

Universidad de las Ciencias Informáticas

Facultad 4



Título: Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad.

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autor: Amari Jacsson Morrell Rojas.

Tutor: Ing. Arturo César Arias Orizondo.

Julio 2007

Año del 48 Aniversario del Triunfo de la Revolución

La Habana

Cuba



La sabiduría suprema es tener sueños bastante grandes para no perderlos de vista mientras se persiguen.

William Faulkner.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los _____ días del mes de _____ del _____.

Nombre completo del primer autor

Nombre completo del segundo autor

Nombre completo del tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma titulado "Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad"; fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente _____

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR

Título: Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad.

Autor: Amari Jacsson Morrell Rojas.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

DATOS DE CONTACTO

- Ingeniero Arturo César Arias Orizondo.
- Profesor Instructor.
- Graduado en el 2003 de Ingeniero Informático del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Título de Oro, promedio 5 puntos.
- Ha impartido las asignaturas de Sistemas de Bases de Datos, Ingeniería de Software I y II, Gestión de Software en la Universidad de Ciencias Informáticas, participando en la elaboración de los programas de estas asignaturas.
- Miembro del tribunal de acreditación de competencias del Departamento de Ingeniería de Software.
- Vicedecano de producción e investigación de la UCI por dos cursos: 2004-2005, 2005-2006.
- Miembro del consejo científico de la UCI.
- Miembro de la comisión de carrera durante los cursos 2004-2005, 2005-2006.
- Ha impartido cursos de postgrado sobre RUP y UML.
- Participó en el Congreso y Feria TechnoInternet 2004 celebrado en Santiago de Chile, Chile.
- Ha participado en evaluaciones de calidad de productos de software para Venezuela.
- Diseñador de base de datos del proyecto SAFRE, primero de exportación de la UCI.
- Líder del proyecto de desarrollo de software para el sistema penitenciario venezolano desde el 2005.

AGRADECIMIENTOS

De Amari Jacsson:

A mis padres y hermana por los años de esfuerzo, sacrificio, confianza y el apoyo que me brindan.

A mis amigas y amigos por su dedicación, cariño, apoyo, paciencia e incondicionalidad.

A mi tutor por la paciencia y la confianza depositada en la realización de este trabajo.

A todas aquellas personas que de alguna manera han ayudado a que realice mis sueños.

DEDICATORIA

Dedico este trabajo a dos personas maravillosas: Gricelia Rojas Hernández y Amari René Morrell García, que me dieron la vida, la guía, el apoyo, la confianza, la voluntad y la fuerza para luchar por lograr mis sueños. Dos seres a los cuales agradezco por existir, por confiar en mí, por no cansarse, por estar siempre, por ser sencillamente como son conmigo, por ser sencillamente especiales.

Aunque toda una vida no alcanzará para recompensarlos, quisiera que alcanzará al menos para hacerlos sentir orgullosos, para agradecerles, para demostrarles que cuanto soy y tengo es gracias a ustedes.

Sinceramente Amari Jacsson.

RESUMEN

Tras la realización de las pruebas de calidad de software se genera gran volumen de información la que se almacena como no conformidades, para su posterior análisis. Actualmente la tabulación de los datos se realiza manualmente, introduciéndose en el proceso un número considerable de errores, unido al tiempo que se debe invertir en ello, lo que atenta contra la calidad, los costos y tiempo de desarrollo.

Se hace necesario registrar las no conformidades de forma centralizada siguiendo un estándar que permita tabular esta información y obtener reportes, que posibiliten la toma de decisiones sobre qué acciones realizar en aras de resolver los errores detectados en el menor período de tiempo posible.

El presente trabajo tiene como objetivos la modelación e implementación de un sistema que posibilite informatizar el análisis de la información registrada a través de las no conformidades para su posterior consulta, hacer reportes a partir de la información con que se cuenta para conocer en que 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.

Para la implementación del mismo se ha hecho uso de los Web Services.Net, por las facilidades y potencialidades que los mismos presentan; del lenguaje de programación C#.Net, ya que el mismo engloba las mejores características de lenguajes como C, C++ y Java; y de la arquitectura de tres capas por la flexibilidad y facilidades que esta brinda cuando se hace necesario realizar modificaciones. Se cuenta además con una base de datos para almacenar la información persistentemente para que pueda ser consultada en todo momento.

ÍNDICE

INTRODUCCIÓN1

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....4

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA 11

2.1 INTRODUCCIÓN. 11

2.2 PROPUESTA DEL SISTEMA..... 11

2.3 HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL SISTEMA..... 12

2.4 MODELO DE DOMINIO 15

 2.4.1 Modelo de Objetos. 15

 2.4.2 Descripción de los Trabajadores. 16

2.5 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE. 17

 2.5.1 Requisitos Funcionales. 17

 2.5.2 Requisitos No Funcionales. 18

2.6 DEFINICIÓN DE LOS ACTORES DEL SISTEMA. 20

 2.6.1 Especificación de los Casos de Uso. 20

2.7 DIAGRAMA DE CASOS DE USO DEL SISTEMA. 23

2.8 DESCRIPCIÓN DETALLADA DE CASOS DE USO DEL SISTEMA. 24

2.9 CONCLUSIONES..... 69

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA 70

3.1 INTRODUCCIÓN. 70

3.2 DIAGRAMA DE CLASES DEL ANÁLISIS..... 71

3.3 DIAGRAMAS DE INTERACCIÓN (COLABORACIÓN)..... 74

3.4 DIAGRAMA DE CLASES DEL DISEÑO. 79

 3.4.1 Paquete de Informes de No Conformidad. 79

 3.4.1.1 Descripción de las Clases del Diseño. 79

 3.4.2 Paquete de Administración. 83

 3.4.2.1 Descripción de las Clases del Diseño. 83

3.5 DISEÑO DE LA BASE DE DATOS..... 88

 3.5.1 Descripción de las Tablas de la Base de Datos. 89

3.6 MECANISMO DE SEGURIDAD Y TRATAMIENTO DE ERRORES. 94

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA.....95

4.1 INTRODUCCIÓN.....95

4.2 MODELO DE IMPLEMENTACIÓN.....95

4.3 DIAGRAMA DE DESPLIEGUE.....96

4.4 MODELO DE PRUEBA.....96

4.5 CONCLUSIONES.....99

CONCLUSIONES.....101

RECOMENDACIONES.....102

BIBLIOGRAFÍA REFENRECIADA.....103

BIBLIOGRAFÍA.....104

GLOSARIO DE TÉRMINOS.....105

INTRODUCCIÓN

El desarrollo de sistemas hoy en día ha tomado gran importancia en el mundo siendo ésta cada vez más creciente. Aunque esta importancia tienda a aumentar, no todo tiene una buena perspectiva, existen inconvenientes en el desarrollo de los sistemas: grandes retrasos en la programación, inconsistencia en su funcionamiento, etc.; pero lo más importante es la falta de calidad, punto de gran interés e importancia para el logro de eficiencia y productividad de los sistemas.

Es claro que si un sistema presenta errores al momento de ser utilizado, ese producto pierde confiabilidad a los ojos del usuario hasta el nivel que podría ser desechado como un producto defectuoso.

Por esta razón la Universidad de las Ciencias Informáticas, como centro que se dedica al desarrollo de programas informáticos, se ve obligada a velar celosamente por la calidad de los productos que se desarrollan, para lograr que los mismos sean competitivos y paralelamente a ello minimizar en lo posible los inconvenientes mencionados anteriormente.




En el proyecto de Sistema de Gestión Penitenciaria (SIGEP), como parte de su desarrollo se gestiona y controla la calidad durante cada una de las fases de desarrollo, desarrollándose para ello un conjunto de acciones necesarias y sistemáticas para proporcionar la confianza de que el sistema que se desarrolla cumple con las exigencias del cliente y satisface los requisitos de calidad establecidos. Este proceso es esencial para asegurar que estas actividades técnicas se lleven a cabo correctamente, dentro de restricciones de costo, tiempo y calidad. Se aplica a lo largo de todo el proceso de Ingeniería de Software, la cual engloba métodos y herramientas de análisis, diseño, codificación y pruebas; revisión de técnicas formales, control de la documentación del software y de los cambios realizados; procedimientos que aseguren un ajuste a los estándares de desarrollo del software, y mecanismos de medida y de información.

Estas tareas de gestión y control de la calidad, en el proyecto SIGEP, actualmente se realizan de forma manual, recopilándose un gran volumen de información, la cual se almacena en documentos, para luego ser tabulada manualmente; lo que provoca la generación de un número considerable de errores mediante su análisis, además de que en ocasiones el proceso puede demorar considerablemente en dependencia del volumen de datos con el que se trabaje. Como consecuencia de estos retrasos en ocasiones no se pueda conocer con exactitud la situación actual en que se encuentra un determinado indicador de calidad que se desee; los integrantes del equipo de calidad se ven en la necesidad de realizar grandes esfuerzos para entregar los resultados de su trabajo en el tiempo definido para ello; inconvenientes, que al final atentan contra la calidad y terminación del producto final que debe ser entregado.

De lo anterior puede plantearse la necesidad de registrar las No Conformidades de forma centralizada siguiendo un estándar que permita tabular esta información y obtener reportes, que posibiliten la toma de decisiones sobre qué acciones realizar en aras de resolver los errores detectados en el menor período de tiempo posible.

Como base para resolver la necesidad que se ha planteado se hace necesaria la modelación e implementación de un sistema que permita, a partir de las no conformidades registradas, realizar un análisis estadístico de los diferentes indicadores de calidad que se miden en el proyecto de Sistema de Gestión Penitenciaria.

Este producto de software, que se propone, tendrá como objetivos específicos:

-  Automatizar el análisis de la información registrada a través de las no conformidades para su posterior consulta.
-  Permitir hacer reportes a partir de la información con que se cuenta para conocer en que estado se encuentran los indicadores de calidad que se miden en el proyecto.
-  Gestionar el acceso a la información de los usuarios que harán uso del sistema.

Dentro de las tareas que se proponen para dar solución a los objetivos planteados están:

1. Realizar un estudio de los indicadores de la calidad que se miden en el proyecto y como se ven estos reflejados en las no conformidades.
2. Analizar y discutir la arquitectura de la aplicación.
3. Desarrollar el Modelado de la aplicación siguiendo la metodología de RUP.
4. Implementar el sistema.

El contenido en este Trabajo de Diploma estará estructurado de la manera siguiente:

Capítulo I

Incluye una Fundamentación Teórica, en la que se aborda qué es la Calidad de Software, Ventajas de controlar la calidad y realizar pruebas de software. Inconvenientes. Importancia de realizar dichos procesos.

Capítulo II

Incluye el modelo de dominio. El análisis del objeto de automatización. Descripción de la información que se maneja, de la aplicación que se propone, así como de todos los requisitos funcionales y no funcionales que esta debe cumplir.

Capítulo III

Incluye la definición del modelo de análisis del sistema y de este el modelo de clases. Describe los diagramas de secuencia del modelo del diseño para cada realización de los casos de uso del primer ciclo



de desarrollo del sistema. Muestra el diagrama de clases del diseño, la descripción de cada una de las clases y por último el diseño de la Base de Datos y la descripción de cada una de las tablas relacionadas en el Diagrama de Entidad Relación.

Capítulo IV

Incluye los Modelos de Implementación y Pruebas. Muestra el diagrama de despliegue y el análisis de pruebas realizadas al sistema.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Propiciar la calidad en el software es una actividad que ha surgido como consecuencia de la fuerte demanda de sistemas de software en todos los procesos que se desarrollan en la actualidad. De allí el esfuerzo que se ha desplegado para obtener software de alta calidad. Es uno de los problemas que se afrontan actualmente en la esfera de la computación. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, producto a que los sistemas que se desarrollan son cada vez más complejos, ya que se van creando nuevas necesidades, lo que ha traído como consecuencia que existan inconvenientes en el desarrollo de los sistemas, entre los que pueden enumerarse: grandes retrasos en la programación, inconsistencia en su funcionamiento, aumento del tiempo y los costos de desarrollo, etc.; pero lo más importante es la falta de calidad, punto de gran interés e importancia para el logro de eficiencia y productividad de los sistemas.

Unido a ello el surgimiento o desarrollo de nuevas tecnologías o ámbitos como la orientación a objetos, mejora de procesos de software, el estudio de componentes, la medición del software, el desarrollo y mantenimiento de software, la gestión y sistemas de calidad de software, normativas y certificación, Internet, Intranet, plantean nuevos retos futuros y una constante actualización de los medios actuales. En cualquier caso siempre se tendrán que evaluar los niveles de concordancia del software, lo que plantea la búsqueda constante de nuevos medios de medición de las especificaciones, es decir, la medición en los distintos ámbitos de la creación del software se plantea como el gran reto.

Se han realizado gran cantidad de investigaciones al respecto entre las que se han planteado muchas interrogantes y aspectos, que serán abordados.

La Calidad, según la ISO (International Standard Organization), se define como la ausencia de deficiencias: "Es la totalidad de aspectos y características de un producto o servicio que se refieren a su capacidad para satisfacer necesidades dadas en la adecuación de sus objetivos". Engloba todos aquellos aspectos del producto que satisfacen las necesidades del cliente y de ese modo proporcionan la satisfacción del producto.

Podría plantearse entonces que la Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia [1], es decir, la concordancia entre los requerimientos funcionales y

no funcionales, explícitamente establecidos, con los estándares de desarrollo y con las características que se esperan de todo software desarrollado profesionalmente.

La Calidad de Software es un proceso medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para uso médico debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación. Aunque la calidad puede medirse después de elaborado el producto, no se recomienda ya que puede resultar muy costoso si se detectan problemas, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

Para realizar mediciones se deben tener en cuenta los siguientes atributos: fiabilidad¹, flexibilidad o adaptabilidad², robustez³, comprensión⁴, modularidad⁵, complejidad⁶, portabilidad⁷, usabilidad⁸, reutilización⁹, eficiencia¹⁰. Conocer los valores de cada uno de estos indicadores posibilitará caracterizar o medir la calidad del software en cuestión.

Por otro lado no puede olvidarse que los objetivos que persigue la Calidad en los sistemas deben estar orientados a incrementar la productividad y satisfacción al trabajo de los profesionales afines, mejorar la calidad del producto del software, proveer técnicas aplicadas para automatizar el manejo de datos, realizar una planeación eficaz de los sistemas, documentar, validar y controlar formalmente la calidad del trabajo realizado y cumplir con los objetivos de la organización en cuanto a productividad de sus sistemas de cómputo.

Para el logro de estos objetivos se hace necesaria la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. La política establecida debe estar sustentada sobre tres principios básicos: tecnológico¹¹, administrativo¹² y ergonómico.¹³

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación.

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición que posee todo software y que son la base para la calidad, el control y el perfeccionamiento de la productividad.

Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, que requiere los siguientes pasos:

- Definir el software que va a ser controlado: Clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- Crear o determinar los métodos de valoración de los indicadores: Métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: Quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados, etc.

El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad. Se basa en normas o estándares genéricos y en procedimientos particulares, entre los que se tienen la familia de normas ISO¹⁴, la IEEE¹⁵, la CMM¹⁶ entre otros.

Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y, lo que es más importante, que se cumplan. La Calidad del Software debe implementarse a lo largo de todo el ciclo de vida, debe correr paralela desde la planificación del producto hasta la fase de producción del mismo.

Para ello deben desarrollarse distintas actividades para la implantación del control de calidad en el desarrollo de software, entre las que se encuentran la aplicación de metodologías y técnicas de desarrollo, la reutilización de procesos de revisión formales, pruebas del software, ajustes a los estándares de desarrollo, control de cambios, mediciones y recopilación de gestión de informes sobre el control de calidad.

Entre los elementos críticos en la calidad de software se tienen las pruebas de software que pueden definirse como: “Un proceso que corre en paralelo al proceso de desarrollo de software, y que se realiza por el convencimiento de que todo sistema debe ser “revisado” con el objetivo de establecer el nivel de

calidad requerido”. [2] En ese proceso se ejecuta el sistema a probar bajo condiciones específicas y se le aplica (eventualmente con apoyo de herramientas especializadas un conjunto de casos de prueba con el objetivo de detectar insatisfacción de los requerimientos planteados). Todas las actividades y sus resultados son documentados.

Este proceso de prueba genera información muy valiosa acerca de la madurez del producto de software que permite tomar decisiones importantes como no comprar o pagar el sistema entregado por falta de calidad, o rehacer uno de los componentes por la misma razón.

Hoy en día, para poder competir en este mercado globalizado y altamente competido de las tecnologías de información y comunicación, la calidad no representa un valor agregado, sino más bien un prerrequisito indispensable para permanecer en el mercado.

Aplicando pruebas de software, se puede reducir el costo y el tiempo global del desarrollo, pues se detectan una buena cantidad de problemas en fases tempranas del proceso.

En ese contexto, algunos factores particulares que dan razón de ser a la prueba de software son [2]:

- Liberar un producto inmaduro trae como consecuencia una labor penosa y costosa de soporte a usuarios insatisfechos, así como altos costos de mantenimiento.
- La insatisfacción del cliente trae como consecuencia pérdida de imagen de la empresa, así como una baja de la credibilidad en la misma.
- La mala calidad de un producto no tarda en verse reflejada en una baja en las ventas.
- En sistemas críticos la falla del sistema puede traer como consecuencia pérdidas de vidas humanas y/o monetarias, y repercusiones económicas (incluso penales) muy severas.

El proceso de prueba debiera comenzar en la fase misma de la definición de requerimientos y terminar con la finalización de la aplicación de los casos de prueba al sistema ejecutable. Esto es, la prueba de software representa un subproceso que permea prácticamente todo el proceso de desarrollo de software:

- Fase de Requerimientos: Se verifica si se siguen estándares, y si los requerimientos son completos, consistentes, rastreables, probables y modulares.
Entradas: Estándar de requerimientos; documento(s) de requerimientos.

Salidas: Recomendaciones para que el documento de requerimientos presente las características deseables mencionadas arriba; documento(s) con diseños preliminares de planes y de casos de prueba de aceptación (ver más abajo).

- **Fase de Diseño:** Verificar si las propuestas de diseño satisfacen los requerimientos, si presentan características deseables, y si se siguieron los estándares de diseño. En particular, verificar si la arquitectura del sistema y de la base de datos son apropiadas, si los diseños tienen alta cohesión y bajo acoplamiento, y si la base de datos tiene la normalización adecuada (no redundancia ni problemas de integridad referencial).

Entradas: Estándar de diseño; documento(s) de diseño; documento(s) de requerimientos.

Salidas: Recomendaciones para que el documento de diseño presente las características deseables mencionadas arriba; documento(s) con refinamientos de los planes y casos de prueba de la fase anterior, mas diseños preliminares de casos de prueba de interacción.

- **Fase de Construcción:** Verificar si se siguen estándares, si los programas son consistentes con el diseño, y si el código es modular, además de cohesivo y bajo en acoplamiento; eventualmente, verificar el nivel y calidad de reuso.

Durante la programación debieran llevarse a cabo de manera gradual varias fases de prueba al sistema:

Pruebas de Unidad: Usualmente realizadas por el mismo desarrollador, en las que prueba constituyentes “atómicos” del sistema que él programó (funciones/objetos/componentes).

Pruebas de Integración: Se verifica si el conjunto de componentes que conforma el sistema completo interactúa correctamente. Por cuestiones de objetividad y eficiencia, es recomendable que esta fase sea llevada a cabo una entidad distinta al grupo desarrollador.

Pruebas de Aceptación: Usualmente llevadas a cabo por el destinatario del software, en las que verifica si el sistema satisface los requerimientos. Por cuestiones de objetividad y eficiencia, es recomendable que esta fase sea llevada a cabo una entidad distinta al grupo desarrollador.

En cada una de estas fases podría aplicarse al menos uno de los siguientes tipos de prueba:



Pruebas de Caja Negra: Se analiza la funcionalidad del software. Basándose en los requerimientos, verificar si el sistema hace lo que debe, y no hace lo que no debe. Las pruebas de stress y de volumen, enfocadas a verificar si el sistema soporta una cantidad mínima de usuarios, o puede almacenar una cierta cantidad de información mínima/máxima, pueden verse como casos especiales de este tipo de prueba.

Entradas: archivos ejecutables; documento(s) de requerimientos.

Salidas: Documento(s) con defectos encontrados.

Pruebas de Caja Blanca: Se analiza el código de la aplicación y la funcionalidad del mismo. Basada en los requerimientos, en el diseño y en el código fuente.

Entradas: Estándar de programación; archivos fuente; documento(s) de diseño

Salidas: Documento(s) con defectos encontrados. Estadísticas acerca del seguimiento al estándar, así como de algunas características del sistema.

Durante todo ese proceso se aplican las denominadas “pruebas progresivas”. Esas pruebas detectan defectos, a los que se da seguimiento para verificar que sean descartados o corregidos. La verificación de que un error haya sido corregido y no “destapó” otros errores se realiza mediante lo que suele llamarse “pruebas regresivas”, consistentes en aplicar ya sea todas o sólo un subconjunto de las pruebas progresivas (implica la definición de algún(os) criterios de selección de casos de prueba).

En la medida que se vayan realizando las pruebas de calidad de software, se obtiene información referente a errores que presenta el software en relación con los requerimientos definidos; estos se reflejan en documentos conocidos como Informes de No Conformidades, los cuales están conformados por no conformidades, las que pueden definirse como insatisfacción o no cumplimiento con lo pactado en un requerimiento del software, es decir, cuando el sistema no refleja o implementa una funcionalidad definida mediante un requerimiento se tiene una no conformidad.

Existen un grupo de herramientas y programas de software que ayudan a obtener y analizar información a partir de la realización de las pruebas de software entre los que pueden mencionarse: Rational ClearQuest and Functional Testing, Rational Funcional Tester, Rational Funcional Tester Extension for Siebel Test Automation, Rational Funcional Tester Extension for Terminal-based Applications, Rational Funcional Tester Plus, Rational Manual Tester, Rational Robot, Rational Test RealTime, IBM Rational Purify for Linux and UNIX, IBM Rational Purify for Windows®, los cuales permiten la creación de informes, gestión

de pruebas y seguimiento de cambios y defectos durante todo el proyecto, la realización de pruebas a aplicaciones de web y de escritorio, la realización de test a lenguajes de script, pruebas de componentes y análisis en tiempo de ejecución de código, pruebas para detectar errores relacionados con el uso de la memoria, de fiabilidad y rendimiento en aplicaciones Windows. Tienen un costo que puede variar entre los 2 485.00 y 16 000.00 euros por cada licencia, en dependencia de las características que posea y de las especificaciones de la licencia que se compre. [3]

Con lo analizado hasta aquí puede apreciarse la importancia que reviste la gestión y control de la calidad de software en el proceso de desarrollo desde sus inicios para detectar las fallas lo antes posible para no liberar productos inmaduros, para reducir los costos y tiempo de desarrollo, para reducir los costos de soporte y la mala imagen que pueden provocar los continuos retrasos y la entrega de un producto defectuoso ante los clientes, para obtener mayor certeza sobre los costos de mantenimiento, para mejorar el proceso de desarrollo, para evitar problemas legales que pudieran surgir.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

Como consecuencia de la situación que actualmente se presenta en el Proyecto de Gestión Penitenciaria con el análisis de la información que se genera del proceso de gestión y control de la calidad de software, y de lo que ello trae como consecuencia en desarrollo y en la entrega del producto final, unido a la importancia que reviste el control de la calidad en todas las fases de un proyecto de software, vistos anteriormente, se propone la creación de un sistema de gestión que analice los datos generados automáticamente y brinde reportes de dicha información en el momento que se desee.

2.2 Propuesta del sistema.

Este sistema, para esta etapa de desarrollo contará con 3 módulos los que están distribuidos de la siguiente forma:

1. Módulo de Informes de No Conformidades: Permitirá gestionar la información de las no conformidades que se generan en el proyecto. Permitirá la creación, modificación y eliminación de los informes de no conformidades y las no conformidades que componen a estos.
2. Módulo de Reportes: Permitirá obtener la información referente a los informes de no conformidades y de los usuarios. Cuenta además con un buscador que le permitirá al usuario realizar consultas y obtener reportes con la información específica que desea. Estos reportes podrán almacenarse en formato pdf, doc, xls y rtf, además de que la aplicación permitirá la impresión de los mismos.
3. Módulo de Administración: Permitirá gestionar la información referente a los usuarios que harán uso de la aplicación. Se han diseñado roles de usuarios, los cuales tendrán un conjunto de permisos que definirán las acciones que pueden realizar los usuarios que pertenezcan al mismo. Los roles podrán ser adicionados, modificados o eliminados. Además se gestionarán los usuarios que harán uso del sistema los cuales podrán ser creados, modificados o eliminados

2.3 Herramientas utilizadas en el desarrollo del sistema.

Para el modelado de esta aplicación se propone utilizar UML¹⁷, ya que está ampliamente difundido y permite especificar, visualizar, construir y documentar artefactos del sistema de un software, lo que posibilita entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Como herramienta para modelar los diagramas en cada unas de las fases del desarrollo se utilizará Rational Rose por ser la mejor solución de modelado visual en el mundo, y la mejor herramienta para traducir requisitos de alto nivel a una arquitectura flexible basada en componentes. Domina el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen, un soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada. Posee una plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Para la realización del sistema se propone organizar los elementos de la aplicación en componentes independientes buscando alcanzar una mayor efectividad a la hora de administrarlos.

Para ello se seguirán los fundamentos del modelo de la programación multicapas¹⁸, ya que esta además de facilitar una administración eficiente de los componentes que la integran, proporciona rapidez a todas las funcionalidades Cliente-Servidor y la magnitud de la aplicación lo exige, posibilita desarrollos en paralelo, la creación de aplicaciones más robustas debido al encapsulamiento, mantenimiento y soporte más sencillo, mayor flexibilidad, ya que brinda la posibilidad de agregar nuevos módulos para dotar al sistema de nuevas funcionalidades, y posibilita una alta escalabilidad, permitiendo manejar múltiples peticiones. Tal y como plantea la arquitectura de esta tecnología, los componentes primarios de la aplicación serán divididos y programados por separados y en tiempo de ejecución serán unidos. De forma tal que si una de las capas definidas sufre cambios, no se vean afectados el resto de las capas ni el resultado final del producto.

Como gestor de base de datos se utilizará SQL Server 2000 que aunque no es considerado el mejor sistema gestor de bases de datos relacionales (RDBMS), aventajado en algunos aspectos como la disponibilidad de versiones para múltiples plataformas, por otros como Oracle y MySQL, Cuenta con un lenguaje para programar procedimientos almacenados, permite definir tablas, índices, vistas, etc., es distribuido y escalable, con soporte para 32 procesadores y 64 GB de RAM, es más fácil de usar que el Oracle y más potente que MySQL, brinda facilidades de replicación, seguridad administrada según perfiles configurables, que puede ser sobre cuentas de usuarios propias o integrada con las de Windows, ofrece además mecanismos de salva y restauración, y posibilidad de importar y exportar los datos en múltiples formatos.

Se propone además la utilización de servicios web¹⁹, ya que son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos web²⁰ estándar, como XML²¹, SOAP²², UDDI²³ o WSDL²⁴. El objetivo final es la creación de un directorio de online de Web Services, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.

