clases del Modelo de Objetos	50
2.3.	Especificación de los Requisitos del Software	50
2.3.1.	Requisitos Funcionales	50
2.3.2.	Requisitos No Funcionales	52
2.4.	Definición de los Casos de Uso	53
2.4.1.	Definición de los actores	53
2.4.2.	Listado de los Casos de Uso	54
2.4.3.	Diagrama de los Casos de Uso del Sistema	59
2.4.4.	Listado de los Casos de Uso expandidos	59
CAPÍTULO 3: ANÁLISIS Y DISEÑO		66
3.1.	Análisis	66
3.1.1.	Diagrama de Clases del Análisis.....	66
3.2.	Diseño	68
3.2.1.	Fundamentación del uso de patrones	68
3.2.2.	Definición de elementos de Diseño	68
3.2.3.	Diagrama de Clases de Diseño.....	69
3.2.4.	Diagramas de Interacción (Secuencia)	71
3.2.5.	Descripción de las Clases.....	74
CAPÍTULO 4: IMPLEMENTACIÓN		80
4.1.	Modelo de Datos.....	80
4.1.1.	Diseño de la Base de Datos.....	80
4.1.2.	Descripción de las tablas presentes en la Base de Datos	81
4.2.	Implementación.....	87
4.2.1.	Diagrama de Despliegue.....	87
4.2.2.	Diagrama de Componentes	87

4.3. Tratamiento de errores.....	89
4.4. Seguridad	89
4.5. Estrategia de codificación. Estándares y estilos a utilizar.....	90
CONCLUSIONES	94
RECOMENDACIONES	95
REFERENCIAS BIBLIOGRÁFICAS	96
BIBLIOGRAFÍA.....	101
ANEXOS.....	107
Anexo 1: Diagrama de Actividades.....	107
Anexo 2: Diagrama de Clases del Modelo de Objetos.....	107
Anexo 3: Descripción expandida de los Casos de Uso	108
Anexo 4: Diagramas de Clases del Análisis.....	127
Anexo 5: Diagramas de Clases del Diseño.....	130
Anexo 6: Diagramas de Clases del Diseño.....	136
Anexo 7: Descripción de las Clases del Diseño	152

INTRODUCCIÓN

El desarrollo de las Tecnologías de la Información y las Comunicaciones ha generado a nivel mundial la creación de un nuevo modelo económico basado en el conocimiento, evidenciado en la creciente evolución de las soluciones informáticas. En Cuba, con el fin de llevar este proceso a toda la sociedad, se han creado una serie de condiciones para el crecimiento de la industria de desarrollo de software, siendo uno de los puntos más significativos dentro de este esfuerzo la creación de la Universidad de las Ciencias Informáticas (UCI).

La UCI es una institución en la que se logra el principio de estudio-producción a través de la vinculación a proyectos productivos de los estudiantes. Está constituida por diez facultades docentes en la sede central y 3 facultades regionales, vinculadas a un área específica de la economía en Cuba. Tiene dentro de sus principales objetivos impulsar la informatización de la sociedad e ir a la vanguardia en el desarrollo de software. Para lograr estos lineamientos posee una estrategia centrada en la creación de los Grupos de Calidad, regidos por la Dirección de Calidad de Software. Estos están orientados al logro de la excelencia en las soluciones informáticas de los proyectos productivos existentes en cada facultad.

Actualmente el Proyecto Calidad de la Facultad 7 está orientado a agilizar los procesos de pruebas que en él se realizan. Lo que está determinado por la existencia de múltiples proyectos que abarcan diferentes polos productivos: Polo de Imágenes y Polo de Informática para la Salud. Los distintos proyectos, a la par del desarrollo de los productos de software, deben generar un conjunto de documentos durante todo el transcurso del proceso de desarrollo, los cuales son agrupados en el Expediente de Proyecto. Estos documentos son sometidos a revisión, pues constituyen un elemento de gran importancia en el ciclo de vida que da lugar al producto final.

Las Pruebas a la Documentación realizadas en el Proyecto Calidad de la Facultad 7 se ven afectadas por problemas que entorpecen la obtención de buenos resultados. Entre estos, la necesidad del Probador de interactuar con varios documentos y aplicaciones (Word¹, Paint²) de forma simultánea, pues no se tiene la posibilidad de un entorno unificado de trabajo. El Probador necesita desplazarse por el documento a revisar y además ir generando un

¹ Herramienta del Microsoft Office: Microsoft Word

² Herramienta del Sistema Operativo Microsoft Windows: Microsoft Paint

Registro de No Conformidades. El constante cambio entre ambos documentos limita las posibilidades de detección de errores y la especificidad en la redacción de los ya encontrados. Además, al insertar los anexos, editados en otra aplicación (Ejemplo: Paint), se favorece la inserción de errores, siendo los más comunes el enumerado de los anexos y su correcta correspondencia con la No Conformidad detectada.

Como otro elemento determinante, todos los datos del Registro de No Conformidades son llenados de forma manual; cuando algunos pueden ser generados automáticamente. Lo que disminuiría el periodo de pruebas, evitando omisiones y entradas de datos incorrectos. Todo este proceso finaliza parcialmente con la generación, por parte de cada uno de los Probadores, de un Registro de No Conformidades. Estos tienen que ser unificados por el Jefe de Pruebas, quien revisa los registros recibidos con el fin de generar un único documento que es entregado a los desarrolladores.

Entre los principales objetivos del Proyecto Calidad de la Facultad 7 se encuentra facilitar la interacción entre los Grupos de Desarrolladores y los Probadores. Esta meta se ha visto entorpecida por dificultades en la corrección y seguimiento de los errores por el Equipo de Desarrollo. Pues por lo general, sus miembros no están familiarizados con las formas de trabajo del Proyecto Calidad. Esta dificultad hace difícil encontrar los errores señalados por los Probadores y dar la respuesta requerida. El no contar con un entorno único de trabajo para las partes involucradas en el proceso, acentúa esta diferencia y genera trabajo extra al Equipo de Desarrollo.

Actualmente existen en el mundo varios sistemas que ayudan a la gestión y seguimiento de las No Conformidades detectadas a un producto de software. Estas herramientas tienen como inconveniente principal que no están orientadas a la mejora del Proceso de Pruebas a la Documentación.

En la Universidad se han realizado algunas investigaciones referentes al tema, aunque la mayoría están centradas en la Gestión de las No Conformidades. En estas se obvia todo el proceso que da lugar a las No Conformidades, tanto para las aplicaciones como para su documentación asociada, contribuyendo de forma parcial a la solución del problema. Algunos de los trabajos realizados anteriormente llegan incluso a la propuesta de una aplicación, pero no la implementan o no se tienen resultados de su puesta en práctica.

Por lo anteriormente expuesto y analizado, acorde a las experiencias obtenidas en el Proyecto Calidad de la Facultad 7, se evidencia que el proceso de pruebas realizado a la

documentación no es el más adecuado, identificándose como problema a resolver: ¿Cómo agilizar el proceso de revisión y seguimiento de errores de la documentación generada en los proyectos de 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 la documentación generada en los proyectos de la Facultad 7.

Objetivo general: Desarrollar una aplicación web para la revisión y seguimiento de errores de la documentación generada en los proyectos de la Facultad 7, la cual favorezca el proceso de control de calidad.

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

- Analizar las soluciones informáticas existentes relacionadas con la revisión y seguimiento de errores.
- Valorar las tendencias actuales de las herramientas y tecnologías.
- Seleccionar las herramientas y tecnologías más adecuadas para el desarrollo de la aplicación.
- Describir las problemáticas existentes en el Proyecto Calidad de la Facultad 7.
- Determinar las necesidades de funcionamiento de la aplicación a desarrollar.
- Elaborar la documentación correspondiente al diseño e implementación de la aplicación.
- Implementar la aplicación.

El presente trabajo consta de cuatro capítulos:

En el Capítulo uno **Fundamentación Teórica**, se muestra en detalles el estudio realizado sobre las diferentes herramientas, tecnologías y metodologías que fueron analizadas con el fin de tomar decisiones para la futura construcción del sistema. Además, se hace un análisis sobre el estado de los sistemas existentes en el mundo que abordan la problemática que se desea resolver con la implementación del sistema a desarrollar.

En el Capítulo dos **Características del Sistema**, se hace una valoración del flujo actual de los procesos, realizándose un análisis crítico y valorativo de cómo se ejecutan estos, la información que se maneja en ellos y el objeto de automatización; aspectos que están relacionados con el objeto de estudio; también se muestra una propuesta del sistema,

acompañada de una valoración comparativa en cuanto a ventajas y desventajas con las herramientas ya existentes.

En el capítulo tres **Análisis y Diseño**, se hace referencia al Modelo de Análisis y al Modelo de Diseño. Estos modelos contienen los diagramas de clases de diseño, los diagramas de interacción y la descripción de las clases de diseño.

En el capítulo cuatro **Implementación**, se muestran el diseño de la Base de Datos, el Diagrama de Despliegue y el Diagrama de Componentes. Además se describe el tratamiento de errores, las acciones para garantizar la seguridad y los estándares usados en la codificación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo, se explican los principales conceptos referentes al negocio en cuestión, se muestra el estado del arte de las diferentes herramientas, tecnologías y metodologías existentes para la gestión de errores en general y específicamente para la gestión de errores en la documentación. Por último se hace un análisis de las tendencias, tecnologías y metodologías más usadas en la actualidad y se concluye con la selección de las herramientas a utilizar para el desarrollo de la aplicación.

1.1. Conceptos básicos asociados al Negocio

No Conformidades (NC): Según la definición de ISO³ 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 (RNC): Documento donde se registran todas las no conformidades que se le detectan al producto.

1.2. Sistemas automatizados existentes

Se desee o no, en todo proyecto hay errores, estos deben ser detectados y solucionados debido a que atentan contra la calidad del producto que se desea lograr. Generalmente el control de los errores en los proyectos se ha realizado mediante emails, documentos Word, hojas Excel, etc. En estos momentos se han realizado importantes mejoras en el proceso de seguimiento de errores, mostrándose estas mejoras en la creación de varias aplicaciones a nivel mundial que facilitan y agilizan la gestión de errores durante el proceso de aseguramiento de la calidad del software.

1.2.1. Ámbito Internacional

Bugzilla

Con seguridad es el más conocido de todos los sistemas. Creado por Mozilla, 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, permite llevar un control de tiempo, distintas formas de autenticación, gráficos y

³ Organización Internacional para la Estandarización (International Organization for Standardization)

multilenguaje. Requiere Perl⁴, MySQL o PostgreSQL⁵, servidor web (Apache recomendado), módulos Perl específicos y Mail Transfer Agent⁶.

Características

- Estructura de base de datos optimizada para un mayor rendimiento y escalabilidad.
- Excelente seguridad para proteger la confidencialidad.
- Avanzada herramienta de consultas que pueden recordar sus búsquedas.
- Capacidades de correo electrónico integrado.
- Editar los perfiles de usuario y preferencias de correo electrónico completa.
- Amplio sistema de permisos.
- Probado bajo el fuego de Mozilla como sistema de seguimiento de fallos.

Beneficios

- Mejorar la comunicación.
- Aumentar la calidad de los productos.
- Mejorar la satisfacción del cliente.
- Garantizar la rendición de cuentas.
- Aumentar la productividad.
- Bugzilla puede adaptarse a múltiples situaciones. (Bugzilla.org, 2005)

BUGS - the Bug Genie

Desarrollado por Ruffdogs, fácil de usar, "BUGS" - Bug Genie es una empresa de código abierto del sistema de seguimiento en cuestión, el cual aumenta su proceso de desarrollo, ofreciendo una herramienta avanzada para la gestión de informes de errores, peticiones y comentarios de los usuarios de sus productos, seguro, con historial sobre los errores, mensajería privada, temas y multilenguaje, entre otras características.

Beneficios

- Mejor gestión de usuarios con control de acceso detallado.
- La mejora de funcionalidad de búsqueda con apoyo de la agrupación.
- Mejora de la funcionalidad de mensajes con carpetas de búsqueda.

⁴ Lenguaje de programación web (Practical Extraction and Report Language)

⁵ Gestores de Base de Datos (Ver Epígrafe 1.3.5)

⁶ Agente de Transferencia de Correo

- Mejora de la presentación de informes en cuestión, asistente automatizado con duplicado de búsqueda.
- Artículos y noticias.
- La integración de Subversion (SVN)⁷.
- Calendario de eventos y reuniones.
- Carteleras mundiales y equipo.
- Generación automática de hoja de ruta.
- Apoyo a la cuestión sub-tareas.
- Gran apoyo lingüístico - BUGS 2 viene parcialmente traducido en varios idiomas. (Moody, 2008)

GNATS

Proyecto GNU⁸, soporta distintas bases de datos, además existe una comunidad que desarrolla nuevas extensiones. GNU GNATS es un conjunto de herramientas para el seguimiento de los errores denunciados por los usuarios a un sitio central. Permite la gestión y el informe de problemas de comunicación con los usuarios a través de diversos medios. GNATS almacena toda la información sobre los informes de problemas en sus bases de datos y proporciona herramientas para la consulta, edición y mantenimiento de las bases de datos.

Gracias a su arquitectura, GNATS no está vinculada a una única interfaz de usuario - puede ser utilizado a través de línea de comandos, e-mail, Emacs⁹, o un dominio de red, por lo general se usa con una interfaz web. A esto se suma el hecho de que toda la configuración se puede almacenar en archivos de texto, permitiendo un fácil uso y proporcionando una buena flexibilidad. (Foundation, 2005)

CodeTrack

La 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 se puedan obtener en un

⁷ Sistema de Control de Versiones

⁸ Proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: GNU No es Unix (GNU is Not Unix)

⁹ Editor de texto con varias funciones, muy popular entre los programadores y usuarios técnicos

proyecto). La información la guarda en archivos de tipo XML¹⁰ y tan sólo es necesario tener instalado Apache y PHP¹¹. De fácil instalación, funciona en navegadores antiguos y seguro, con características que hacen que sea una aplicación sencilla y útil para crear, revisar y gestionar los informes de error y respuestas de desarrolladores para múltiples proyectos de una interfaz común. Especialmente adecuado para entornos de intranet y extranet, CodeTrack incluye una fuerte autenticación, y permite el control de acceso personalizado a los proyectos individuales.

Beneficios

- CodeTrack es rápida: búsqueda de miles de errores en cuestión de segundos.
- Instalación Trivial- Se instala en menos de cinco minutos.
- Realiza un seguimiento de las cuestiones a través de múltiples proyectos.
- Completa la historia y la auditoría de error por defecto.
- Fuerte autenticación fuera de la caja.
- Soporte completo para navegadores más antiguos (como Netscape 4.x).
- No hay base de datos necesarios para el backend¹².
- No se necesita servidor de correo.
- No hay necesidad de ocultar las bibliotecas compartidas para extender su sistema a través de los directorios.
- Permite restringir el acceso según el usuario o papel, por sistema o por proyecto.
- La aplicación entera es un solo programa PHP. (Sourceforge.net, 2008)

Mantis

Desarrollado en PHP, siendo necesario MySQL y un servidor web (Apache a ser posible), este bug tracker¹³ tiene entre sus características el ser fácil de instalar, permitir varios proyectos, varios lenguajes, emails, búsquedas y filtros.

¹⁰ Lenguaje de Marcas Ampliable (eXtensible Markup Language)

¹¹ Acrónimo recursivo: PHP Hypertext Pre-processor (inicialmente Personal Home Page Tools)

¹² Término usado para definir el final de un proceso

¹³ Término usado para definir algunas herramientas usadas en la gestión de errores

MantisBT es un servicio gratuito basado en la web. Está escrito en el lenguaje de scripting PHP, y trabaja con MySQL, MS SQL¹⁴, bases de datos PostgreSQL y un servidor de web. MantisBT se ha instalado en Windows, Linux, Mac OS, OS / 2, y otros. Casi cualquier navegador debería ser capaz de funcionar como un cliente. Es liberado bajo los términos de la GNU (GPL¹⁵).

La última versión estable es la 1.1.6. La última versión en desarrollo es 1.2.0a3. (Group, 2009)

Bug-a-Boo

Solo para Linux, permite temas¹⁶ y está disponible en inglés y alemán. Bug-A-Boo es una web basada en el sistema de reporte de errores para Linux. Es liberado bajo los términos de la GNU (GPL).

Funciona en cualquier servidor web, proporciona funcionalidad CGI¹⁷. Bug-A-Boo no necesita ningún servidor de base de datos, pero trae sus propias tablas locales, se accede con Tdbengine¹⁸. Por lo tanto, es fácil de configurar y un verdadero monstruo de rendimiento. Cuenta con múltiples idiomas, tres diferentes niveles de usuario, diseños y temas, las notificaciones por correo electrónico y los perfiles de visualización para todos los usuarios. (Bug-a-Boo.org, 2005)

Zentrack

También diseñado para ayuda de helpdesk¹⁹, con una interfaz bastante configurable, permite controlar el flujo de trabajo y el manejo de los datos, control de los tiempos y de las funcionalidades para los usuarios y lista de contactos.

Beneficios

¹⁴ Siglas para definir el gestor de base de datos Microsoft SQL (Ver epígrafe 1.3.5)

¹⁵ General Public License

¹⁶ Término usado para referirse a los cambios y la configuración de un ambiente de trabajo en un software

¹⁷ Interfaz de entrada común (Common Gateway Interface)

¹⁸ Sistema gestor de base de datos con un lenguaje de programación integrado, especificado para el manejo de bases de datos en la web

¹⁹ Término para definir un sistema de ayuda brindado por las organizaciones a sus usuarios

- Campos personalizados - agrega sus propias fechas, números, cadenas.
- Comportamientos - dependencias complejas y listas.
- Realiza un seguimiento de horas, las estimaciones, y los plazos.
- Interfaces y adaptación.
- Pasarela de correo electrónico para crear y responder a las entradas.
- Recordatorios de artículos atrasados.
- Usabilidad y la interfaz.
- Aspecto personalizable.
- Control de acceso en la pantalla por pantalla, campo por campo base.
- Atajos de teclado.
- Diseño compacto. (Zentrack.net, 2009)

Hipergate

Una herramienta muy completa que ofrece un módulo de Gestión de Proyectos (seguimiento de tareas y control de incidencias). Todo esto además de herramientas colaborativas, gestión de contactos, etc. Con una licencia basada en GPL y desarrollado en Java (necesita un servidor de aplicaciones J2EE²⁰, al menos un contenedor servlets/jsp) y admite varias bases de datos: PostgreSQL, Oracle, y SqlServer. Hipergate es una suite de aplicaciones de código abierto basadas en web.

El propósito del conjunto de programas es cubrir un amplio rango de necesidades de tecnología de información en cualquier organización. Todas las aplicaciones se acceden desde Internet Explorer sin necesidad de descargar ningún software en el puesto cliente.

La suite tiene capacidad multi-entidad y puede utilizarse indistintamente para dar servicio a una empresa, a un grupo de empresas, o en modalidad ASP²¹ para alojar un número ilimitado de entidades clientes diferentes.

¿Qué hace a Hipergate único?

- Es completamente gratuito.
- Es la única aplicación Open Source con una cobertura funcional tan amplia.

²⁰ Java 2 Platform, Enterprise Edition

²¹ Active Server Pages (Ver Epígrafe 1.3.4)

- Funciona sobre múltiples bases de datos y sistemas operativos.
- Tiene un diseño conceptual e implementación unitarios.
- Es apropiado para su uso en modalidad ASP.
- Es multi-idioma.
- Dispone de una red de servicios profesionales de soporte.
- Está bien documentado.
- Hipergate es software de última generación; diseñado, programado y testeado con las herramientas y técnicas más modernas. (Hipergate.org, 2008)

Trac

Se integra con SVN y permite navegar por el código, abrir tickets y hacer una buena gestión del proyecto. Escrito en Python, admite las bases de datos SQLite, PostgreSQL y MySQL.

Trac utiliza un enfoque minimalista basado en la web de software de gestión de proyectos. Su misión es ayudar a los desarrolladores a escribir software de gran mantenimiento durante el camino que acompaña la construcción del software. Trac debe imponer lo menos posible sus prácticas en un equipo establecido en proceso de desarrollo y políticas. Proporciona un interfaz para Subversion. (Project, 2008)

1.2.2. Ámbito Nacional

En la UCI actualmente, al igual que en el resto de las instituciones que desarrollan software en el país, se realizan acciones para incidir en la mejora del proceso de detección y corrección de errores de los productos informáticos. Para ello se comienzan a utilizar algunas herramientas de las antes mencionadas. A su vez se han llevado a cabo algunas investigaciones dirigidas fundamentalmente a buscar otras soluciones que se adapten a las características particulares de los procesos que se llevan a cabo en la Universidad.

Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad: La investigación tiene como objetivos la modelación e implementación de un sistema que posibilite obtener reportes a partir de la información contenida en los Registros de No Conformidades, conocer en qué estado se encuentran los indicadores de calidad que se miden en el proyecto y gestionar el acceso a la información de los usuarios que harán uso de la misma. Los principales requerimientos del sistema son los siguientes:

1. Gestión de Usuarios y sus roles en el Sistema.
2. Gestión de Reportes basados en los Registros de No Conformidades.
3. Gestión de No Conformidades.

Este sistema no contemplaba dentro de sus funcionalidades las respuestas brindadas por los desarrolladores a las No Conformidades, tampoco estaba centrado en las particularidades de un Proceso de Pruebas a la Documentación. Actualmente no existe información sobre la completa implementación de la herramienta, pues en las conclusiones del trabajo no se menciona. (Morrell Rojas, 2007)

Herramienta Informática para automatizar los procesos en el Laboratorio de Calidad de Software: Módulo Gestión de las No Conformidades: La investigación está asociada al surgimiento del Laboratorio Central de Certificación de Software en la Universidad de las Ciencias Informáticas. Esta investigación culminó con el desarrollo de un producto informático. El sistema tenía concebido como sus principales funcionalidades las siguientes:

1. Gestionar los trabajadores de la revisión.
2. Gestionar las No Conformidades y sus respuestas.
3. Emitir reportes
4. Realizar la distribución de trabajo.

Como se aprecia, la herramienta estaba orientada a la gestión de forma general de las No Conformidades, sin tener en cuenta las particularidades del Proceso de Pruebas a la Documentación. Actualmente no existe información sobre los resultados que se lograron con la puesta en práctica de esta herramienta dentro o fuera de la Universidad. (Nieves Borrero, et al., 2007)

Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7: El objetivo del trabajo se enmarca en 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. Lo cual facilite el trabajo de los miembros del Proyecto Calidad de la Facultad 7 en cuanto a la detección y seguimiento de errores, tanto del software como de la documentación generada durante el proceso de desarrollo, además de brindar facilidades en la elaboración de los documentos legales que se generan en el proceso de revisión. Culminó con la propuesta de desarrollo de un sistema informático, constituyendo la base del presente Trabajo de Diploma, el cual se enmarca en la implementación del proceso correspondiente al Proceso de Pruebas a la Documentación. (Díaz Colomé, et al., 2007)

1.3. Acercamiento a las tendencias tecnológicas actuales

1.3.1. Sobre internet

Algunos de los nombres más usados para definir Internet son “Red de Redes” o “Autopista de la Información”. Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores. Además, es considerada “La Red de Redes” porque es la más grande.

Por Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Actualmente existen alrededor de 50 millones de personas con acceso a este recurso. La información circula de manera casi instantánea, lo que hace que el mundo se encuentre conectado a cada segundo, y con ello, se pueden realizar actividades como transacciones económicas, comunicación a distancia, etc. (Zamora, 2008)

Uno de los fundamentos de Internet es el TCP/IP²², un protocolo de transmisión que asigna a cada máquina que se conecta un número específico, llamado «número IP». El protocolo TCP/IP sirve para establecer una comunicación entre dos puntos remotos mediante el envío de información en paquetes.

Internet brinda además una serie de servicios, los más usados son Correo Electrónico, World Wide Web, FTP²³, Grupos de Noticias, IRC²⁴ y Servicios de Telefonía. (Hervás Gómez, 2006)

1.3.2. Aplicaciones WEB. Ventajas y Desventajas

Con la llegada de la Web 2.0²⁵, propiciada esta por el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y los nuevos retos de Internet, se abre paso una nueva idea en la concepción del software; el software orientado a servicios o SaaS (acorde a sus siglas en inglés). SaaS es un modelo de distribución del software que proporciona a los clientes el acceso al mismo a través de la red (generalmente Internet), de manera que

²² Protocolo de Control de Transmisión/ Protocolo de Internet (Transmisión Control Protocol/Internet Protocol). Usados para definir la familia de protocolos de Internet

²³ Protocolo de Transferencia de Archivos (File Transfer Protocol)

²⁴ Protocolo de Comunicación en Tiempo Real basado en texto (Internet Relay Chat)

²⁵ El término **Web 2.0** fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

les libra del mantenimiento de las aplicaciones, de operaciones técnicas y de soporte. Las aplicaciones distribuidas en la modalidad SaaS pueden llegar a cualquier tipo de empresa sin importar su tamaño o su ubicación geográfica. Se trata de un modelo que une el producto (software) al servicio, para dotar a las empresas de una solución completa que permita optimizar sus costes y sus recursos. (CIIN)

Ventajas WEB: (Graham, 2001)

- **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, pues no se hace necesario iniciar nuevas descargas y/o 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 RAM²⁶ 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 sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs²⁷:** 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.

²⁶ Memoria de Acceso Aleatorio (Random Access Memory)

²⁷ Errores presentes en la aplicación que atentan contra su correcto funcionamiento.

- **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:** El hecho de que el manejo de los datos sea realizado de forma remota libra al usuario de la responsabilidad en la protección de los mismos, y al mismo tiempo logra que los recursos sean accesibles en cualquier momento.
- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en Web pueden ser utilizadas 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.
- **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 hagan 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 proveen 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.
- **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. Debido a que las aplicaciones Web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación existente.

Desventajas WEB: (Masternewmedia.org, 2005)

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera hasta que se obtiene la reacción del sistema.

- 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.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.3.3. Servidores WEB

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP²⁸ 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)

Actualmente los servidores Web más difundidos son: Apache e Internet Information Server o Services (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 parches (patch) para mejorar su fiabilidad y sus características; de ahí su nombre. (Htmlpoint.com, 2008)

Características: (Ciberaula.com, 2005)

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita con un código fuente disponible. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esta característica le ofrece al software un grado de transparencia tal que es posible determinar en todo momento qué es lo que se está instalando, sin secretos ni puertas traseras.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se

²⁸ Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol)

necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

- Trabaja con Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas JSP²⁹. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs³⁰. Permite la creación de ficheros de log a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Se pueden extender las características de Apache hasta donde la imaginación y los conocimientos lleguen, debido a que el equipo de desarrollo está formado por voluntarios, diseminados por todo el mundo, que sigue manteniendo este servidor de web libre.

IIS (Internet Information Server o Services)

Es un servidor de Microsoft, destinado a la publicación, mantenimiento y gestión de páginas y portales Web.

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, SMTP³¹, NNTP³² y HTTP/HTTPS³³. (González, 2007)

²⁹ Java Server Pages (Ver Epígrafe 1.3.4)

³⁰ Término para definir la creación de un fichero donde se registran las acciones de los usuarios

³¹ Protocolo Simple de Transferencia de Correo (Simple Mail Transfer Protocol)

³² Protocolo para la Transferencia de Noticias en red (Network News Transfer Protocol)

³³ Protocolo Seguro para la Transferencia de Hipertexto (HyperText Transfer Protocol Secure)

Este servicio convierte a un ordenador 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).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. (Microsoft.com, 2007)

1.3.4. Lenguajes de Programación WEB

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK³⁴.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994. (Achour, et al.)

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.
- 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.
- Biblioteca nativa de funciones sumamente amplia e incluida.

³⁴ Conjunto multiplataforma de bibliotecas para el desarrollo de interfaces gráficas de usuarios

- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

Desventajas de PHP

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, unicode, etc.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Está diseñado especialmente para un modo de hacer aplicaciones web que es ampliamente considerado problemático y obsoleto (mezclar el código con la creación de la página web). (Agustín, 2007)

JSP

La tecnología Java para la creación de páginas web con programación en el servidor.

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java. Java Server Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. (Valparaiso, 2006)

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

Con JSP se puede crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podrán escribirse con el editor HTML/XML habitual. (Millán Tejedor, 2005)

ASP

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). (Álvarez, 2001)

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.

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 Microsoft Visual Basic, C#, o cualquier otro lenguaje que soporte la plataforma .NET. El código MSIL se compila con posterioridad a código nativo. (Villaverde Masa, 2007)

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

- **Rendimiento:** la aplicación se compila 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 mejor rendimiento. También permite el almacenamiento del caché en el servidor
- **Rapidez en programación:** mediante diversos controles, se puede 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 las aplicaciones.

ASP.NET

No es simplemente una actualización de la tecnología para desarrollo de aplicaciones web, sino que es un nuevo enfoque de cómputo para el desarrollo de software. Adquiere de la tecnología de JAVA muchas de sus funcionalidades pero con acceso más sencillo y con la

facilidad de poder emplear diferentes lenguajes de codificación sin perder potencia de diseño, ya que los principales: C#, Visual Basic.NET y Jscript.NET son orientados a objetos. Además, gracias a su nuevo modo de desarrollo de formularios web empleando los controles web se puede concluir diciendo que, en conjunto, el desarrollo de cliente a nivel funcional, no en términos estéticos, es más poderoso que los mejores clientes de Flash MX.

- 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.
- 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 es muy útil almacenar datos de la base de datos en la caché, reduciendo enormemente el consumo de recursos.
- Carpetas especializadas, como 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 la 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 servir para el Internet Explorer, para el Pocket Internet Explorer desde una PDA³⁵ 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.

³⁵ Asistente Digital Personal (Personal Digital Assistant)

- Se puede utilizar hasta cuarenta lenguajes distintos para el desarrollo en ASP.NET, aunque en el 95% de las aplicaciones se usa C#, Visual Basic.Net (VB.NET) o J#. (Subgurim.NET, 2006)

Lenguaje de Programación C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la Ecma International³⁶ e ISO.

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. (Katrib, et al., 2008)

Visual Studio Tools for 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 y las versiones posteriores a este. Los desarrolladores pueden utilizar lenguajes .NET tales como Visual Basic .NET o C#. Además, las Primary Interop Assemblies (PIAs) permiten al código .NET comunicarse con el modelo de objetos no administrado que cada una de las aplicaciones Office expone. (Díaz Colomé, et al., 2007)

¿Qué son las PIAs?

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 COM³⁷. El código administrado (C# o Visual Basic) para interoperar con estas interfaces se

³⁶ Organización internacional basada en membresías de estándares para la comunicación y la información

³⁷ Por sus siglas en inglés: Component Object Model

comunica a través de un componente. Este componente no es más que un conjunto de clases .NET compiladas dentro de un ensamblado llamado PIA.

Estos ensamblados de interoperabilidad primarios de Microsoft Office facilitan la interoperabilidad entre el código administrado y las bibliotecas de tipos COM de Office. Los ensamblados de interoperabilidad COM (modelo de objetos componentes) de Microsoft permiten que el código no administrado (COM) se llame desde código administrado (.NET) mediante Microsoft .NET Framework y el módulo de tiempo de ejecución de lenguaje común. Los ensamblados de interoperabilidad COM permiten que las aplicaciones administradas se enlacen con tipos no administrados en la compilación. A continuación, la aplicación puede proporcionar información al entorno de tiempo de ejecución de lenguaje común acerca de cómo se deben organizar los tipos no administrados en tiempo de ejecución. (Support, 2007)

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.

1.3.5. Sistemas Gestores de Base de Datos (SGBD)

Un **Sistema Gestor de Bases de Datos (SGBD) o DBMA**³⁸ es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Algunos ejemplos de SGBD son Oracle, PostgreSQL, MySQL, MS SQL Server, etc.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.

³⁸ Por sus siglas en inglés: Database Management System

- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Características de un Sistema Gestor de Base de Datos

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra protegida frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipularla o destruirla; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

- **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias. (CAVSI, 2004)

Oracle

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado 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.
- Soporte multiplataforma.

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio provocan que sólo se vea en empresas muy grandes y multinacionales, por norma general. (Masip, 2002)

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS³⁹ con licencia libre como PostgreSQL, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

PostgreSQL

Es un servidor de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD⁴⁰.

³⁹ Sistema Administrador de Bases de Datos Relacionales (Relational Database Management System)

⁴⁰ Licencia de software libre permisiva (Berkeley Software Distribution)

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Principales características:

- Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente Multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- Amplia variedad de tipos nativos. PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria.
 - Texto de largo ilimitado.
 - Figuras geométricas (con una variedad de funciones asociadas)
 - Direcciones IP (IPv4 e IPv6).
 - Bloques de direcciones estilo CIDR⁴¹.
 - Direcciones MAC.
- Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS. (Eaprende.com, 2001)

Otras características:

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*)
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

⁴¹ Encaminamiento Inter-Dominios sin Clases (Classless Inter-Domain Router)

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x.

- **DBMS Objeto-Relacional:** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.
- **Altamente Extensible:** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Soporte SQL Comprensivo:** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **API Flexible:** La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL⁴², C/C++, y Pike.
- **Lenguajes Procedurales:** PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- **Cliente/Servidor:** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. (Sánchez González, 2004)

Gracias a su licencia BSD, se permite la utilización del código para ser comercializado. Uno de los casos ejemplo es la de Enterprise DB (Postgresql Plus), la cual incluye varios agregados y una interfaz de desarrollo basada en Java. Entre otras empresas que utilizan Postgresql para comercializar se encuentra CyberTech (Alemania), con su producto CyberCluster.

⁴² Lenguaje de Herramientas de Comando (Tool Command Language)

MySQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM⁴³ en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. (Nieves Borrero, et al., 2007)

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

⁴³ International Business Machines (conocida también como el Gigante Azul)

VENTAJAS

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

DESVENTAJAS

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS). (Eaprende.com, 2001)

SQL Server

SQL Server es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas. En el centro de SQL Server está el motor de SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.

SQL Server incluye herramientas para la administración de los recursos que el ordenador proporciona y los gestiona para un mejor rendimiento de la base de datos.

Transact-SQL es el lenguaje que utiliza SQL Server para poder enviar peticiones tanto de consultas, inserciones, modificaciones, y de borrado a las tablas, así como otras peticiones que el usuario necesite sobre los datos. En definitiva, es un lenguaje que utiliza SQL Server para poder gestionar los datos que contienen las tablas.

El lenguaje estándar SQL (Structured Query Language) se emplea para los sistemas de bases de datos relacionales RDBMS (Relational Database Management System), es el estándar ANSI (American National Standards Institute). (FormaSelect)

Características de Microsoft SQL Server:

- Soporte de transacciones.

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL⁴⁴ y DML⁴⁵ gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita. (Islasoft.com, 2006)

SQL Server Express Edition

SQL Server Express es una versión de SQL Server 2005 diseñada para ayudar a los desarrolladores a construir aplicaciones robustas y fiables ofreciendo una sencilla pero potente base de datos que es además gratuita. SQL Server Express ya está incluida como opción al instalar cualquier herramienta Visual Studio Express, por lo que no necesitas descargarla aparte si ya estás instalando cualquiera de ellas.

SQL Server 2005 Express Edition proporciona muchas nuevas funcionalidades mejoradas, como son:

- Una base de datos robusta para crear aplicaciones dinámicas
- Fuerte soporte de XML
- Herramientas y características para mejorar la gestión y facilidad de uso
- Instalación y setup⁴⁶ sencillos
- Escalabilidad y rendimientos
- Gran integración con Visual Studio 2005
- Robusta seguridad (Microsoft.com)

(Augusto de la Paz 2009)

1.3.6. Arquitectura

⁴⁴ Lenguaje de Definición de Datos (Data Definition Language)

⁴⁵ Lenguaje de Manipulación de Datos (Data Manipulation Language)

⁴⁶ Término para definir un programa de preparación, montaje y configuración de un sistema

Modelo Cliente-Servidor

El esquema cliente-servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí".

Los principales componentes del esquema cliente-servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones. (Monografias.com, 2007)

Modelo cliente-servidor de 2 capas

Las aplicaciones cliente-servidor clásicas o de 2 capas, como su nombre lo indica, agrupan la lógica de presentación (interfaz) y la lógica de aplicación en la máquina cliente y acceden a fuentes de datos compartidos a través de una conexión de red que se encuentran en el servidor de datos.

La ventaja que presenta este tipo de aplicaciones es que los datos están centralizados. Esta centralización beneficia a la empresa pues es más fácil compartir los datos, se simplifica la generación de reportes y se proporciona consistencia en el acceso a los datos.

A continuación se enumeran las principales desventajas que presentan este tipo de aplicaciones:

- **Difíciles de mantener:** Esto viene dado por el hecho de que son difíciles de mantener las reglas de negocio de la lógica de aplicación ya que estas están programadas en cada cliente y esto implica que cualquier cambio tiene que ser redistribuido en todos los clientes.
- **Se compromete la confidencialidad:** Al tener programada la lógica de aplicación en el cliente este tiene a su disposición todas las reglas de negocio de la empresa.
- **Están estrechamente limitadas a una fuente de datos:** Los clientes casi siempre están configurados para acceder a una base de datos en particular por lo que mover los datos a una base de datos diferente se hace realmente complicado. (Janium.com, 2009)

Modelo cliente-servidor de 3 capas

En la arquitectura de 3 capas la presentación, la lógica de aplicación y los elementos de datos están conceptualmente separados. Los componentes de la capa de presentación manejan la interacción con el usuario y realizan las peticiones del cliente a los componentes de la capa intermedia. Los componentes de la capa intermedia, manipulan la lógica de negocio y hacen las peticiones a la base de datos.

A continuación se enumeran algunas ventajas de las aplicaciones de 3 capas:

- Los componentes de la aplicación pueden ser desarrollados en cualquier lenguaje general lo que posibilita que el grupo de desarrolladores no se centre en el uso de un solo lenguaje.
- Los componentes están centralizados lo que posibilita su fácil desarrollo, mantenimiento y uso.
- Los componentes de la aplicación pueden estar esparcidos en múltiples servidores permitiendo una mayor escalabilidad.
- Los problemas de limitación para las conexiones a las bases de datos se minimizan ya que la base de datos solo es vista desde la capa intermedia y no desde todos los clientes. Además que las conexiones y los drivers de las bases de datos no tienen que estar en los clientes.
- Los componentes de aplicación de la capa intermedia pueden ser asegurados centralmente usando una infraestructura común. Se pueden conceder o denegar los permisos componente a componente simplificando la administración. (Ronda Amador, et al., 2002)

Modelo–Vista–Controlador

La arquitectura Model-View-Controller surgió como patrón arquitectónico para el desarrollo de interfaces gráficas de usuario en entornos Smalltalk. Su concepto se basa en la necesidad de reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, mediante la división de la aplicación en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización. (Aguilar, 2009)

Esta arquitectura ha demostrado ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma J2EE. (Froufe, 2005)

1.3.7. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Pueden ser comparadas con un plan de contingencias en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quienes deben participar en el desarrollo de las actividades y qué papel deben de tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Metodologías Ágiles

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía "ágil". (Methodologies, 2008)

XP

Este es el método que más popularidad ha alcanzado entre las metodologías ágiles, y posiblemente sea también el más transgresor de la ortodoxia basada en procesos. Su creador, Kent Beck fue el alma mater del Manifiesto Ágil.

Extreme Programming (XP) se levanta sobre la suposición de que es posible desarrollar software de gran calidad a pesar, o incluso como consecuencia del cambio continuo. Su principal asunción es que con un poco de planificación, un poco de codificación y unas pocas pruebas se puede decidir si se está siguiendo un camino acertado o equivocado, evitando así tener que echar marcha atrás demasiado tarde.

Valores que promueve XP:

- **Coraje.**
- **Comunicación:** XP pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma ve el avance día a día, y es posible ajustar la agenda y las funcionalidades de forma consecuyente.
- **Feedback rápido y continuo:** Una metodología basada en el desarrollo incremental de pequeñas partes, con entregas y pruebas frecuentes y continuas, proporciona un flujo de retro-información valioso para detectar los problemas o desviaciones. De esta forma los fallos se localizan muy pronto, pues se detectan los errores de planificación que solo se evidencian durante el desarrollo del sistema. La retro-información constituye la herramienta que permite reajustar la agenda y los planes.
- **Simplicidad:** La simplicidad consiste en desarrollar sólo el sistema que realmente se necesita e implica resolver en cada momento sólo las necesidades actuales. Con este principio de simplicidad, junto con la comunicación y el feedback resulta más fácil conocer las necesidades reales.

XP no es un modelo de procesos ni un marco de trabajo, sino un conjunto de 12 prácticas que se complementan unas a otras y deben implementarse en un entorno de desarrollo cuya cultura se base en los cuatro valores citados. (Letelier, et al., 2006)

SCRUM

No es propiamente un método o metodología de desarrollo, e implantarlo como tal resulta insuficiente.

Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas.

Los principios de Scrum son:

- Equipos autogestionados.
- Una vez dimensionadas las tareas no es posible agregarles trabajo extra.
- Reuniones diarias en las que los miembros del equipo se plantean 3 cuestiones:
 - ¿Qué has hecho desde la última revisión?
 - ¿Qué obstáculos te impiden cumplir la meta?
 - ¿Qué vas a hacer antes de la próxima reunión?

- Iteraciones de desarrollo de frecuencia inferior a un mes, al final de las cuales se presenta el resultado a los externos del equipo de desarrollo, y se realiza una planificación de la siguiente iteración, guiada por cliente. (Gómez, 2008)

DSDM (Dynamic Systems Development Method)

Es la metodología más antigua de las auto-denominadas ágiles y a su vez la más próxima a los métodos formales, de hecho la implantación de un modelo DSDM en una organización la lleva a alcanzar lo que CMM⁴⁷ consideraría un nivel 2 de madurez.

En común con los métodos ágiles, DSDM considera imprescindible una implicación y una relación estrecha con el cliente durante el desarrollo, así como la necesidad de trabajar con métodos de desarrollo incremental y entregas evolutivas.

DSDM cubre los aspectos de gestión de proyectos, desarrollo de los sistemas, soporte y mantenimiento y se autodefine como un marco de trabajo para desarrollo rápido más que como un método específico para el desarrollo de sistemas. (Miguel, 2009)

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 (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. 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). (Methodologies, 2008)

Metodologías Clásicas

RUP

El Proceso Unificado Rational (Rational Unified Process) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a

⁴⁷ Modelo de Capacidad y Madurez (Capability Maturity Model)

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. 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. (Valencia, 2007)

1.3.8. Herramientas CASE (Computer – Aided Software Engineering)

La consolidación de metodologías de desarrollo integrando diferentes técnicas impulsó la aparición de paquetes de propósito más amplio. Surgió la necesidad de un diccionario de datos del sistema que almacene las definiciones usadas en las diferentes fases del desarrollo (este diccionario es lo que comúnmente se denomina repositorio). Esto contribuyó a implementar funciones de integración y verificación de consistencia entre técnicas (asociadas a distintas actividades en el desarrollo). La automatización de tareas también ha sido un aspecto de interés. En programación automática esto se ha traducido en: generadores de pantallas e informes, generadores de esquemas físicos de bases de datos y generadores de código para prototipos o partes de programas.

Actualmente, en Ingeniería de Software todos los desafíos y los correspondientes enfoques de solución están normalmente concebidos y llevados a la práctica dentro del contexto de un CASE. (Valencia, 2001)

El propósito de una herramienta CASE es dar soporte automatizado para la aplicación de todas o algunas técnicas usadas por una o varias metodologías. El énfasis en planificación, análisis y diseño promovido por una herramienta CASE tiene un fuerte impacto y recompensa en la mejora de la calidad del producto obtenido y en el aumento de

productividad (disminución de tiempos, costes y esfuerzos) en las actividades de desarrollo y mantenimiento.

El beneficio adicional obtenido por la utilización de un CASE actual (si se compara con la utilización de una metodología sin el uso de un CASE) se representa en los siguientes aspectos:

- Facilita la verificación y mantenimiento de la consistencia de la información del proyecto.
- Facilita el establecimiento de estándares en el proceso de desarrollo y documentación.
- Facilita el mantenimiento del sistema y las actualizaciones de su documentación.
- Facilita la aplicación de las técnicas de una metodología.
- Disponibilidad de funciones automatizadas tales como: obtención de prototipos, generación de código, generación de pantallas e informes, generación de diseños físicos de bases de datos, verificadores automáticos de consistencia.
- Facilita la aplicación de técnicas de reutilización y reingeniería.
- Facilita la planificación y gestión del proyecto informático. (Chacón, 2007)

Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (Freedownloadmanager.org, 2004)

Además es una herramienta colaborativa, pues soporta múltiples usuarios trabajando sobre el mismo proyecto. La documentación del proyecto puede ser generada automáticamente en varios formatos (Web o .Pdf), y permite control de versiones.

Enterprise Architect

Enterprise Architect combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación.

Alta capacidad: Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multiusuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea.

Trazabilidad de extremo a extremo: Enterprise Architect provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

Velocidad, estabilidad y buen rendimiento: El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible.

Construido sobre las bases de UML 2.1: Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0. Usa Perfiles UML para extender el dominio de modelado, mientras que la Validación del Modelo asegura integridad. Combina Procesos de Negocio, Información y Flujos de trabajo en un modelo usando extensiones gratuitas para BPMN⁴⁸ y el perfil Eriksson-Penker. (Sparxsystems.com, 2006)

Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML [Booch, Rumbaugh y Jacobson] y que soporta de forma completa la especificación del UML, 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. Esta herramienta propone la utilización de cuatro tipos de modelos

⁴⁸ Notación para el Modelado de Procesos del Negocio (Business Process Management Notation)

para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (De Nobrega, 2005) A continuación se muestran algunas de las características que tiene Rational:

- **Desarrollo Iterativo:** Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.
- **Generador de Código:** Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.
- **Ingeniería Inversa:** Proporciona mecanismos para realizar la denominada Ingeniería Inversa, a partir del código de un programa, se puede obtener su diseño.
- **Trabajo en Grupo:** Permite varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. (Martínez Pérez, et al., 2006)

El navegador UML de Rational Rose 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.

1.3.9. Lenguaje Unificado de Modelado (UML)

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, no es un lenguaje de programación.

Las herramientas CASE pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a

partir de programas existentes. UML es un lenguaje de propósito general para el modelado orientado a objetos, es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes. (Corp, 2006)

Objetivos del UML

- Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso, incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial. (Vico.org, 2002)

1.4. Tecnologías y Herramientas a utilizar

Luego del estudio realizado se arriba a la conclusión de que el Sistema estará guiado por la metodología de desarrollo RUP, por ser una de las más utilizadas para el diseño, implementación y documentación de sistemas web; además esta basada en UML. Por las potencialidades ofrecidas se escoge la herramienta Rational Rose Enterprise Edition 2003 para llevar la documentación del software.

En la implementación del sistema, considerando la importancia de la integración con el Microsoft Word, se utilizará el Visual Studio .NET 2008, tecnología ASP.NET –pues se trata de una aplicación web- con lenguaje de programación C#. Se incluye la herramienta Visual Studio Tools for Office Second Edition, la cual facilita 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 páginas ASP.Net se ejecuten en dicho servidor web.

Como Sistema Gestor de Base de Datos se determina el uso de SQL Server 2005 Express Edition, por su rapidez cuando se trabaja con una base de datos pequeña, además de estar totalmente integrado con los componentes del Visual Studio. La aplicación se sustentará en una Arquitectura Cliente – Servidor de tres capas.

Como resultado del análisis realizado durante el presente capítulo se concluye que: las herramientas estudiadas relacionadas con el proceso de seguimiento de errores no responden a las necesidades que se presentan en el Proyecto Calidad de la Facultad 7. Debido a esto se necesita implementar un sistema informático para la revisión y seguimiento de errores. Para ello se realizó un estudio de las tecnologías actuales y se seleccionaron las más adecuadas para la implementación del sistema.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se tratan los aspectos fundamentales relacionados con el objeto de estudio. Se explican los procesos del negocio sobre los que se basa la organización para su funcionamiento, se describe el flujo actual de los procesos y se realiza un análisis crítico de cómo estos se realizan actualmente -incluyendo toda la información que se maneja, o sea documentos específicos que se procesan-. Además se presenta 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 señalar puntualmente los requerimientos del sistema que va a soportar la organización.

Se muestra la Especificación de los requisitos de software (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

En el Proceso de Aseguramiento y Control de la Calidad en los proyectos de la Facultad 7, es de vital importancia la detección, seguimiento y control de errores de la documentación generada en los mismos. A continuación se describen los principales elementos del proceso:

Éste proceso permite que los Grupos de Desarrollo de los diferentes polos productivos existentes en la Facultad 7 (Polo de Imágenes, Polo de Salud), puedan certificar la calidad de sus productos. Dicho proceso comienza cuando el cliente (Líder de un Proyecto o el Líder del Equipo de Calidad Interno) desea liberar un artefacto, para lo cual se envía la solicitud y de ser aceptada por el Asesor de Calidad de la Facultad se entrega la documentación del producto a revisar al Proyecto Calidad.

En caso de haber aceptado la solicitud, la documentación es recibida por el Asesor de Calidad de la Facultad o por el Asesor de Calidad que atiende el Área Temática, a partir de entonces comienza el proceso de pruebas a la documentación entregada. El Probador revisa la documentación y genera el RNC (en caso de existir demasiadas NC el Jefe de Pruebas en ese momento puede abortar el proceso de pruebas al artefacto en cuestión).

Posteriormente los RNC generados se entregan al Jefe de Pruebas, el que se encarga de unificarlos en caso de que sea necesario y revisar que estén correctamente elaborados, estos registros se incluyen en el Expediente de Prueba y se le notifica al cliente, vía correo electrónico, que ya puede acceder a los mismos.

El cliente define la respuesta a las NC (si proceden o no) y justifica en caso negativo, corrige las NC que proceden y envía la respuesta a las NC; estas son incluidas en el expediente de pruebas y notificadas al Jefe de Pruebas. Se revisa la respuesta a las NC, se actualiza el RNC verificando las soluciones proporcionadas por el Equipo de Desarrollo y se da paso a una nueva iteración.

El proceso de pruebas termina una vez liberados todos los artefactos solicitados, para ello existen tres tipos de liberación:

- **Liberación total:** Se define cuando son liberados satisfactoriamente todos los elementos probados.
- **Liberación parcial:** Se define cuando se interrumpen las pruebas de alguno de los elementos solicitados y los demás concluyen satisfactoriamente la etapa de pruebas, liberándose estos últimos y pendientes los rechazados.
- **Liberación bajo concesión:** Se define cuando es necesario liberar los elementos probados pero estos, una vez concluida la última iteración, aún presentan NC no significativas, es decir, que no afectan el funcionamiento del sistema.

Causas:

- Surgimiento de NC con posterioridad a la aprobación por parte del cliente de un determinado elemento, quedando pendiente la revisión y/o aceptación por el mismo.
- Presentar NC no significativas de cuya solución se hace responsable la dirección del proyecto.

2.1.2. Análisis crítico de la ejecución de los procesos

Las Pruebas a la Documentación realizadas hasta el momento en el Proyecto Calidad se ven afectadas por una serie de aspectos que entorpecen la obtención de buenos resultados. Entre estos problemas se encuentra la necesidad del Probador de interactuar con varios documentos y aplicaciones (Word, Paint) de forma simultánea, pues no se tiene

la posibilidad de un entorno unificado de trabajo. El Probador necesita desplazarse por el documento a revisar y además ir generando un Registro de No Conformidades; el constante cambio entre ambos documentos limita las posibilidades de detección de errores y la especificidad en la redacción de los ya encontrados. Además, al insertar los anexos, editados en otra aplicación (Ejemplo: Paint), se favorece la inserción de errores, siendo los más comunes el enumerado de los anexos y su correcta correspondencia con la No Conformidad detectada.

Como otro elemento determinante se tiene que todos los datos del Registro de No Conformidades son llenados de forma manual, alargando el periodo de pruebas, favoreciendo omisiones y entradas de datos incorrectos, por lo que repercute de forma notoria en la buena calidad del proceso.

A estos problemas se suma la necesidad de unificar los Registros de No Conformidades que genera cada Probador en un Registro de No Conformidades General, ya que los desarrolladores necesitan de este último para realizar las transformaciones necesarias en el software sometido a revisión. Este traspaso de información también provoca errores, los cuales se evidencian en ocasiones en la omisión de No Conformidades.

En el Proyecto Calidad de la Facultad 7 se hace especial énfasis en el logro de un entendimiento con los desarrolladores, quienes en su mayoría no están familiarizados con el trabajo realizado por los Probadores. Estas dificultades se hacen evidentes cuando se necesita dar seguimiento a ciertos errores detectados, pues a los desarrolladores se les hace difícil en ocasiones entender lo descrito por el Probador en el RNC.

Existen herramientas en el Proyecto Calidad de la Facultad 7 que ayudan a ganar en cuanto a organización y centralización del trabajo (fundamentalmente Subversion), pero se carecen de mecanismos para darle solución a las problemáticas presentadas.

2.1.3. Información que se maneja

El Registro de No Conformidades: es un documento para el control y seguimiento de las No Conformidades. Posee nombre del proyecto revisado, número de revisión, fecha del documento y nombre del probador, lo cual favorece el control de posibles errores internos en el proceso. Este documento está formado fundamentalmente por una lista de todas las No Conformidades encontradas durante el Proceso de Pruebas y la mayor cantidad de información posible para la fácil identificación de las mismas, así como las circunstancias en las que fueron generadas.

La documentación del producto de software: incluye Manual de Usuario, Modelo de Casos de Uso del Negocio, Modelo de Casos de Uso del Sistema, Ayuda del software, Especificación de Requisitos, entre otros (estos elementos pueden variar en dependencia del producto de software en revisión). Todos estos elementos constituyen el punto de partida del Proceso de Revisión de la Documentación a la que se somete el producto.

2.1.4. Objeto de automatización

El sistema propuesto ha sido concebido con el fin de mejorar el Proceso de Revisión de la Documentación que se desarrolla actualmente en el Laboratorio del Proyecto Calidad de la Facultad 7 a los distintos productos informáticos.

El sistema automatizará el Registro de No Conformidades y facilitará la comunicación entre Probadores y Desarrolladores durante el Proceso de Pruebas, pues ambos trabajarán sobre la aplicación actualizando de manera inmediata los datos generados.

La aplicación posibilitará que los Probadores presten una mejor atención al trabajo que están realizando, pues no dependerán de herramientas auxiliares para la obtención de los resultados de su trabajo. Al mismo tiempo los datos estarán disponibles para los Jefes de Prueba, el Asesor de Calidad de la Facultad y el Administrador de Calidad, lo que posibilitará la obtención de un informe sobre los resultados de las pruebas en cualquier momento.

Los Desarrolladores también se verán beneficiados, pues la herramienta les permitirá visualizar los errores detectados y la realización de las correcciones pertinentes en un mismo ambiente de trabajo, facilitando el seguimiento de los errores y su localización, independientemente de la existencia de los errores que puedan ser insertados durante la detección de los mismos.

2.2. Modelo de Negocio

2.2.1. 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	Personal externo al negocio que se beneficia e interactúa con este. Comprende a los interesados en la revisión de la documentación generada por el producto informático desarrollado, con el fin de que esta sea certificada.

Tabla 2.1 -Descripción de los actores del negocio.

2.2.2. 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
Jefe de Pruebas	Encargado de la coordinación, establecimiento y control de las pautas para la realización del Proceso de Pruebas. Realiza la distribución del trabajo y, según los resultados obtenidos, evalúa el tipo de liberación que se aplicará al producto.
Administrador de Calidad	Es el encargado de socializar la información desde y hacia el Grupo de Pruebas, actualizando los documentos revisados y corregidos.
Probador	Encargado del proceso de detección, registro y control de las No Conformidades detectadas en el

	Proceso de Pruebas a la Documentación. (Miembros del Grupo de Calidad)
Desarrollador	Encargado del proceso de corrección de las No Conformidades detectadas por el Probador y responsable de la actualización de la documentación que se está revisando. (Miembros del Equipo de Desarrollo)

Tabla 2.2 -Descripción de los trabajadores del negocio.

2.2.3. Diagrama de Casos de Uso del Negocio

Un Caso de Uso del Negocio representa un proceso que se ejecuta en el mismo, constituyendo una serie de acciones que muestran un resultado palpable para ciertos actores de dicho negocio.

Para el presente trabajo se define como un proceso que muestra resultados observables el Proceso de Pruebas a la Documentación. Este muestra como resultado más significativo el Registro de No Conformidades.

El diagrama de Casos de Uso del Negocio que se presenta a continuación representa gráficamente la interacción existente entre el proceso del negocio (Revisar Documentación) y el actor del negocio (Cliente).

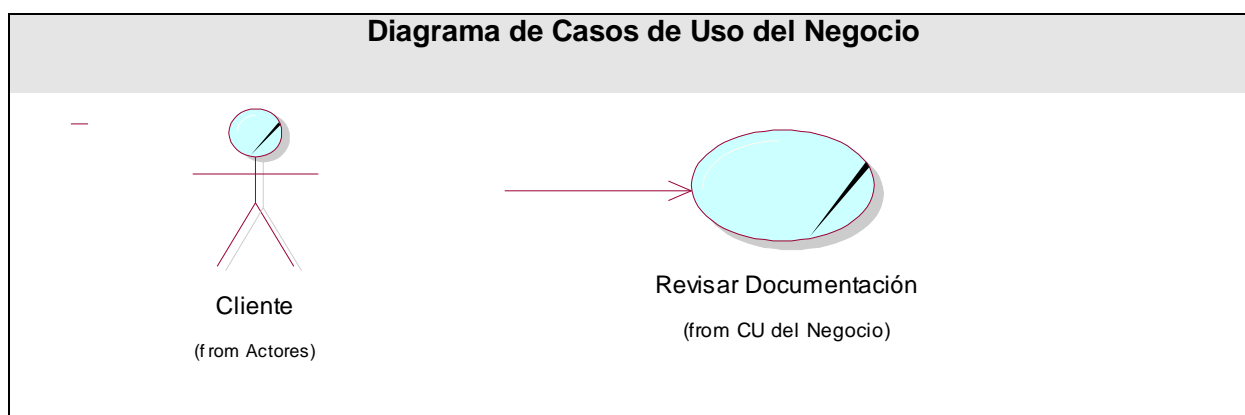


Fig. 2.1 -Diagrama de Casos de Uso del Negocio.

2.2.4. Descripción Textual de los Casos de Uso del Negocio

Caso de Uso:	Revisar Documentación
Actores:	Cliente (inicia)
Trabajadores:	Jefe de Pruebas, Administrador de Calidad, Probador, Desarrollador
Resumen: El caso de uso se inicia cuando el cliente le entrega la documentación de su producto al Jefe de Pruebas 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 la documentación no tenga errores.	
Precondiciones:	La solicitud de revisión debe haber sido aceptada por el Equipo de Pruebas.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Entrega la Documentación	2. El Jefe de Pruebas recibe la documentación y orienta el trabajo. 3. El Probador revisa la documentación. 4. El Probador registra las No Conformidades detectadas en el Registro de No Conformidades (RNC). 5. El Probador, cuando termina la revisión, envía al Jefe de Pruebas el RNC. 6. El Jefe de Pruebas revisa e integra los RNC enviados por los Probadores en un Registro de No Conformidades General (RNCG). 7. El Jefe de Pruebas le envía al Cliente el RNCG.
8. El Cliente revisa el RNCG enviado por el Jefe de Pruebas.	13. El Jefe de Pruebas recibe la documentación corregida y el RNCG actualizado, orienta el trabajo.
9. Actualiza el RNCG con el estado de	

Capítulo 2: Características del Sistema

<p>aceptación de las No Conformidades (Aceptada-rechazada).</p> <p>9.1. Si se rechaza alguna No Conformidad se describen las causas.</p> <p>10. Corrige la documentación acorde a los señalamientos</p> <p>11. Actualiza el RNCG con el estado de solución (Resuelta-Pendiente) de las No Conformidades.</p> <p>12. El Cliente envía RNCG actualizado y la documentación corregida.</p>	<p>13.1. El Jefe de Pruebas decide realizar una nueva iteración. (Ver Flujo Alterno 1).</p> <p>13.2. El Jefe de Pruebas decide liberar el Producto (Ver Flujo Alterno 2).</p>
Flujos Alternos	
Flujo Alterno 1	
Acción del Actor	Respuesta del Negocio
	<p>14. El Jefe de Pruebas orienta el trabajo.</p> <p>15. El Probador revisa la documentación para verificar la corrección de las No Conformidades detectadas y la existencia de nuevas No Conformidades.</p> <p>16. El Probador actualiza el RNC con el estado de solución de las No Conformidades detectadas y la descripción de las nuevas No Conformidades detectadas.</p> <p>17. Ver paso 5 del Flujo Normal de Eventos.</p>
Flujos Alternos	
Flujo Alterno 2	

Acción del Actor	Respuesta del Negocio
	<p>14. Acorde a los resultados obtenidos en el Proceso de Pruebas a la Documentación el Jefe de Pruebas selecciona el tipo de liberación.</p> <p>15. El Jefe de Pruebas emite el acta de Liberación Parcial con las especificaciones según el tipo de liberación</p>

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

2.2.5. Diagrama 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)

2.2.6. 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)

2.3. Especificación de los Requisitos del Software

2.3.1. Requisitos Funcionales

1. Realizar autenticación
2. Gestionar Usuarios
 - 2.1. Crear usuario
 - 2.2. Modificar usuario
 - 2.3. Eliminar usuario
 - 2.4. Mostrar detalles de un usuario
3. Gestionar Elementos del Expediente de Proyecto

- 3.1. Crear Elemento del Expediente de Proyecto
- 3.2. Modificar Elemento del Expediente de Proyecto
- 3.3. Eliminar Elemento del Expediente de Proyecto
- 3.4. Mostrar detalles del Elemento del Expediente de Proyecto
- 4. Gestionar Revisiones
 - 4.1. Crear revisión
 - 4.2. Terminar revisión
 - 4.3. Mostrar detalles de una revisión
- 5. Subir Documentación
- 6. Gestionar Tareas
 - 6.1. Asignar tarea
 - 6.2. Eliminar asignación de tarea
 - 6.3. Mostrar detalles de una tarea
 - 6.4. Listar tareas
- 7. Gestionar No Conformidades
 - 7.1. Crear No Conformidad
 - 7.2. Modificar una No Conformidad
 - 7.3. Eliminar una No Conformidad
 - 7.4. Mostrar detalles de una No Conformidad
 - 7.5. Asignar Respuesta del Equipo de Desarrollo
 - 7.6. Corregir No Conformidad
- 8. Gestionar Reportes
- 9. Listar Usuarios

10. Listar Revisiones

11. Listar No Conformidades

2.3.2. Requisitos No Funcionales

Interfaz Externa:

La interfaz será de diseño sencillo, optimizando las opciones de navegabilidad entre las funcionalidades. Los colores a usar estarán acordes a los fines de la aplicación.

Usabilidad:

El sistema está orientado a personas con conocimientos básicos de computación. Para una correcta explotación de las facilidades que brinda se recomienda principalmente para personas medianamente capacitadas en el Proceso de Pruebas a la Documentación.

Rendimiento:

Las páginas estarán diseñadas persiguiendo el objetivo de optimizar el rendimiento del servidor, evitando la inclusión de elementos gráficos innecesarios. Los tiempos para la ejecución de las funcionalidades no excederán los 5 segundos.

Soporte:

Se tendrá un sistema de ayuda insertado en la aplicación. La documentación del producto abarcará todos los aspectos necesarios para el mantenimiento y la ejecución de mejoras en caso de ser necesario.

Portabilidad:

El hecho de ser una herramienta Web hace que el sistema pueda ejecutarse en múltiples Sistemas Operativos (Windows, Linux, etc.).

Seguridad:

La seguridad del sistema estará dada por un sistema de autenticación de usuarios, cuyos permisos están definidos según el rol que desempeñe el usuario que se autentica.

Software (Cliente)

Navegador Mozilla Firefox 2.0 o superior, Internet Explorer 6.0 o superior

Sistemas Operativo Windows 98 o superior, Linux

Microsoft .Net Framework 3.5 (SO Windows) Mono Framework 3.5 (Linux)

Software (Servidor)

Sistema Operativo Linux o Windows Advanced Server 2000 o superior

Servidor de aplicaciones web Apache 1.13.19, con módulo mod_mono

Servidor de Base de Datos Microsoft SQL Server Express 2005

Requerimientos mínimos de Hardware (Cliente):

Procesador Pentium III a 1GHz de velocidad de procesamiento y 128 Mb de memoria RAM

Monitor VGA o superior

Tarjeta de red

Requerimientos mínimos de Hardware (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

2.4. Definición de los Casos de Uso

2.4.1. Definición de los actores

Actores	Justificación
Administrador de Pruebas	Encargado del manejo y configuración de las acciones que se desarrollan en la aplicación. Realiza la gestión de usuarios que intervienen en los procesos de la aplicación. Responsable por el Grupo de Calidad de llevar el control de las Revisiones y los Probadores que forman parte de ellas. Forma parte del flujo de información que se establece con el Equipo de Desarrollo del producto en revisión.
Usuario Avanzado	Realiza las acciones comunes definidas para el Jefe de Pruebas y el Administrador de Calidad.

Capítulo 2: Características del Sistema

Usuario	Realiza las acciones comunes definidas para todos los que interactúan con la aplicación.
Jefe de Pruebas	Controla las No Conformidades generadas en el proceso.
Administrador de Calidad	Representa al Equipo de Desarrollo designado para dar respuestas y soluciones a las acciones realizadas por el Equipo de Pruebas. Es el responsable, por parte de los Desarrolladores, de llevar el control del equipo que interviene en la revisión correspondiente a su producto de software. Controla las No Conformidades generadas en el proceso.
Probador	Encargado de definir las No Conformidades detectadas a un producto en revisión. También brinda seguimiento a las respuestas y acciones definidas por el Equipo de Desarrollo.
Desarrollador	Encargado de atender las No Conformidades detectadas al producto en revisión, brindando respuestas y acciones concretas. Responsable del seguimiento de las No Conformidades hasta su erradicación o hasta la terminación de las pruebas.

Tabla 2.4 -.Descripción de los actores del sistema.

2.4.2. Listado de los Casos de Uso

CU Autenticar

CU #1	CU Autenticar
Usuario	Usuario
Descripción	Comienza cuando un usuario decide empezar a interactuar con la aplicación; ingresa su identificador y contraseña. El

Capítulo 2: Características del Sistema

	sistema determina si la autenticación se realizó o no con éxito.
Referencia	RF 1

Tabla 2.5 -.CU Autenticación.

CU Gestionar Usuarios

CU #2	CU Gestionar Usuarios
Usuario	Administrador de Pruebas
Descripción	Comienza cuando el Administrador de Pruebas decide realizar una de las acciones para efectuar la gestión de los usuarios que interactúan con el sistema; puede ser para agregar, eliminar y modificar un usuario. Permite además visualizar los detalles de un usuario seleccionado.
Referencia	RF 2

Tabla 2.6 -.CU Gestionar Usuario.

CU Gestionar Elementos del Expediente de Proyecto

CU #3	CU Gestionar Elementos del Expediente de Proyecto
Usuario	Administrador de Pruebas
Descripción	Se inicia cuando el Administrador de Pruebas decide realizar acciones como agregar, modificar o eliminar un elemento del Expediente de Proyecto. Permite además visualizar los detalles de un elemento seleccionado.
Referencia	RF 3

Tabla 2.7 -.CU Gestionar Elementos del Expediente de Proyecto.

CU Gestionar Revisiones

CU #4	CU Gestionar Revisiones
Usuario	Administrador de Pruebas
Descripción	Se inicia cuando el Administrador de Pruebas decide realizar acciones como agregar, modificar o eliminar una revisión. Permite además visualizar los detalles de una revisión seleccionada.
Referencia	RF 4

Tabla 2.8 -.CU Gestionar Revisión.

CU Subir Documentación

CU #5	CU Subir Documentación
Usuario	Administrador de Calidad
Descripción	Se inicializa cuando el Administrador de Calidad determina poner a disposición del Equipo de Pruebas la documentación que será sometida a revisión.
Referencia	RF 5

Tabla 2.9 -.CU Subir Documentación.

CU Gestionar Tareas

CU #6	CU Gestionar Tareas
Usuario	Usuario Avanzado
Descripción	Se inicializa cuando un Usuario Avanzado (Jefe de Pruebas, Administrador de Calidad) decide realizar acciones como agregar y/o eliminar una tarea (asignarle a

Capítulo 2: Características del Sistema

	un Probador o Desarrollador el trabajo en una revisión). Permite además visualizar todas las tareas que han sido asignadas.
Referencia	Rf 4.3, RF 6, Rf 9, RF 10

Tabla 2.10 -.CU Gestionar Tareas.

CU Gestionar No Conformidades

CU #7	CU Gestionar No Conformidades
Usuario	Probador
Descripción	Se inicializa cuando un Probador decide realizar acciones para agregar, eliminar y/o modificar una No Conformidad. Permite además mostrar los detalles de una No Conformidad seleccionada.
Referencia	RF 6.3, FR 7.1, FR 7.2, RF 7.3, RF 7.4, RF 11

Tabla 2.11 -.CU Gestionar No Conformidades.

CU Responder No Conformidades

CU #8	CU Responder No Conformidades
Usuario	Desarrollador
Descripción	Se inicializa cuando un Desarrollador decide dar respuesta a las No Conformidades emitidas por el Equipo de Pruebas. Permite además mostrar los detalles de una No Conformidad seleccionada.
Referencia	RF 6.3, RF 7.4, RF 7.5, RF 7.6, FR 11

Tabla 2.12 -.CU Responder No Conformidades.

CU Gestionar Reportes

CU #9	CU Gestionar Reportes
Usuario	Usuario Avanzado
Descripción	Se inicializa cuando se desea obtener un reporte del proceso de pruebas realizado a la documentación.
Referencia	RF 8, RF 9, RF 10, RF 11

Tabla 2.13 -.CU Gestionar Reportes

2.4.3. Diagrama de los Casos de Uso del Sistema

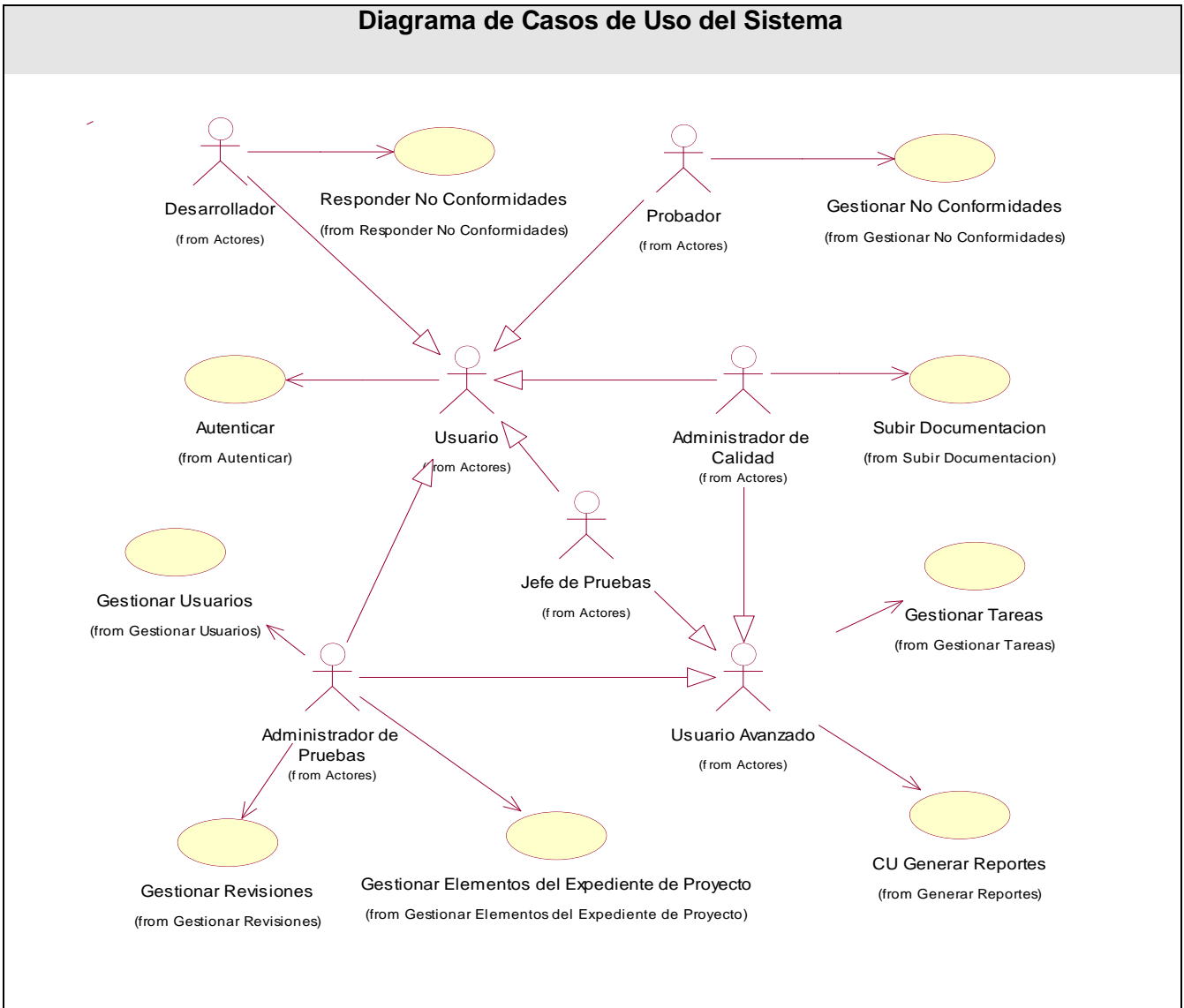


Fig. 2.3 -Diagrama de Casos de Uso del Sistema.