Para la implementación se propone usar el entorno de desarrollo integrado que nos brinda Microsoft Visual Studio 2005, ya que permite crear aplicaciones de línea de negocio, aplicaciones para windows, la web y dispositivos móviles, además del desarrollo de aplicaciones cliente-servidor. Utiliza como base la plataforma.net²⁵, la está constituida por un conjunto de tecnologías que permiten desarrollar y distribuir software en forma de servicios (Servicios Web) que son suministrables remotamente a través de internet, capaces de comunicarse y combinarse, correr sobre diferentes sistemas operativos, y ser desarrollados en diversas plataformas de desarrollo y lenguajes de programación. Además cuenta con ADO.Net que es lo más reciente en una extensa línea de tecnologías de acceso a bases de datos. Se ha diseñado siguiendo específicamente unas directrices más generales y menos orientadas a la base de datos. ADO.NET reúne todas las clases que permiten el manejo de datos. Estas clases representan los objetos que contienen datos y que muestran las capacidades normales de las bases de datos: índices, ordenación, vistas. Aunque ADO.NET es la solución definitiva para las aplicaciones de base de datos de .NET, destaca por un diseño global que no se centra tanto en las bases de datos como el modelo ADO. Se persigue, con .NET, un nuevo modelo informático que sustituya el actual, compuesto por dispositivos y sitios Web individuales conectados a través de Internet, por un conjunto de recursos, ordenadores y servicios cooperando entre sí para satisfacer las demandas de los usuarios.



Entre las ventajas que presenta esta tecnología se tienen: modelo de programación orientado a objetos consistente, eliminación del “infierno de las DLLs” (las versiones nuevas de las DLLs pueden coexistir con las viejas, simplificando mucho la instalación y desinstalación de software), ejecución multiplataforma, integración de lenguajes (C#, Visual Basic, Visual C++, J#), aunque pueden añadirse otros y la gestión automática de memoria.

Dentro de los lenguajes con que cuenta este entorno de desarrollo se ha decidido utilizar C#.Net porque Microsoft ha creado C# que combina algunas de las características más avanzadas de Java con algunas de las más potentes de C y C++, construido especialmente para adaptarse de manera natural al framework.net.

Las principales características que identifican al lenguaje C# son:

Sencillez: Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

Modernidad: Incorpora en el propio lenguaje elementos que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.

Orientación a objetos: Es más puro pues no admiten funciones ni variables globales, todo el código y datos han de especificarse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

Orientación a componentes: Su sintaxis incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

Seguridad de tipos: Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.

Eficiencia: El código incluye numerosas restricciones para alcanzar su seguridad y no permite el uso de punteros. A diferencia de Java, en el mismo es posible saltarse dichas restricciones manipulando objetos a través de punteros.

2.4 Modelo de Dominio.

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema.

El modelo del dominio se considera en RUP un subconjunto del llamado modelo de objetos del negocio. Se puede desarrollar un modelo de objetos del negocio enfocado a la explicación de los productos, entregables o eventos que son importantes en el negocio. Tiene como objetivo comprender y describir las clases más importantes dentro del contexto del sistema. [4]

2.4.1 Modelo de Objetos.

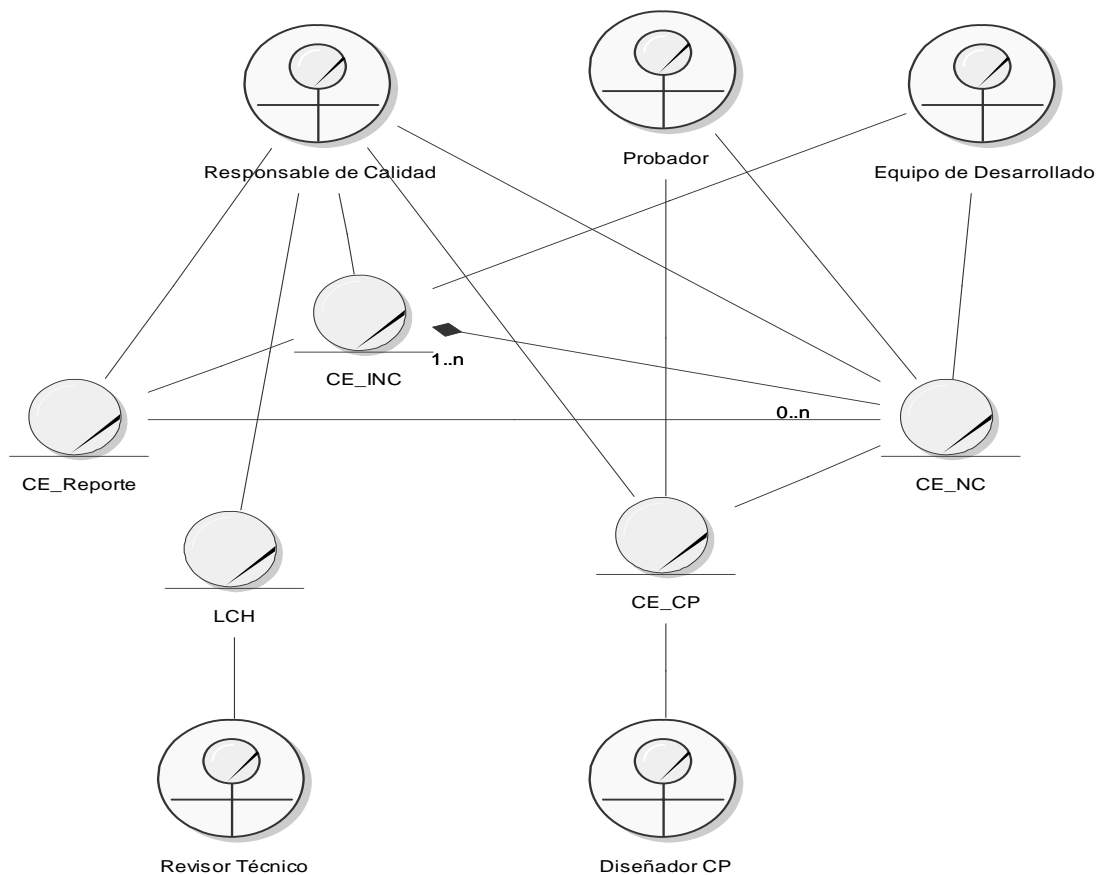


Figura 1. Modelo de Objeto.

2.4.2 Descripción de los Trabajadores.

Trabajador	Justificación
Responsable de Calidad	Es una persona orientada al detalle. Asegura que la aplicación producida se ajusta a las especificaciones y está razonablemente libre de errores. Proporciona una metodología para realizar las pruebas. Coordina las pruebas de calidad internas, las pruebas de aceptación del cliente y pilotos. Evalúa los resultados que se obtienen en las pruebas de calidad. Debe poseer conocimientos mínimos sobre RUP, UML, calidad de software, ingeniería de software y negocio
Diseñador de Pruebas	Diseña los casos de prueba. Evalúa y documenta el resultado de las pruebas realizadas al software. Define listas de chequeo. Debe poseer conocimientos mínimos sobre RUP, UML y pruebas de software.
Revisor Técnico	Chequea que los artefactos generados se ajusten a las pautas y lineamientos establecidos para su confección. Debe poseer conocimientos mínimos sobre RUP y UML.
Probador	Ejecuta las pruebas diseñadas. Anota los resultados obtenidos. Debe poseer conocimientos mínimos sobre el negocio. Habilidades mínimas de computación.
Equipo de Desarrollo	Son los demás integrantes que participan en el proyecto y que se relacionan con los informes de no conformidad y las no conformidades, exceptuando al personal de calidad. Deben poseer los conocimientos afines con el rol que desempeñen.

2.5 Especificación de los Requisitos de Software.

2.5.1 Requisitos Funcionales.

R.1. Gestionar Informes de No Conformidades (INC).

1.1. Insertar Informes de No Conformidad (INC).

1.2. Eliminar Informes de No Conformidad (NC).

R.2. Gestionar No Conformidades (NC).

2.1. Crear No Conformidades.

2.1.1. Notificar por Correo al Responsable de Resolver la No Conformidad que tiene No Conformidades Asignadas.

2.2. Modificar No Conformidades.

2.2.1. Notificar por Correo al Responsable de Resolver la No Conformidad que tiene No Conformidades Asignadas.

2.3. Eliminar No Conformidades.

R.3. Generar Reportes.

3.1. Mostrar los Reportes con toda la Información de los INC.

3.1.1. Permitir obtener consultas por parámetros.

3.2. Mostrar los Reportes con toda la Información de los Usuarios.

R.4. Autenticar Usuario.

R.5. Cambiar la Contraseña.

R.6. Gestionar Roles.

6.1. Crear Roles.

6.2. Modificar Permisos de un Rol.

6.3. Eliminar un Rol.

R.7. Gestionar Usuarios.

7.1. Insertar Nuevos Usuarios.

7.2. Modificar Datos de algún Usuario.

7.3. Eliminar Usuarios.

2.5.2 Requisitos No Funcionales.

- **Apariencia o interfaz externa.**

La interfaz de este sistema debe ser sencilla y de fácil uso con rápida respuesta, para agilizar la búsqueda de información y elaboración de los documentos a procesar. Debe tener diferentes opciones para mostrar y confeccionar documentos, así como para visualizar los reportes. Contendrá un menú para que actúe con el usuario en dependencia de sus necesidades.

- **Usabilidad.**

El sistema debe ser de fácil manejo para los usuarios estén vinculados al proceso de Control y Gestión de la Calidad de Software.

- **Rendimiento.**

La aplicación está concebida para un ambiente cliente/servidor así que los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información. Para lograr una respuesta rápida debe garantizarse una velocidad de conexión aceptable, específicamente una Red de Área Local (LAN). El tiempo de respuesta y de búsqueda de información así como la generación de reportes y documentos debe ser relativamente corto, teniendo en cuenta el volumen de información que se requiera analizar para ello; implicando esto que el acceso a la base de datos debe ser lo más disponible, rápido y consistente posible.

- **Soporte.**

Para garantizar el soporte, se realizarán encuentros con los integrantes de calidad, con el fin de explicarles como funciona la herramienta. Se requiere de un servidor para instalar el servicio web y SQL Server para garantizar un manejo eficiente con la base de datos correspondiente.

- **Portabilidad.**

El sistema deberá ser compatible con los sistemas operativos de la familia Windows.

- **Requerimientos de Hardware.**

Se requiere que las computadoras de los usuarios tengan tarjeta de red y al menos 256 MB de RAM. En el caso del Web Service y de la Base de Datos se recomienda que se ubiquen en una estación de trabajo con 512 MB o 1 GB de memoria en aras de gestionar con rapidez las peticiones de los usuarios que harán uso del sistema.

- **Requerimientos de Software.**

La estación de trabajo que actuará como servidor donde estarán el Web Service y la base de datos, así como cada una de las estaciones que harán uso de este servicio web deberán tener instalado el sistema operativo Windows XP o Server 2003. Se recomienda que las estaciones de trabajo de los clientes tengan instalado Microsoft Outlook, ya que el sistema hará uso de este para enviar y recibir la mensajería de correo electrónico.

En el caso del servidor debe tener instalado Microsoft SQL Server 2000, como gestor de base de datos; además, Internet Information Server para poder gestionar las peticiones del cliente; el Framework 2.0 de Microsoft Visual Studio, ya que esta aplicación se desarrolló con el lenguaje de programación C#.Net.

- **Seguridad.**

Permitir que a la información solo acceda quien está autorizado para ello y dándole el uso al cual está dirigida. Identificar a la persona a la que se autoriza, a quien se le concede permiso para determinadas tareas (son los procesos de identificación, autorización y asignación de perfiles y roles). Mantener la integridad de la información, es decir, que no se perderá durante su almacenamiento o transporte. Permitir que cuando se intente borrar cualquier documento o información pueda existir una opción de advertencia para verificar el hecho.

- **Disponibilidad.**

Significa que a los usuarios autorizados se les garantizará el acceso al sistema y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para realizar las acciones correspondientes que se le asignan en el sistema. Esto se logra a través de la creación de roles de usuarios, los cuales tendrán asignados permisos de acción sobre cada información manejada por el sistema, para lo cual se requiere la autenticación del usuario. Si no se autentifica no podrá hacer uso del sistema.

- **Confiabilidad.**

La información manejada por el sistema está protegida de acceso no autorizado y de divulgación.

- **Integridad**

La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción, de la misma forma será considerada igual a la fuente o autoridad de los datos.

2.6 Definición de los Actores del Sistema.

Estos no son más que trabajadores que participan directamente en el negocio y/o nuevos actores que surgen para ejecutar funcionalidades surgidas debidas a la presencia de requisitos especiales (Ver Tabla 2).

Tabla 2. Actores del Sistema.

Actores	Justificación
Usuario	Es quien hará uso de la aplicación. Podrá Crear, Actualizar y Eliminar Roles, Usuarios, No Conformidades (NC), Informes de No Conformidad (INC); Obtener Reportes de INC y de Usuarios; partiendo de los Permisos que estén definidos para los Roles a los que pertenezca.

2.6.1 Especificación de los Casos de Uso.

Tabla 3. Especificación del Caso de Uso “Gestionar INC”.

CU-1	Gestionar INC
Actor	Usuario
Descripción	Permite Insertar y Eliminar Informes de No Conformidad (INC). Para adicionar un INC se verifica que este no exista, de existir se notifica un error. Para eliminar un INC este no puede tener No Conformidades (NC). Permite modificar y eliminar NC, ejecutándose el CU Gestionar NC.
Referencia	R 1, R 1.1, R 1.2

Tabla 4. Especificación del Caso de Uso “Gestionar NC”.

CU-2	Gestionar NC
Actor	Usuario

Descripción	Permite insertar, eliminar y modificar no conformidades (NC). Para adicionar una NC se verifica que esta no exista en el INC al que se desea adicionar. Cuando se desea modificar una NC se verifica que no exista otra NC similar dentro del mismo INC, al que pertenecerá esta. En caso de existir algún error con las verificaciones se notifica. Cada vez que se adicione o actualice una NC se enviará un correo al responsable de resolver la NC, si así lo desea quien lleve a cabo la acción.
Referencia	R 2, R 2.1, R 2.1.1, R 2.2, R 2.2.1, 2.3

Tabla 5. Especificación del Caso de Uso “Generar Reportes”.

CU-3	Generar Reportes
Actor	Usuario
Descripción	Permite obtener reportes de no conformidades (NC), agrupadas por informes de no conformidad (INC) y de los usuarios. Una vez que se tienen los datos se pueden realizar consultas según criterios, en el caso de los Reportes de INC los criterios serían: Nombre INC, Nombre de quien conformó el INC, Rango de Fechas en los que se creó el INC, Nombre del Responsable de Resolver la NC, Rango de Versiones de las NCs, Estado de la NC, Categoría de la Falla de la NC, Nivel de Importancia de la NC, Rango de Fechas de creación de la NC.
Referencia	R 3, R 3.1, R 3.1.1, R 3.2

Tabla 6. Especificación del Caso de Uso “Autenticar Usuario”.

CU-4	Autenticar Usuario
Actor	Usuario
Descripción	Permite verificar que el usuario y contraseña suministrados para poder acceder al sistema sean correctos. Si los datos suministrados son incorrectos se notificará el error y se esperará a que los datos sean ingresados nuevamente y se repetirá el proceso de verificación.
Referencia	R 4

Tabla 7. Especificación del Caso de Uso “Cambiar Contraseña”.

CU-5	Cambiar Contraseña
Actor	Usuario
Descripción	Permite cambiar la contraseña del usuario, pero para ello este debe ingresar el usuario, contraseña actual, contraseña nueva y la confirmación de la contraseña nueva. Si los datos suministrados son correctos se cambiará la contraseña actual por la nueva, si no se notificará el error y se esperará a que sean suministrados correctamente los datos.
Referencia	R 5

Tabla 8. Especificación del Caso de Uso “Gestionar Roles”.

CU-6	Gestionar Roles
Actor	Usuario
Descripción	Permite insertar roles con sus permisos, modificar los permisos de los roles y eliminar roles. Solo se podrán eliminar aquellos roles a los cuales no pertenezca ningún usuario
Referencia	R 6, R 6.1, R 6.2, R 6.3

Tabla 9. Especificación del Caso de Uso “Gestionar Usuarios”.

CU-7	Gestionar Usuarios
Actor	Usuario
Descripción	Permite insertar, eliminar y modificar la información de los usuarios. Para adicionar un usuario se debe tener en cuenta que este no exista; para eliminarlo se verifica que no tenga informes de no conformidad creados, no conformidades asignadas y que no esté actualmente conectado. En caso de que exista algún problema en las verificaciones se notifica.
Referencia	R 7, R 7.1, R 7.2, R 7.3

2.7 Diagrama de Casos de Uso del Sistema.

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

Se ha dividido el diagrama de casos de uso del sistema por paquetes para estructurar las funcionalidades de los mismos (Ver Figura 2)

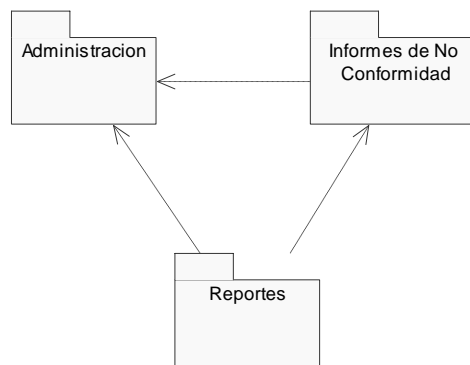


Figura 2. Estructuración del Sistema por Paquetes.

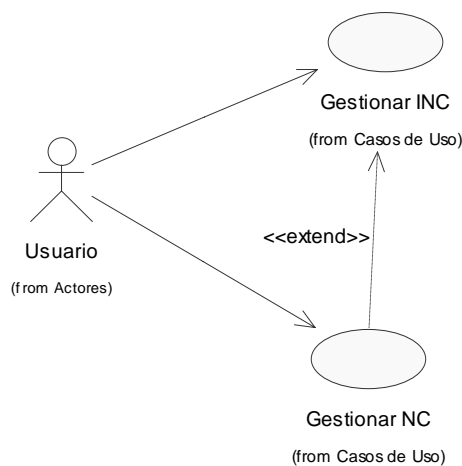


Figura 3. Diagrama de Casos de Uso del Sistema del Paquete Informe de No Conformidades.

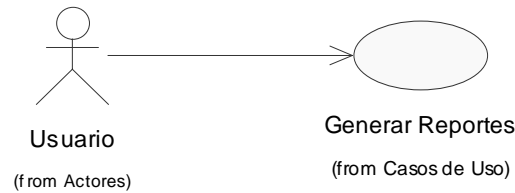


Figura. 4. Diagrama de Casos de Uso del Sistema del Paquete Reportes.

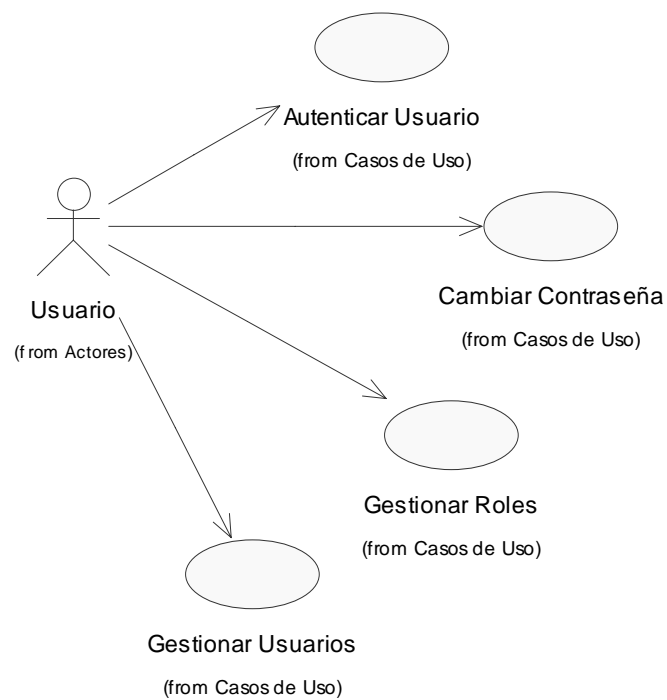


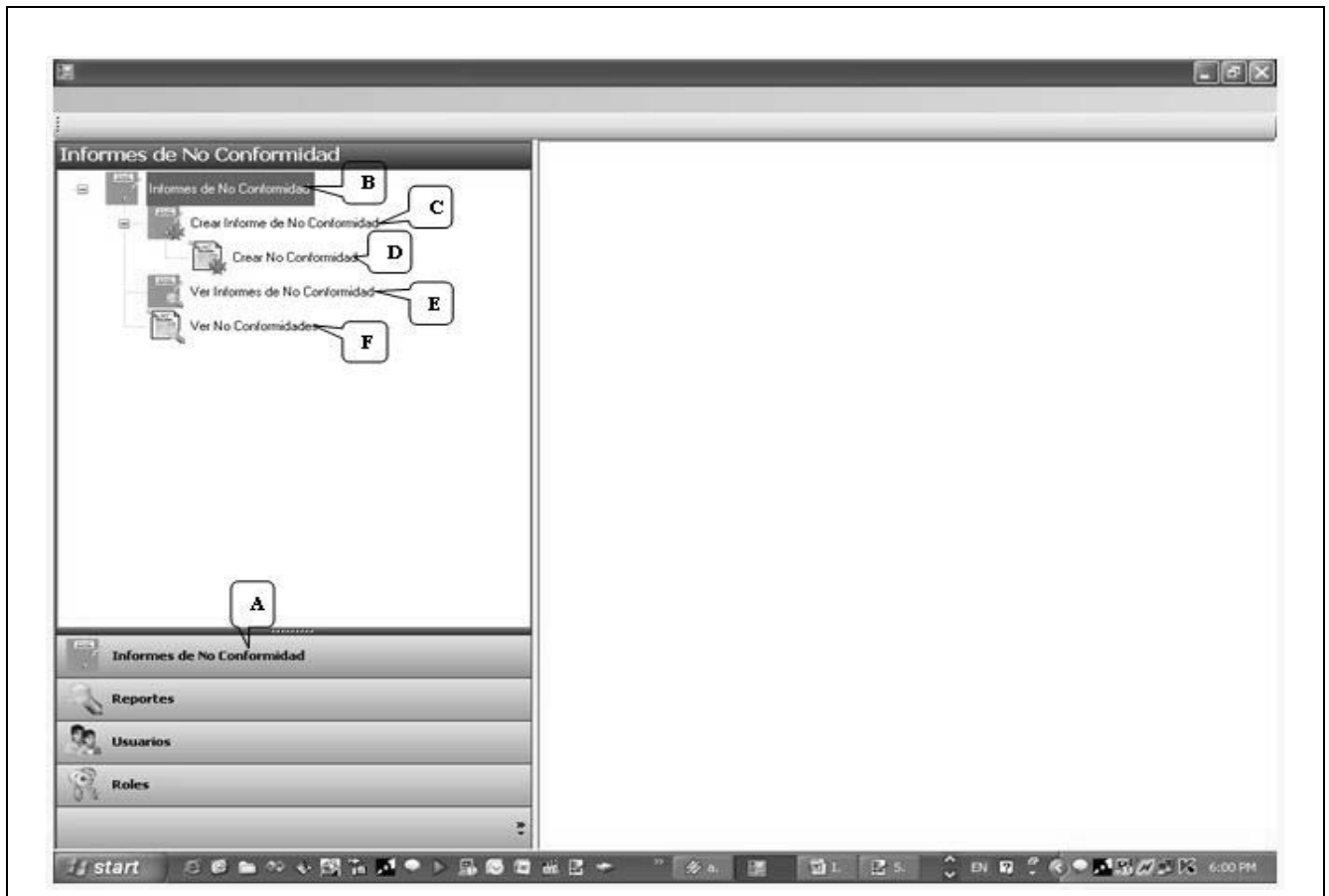
Figura 5. Diagrama de Casos de Uso del Sistema del Paquete Administración.

2.8 Descripción Detallada de Casos de Uso del Sistema.

No es más que una plantilla estándar donde se muestra una descripción para ayudar a comprender los Casos de Uso del Sistema.

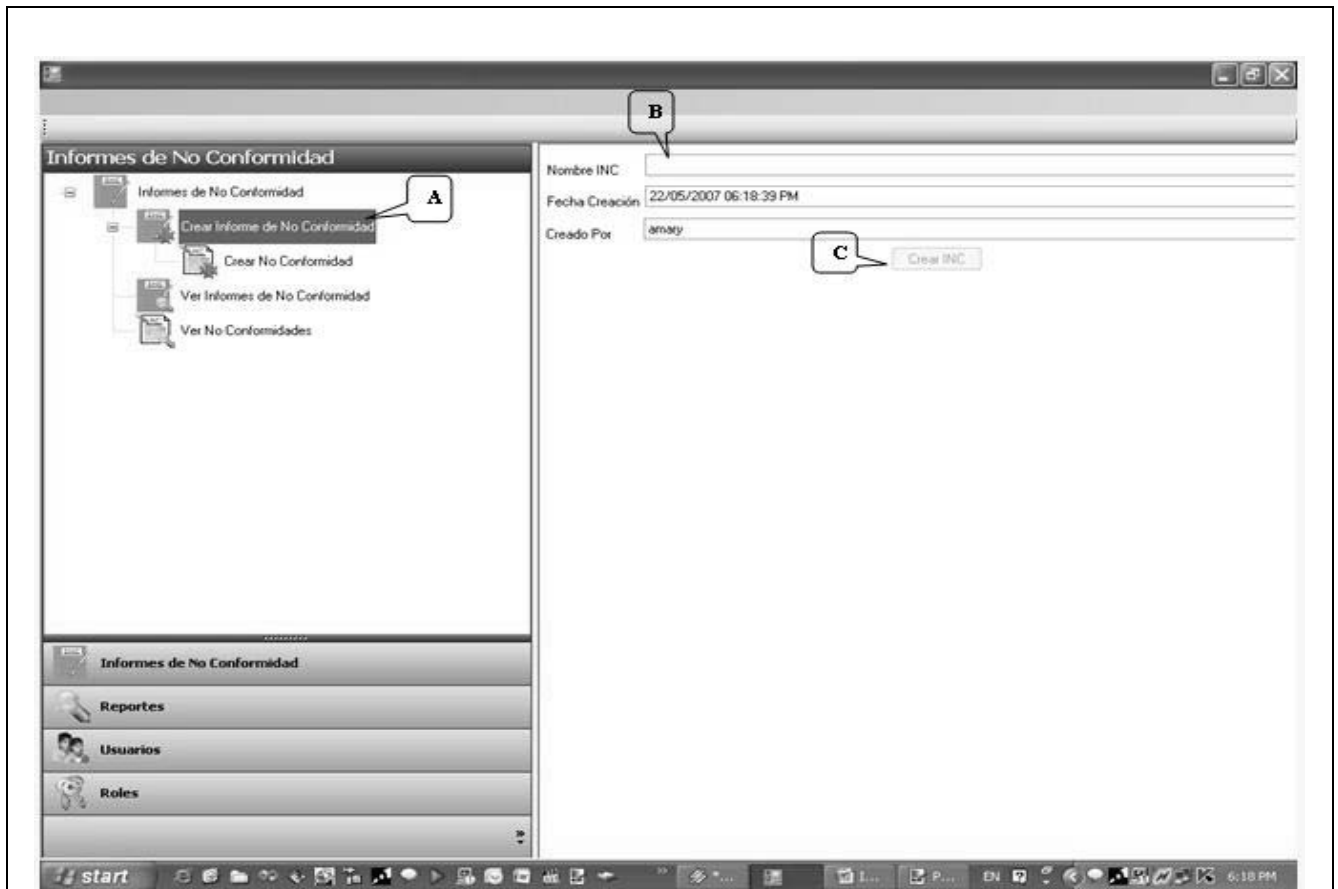
Tabla 10. Descripción Textual del caso de uso “Gestionar Informes de No Conformidades”.