2.4.4. Listado de los Casos de Uso expandidos

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

CU Gestionar No Conformidades

Caso de uso	
CU#7	CU Gestionar No Conformidades

Capítulo 2: Características del Sistema

Propósito	Permitir realizar la gestión de las No Conformidades de una revisión.	
Actores: Probador		
Resumen: El Caso de Uso se inicializa cuando el Probador desea realizar acciones como agregar, modificar o eliminar una No Conformidad, puede además ver los detalles de una previamente seleccionada. El Caso de Uso termina cuando se termina la gestión de los elementos.		
Referencias	RF 6.3, FR 7.1, FR 7.2, RF 7.3, RF 7.4, RF 11	
Acción del actor	Respuesta del sistema	
1. Escoge la opción “Gestionar No Conformidades”.	2. Muestra la vista “Gestionar No Conformidades” 3. Muestra los detalles de la tarea que tiene asignada. 4. Se muestra el listado de las No Conformidades detectadas al elemento correspondiente a la tarea asignada.	
5. Selecciona la Etapa de Detección (etapa en la que se realiza la revisión del elemento). 6. Escoge realizar una de las acciones: a) Detectar Nuevas No Conformidades (Ver Sección “Nuevas No Conformidades”) b) Revisar No Conformidades Detectadas (Ver Sección “Revisar No Conformidades”)	7. Abre la aplicación para interactuar con el documento dentro del editor.	
Sección “Nuevas No Conformidades”		
1. Detecta una No Conformidad y se escoge la opción “Registrar”.	2. Se muestran los campos para recoger los datos: <ul style="list-style-type: none"> ○ Descripción ○ Clasificación ○ Epígrafe ○ Sub-epígrafe 	

Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> ○ Párrafo
3. Entra los datos solicitados y selecciona "Aceptar".	<p>3.1. El sistema verifica que los datos estén completos y sean correctos.</p> <p>3.2. Los datos son registrados y se muestra un mensaje informando del éxito de la operación.</p> <p>3.3. Los campos de entrada de datos se ocultan.</p>
Flujo Alternativo 1	
	<p>3.1. El sistema muestra un error en los datos suministrados.</p> <p>3.2. Permite volver a realizar la operación.</p>
Flujo Alternativo 2	
3. Entra los datos solicitados y selecciona "Cancelar".	<p>3.1. Los campos borran los datos suministrados.</p> <p>3.2. Los campos de entrada de datos se ocultan.</p> <p>3.3. La No Conformidad no es registrada.</p>
Sección "Revisar No Conformidades"	
1. Verifica si las No Conformidades fueron corregidas o no.	2. Se muestran los detalles de la No Conformidad seleccionada con los campos listos para realizar la modificación.
3. Actualiza la No Conformidad. Y escoge la opción "Aceptar".	<p>3.1. El sistema verifica que los datos estén completos y sean correctos.</p> <p>3.2. Los datos son registrados y se muestra un mensaje informando del éxito de la operación.</p> <p>3.3. Los campos de entrada de datos se ocultan.</p>
Flujo Alternativo 1	
2. Detecta una nueva No Conformidad (Ver Sección "Nuevas No Conformidades")	

Flujo Alternativo 2	
	<p>3.1. El sistema muestra un error en los datos suministrados.</p> <p>3.2. Permite volver a realizar la operación.</p>
Flujo Alternativo 2	
<p>3. Actualiza la No Conformidad. Y escoge la opción "Cancelar".</p>	<p>3.1. Los campos borran los datos suministrados.</p> <p>3.2. Los campos de entrada de datos se ocultan.</p> <p>3.3. La No Conformidad no es modificada.</p>

Tabla 2.14 -.CU Gestionar No Conformidades

CU Responder No Conformidades

Caso de uso	
CU#8	CU Responder No Conformidades
Propósito	Permitir que la actualización de las No Conformidades con la respuesta emitida por el Equipo de Desarrollo.
Actores: Desarrollador	
Resumen: El Caso de Uso se inicializa cuando el Desarrollador desea responder las No Conformidades detectadas a un elemento, muestra los detalles de una No Conformidad seleccionada. El Caso de Uso termina cuando la No Conformidad es modificada acorde a la respuesta emitida por el Equipo de Desarrollo.	
Referencias	RF 6.3, RF 7.4, RF 7.5, RF 7.6, FR 11
Acción del actor	Respuesta del sistema
1. Escoge la opción "Responder No Conformidades".	<p>2. Muestra la vista "Responder No Conformidades"</p> <p>3. Muestra los detalles de la tarea que tiene asignada.</p>

Capítulo 2: Características del Sistema

	4. Se muestra el listado de las No Conformidades del elemento asociada a la tarea.
5. Selecciona una No Conformidad. 6. Escoge una de las opciones: a) Brindar Respuesta (Ver Sección “Responder No Conformidades”) b) Corregir No Conformidad (Ver Sección “Corregir No Conformidades”)	7. Abre la aplicación para interactuar con el documento dentro del editor.
Sección “Responder No Conformidades”	
	1. Se muestran los detalles de la No Conformidad, solo aparece como editable el campo: <ul style="list-style-type: none"> ○ Estado de aceptación.
2. Entra los datos solicitados y selecciona “Aceptar”.	3.1. El sistema verifica que los datos estén completos y sean correctos. 3.2. Los datos son registrados y se muestra un mensaje informando del éxito de la operación. 3.3. Los campos de entrada de datos se ocultan. 3.4. El sistema muestra un mensaje para realizar la corrección de la No Conformidad.
3. En el mensaje escoge la opción “Aceptar”.	3.1. Se habilitan los controles de edición (Ver Sección “Corregir No Conformidades”).
Flujo Alternativo 1	
	3.1. El sistema muestra un error en los datos suministrados. 3.2. Permite volver a realizar la operación.
Flujo Alternativo 2	

Capítulo 2: Características del Sistema

4. Entra los datos solicitados y selecciona “Cancelar”.	4.1. Los campos borran los datos suministrados. 4.2. Los campos de entrada de datos se ocultan. 4.3. La No Conformidad no es modificada.
Flujo Alternativo 3	
5. En el mensaje escoge la opción “Cancelar”.	5.1. Los campos borran los datos suministrados. 5.2. Los campos de entrada de datos se ocultan. 5.3. La No Conformidad no es modificada.
Sección “Corregir No Conformidades”	
	1. Se muestran los detalles de la No Conformidad seleccionada con el campo listo para realizar la modificación: <ul style="list-style-type: none"> ○ estado de solución.
2. La No Conformidad es corregida en el elemento y se oprime la opción “Salvar” del editor. 3. Entra el dato solicitado y escoge la opción “Aceptar”.	4.1. El sistema verifica que los datos estén completos y sean correctos. 4.2. Los datos son registrados y se muestra un mensaje informando del éxito de la operación. 4.3. Los campos de entrada de datos se ocultan.
Flujo Alternativo 1	
	4.1. El sistema muestra un error en los datos suministrados. 4.2. Permite volver a realizar la operación.
Flujo Alternativo 2	
4. Entra el dato solicitado y escoge la opción “Cancelar”.	4.1. Los campos borran los datos suministrados. 4.2. Los campos de entrada de datos se ocultan. 4.3. La No Conformidad no es modificada.

Tabla 2.14 -.CU Responder No Conformidades

Capítulo 2: Características del Sistema

En este capítulo se realizó un análisis crítico de los procesos actuales del negocio. Este análisis permitió precisar el estado actual del Proceso de Pruebas a la Documentación en los proyectos productivos de la Facultad 7; con respecto a las actividades para asegurar la calidad del mismo, así como de los productos desarrollados. De la información obtenida, se definieron las necesidades de funcionamiento de la aplicación a implementar.

CAPÍTULO 3: ANÁLISIS Y DISEÑO

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 abordarán 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. Diagrama 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 sus funcionalidades. 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 identificados como principales (CU Gestionar No Conformidades, CU Responder No Conformidades). El resto de los Diagramas de clases se podrán encontrar en el Anexo 5.

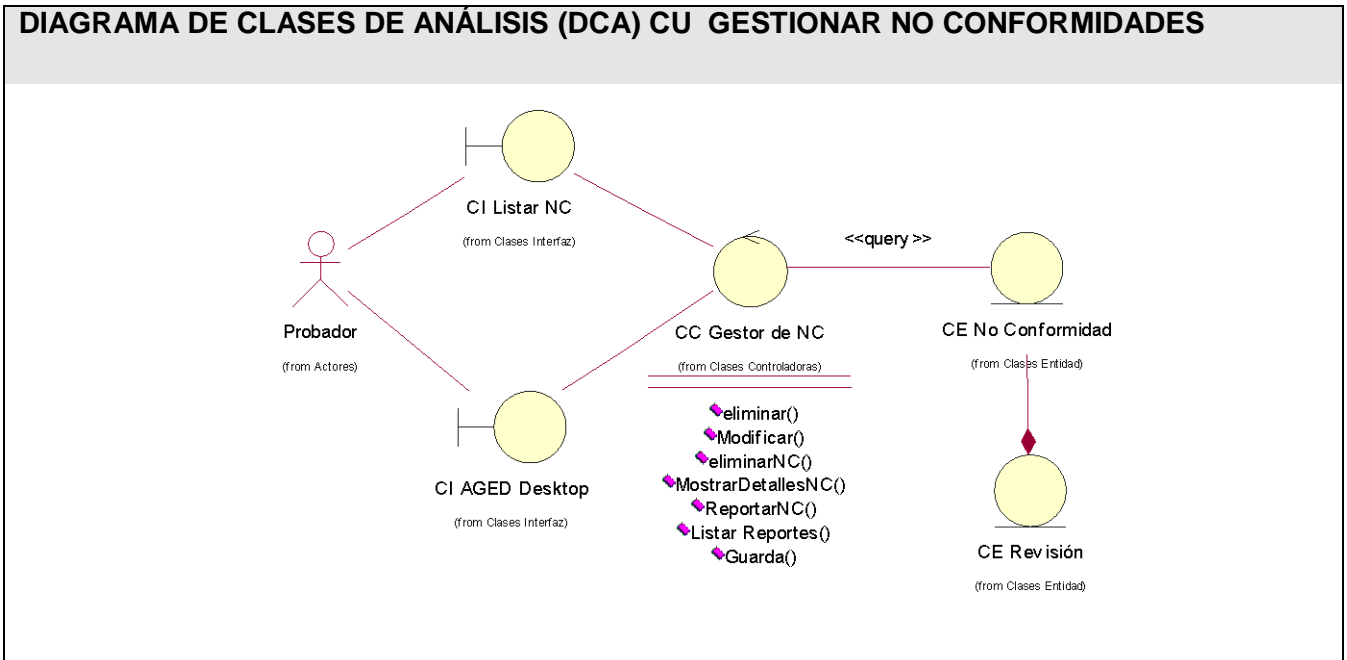


Fig. 3.1 –DCA CU Gestionar No Conformidades.

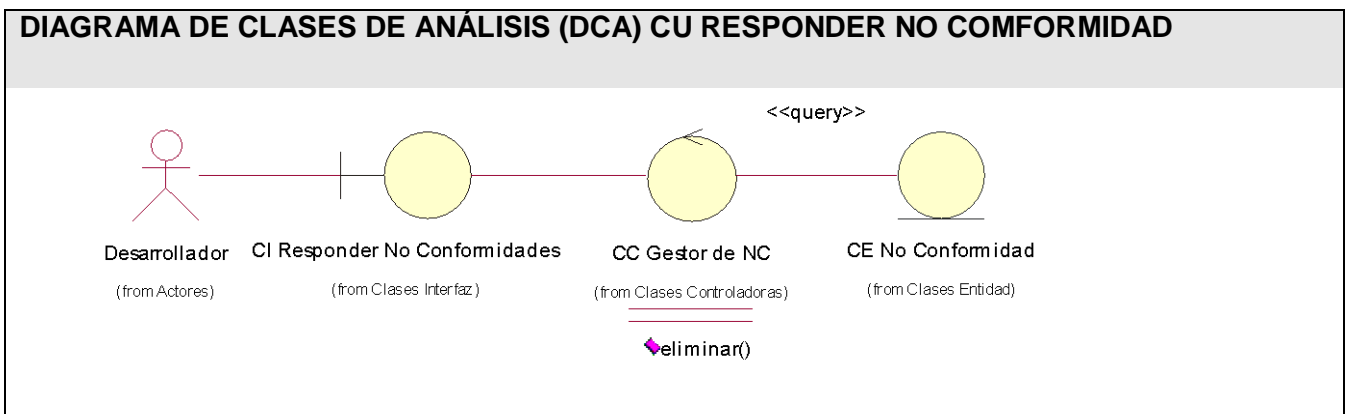


Fig. 3.2 –DCA CU Responder No Conformidades.

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. Los patrones GRASP⁴⁹ 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.

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

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.

⁴⁹ Patrones Generales de Software para Asignación de Responsabilidades (General Responsibility Assignment Software Patterns)

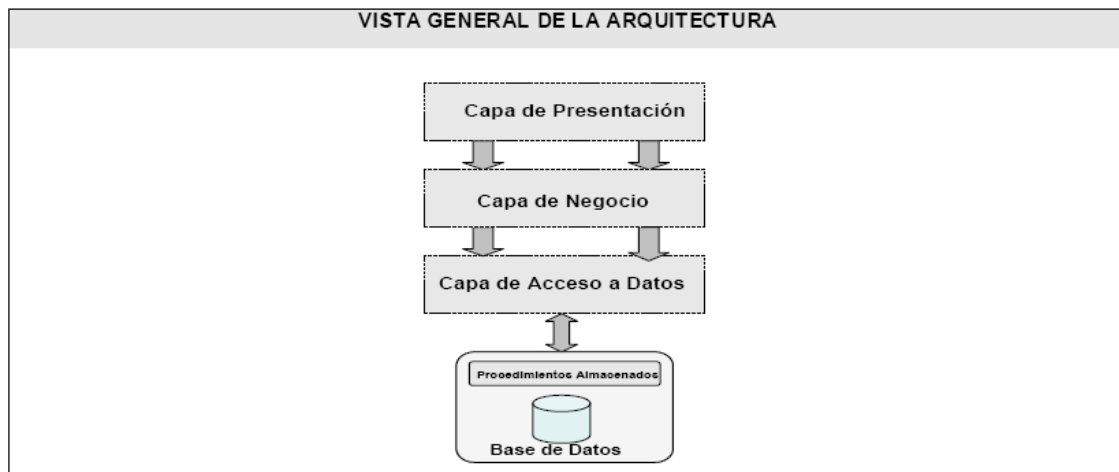


Fig. 3.3 –Vista General de la Arquitectura.

3.2.3. Diagrama de Clases de 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 y Responder No Conformidades. El resto de los diagramas de clases del diseño se podrán encontrar en el Anexo 6.

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GESTIONAR NO CONFORMIDADES

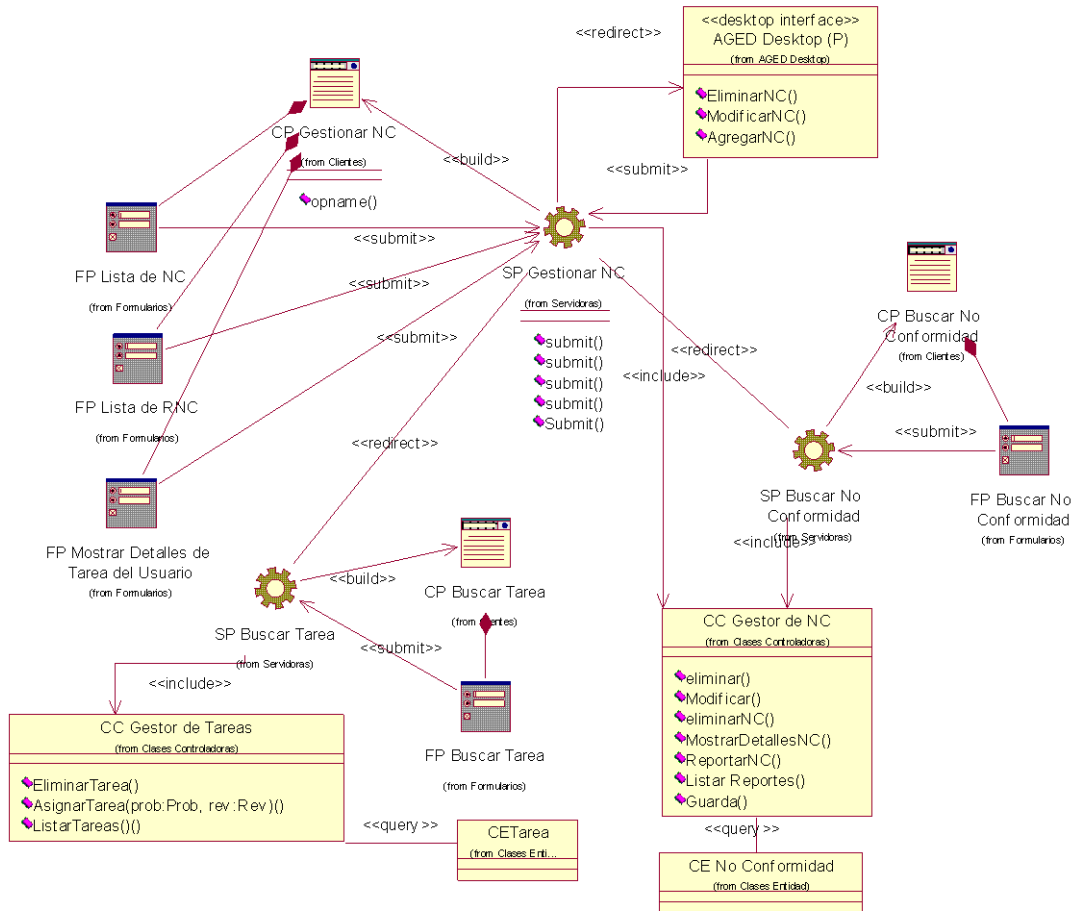


Fig. 3.4 –DCD CU Gestionar No Conformidades.

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU RESPONDER NO CONFORMIDAD

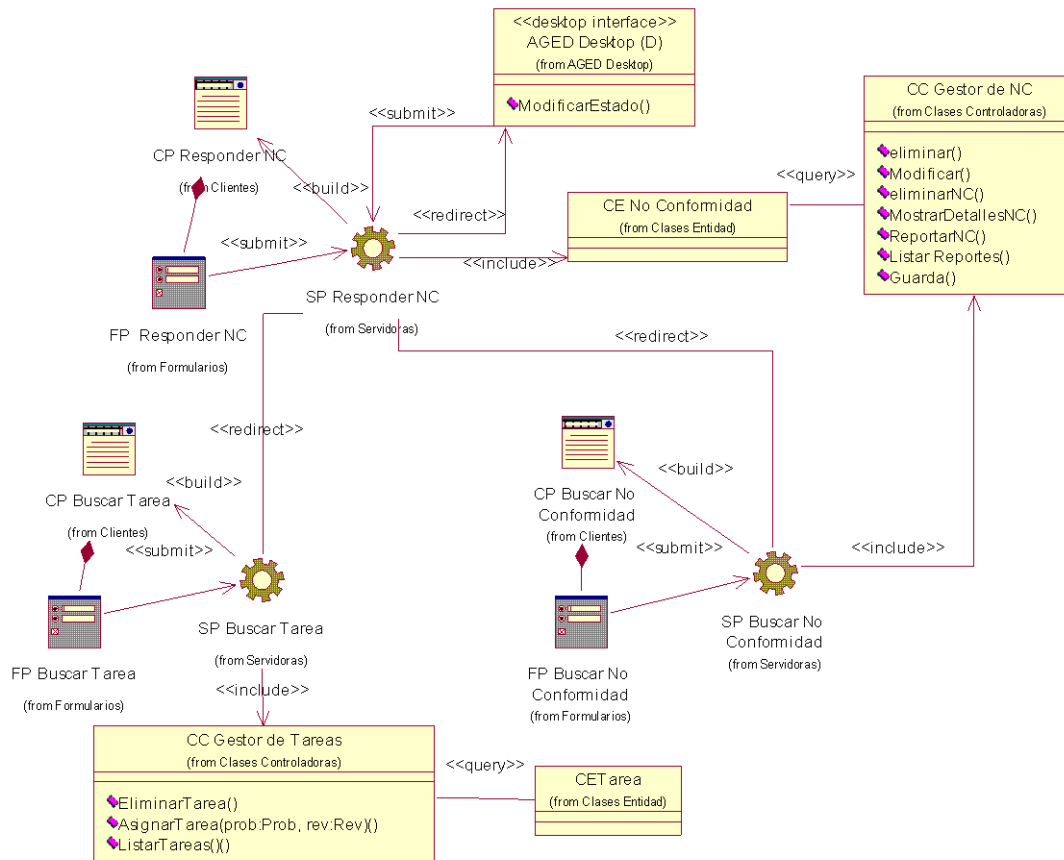


Fig. 3.5 –DCD CU Responder No Conformidades.

3.2.4. Diagramas de Interacción (Secuencia)

Se presentarán los diagramas de interacción de los Caso de Uso Gestionar No Conformidades y Responder No Conformidades, dividido por escenarios, para lograr una mejor comprensión. El resto de los diagramas de interacción se podrán encontrar en el Anexo 7.

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR NO CONFORMIDADES (Gestionar NC)

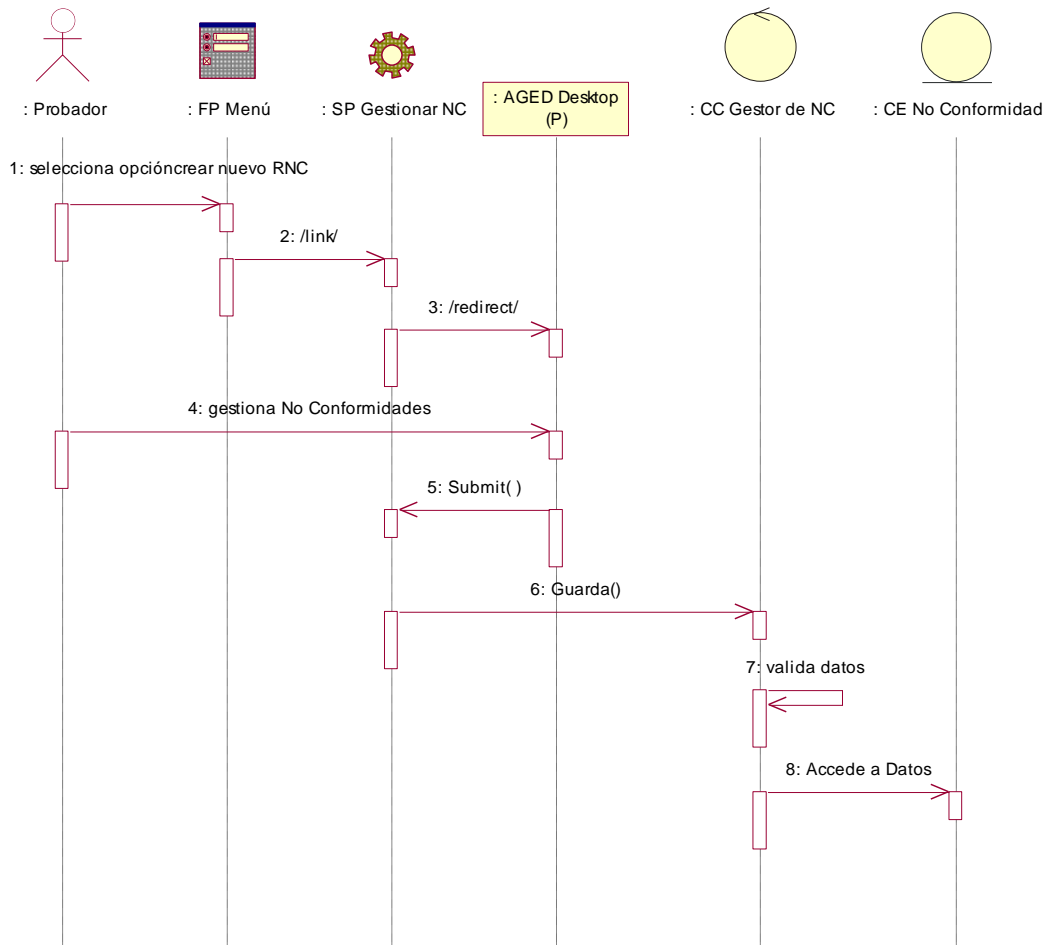


Fig. 3.6 –DS CU Gestionar No Conformidades (Gestionar NC).

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR NO CONFORMIDADES (Lista NC)

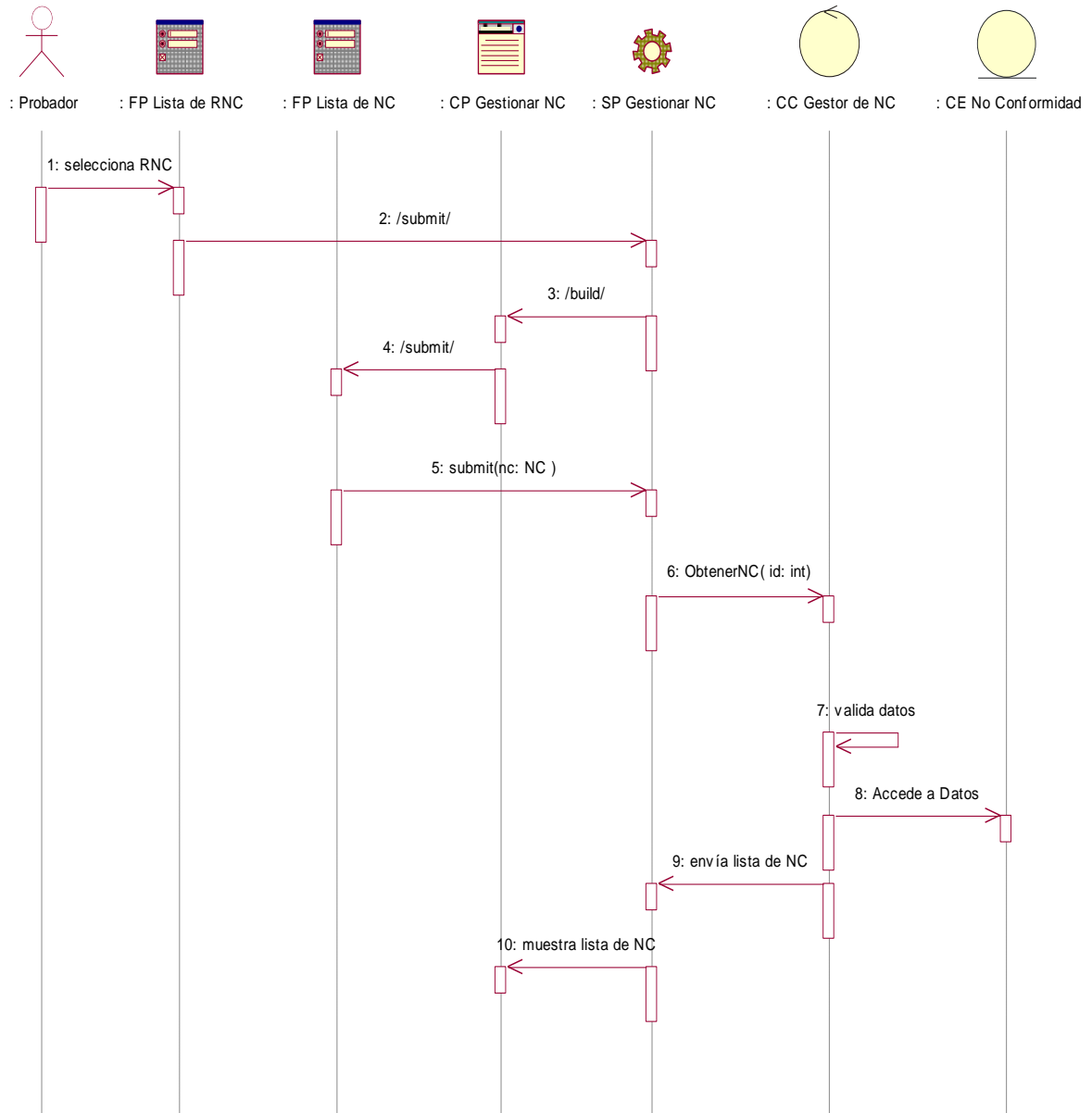


Fig. 3.7 –DS CU Gestionar No Conformidades (Listar NC).

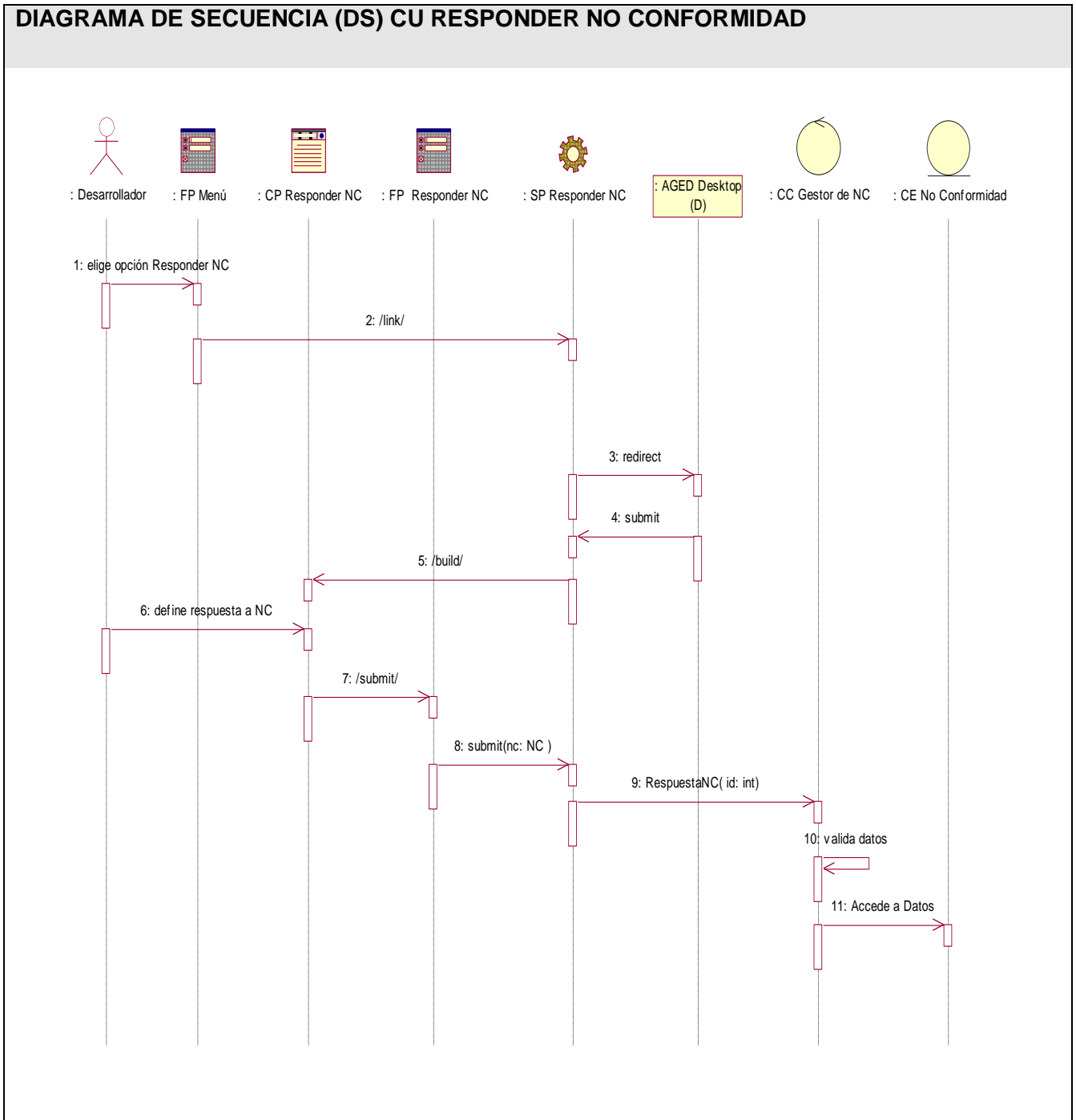


Fig. 3.8 –DS CU Responder No Conformidades.

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 8

Nombre: CI Gestionar No Conformidad

Tipo de clase: Interfaz	
Atributo	Tipo
Área Temática	Textbox
Proyecto	Textbox
Producto	Textbox
Tipo de Elemento	Textbox
Nombre del Elemento	Textbox
Lista de Registros de NC	GridView
Lista de No Conformidades	GridView
Nuevo Registro de NC	Button
Agregar NC	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Nuevo_Click(Object sender, EventArgs e).
Descripción:	Este método se ejecuta cuando se presiona el botón Nuevo Registro de NC. Invoca a la funcionalidad encargada de crear Registros de NC.
Nombre:	void Nuevo_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Agregar NC. Invoca a la funcionalidad encargada de crear No Conformidades.
Nombre:	void Cancelar_Click(Object sender, EventArgs e).
Descripción:	Este método se ejecuta cuando se presiona el botón Cancelar. Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla 3.1 -.CI Gestionar No Conformidades

Nombre: CI Responder No Conformidad	
Tipo de clase: Interfaz	
Atributo	Tipo
Área Temática	Textbox
Proyecto	Textbox
Producto	Textbox
Tipo de Elemento	Textbox
Nombre del Elemento	Textbox
Lista de Registros de NC	GridView
Lista de No Conformidades	GridView
Responder NC	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Responder_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón Responder NC. Invoca a la funcionalidad encargada de cambiar los estados de las No Conformidades.
Nombre:	void Cancelar_Click(Object sender, EventArgs e).
Descripción:	Este método se ejecuta cuando se presiona el botón Cancelar. Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla 3.2 -.CI Responder No Conformidades

Nombre: CC Gestionar No Conformidades
Tipo de clase: Controladora

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void AdicionarNC(NC nc)
Descripción:	Este método es el encargado de adicionar una nueva No Conformidad.
Nombre:	void ModoficarNC(NC nc)
Descripción:	Este método modifica una No Conformidad.
Nombre:	void EliminarNC(NC nc)
Descripción:	Este método es el encargado de eliminar una No Conformidad.
Nombre:	void Cambiar_SoluciónNC(NC nc, String solucion)
Descripción:	Este método es el encargado de cambiar el estado de solución de una No Conformidad.
Nombre:	void Cambiar_AceptaciónNC(NC nc, String aceptacion)
Descripción:	Este método es el encargado de cambiar el estado de aceptación de una No Conformidad.

Tabla 3.3 -.CC Gestionar No Conformidades

Nombre: CE Reporte_No_Conformidad	
Tipo de clase: Entidad	
Atributo	Tipo
ident_Registro	Integer
usuario	String
ident_Elem	Integer
etapa_Deteccion	String

fecha	DateTime
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.4 -.CE Reporte_No_Conformidad

Nombre: CE No_Conformidad	
Tipo de clase: Entidad	
Atributo	Tipo
numero_NC	Integer
ident_Registro	Integer
clasificacion	String
estado_Solucion	String
estado_Aceptacion	String
descripcion	String
epígrafe	Integer
sub_Epigrafe	Integer
parrafo	Integer
explicacion	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla 3.5 -.CE No_Conformidad

En el presente capítulo se definieron los patrones de diseño que acompañan la documentación e implementación durante el ciclo de desarrollo de software. La arquitectura definida está orientada al logro de una base sólida para la construcción del sistema. Fueron obtenidos los artefactos correspondientes a los flujos de trabajo desarrollados acorde a la metodología utilizada.

CAPÍTULO 4: IMPLEMENTACIÓN

En el presente capítulo, se muestran el diseño de la Base de Datos, el Diagrama de Despliegue y el Diagrama de Componentes. Además se describe el tratamiento de errores, las acciones para garantizar la seguridad y los estándares usados en la codificación con el objetivo de lograr una mejor comprensión de cómo funciona la herramienta presentada.