Caso de uso	
CU-1	Gestionar Informes de No Conformidades (INC)
Propósito	Crear y Eliminar Informes de No Conformidades (INC).
Actores	Usuario
Resumen	El Usuario puede adicionar un nuevo informe de no conformidades, para ello deberá escribir el nombre que desea ponerle (no puede repetirse). Además podrá eliminar cualquiera de los INC que haya creado si no tienen No Conformidades (NC). Puede adicionar, eliminar o modificar NCs. En el caso de la eliminación de NCs, solo podrá eliminar aquellas que pertenezcan a INC que hayan sido creados por el Usuario. En el caso de la modificación de NCs podrá modificar aquellas que pertenezcan a INC que hayan sido creados por el Usuario o aquellas que hayan sido asignadas al Usuario.
Referencias	R 1, R 1.1, R 1.2, R 2, R 2.1, R 2.1.1, R 2.2, R 2.2.1, R 2.3
Precondiciones	<ul style="list-style-type: none"> • El Usuario debe de estar autenticado. • El Usuario debe pertenecer a un Rol que tenga entre sus permisos “Crear Informes de No Conformidad” y/o “Ver Informes de No Conformidad”. • Tener instalado Outlook.
Interfaz I “Principal”	



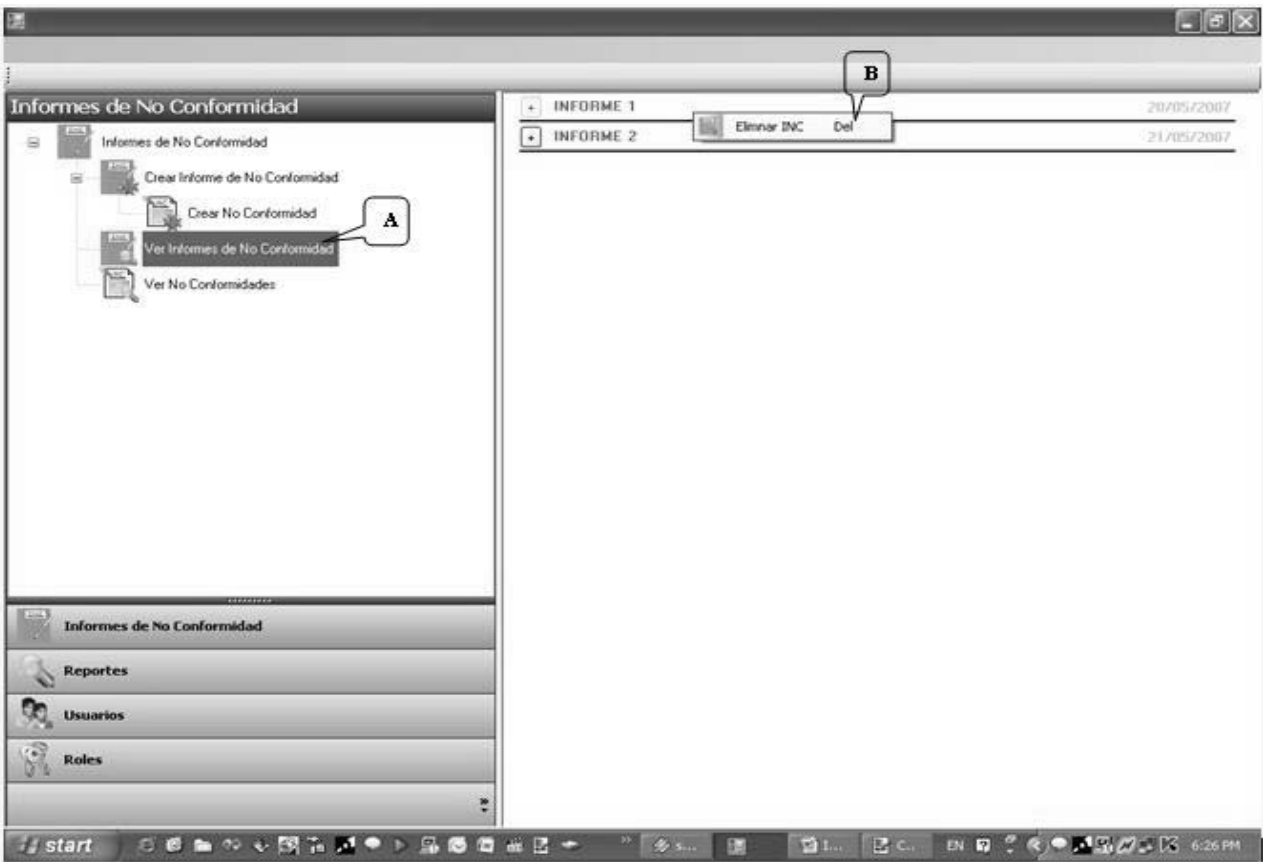
- ✚ A: Permite gestionar los Informes de No Conformidad (INC) y las No Conformidades(NC).
- ✚ B: Carga las Opciones que puede realizar el Usuario con los INCs y las NCs.
- ✚ C: Muestra una Interfaz para crear un Informe de No Conformidad.
- ✚ D: Muestra una Interfaz para crear una No Conformidad.
- ✚ E: Muestra en una Interfaz las INC creados por el Usuario con las NC pertenecientes al mismo.
- ✚ F: Muestra las NC que deben ser resueltas por el Usuario.

Interfaz II “Crear Informe de No Conformidad”



- ✚ A: Muestra la Interfaz “Crear Informe de No Conformidad”.
- ✚ B: Campo para escribir el nombre del Informe de No Conformidad.
- ✚ C: Botón para adicionar el Informe de No Conformidad al Sistema.

Interfaz III “Eliminar Informe de No Conformidad”



Informes de No Conformidad

- Informes de No Conformidad
 - Crear Informe de No Conformidad
 - Crear No Conformidad
 - Ver Informes de No Conformidad
 - Ver No Conformidades

INFORME 1	Eliminar INC	Del	20/05/2007
INFORME 2			21/05/2007

Informes de No Conformidad

Reportes

Usuarios

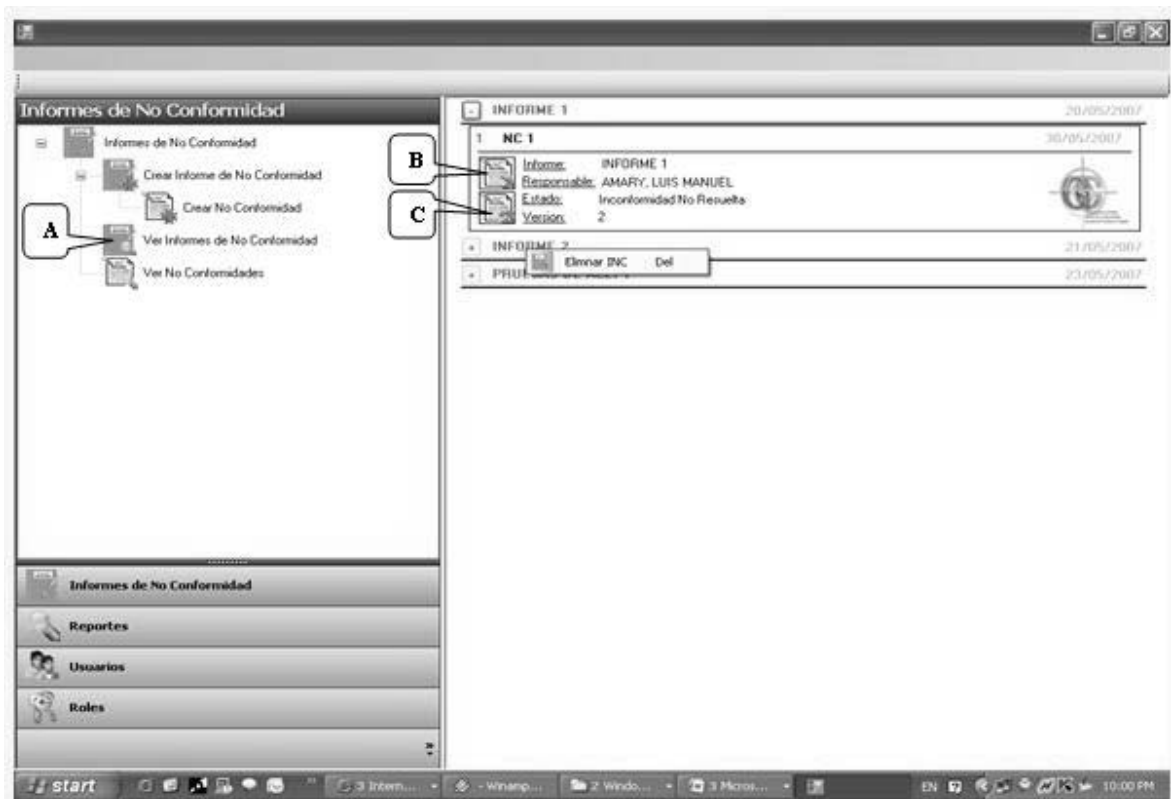
Roles

start 6:26 PM

⚡ A: Muestra una Interfaz con los INC creados por el Usuario y las NC de cada uno.

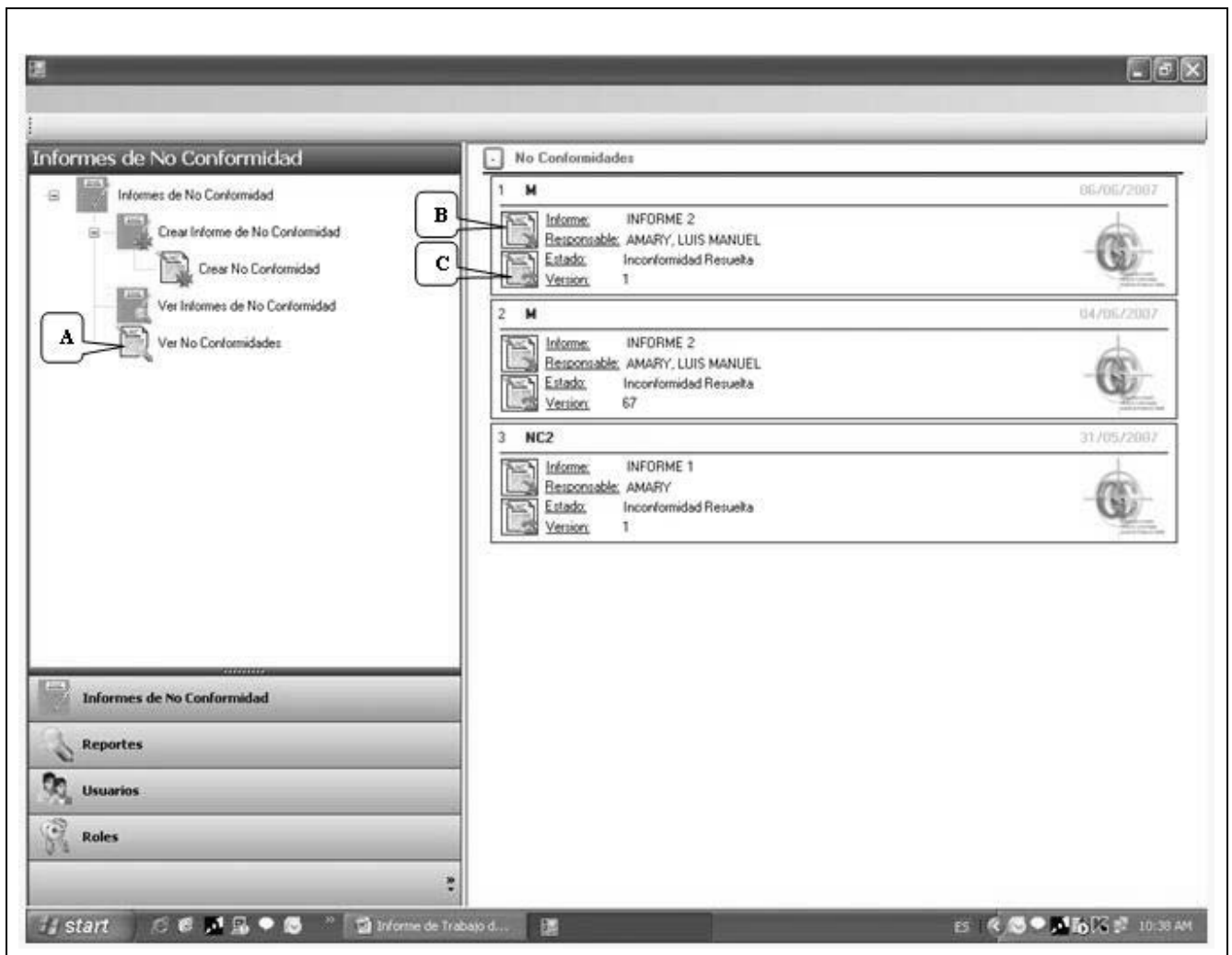
⚡ B: Muestra una Interfaz de confirmación de que se desea eliminar el INC.

Interfaz IV “Ver Informes de No Conformidad”



- ✚ A: Muestra una Interfaz con los INC creados por el Usuario y las NC de cada uno.
- ✚ B: Muestra una Interfaz para confirmar la eliminación de la NC.
- ✚ C: Muestra una Interfaz para modificar la NC.

Interfaz V “Ver No Conformidades”



- ✚ A: Muestra una Interfaz con las NC que se han asignado al Usuario.
- ✚ B: Muestra una Interfaz para confirmar la eliminación de la NC.
- ✚ C: Muestra una Interfaz para modificar la NC.

Acción del Actor	Respuesta del Sistema
1. Selecciona "Informes de No Conformidad". Interfaz I (A).	2. Muestra una Interfaz con las opciones que puede realizar el usuario con los

	INCs y las NCs. Interfaz I (B).
<p>3. Selecciona una Opción. Interfaz I (C, D, E, F).</p>	<p>4. Muestra la Interfaz correspondiente.</p> <ul style="list-style-type: none"> • En caso de que sea “Crear Informe de No Conformidad” Ver Sección “Crear Informe de No Conformidad”. Interfaz II. • En caso de que sea “Eliminar Informe de No Conformidad” Ver Sección “Eliminar Informe de No Conformidad”. Interfaz III. • En caso de que sea “Crear No Conformidad” Ver Sección “Crear No Conformidad”. CU Gestionar No Conformidades. • En caso de que sea Interfaz III o IV (B) [Eliminar NC] Ver Sección “Eliminar No Conformidad”. CU Gestionar No Conformidades. • En caso de que sea Interfaz III o IV (C) [Modificar NC] Ver Sección “Modificar No Conformidad”. CU Gestionar No Conformidades. • En caso de que sea “Ver Informes de No Conformidad” Ver Sección “Ver Informes de No Conformidad”. Interfaz



	<p>IV.</p> <ul style="list-style-type: none"> • En caso de que sea “Ver No Conformidades” Ver Sección “Ver No Conformidades”. Interfaz V.
Sección “Crear Informe de No Conformidad”	
	<p>5. Muestra la Interfaz “Crear Informe de No Conformidad” (INC). Interfaz II</p>
<p>6. Escribe el Nombre del INC que será creado. Interfaz II (B).</p> <p>7. Presiona el Botón “Crear INC”.</p>	<p>8. Verifica que la información suministrada sea correcta.</p> <ul style="list-style-type: none"> • En caso de que no se pueda Insertar el Informe de No Conformidad Ver Sección “Datos Incorrectos”. <p>9. Crea el Informe de No Conformidad en el Sistema.</p> <p>10. Notifica el Resultado de la operación.</p> <p>11. Muestra la Interfaz correspondiente para crear una No Conformidad en el Informe de No Conformidad que se ha adicionado.</p>
Sección “Eliminar Informe de No Conformidad”	
	<p>5. Verifica que el Informe de No Conformidad que desea ser eliminado no tenga No Conformidades.</p>

	<ul style="list-style-type: none"> • En caso de que no se pueda ser eliminado el Informe de No Conformidad Ver Sección “Datos Incorrectos”. <p>6. Muestra la Interfaz “Eliminar INC”. Interfaz III (B).</p>
<p>7. Presiona la opción “Eliminar INC”. Interfaz III (B).</p> <ul style="list-style-type: none"> • En caso de que no presione la opción “Eliminar INC”. Interfaz III (B). Ver Sección “No Eliminar INC”. 	<p>8. Muestra una Ventana de Confirmación “Preguntando si se desea eliminar el INC”.</p>
<p>9. Presiona el Botón “Sí”.</p> <ul style="list-style-type: none"> • En caso de que presione el Botón “No” Ver Sección “No Eliminar INC”. 	<p>10. Elimina el INC del Sistema.</p> <p>11. Notifica el resultado de la Operación.</p>
<p>Sección “Ver Informes de No Conformidad”.</p>	
	<p>5. Muestra una interfaz con los Informes de No Conformidad que han sido creados por el Usuario y las No Conformidades pertenecientes al mismo. Interfaz IV.</p> <p>En caso de que no tenga INC creados muestra un mensaje “No tiene Informes de No Conformidad Creados.”</p>
<p>Sección “Ver No Conformidades”</p>	

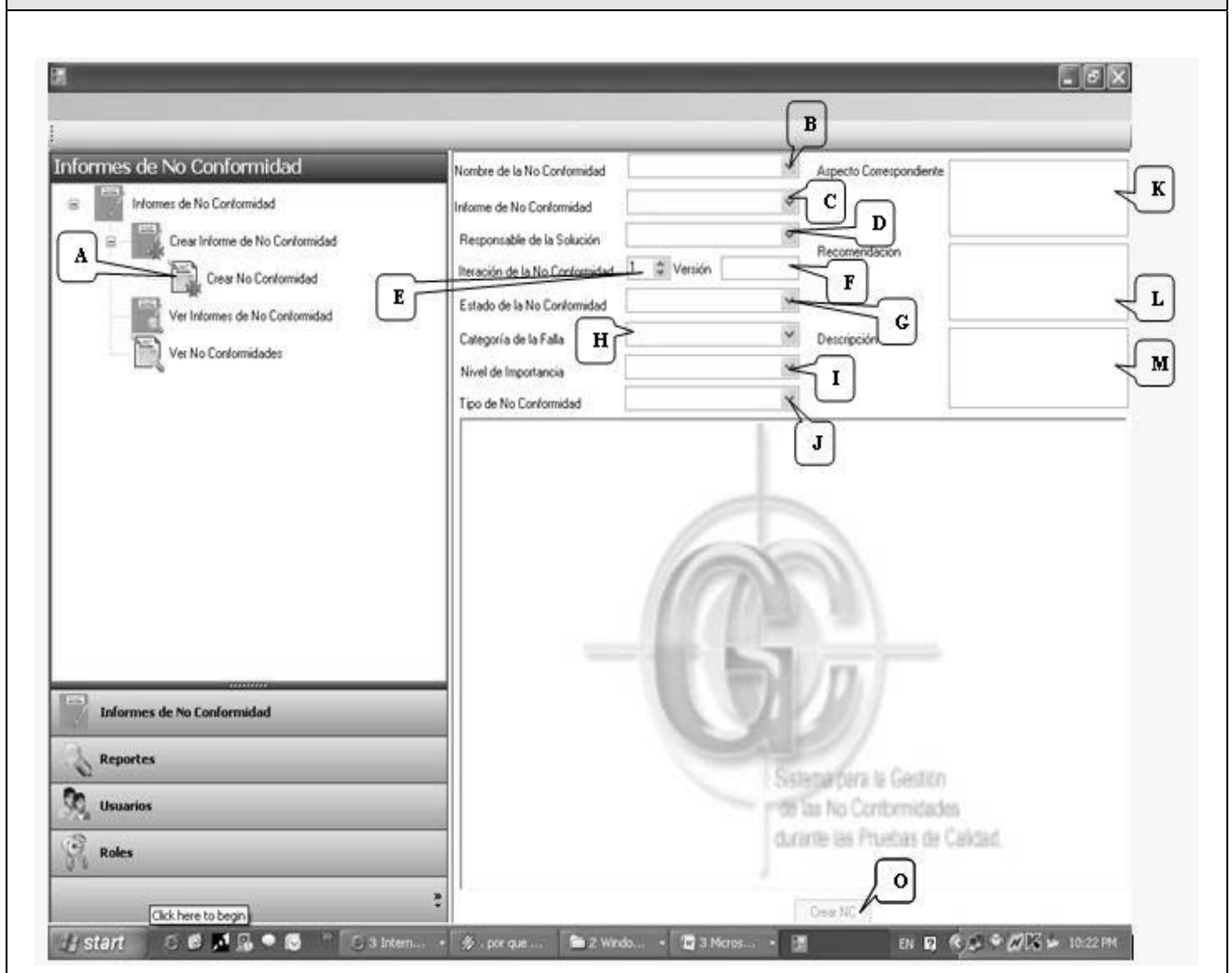
	<p>5. Muestra una interfaz con las No Conformidades que deben ser resueltas por el Usuario. Interfaz V.</p> <p>En caso de que no tenga NC asignadas muestra un mensaje “No tiene No Conformidades Asignadas.”</p>
Flujo Alternativo	
Sección “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	Notifica el Error Ocurrido.
Sección “No Eliminar INC”.	
	Cancela el Proceso.
Puntos de Extensión	
Prioridad	Alta

Tabla 11. Descripción Textual del caso de uso “Gestionar NC”.

Caso de uso	
CU-2	Gestionar NC
Propósito	Adicionar, modificar y eliminar las No Conformidades (NC) pertenecientes a los Informes de No Conformidad (INC).
Actores	Usuario
Resumen	El caso de uso permite la creación de No Conformidades (NC), Las cuales deben pertenecer a un informe de no conformidad (INC) y tener a un usuario asignado para su resolución. No puede existir más de una NC similar en un mismo INC. Estas también podrán ser modificadas o eliminadas, pero solo por quien creó el INC al cual pertenece la misma y por el responsable de resolverla. Si se diese el caso de que el responsable de resolverla no fue quien creó el INC al que pertenece la NC, solo podrá modificarla.

Referencias	R 2, R 2.1, R 2.1.1, R 2.2, R 2.2.1, R 2.3
Precondiciones	<ul style="list-style-type: none"> • El Usuario debe de estar autenticado. • El Usuario debe pertenecer a un Rol que tenga entre sus permisos “Crear Informes de No Conformidad” y/o “Crear No Conformidad” y/o “Ver Informes de No Conformidad”. • Tener instalado Outlook.

Interfaz I “Crear No Conformidad”



- ✚ A: Carga una interfaz para crear una No Conformidad (NC).
- ✚ B: Campo para seleccionar o escribir el nombre de la NC que será adicionada.
- ✚ C: Campo para seleccionar el nombre del Informe de No Conformidad (INC) al que pertenecerá la NC ue será creada.
- ✚ D: Campo para seleccionar el nombre del responsable de resolver la NC.
- ✚ E: Campo para seleccionar la iteración de la NC que se revisa.
- ✚ F: Campo para escribir la versión.
- ✚ G: Campo para seleccionar el estado en el que se encuentra la NC.
- ✚ H: Campo para seleccionar la categoría de falla que presenta la NC.
- ✚ I: Campo para seleccionar el nivel de importancia de la NC
- ✚ J: Campo para seleccionar el tipo de NC.
- ✚ K: Campo para describir otros aspectos.
- ✚ L: Campo para escribir las recomendaciones.
- ✚ M: Campo para describir el problema que ha generado la NC.
- ✚ N: Campo para cargar una imagen que represente el error de forma visual.
- ✚ O: Permite adicionar la NC al sistema.

Interfaz II “Modificar No Conformidad”



- ✚ Es una interfaz similar a la descrita anteriormente.
- ✚ O: Permite actualizar una NC en el sistema.

Sección “Crear No Conformidad”

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Escribe o selecciona el Nombre de la No Conformidad (NC). Interfaz I (B). 2. Selecciona el nombre del INC al que pertenecerá la NC que será creada. 	<ol style="list-style-type: none"> 15. Verifica que los datos suministrados sean válidos. <ul style="list-style-type: none"> • En caso de que no se pueda crear la NC. Ver Sección “Datos Incorrectos”

<p>Interfaz I (C).</p> <p>3. Selecciona el nombre del responsable de resolver la NC. Interfaz I (D).</p> <p>4. Selecciona la iteración de la NC que se revisa. Interfaz I (E).</p> <p>5. Define la versión de la NC que se revisa Interfaz I (F).</p> <p>6. Selecciona el estado de la NC. Interfaz I (G).</p> <p>7. Selecciona la categoría de la NC. Interfaz I (H).</p> <p>8. Selecciona el nivel importancia de la NC. Interfaz I (I).</p> <p>9. Selecciona el tipo de la NC. Interfaz I (J).</p> <p>10. Escribe otros aspectos que desee. Interfaz I (K).</p> <p>11. Si lo desea escribe recomendaciones. Interfaz I (L).</p> <p>12. Escribe la causa que dio origen al surgimiento de la NC. Interfaz I (M).</p> <p>13. Si lo desea carga una imagen que represente el error que ha dado origen a la NC. Interfaz I (N).</p> <p>14. Presiona el Botón “Crear NC”. Interfaz I</p>	<p>16. Adiciona la NC al sistema.</p> <p>17. Pregunta si se desea enviar una notificación por correo a las responsables de resolver la NC.</p>
---	--

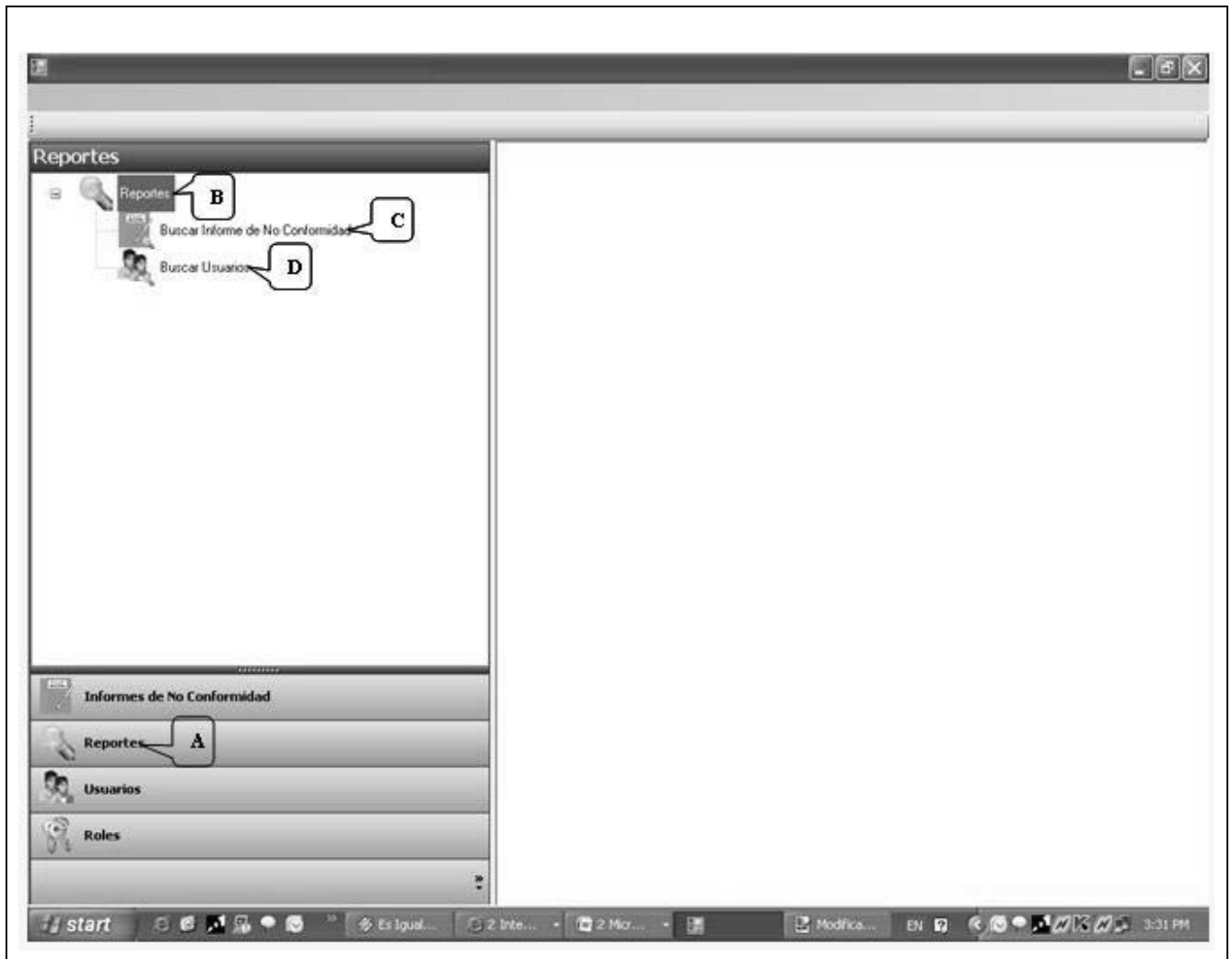
(O).	
<p>18. Presiona el Botón “Sí”.</p> <p>En caso de que presione “No” se cierra la ventana de confirmación.</p>	<p>19. Notifica el resultado de la operación.</p> <p>20. Vuelve a la Interfaz donde se encontraba anteriormente.</p>
Sección “Modificar No Conformidad”	
	<p>5. Carga una interfaz con los datos actuales de la No Conformidad (NC), deshabilitando los campos Nombre de la NC, Nombre del INC y Responsable si el Usuario no fue quien creó el INC al que pertenece la NC. Interfaz II.</p>
<p>6. Modifica los campos que desee.</p> <p>7. Presiona el Botón “Actualizar NC”. Interfaz II (O).</p>	<p>8. Verifica que la información suministrada sea correcta.</p> <ul style="list-style-type: none"> • En caso de que la NC no pueda ser actualizada Ver Sección “Datos Incorrectos”. <p>9. Actualiza los datos de la NC en el sistema.</p> <p>10. Pregunta si se desea enviar una notificación por correo a las responsables de resolver la NC.</p>
<p>11. Presiona el Botón “Sí”.</p> <p>En caso de que presione “No” se cierra la</p>	<p>12. Notifica el resultado de la operación.</p> <p>13. Vuelve a la Interfaz donde se encontraba</p>



ventana de confirmación.	anteriormente.
Sección “Eliminar No Conformidad”	
	5. Muestra una interfaz de confirmación de eliminación de la NC.
6. Presiona el botón “Sí”. <ul style="list-style-type: none"> En caso de que presione el botón “No”. Ver Sección “No Eliminar NC” 	
Flujo Alternativo	
Sección “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	Notifica el Error Ocurrido.
Sección “No Eliminar NC”	
	Cancela la operación.
Puntos de Extensión	
Prioridad	Alta

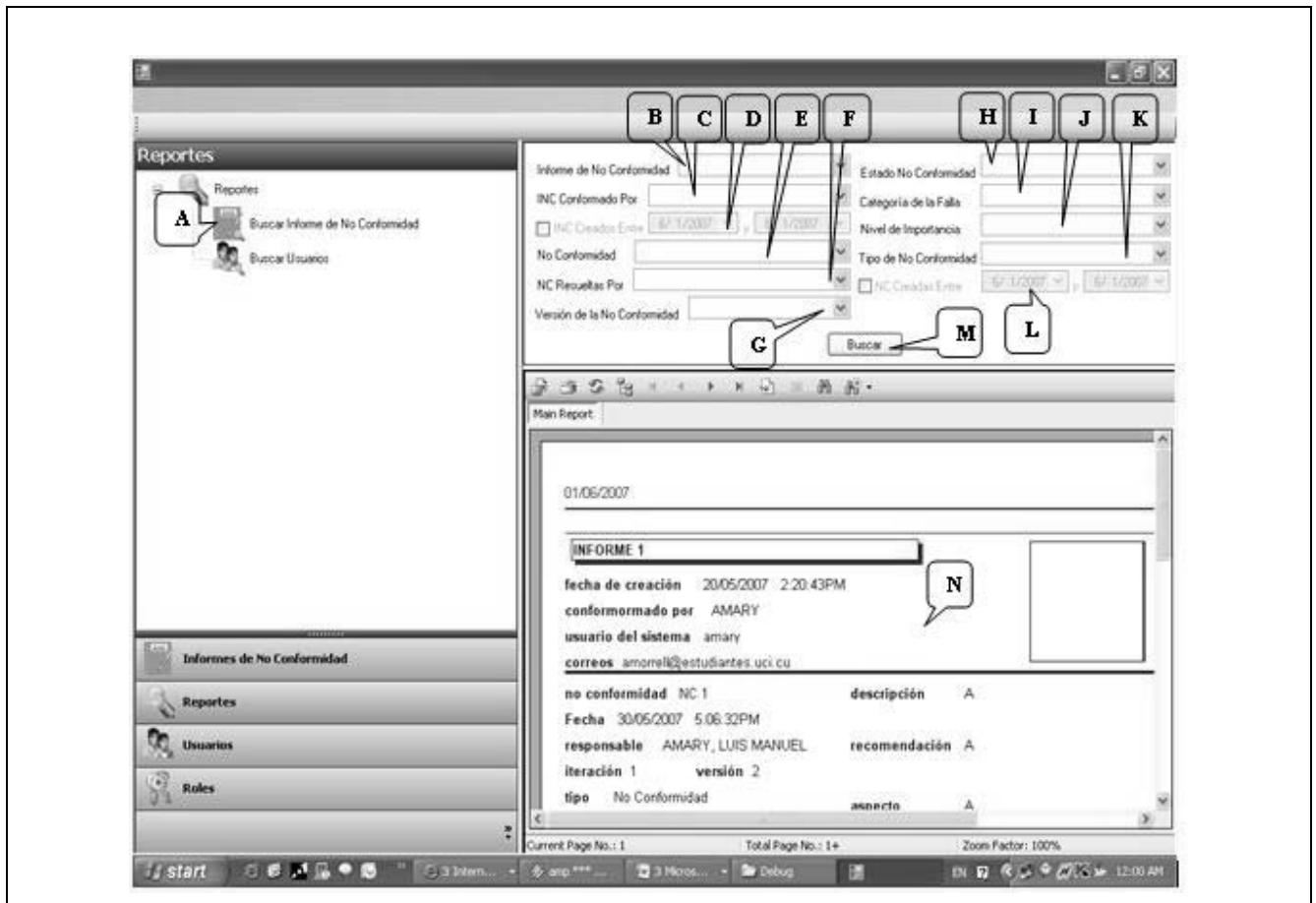
Tabla 12. Descripción Textual del caso de uso “Gestionar Reportes”.

Caso de uso	
CU-3	Gestionar Reportes
Propósito	Obtener Reportes a partir de la información con que se cuenta.
Actores	Usuario
Resumen	Podrán obtenerse reportes a partir de los Informes de No conformidades y de los Usuarios del sistema. Inicialmente el sistema mostrará todos los datos los que luego el Usuario podrá reducir a sus necesidades de acuerdo a determinados criterios de búsqueda que se han definido.
Referencias	R 3, R 3.1, R 3.1.1, R 3.2, R 3.2.1
Precondiciones	<ul style="list-style-type: none"> • Que el Usuario esté autenticado.
Interfaz I “Reportes”	



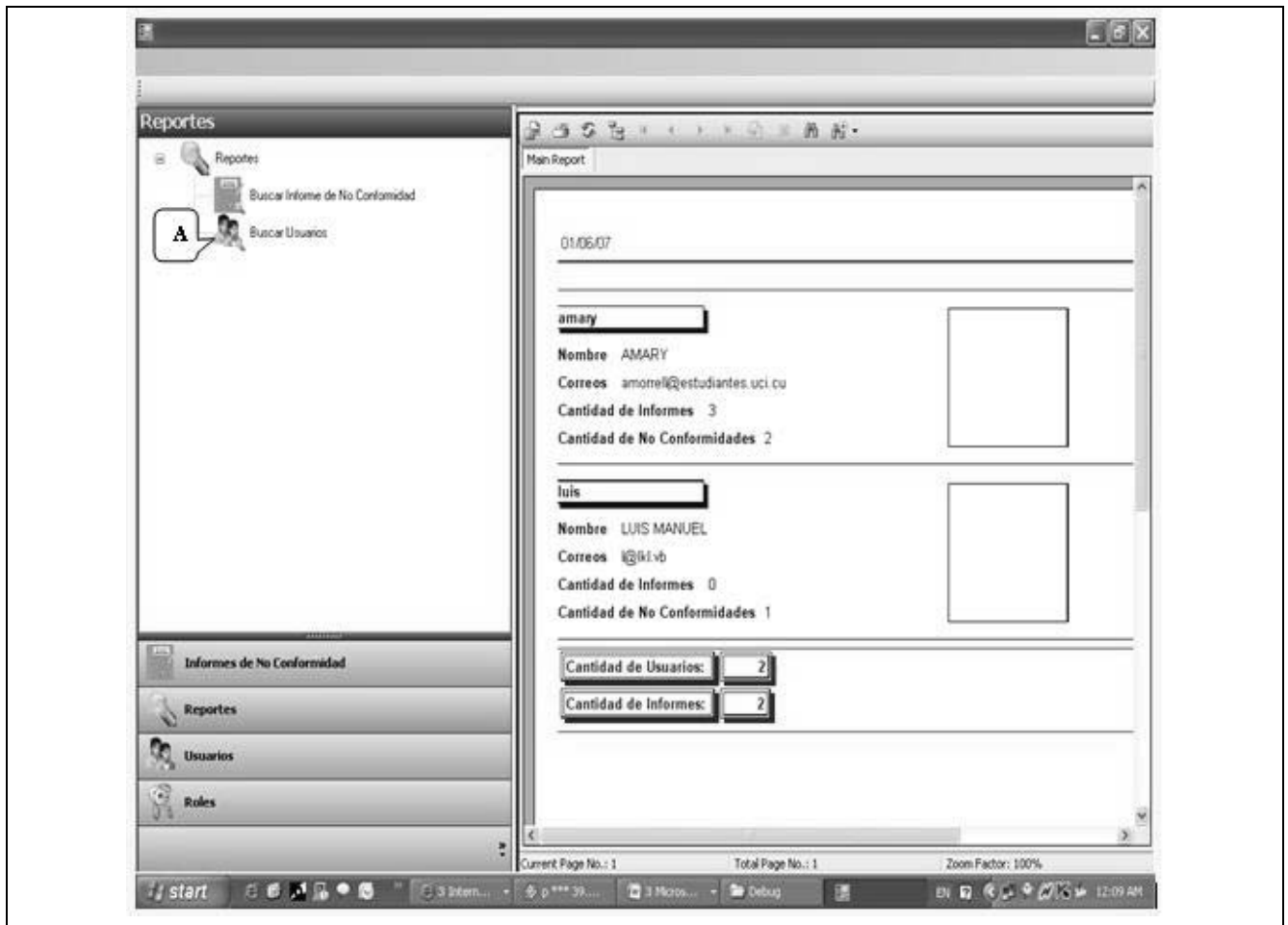
- ✚ A: Muestra una interfaz que contiene todas las opciones que podrá realizar el Usuario con los reportes.
- ✚ B: Muestra todas las opciones que podrá realizar el Usuario con los reportes.
- ✚ C: Muestra una interfaz para trabajar con reportes de Informes de No Conformidaes.
- ✚ D: Muestra una interfaz para trabajar con reportes de Usuarios.

Interfaz II “Buscar Informes de No Conformidad”



- ✚ A: Muestra una interfaz para realizar consultas con los Informes de No Conformidad (INC).
- ✚ B: Muestra todos los nombres de los Informes de No Conformidad que han sido creados.
- ✚ C: Muestra todos los nombres de los usuarios del sistema que han creado Informes de No Conformidad (INC).
- ✚ D: Proporciona un rango de Fechas para buscar los INC que han sido creados.
- ✚ E: Muestra todos los nombres de las No Conformidades que han sido creadas.
- ✚ F: Muestra todos los nombres de los usuarios del sistema a los cuales se les han asignado No Conformidades (NC).
- ✚ G: Proporciona las versiones para buscar las NCs.
- ✚ H: Muestra todos estados de las No Conformidades (NC).
- ✚ I: Muestra todas las categorías de las No Conformidades (NC).
- ✚ J: Muestra todos los niveles de importancia de las No Conformidades (NC).
- ✚ K: Muestra los tipos de No Conformidades (NC).
- ✚ L: Proporciona un rango de Fechas para buscar las NC que han sido creadas.
- ✚ M: Permite ejecutar una búsqueda de los datos que cumplan con los criterios definidos.
- ✚ N: Muestra el resultado de la consulta realizada.

Interfaz III “Buscar Usuarios”



⊕ A: Muestra una interfaz para realizar consultas con los Usuarios del Sistema.

Sección “Obtener Reporte”

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “Reportes”. Interfaz I (A).	2. Muestra una interfaz con las opciones disponibles para obtener el reporte. Interfaz I (B).
3. Selecciona el tipo de reporte. Interfaz II (A) para INC e Interfaz III (A) para	4. Carga toda la información que existe referente al tipo de reporte seleccionado.

Usuario.	5. Muestra el reporte. Interfaz II (N) , similar en la Interfaz III .
6. Selecciona los criterios de búsqueda por los que desea obtener los resultados. (descritos anteriormente). 7. Presiona el botón "Buscar".	8. Realiza una consulta sobre el reporte con los criterios definidos. 9. Muestra el resultado de la operación.
Flujo Alternativo	
Puntos de Extensión	
Prioridad	Alta

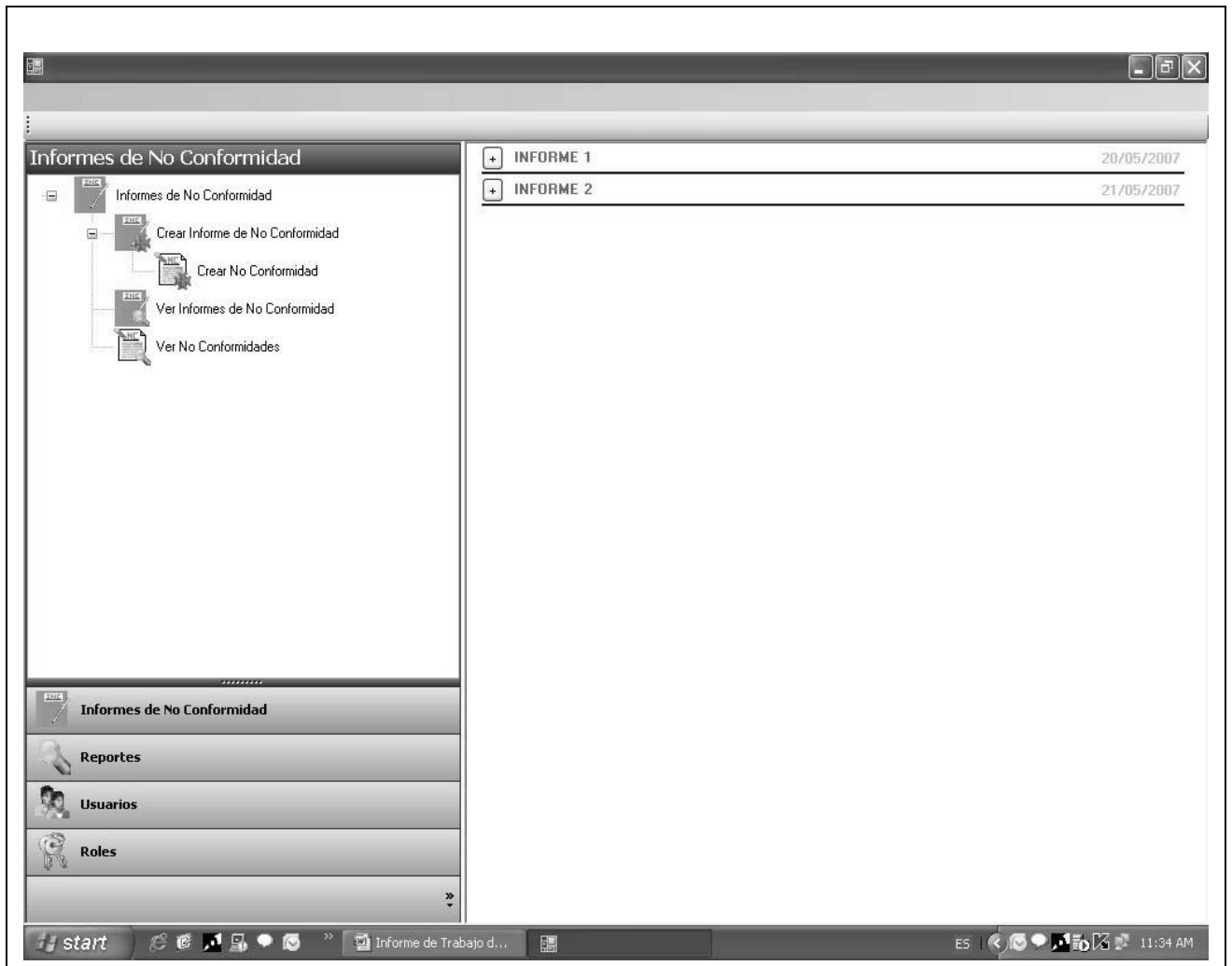
Tabla 13. Descripción Textual del caso de uso "Autenticar Usuario".

Caso de uso	
CU-4	Autenticar Usuario
Propósito	Que el Usuario pueda autenticarse para acceder al sistema
Actores	Usuario
Resumen	El Caso de Uso se inicia cuando el Usuario desea hacer uso del sistema. Este (Sistema) le muestra la Interfaz de Autenticación para validar la validez de este (Usuario).
Referencias	R 4
Precondiciones	
Interfaz I "Autenticarse"	



- ✚ A: Campo para Escribir el usuario de la aplicación.
- ✚ B: Campo para escribir la Contraseña de acceso al sistema.
- ✚ C: Permite el acceso al sistema.
- ✚ D: Cierra la aplicación.

Interfaz II "Principal"



Acción del actor	Respuesta del sistema
1. Ejecuta la aplicación	2. Muestra la Interfaz de Autenticarse. Interfaz I
3. Introduce el Usuario Interfaz I (A) y la Contraseña Interfaz I (B) . 4. Presiona el Botón “Entrar” Interfaz I (C) . En caso de que presione el Botón	5. Valida los Datos y verifica que son correctos. <i>En caso de que los datos sean Incorrectos Ver Sección “Datos</i>

<p>“Cancelar” Interfaz I (D) Ver Sección “Cancelar Entrada al Sistema”.</p>	<p><i>Incorrectos”</i></p> <p>6. Muestra la Interfaz del Sistema. Interfaz II.</p> <p>7. Si el Usuario tiene INCs los Carga, sino carga las NCs.</p>
Flujos Alternativos	
Sección “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	5.1 Informa al Usuario que los Datos son Incorrectos.
Sección “Cancelar Entrada al Sistema”	
Acción del actor	Respuesta del Sistema
4.1 Presiona el Botón “Cancelar” Interfaz I (D).	5 Cierra la Aplicación.
Puntos de Extensión	
Prioridad	Alta

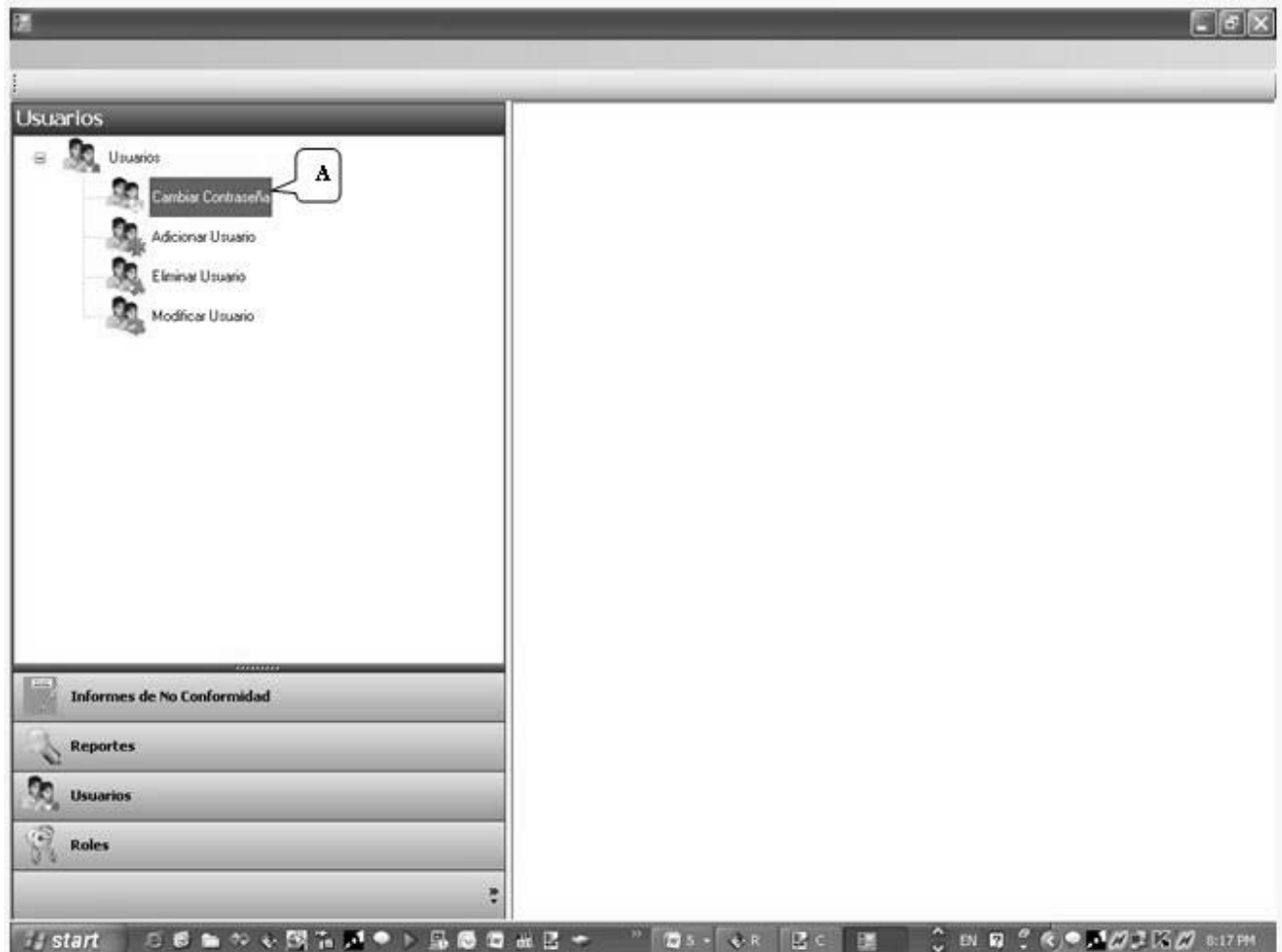
Tabla 14. Descripción Textual del caso de uso “Cambiar Contraseña”.

Caso de uso	
CU-5	Cambiar Contraseña
Propósito	Cambiar la contraseña actual del usuario por una nueva.
Actores	Usuario
Resumen	Permite al Usuario a partir de su usuario del sistema y contraseña actual, cambiar su contraseña por una nueva. Para ello el usuario y la contraseña actual deben ser válidos y la contraseña nueva y la confirmación de esta deben coincidir.
Referencias	R 5.

Precondiciones

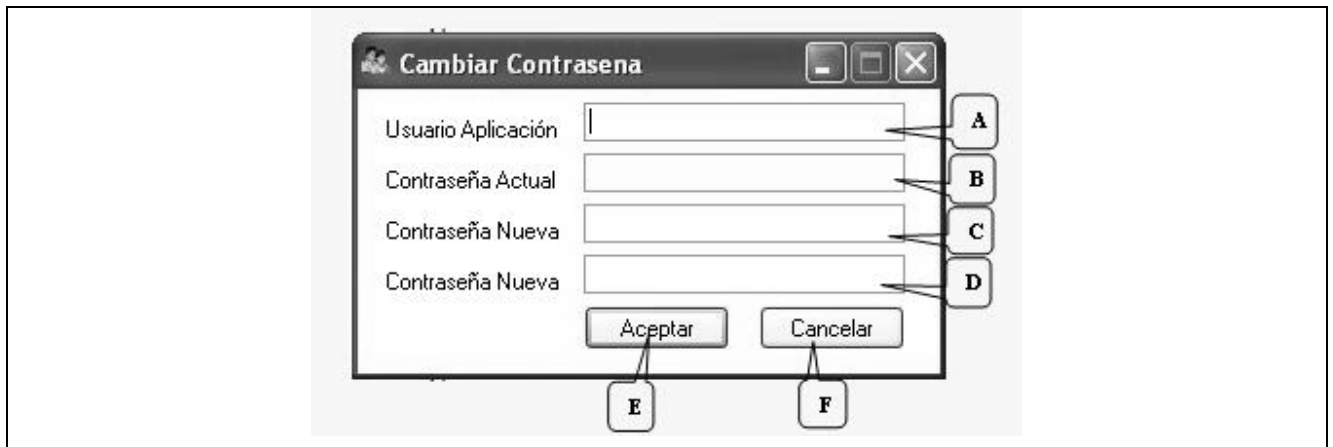
- El Usuario tiene que estar autenticado.

Interfaz I “Principal”



⊕ A: Opción que permite mostrar la interfaz para cambiar la contraseña.

Interfaz II “Cambiar Contraseña”



- ✚ A: Campo para escribir el usuario de la aplicación.
- ✚ B: Campo para escribir la contraseña actual.
- ✚ C y D: Campo para escribir la contraseña nueva.
- ✚ E: Actualiza la Contraseña del Usuario en el Sistema.
- ✚ F: Cancela el cambio de la contraseña.