4.1. Modelo de Datos

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

4.1.1. Diseño de la Base de Datos

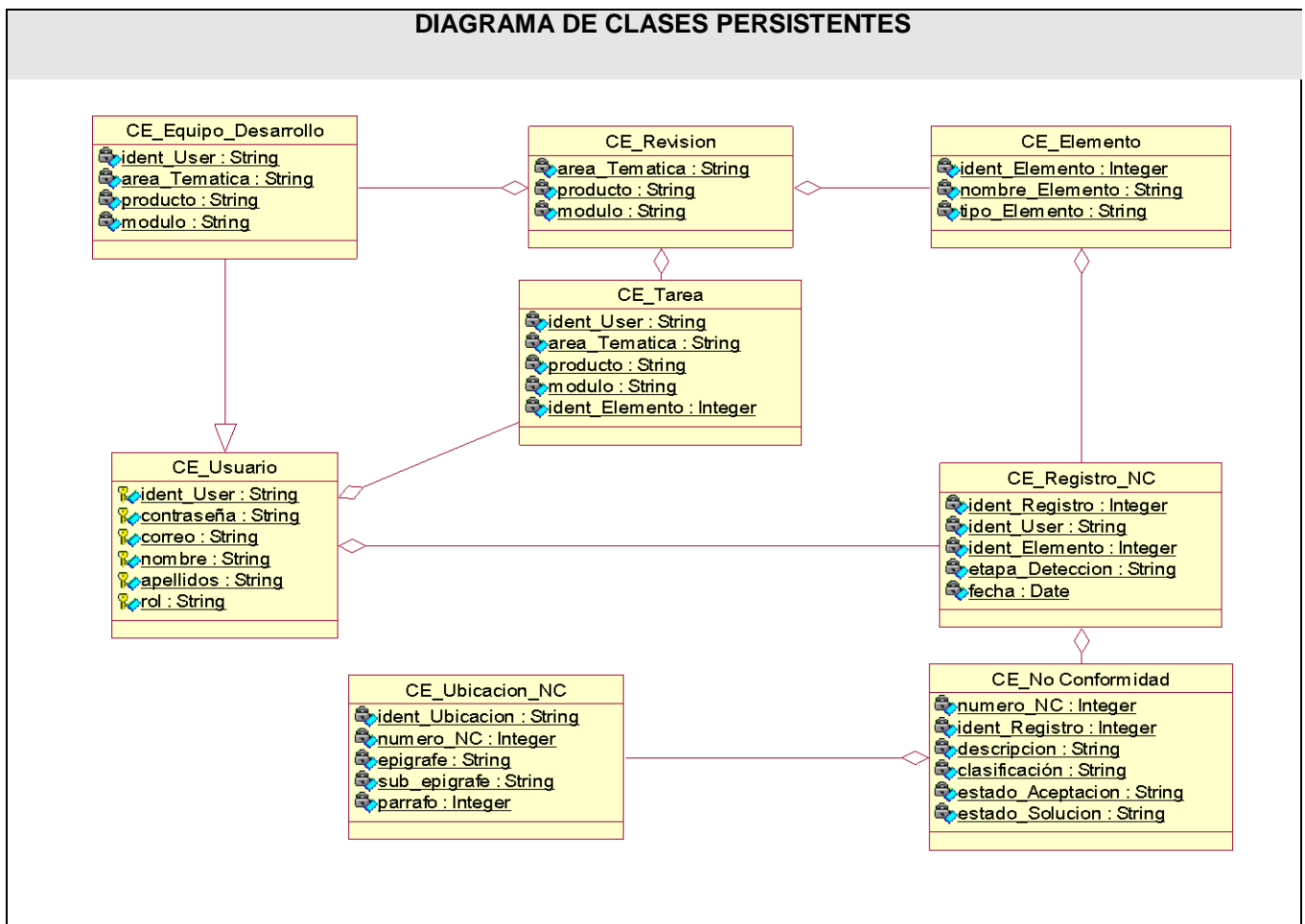


Fig. 4.1 –Diagrama de Clases Persistentes.

estado	varchar(50)	Contiene el estado de ejecución en el que se encuentra el Proceso de Pruebas aplicado a la revisión (sin comenzar, iniciada, terminada).
fecha_Inicio	datetime	Contiene el valor de la fecha en la que se inició el Proceso de Pruebas a la revisión. (Se obtiene automáticamente cuando se cambia el estado de la revisión a “Iniciada”).
fecha_Fin	datetime	Contiene el valor de la fecha en la que se terminó el Proceso de Pruebas a la revisión. (Se obtiene automáticamente cuando se cambia el estado de la revisión a “Terminada”).

Tabla 4.1 -.Tabla de la BD - Revision

Nombre: Usuario		
Descripción: Almacena los datos correspondientes a los Usuarios que interactúan con la aplicación.		
Atributo	Tipo	Descripción
ident_User	varchar	Llave primaria: Representa el identificador del usuario, atributo único. Permite realizar las acciones de búsqueda y modificaciones. Atributo que no cambia su valor y mantiene la integridad y relación entre los datos.
contraseña	varchar	Permite la autenticación dentro de la aplicación, corresponde a un usuario determinado.
correo	varchar	Propiedad que contiene el valor de e-mail de un usuario (en entornos donde la comunicación fluya a través de esta vía es útil para la notificación de las tareas).

nombre	varchar	Contiene el (los) nombre(s) del usuario.
apellidos	varchar	Contiene el (los) apellido(s) del usuario.
rol	varchar	Permite garantizar que el usuario acceda a las funcionalidades acordes a sus necesidades dentro de la aplicación, sirve para establecer los permisos. Se basa en el papel que juega el usuario dentro del Proceso de Pruebas a la Documentación: “Administrador de Pruebas”, “Jefe de Pruebas”, “Administrador de Calidad”, “Probador”, “Desarrollador”.

Tabla 4.2 -.Tabla de la BD - Usuario

Nombre: Equipo_Desarrollo		
Descripción: Almacena los datos correspondientes a los usuarios que forman parte de un Equipo de Desarrollo (Usuarios designados como responsables del Proceso de Pruebas por parte del proyecto que se somete a la evaluación).		
Atributo	Tipo	Descripción
ident_User	varchar	Llave foránea: El atributo es heredado del usuario correspondiente (usuario al que pertenece la especialización del Equipo de Desarrollo).
Area_Tematica	varchar	Llave foránea: El atributo es heredado de la revisión correspondiente (revisión a la que pertenece la especialización del Equipo de Desarrollo).
producto	varchar	Llave foránea: El atributo es heredado de la revisión correspondiente (revisión a la que pertenece la especialización del Equipo de Desarrollo).

modulo	varchar	Llave foránea: El atributo es heredado de la revisión correspondiente (revisión a la que pertenece la especialización del Equipo de Desarrollo).
--------	---------	--

Tabla 4.3 -.Tabla de la BD – Equipo_Desarrollo

Nombre: Elemento		
Descripción: Almacena los datos correspondientes a los documentos que pueden estar incluidos en una revisión. Sobre estos documentos se ejecuta la acción de los Desarrolladores y Probadores para detectar o corregir errores.		
Atributo	Tipo	Descripción
ident_Elemento	integer	Llave primaria: Representa el identificador del elemento, atributo único. Permite realizar las acciones de búsqueda y modificaciones. Atributo que no cambia su valor y mantiene la integridad y relación entre los datos. (Autogenerado).
nombre_ Elemento	varchar	Contiene el nombre del elemento (está dado acorde a la configuración actual determinada por el Expediente de Proyecto).
tipo_Elemento	varchar	Contiene el tipo del elemento (está dado acorde a la configuración actual determinada por el Expediente de Proyecto: "Ingeniería", "Gestión de Proyectos", "Soporte").

Tabla 4.4 -.Tabla de la BD - Elemento

Nombre: Registro_NC
Descripción: Almacena los datos correspondientes a los Registro de No Conformidades (RNC) que contiene un elemento y son generados por un usuario. Se encarga de contener todas las No

Conformidades (NC) detectadas por los Probadores, así como de mantener el control del historial de las NC corregidas por los Desarrolladores.		
Atributo	Tipo	Descripción
ident_Registro	integer	Llave primaria: Representa el identificador del RNC, atributo único. Permite realizar las acciones de búsqueda y modificaciones. Atributo que no cambia su valor y mantiene la integridad y relación entre los datos. (Autogenerado).
ident_Elemento	integer	Llave foránea: El atributo es heredado del elemento correspondiente (elemento al que pertenece el RNC).
ident_user	varchar	Llave foránea: El atributo es heredado del usuario correspondiente (usuario que generó el RNC).
fecha	datetime	Contiene el valor de la fecha en la que fue generado el RNC.

Tabla 4.5 -.Tabla de la BD – Registro_NC

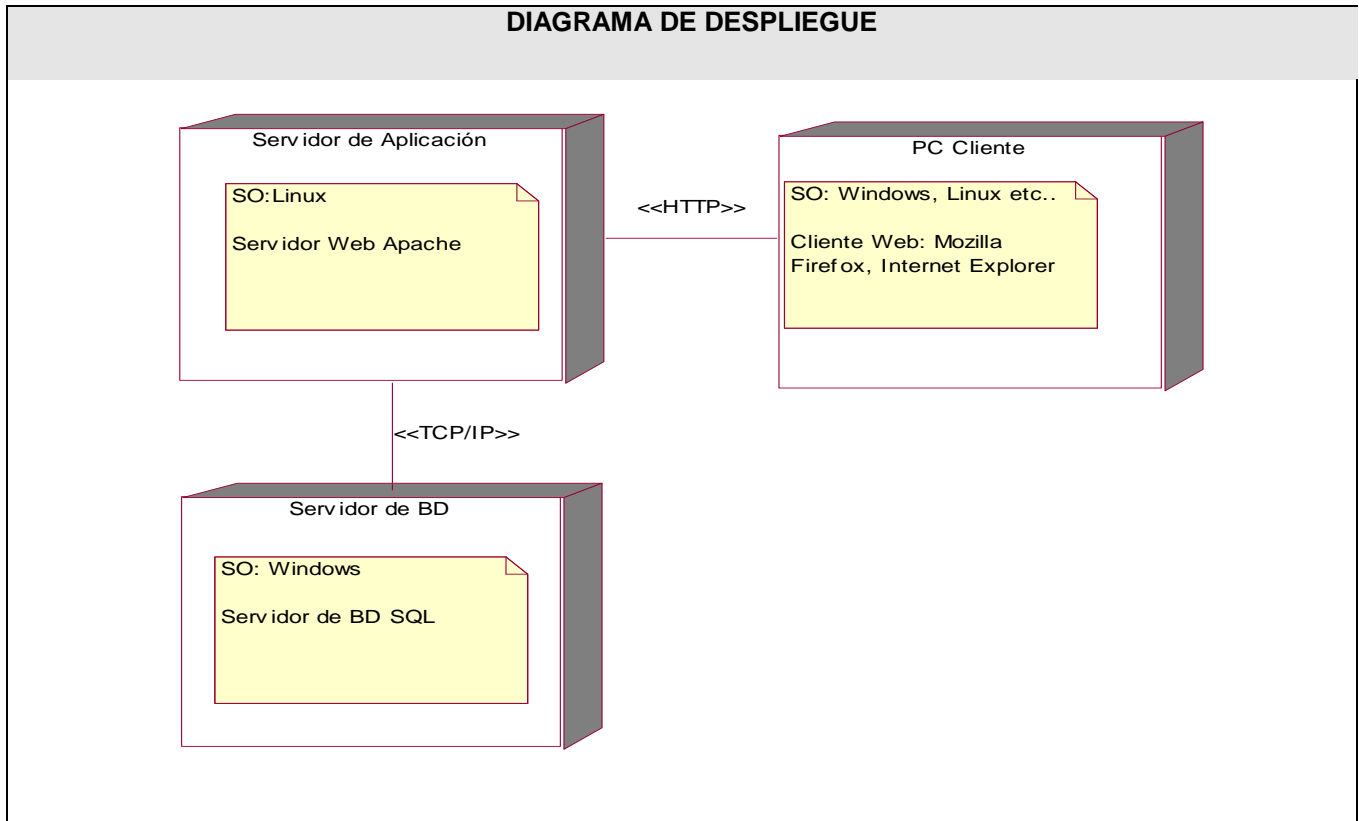
Nombre: No_Conformidad		
Descripción: Almacena los datos correspondientes a las No Conformidades (NC) que son almacenadas en un RNC. Se encarga de contener todas las especificaciones de la NC detectada por los Probadores, así como de mantener el control del historial de los cambios de estados efectuados por los Desarrolladores.		
Atributo	Tipo	Descripción
numero_NC	integer	Llave primaria: Representa el identificador del RNC, atributo único. Permite realizar las acciones de búsqueda y modificaciones. Atributo que no cambia su valor y mantiene la

		integridad y relación entre los datos. (Autogenerado).
ident_Registro	integer	Llave foránea: El atributo es heredado del RNC correspondiente (RNC donde se encuentra contenida la NC).
descripción	varchar	Contiene la descripción detallada de la NC detectada.
etapa_Deteccion	varchar	Contiene el momento, durante el Proceso de Pruebas, en el que se detectó la NC
clasificacion	varchar	Contiene la clasificación en cuanto a importancia de la NC detectada: "Significativa", "No Significativa", "Recomendación".
estado_Aceptacion	varchar	Contiene el valor de aceptación otorgado por el Equipo de Desarrollo a la NC detectada: "Procede", "No procede".
estado_Solucion	varchar	Contiene el valor de solución otorgado por el Equipo de Desarrollo a la NC detectada: "Pendiente", "Resuelta", "No resuelta".

Tabla 4.6 -.Tabla de la BD – No_Conformidad

4.2. Implementación

4.2.1. Diagrama de Despliegue



4.2.2. Diagrama de Componentes

Debido a que la aplicación está integrada a una aplicación de escritorio, la cual es invocada para la realización de algunas operaciones se desglosará el Diagrama de Componentes teniendo en cuenta las características individuales de cada parte del sistema. Luego se mostrará la integración general.

DIAGRAMA DE COMPONENTES AGED-DESKTOP

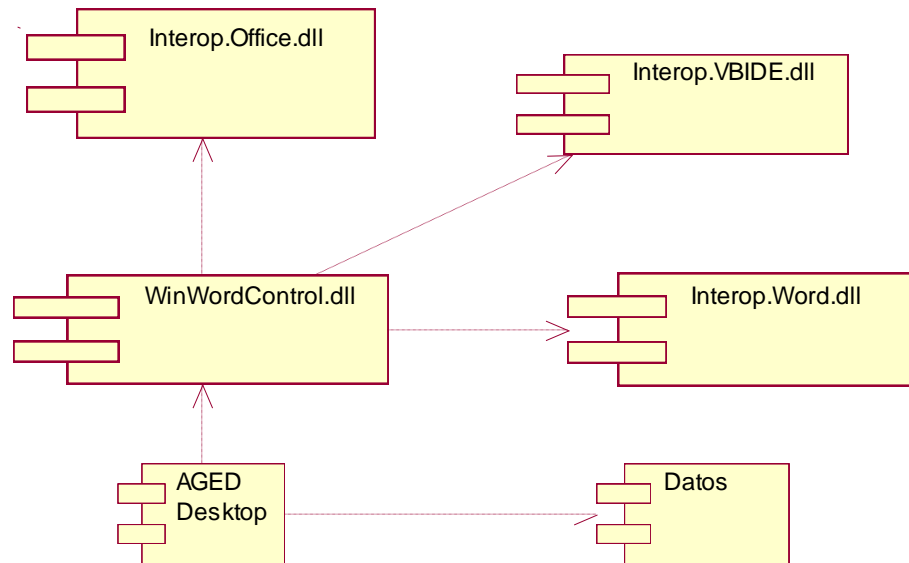
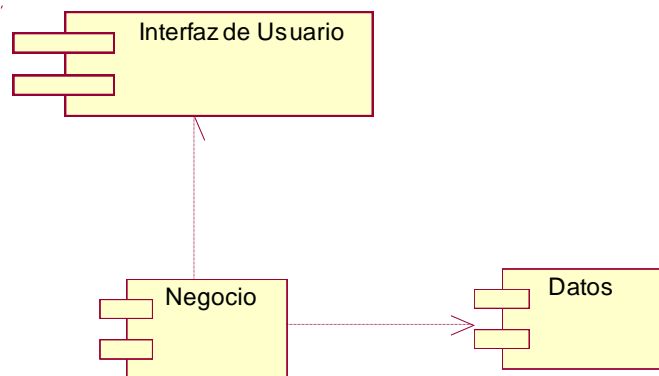
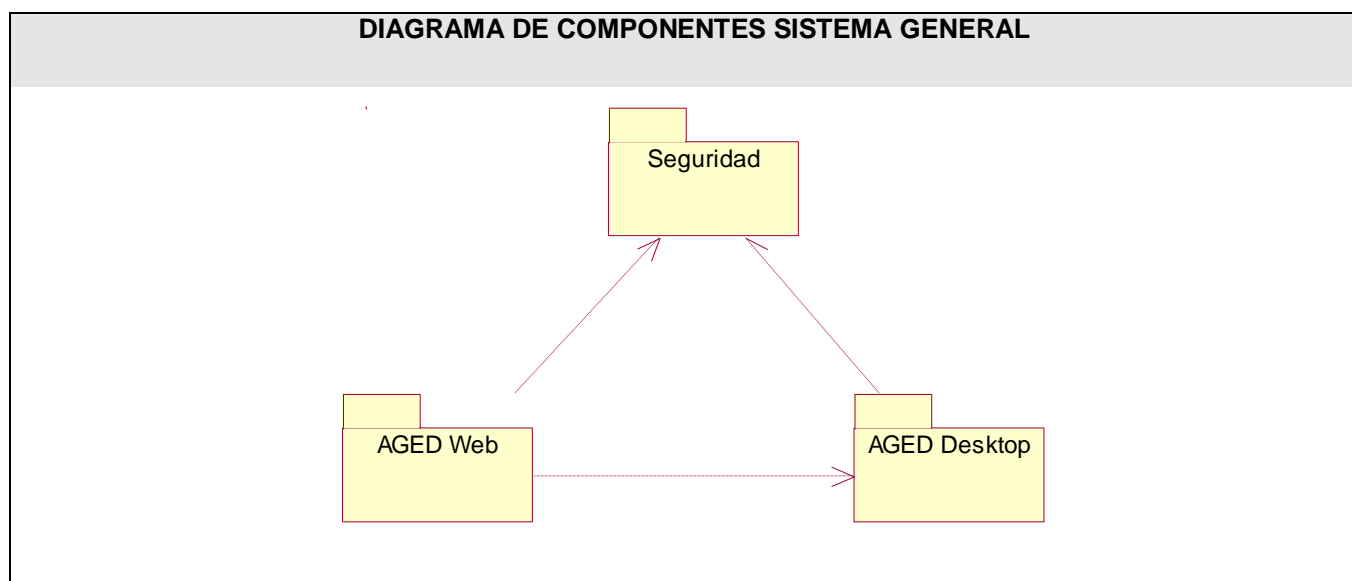


DIAGRAMA DE COMPONENTES AGED-WEB





4.3. Tratamiento de errores

El sistema cuenta con varios mecanismos para el trabajo con los errores, enfocándose en la mitigación de los mismos. El uso de ASP.NET permite usar controles predefinidos que validan los datos de entradas definidos para el usuario sin necesidad de realizar peticiones al servidor, sumándose a esta ventaja el hecho de poder ser utilizados sin necesidad de incluir líneas de código.

La interfaz Web está diseñada para que el usuario realice sus acciones entrando al sistema solo los campos indispensables, ahorrando tiempo al usuario y evitando acciones innecesarias que puedan provocar errores. Para cumplir este objetivo se cuenta con elementos configurables como campos de selección, listas desplegadas con varias opciones y campos autogenerados por el sistema de forma transparente al usuario.

El sistema consta con funciones de validación y captación de errores, informando al usuario en momentos importantes sobre la realización de acciones incorrectas y la forma de corregir dicha acción. El sistema, junto al manejo de excepciones, cuenta con la posibilidad de enviar mensajes explícitos al usuario informando sobre la realización exitosa o cancelación de una acción predeterminada, evitando siempre abusar de los mensajes de información.

4.4. Seguridad

Debido a que el sistema implementado está basado en la Web la seguridad juega un papel fundamental para el correcto funcionamiento. El uso de ASP.NET permite hacer uso de la

gestión de permisos de acceso sobre los recursos del sistema de forma dinámica; mediante las variables de sesión se controla en todo momento la información vital sobre el usuario que se encuentra haciendo uso de las funcionalidades del sistema (las variables de sesión, a diferencia de otras tecnologías, se manejan del lado del servidor, evitando la modificación con fines dañinos al sistema). Además, el uso de los ficheros web_config permite validar desde un nivel general hasta el nivel más simple los permisos acorde a los roles de usuarios.

Para la interacción con el sistema se han definido una serie de roles que se enumeran según el orden jerárquico de la siguiente forma: Administrador de Pruebas, Jefe de Pruebas. Administrador de Calidad, Probador y Desarrollador. Los permisos para cada grupo de usuarios se obtienen al realizarse una autenticación exitosa.

La integración existente entre el SGBD (SQL Server Express Edition) y el lenguaje de programación utilizados permite obtener de forma puntual los datos necesarios para la gestión de la información necesaria e indispensable para la realización de las acciones referentes al usuario (acorde a su nivel de acceso). Esto garantiza que a cada usuario se le muestre solo la información que necesita para desempeñarse de manera efectiva en el sistema.

Los mensajes informativos que se muestran al usuario evitarán en todo momento mostrar información sensible del sistema.

4.5. Estrategia de codificación. Estándares y estilos a utilizar

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

A continuación se presentan algunas de estas convenciones para la programación en el lenguaje C# usadas para la programación de la aplicación implementada.

Comentarios

Deberán incluir:

- Autor
- Fecha
- Descripción

Algoritmo.

Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.
- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.

Identificadores de variables

Para distinguir palabras dentro del nombre deberá emplearse una letra mayúscula o un guión bajo (_), sin mezclar ambas formas en un mismo programa. A partir de la segunda palabra se usará mayúscula; preferiblemente no usar más de 2 palabras por identificador siempre y cuando en el nombre quede implícita la acción.

Ejemplos:

descripcion_NC

ubicaciones

Identificadores de clases y métodos

Los identificadores de clases comenzarán escritos con letra mayúscula. En caso de tener más de una palabra estas se separarán por un guión bajo (_) y a continuación se comenzará de nuevo con mayúscula.

Nota: las clases serán implementadas por separado, incluyendo en cada una todas las funcionalidades que sean necesarias y pertenezcan al dominio de la clase.

Ejemplos:

No_Conformidad

Organización Visual del Programa

Generales

- No manejar en los programas más de una instrucción por línea.
- Declarar las variables del mismo tipo en una línea.

Sangrías

- Las sangrías tendrán una longitud de tres espacios.
- Para las llaves que definen el cuerpo de una función, sangría un nivel.

Ejemplo: void Funcion ()

```
{  
//Instrucciones de la función  
}
```

- Sangría para las instrucciones del cuerpo de cada estructura de control.

Ejemplo: for (int x = 0; x < 5; x++)

```
{  
//Instrucciones a ejecutar  
}
```

Líneas y espacios en blanco

- Insertar una línea en blanco antes y después de una declaración de datos que aparezca entre instrucciones ejecutables.
- Las declaraciones de datos dentro de una función, deberán ir al inicio y separadas de las instrucciones ejecutables de la función por medio de una línea en blanco.
- Los operadores unarios (++ , -- , etc.) deben ponerse junto a sus operandos, sin espacios intermedios.

En el capítulo se implementó la herramienta para la revisión y seguimiento de errores de la documentación en los proyectos productivos de la Facultad 7. Con esta aplicación, se espera facilitar y agilizar el proceso de aseguramiento de la calidad de la documentación en la misma.

CONCLUSIONES

Al concluir la investigación se han cumplido el objetivo y las tareas propuestas:

Las herramientas existentes relacionadas con la revisión y seguimiento de errores no responden a las necesidades que se presentan en el Proyecto de Calidad de La Facultad 7, fundamentalmente enfocadas a la revisión de la documentación.

Se realizó un análisis de las tendencias actuales de las herramientas y tecnologías lográndose seleccionar las más adecuadas para la implementación del sistema teniendo en cuenta las particularidades del mismo.

Después de valorar las principales problemáticas existentes en el proyecto de calidad de la Facultad 7, se definieron una serie de aspectos que afectan el proceso de revisión y seguimiento de errores de la documentación de los diferentes proyectos de la facultad que deben ser liberados por el laboratorio de Calidad.

Se definieron las necesidades de funcionamiento de la aplicación así como los artefactos correspondientes a los flujos de trabajo definidos por la metodología seleccionada.

Se obtuvo una aplicación para la Revisión y Seguimiento de los Errores de la Documentación de los Proyectos de la Facultad 7, que agiliza y facilita este proceso.

RECOMENDACIONES

Para futuras investigaciones, se recomienda:

Incorporar al sistema un módulo para el trabajo con Open Office, con vistas a la migración.

Realizar un análisis más detallado sobre los reportes de utilidad para los usuarios del sistema que se puedan incorporar además de los que se encuentran en la aplicación.

Incorporar al sistema un módulo para la Gestión de No Conformidades de Software con el fin de lograr una herramienta más completa.

REFERENCIAS BIBLIOGRÁFICAS

- Achour, Mehdi, y otros.** Manual de PHP. *PHP*. [En línea] <http://es.php.net/manual/es/>.
- Aguilar, José M. 2009.** Diez preguntas básicas (con respuesta) sobre ASP.Net MVC. [En línea] Abril de 2009. <http://geeks.ms/blogs/jmaguilar/archive/2008/03/24/diez-preguntas-b-225-sicas-con-respuesta-sobre-asp-net-mvc.aspx>.
- Agustín Dondo.** ¿Por qué elegir PHP? *PHP en castellano*. [En línea] <http://www.programacion.com/php/articulo/porquephp/>.
- Álvarez, Miguel Ángel. 2001.** ¿Qué es ASP? *Desarrollo Web*. [En línea] Mayo de 2001. <http://www.desarrolloweb.com/articulos/393.php>.
- Avidos.net. 2005.** ventaja e inconvenientes de las aplicaciones web. *Blogold*. [En línea] Enero de 2005. <http://www.avidos.net/blogold/aplicaciones-web>.
- Bug-a-Boo.org. 2005.** Bug-A-Boo - A web based bug tracking system. [En línea] Febrero de 2005. <http://www.bug-a-boo.org>.
- Bugzilla.org.** Bugzilla. *Bugzilla*. [En línea] <http://www.bugzilla.org>.
- CAVSI. 2004.** ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] Junio de 2004. <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
- Chacón, Andrés. 2007.** HERRAMIENTAS ASISTIDAS POR COMPUTADORA PARA LA INGENIERÍA DE SISTEMAS. [En línea] 2007. http://pdf.rincondelvago.com/herramientas-case_1.html.
- Ciberaula.com. 2005.** Una Introducción a APACHE. *Ciberaula*. [En línea] Septiembre de 2005. http://linux.ciberaula.com/articulo/linux_apache_intro.
- CIIN, Blog del.** Software As a Service (SaaS): ¿Qué es? *Blog del CIIN (Blog del Centro de Innovación en Integración de Cantabria)*. [En línea] <http://geeks.ms/blogs/ciin/archive/2007/10/05/software-as-a-service-sas-191-qu-233-es.aspx>.
- Corp, Neuron. 2006.** ¿Qué es UML? [En línea] 2006. http://www.neuronsrl.com.ar/training/uml/uml_intro.html.
- De Nobrega, Maria. 2005.** Herramientas CASE: Rational Rose. [En línea] Junio de 2005. http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php.
- Díaz Colomé, Yanersy y Molina Hernández, Yenisel. 2007.** *Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7*. Ciudad de La Habana : s.n., 2007.

Eaprende.com. 2001. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. *Aprende.com*. [En línea] Enero de 2001. <http://www.eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>.

FormaSelect. Introduccion a SQL Server 2000. *FormaSelect*. [En línea] <http://www.formaselect.com/curso/experto-en-sql-server-2000/Introduccion-a-SQL-Server%202000.pdf>.

Foundation, Free Software. 2005. GNATS the GNU Bug Tracking System. *GNATS-GNU Project*. [En línea] Marzo de 2005. <http://www.gnu.org/software/gnats/>.

Freedownloadmanager.org. 2004. Visual Paradigm for UML (ME) 6.0. [En línea] Mayo de 2004. [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

Froufe, Agustín. 2005. ARQUITECTURA Modelo/Vista/Controlador. [En línea] 2005. http://www.ulpgc.es/otros/tutoriales/java/Apendice/arq_mvc.html.

Gold, Mike. 2006. Creating and Opening Microsoft Word Documents from .NET Using C#. *C# Corner*. [En línea] Octubre de 2006. <http://www.c-sharpcorner.com/UploadFile/mgold/WordFromDotNet11082005235506PM/WordFromDotNet.aspx>.

Gómez, Diego. 2008. Scrum garantiza un alto rendimiento. [En línea] Septiembre de 2008. <http://www.dosideas.com/metodologias/223-scrum-garantiza-un-alto-rendimiento-empresarial.html>.

González, Iván. 2007. Introducción a Internet Information Server 7. [En línea] Enero de 2007. <http://geeks.ms/blogs/iis/>.

Graham, Paul. 2001. The Other Road Ahead. [En línea] Septiembre de 2001. <http://www.paulgraham.com/road.html>.

Group, Mantis. 2009. My View - MantisBT DEMO site. *Mantis*. [En línea] Abril de 2009. http://www.mantisbt.org/demo/my_view_page.php.

Hänel, Matthias. 2003. Word Control for .NET. *The Code Project*. [En línea] Enero de 2003. <http://www.codeproject.com/KB/miscctrl/winwordcontrol.aspx>.

Hervás Gómez, Carlos. Las nuevas tecnologías en la Educación Primaria. Internet: los procesos de enseñanza-aprendizaje con la world wide web. *Grupo de Tecnología Educativa, Universidad de Sevilla*. [En línea] <http://tecnologiaedu.us.es/edutec/2libroedutec99/4.9.htm>.

Hipergate.org. 2008. Hipergate: Free Open Source CRM Groupware and Intranet Software. [En línea] Noviembre de 2008. <http://www.hipergate.org/>.

Htmlpoint.com. 2008. ¿Qué es Apache? [En línea] Diciembre de 2008. <http://www.htmlpoint.com/faq/apache/01.htm>.

Islasoft.com. 2006. Diferencias entre Microsoft Access y Microsoft SQL Server/MSDE. *IsaSoft.* [En línea] Septiembre de 2006. http://www.islasoft.com/Producto/Diferencias_SQL_Server_Office.pdf.

Janium.com. 2009. Aplicaciones basadas en Web. [En línea] 2009. <http://www.janium.com/page2/page1/page6/page7/page7.html>.

Letelier, Patricio y Carmen Penadés, María. 2006. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* 2006.

Martínez Pérez, Yancy, Díaz Domínguez, Alexey y Lorente Rodríguez, Abel Ernesto. 2006. *PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS PRODUCTOS DE LA COLECCIÓN MULTISABER.* Ciudad de La Habana : s.n., 2006.

Masip, David. 2002. ¿Qué es Oracle? *Desarrolloweb.com.* [En línea] Julio de 2002. <http://www.desarrolloweb.com/articulos/840.php>.

Masternewmedia.org. 2005. Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era “En Vivo”. [En línea] Noviembre de 2005. 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.

Methodologies, Crystal. 2008. Crystal Methodologies. *Crystal Methodologies.* [En línea] Mayo de 2008. <http://crystalmethodologies.blogspot.com/>.

Microsiervos.com. 2005. ¿Qué es Internet? [En línea] Junio de 2005. <http://www.microsiervos.com/archivo/internet/que-es-internet.html>.

Microsoft.com. 2007. Internet Information Services. [En línea] Julio de 2007. <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.

—. SQL Server 2005 Express Edition. *Microsoft.com.* [En línea] <http://www.microsoft.com/spanish/msdn/vstudio/express/SQL/default.msp>.

Miguel, Luis. 2009. DSDM (Método de Desarrollo de Sistemas Dinámicos). [En línea] Febrero de 2009. <http://matemiguel-usac.spaces.live.com/blog/cns!934A4D34E7CB27EE!164.entry>.

Millán Tejedor, Ramón Jesús. 2005. Desarrollo de Sitios web dinámicos. [En línea] Diciembre de 2005. <http://www.ramonmillan.com/documentos/dhtml.pdf>.

Millhollon, Mary y Murray, Katherine. 2004. *Microsoft Word 2003 Inside Out.* 2004.

Monografias.com. 2007. Definición arquitectura cliente servidor. [En línea] 2007. <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.

- Moody, David.** The Bug Genie. *The Bug Genie*. [En línea] <http://www.thebuggenie.com>.
- Morrell Rojas, Amari Jacsson.** 2007. *Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad*. Ciudad de la Habana : s.n., 2007.
- Nieves Borrero, Martha y Góngora Rodríguez, Asnier.** 2007. *Herramienta Informática para automatizar los procesos en el Laboratorio de Calidad de Software: Módulo Gestión de las No Conformidades*. Ciudad de La Habana : s.n., 2007.
- Project, Trac Open Source.** 2008. The Trac Project. *The Trac Project*. [En línea] Octubre de 2008. <http://trac.edgewall.org/>.
- Ramírez Mastrap, Yudenia.** 2007. *LA GESTIÓN DE ERRORES EN EL DESARROLLO DE SOFTWARE*. 2007.
- Reynolds-Haertle, Robin.** 2002. Tips and Tricks: Building Microsoft Office Add-ins with Visual C# .NET and Visual Basic .NET. *MSDN*. [En línea] Septiembre de 2002. <http://msdn.microsoft.com/en-us/library/aa289518.aspx>.
- Ronda Amador, Yoelys, Cobas Santos, Kadir y Marrero Viñas, Daymel.** 2002. *El modelo cliente-servidor de tres capas para el trabajo en redes de información*. 2002.
- Sánchez González, Carlos.** 2004. *ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras*. Da Coruña : s.n., 2004.
- Sourceforge.net.** 2008. CodeTrack: Bug Reporting and Tracking for the Rest of Us! [En línea] Octubre de 2008. <http://kennwhite.sourceforge.net/codetrack>.
- Sparxsystems.com.** 2006. Enterprise Architect - Herramienta de diseño UML. [En línea] 2006. <http://www.sparxsystems.com.ar/products/ea.html>.
- Subgurim.NET.** 2006. ¿Por qué ASP.NET? [En línea] Febrero de 2006. <http://www.subgurim.net/Articulos/asp-net-general/3/por-que-asp-net.aspx>.
- Support, Microsoft Help and.** 2006. Cómo utilizar el control WebBrowser para abrir un documento de Office en Visual C# 2005 o en Visual C# .NET. *Microsoft Help and Support*. [En línea] Octubre de 2006. <http://support.microsoft.com/kb/304662/es>.
- Valencia, Universidad Politécnica de.** 2001. Introducción a Herramientas CASE y System Architect. *Universidad Politécnica de Valencia*. [En línea] Abril de 2001. http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.
- . 2007. Rational Unified Process (RUP). *Universidad Politécnica de Valencia*. [En línea] Mayo de 2007. <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>.
- Valparaiso, Universidad de.** 2006. Evaluación del lenguaje. *Educnet-Universidad de Valparaíso*. [En línea] Febrero de 2006. <http://educnet.decom-uv.cl/educnet/uploads/Implementacion.pdf?nombre=p379/Implementacion.pdf>.

Referencias Bibliográficas

Vegas, Jesús. 2002. El Servidor Web. *Departamento de Informática (Universidad de Valladolid)*. [En línea] Marzo de 2002. <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.

Vico.org. 2002. Unified Modeling Language. *Vico.org*. [En línea] 2002. <http://www.vico.org/FormMentorOutsourcingUML.pdf>.

Villaverde Masa, Pablo. 2007. ASP. Recursos y características. [En línea] Noviembre de 2007. http://tgp0607.awardspace.com/Recursos_ASP.pdf.

Zamora, Universidad Nacional de Lomas de. 2008. ¿Qué es Internet? [En línea] Diciembre de 2008. <http://www.unlz.edu.ar/biblioteca/tutores/histoweb/datosgral.html>.

Zentrack.net. 2009. Zentrack. [En línea] Abril de 2009. <http://www.zentrack.net>.

BIBLIOGRAFÍA

- Achour, Mehdi, et al.** Manual de PHP. *PHP*. [Online] <http://es.php.net/manual/es/>.
- Aguilar, José M. 2009.** Diez preguntas básicas (con respuesta) sobre ASP.Net MVC. [Online] Abril 2009. <http://geeks.ms/blogs/jmaguilar/archive/2008/03/24/diez-preguntas-b-225-sicas-con-respuesta-sobre-asp-net-mvc.aspx>.
- Agustín Dondo.** ¿Por qué elegir PHP? *PHP en castellano*. [Online] <http://www.programacion.com/php/articulo/porquephp/>.
- Álvarez, Miguel Ángel. 2001.** ¿Qué es ASP? *Desarrollo Web*. [Online] Mayo 2001. <http://www.desarrolloweb.com/articulos/393.php>.
- Avidos.net. 2005.** ventaja e inconvenientes de las aplicaciones web. *Blogold*. [Online] Enero 2005. <http://www.avidos.net/blogold/aplicaciones-web>.
- Bug-a-Boo.org. 2005.** Bug-A-Boo - A web based bug tracking system. [Online] Febrero 2005. <http://www.bug-a-boo.org>.
- Bugzilla.org.** Bugzilla. *Bugzilla*. [Online] <http://www.bugzilla.org>.
- CAVSI. 2004.** ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [Online] Junio 2004. <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
- Chacón, Andrés. 2007.** HERRAMIENTAS ASISTIDAS POR COMPUTADORA PARA LA INGENIERÍA DE SISTEMAS. [Online] 2007. http://pdf.rincondelvago.com/herramientas-case_1.html.
- Ciberaula.com. 2005.** Una Introducción a APACHE. *Ciberaula*. [Online] Septiembre 2005. http://linux.ciberaula.com/articulo/linux_apache_intro.
- CIIN, Blog del.** Software As a Service (SaaS): ¿Qué es? *Blog del CIIN (Blog del Centro de Innovación en Integración de Cantabria)*. [Online] <http://geeks.ms/blogs/ciin/archive/2007/10/05/software-as-a-service-sas-191-qu-233-es.aspx>.
- Corp, Neuron. 2006.** ¿Qué es UML? [Online] 2006. http://www.neuronsrl.com.ar/training/uml/uml_intro.html.
- De Nobrega, Maria. 2005.** Herramientas CASE: Rational Rose. [Online] Junio 2005. http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php.
- Desarrolloweb.com. 2006.** Ejecutar archivo .exe en página web. *Desarrollo web.com*. [Online] Agosto 2006. <http://www.desarrolloweb.com/faq/554.php>.

Díaz Colomé, Yanersy and Molina Hernández, Yenisel. 2007. *Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7.* Ciudad de La Habana : s.n., 2007.

Eaprende.com. 2001. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. *Aprende.com.* [Online] Enero 2001. <http://www.eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>.