Acción del actor	Respuesta del sistema
1. Selecciona la opción “Cambiar Contraseña”. Interfaz I (A).	2. Muestra la Interfaz para el cambio de contraseña. Interfaz II.
3. Escribe el usuario de la aplicación. Interfaz II (A). 4. Escribe la contraseña actual. Interfaz II (B). 5. Escribe la contraseña Nueva Interfaz II (C, D). 6. Presiona el Botón “Aceptar”. Interfaz II	7. Valida los Datos y Verifica que son correctos. <ul style="list-style-type: none"> • En caso de que la verificación sea incorrecta Ver Sección “Datos Incorrectos”. 8. Actualiza la contraseña en el sistema. 9. Cierra la interfaz Cambiar Contraseña

<p>(E).</p> <ul style="list-style-type: none"> En caso de que presione el Botón “Cancelar” Ver Sección “Cancelar Cambio de Contraseña”. 	<p>Interfaz II</p> <p>10. Muestra un Mensaje notificándole al Usuario el resultado de la operación.</p>
Flujo Alternativo	
Sección “Datos Incorrectos”	
Acción del actor	Respuesta del sistema
	7.1 Muestra un mensaje de error.
Sección “Cancelar Cambio de Contraseña”	
Acción del actor	Respuesta del sistema
	8. Cierra la Interfaz de Cambiar Contraseña.
Puntos de Extensión	
Prioridad	Baja

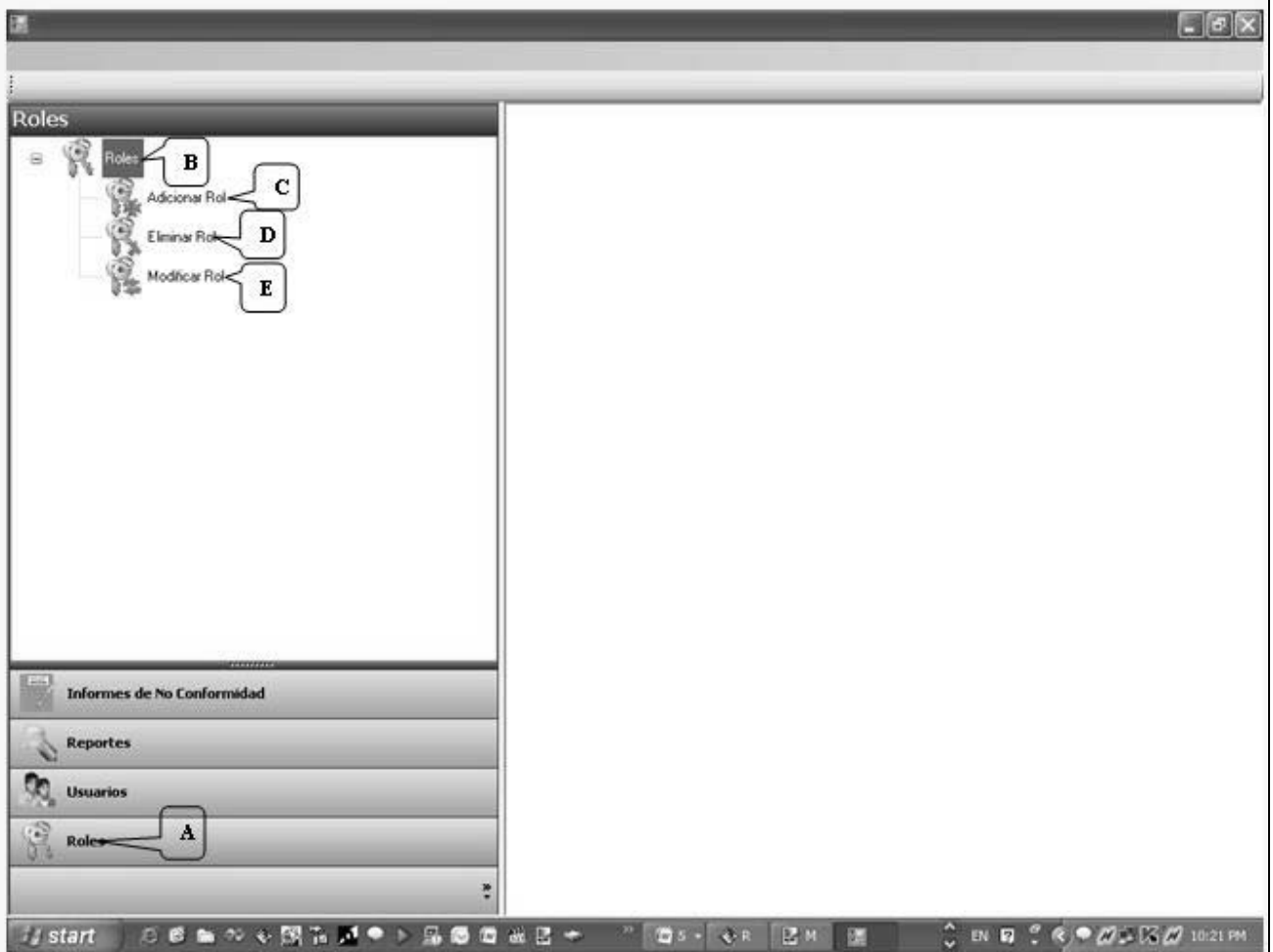
Tabla 15. Descripción Textual del caso de uso “Gestionar Roles”.

Caso de uso	
CU-6	Gestionar Roles
Propósito	Insertar, Modificar, Eliminar Roles de Usuarios del Sistema.
Actores	Usuario
Resumen	El usuario podrá Insertar, Modificar o Eliminar Roles en el sistema. Cada Rol tendrá Permisos (los que podrán modificarse) y Usuarios Asociados. No se Pueden Insertar Roles con el mismo nombre. Para Eliminar un Rol, este no puede tener Usuarios Asociados.
Referencias	R 6, R 6.1, R 6.2, R 6.3

Precondiciones

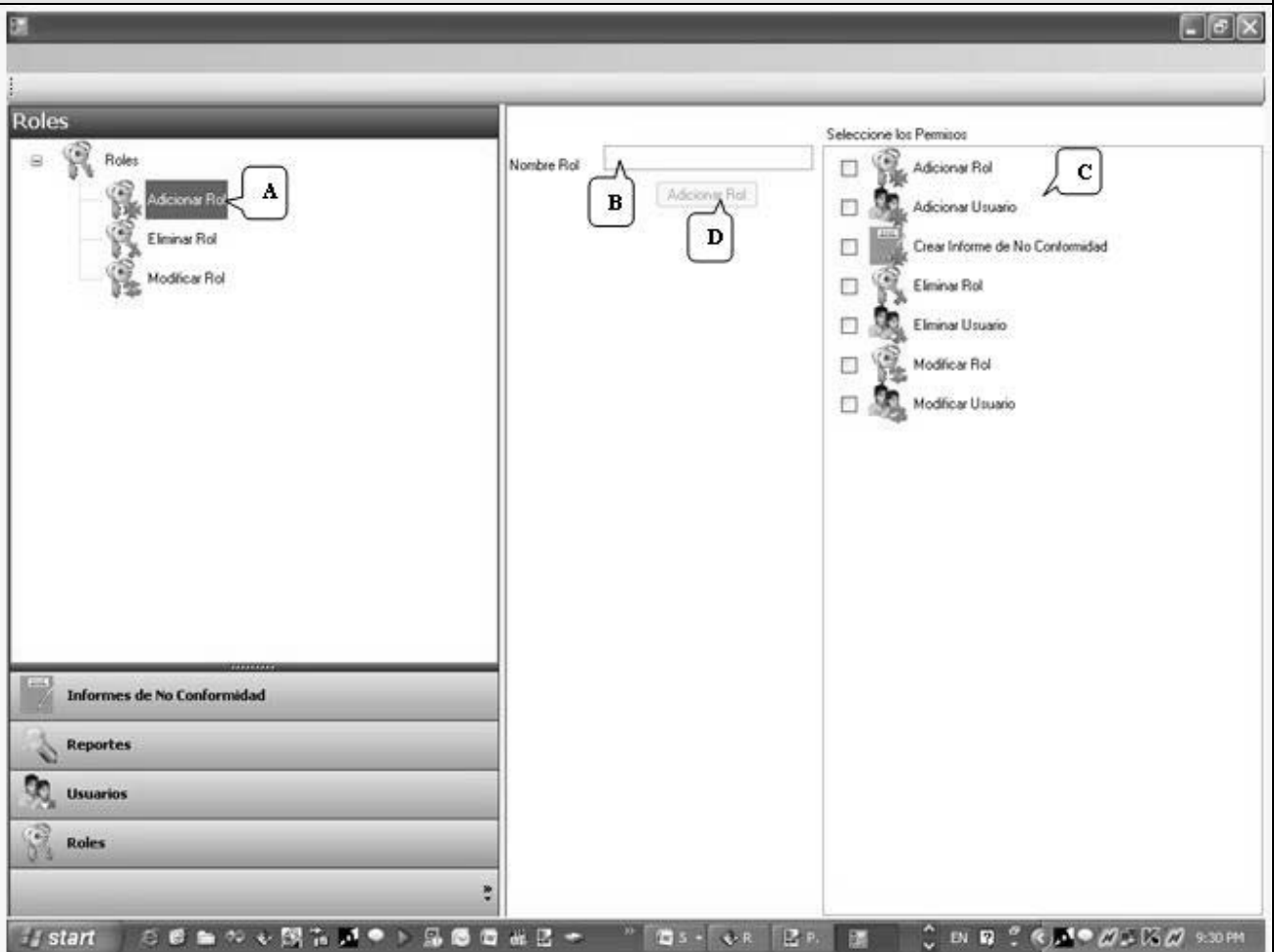
- El usuario tiene que estar autenticado.
- El usuario debe pertenecer a algún Rol que tenga dentro de sus Permisos “Adicionar Rol” y/o “Eliminar Rol” y/o “Modificar Rol”.

Interfaz I “Principal”



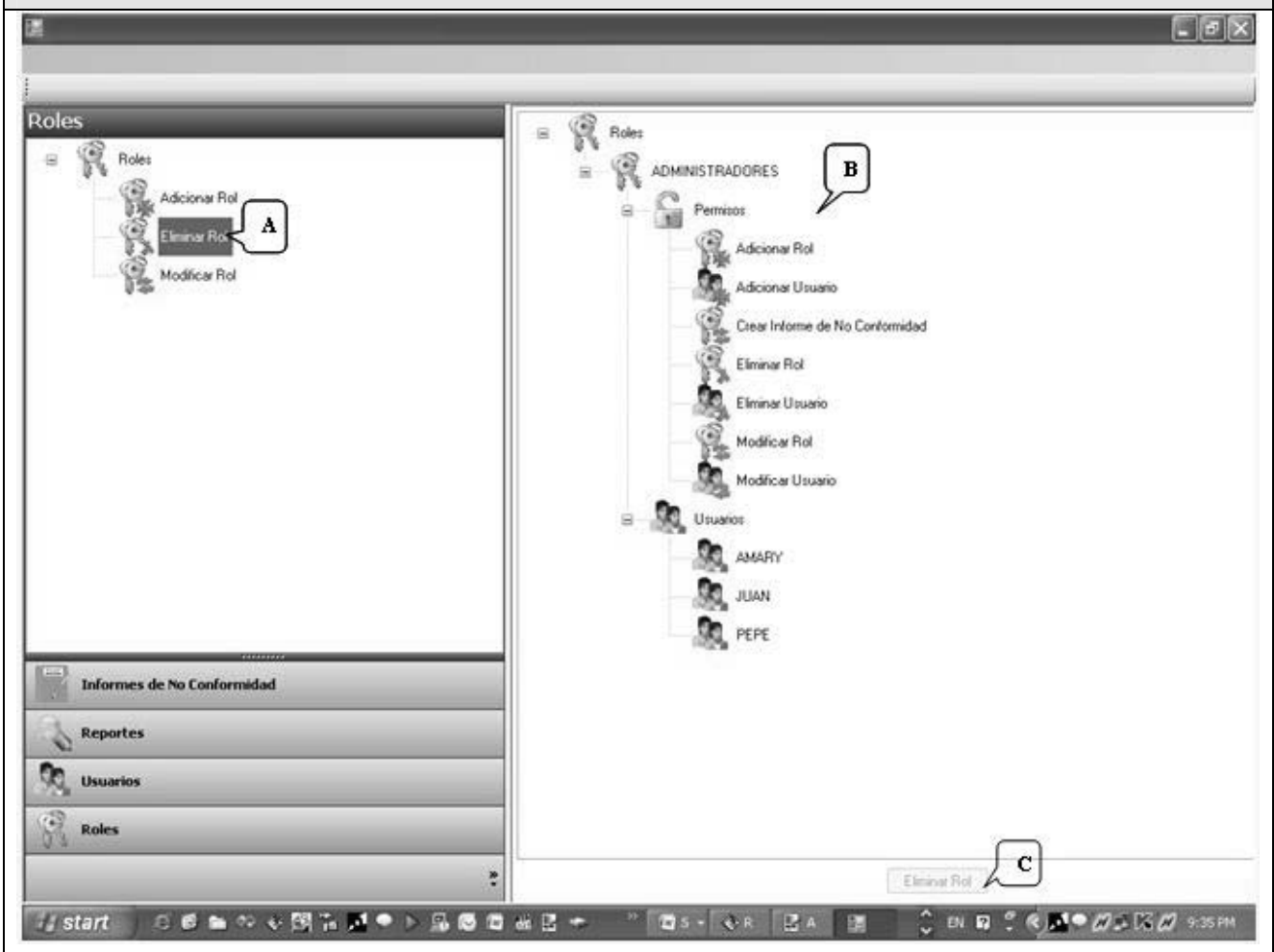
- ✚ A: Opción que permite gestionar los Roles.
- ✚ B: Muestra las Opciones que podrá realizar el Usuario con los Roles.
- ✚ C: Muestra Interfaz “Adicionar Rol”.
- ✚ D: Muestra Interfaz “Eliminar Rol”.
- ✚ E: Muestra Interfaz “Modificar Rol”.

Interfaz II “Adicionar Rol”



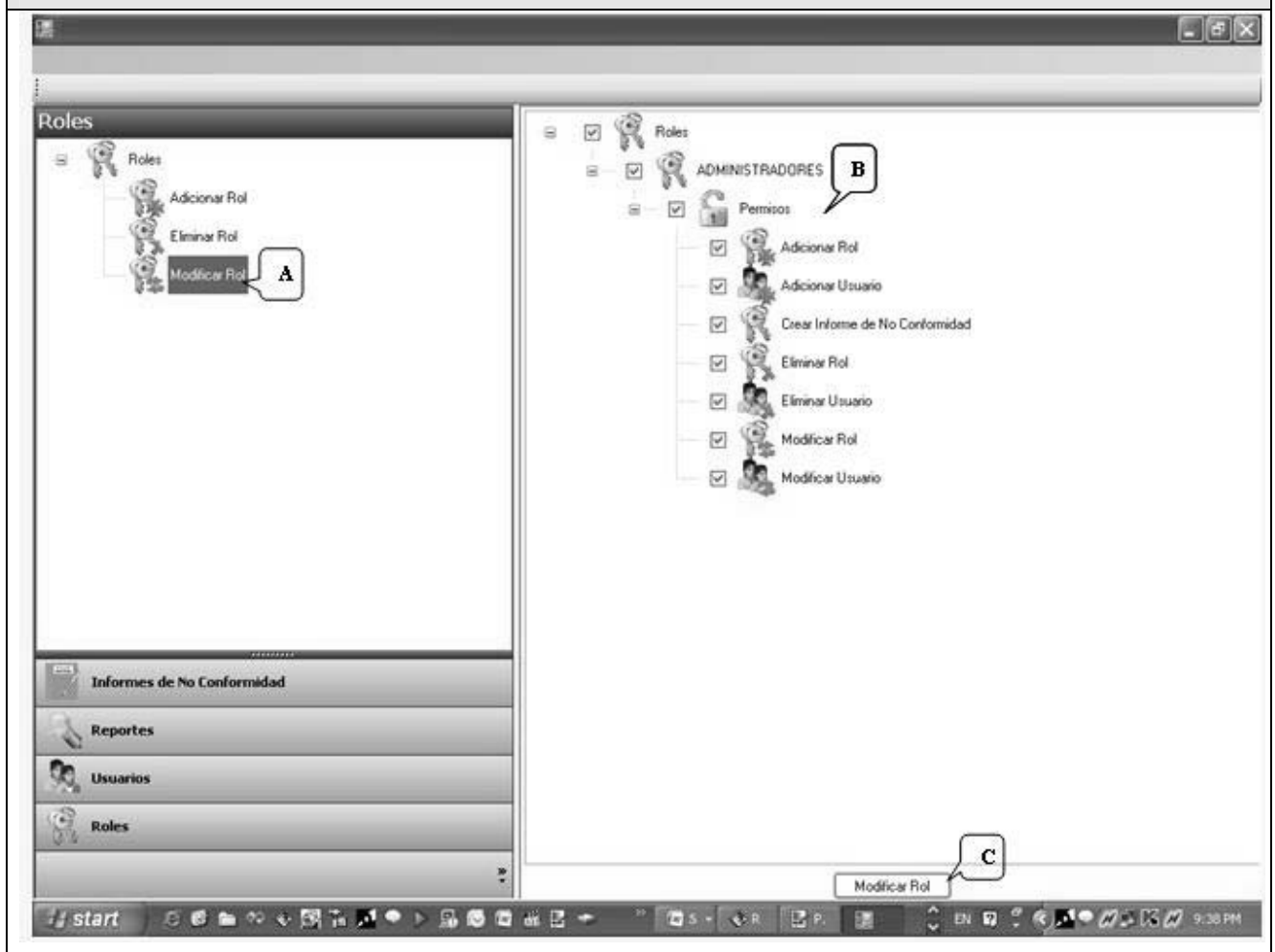
- ✚ A: Opción que posibilita ver la Interfaz para Adicionar un Rol al Sistema.
- ✚ B: Campo para escribir el Nombre del Nuevo Rol que será adicionado.
- ✚ C: Permisos que pueden ser seleccionados para el Rol que será adicionado.
- ✚ D: Botón que posibilita adicionar el Rol con sus permisos al Sistema.

Interfaz III “Eliminar Rol”



- ✚ A: Opción que posibilita ver la Interfaz para Eliminar un Rol del Sistema.
- ✚ B: Todos los Roles que se han definido en el Sistema con sus Permisos y Usuarios Asociados, para que seleccione el Rol a Eliminar.
- ✚ C: Botón que posibilita Eliminar un Rol con sus Permisos del Sistema.

Interfaz IV “Modificar Rol”



- ✚ A: Opción que posibilita ver la Interfaz para Modificar un Rol del Sistema.
- ✚ B: Todos los Roles que se han definido en el Sistema con sus Permisos para que seleccione el Rol a Modificar.
- ✚ C: Botón que posibilita Modificar un Rol en el Sistema.

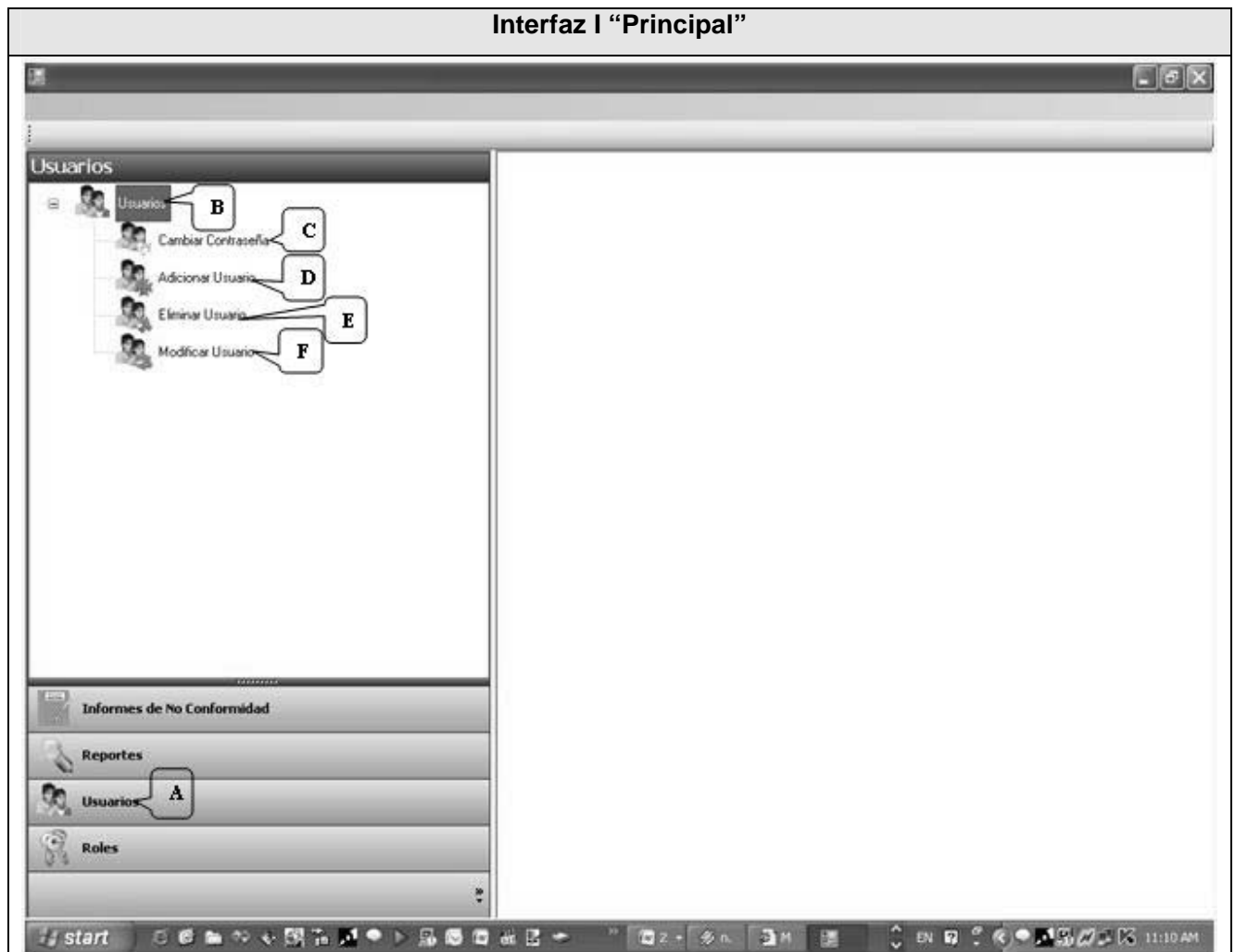
Acción del Actor	Respuesta del Sistema
1. Selecciona la Opción “Roles”. Interfaz I (A).	2. Muestra la Interfaz para Gestionar los Roles. Interfaz I (B).
3. Selecciona una de las Opciones.	4. Muestra la Interfaz correspondiente <ul style="list-style-type: none"> • En caso de que seleccione “Adicionar Rol” Interfaz I (C), Ver Sección “Adicionar Rol”. • En caso de que seleccione “Eliminar Rol” Interfaz I (D), Ver Sección “Eliminar Rol”. • En caso de que seleccione “Modificar Rol” Interfaz I (E), Ver Sección “Modificar Rol”.
Sección “Adicionar Rol”	
Acción del Actor	Respuesta del Sistema
	5. Muestra la Interfaz para Adicionar un Rol. Interfaz II
6. Escribe el nombre del nuevo Rol que será	9. Verifica que la información suministrada

<p>adicionado. Interfaz II (B).</p> <p>7. Selecciona el o los Permisos del Rol que será Adicionado. Interfaz II (C).</p> <p>8. Presiona el Botón “Adicionar Rol”. Interfaz II (D).</p>	<p>es correcta.</p> <ul style="list-style-type: none"> •En caso de haber Errores Ver Sección “Datos Incorrectos”. <p>10. Adiciona el nuevo Rol con sus Permisos al Sistema.</p> <p>11. Notifica el Resultado de la Operación al Usuario.</p>
Sección “Eliminar Rol”	
Acción del Actor	Respuesta del Sistema
	<p>5. Muestra la Interfaz para Eliminar un Rol. Interfaz III.</p>
<p>6. Selecciona el Rol que desea Eliminar. Interfaz III (B).</p> <p>7. Presiona el Botón “Eliminar Rol.” Interfaz III (C).</p>	<p>8. Verifica que el Rol se pueda Eliminar.</p> <ul style="list-style-type: none"> • En caso de que el Rol No se pueda Eliminar Ver Sección “Datos Incorrectos”. <p>9. Elimina el Rol con sus Permisos del Sistema.</p> <p>10. Notifica el Resultado de la Operación al Usuario.</p>
Sección “Modificar Rol”	
Acción del Actor	Respuesta del Sistema
	<p>5. Muestra la Interfaz para Modificar Roles. Interfaz IV.</p>

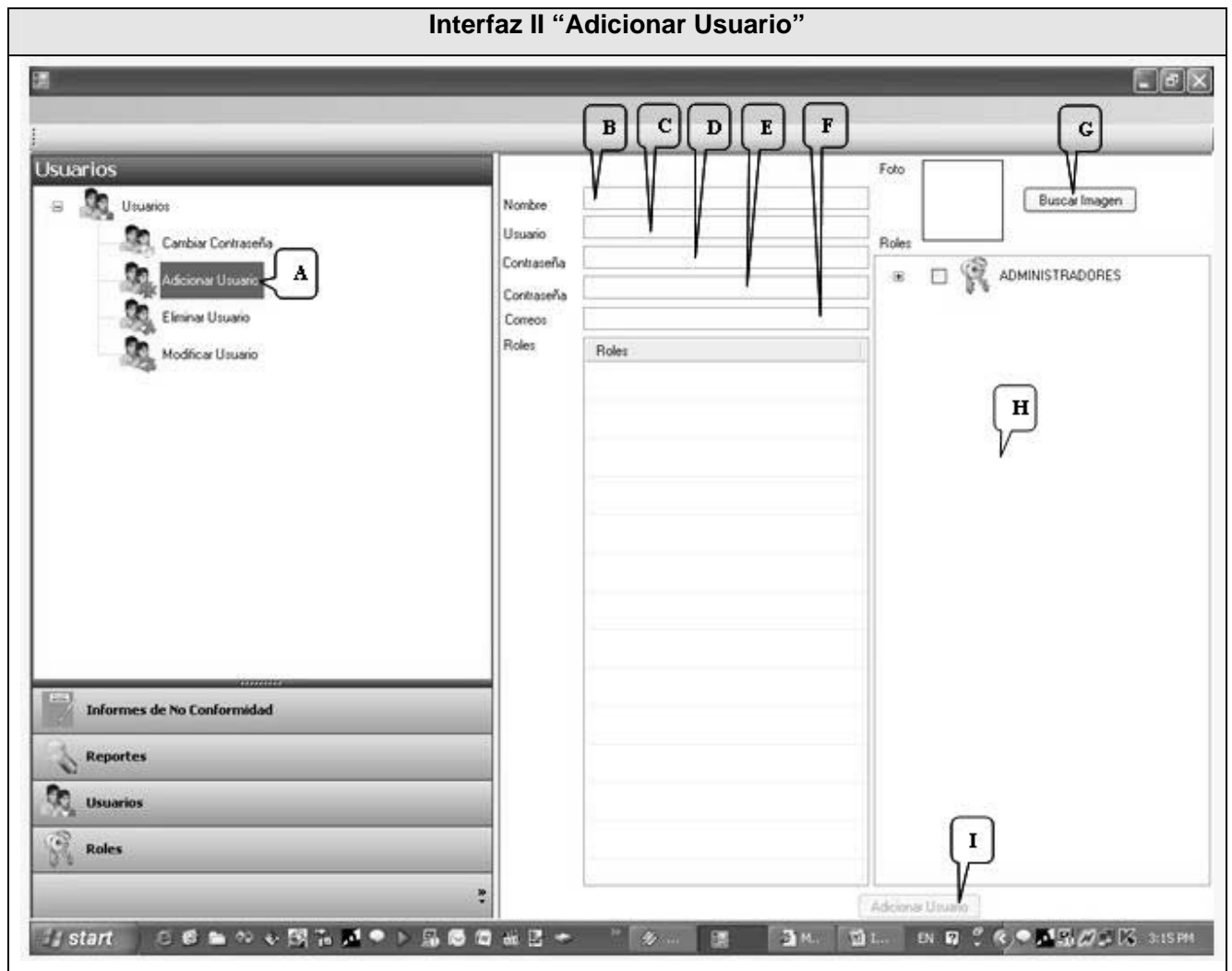
6. Modifica los Permisos del Rol o los Roles que desee. Interfaz IV (B).	8. Actualiza la información en el Sistema.
7. Presiona el Botón “Eliminar Rol.” Interfaz IV (C).	9. Notifica al Usuario el Resultado de la Operación.
Flujo Alternativo	
Sección “Datos Incorrectos”	
Acción del actor	Respuesta del sistema
	Notifica el Error Ocurrido.
Puntos de Extensión	
Prioridad	Alta

Tabla 16. Descripción Textual del caso de uso “Gestionar Usuarios”.

Caso de uso	
CU-7	Gestionar Usuarios
Propósito	Adicionar, Eliminar y Modificar Usuarios en el Sistema.
Actores	Usuario
Resumen	El usuario podrá Adicionar, Modificar o Eliminar Usuarios en el sistema. No se pueden Adicionar Usuarios con el mismo usuario de sistema. Para Eliminar un Usuario este no podrá tener Informes de No Conformidad creados, ni No Conformidades Asignadas. Para Modificar la Información de un Usuario este no puede estar conectado al Sistema.
Referencias	R 7, R 7.1, R 7.2, R 7.3
Precondiciones	<ul style="list-style-type: none"> • El usuario tiene que estar autenticado. • El usuario debe pertenecer a algún Rol que tenga dentro de sus Permisos “Adicionar Usuario” y/o “Eliminar Usuario” y/o “Modificar Usuario”.

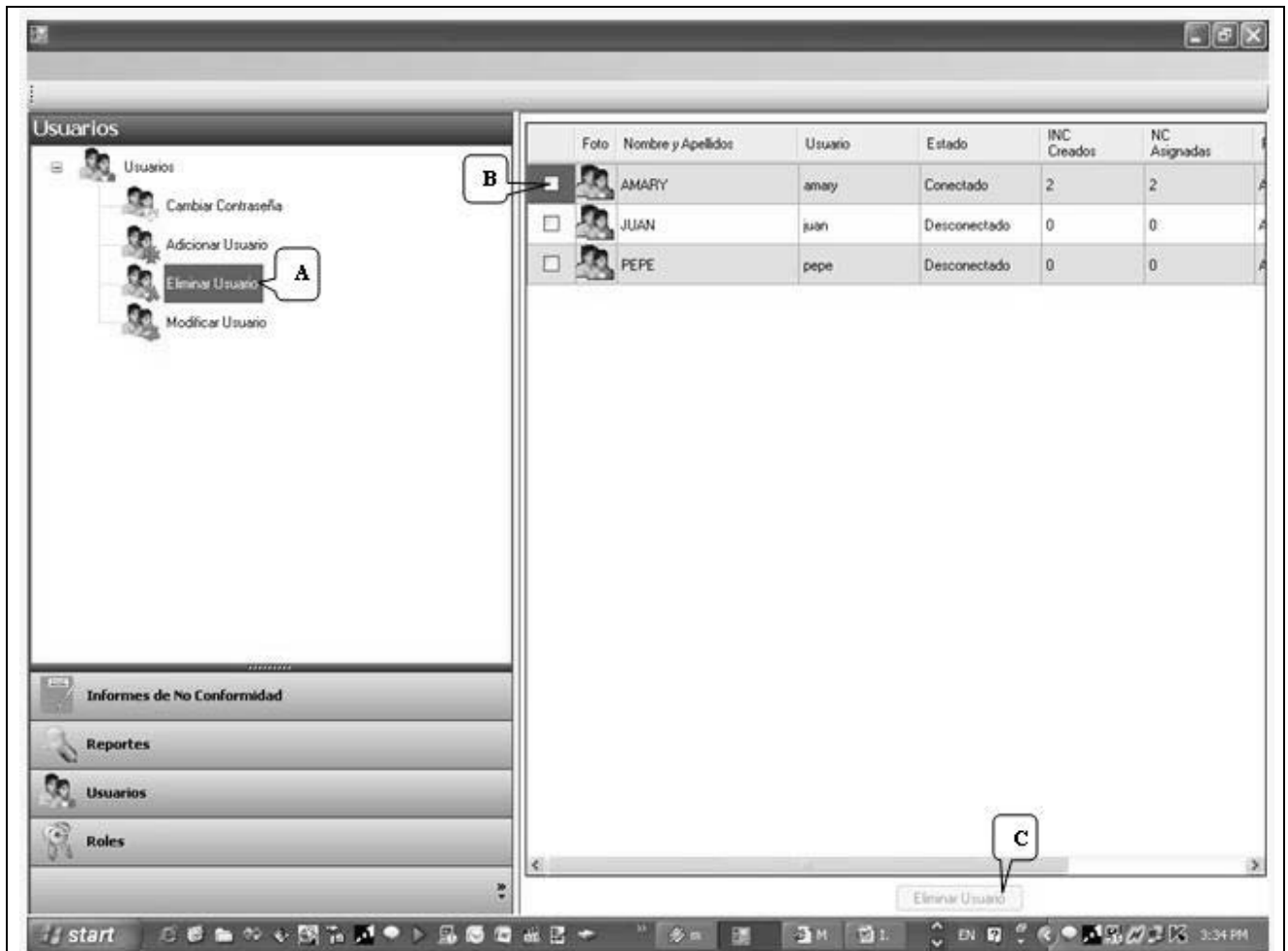


- ✚ A: Opción que permite gestionar los Usuarios.
- ✚ B: Muestra las Opciones que podrá realizar el Usuario con los Usuarios que existen en el Sistema.
- ✚ C: Muestra Interfaz "Adicionar Usuario".
- ✚ D: Muestra Interfaz "Eliminar Usuario".
- ✚ E: Muestra Interfaz "Modificar Usuario".



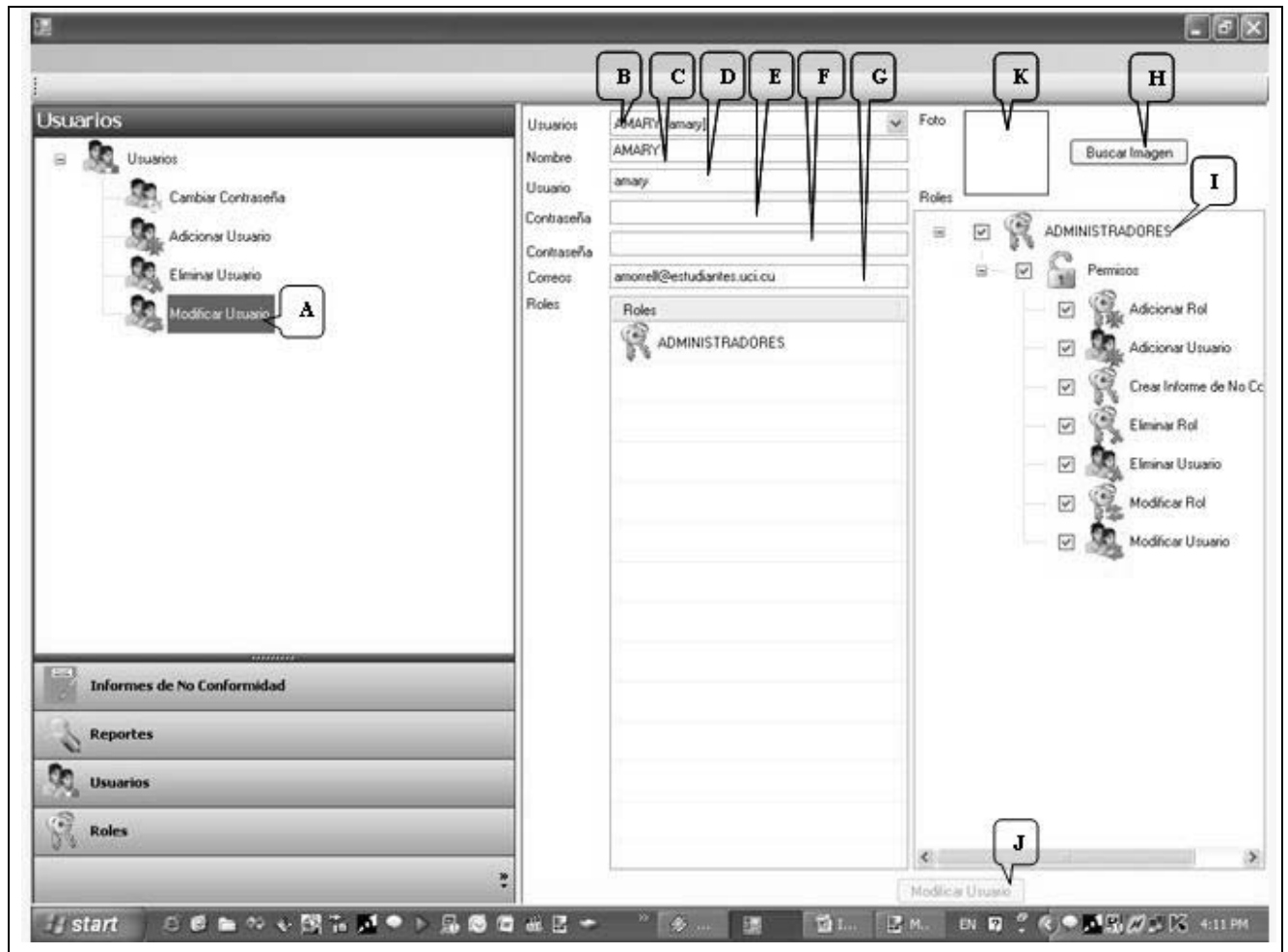
- ✚ A: Carga la Interfaz “Adicionar Usuario”.
- ✚ B: Campo para escribir el Nombre y Apellidos del nuevo Usuario que será adicionado.
- ✚ C: Campo para escribir el usuario del sistema del nuevo Usuario que será adicionado.
- ✚ D: Campo para escribir el contraseña de acceso al sistema del nuevo Usuario que será adicionado.
- ✚ E: Campo para escribir la confirmación de la contraseña de acceso al sistema del nuevo Usuario que será adicionado.
- ✚ F: Campo para escribir los correos electrónicos del nuevo Usuario que será adicionado.
- ✚ G: Botón que permite cargar la imagen del nuevo Usuario que será adicionado.
- ✚ H: Área donde se podrán seleccionar los Roles a los que pertenecerá el nuevo Usuario que será adicionado.
- ✚ I: Adiciona la información del usuario al sistema.

Interfaz III “Eliminar Usuario”



- ✚ A: Muestra la Interfaz “Eliminar Usuario”.
- ✚ B: Muestra los datos de todos los usuarios para que seleccione los que desea eliminar.
- ✚ C: Botón para eliminar los datos del o los usuarios del sistema.

Interfaz IV “Modificar Usuario”



- ✚ A: Carga la Interfaz “Modificar Usuario”.
- ✚ B: Muestra los Usuarios que Existen en el Sistema.
- ✚ C: Campo para escribir el Nombre y Apellidos del nuevo Usuario que será adicionado.
- ✚ D: Campo para escribir el usuario del sistema del nuevo Usuario que será adicionado.
- ✚ E: Campo para escribir el contraseña de acceso al sistema del nuevo Usuario que será adicionado.
- ✚ F: Campo para escribir la confirmación de la contraseña de acceso al sistema del nuevo Usuario que será adicionado.
- ✚ G: Campo para escribir los correos electrónicos del nuevo Usuario que será adicionado.
- ✚ H: Botón que permite cargar la imagen del nuevo Usuario que será adicionado.
- ✚ I: Área donde se podrán seleccionar los Roles a los que pertenecerá el nuevo Usuario que será adicionado.
- ✚ J: Adiciona la información del usuario al sistema.
- ✚ K: Muestra la foto del usuario.

Acción del Actor	Respuesta del Sistema
1. Selecciona la Opción “Usuarios”. Interfaz I (A).	2. Muestra la Interfaz para Gestionar los Usuarios. Interfaz I (B).
3. Selecciona una de las Opciones.	4. Muestra la Interfaz correspondiente: <ul style="list-style-type: none"> • En caso de que seleccione “Adicionar Rol” Interfaz I (C), Ver Sección “Adicionar Usuario”.

	<ul style="list-style-type: none"> • En caso de que seleccione “Eliminar Rol” Interfaz I (D), Ver Sección “Eliminar Usuario”. • En caso de que seleccione “Modificar Rol” Interfaz I (E), Ver Sección “Modificar Usuario”.
Sección “Adicionar Usuario”	
	<p>5. Muestra la Interfaz “Adicionar Usuario”. Interfaz II.</p>
<p>6. Escribe el Nombre y los Apellidos del nuevo Usuario que será adicionado. Interfaz II (B).</p> <p>7. Escribe el usuario de acceso al sistema del nuevo Usuario que será adicionado. Interfaz II (C).</p> <p>8. Escribe la contraseña (como confirmación) de acceso al sistema del nuevo Usuario que será adicionado. Interfaz II (D).</p> <p>9. Escribe la contraseña (como confirmación) de acceso al sistema del nuevo Usuario que será adicionado. Interfaz II (E).</p> <p>10. Escribe las direcciones de correo electrónico del nuevo Usuario que será</p>	<p>11. Verifica que la información suministrada sea correcta.</p> <ul style="list-style-type: none"> • En caso de que el Usuario no se pueda Adicionar: Ver Sección “Datos Incorrectos”.

<p>adicionado. Interfaz II (F).</p>	
<p>12. [Opcional] Presiona el Botón “Buscar Imagen” para cargar la Foto del nuevo Usuario que será adicionado. Interfaz II (G).</p> <p>13. Selecciona los Roles a los que pertenecerá el nuevo Usuario que será adicionado.</p> <p>14. Presiona el Botón “Adicionar Usuario”.</p>	<p>15. Verifica que la información suministrada sea correcta.</p> <ul style="list-style-type: none"> • En caso de que el Usuario no se pueda Adicionar: Ver Sección “Datos Incorrectos”. <p>16. Adiciona los Datos del nuevo Usuario al Sistema.</p> <p>17. Notifica al Usuario el resultado de la operación.</p>
<p>Sección “Eliminar Usuario”</p>	
	<p>5. Muestra la Interfaz “Eliminar Usuario”. Interfaz III.</p>
<p>6. Selecciona los Usuarios que desea Eliminar. Interfaz III (B)</p>	<p>7. Verifica que el Usuario seleccionado se pueda Eliminar.</p> <ul style="list-style-type: none"> • En caso de que el Usuario no se pueda Eliminar: Ver Sección “Datos Incorrectos”.
<p>8. Presiona el Botón “Eliminar Usuario”. Interfaz III (C)</p>	<p>9. Elimina los datos de los Usuarios seleccionados del Sistema.</p> <p>10. Notifica el Resultado de la Operación.</p>

Sección “Modificar Usuario”	
	<p>5. Muestra la Interfaz “Modificar Usuario”. Interfaz IV.</p>
<p>6. Selecciona el Nombre del Usuario que desea Modificar. Interfaz IV (B).</p>	<p>7. Carga los datos del Usuario Seleccionado:</p> <ul style="list-style-type: none"> • Nombre y Apellidos. Interfaz IV (C). • Usuario de acceso al sistema. Interfaz IV (D). • Correos. Interfaz IV (G). • Foto. Interfaz IV (K). • Selecciona los Roles a los que pertenece el Usuario. Interfaz IV (I).
<p>8. Modifica la los datos que desee.</p> <p>9. Escribe la contraseña de acceso al sistema del Usuario. Interfaz IV (E).</p> <p>10. Escribe la contraseña de acceso al sistema (como confirmación) del Usuario. Interfaz IV (E).</p> <p>11. Presiona el Botón “Modificar Usuario”. Interfaz IV (J).</p>	<p>12. Verifica que se puedan modificar los datos del usuario seleccionado.</p> <ul style="list-style-type: none"> • En caso de que no se puedan Modificar los datos del Usuario: Ver Sección “Datos Incorrectos”. <p>13. Actualiza los datos el usuario en el sistema.</p> <p>14. Notifica el resultado de la Operación.</p>
Flujo Alternativo	
Sección “Datos Incorrectos”	



Acción del Actor	Respuesta del Sistema
	Notifica el Error Ocurrido.
Puntos de Extensión	
Prioridad	Media

2.9 Conclusiones.

En este capítulo se ha hecho la propuesta del sistema sobre la base del problema que se debe resolver, se han descrito las funcionalidades básicas del mismo, las herramientas de modelación y diseño, el gestor de base de datos, se ha realizado el modelo de objetos, se han definido los requisitos funcionales y no funcionales, los actores del sistema, los modelos de casos del sistema para cada uno de los módulos que conformarán el mismo, y por último se han descrito detalladamente cada uno de los casos de uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

El análisis y diseño del sistema son uno de los mecanismos más importantes dentro del proceso unificado del desarrollo de un software. Se consideran más cercanos al acceso de datos y a la arquitectura física de la aplicación.

El modelo de análisis ofrece una especificación más precisa de los requisitos, los estructura de modo que facilita su preparación, su comprensión, su modificación y en general su mantenimiento. Este es tomado como una primera aproximación al modelo del diseño. Por su parte, este se encargará de moldear el sistema y buscar una forma de arquitectura que de vida a los requerimientos del sistema, que incluya componentes de código que se compilan e integran en versiones ejecutables del mismo.

Durante el análisis se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos, buscando la forma de conseguir una comprensión más precisa y una descripción que sea fácil de mantener, que ayude a estructurar el sistema entero incluyendo la arquitectura.

En el modelo de clases del análisis se definen varios artefactos, tal es el caso de las clases de interfaz, entidad y control.

Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del modelo del diseño.

Por su parte el modelo de diseño como ya se plantea anteriormente intenta preservar la estructura definida por el modelo de análisis. Es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto a otras restricciones del entorno de implementación tienen su impacto en el sistema que se desarrolla. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación.

3.2 Diagrama de Clases del Análisis.

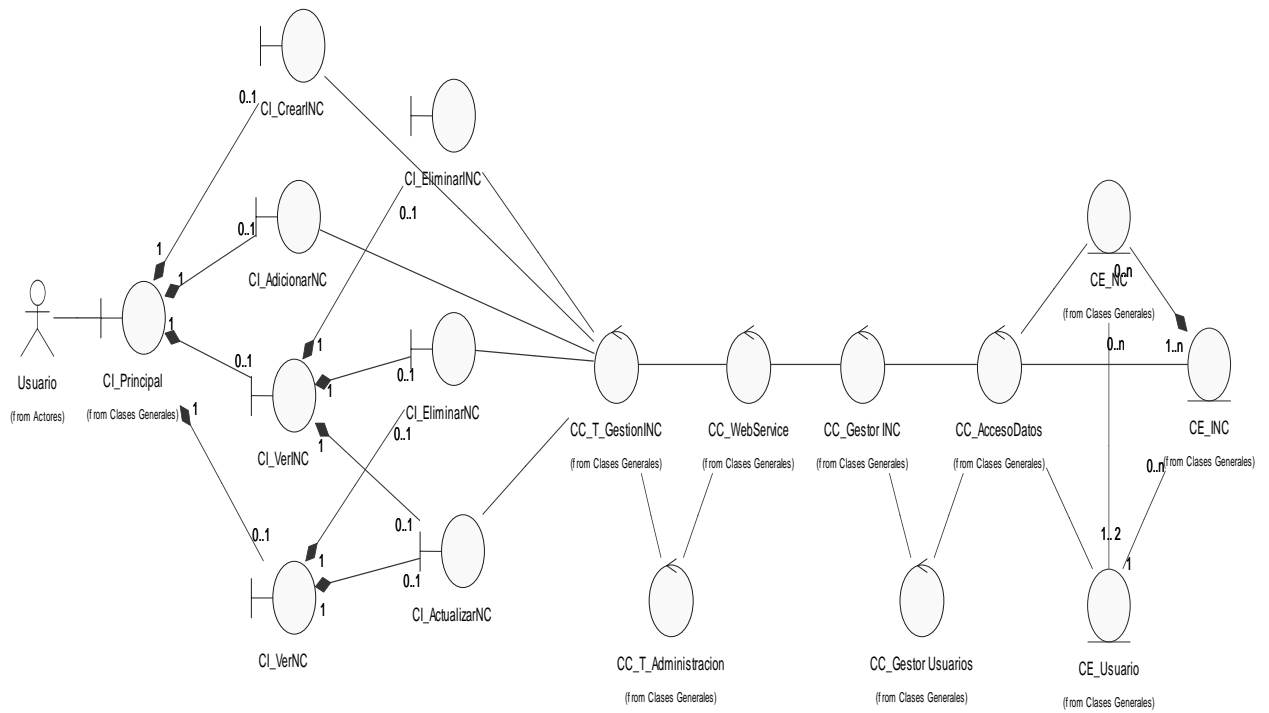


Figura 6. Diagrama de clases del análisis “Paquete de Informe de No Conformidades”.

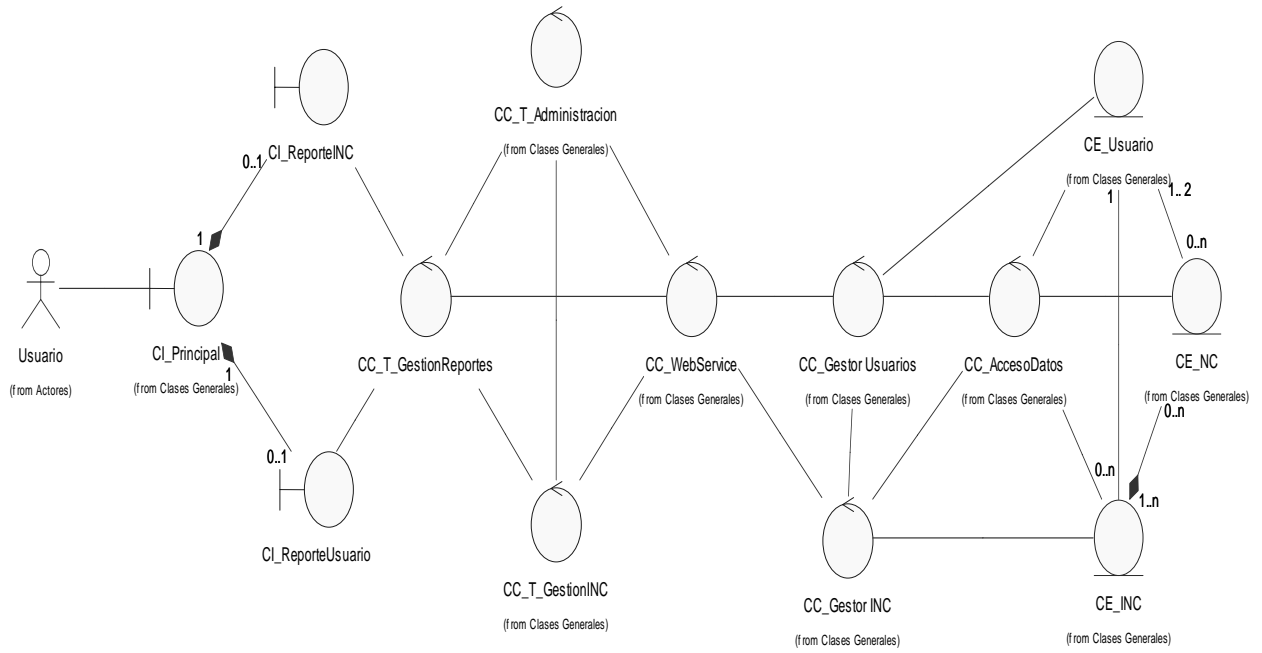


Figura 7. Diagrama de clases del análisis “Paquete de Reportes”

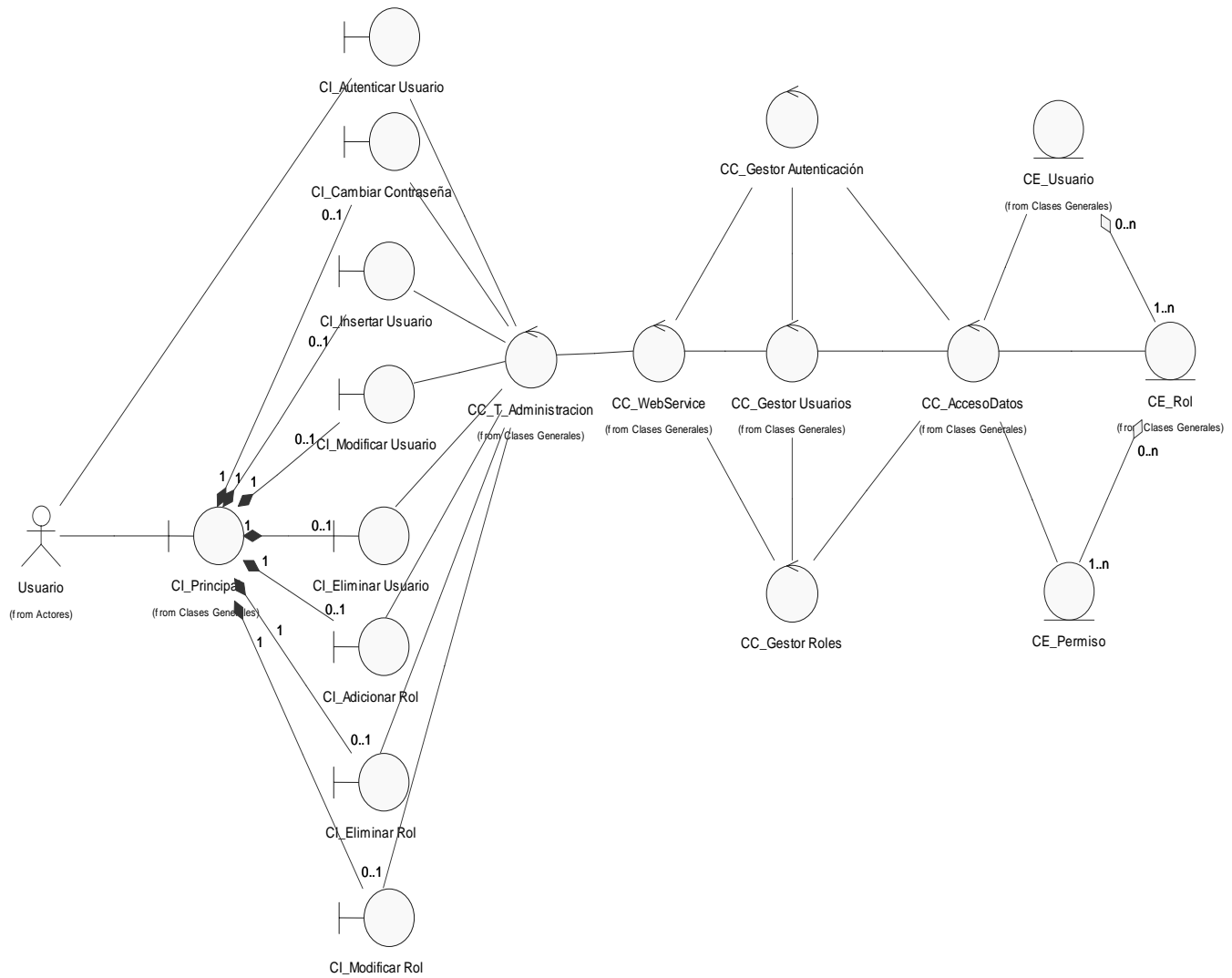


Figura 8. Diagrama de clases del análisis “Paquete de Administración.”