FormaSelect. Introduccion a SQL Server 2000. *FormaSelect.* [Online] <http://www.formaselect.com/curso/experto-en-sql-server-2000/Introduccion-a-SQL-Server%202000.pdf>.

Foundation, Free Software. 2005. GNATS the GNU Bug Tracking System. *GNATS-GNU Project.* [Online] Marzo 2005. <http://www.gnu.org/software/gnats/>.

Freedownloadmanager.org. 2004. Visual Paradigm for UML (ME) 6.0. [Online] Mayo 2004. [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

Froufe, Agustín. 2005. ARQUITECTURA Modelo/Vista/Controlador. [Online] 2005. http://www.ulpgc.es/otros/tutoriales/java/Apendice/arq_mvc.html.

Gold, Mike. 2006. Creating and Opening Microsoft Word Documents from .NET Using C#. *C# Corner.* [Online] Octubre 2006. <http://www.c-sharpcorner.com/UploadFile/mgold/WordFromDotNet11082005235506PM/WordFromDotNet.aspx>.

Gómez, Diego. 2008. Scrum garantiza un alto rendimiento. [Online] Septiembre 2008. <http://www.dosideas.com/metodologias/223-scrum-garantiza-un-alto-rendimiento-empresarial.html>.

González, Iván. 2007. Introducción a Internet Information Server 7. [Online] Enero 2007. <http://geeks.ms/blogs/iis/>.

Graham, Paul. 2001. The Other Road Ahead. [Online] Septiembre 2001. <http://www.paulgraham.com/road.html>.

Group, Mantis. 2009. My View - MantisBT DEMO site. *Mantis.* [Online] Abril 2009. http://www.mantisbt.org/demo/my_view_page.php.

Hänel, Matthias. 2003. Word Control for .NET. *The Code Project.* [Online] Enero 2003. <http://www.codeproject.com/KB/miscctrl/winwordcontrol.aspx>.

Hervás Gómez, Carlos. Las nuevas tecnologías en la Educación Primaria. Internet: los procesos de enseñanza-aprendizaje con la world wide web. *Grupo de Tecnología Educativa, Universidad de Sevilla.* [Online] <http://tecnologiaedu.us.es/edutec/2libroedutec99/4.9.htm>.

- Hipergate.org. 2008.** Hipergate: Free Open Source CRM Groupware and Intranet Software. [Online] Noviembre 2008. <http://www.hipergate.org/>.
- Htmlpoint.com. 2008.** ¿Qué es Apache? [Online] Diciembre 2008. <http://www.htmlpoint.com/faq/apache/01.htm>.
- Islasoft.com. 2006.** Diferencias entre Microsoft Access y Microsoft SQL Server/MSDE. *IslaSoft*. [Online] Septiembre 2006. http://www.islasoft.com/Producto/Diferencias_SQL_Server_Office.pdf.
- Jahedur Rahman, Mohammad. 2009.** File Uploader in ASP.NET and ASP.NET MVC. *The Code Project*. [Online] Enero 2009. <http://www.codeproject.com/KB/aspnet/Uploader.aspx>.
- Janium.com. 2009.** Aplicaciones basadas en Web. [Online] 2009. <http://www.janium.com/page2/page1/page6/page7/page7.html>.
- Katrib, Miguel, et al. 2008.** *Visual Studio 2008 Desafía todos los retos*. 2008.
- Letelier, Patricio and Carmen Penadés, María. 2006.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2006.
- LN, Krishnan. 2006.** Reading a word document using C#. *C# Corner*. [Online] Febrero 2006. <http://www.c-sharpcorner.com/UploadFile/Globalking/fileAccessingusingcsharp02242006050207AM/fileAccessingusingcsharp.aspx>.
- Martínez Pérez, Yancy, Díaz Domínguez, Alexey and Lorente Rodríguez, Abel Ernesto. 2006.** *PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS PRODUCTOS DE LA COLECCIÓN MULTISABER*. Ciudad de La Habana : s.n., 2006.
- Masip, David. 2002.** ¿Qué es Oracle? *Desarrolloweb.com*. [Online] Julio 2002. <http://www.desarrolloweb.com/articulos/840.php>.
- Masternewmedia.org. 2005.** Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era “En Vivo”. [Online] Noviembre 2005. 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.
- Methodologies, Crystal. 2008.** Crystal Methodologies. *Crystal Methodologies*. [Online] Mayo 2008. <http://crystalmethodologies.blogspot.com/>.
- Microsiervos.com. 2005.** ¿Qué es Internet? [Online] Junio 2005. <http://www.microsiervos.com/archivo/internet/que-es-internet.html>.
- Microsoft.com. 2007.** Internet Information Services. [Online] Julio 2007. <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.
- . SQL Server 2005 Express Edition. *Microsoft.com*. [Online] <http://www.microsoft.com/spanish/msdn/vstudio/express/SQL/default.msp>.

- Miguel, Luis. 2009.** DSDM (Método de Desarrollo de Sistemas Dinámicos). [Online] Febrero 2009. <http://matemiguel-usac.spaces.live.com/blog/cns!934A4D34E7CB27EE!164.entry>.
- Millán Tejedor, Ramón Jesús. 2005.** Desarrollo de Sitios web dinámicos. [Online] Diciembre 2005. <http://www.ramonmillan.com/documentos/dhtml.pdf>.
- Millhollon, Mary and Murray, Katherine. 2004.** *Microsoft Word 2003 Inside Out*. 2004.
- Monografias.com. 2007.** Definición arquitectura cliente servidor. [Online] 2007. <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
- Moody, David.** The Bug Genie. *The Bug Genie*. [Online] <http://www.thebuggenie.com>.
- Morrell Rojas, Amari Jacsson. 2007.** *Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad*. Ciudad de la Habana : s.n., 2007.
- Nieves Borrero, Martha and Góngora Rodríguez, Asnier. 2007.** *Herramienta Informática para automatizar los procesos en el Laboratorio de Calidad de Software: Módulo Gestión de las No Conformidades*. Ciudad de La Habana : s.n., 2007.
- Project, The Code. 2007.** Embed PDFs into a Web Page with a Custom Control. *The Code Project*. [Online] Enero 2007. <http://www.codeproject.com/KB/applications/EmbedPDFinWebpages.aspx>.
- Project, Trac Open Source. 2008.** The Trac Project. *The Trac Project*. [Online] Octubre 2008. <http://trac.edgewall.org/>.
- Ramírez Mastrap, Yudenia. 2007.** *LA GESTIÓN DE ERRORES EN EL DESARROLLO DE SOFTWARE*. 2007.
- Reynolds-Haertle, Robin. 2002.** Tips and Tricks: Building Microsoft Office Add-ins with Visual C# .NET and Visual Basic .NET. *MSDN*. [Online] Septiembre 2002. <http://msdn.microsoft.com/en-us/library/aa289518.aspx>.
- Ronda Amador, Yoelys, Cobas Santos, Kadir and Marrero Viñas, Daymel. 2002.** *El modelo cliente-servidor de tres capas para el trabajo en redes de información*. 2002.
- Sánchez González, Carlos. 2004.** *ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras*. Da Coruña : s.n., 2004.
- Shaikh, Munir. 2007.** Word to HTML Converter using ASP.Net 2.0 and Microsoft 11.0 object Library. *C# Corner*. [Online] Marzo 2007. <http://www.c-sharpcorner.com/UploadFile/munnamax/WordToHtml03252007065157AM/WordToHtml.aspx>.
- Shehzad, Khurram. 2008.** Automatic Reading and Verification of Microsoft Word document in C# using Aspose.Words. *C# Corner*. [Online] Enero 2008. http://www.c-sharpcorner.com/UploadFile/shehzadgee/Automatic_verification_of_Microsoft_Word_Docum

ent_using_Csharp01082008142038PM/Automatic_verification_of_Microsoft_Word_Document_using_Csharp.aspx.

Sourceforge.net. 2008. CodeTrack: Bug Reporting and Tracking for the Rest of Us! [Online] Octubre 2008. <http://kennwhite.sourceforge.net/codetrack>.

Sparxsystems.com. 2006. Enterprise Architect - Herramienta de diseño UML. [Online] 2006. <http://www.sparxsystems.com.ar/products/ea.html>.

Subgurim.NET. 2006. ¿Por qué ASP.NET? [Online] Febrero 2006. <http://www.subgurim.net/Articulos/asp-net-general/3/por-que-asp-net.aspx>.

Support, Microsoft Help and. 2006. Cómo utilizar el control WebBrowser para abrir un documento de Office en Visual C# 2005 o en Visual C# .NET. *Microsoft Help and Support*. [Online] Octubre 2006. <http://support.microsoft.com/kb/304662/es>.

—. **2007.** How to build an Office COM add-in by using Visual C# .NET. *Microsoft Help and Support*. [Online] Mayo 2007. <http://support.microsoft.com/Default.aspx?kbid=302901>.

—. **2007.** Office 2003 Update: Redistributable Primary Interop Assemblies está disponible para descarga. [Online] Mayo 2007. <http://support.microsoft.com/kb/897646/es>.

Valencia, Universidad Politécnica de. 2001. Introducción a Herramientas CASE y System Architect. *Universidad Politécnica de Valencia*. [Online] Abril 2001. http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.

—. **2007.** Rational Unified Process (RUP). *Universidad Politécnica de Valencia*. [Online] Mayo 2007. <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>.

Valparaiso, Universidad de. 2006. Evaluación del lenguaje. *Educnet-Universidad de Valparaíso*. [Online] Febrero 2006. <http://educnet.decom-uv.cl/educnet/uploads/Implementacion.pdf?nombre=p379/Implementacion.pdf>.

Vegas, Jesús. 2002. El Servidor Web. *Departamento de Informática (Universidad de Valladolid)*. [Online] Marzo 2002. <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.

Vico.org. 2002. Unified Modeling Language. *Vico.org*. [Online] 2002. <http://www.vico.org/FormMentorOutsourcingUML.pdf>.

Villaverde Masa, Pablo. 2007. ASP. Recursos y características. [Online] Noviembre 2007. http://tgp0607.awardspace.com/Recursos_ASP.pdf.

Zamora, Universidad Nacional de Lomas de. 2008. ¿Qué es Internet? [Online] Diciembre 2008. <http://www.unlz.edu.ar/biblioteca/tutores/histoweb/datosgral.html>.

Zentrack.net. 2009. Zentrack. [Online] Abril 2009. <http://www.zentrack.net>.

ANEXOS

Anexo 1: Diagrama de Actividades

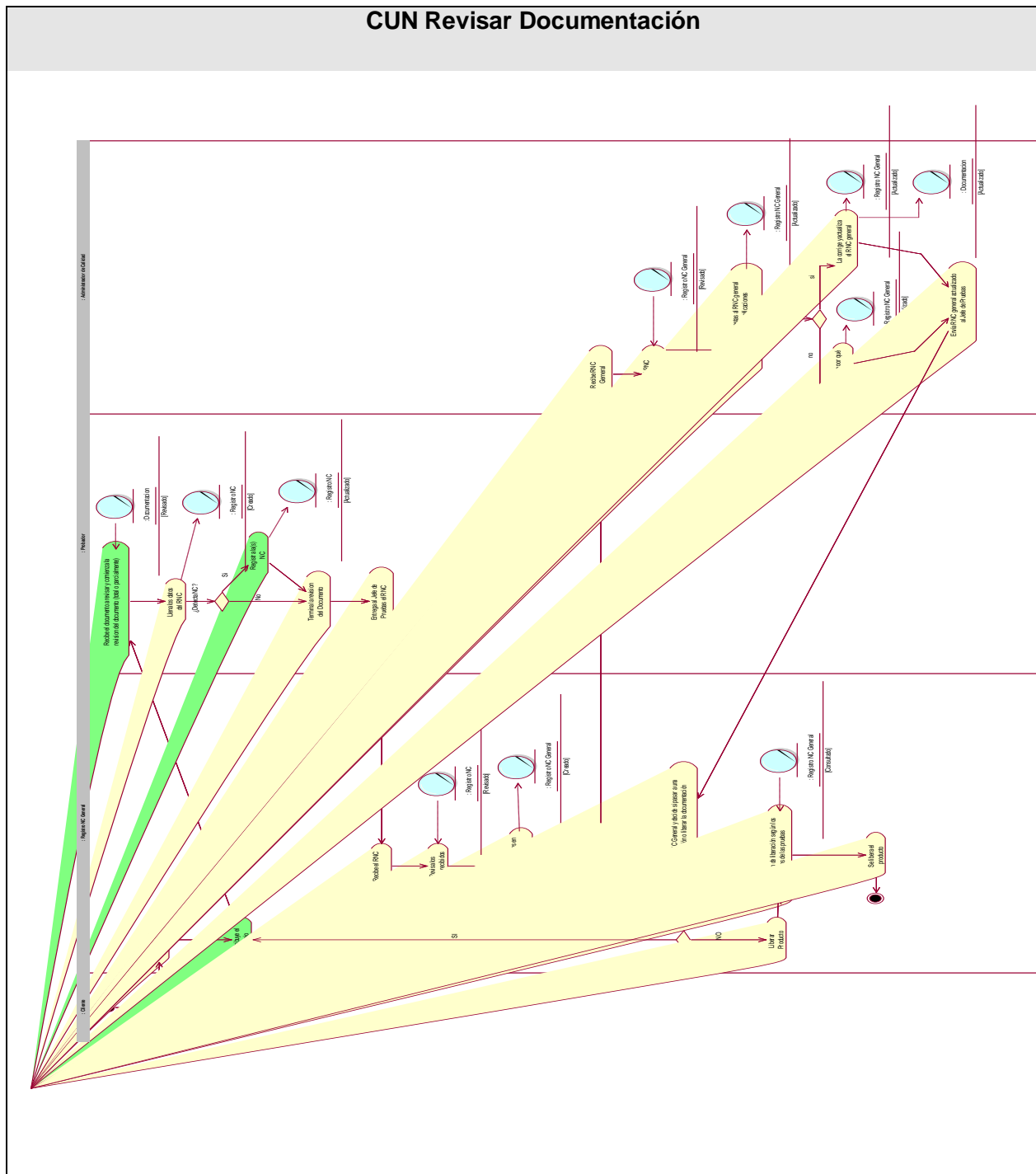


Fig. A1.1 -Diagrama de actividades <CUN Revisar Documentación >.

Anexo 2: Diagrama de Clases del Modelo de Objetos

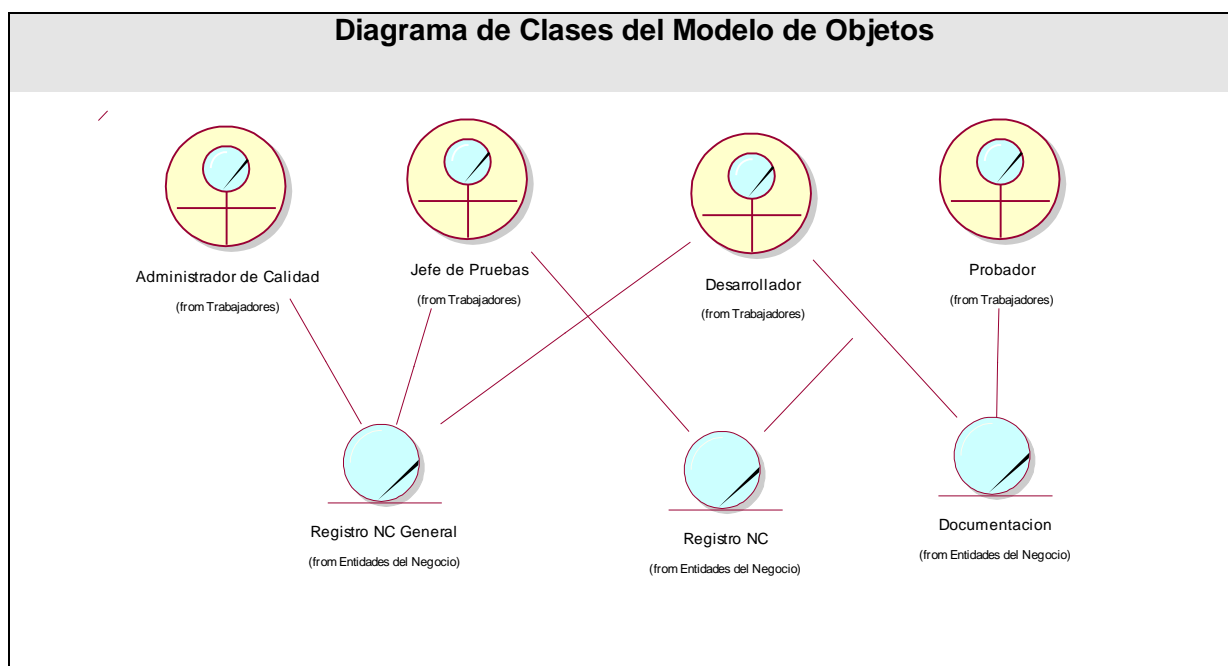


Fig. A2.1 -Diagrama de Clases del Modelo de Objetos.

Anexo 3: Descripción expandida de los Casos de Uso

CU Autenticar

Caso de uso	
CU#1	CU Autenticar
Propósito	Permitir o denegar el acceso a la aplicación.
Actores: Usuario	
Resumen: EL Caso de Uso se inicializa cuando el usuario decide autenticarse, para ello ingresa su identificador y la contraseña, el sistema determina si puede acceder o no a la aplicación. El Caso de Uso finaliza cuando el usuario se autentica en el sistema.	
Referencias	RF 1
Acción del actor	Respuesta del sistema
	1. Muestra la vista para la realización de la autenticación: <ul style="list-style-type: none"> ○ Identificador

	<ul style="list-style-type: none"> ○ Contraseña
2. El usuario introduce su identificador y la contraseña correspondiente. Oprime el botón "Entrar".	<p>3. Verifica que todos los datos hallan sido proporcionados.</p> <p>4. Verifica la existencia del usuario y la correspondencia con la contraseña suministrada.</p> <p>5. Permite el acceso a la aplicación acorde a los permisos que posee el usuario autenticado.</p>
Flujo Alternativo 1	
Acción del actor	Respuesta del sistema
	3. Se muestra un mensaje informando que faltan datos.
4. Acepta el mensaje.	
Flujo Alternativo 1	
Acción del actor	Respuesta del sistema
	4. Muestra un mensaje de error informando de un error en la autenticación.
5. Acepta el mensaje	6. Permite volver a realizar la operación.

Tabla A3.1 -.Descripción expandida del CU Autenticación

CU Gestionar Usuarios

Caso de uso	
CU#2	CU Gestionar Usuarios
Propósito	Permitir realizar la gestión de los usuarios.

Actores: Administrador de Pruebas	
Resumen: EL Caso de Uso se inicializa cuando el Administrador de Pruebas decide realizar acciones como agregar, modificar o eliminar un usuario, además puede ver los detalles de un usuario previamente seleccionado. EL Caso de Uso finaliza cuando se culmina la acción seleccionada previamente (agregar, eliminar, modificar).	
Referencias	RF 2
Acción del actor	Respuesta del sistema
1. El Administrador de Pruebas escoge la opción "Usuarios".	2. Muestra la vista "Gestionar Datos del Usuario" con un listado de los usuarios existentes.
3. Decide realizar una de las acciones: a) Agregar un nuevo usuario (Ver Sección "Agregar Usuario"). b) Modificar un usuario (Ver Sección "Modificar Datos del Usuario"). c) Eliminar un usuario (Ver Sección "Eliminar Usuario").	
Sección "Agregar Usuario"	
Acción del actor	Respuesta del sistema
	1. Muestra los campos para la inserción de los datos necesarios: <ul style="list-style-type: none"> ○ Nombre(s) ○ Apellidos(s) ○ Usuario ○ Responsabilidad ○ Usuario de correo

	<ul style="list-style-type: none"> ○ Servidor de Correo ○ Contraseña <p>Datos de un miembro del Equipo de Desarrollo</p> <ul style="list-style-type: none"> ○ Área Temática ○ Producto ○ Módulo
2. Entra los datos especificados y escoge la opción "Aceptar".	<p>2.1. Verifica que los datos necesarios están completos y son correctos.</p> <p>2.2. El usuario nuevo es agregado a los usuarios del sistema.</p> <p>2.3. Se muestra un mensaje informando del éxito de la operación.</p> <p>2.4. Los campos de entrada de datos se muestran vacíos.</p>
Flujo Alternativo 1	
	2.1. Muestra un mensaje informando del error en los datos proporcionados.
3. Acepta el mensaje	4. Permite volver a realizar la operación
Flujo Alternativo 2	
2. Entra los datos especificados y escoge la opción "Cancelar".	<p>2.1. El usuario nuevo no es agregado.</p> <p>2.2. Los campos de entrada de datos se muestran limpios.</p>
Sección "Modificar Datos del Usuario"	
Acción del actor	Respuesta del sistema

<p>1. Selecciona un usuario del listado.</p>	<p>2. Muestra los datos del usuario seleccionado.</p> <ul style="list-style-type: none"> ○ Nombre(s) ○ Apellidos(s) ○ Usuario ○ Responsabilidad ○ Usuario de correo ○ Servidor de Correo ○ Contraseña <p>Datos de un miembro del Equipo de Desarrollo</p> <ul style="list-style-type: none"> ○ Área Temática ○ Producto ○ Módulo
<p>3. Modifica los datos y escoge la opción “Modificar”.</p>	<p>3.1. Verifica que los datos necesarios están completos y son correctos.</p> <p>3.2. Los datos del usuario son modificados.</p> <p>3.3. Se muestra un mensaje informando del éxito de la operación.</p> <p>3.4. Los campos de entrada de datos se muestran vacíos.</p>
<p>Flujo Alternativo 1</p>	
	<p>3.1. Muestra un mensaje informando del error en los datos proporcionados.</p>
<p>4. Acepta el mensaje</p>	<p>5. Permite volver a realizar la operación</p>
<p>Flujo Alternativo 2</p>	

3. Modifica los datos y escoge la opción "Cancelar".	3.1. Los datos del usuario no son modificados. 3.2. Los campos de entrada de datos se muestran limpios.
Sección "Eliminar Usuario"	
Acción del actor	Respuesta del sistema
1. Selecciona un usuario del listado.	2. Muestra los datos del usuario seleccionado. <ul style="list-style-type: none"> ○ Nombre(s) ○ Apellidos(s) ○ Usuario ○ Responsabilidad ○ Usuario de correo ○ Servidor de Correo ○ Contraseña <p>Datos de un miembro del Equipo de Desarrollo</p> <ul style="list-style-type: none"> ○ Área Temática ○ Producto ○ Módulo
3. Escoge la opción "Eliminar".	4. Muestra un mensaje para confirmar la acción.
5. En el mensaje escoge "Aceptar".	6. El usuario es eliminado del sistema.
Flujo Alternativo 1	
3. En el mensaje escoge "Cancelar".	4. El usuario no es eliminado del sistema.

Tabla A3.2 -.Descripción expandida del CU Gestionar Usuarios

CU Gestionar Elementos del Expediente de Proyecto

Caso de uso	
CU#3	CU Gestionar Elementos del Expediente de Proyecto
Propósito	Permitir realizar la gestión de los elementos del Expediente de Proyecto.
Actores: Administrador de Pruebas	
Resumen: EL Caso de Uso se inicializa cuando el Administrador de Pruebas decide realizar acciones como agregar, modificar o eliminar un elemento, además puede ver los detalles de un elemento previamente seleccionado. EL Caso de Uso finaliza cuando se culmina la acción seleccionada previamente (agregar, eliminar, modificar).	
Referencias	RF 3
Acción del actor	Respuesta del sistema
1. El Administrador de Pruebas escoge la opción "Expediente de Proyecto".	2. Muestra la vista "Gestionar Elementos del Expediente de Proyecto" con un listado de los elementos existentes.
3. Decide realizar una de las acciones: d) Agregar un nuevo elemento (Ver Sección "Agregar Elemento"). e) Modificar un elemento (Ver Sección "Modificar Datos del Elemento"). f) Eliminar un elemento (Ver Sección "Eliminar Elemento").	
Sección "Agregar Elemento"	
Acción del actor	Respuesta del sistema
	3. Muestra los campos para la inserción de los datos necesarios: ○ Tipo del Documento

	<ul style="list-style-type: none"> ○ Nombre del Documento
4. Entra los datos especificados y escoge la opción "Aceptar".	<p>4.1. Verifica que los datos necesarios están completos y son correctos.</p> <p>4.2. El elemento nuevo es agregado a los elemento del sistema.</p> <p>4.3. Se muestra un mensaje informando del éxito de la operación.</p> <p>4.4. Los campos de entrada de datos se muestran vacíos.</p>
Flujo Alterno 1	
	4.1. Muestra un mensaje informando del error en los datos proporcionados.
5. Acepta el mensaje	6. Permite volver a realizar la operación
Flujo Alterno 2	
4. Entra los datos especificados y escoge la opción "Cancelar".	<p>5.1. El elemento nuevo no es agregado.</p> <p>5.2. Los campos de entrada de datos se muestran limpios.</p>
Sección "Modificar Datos del Elemento"	
Acción del actor	Respuesta del sistema
1. Selecciona un elemento del listado.	<p>2. Muestra los datos del elemento seleccionado.</p> <ul style="list-style-type: none"> ○ Tipo del Documento ○ Nombre del Documento
3. Modifica los datos y escoge la opción "Modificar".	<p>3.1. Verifica que los datos necesarios están completos y son correctos.</p> <p>3.2. Los datos del elemento son modificados.</p>

	<p>3.3. Se muestra un mensaje informando del éxito de la operación.</p> <p>3.4. Los campos de entrada de datos se muestran vacíos.</p>
Flujo Alternativo 1	
	3.1. Muestra un mensaje informando del error en los datos proporcionados.
4. Acepta el mensaje	5. Permite volver a realizar la operación
Flujo Alternativo 2	
3. Modifica los datos y escoge la opción "Cancelar".	<p>4.1. Los datos del elemento no son modificados.</p> <p>4.2. Los campos de entrada de datos se muestran limpios.</p>
Sección "Eliminar Elemento"	
Acción del actor	Respuesta del sistema
1. Selecciona un elemento del listado.	<p>2. Muestra los datos del elemento seleccionado.</p> <ul style="list-style-type: none"> ○ Tipo del Documento ○ Nombre del Documento
3. Escoge la opción "Eliminar".	4. Muestra un mensaje para confirmar la acción.
5. En el mensaje escoge "Aceptar".	6. El elemento es eliminado del sistema.
Flujo Alternativo 1	
5. En el mensaje escoge "Cancelar".	6. El elemento no es eliminado del sistema.

Tabla A3.3 -.Descripción expandida del CU Gestionar Expediente

CU Gestionar Revisiones

Caso de uso

CU#4	CU Gestionar Revisiones	
Propósito	Permitir realizar la gestión de las revisiones.	
Actores: Administrador de Pruebas		
Resumen: El Caso de Uso se inicia cuando el Administrador de Pruebas desea realizar acciones como agregar, modificar o eliminar una revisión, permitiendo además ver los detalles de una revisión previamente seleccionada.		
Referencias	RF 4	
Acción del actor	Respuesta del sistema	
1. Escoge la opción "Revisiones".	2. Muestra la vista "Gestionar Revisiones" con un listado de las revisiones existentes.	
3. Decide realizar una de las acciones: a) Agregar una nueva revisión (Ver Sección "Agregar Revisión"). b) Modificar una revisión (Ver Sección "Modificar Datos de la Revisión"). c) Eliminar una revisión (Ver Sección "Eliminar Revisión").		
Sección "Agregar Revisión"		
Acción del actor	Respuesta del sistema	
	1. Muestra los campos para la inserción de los datos necesarios: <ul style="list-style-type: none"> ○ Área Temática ○ Nombre del Producto ○ Nombre del Módulo ○ Estado de la Revisión 	

	<ul style="list-style-type: none"> ○ Elementos de la Revisión
2. Entra los datos especificados y escoge la opción "Aceptar".	<p>2.1. Verifica que los datos necesarios están completos y son correctos.</p> <p>2.2. La nueva revisión es agregada al sistema.</p> <p>2.3. Se muestra un mensaje informando del éxito de la operación.</p> <p>2.4. Los campos de entrada de datos se muestran vacíos.</p>
Flujo Alternativo 1	
	2.1. Muestra un mensaje informando del error en los datos proporcionados.
3. Acepta el mensaje	4. Permite volver a realizar la operación
Flujo Alternativo 2	
2. Entra los datos especificados y escoge la opción "Cancelar".	<p>2.1. La nueva revisión no es agregada.</p> <p>2.2. Los campos de entrada de datos se muestran limpios.</p>
Sección "Modificar Datos de la Revisión"	
Acción del actor	Respuesta del sistema
1. Selecciona una revisión del listado.	<p>2. Muestra los datos de la revisión seleccionada.</p> <ul style="list-style-type: none"> ○ Área Temática ○ Nombre del Producto ○ Nombre del Módulo ○ Estado de la Revisión ○ Elementos de la Revisión

<p>3. Modifica los datos y escoge la opción "Aceptar".</p>	<p>3.1. Los datos de la revisión son modificados.</p> <p>3.2. Se muestra un mensaje informando del éxito de la operación.</p> <p>3.3. Los campos de entrada de datos se muestran limpios.</p>
<p>Flujo Alternativo 1</p>	
	<p>3.1. Muestra un mensaje informando del error en los datos proporcionados.</p>
<p>4. Acepta el mensaje</p>	<p>5. Permite volver a realizar la operación</p>
<p>Flujo Alternativo 2</p>	
<p>3. Modifica los datos y escoge la opción "Cancelar".</p>	<p>3.1. Los datos de la revisión no son modificados.</p> <p>3.2. Los campos de entrada de datos se muestran limpios.</p>
<p>Sección "Eliminar Revisión"</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. Selecciona una revisión del listado.</p>	<p>2. Muestra los datos de la revisión seleccionada.</p> <ul style="list-style-type: none"> ○ Área Temática ○ Nombre del Producto ○ Nombre del Módulo ○ Estado de la Revisión ○ Elementos de la Revisión
<p>3. Escoge la opción "Eliminar".</p>	<p>4. Muestra un mensaje para confirmar la acción.</p>
<p>5. En el mensaje escoge "Aceptar"</p>	<p>6. La revisión es eliminada del sistema.</p>
<p>Flujo Alternativo 1</p>	

3. En el mensaje escoge "Cancelar".	4. La revisión no es eliminada del sistema.
-------------------------------------	---

Tabla A3.4 -.Descripción expandida del CU Gestionar Revisión

CU Subir Documentación

CU#5	CU Subir Documentación	
Propósito	Permitir al Administrador de Calidad que la documentación de un producto a revisar sea puesta a disposición del Equipo de Pruebas.	
Actores: Administrador de Calidad		
Resumen: EL Caso de Uso se inicializa cuando el Administrador de Calidad decide poner la documentación de un producto a revisar a disposición del Equipo de Pruebas. El Administrador de Calidad selecciona la ubicación donde se encuentran los documentos, la aplicación los guarda en una carpeta donde el Equipo de Pruebas los tiene a su completa disposición.		
Referencias	RF 5	
Acción del actor	Respuesta del sistema	
1. Escoge la opción "Subir Documentos".	2. Muestra la vista "Subir Documentación".	
3. Busca los documentos en su entorno de trabajo y selecciona el (los) documento(s) para poner disponible(s). 4. Escoge la opción "Aceptar".	5. El (los) documento(s) se guarda(n) en una carpeta previamente definida por el Equipo de Pruebas.	

Tabla A3.5 -.Descripción expandida del CU Subir Documentación

CU Gestionar Tareas

Caso de uso	
CU#6	CU Gestionar Tareas
Propósito	Permitir realizar la gestión de las tareas de los desarrolladores y los probadores por parte de sus respectivos líderes.

Actores: Usuario Avanzado (Jefe de Pruebas, Administrador de Calidad)	
Resumen: El Caso de Uso se inicializa cuando el Usuario Avanzado desea realizar acciones como agregar, modificar o eliminar una tarea. Las tareas son asignadas a probadores o desarrolladores en dependencia de quien realiza la distribución (Administrador de Calidad, Jefe de Pruebas).	
Referencias	Rf 4.3, RF 6, Rf 9, RF 10
Acción del actor	Respuesta del sistema
1. Escoge la opción "Tareas".	<ul style="list-style-type: none"> a) Si el usuario autenticado es un Jefe de Pruebas ver Sección "Tareas de los Probadores", Sección "Eliminar Tarea a un Probador". b) Si el usuario autenticado es un Administrador de Calidad ver Sección "Asignar Tarea a un Desarrollador" ", Sección "Eliminar Tarea de Desarrollador".
Sección "Tareas de los Probadores".	
Sección "Asignar Tarea a un Probador"	
Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> 1. Muestra la vista "Gestionar Tareas" con un listado de las tareas que han sido asignadas. 2. Muestra todas las revisiones que se encuentran excepto aquellas que han sido terminadas. 3. Muestra todos los usuarios que poseen el rol de Probador.
4. Selecciona la revisión.	5. Se muestran los elementos del Expediente de Proyecto que componen la revisión.
6. Selecciona el usuario que participará en ella y el elemento que revisará, escoge la	<ul style="list-style-type: none"> 7.1. La nueva tarea es registrada en el sistema. 7.2. Se muestra un mensaje informando del éxito

opción "Aceptar".	de la operación.
Flujo Alternativo 1	
	1. Muestra un mensaje informando que no existen tareas para revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 2	
	3. Muestra un mensaje informando que no existen revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 3	
	4. Muestra un mensaje informando que no existen usuarios con el rol de Probador.
Flujo Alternativo 4	
6. Selecciona la revisión y el usuario que participará en ella, escoge la opción "Cancelar".	7.1. La nueva tarea no es agregada.
Sección "Eliminar Tarea a un Probador"	
Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> Muestra todas las tareas que han sido asignadas a revisiones que no han sido terminadas.
<ul style="list-style-type: none"> Selecciona una tarea. Escoge la opción "Eliminar Tarea Asignada". 	<ul style="list-style-type: none"> Muestra los detalles de la tarea seleccionada. Se muestra un mensaje para confirmar la acción.
<ul style="list-style-type: none"> En el mensaje escoge "Aceptar". 	.1. La tarea es eliminada del sistema.

Flujo Alternativo 1	
	1. Muestra un mensaje informando que no existen tareas para revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 2	
5. En el mensaje escoge "Cancelar".	5.1. La tarea no es eliminada del sistema.
Sección "Tareas de los Desarrolladores".	
Sección "Asignar Tarea a un Desarrollador"	
Acción del actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. Muestra la vista "Gestionar Tareas" con un listado de las tareas que han sido asignadas. 2. Muestra todas las revisiones que se encuentran excepto aquellas que han sido terminadas. 3. Muestra todos los usuarios que poseen el rol de Desarrollador.
4. Selecciona la revisión.	5. Se muestran los elementos del Expediente de Proyecto que componen la revisión.
6. Selecciona la revisión y el usuario que participará en ella, escoge la opción "Aceptar".	<ol style="list-style-type: none"> 6.1. La nueva tarea es registrada en el sistema. 6.2. Se muestra un mensaje informando del éxito de la operación.
Flujo Alternativo 1	
	1. Muestra un mensaje informando que no existen tareas para revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 2	

	3. Muestra un mensaje informando que no existen revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 3	
	4. Muestra un mensaje informando que no existen usuarios (con el rol de Desarrollador) asociados a su producto.
Flujo Alternativo 4	
6. Selecciona la revisión y el usuario que participará en ella, escoge la opción "Cancelar".	6.1. La nueva tarea no es agregada.
Sección "Eliminar Tarea a un Desarrollador"	
Acción del actor	Respuesta del sistema
	1. Muestra todas las tareas que han sido asignadas a revisiones que no han sido terminadas.
2. Selecciona una tarea. 3. Escoge la opción "Eliminar Tarea Asignada".	4. Muestra los detalles de la tarea seleccionada. 5. Se muestra un mensaje para confirmar la acción.
6. En el mensaje escoge "Aceptar".	7. La tarea es eliminada del sistema.
Flujo Alternativo 1	
	1. Muestra un mensaje informando que no existen tareas para revisiones en estado de ejecución o comenzadas.
Flujo Alternativo 2	
5. En el mensaje escoge "Cancelar".	5.1. La tarea no es eliminada del sistema.

Tabla A3.6 -.Descripción expandida del CU Gestionar Tareas

CU Gestionar Reportes

CU#9	CU Gestionar Reportes	
Propósito	Permitir gestionar un reporte con las No Conformidades detectadas en un proceso de pruebas.	
Actores:	Administrador de Pruebas, Usuario Avanzado (Administrador de Calidad, Jefe de Pruebas)	
Resumen:	El Caso de Uso se inicializa cuando uno de los actores necesita visualizar un reporte con las estadísticas de las No Conformidades. El Caso de Uso termina cuando se muestran los resultados.	
Referencias	RF 8, RF 9, RF 10, RF 11	
Acción del actor	Respuesta del sistema	
<p>1. Escoge obtener el reporte por diferentes criterios:</p> <p>a) Por elementos de una Revisión (Ver Sección “Reporte por elementos de una Revisión”).</p> <p>b) Por iteración (Ver Sección “Reporte por iteración”).</p> <p>c) Reporte completo de una Revisión (Ver Sección “Reporte de una Revisión”).</p>		
Sección “Reporte por elementos de una Revisión”		
Acción del actor	Respuesta del sistema	
	1. Muestra las revisiones existentes en las que participa el usuario.	
2. Escoge la revisión para obtener el listado.	3. Muestra los elementos de la revisión	

	seleccionada.
4. Selecciona el elemento. 5. Escoge la opción "Generar Reporte".	6. Genera el reporte solicitado.
Flujo Alterno 1	
	1. Muestra un mensaje informando que no existen revisiones.
Flujo Alterno 2	
	6. Muestra un mensaje informando que no existen datos para generar el reporte solicitado.
Sección "Reporte por iteración"	
Acción del actor	Respuesta del sistema
	1. Muestra las revisiones existentes en las que participa el usuario.
2. Escoge la revisión para obtener el listado.	3. Genera el reporte solicitado con los datos ordenados por iteraciones.
Flujo Alterno 1	
	1. Muestra un mensaje informando que no existen revisiones.
Flujo Alterno 2	
	3. Muestra un mensaje informando que no existen datos para generar el reporte solicitado.
Sección "Reporte de una Revisión"	
Acción del actor	Respuesta del sistema

	1. Muestra las revisiones existentes en las que participa el usuario.
2. Escoge la revisión para obtener el listado.	3. Genera el reporte solicitado con los datos ordenados por iteraciones y por elementos.
Flujo Alternativo 1	
	1. Muestra un mensaje informando que no existen revisiones.
Flujo Alternativo 2	
	3. Muestra un mensaje informando que no existen datos para generar el reporte solicitado.

Tabla A3.6 -.Descripción expandida del CU Gestionar Reportes

Anexo 4: Diagramas de Clases del Análisis

DIAGRAMA DE CLASES DE ANÁLISIS (DCA) CU AUTENTICAR

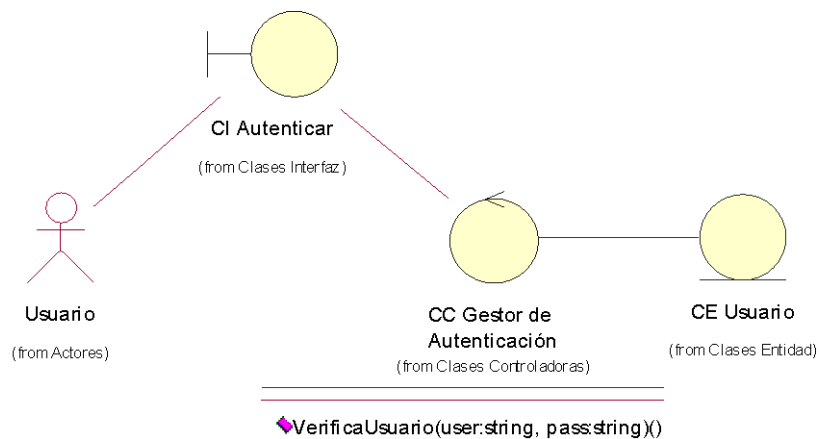


Fig. A4.1 -.DCA CU Autenticar

DIAGRAMA DE CLASES DE ANÁLISIS (DCA) CU GESTIONAR USUARIOS

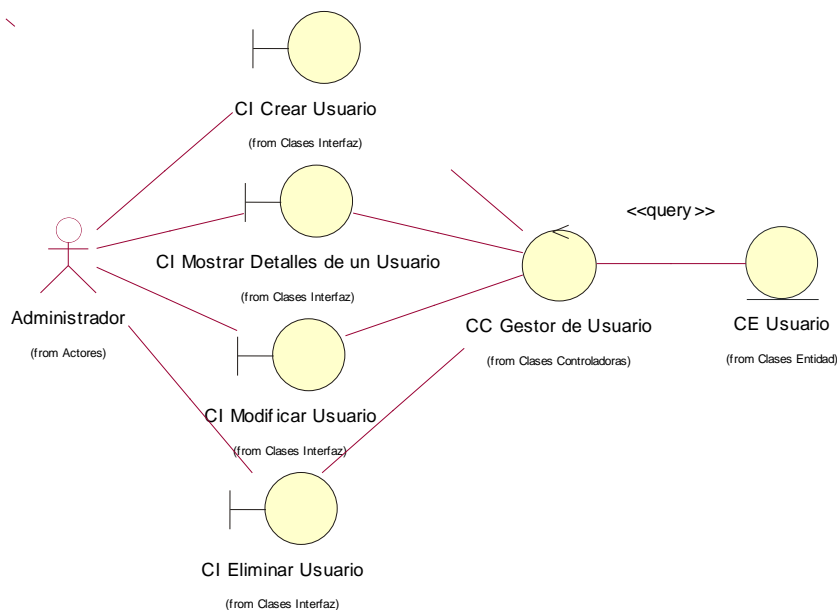


Fig. A4.2 -.DCA CU Gestionar Usuarios

DIAGRAMA DE CLASES DE ANÁLISIS (DCA) CU GESTIONAR ELEMENTOS DE EXPEDIENTE DE PROYECTO

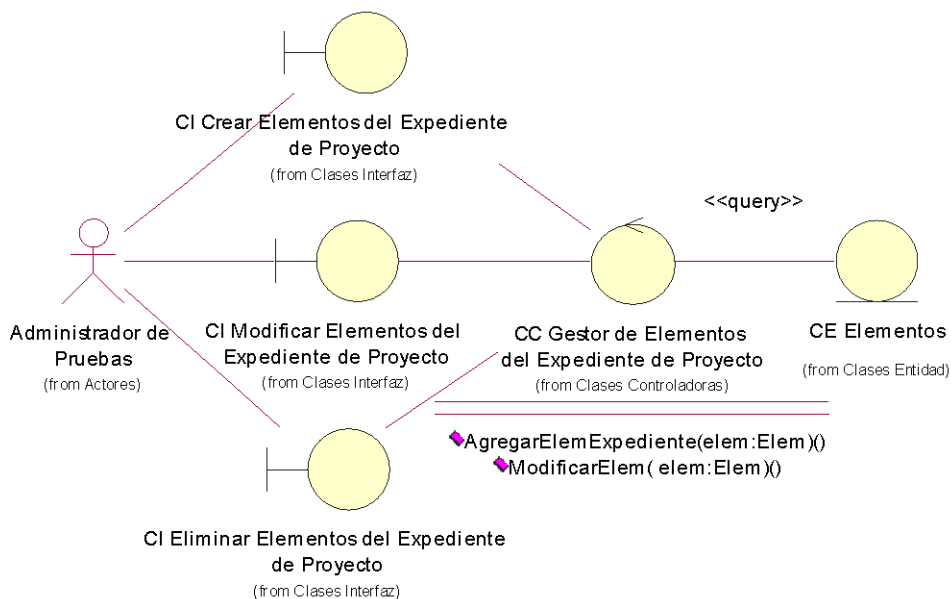


Fig. A4.3 -.DCA CU Gestionar Elementos del Expediente de Proyecto

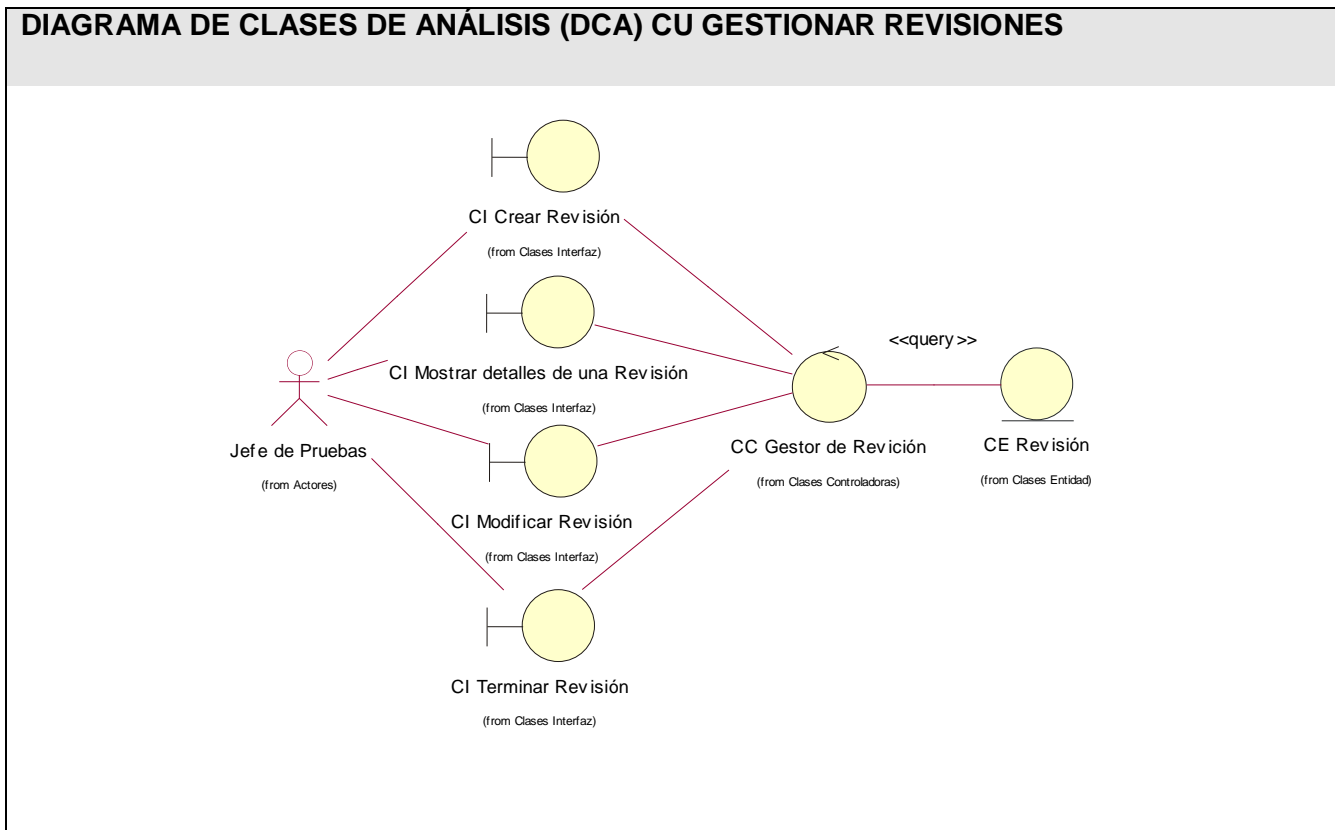


Fig. A4.4 -.DCA CU Gestionar Revisiones

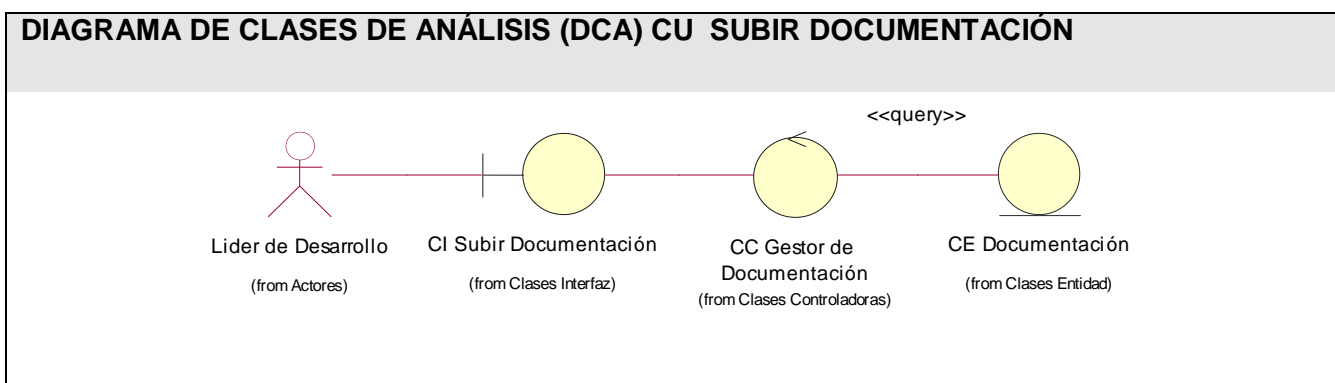


Fig. A4.5 -.DCA CU Subir Documentación

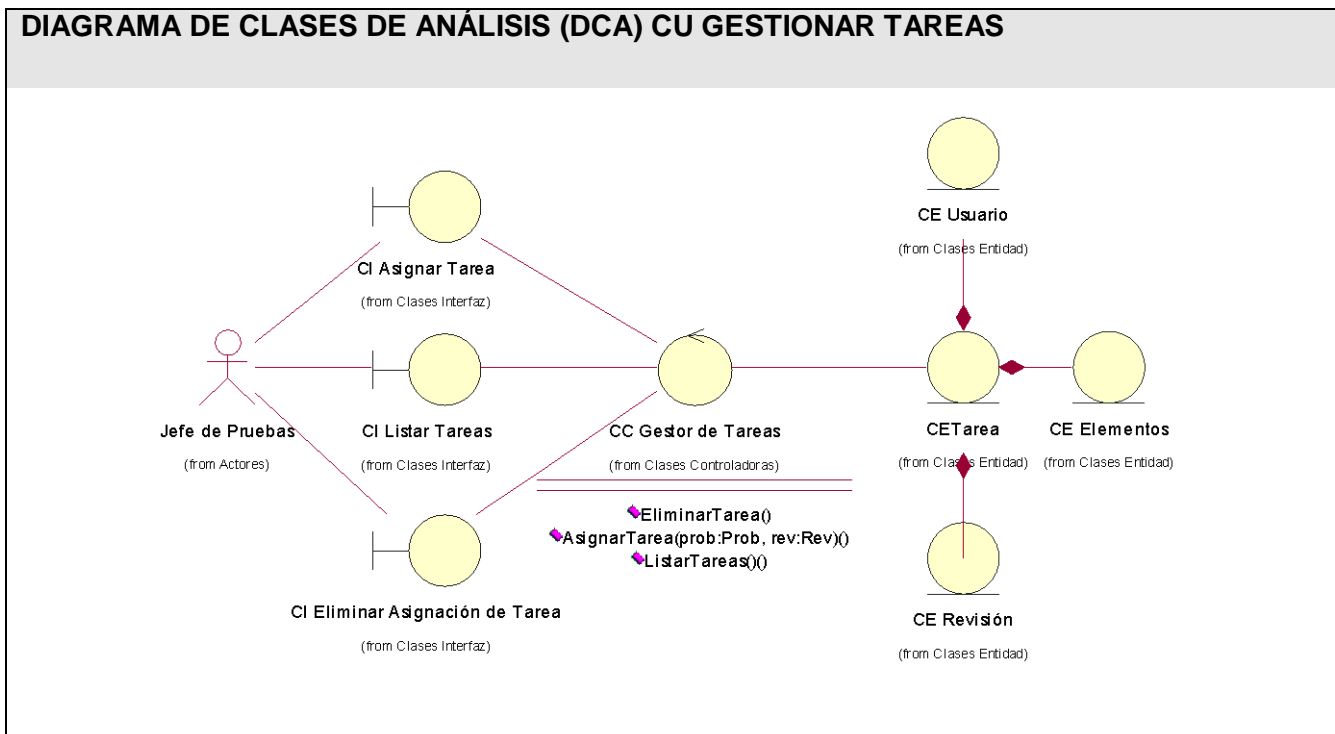


Fig. A4.6 -.DCA CU Gestionar Tareas

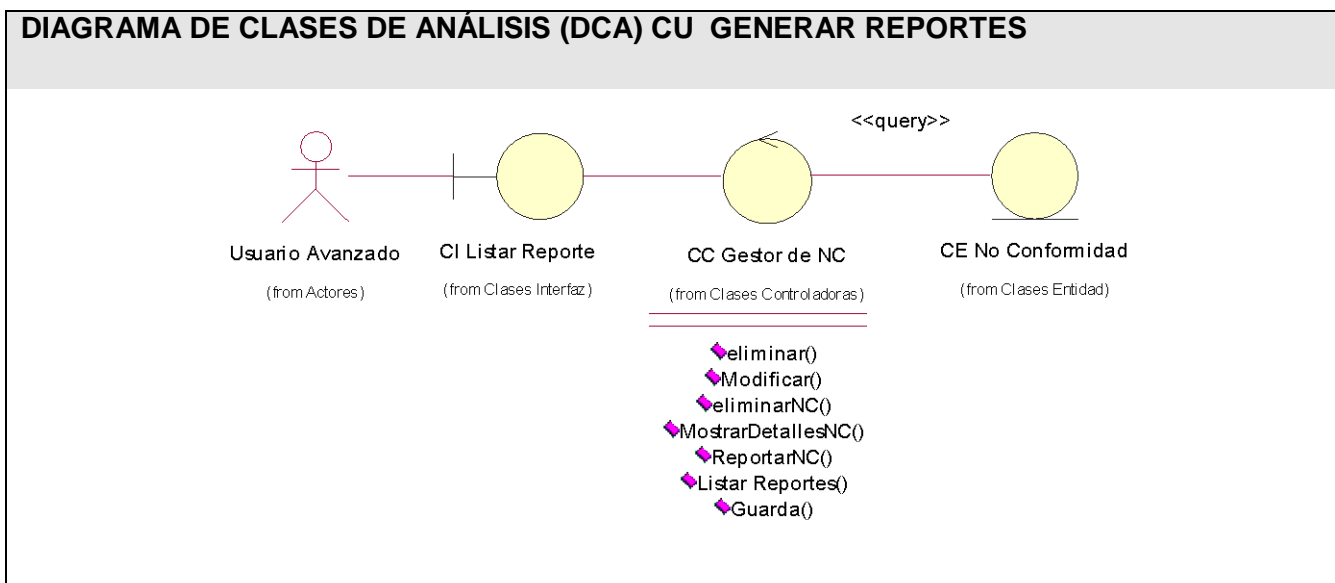


Fig. A4.7 -.DCA CU Gestionar Reportes

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU AUTENTICAR

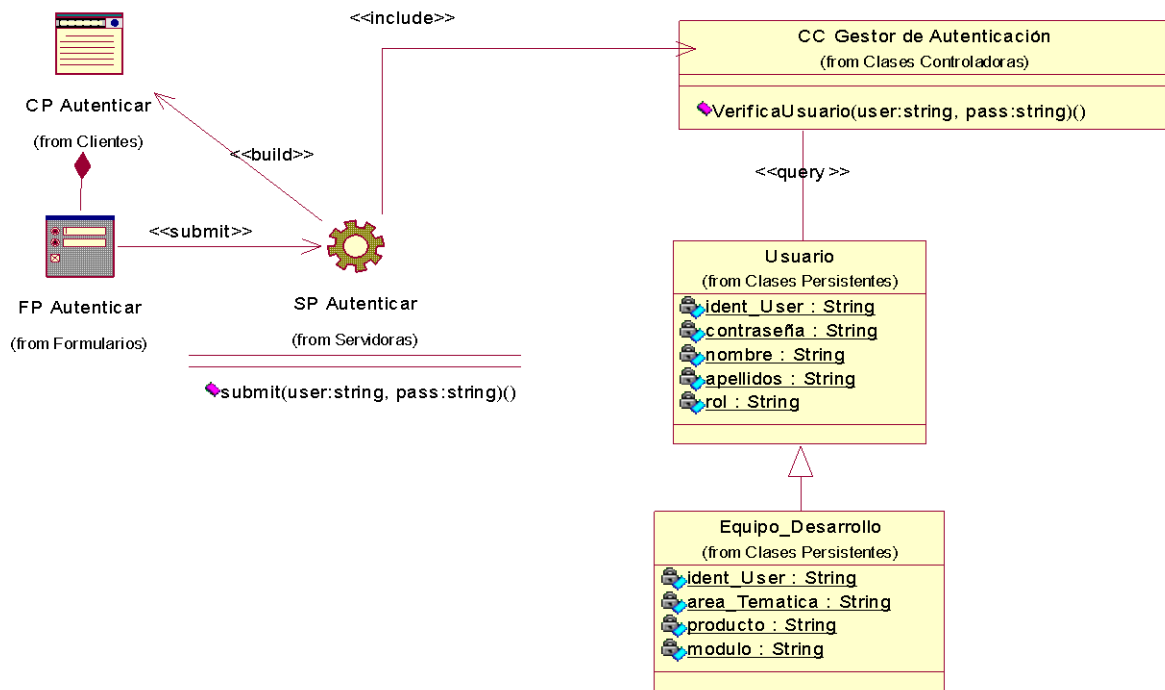


Fig. A5.1 -.DCD CU Autenticar

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GESTIONAR USUARIOS

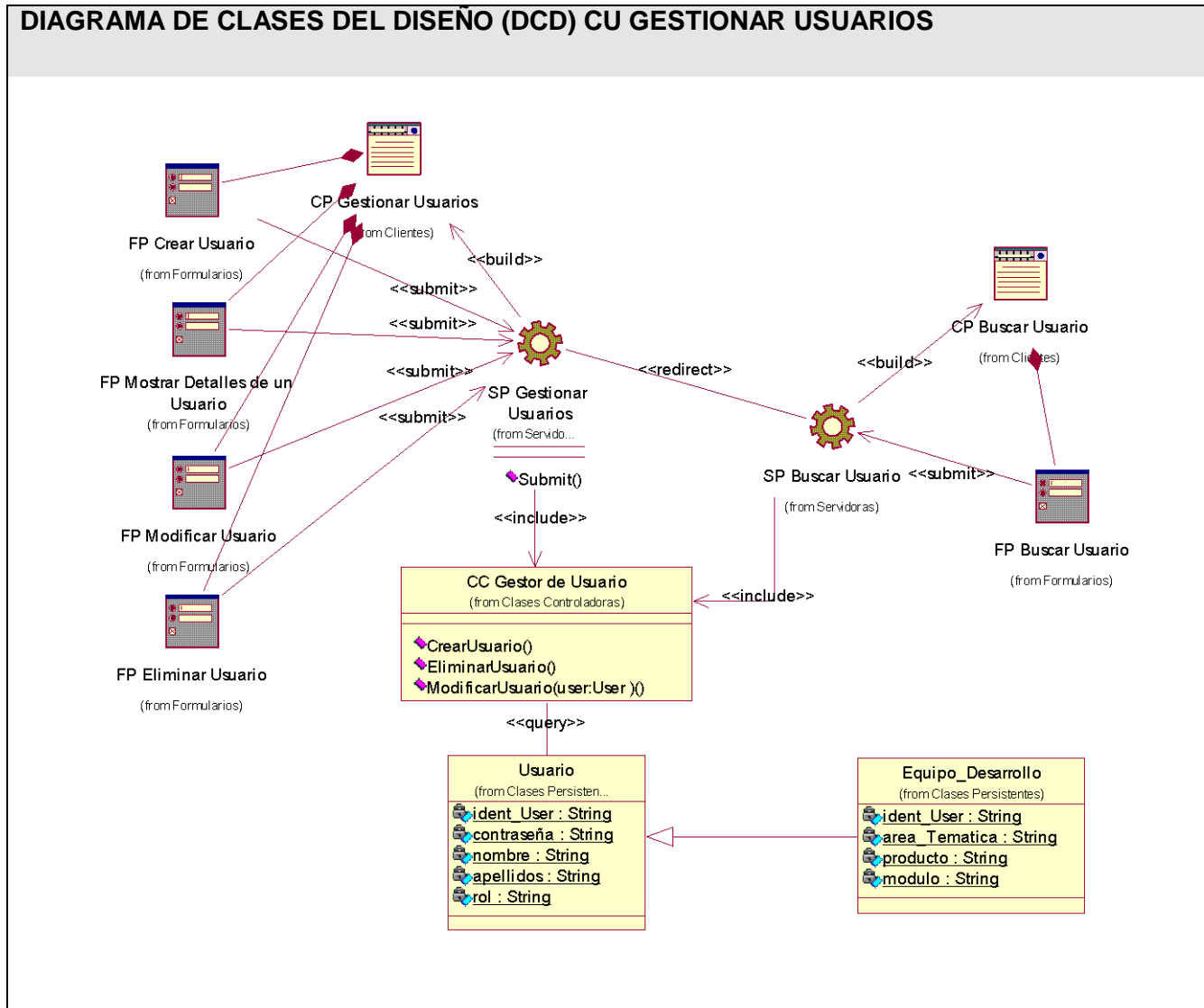


Fig. A5.2 -.DCD CU Gestionar Usuarios

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GESTIONAR ELEMENTOS DEL EXPEDIENTE DE PROYECTO

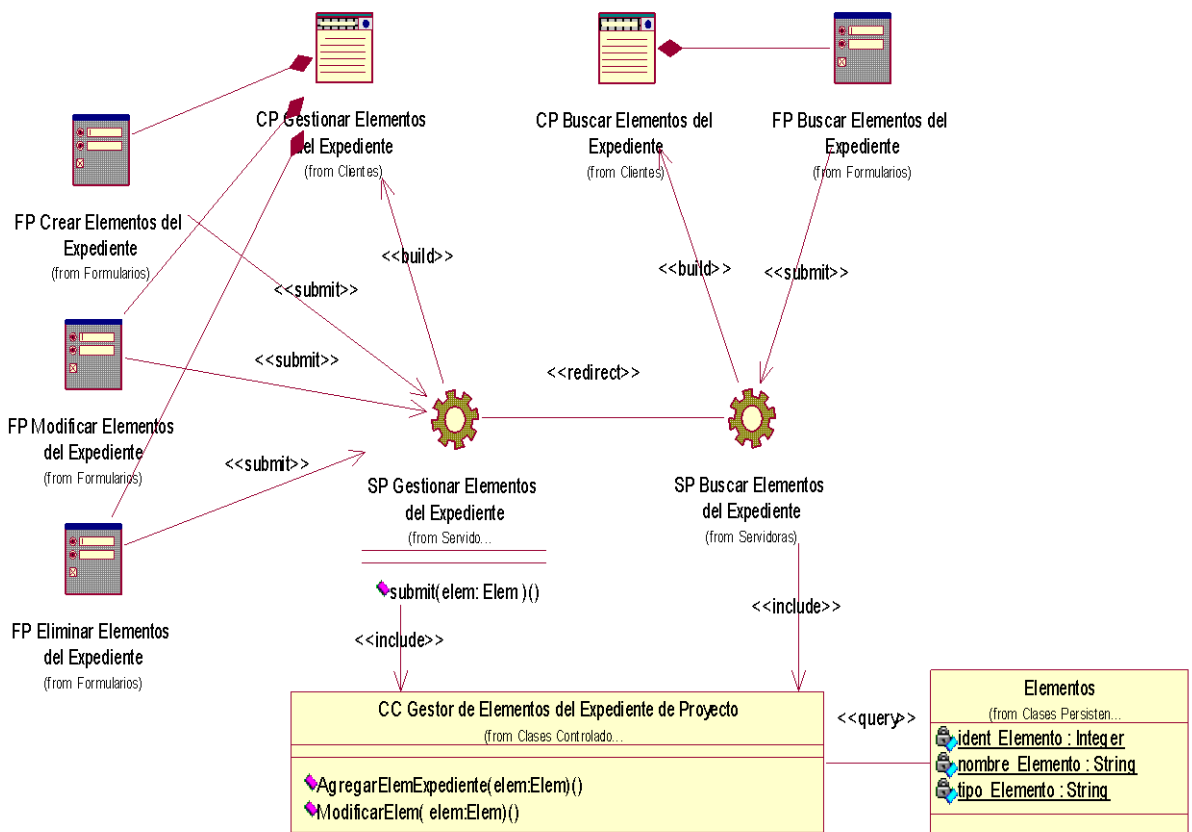


Fig. A5.3 -.DCD CU Gestionar Elementos del Expediente de Proyecto

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GESTIONAR REVISIONES

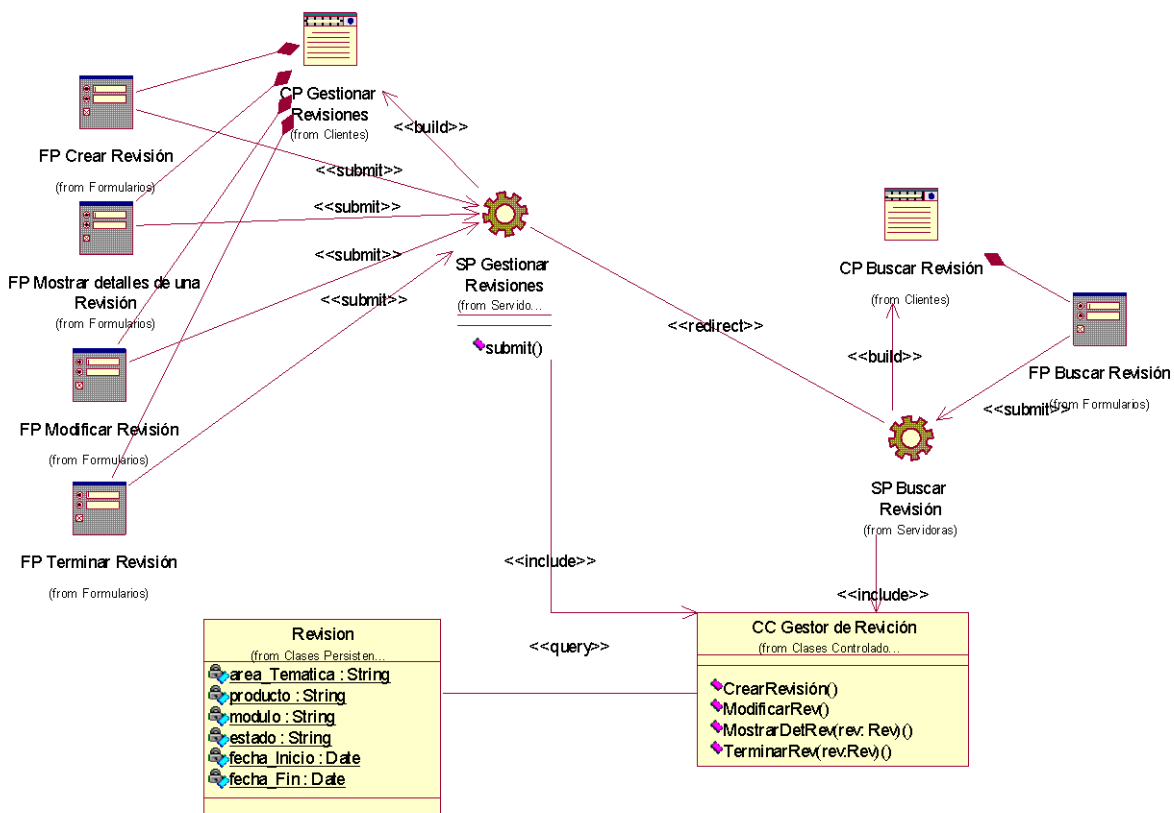


Fig. A5.4 -.DCD CU Gestionar Revisiones

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU SUBIR DOCUMENTACIÓN

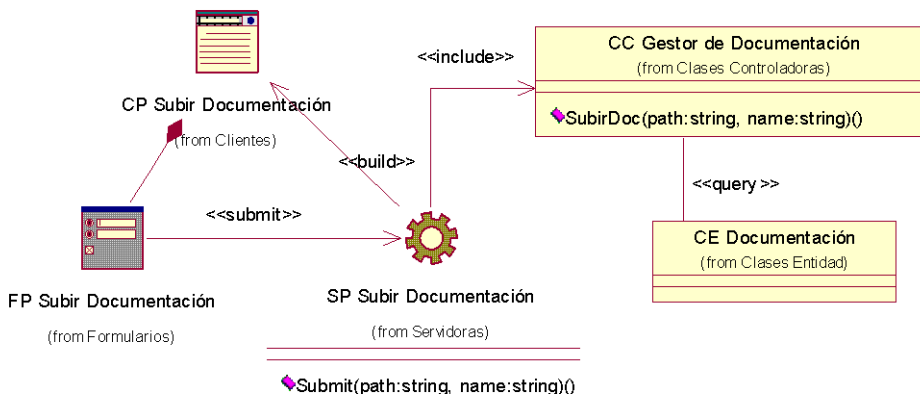


Fig. A5.5 -.DCD CU Subir Documentación

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GESTIONAR TAREAS

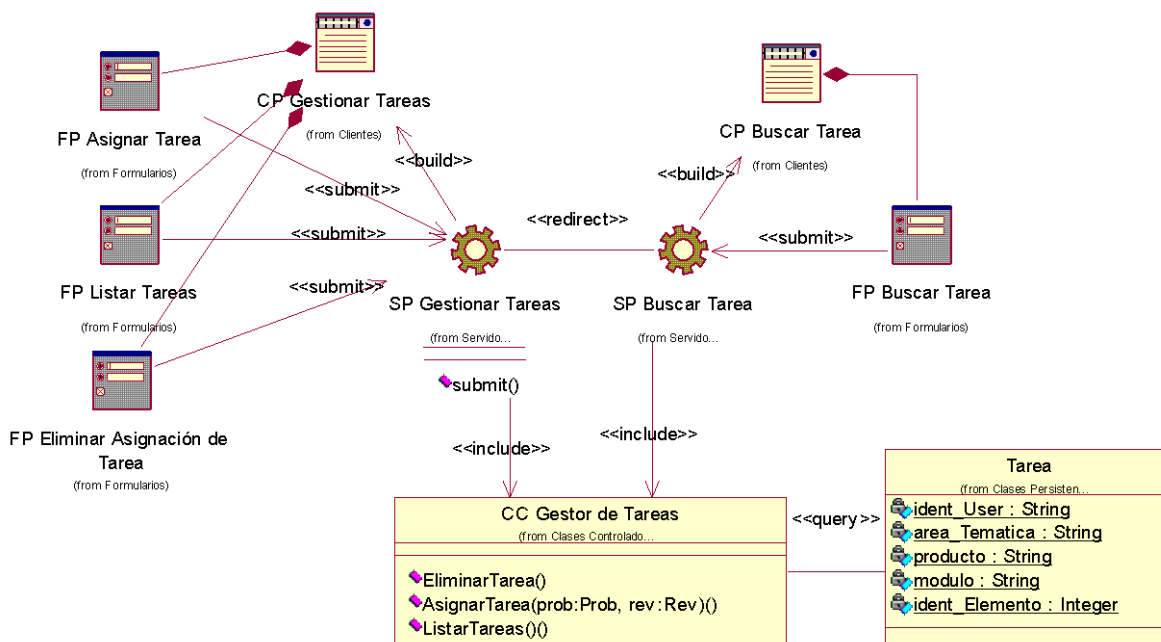


Fig. A5.6 -.DCD CU Gestionar Tareas

DIAGRAMA DE CLASES DEL DISEÑO (DCD) CU GENERAR REPORTE

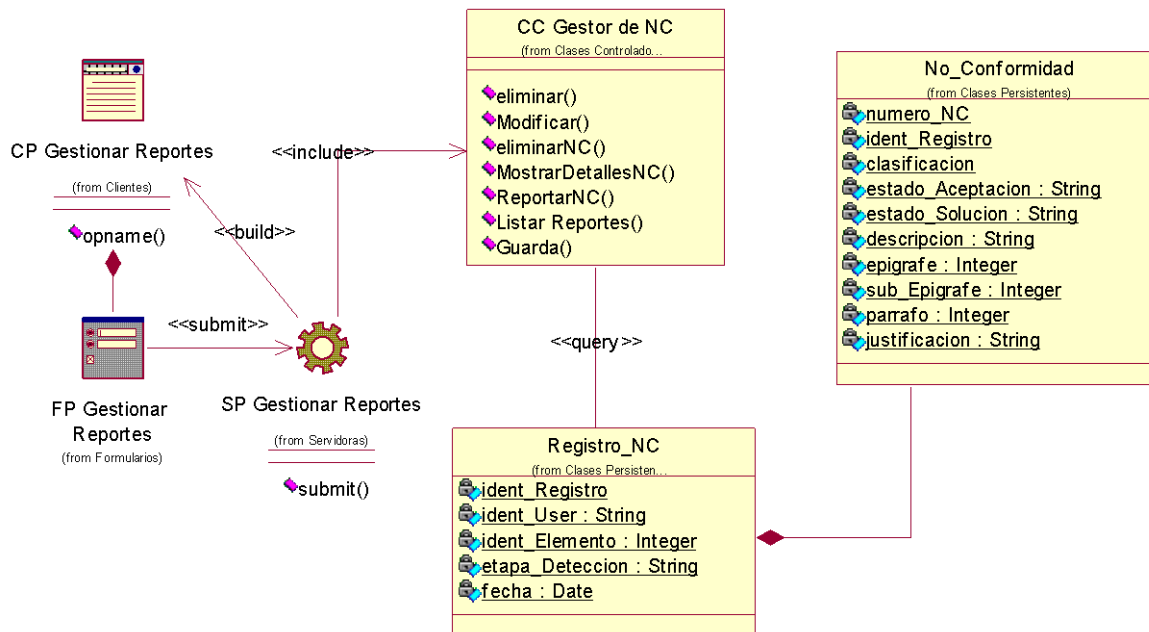


Fig. A5.7 -.DCD CU Gestionar Reportes

Anexo 6: Diagramas de Clases del Diseño

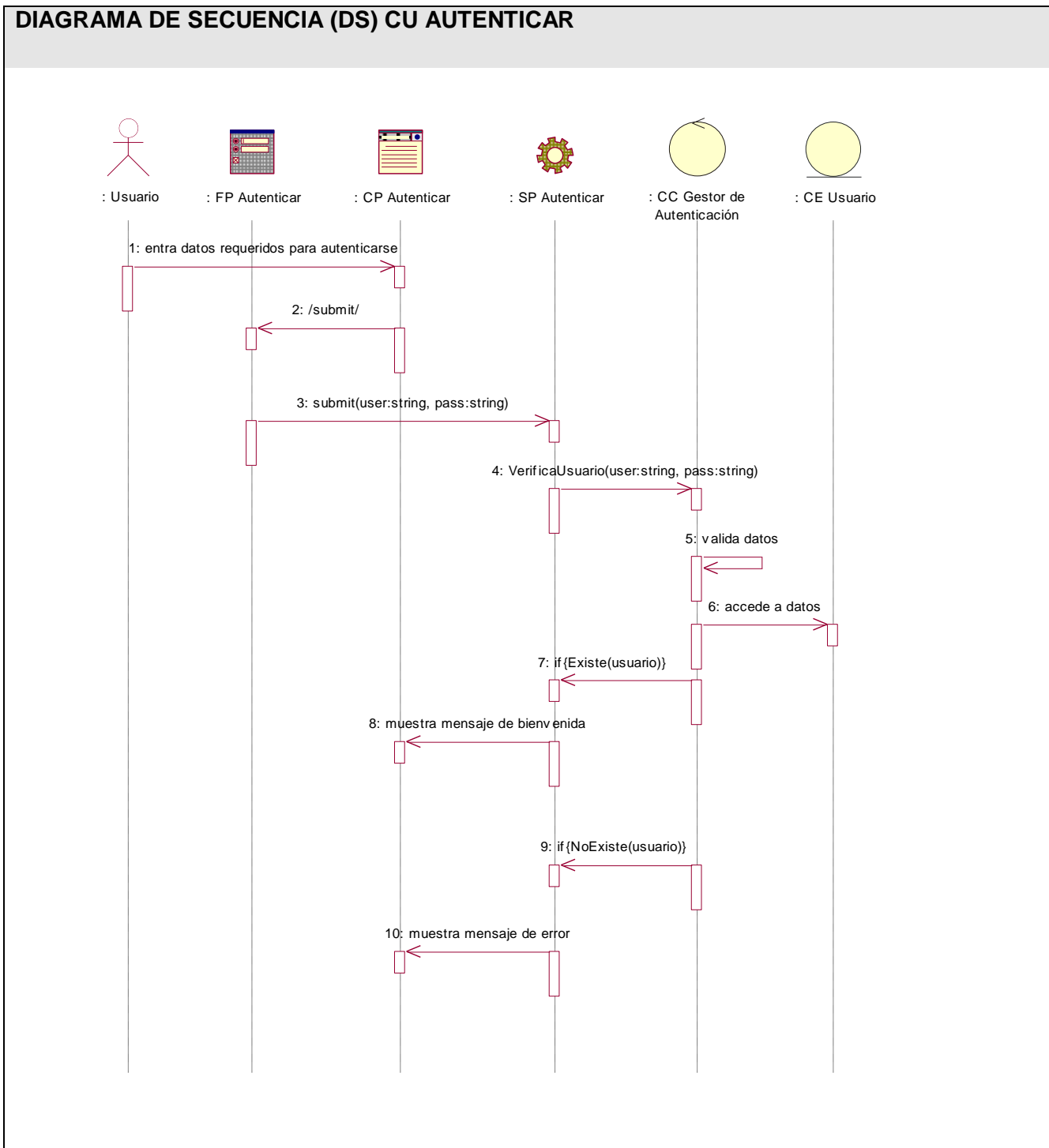


Fig. A6.1 -.DS CU Autenticar

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR USUARIOS (Crear Usuario)