3.3 Diagramas de Interacción (Colaboración).

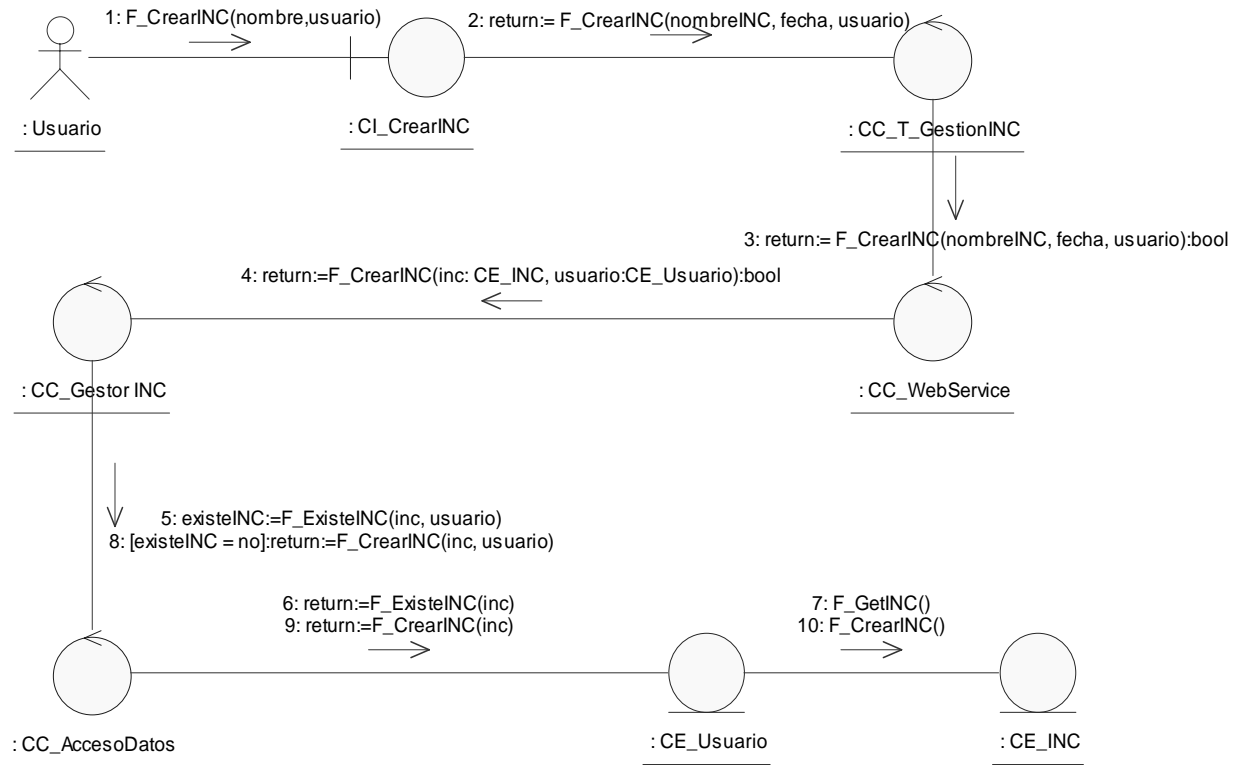


Figura 9. Diagrama de Colaboración “Crear Informe de No Conformidad” (Caso de Uso Gestionar INC).

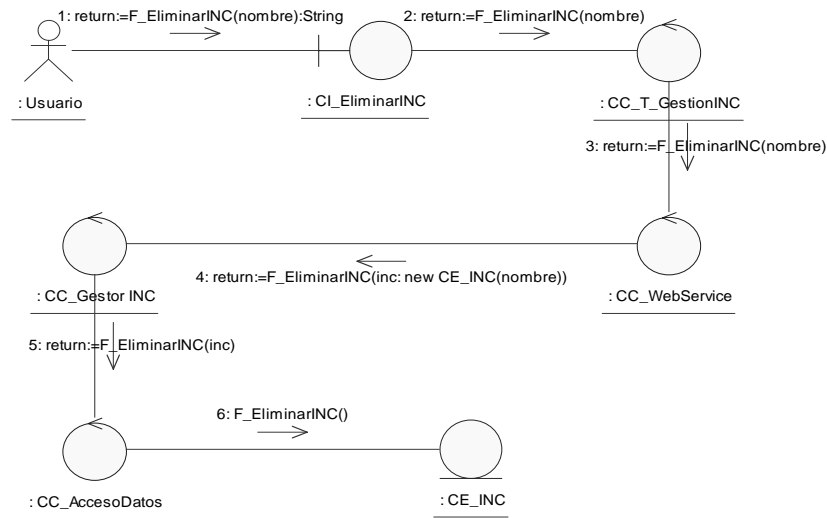


Figura 10. Diagrama de Colaboración “Eliminar Informe de No Conformidad” (Caso de Uso Gestionar INC).

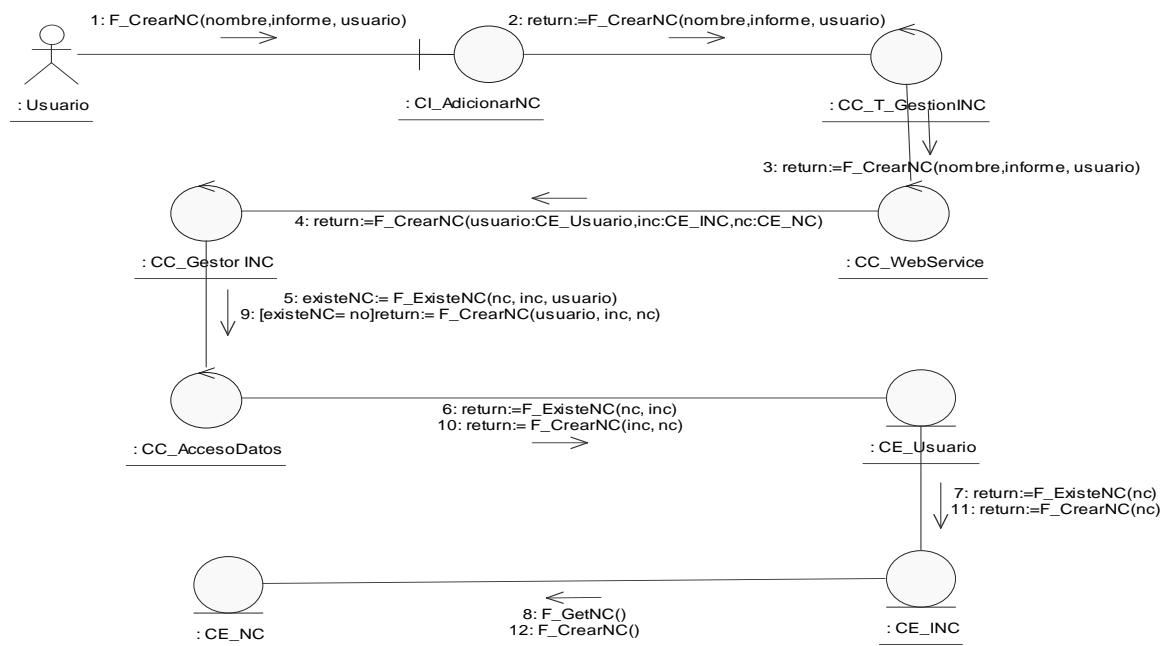


Figura 11. Diagrama de Colaboración “Crear No Conformidad” (Caso de Uso Gestionar NC).

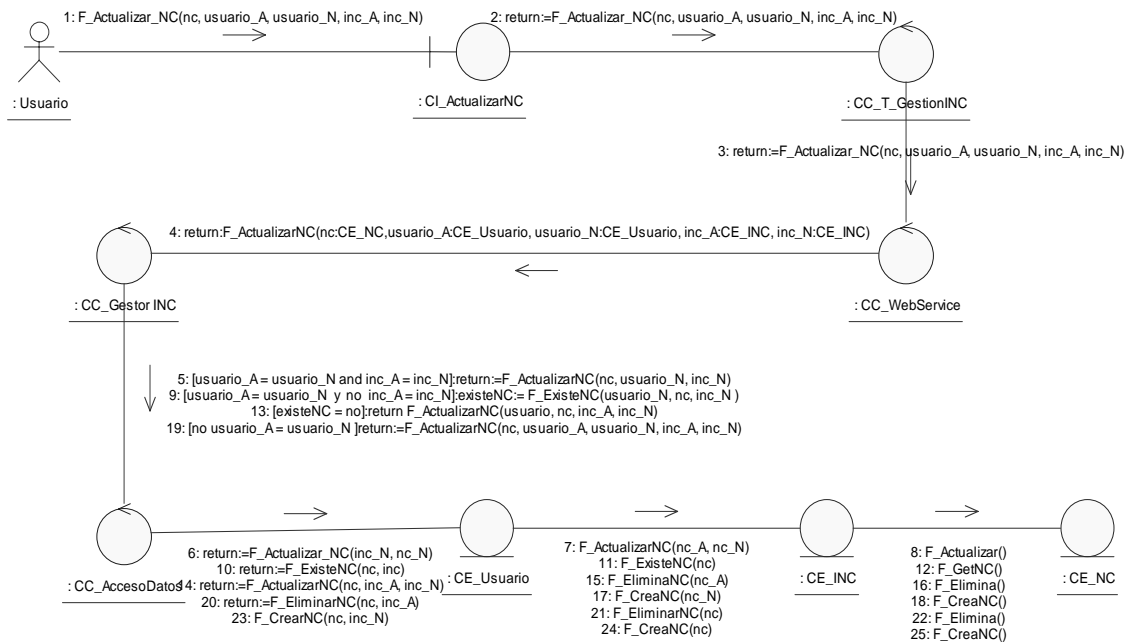


Figura 12. Diagrama de Colaboración “Modificar No Conformidad” (Caso de Uso Gestionar NC).

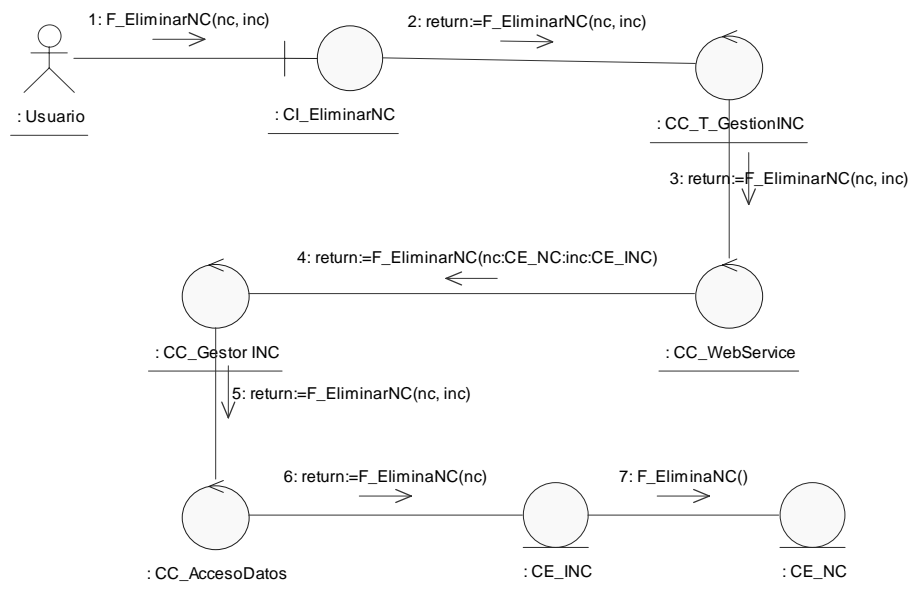


Figura 13. Diagrama de Colaboración “Eliminar No Conformidad” (Caso de Uso Gestionar NC).

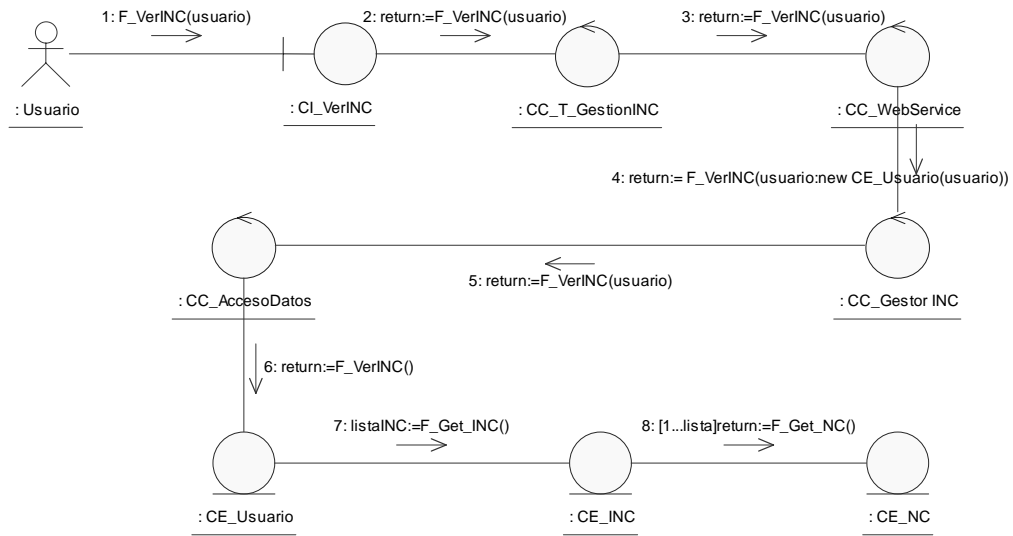


Figura 14. Diagrama de Colaboración “Ver Informes de No Conformidad” (Caso de Uso Gestionar INC).

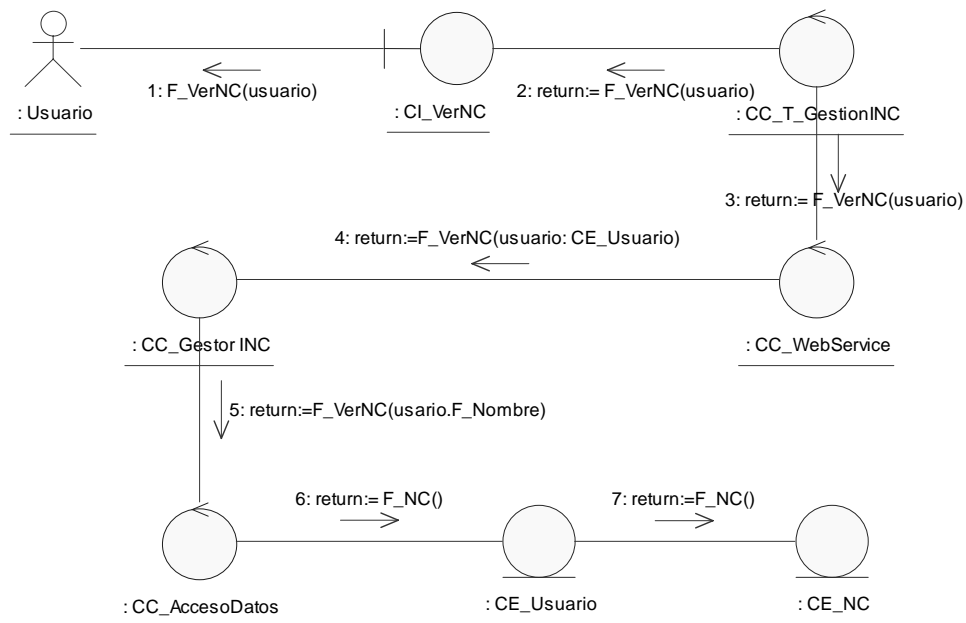


Figura 15. Diagrama de Colaboración “Ver No Conformidad” (Caso de Uso Gestionar NC).

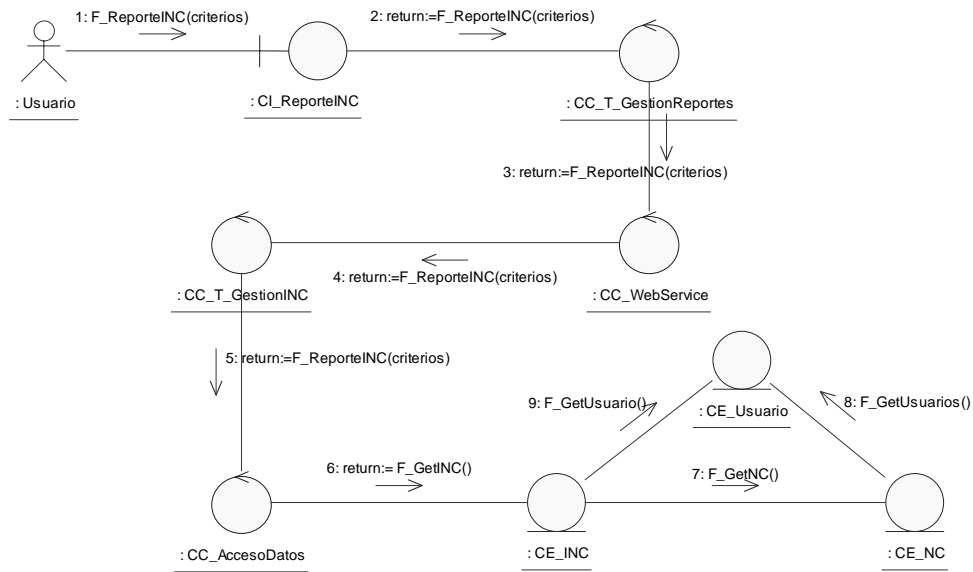


Figura 16. Diagrama de Colaboración “Obtener Reporte de Informes de No Conformidad” (Caso de Uso Gestionar Reportes).

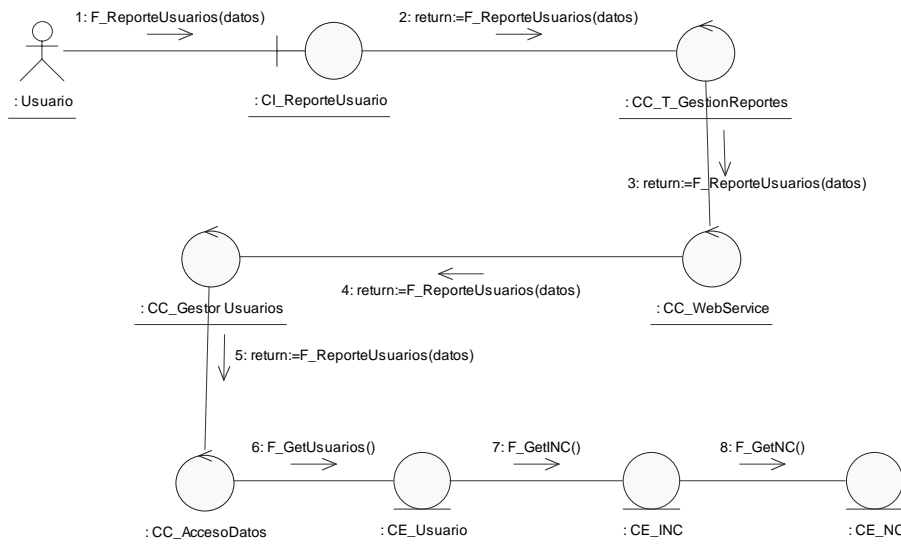


Figura 17. Diagrama de Colaboración “Obtener Reporte de Usuarios” (Caso de Uso Gestionar Reportes).

3.4 Diagrama de Clases del Diseño.

3.4.1 Paquete de Informes de No Conformidad (Clases Persistentes).

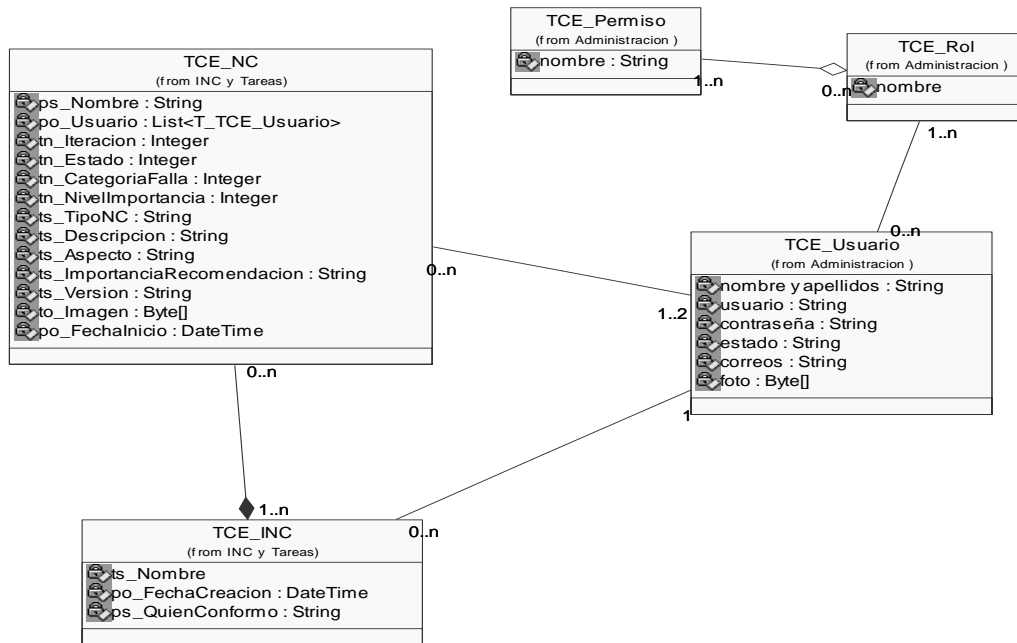


Figura 18. Diagrama de Clases Persistentes Paquete de Informes de No Conformidad.

3.4.1.1 Descripción de las Clases del Diseño.

Tabla 17. Descripción de la Clase “T_CE_Inf_No_Conformidad”.

Nombre: T_CE_Inf_No_Conformidad	
Tipo de clase: Entidad	
Atributo	Tipo
ps_Nombre	string
po_FechaCreacion	DateTime
po_QuienConformo	T_CE_Usuario
po_ListaNoConformidades	List<T_CE_No_Conformidad>
Responsabilidad:	
Nombre:	T_CE_Inf_No_Conformidad(string ts_Nombre, T_CE_Usuario

	to_UsuarioConformo):void
Descripción:	Constructor.
Nombre:	T_CE_Inf_No_Conformidad(string ts_Nombre, DateTime to_FechaCreacion, T_CE_Usuario to_UsuarioConformo):void
Descripción:	Constructor.
Nombre:	T_CE_Inf_No_Conformidad(string ts_Nombre):void
Descripción:	Constructor.
Nombre:	F_Atributo:Tipo de dato de retorno
Descripción:	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.
Nombre:	F_AdicionarNoConformidad(T_CE_No_Conformidad lo_NoConformidad):bool
Descripción:	Añade una No Conformidad al Informe

Tabla 18. Descripción de la Clase “T_CE_No_Conformidad”.

Nombre: T_CE_No_Conformidad	
Tipo de clase: Entidad	
Atributo	Tipo
Is_NombreNC	string
Io_Usuario	List<T_CE_Usuario>
po_Fechnicio	DateTime
pn_Iteracion	int
pn_Estado	int
pn_Categoria	int
pn_NivelImportancia	int
ps_TipoNC	string
ps_Aspecto	string
ps_ImportanciaRecomendacion	string
ps_Version	string
ps_Descripcion	string

po_Image	byte[]
Responsabilidad:	
Nombre:	T_CE_No_Conformidad(string ts_NombreNC,List<T_CE_Usuario> to_Usuario, int tn_iteracion, int tn_Estatus, int tn_CategoriaFalla, int tn_NivelImportancia, string ts_TipoNC, string ts_Descripcion,string ts_Aspecto, string ts_ImportanciaRecomendacion, string ts_Version, byte[] to_Imagen):void
Descripción:	Constructor
Nombre:	F_..... :Tipo de dato de retorno
Descripción:	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.

Tabla 19. Descripción de la Clase “T_CE_No_Conformidad”.

Nombre: T_CC_GestorInformeNoConformidad	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	F_CrearInformeNoConformidad(T_CE_Inf_No_Conformidad to_InformeNoConformidad):bool
Descripción:	Crea un INC.
Nombre:	F_EliminarINC(T_CE_Inf_No_Conformidad to_INC):bool
Descripción:	Elimina el INC, si no tiene NCs.
Nombre:	F_AdicionarNoConformidad(T_CE_Inf_No_Conformidad to_InformeNC, T_CE_No_Conformidad to_NoConformidad):bool
Descripción:	Adiciona una nueva NC al INC, siempre y cuando esta no exista en el informe.
Nombre:	F_EliminarNC(T_CE_No_Conformidad to_NC, T_CE_Inf_No_Conformidad to_INC):bool
Descripción:	Elimina una NC de un INC.
Nombre:	F_ActualizarNC(T_CE_Inf_No_Conformidad to_INC_Nuevo, T_CE_No_Conformidad to_NC_Nueva, T_CE_Inf_No_Conformidad to_INC_Viejo, T_CE_No_Conformidad to_NC_Vieja):bool

Descripción:	Actualiza los datos de una NC.
Nombre:	F_Cantidad_INC_Creados(T_CE_Usuario to_Usuario):int
Descripción:	Devuelve la cantidad de INCs que han sido creados por el Usuario.
Nombre:	F_Cant_NC_Creadas(T_CE_Usuario to_Usuario):int
Descripción:	Devuelve la cantidad de NCs creadas por el Usuario.
Nombre:	F_Cant_NC_Resolver(T_CE_Usuario to_Usuario):int
Descripción:	Devuelve la cantidad de NCs que deben ser resueltas por el Usuario.
Nombre:	F_Cant_NC_Pertenecientes(T_CE_Inf_No_Conformidad to_INC):int
Descripción:	Devuelve la cantidad de NCs pertenecientes a un INC.
Nombre:	F_INC(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve los nombres de los INCs del Usuario con la cantidad de NCs pertenecientes a cada INC.
Nombre:	F_NC_INC(T_CE_Inf_No_Conformidad to_INC): XmlDocument
Descripción:	Devuelve las NCs pertenecientes a un INC.
Nombre:	F_ObtenerResponsableNC(XmlDocument to_Doc_NC): XmlDocument
Descripción:	Coloca los responsables de resolver cada NC.
Nombre:	F_Datos_NC(T_CE_No_Conformidad to_NC, T_CE_Inf_No_Conformidad to_INC): XmlDocument
Descripción:	Devuelve los datos de una NC perteneciente a un INC.
Nombre:	F_NombresINC():XmlDocument
Descripción:	Devuelve los nombres de todos los INCs.
Nombre:	F_NombresNC():XmlDocument
Descripción:	Devuelve todos los nombres de las NCs.
Nombre:	F_INC_CreadosPor():XmlDocument
Descripción:	Devuelve el usuario y el nombre y apellidos de los Usuarios que han creado INC.
Nombre:	F_Resueltos_Por():XmlDocument
Descripción:	Devuelve el usuario y el nombre y apellidos de los Usuarios que tienen NC asignadas para resolver.
Nombre:	F_Version():XmlDocument
Descripción:	Devuelve todas las versiones diferentes que se han insertado, para utilizarlas

	como parámetro en la consulta.
Nombre:	F_NC_Usuario(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve las NC que deben ser resueltas por el Usuario.
Nombre:	F_INC_NC(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve los INC creados por el Usuario con las NC de cada uno.
Nombre:	F_ConsultaNCsINC():XmlDocument
Descripción:	Devuelve todos los datos de las NCs de los INCs.