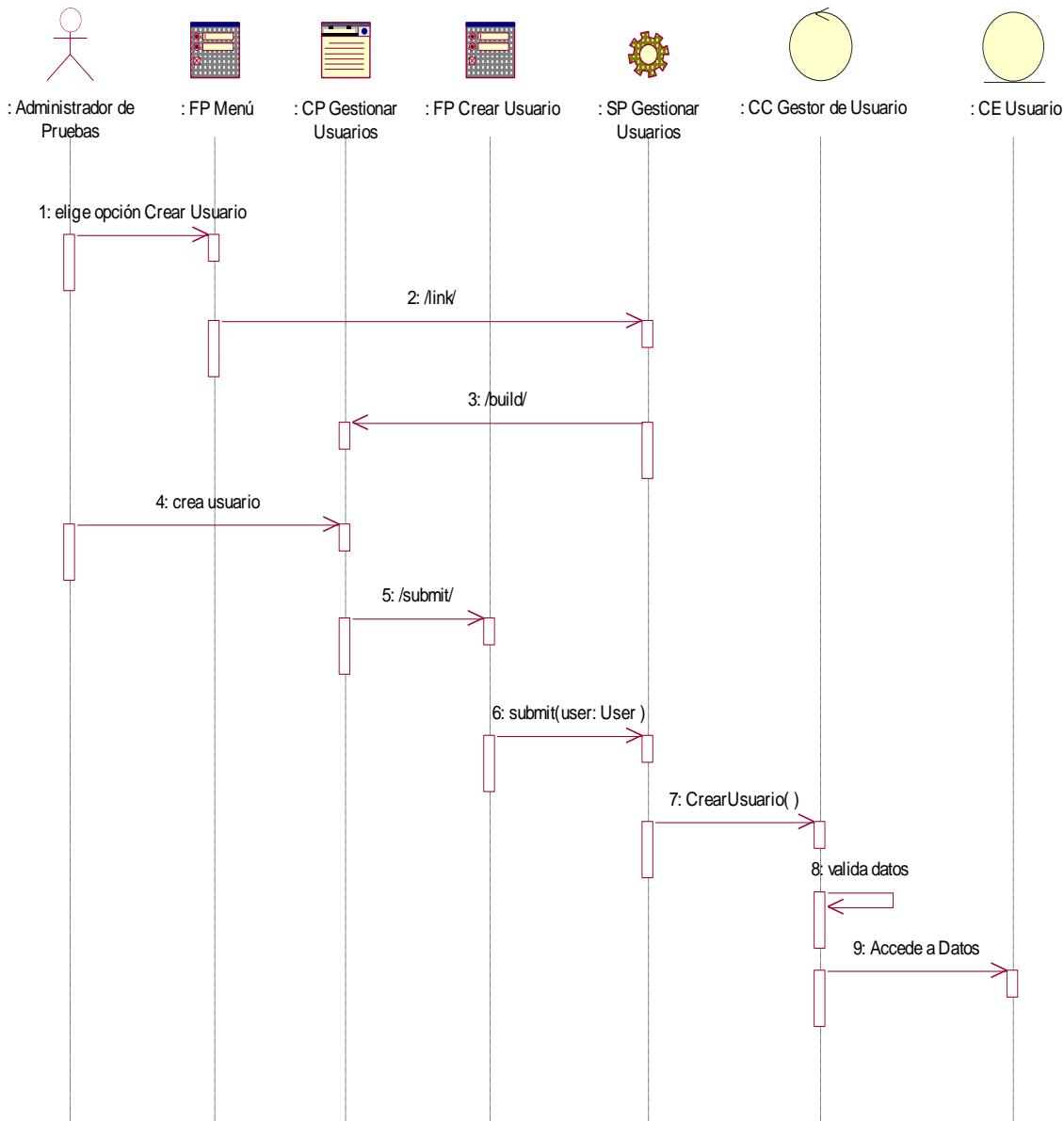


Fig. A6.2 -.DS CU Gestionar Usuarios (Crear Usuario).

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR USUARIOS (Modificar Usuario)

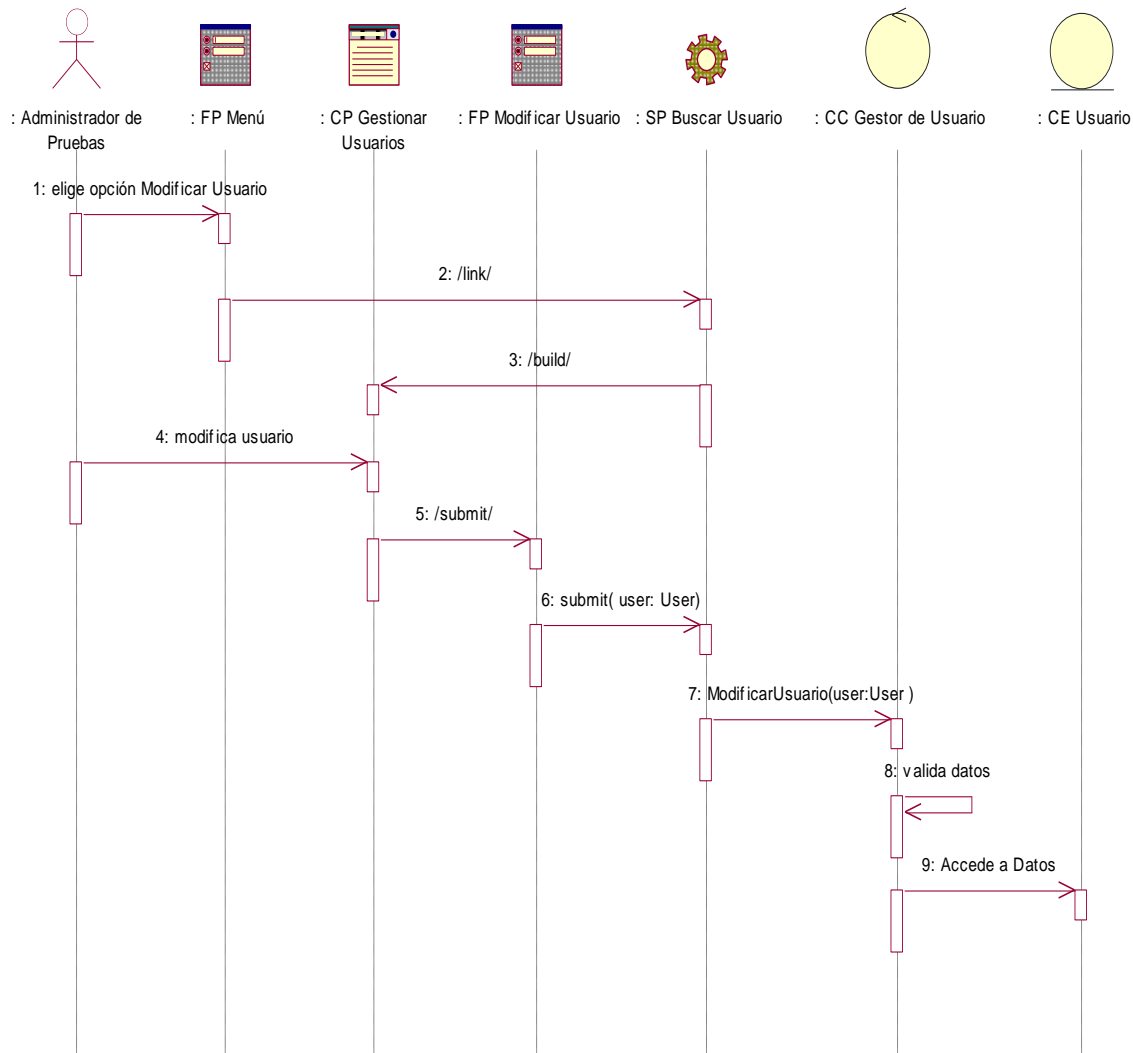


Fig. A6.3 -.DS CU Gestionar Usuarios (Modificar Usuario)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR USUARIOS (Eliminar Usuario)

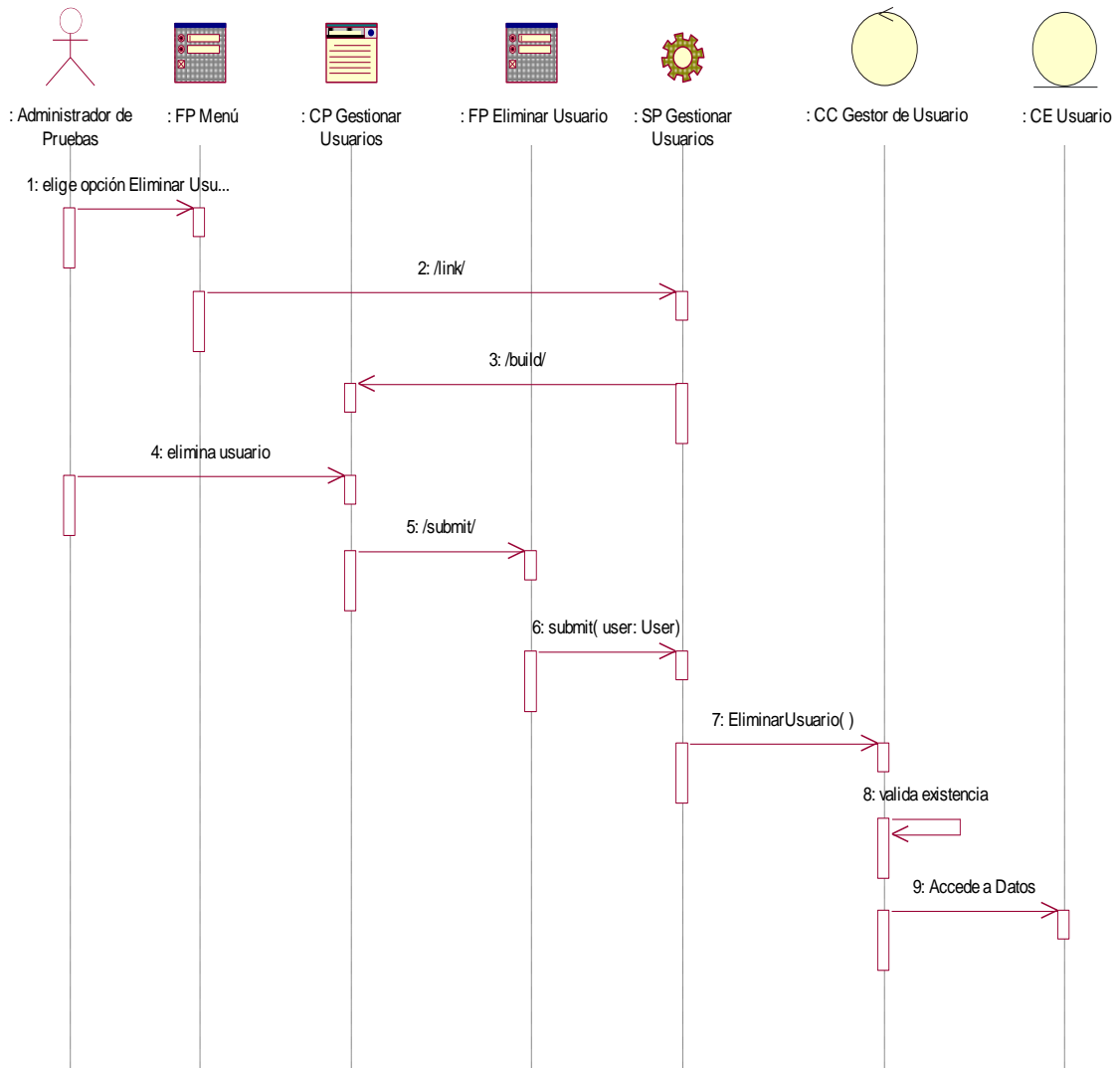


Fig. A6.4 -.DS CU Gestionar Usuarios (Eliminar Usuario)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR USUARIOS (Mostrar Detalles del Usuario)

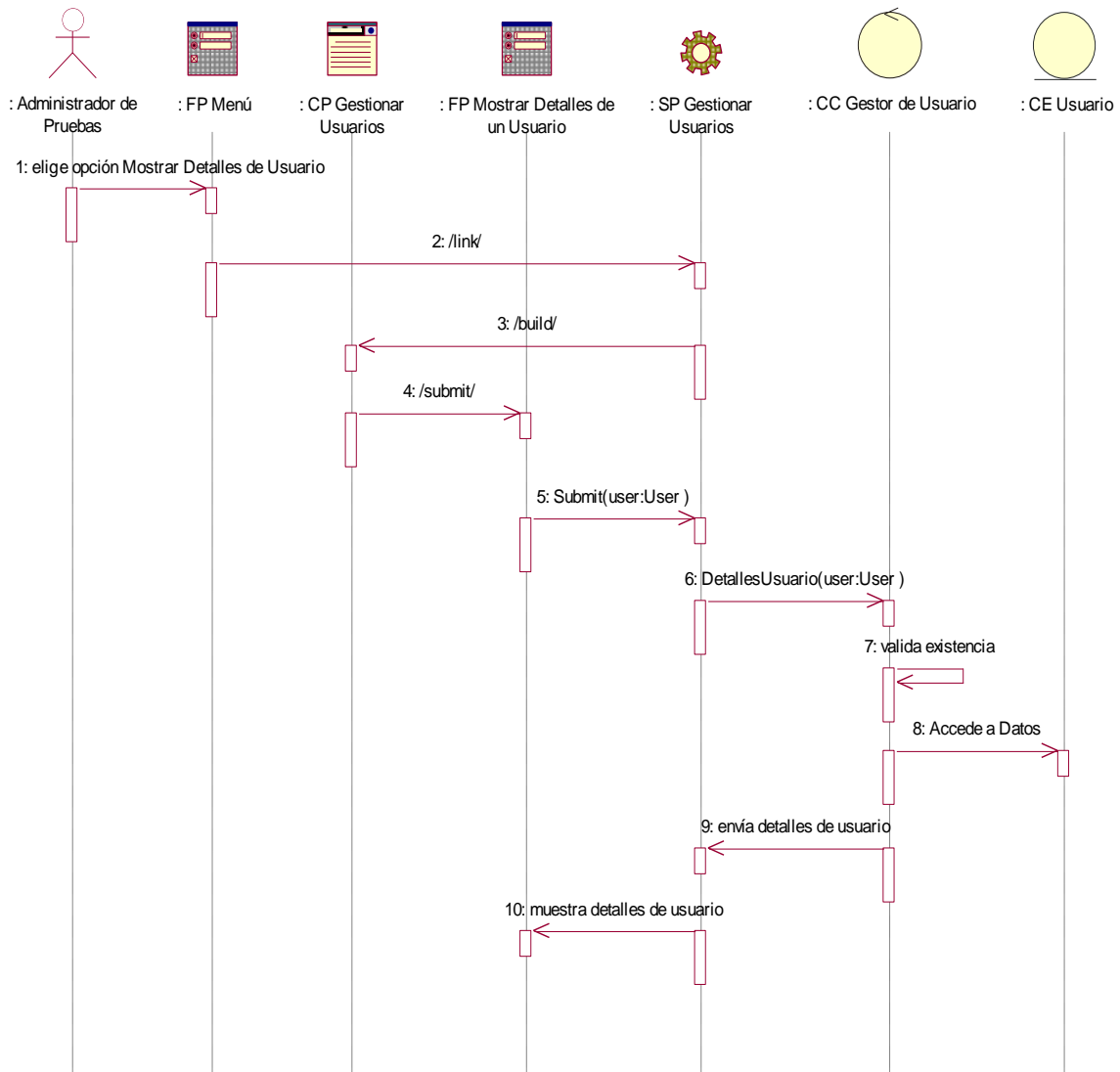


Fig. A6.5 -.DS CU Gestionar Usuarios (Mostrar Detalles del Usuario)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR ELEMENTO DEL EXPEDIENTE DE PROYECTO

(Crear Elementos del Expediente)

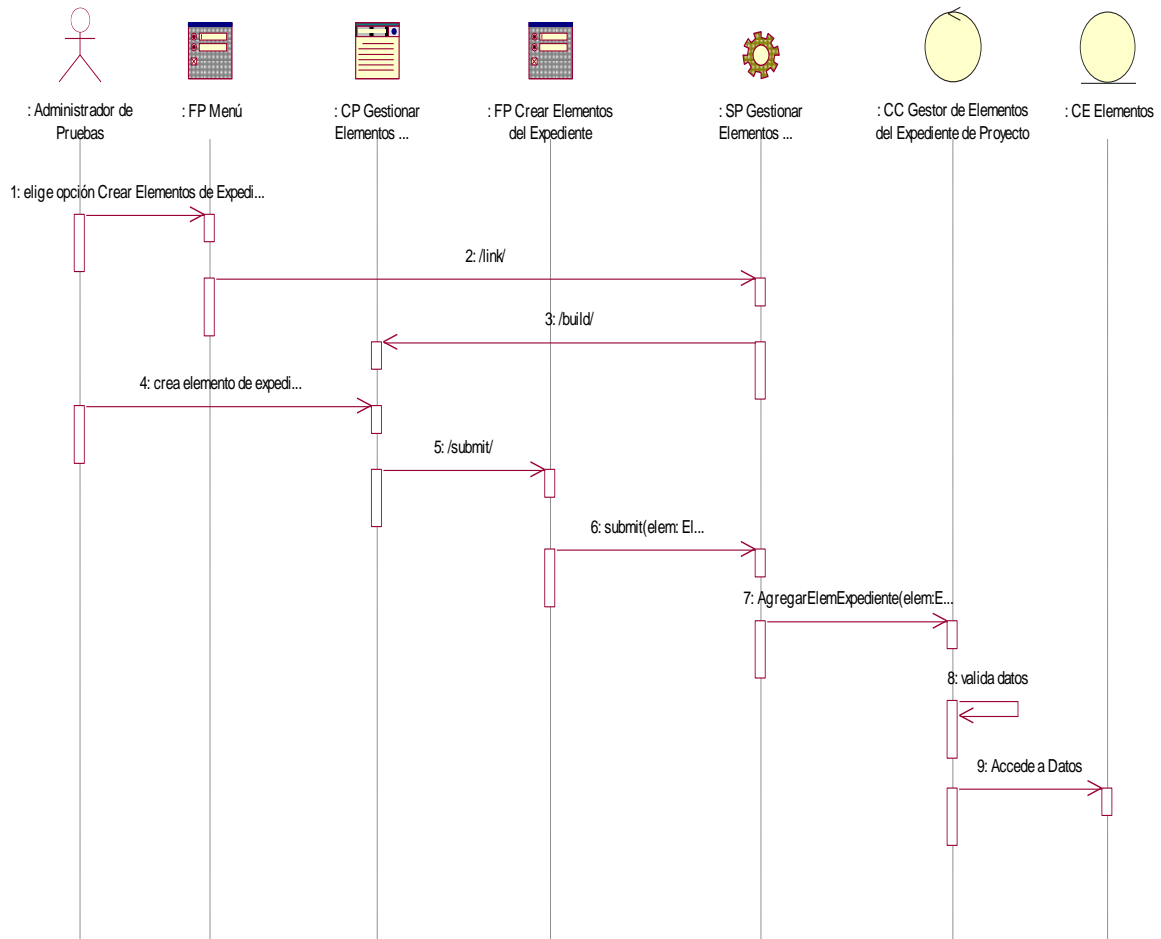


Fig. A6.6 -.DS CU Gestionar Elementos del Expediente de Proyecto (Crear Elementos del Expediente)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR ELEMENTO DEL EXPEDIENTE DE PROYECTO

(Modificar Elementos del Expediente)

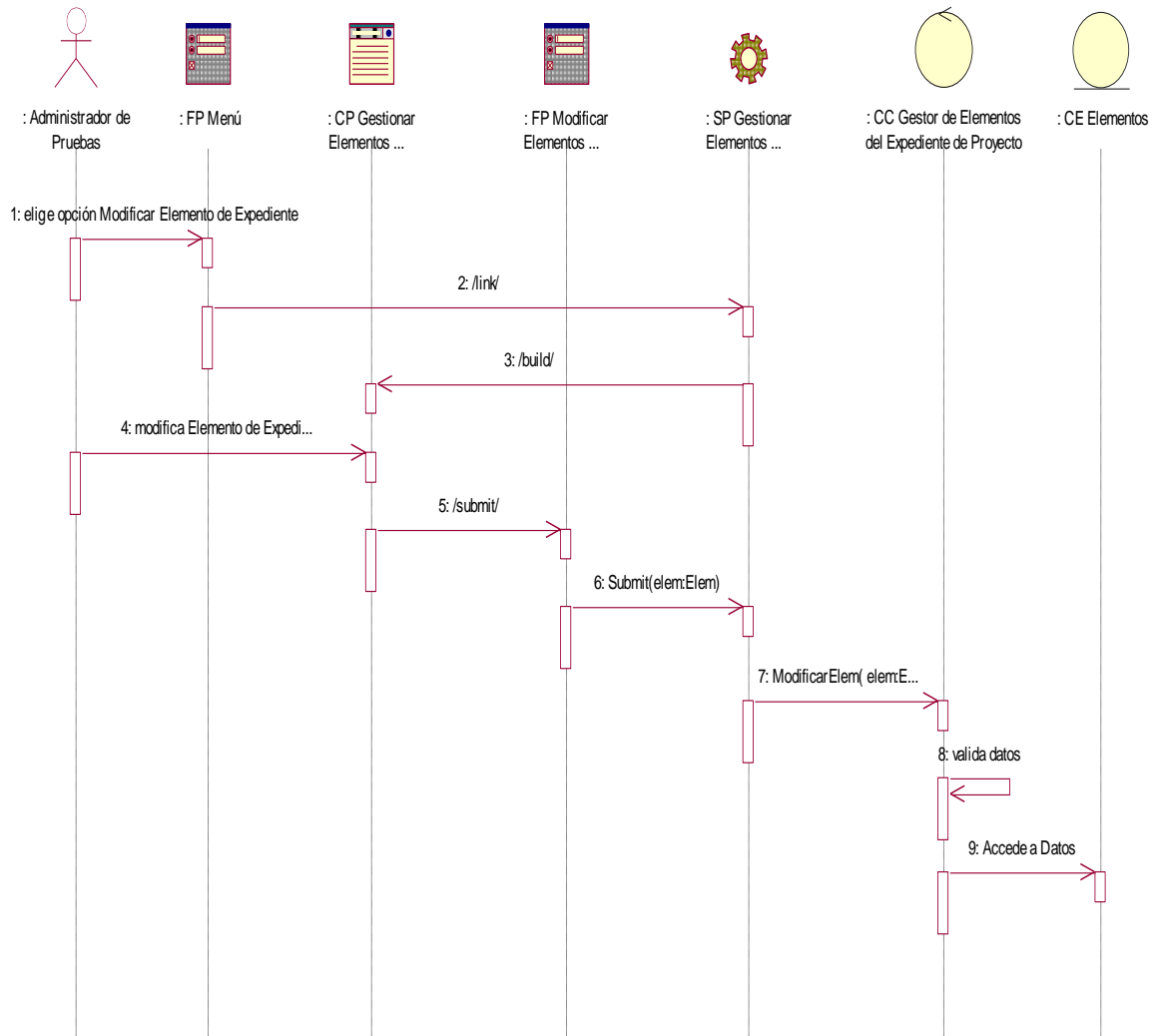


Fig. A6.7 -.DS CU Gestionar Elementos del Expediente de Proyecto (Modificar Elementos del Expediente)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR ELEMENTO DEL EXPEDIENTE DE PROYECTO

(Eliminar Elementos del Expediente)

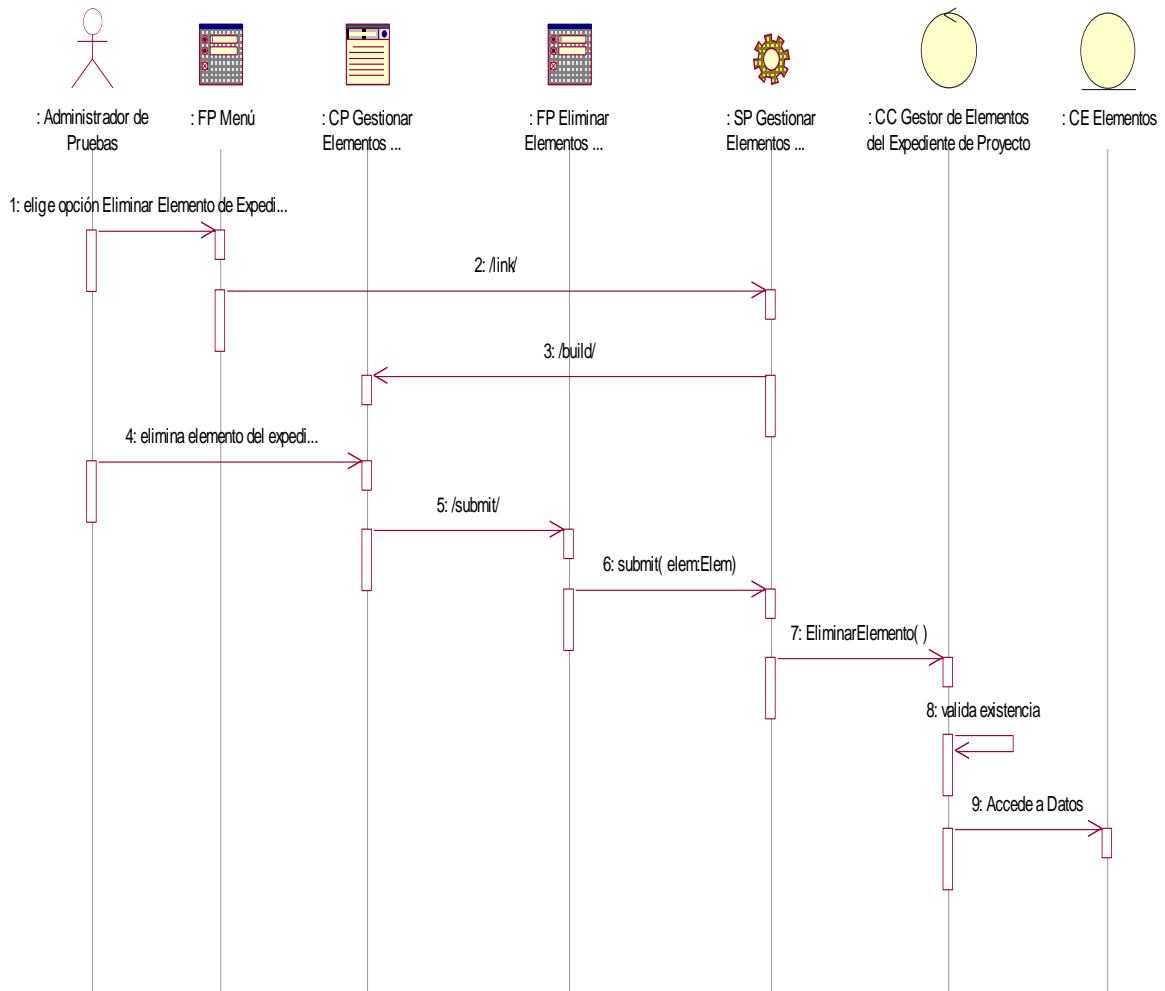


Fig. A6.8 -.DS CU Gestionar Elementos del Expediente de Proyecto (Eliminar Elementos del Expediente)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR REVISIONES (Crear Revisión)

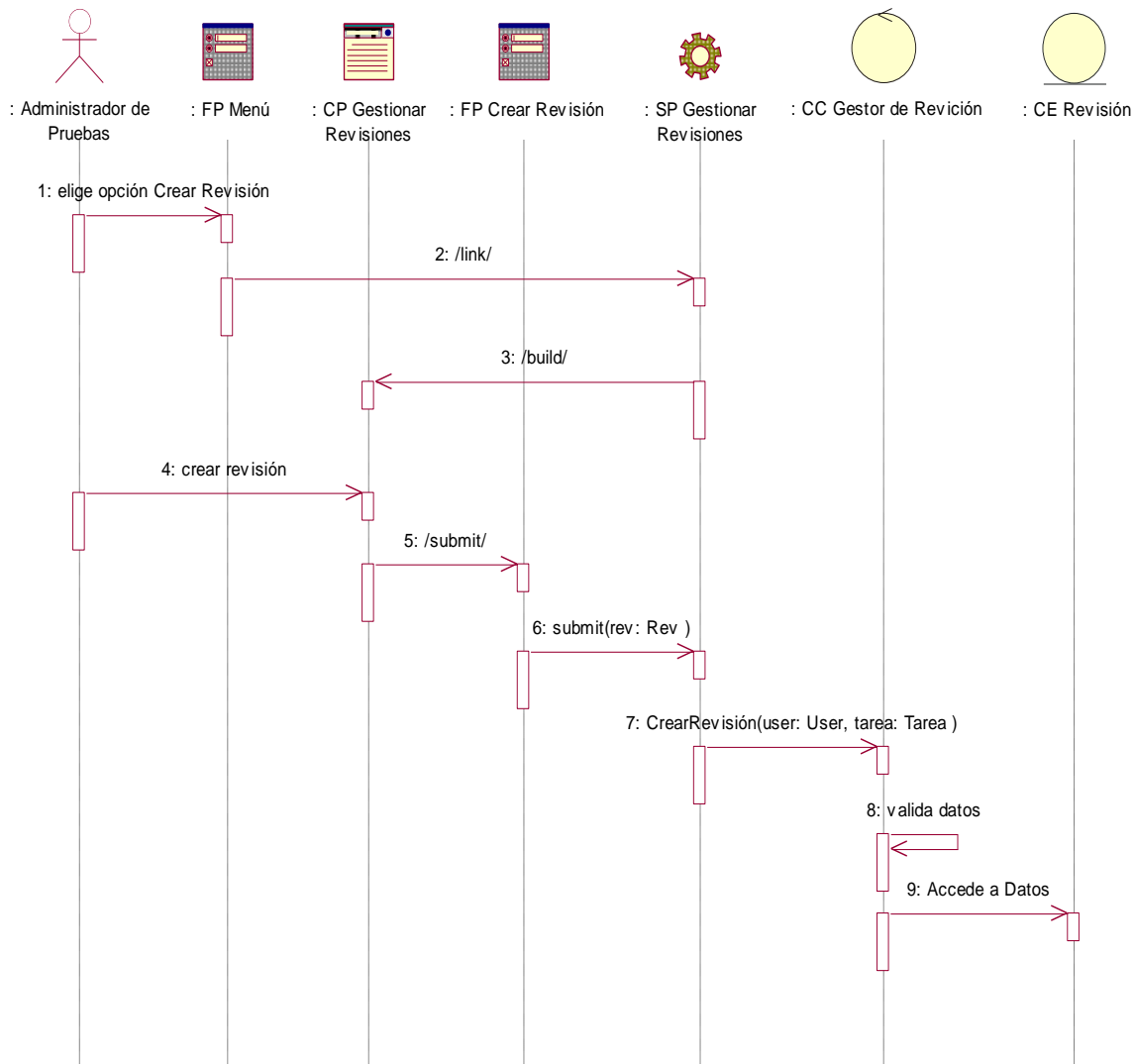


Fig. A6.9 -.DS CU Gestionar Revisiones (Crear Revisión)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR REVISIONES (Modificar Revisión)

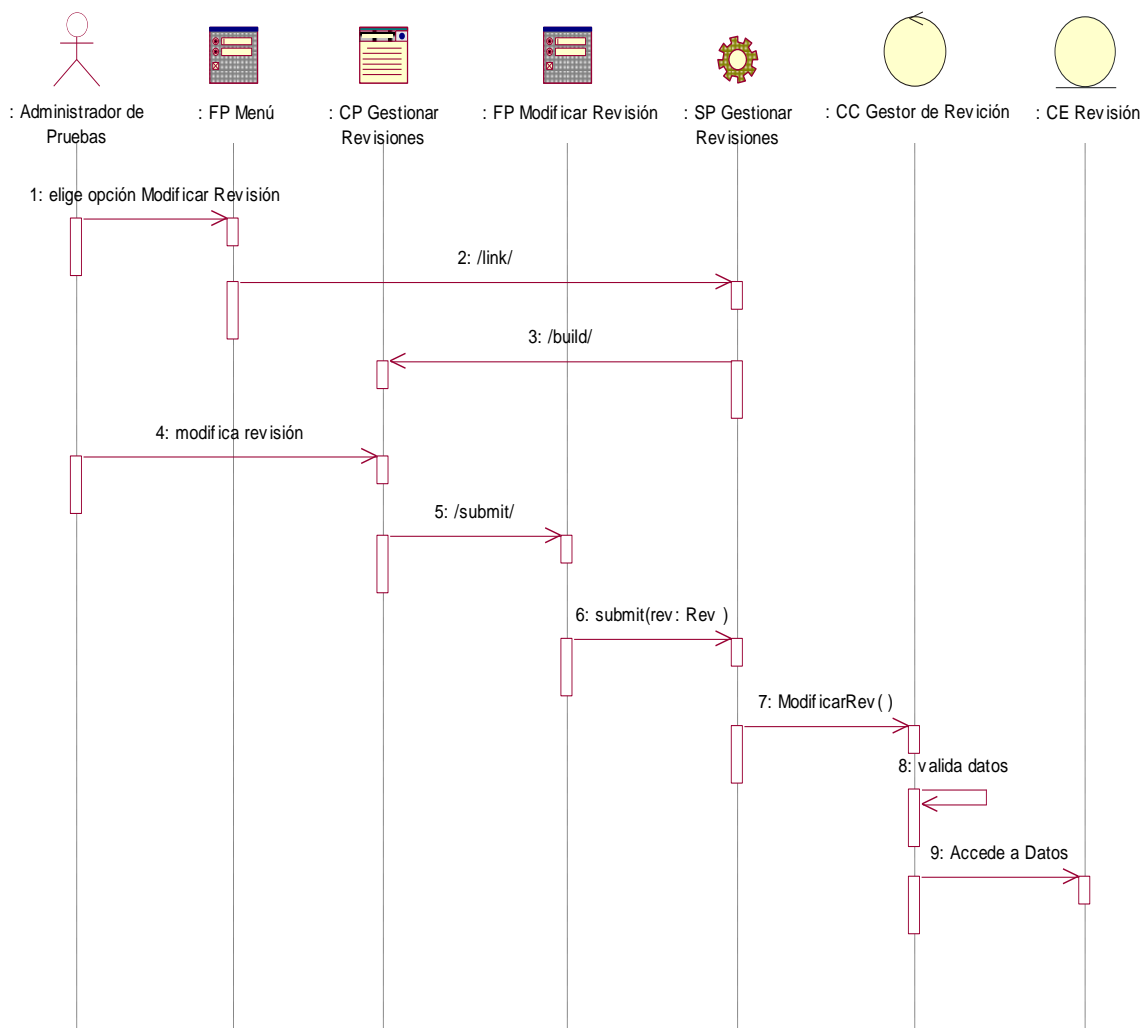


Fig. A6.10 -.DS CU Gestionar Revisiones (Modificar Revisión)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR REVISIONES (Eliminar Revisión)

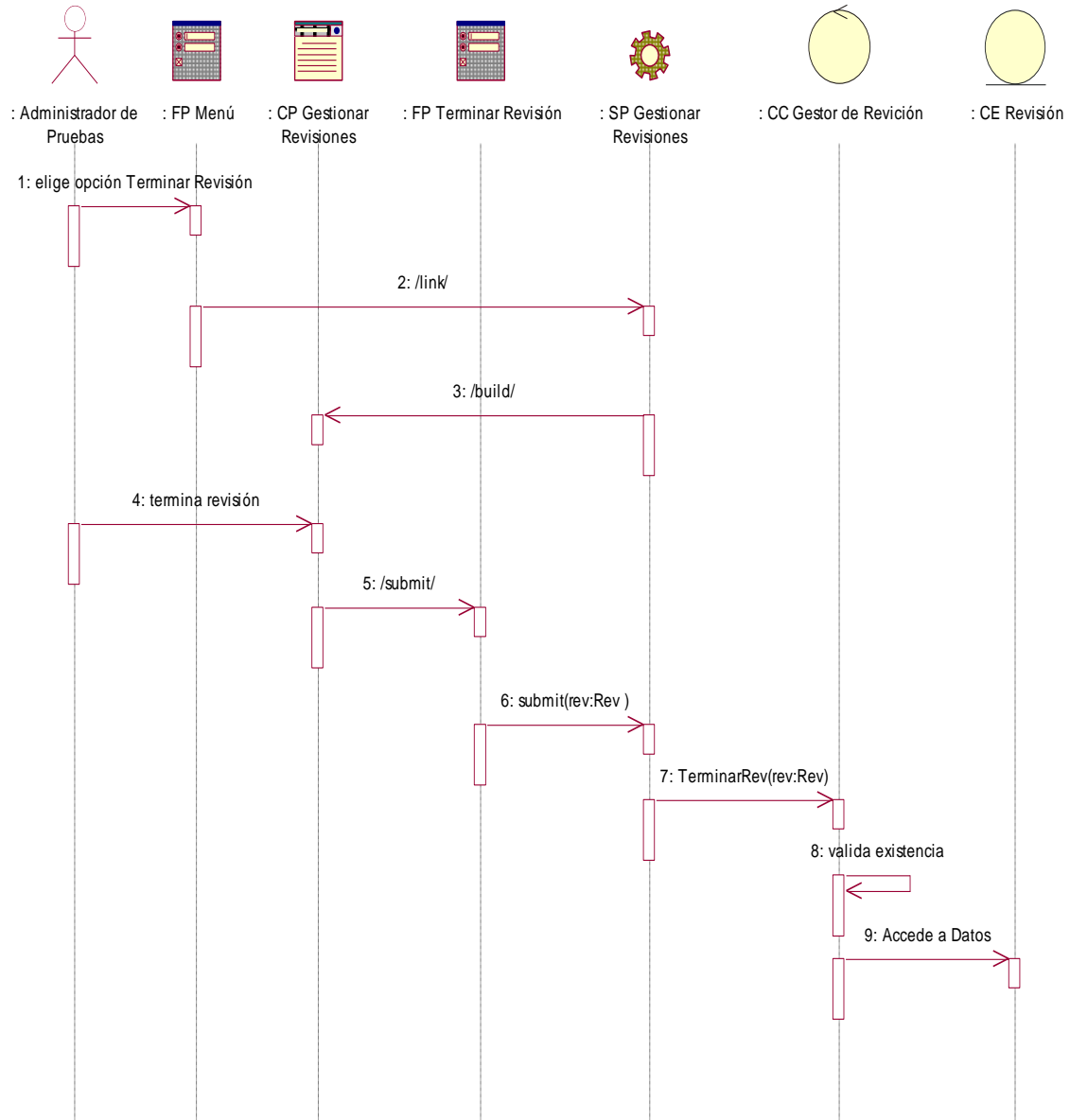


Fig. A6.11 -.DS CU Gestionar Revisiones (Eliminar Revisión)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR REVISIONES (Mostrar Detalles de la Revisión)

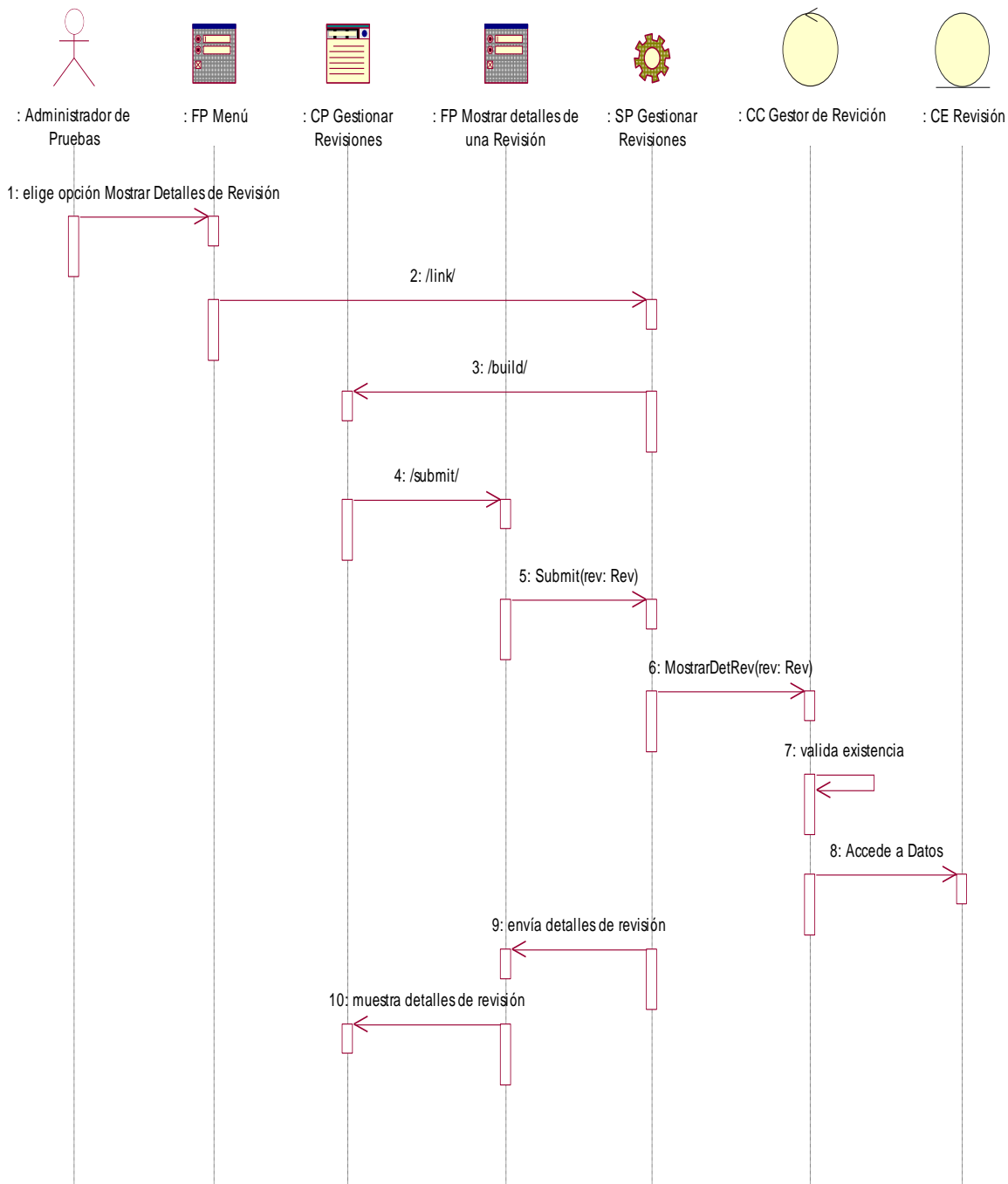


Fig. A6.12 -.DS CU Gestionar Revisiones (Mostrar Detalles de la Revisión)

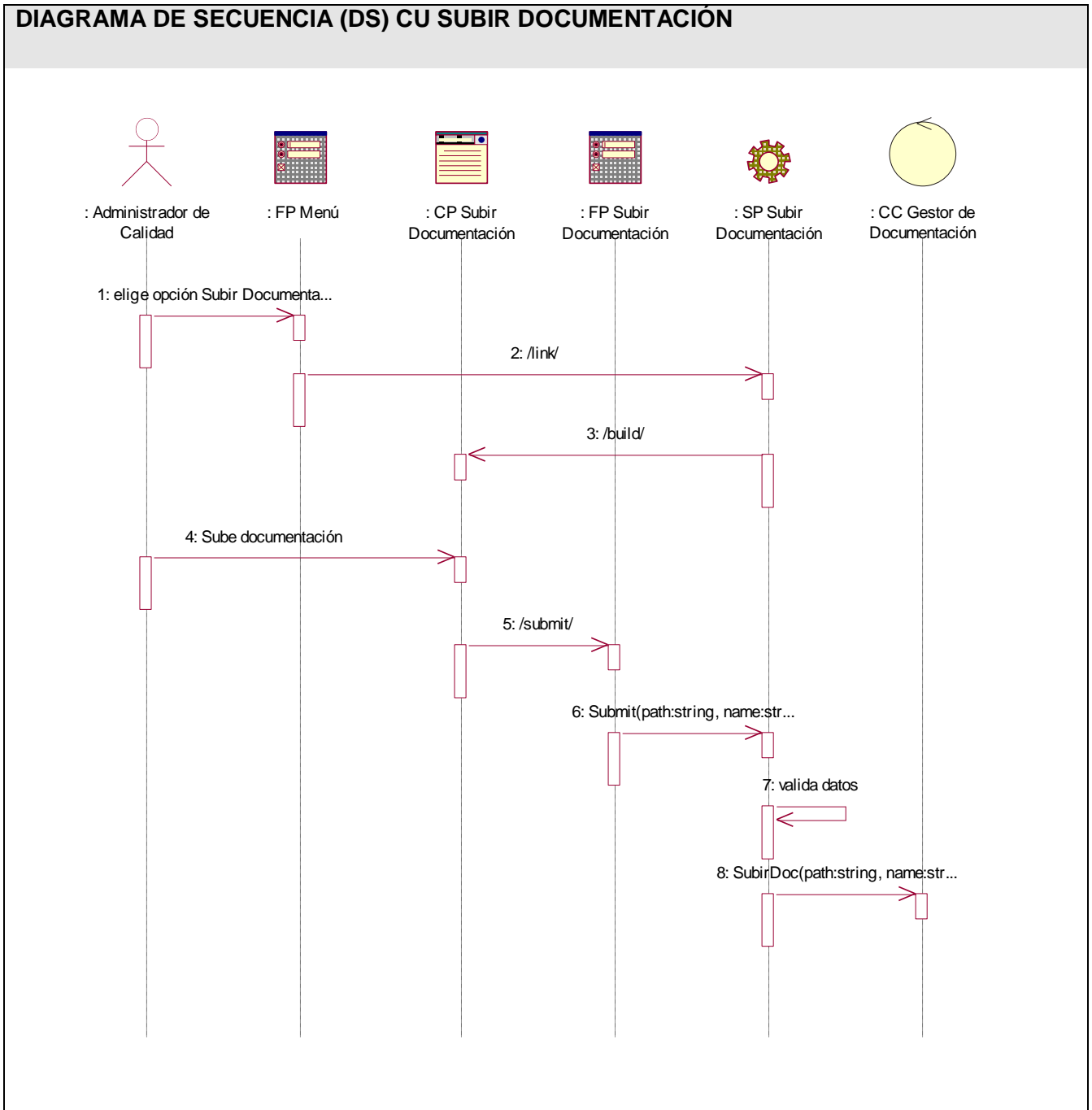


Fig. A6.13 -.DS CU Subir Documentación

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR TAREAS (Asignar Tarea)

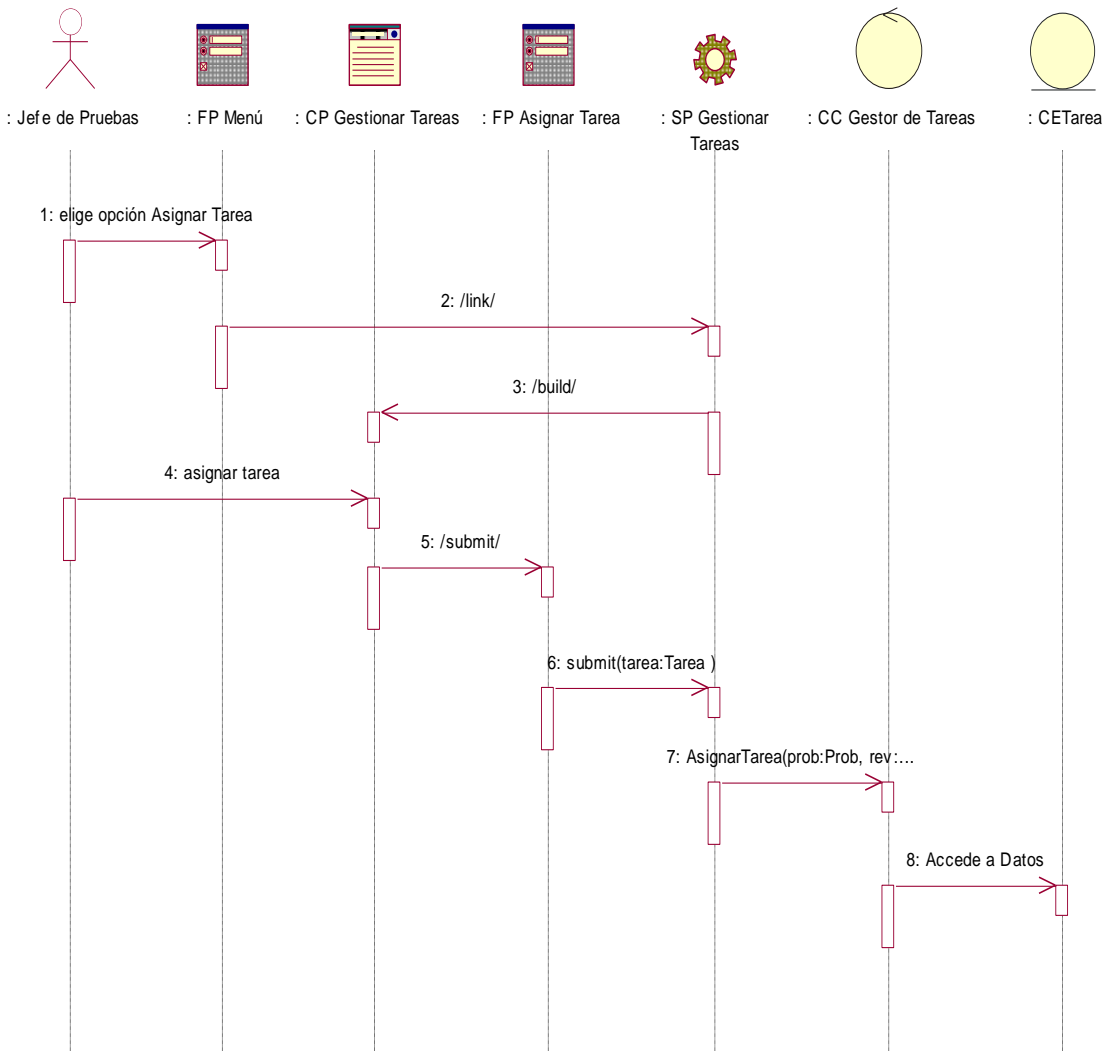


Fig. A6.14 -.DS CU Gestionar Tareas (Asignar Tarea)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR TAREAS (Eliminar Asignación de Tarea)

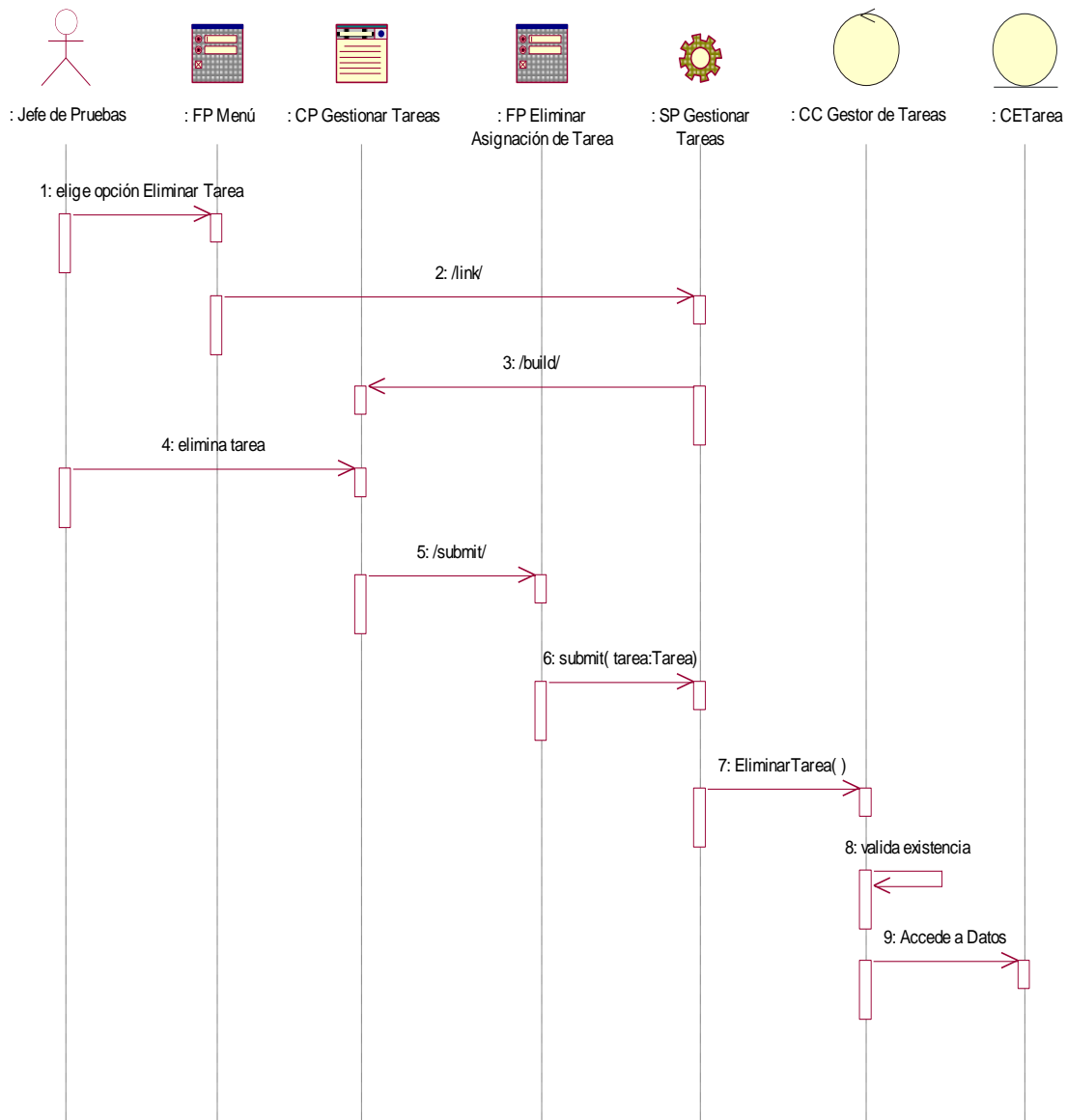


Fig. A6.15 -.DS CU Gestionar Tareas (Eliminar Asignación de Tarea)

DIAGRAMA DE SECUENCIA (DS) CU GESTIONAR TAREAS (Listar Tareas)

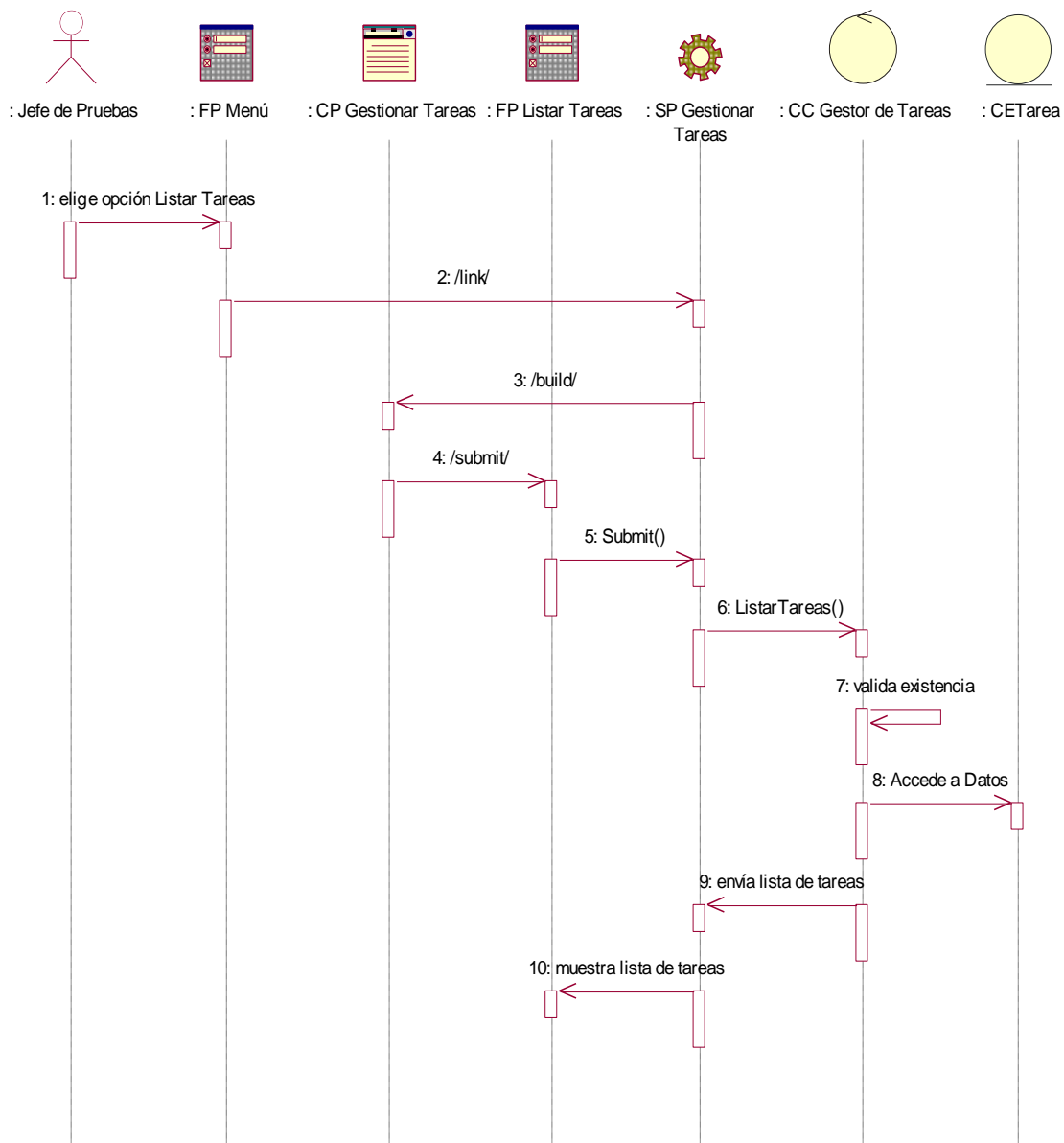


Fig. A6.16 -.DS CU Gestionar Tareas (Listar Tareas)

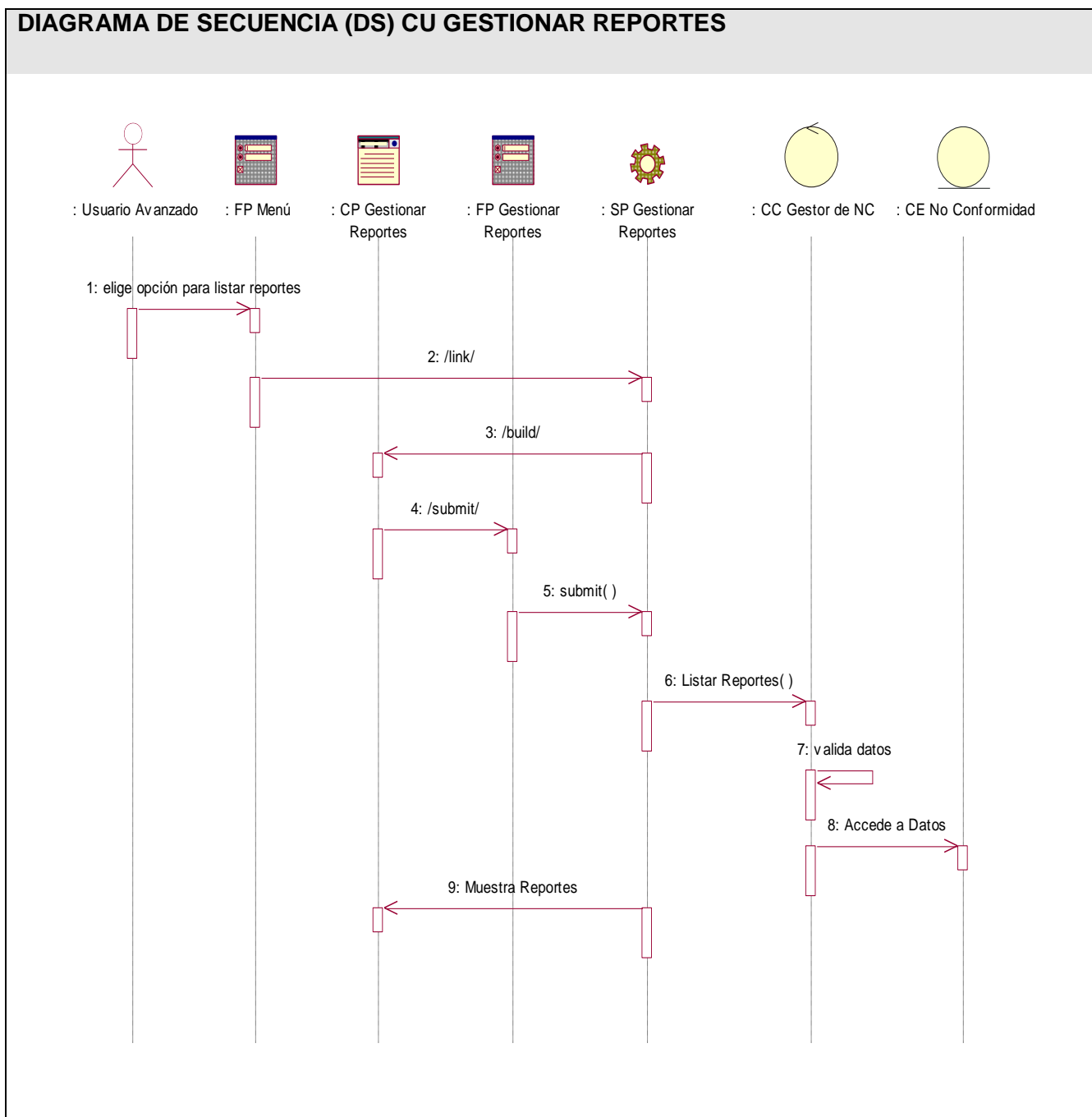


Fig. A6.7 -.DS CU Gestionar Reportes

Anexo 7: Descripción de las Clases del Diseño

Nombre: CI Autenticar	
Tipo de clase: Interfaz	
Atributo	Tipo

Usuario	textbox
Contraseña	textbox
Aceptar	button
Para cada responsabilidad:	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)
Descripción:	Este Método se ejecuta cuando se presiona el botón "Aceptar". Invoca a la funcionalidad encargada de la autenticación de los Usuarios.

Tabla A7.1 -.CI Autenticar

Nombre: CI Gestionar Usuarios	
Tipo de clase: Interfaz	
Atributo	Tipo
Usuario	Textbox
Contraseña	Textbox
Nombre(s)	Textbox
Apellido(s)	Textbox
Usuario de correo	Textbox
Servidor de correo	DropDownList
Rol	DropDownList
Lista de Usuarios	GridView
Aceptar	Button
Modificar	Button
Eliminar	Button

Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Aceptar". Invoca a la funcionalidad encargada de agregar los Usuarios.
Nombre:	void Modificar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Modificar". Invoca a la funcionalidad encargada de modificar los Usuarios.
Nombre:	void Eliminar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Eliminar". Invoca a la funcionalidad encargada de eliminar los Usuarios.
Nombre:	void Cancelar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Cancelar". Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.2 -.CI Gestionar Usuarios

Nombre: CI Gestionar Elementos del Expediente de Proyecto	
Tipo de clase: Interfaz	
Atributo	Tipo
Tipo del Elemento	DropDownList
Nombre del Elemento	Textbox
Lista de Elementos	GridView
Aceptar	Button
Modificar	Button

Eliminar	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Aceptar". Invoca a la funcionalidad encargada de agregar los Elementos.
Nombre:	void Modificar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Modificar". Invoca a la funcionalidad encargada de modificar los Elementos.
Nombre:	void Eliminar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Eliminar". Invoca a la funcionalidad encargada de eliminar los Elementos.
Nombre:	void Cancelar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Cancelar". Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.3 -.CI Gestionar Elementos del Expediente de Proyecto

Nombre: CI Gestionar Revisión	
Tipo de clase: Interfaz	
Atributo	Tipo
Proyecto	DropDownList
Producto	DropDownList
Módulo	DropDownList
Estado	DropDownList

Lista de Elementos	GridView
Lista de Revisiones	GridView
Aceptar	Button
Modificar	Button
Eliminar	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Aceptar". Invoca a la funcionalidad encargada de agregar las Revisiones.
Nombre:	void Modificar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Modificar". Invoca a la funcionalidad encargada de modificar las Revisiones.
Nombre:	void Eliminar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Eliminar". Invoca a la funcionalidad encargada de eliminar las Revisiones.
Nombre:	void Cancelar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Cancelar". Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.4 -.CI Gestionar Revisión

Nombre: CI Subir Documentación	
Tipo de clase: Interfaz	
Atributo	Tipo

Lista de Elementos	GridView
Examinar	UploadFile
Compartir	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Compartir_Click (Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Compartir". Permite subir a la aplicación el archivo buscado.
Nombre:	void Cancelar_Click (Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón "Cancelar". Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.5 -.CI Subir Documentación

Nombre: CI Asignar Tareas	
Tipo de clase: Interfaz	
Atributo	Tipo
Lista de Probadores	GridView
Lista de Revisiones	GridView
Lista de Elementos	GridView
Lista de Tareas	GridView
Aceptar	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Aceptar_Click(Object sender, EventArgs e)

Descripción:	Este método se ejecuta cuando se presiona el botón “Aceptar”. Invoca a la funcionalidad encargada de la creación de las Tareas.
Nombre:	void Cancelar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón “Cancelar”. Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.6 -.CI Asignar Tareas

Nombre: CI Gestionar Reportes	
Tipo de clase: Interfaz	
Atributo	Tipo
Lista de Probadores	GridView
Lista de Revisiones	GridView
Lista de Elementos	GridView
Lista de Tareas	GridView
Obtener Reporte	Button
Cancelar	Button
Para cada responsabilidad:	
Nombre:	void Reporte_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón “Obtener Reporte”. Invoca a la funcionalidad encargada de la creación de los Reportes.
Nombre:	void Cancelar_Click(Object sender, EventArgs e)
Descripción:	Este método se ejecuta cuando se presiona el botón “Cancelar”. Invoca a la funcionalidad encargada de cancelar una acción en progreso.

Tabla A7.6 -.CI Gestionar Reportes

Nombre: CC Login	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Login(string conexion)
Descripción:	Crea conexión con la Base de Datos
Nombre:	void VerificarAcceso(String usuario, String password)
Descripción:	Se encarga de chequear los datos de entrada y permitir o denegar el acceso al sistema, otorga los permisos de navegación.

Tabla A7.7 -.CC Login

Nombre: CC Gestionar Usuarios	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void Adicionar_Usuario (User user)
Descripción:	Este método es el encargado de adicionar un nuevo Usuario
Nombre:	void Modificar_Usuario (User user)
Descripción:	Este método modifica un Usuario.
Nombre:	void Eliminar_Usuario (String user)
Descripción:	Este método es el encargado de eliminar un Usuario.
Nombre:	User Obtener_Usuario (String id_User)
Descripción:	Este método busca un Usuario determinado.

Tabla A7.8 -.CC Gestionar Usuarios

Nombre: CC Gestionar Elementos del Expediente de Proyecto	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void Adicionar_Elemento (Elemento nuevo)
Descripción:	Este método es el encargado de adicionar un nuevo Elemento.
Nombre:	void Modificar_Elemento (Elemento nuevo)
Descripción:	Este método modifica un Elemento.
Nombre:	void Eliminar_Elemento (int ident_Elem)
Descripción:	Este método es el encargado de eliminar un Elemento.
Nombre:	User Obtener_Elemento (int ident_Elem)
Descripción:	Este método busca un Elemento determinado.

Tabla A7.9 -.CC Gestionar Elementos del Expediente de Proyecto

Nombre: CC Gestionar Revisiones	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void Adicionar_Revision (Revision nueva)
Descripción:	Este método es el encargado de adicionar una nueva Revisión.

Nombre:	void Modificar_Revision (Revision nueva)
Descripción:	Este método modifica una nueva Revisión.
Nombre:	void Eliminar_Revision (int ident_Elem)
Descripción:	Este método es el encargado de eliminar una Revisión.
Nombre:	User Obtener_Revision (int ident_Elem)
Descripción:	Este método busca una Revisión determinada.

Tabla A7.10 -.CC Gestionar Revisiones

Nombre: CC Subir Documentos	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void Subir_Doc(String camino)
Descripción:	Subir la documentación a la aplicación.

Tabla A7.11 -.CC Subir Documentos

Nombre: CC Gestionar Tareas	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	void Adicionar_Tarea (Tarea nueva)
Descripción:	Este método es el encargado de adicionar una nueva Tarea.

Nombre:	void Modificar_Tarea (Tarea nueva)
Descripción:	Este método modifica una nueva Tarea.
Nombre:	void Eliminar_Tarea (int ident_Tarea)
Descripción:	Este método es el encargado de eliminar una Tarea.
Nombre:	User Obtener_Tarea (int ident_Tarea)
Descripción:	Este método busca una Tarea determinada.

Tabla A7.12 -.CC Gestionar Tareas

Nombre: CC Gestionar Reportes	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	List<RegistrosNC> Registros_Elementos (Elemento ver)
Descripción:	Este método es el encargado de obtener los Registros de NC de un elemento.
Nombre:	List<RegistrosNC> Registros_Revision (Revision ver)
Descripción:	Este método es el encargado de obtener los Registros de NC de una Revisión.
Nombre:	List<RegistrosNC> Registros_Usuario (Usuario ver)
Descripción:	Este método es el encargado de obtener los Registros de NC de un Usuario.

Tabla A7.13 -.CC Gestionar Reportes

Nombre: CE Usuario

Tipo de clase: Entidad	
Atributo	Tipo
usuario	String
contraseña	String
correo	String
nombre	String
apellidos	String
rol	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla A7.14 -.CE Usuario

Nombre: CE Equipo de Desarrollo	
Tipo de clase: Entidad	
Atributo	Tipo
usuario	String
area_Tematica	String
producto	String
modulo	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla A7.15 -.CE Equipo de Desarrollo

Nombre: CE Elemento del Expediente de Proyecto	
Tipo de clase: Entidad	
Atributo	Tipo
usuario	String
contraseña	String
correo	String
nombre	String
apellidos	String
rol	String
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla A7.16 -.CE Elemento del Expediente de Proyecto

Nombre: CE Revision	
Tipo de clase: Entidad	
Atributo	Tipo
area_Tematica	String
producto	String
modulo	String
estado	String
fecha_Inicio	DateTime
fecha_Fin	DateTime

Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla A7.17 -.CE Revision

Nombre: CE Tarea	
Tipo de clase: Entidad	
Atributo	Tipo
area_Tematica	String
producto	String
modulo	String
usuario	String
ident_Elem	Integer
Para cada responsabilidad:	
Nombre:	
Descripción:	

Tabla A7.18 -.CE Tarea