3.4.2 Paquete de Administración (Clases Persistentes).

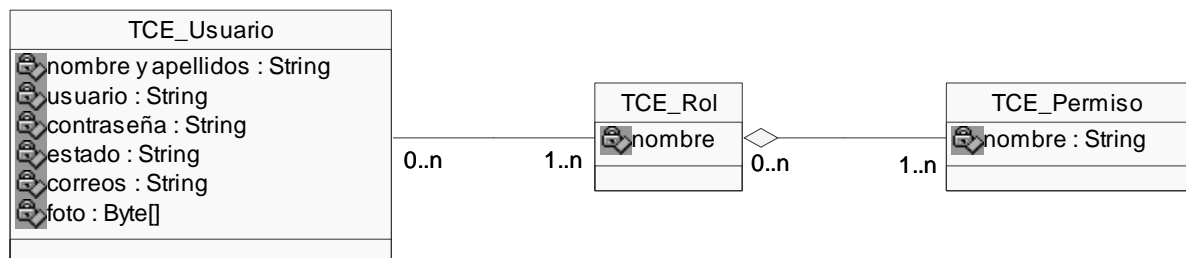


Figura 18. Diagrama de Clases Persistentes Paquete de Administración.

3.4.2.1 Descripción de las Clases del Diseño.

Tabla 20. Descripción de la Clase “T_CE_Permission”.

Nombre: T_CE_Permission	
Tipo de clase: Entidad	
Atributo	Tipo
ps_Nombre	string
Responsabilidad:	
Nombre:	T_CE_Permission (string ts_Nombre):void
Descripción:	Constructor.
Nombre:	F_..... :Tipo de dato de retorno
Descripción:	Las propiedades de actualización y retorno de cada atributo. El nombre se

	corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.
--	---

Tabla 21. Descripción de la Clase “T_CE_Rol”.

Nombre: T_CE_Rol	
Tipo de clase: Entidad	
Atributo	Tipo
ps_Nombre	string
po_ListaPermisos	List<T_CE_Permiso>
Responsabilidad:	
Nombre:	T_CE_Rol(string ts_Nombre):void
Descripción:	Constructor.
Nombre:	T_CE_Rol(string ts_Nombre,List<T_CE_Permiso> to_ListaPermisos):void
Nombre:	F_..... :Tipo de dato de retorno
Descripción:	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.

Tabla 22. Descripción de la Clase “T_CE_Usuario”.

Nombre: T_CE_Usuario	
Tipo de clase: Entidad	
Atributo	Tipo
ps_NombreApellidos	string
po_ListaRoles	List<T_CE_Rol>
ps_Usuario	string
ps_Contrasena	string
pc_Estado	char
pa_foto	byte[]
Responsabilidad:	
Nombre:	T_CE_Usuario(List<T_CE_Rol> to_ListaRoles, string ts_Usuario, string

	ts_Contrasena, string ts_NombreApellidos, byte[] ta_foto, string ts_Correos):void
Descripción:	Constructor.
Nombre:	T_CE_Usuario(string ts_Usuario):void
Descripción:	Constructor.
Nombre:	T_CE_Usuario(string ts_Usuario, string ts_Contrasena, char tc_Estado)
Descripción:	Constructor.
Nombre:	F_..... :Tipo de dato de retorno
Descripción:	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.

Tabla 23. Descripción de la Clase “T_CC_GestorRoles”.

Nombre: T_CC_GestorRoles	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	F_Perminos():XmlDocument
Descripción:	Devuelve Todos los Permisos que se han Definido
Nombre:	F_Roles():XmlDocument
Descripción:	Devuelve Todos los Roles que se han Definido
Nombre:	F_AdicionarRol(T_CE_Rol to_Rol):bool
Descripción:	Adiciona un Nuevo Rol con sus Permisos
Nombre:	F_EliminarRol(T_CE_Rol to_Rol):bool
Descripción:	Elimina un Rol, con sus Permisos, si este no tiene usuarios que pertenezcan al mismo
Nombre:	F_ModificarRol(T_CE_Rol to_Rol, List<T_CE_Permission> to_PerminosNuevos):bool
Descripción:	Reemplaza los Permisos del Rol, por los Nuevos Permisos que se han definido
Nombre:	F_RolExiste(T_CE_Rol to_Rol):bool
Descripción:	Indica si se ha definido o no el Rol

Nombre:	F_Perminos(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve los Permisos que se han definido para el usuario
Nombre:	F_Roles(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve los Roles a los que pertenece el Usuario

Tabla 24. Descripción de la Clase “T_CC_GestorUsuarios”.

Nombre: T_CC_GestorUsuarios	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	F_NombreUsuario(T_CE_Usuario to_Usuario):string
Descripción:	Devuelve el Nombre y Apellidos del usuario
Nombre:	F_Correo(T_CE_Usuario to_Usuario):string
Descripción:	Devuelve las direcciones de correo electrónico del usuario
Nombre:	F_Todos_Nombres_y_Usuarios(): XmlDocument
Descripción:	Devuelve Todos los Nombres y usuarios de la aplicación, que se han definido.
Nombre:	F_DatosUsuario(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve toda la Información del Usuario (Nombre y Apellidos, Usuario, Foto, Roles, Estado, Cantidad de INC creados, Cantidad de NC asignadas)
Nombre:	F_DatosTodosUsuarios(): XmlDocument
Descripción:	Devuelve toda la Información de todos los Usuarios (Nombre y Apellidos, Usuario, Foto, Roles, Estado, Cantidad de INC creados, Cantidad de NC asignadas)
Nombre:	F_AñadirUsuario(T_CE_Usuario to_Usuario):bool
Descripción:	Añade un Nuevo Usuario en caso de que este no exista
Nombre:	F_EliminarUsuario(T_CE_Usuario to_Usuario):bool
Descripción:	Elimina toda la Información referente al Usuario si no está conectado y no tiene INC creados, ni NC asignadas.
Nombre:	F_ModificarUsuario(T_CE_Usuario to_Usuario, T_CE_Usuario to_UsuarioAntiguo):bool
Descripción:	Modifica la Información referente al Usuario, siendo to_UsuarioAntiguo, el usuario

	existente y to_Usuario el Usuario con las nuevas características, si no está conectado al sistema.
Nombre:	F_UsuarioExiste(T_CE_Usuario to_Usuario):bool
Descripción:	Indica si existe o no el usuario.
Nombre:	F_ValidarDatos(T_CE_Usuario to_Usuario):bool
Descripción:	Valida que los Datos de Entrada proporcionados (usuario y contraseña) sean Correctos. La contraseña debe de estar encriptada.
Nombre:	F_ActualizarEstadoConexion(T_CE_Usuario to_Usuario):bool
Descripción:	Actualiza el estado de Conexión del usuario(C= Conectado, D=Desconectado)
Nombre:	F_CambiarContrasena(T_CE_Usuario to_Usuario, string ts_ContrasenaNueva):bool
Descripción:	Cambia la contraseña actual del usuario por la contraseña nueva. La contraseña debe de estar encriptada.
Nombre:	F_PerminosUsuario(T_CE_Usuario to_Usuario): XmlDocument
Descripción:	Devuelve los permisos del usuario.
Nombre:	F_ConsultaNCsUsuario(): XmlDocument
Descripción:	Devuelve toda la información referente al Usuario (Nombre y Apellidos, Usuario, Foto, INCs con sus NCs (creadas y asignadas), abarcando toda la información de los INCs y de las NCs).

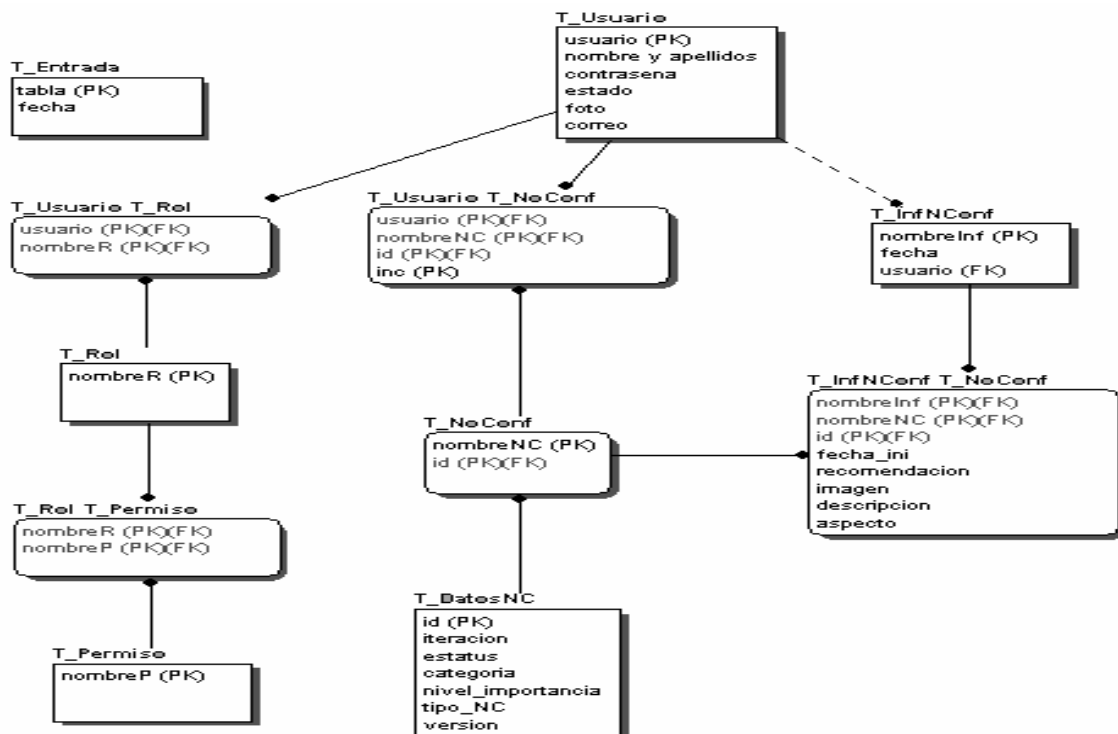
Tabla 25. Descripción de la Clase “T_CC_GestorAumentacion”.

Nombre: T_CC_GestorAumentacion	
Tipo de clase: Controladora	
Atributo	Tipo
Responsabilidad:	
Nombre:	F_ValidarDatos(T_CE_Usuario to_Usuario):bool
Descripción:	Valida que el usuario y contraseña de acceso al sistema sean correctos.
Nombre:	F_ActualizarEstado(T_CE_Usuario to_Usuario):bool
Descripción:	Actualiza el estado de conexión del usuario(C= Conectado, D= Desconectado)
Nombre:	F_CambiarContrasena(T_CE_Usuario to_Usuario, string

	ts_ContrasenaNueva):bool
Descripción:	Cambia la contraseña actual del usuario por la contraseña nueva
Nombre:	F_UltimaModificacion(string ts_tabla): DateTime
Descripción:	Devuelve la fecha y hora de la última modificacion realizada en una tabla.
Nombre:	F_ActualizarUltimaModificacion(string ts_tabla):bool
Descripción:	Actualiza la Fecha-Hora de acceso a una tabla.

El Diagrama Entidad-Relación fue realizado en Edwin, herramienta que se utiliza para modelar, que ayuda a diseñar bases de datos de alto desempeño para cliente/servidor y *Web/Intranet*, La herramienta Erwin no solo ayuda a diseñar modelos de datos lógicos, también construye automáticamente estructuras de datos físicos, incluyendo todas las tablas, índices, procedimientos almacenados, *triggers* de integridad referencial y otros componentes necesarios para manejar exitosamente los datos usados en la organización con solo la información del Diagrama Entidad Relación.

3.5 Diseño de la Base de Datos.



3.5.1 Descripción de las Tablas de la Base de Datos.

Tabla 26. Descripción de la tabla “T_Permission” de la base de datos.

Nombre: T_Permission		
Descripción: Almacena todas las posibles acciones que podrán realizarse en el sistema.		
Atributo	Tipo	Descripción
nombreP	nvarchar(50)	Almacena una acción que se puede realizar en el sistema.

Tabla 27. Descripción de la tabla “T_Rol” de la base de datos.

Nombre: T_Rol		
Descripción: Almacena el nombre de todos los roles que existen en el sistema.		
Atributo	Tipo	Descripción
nombreR	nvarchar(50)	Nombre del Rol.

Tabla 28. Descripción de la tabla “T_Rol T_Permission” de la base de datos.

Nombre: T_Rol T_Permission		
Descripción: Surge de la relación [muchos a muchos] entre las tablas “T_Rol” y “T_Permission”, ya que un Rol puede tener definidos varios Permisos y un Permiso pudiera estar incluido en varios Roles.		
Atributo	Tipo	Descripción
nombreR	nvarchar(50)	Campo llave de la tabla “T_Rol”.
nombreP	nvarchar(50)	Campo llave de la tabla “T_Permission”.

Tabla 29. Descripción de la tabla “T_Usuario” de la base de datos.

Nombre: T_Usuario		
Descripción: Almacena los datos de los Usuarios que harán uso del sistema.		
Atributo	Tipo	Descripción
usuario	nvarchar(20)	Usuario de acceso al sistema de un Usuario.
nombre y apellidos	nvarchar(60)	Nombre y Apellidos del Usuario.
contraseña	nvarchar(50)	Contraseña del Usuario. Se guarda encriptada.

estado	char(1)	Estado de Conexión del Usuario (C = "Conectado", D = "Desconectado").
foto	image	Imagen que identifica al Usuario
correo	char(500)	Almacena las direcciones de correo electrónico del usuario.

Tabla 30. Descripción de la tabla "T_Usuario T_Rol" de la base de datos.

Nombre: T_Usuario T_Rol		
Descripción: Surge de la relación [muchos a muchos] entre las tablas "T_Usuario" y "T_Rol", ya que un usuario puede pertenecer a varios Roles y un Rol pudiera tener varios Usuarios.		
Atributo	Tipo	Descripción
usuario	nvarchar(20)	Campo llave de la tabla "T_Usuario".
nombreR	nvarchar(50)	Campo llave de la tabla "T_Rol".

Tabla 31. Descripción de la tabla "T_InfNConf" de la base de datos.

Nombre: T_InfNConf		
Descripción: Almacena los datos de los Informes de No Conformidad.		
Atributo	Tipo	Descripción
nombreInf	char(50)	Nombre del Informe de No Conformidad.
fecha	datetime	Fecha de creación del Informe de No Conformidad.
usuario	nvarchar(50)	Campo que se agrega a la tabla producto de la relación [1 a muchos] con la tabla "T_Usuario", ya que un Informe de No Conformidad solo podrá ser creado por un Usuario, pero a su vez un Usuario podrá crear varios Informes de No Conformidad.

Tabla 32. Descripción de la tabla “T_NoConf” de la base de datos.

Nombre: T_NoConf		
Descripción: Contendrá la Información de las No Conformidades.		
Atributo	Tipo	Descripción
nombreNC	char(40)	Nombre de la No Conformidad.
id	int	Campo que se agrega a la tabla debido a la relación [1 a muchos] con la tabla “T_DatosNC”, debido a que a cada No Conformidad tendrá asociada una tupla de la tabla “T_DatosNC”, pero una misma tupla de la tabla “T_DatosNC” podrá ser compartida por varias No Conformidades.

Tabla 33. Descripción de la tabla “T_DatosNC” de la base de datos.

Nombre: T_DatosNC		
Descripción: Almacena el conjunto de datos de las No Conformidades que pudiesen ser comunes entre estas, para evitar redundancias.		
Atributo	Tipo	Descripción
id	int	Campo llave de la tabla.
iteracion	int	Representa un valor numérico de la iteración de la No Conformidad.
estatus	int	Representa un valor numérico que se corresponde con el estado de solución en el que se encuentra la No Conformidad. <ul style="list-style-type: none"> 0. Inconformidad no resuelta. 1. Inconformidad resuelta. 2. Inconformidad no procede. 3. Pendiente por parte del cliente.

		4. Reincidencia de la falla.
categoria	int	Representa un valor numérico que se corresponde con la categoría a la que pertenece la No Conformidad. 0. Capa de Presentación. 1. Capa de Negocios. 2. Capa de Negocio y Capa de Datos.
nivel_importancia	int	Representa un valor numérico que se corresponde con la importancia que se confiere a la No Conformidad. 0. Baja. 1. Media. 2. Alta.
tipo_NC	char(2)	Representa el tipo de No Conformidad. NC. No Conformidad. NR. Nuevo Requisito. MR. Modificación de Requisito.
versión	char(10)	Representa la versión de la No Conformidad que se revisa.

Tabla 34. Descripción de la tabla “T_InfNConf T_NoConf” de la base de datos.

Nombre: T_InfNConf T_NoConf
Descripción: Surge de la relación [muchos a muchos] entre las tablas “T_InfNConf” y “T_NoConf”, ya que un Informe de No Conformidad puede contener varias No Conformidades al igual que una No Conformidad puede pertenecer a varios Informes de No Conformidad diferentes. Se han agregado a esta relación los atributos de las No Conformidades que son específicos para cada una de ellas.

Atributo	Tipo	Descripción
nombreInf	char(50)	Nombre del Informe de No Conformidad al que pertenece la No Conformidad.
nombreNC	char(40)	Nombre de la No Conformidad.
Id	int	Identificador correspondiente a los datos de la No Conformidad en la tabla "T_DatosNC".
fecha_ini	datetime	Fecha de creación de la No Conformidad.
recomendación	nvarchar(1000)	Recomendaciones que se hacer respecto al error que describe la No Conformidad.
imagen	image	Imagen que representa el error descrito en la No Conformidad de forma visual.
Descripción	nvarchar(1000)	Descripción del error detectado.
aspecto	nvarchar(1000)	Otros aspectos que vayan a ser descritos.

Tabla 35. Descripción de la tabla "T_Usuario T_NoConf" de la base de datos.

Nombre: T_Usuario T_NoConf		
<p>Descripción: Surge de la Relación [muchos a muchos] que existe entre las tablas "T_Usuario" y "T_NoConf", ya que a un Usuario se le pueden asignar varias No Conformidades y una No Conformidad pudiera ser resuelta por dos Usuarios (Usuario que conformó en Informe de No Conformidad al que pertenece la No Conformidad y otro Usuario al que se le asigne la No Conformidad).</p>		
Atributo	Tipo	Descripción
usuario	nvarchar(50)	Campo de la tabla "T_Usuario" que indica el Usuario al que se le ha asignado una No Conformidad.
nombreNC	char(40)	Campo de la tabla "T_NoConf" que indica el nombre de la No Conformidad que se le ha asignado al Usuario.
id	int	Campo de la tabla "T_DatosNC" que indica el id de los datos pertenecientes a la No Conformidad.

inc	char(50)	Indica el nombre del Informe de No Conformidad al que pertenece la No Conformidad.
-----	----------	--

Tabla 36. Descripción de la tabla “T_Entrada” de la base de datos.

Nombre: T_Entrada		
Descripción: Almacena la fecha y hora del último acceso a una tabla.		
Atributo	Tipo	Descripción
tabla	nvarchar(30)	Nombre de la tabla a la que se ha accedido.
fecha	datetime	Fecha en la que se ha accedido.

3.6 Mecanismo de seguridad y Tratamiento de Errores.

Se abordará ahora el tema relacionado con la seguridad y el tratamiento de errores. En el caso de este sistema existe el módulo de administración, descrito anteriormente, en el cual se implementa el tema de la seguridad en la aplicación. Cada uno de los Usuarios que harán uso de la aplicación cuenta con un usuario y contraseña de acceso para ello. Además la contraseña se almacena en la base de datos encriptada. En aras de mejorar la seguridad se definieron Roles con determinados permisos a ejecutar en el sistema para que cada uno de los Usuarios acceda solamente a lo que realmente necesite. Unido a lo anterior la aplicación cuenta con una validación de SQL Injection, ya que como la información es almacenada en una base de datos en SQL Server 2000, hay que garantizar que la misma no sufra alteraciones producto a la entrada de sentencias sql por parte de los usuarios.

Para el tratamiento de errores cada una de las funciones implementa su funcionalidad dentro de un bloque que garantiza el tratamiento de cualquier error que ocurra, notificándolo a través de las capas definidas, hasta llegar a la aplicación cliente donde se le muestra al usuario.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA.

4.1 Introducción.

En este capítulo se procede a la implementación del sistema y a la realización de pruebas del software para a través de las mismas obtener una revisión final de las especificaciones, del diseño y de la codificación.

Para el desarrollo de este capítulo se tuvieron en cuenta algunos artefactos que fueron generados en el flujo de análisis y diseño. A partir de los cuales se desarrolló la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

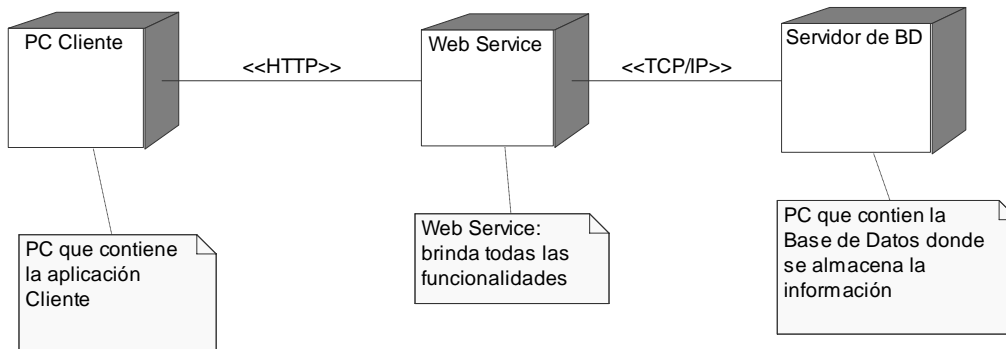
4.2 Modelo de Implementación.

El modelo de implementación constituye la vista de Implementación de la arquitectura, y como tal guía las labores de construcción del sistema. Este contiene fundamentalmente los subsistemas de implementación, incluyendo las dependencias y otras informaciones necesarias para su utilización.

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la última descripción física de la topología del sistema y describe la estructura de las unidades de hardware. Además, representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes.

4.3 Diagrama de Despliegue.



Para lograr una mejor comprensión de los componentes que forman el sistema, se presentarán los diagramas de componentes.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándar de componentes son los siguientes:

- <<executable>> es un programa que puede ser ejecutado en un nodo.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de base de datos.
- <<document>> es un documento.

Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas.

A continuación se muestran los componentes generados para la aplicación. (Ver Figuras (Anexos))

4.4 Modelo de Prueba.

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

Dentro de las pruebas se encuentran dos fundamentales: Prueba de Caja Negra y de Caja Blanca.

La prueba de la **caja blanca** del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

La prueba de caja negra es la que se le aplicó al sistema. A continuación se muestran los casos de prueba.

- 1 Caso de prueba donde el usuario y contraseña son correctos.
- 2 Caso de prueba donde el usuario es correcto y la contraseña incorrecta.
- 3 Caso de prueba donde el usuario es incorrecto y la contraseña es correcta.
- 4. Caso de prueba donde el usuario y la contraseña son incorrectos.
- 5 Caso de prueba donde el usuario y la contraseña existen, pero no corresponden.

Tabla 37. Caso de prueba “Autenticarse”.

Caso de Uso	Autenticarse		
Caso de Prueba	Entrada	Condiciones	Resultados
1	usuario = “amary” contraseña = “amary”	Como los datos son correctos, la autenticación debe ser satisfactoria	Autenticación satisfactoria, entrada al sistema.
2	Usuario =”amary” Contraseña=”vgvg”	Como la contraseña es incorrecta el sistema debe notificar el error e impedir el acceso al sistema.	Mensaje de error “Verifique que el usuario y contraseña sean correctos”.
3	Usuario =”yo” Contraseña=” amary”	Como el usuario es incorrecto el sistema debe notificar el error e impedir el acceso al sistema.	Mensaje de error “Verifique que el usuario y contraseña sean correctos”.
4	Usuario =”yo”	Como el usuario y	Mensaje de error

	Contraseña="yo"	contraseña son incorrectos el sistema debe notificar el error e impedir el acceso al sistema.	"Verifique que el usuario y contraseña sean correctos".
5	Usuario ="amary" Contraseña="luis"	Aunque el usuario y contraseña existen no se corresponden. El sistema debe notificar el error e impedir el acceso al sistema.	Mensaje de error "Verifique que el usuario y contraseña sean correctos".

- 1 Caso de prueba donde el Informe de No Conformidad no existe en el sistema.
- 2 Caso de prueba donde el Informe de No Conformidad existe en el sistema.

Tabla 38. Caso de prueba "Gestionar INC".

Caso de Uso	Gestionar INC [Crear Informe de No Conformidad]		
Caso de Prueba	Entrada	Condiciones	Resultados
1	Nombre INC= "Informe 3"	Como el INC no existe debe ser insertado.	Inserta el INC y muestra el mensaje: "En Informe de No Conformidad se ha creado correctamente".
2	Nombre INC= "Informe 3"	Como el INC ya existe no puede insertarse. Debe notificarse el error.	Muestra el mensaje: "El Informe de No Conformidad ya existe".

- 1 Caso de prueba donde los datos de la No Conformidad no existen en el sistema.
- 2 Caso de prueba donde los datos de la No Conformidad existen en el sistema.

Tabla 39. Caso de prueba “Gestionar NC”.

Caso de Uso	Gestionar NC [Crear No Conformidad]		
Caso de Prueba	Entrada	Condiciones	Resultados
1	Nombre de la NC = “NC 2”, INC =”Informe 1”, Responsable= “Amary [amary]”, Iteración = 1, Versión = 1.6, Estado=”Inconformidad Resuelta”, Falla =”Capa de Presentación”, Importancia = “Baja”, Tipo = “Nuevo Requisito”, Aspecto= “a”, Recomendación = “a”, Descripción = “a”.	Como los datos de la NC no existen, esta debe ser insertada y luego el sistema debe brindar la posibilidad de notificar por correo electrónico a los responsables, además de notificar el resultado de la operación.	Insertó los datos de la NC satisfactoriamente, preguntó si se deseaba o no enviar el correo de notificación, notificó el resultado de la operación y cargó la interfaz donde se encontraba el usuario anteriormente.
2	Los datos del caso de prueba 1.	Como los datos de la NC ya existen en el sistema este debe notificarlo.	Mensaje: “Ya existen los datos de la No Conformidad en el Informe”

4.5 Conclusiones.

En este capítulo se mostró cómo a través de la implementación, se produjo un refinamiento de la vista de la arquitectura del modelo de despliegue, donde los componentes ejecutables fueron asignados a nodos. Además cómo el modelo de implementación fue la entrada principal de las etapas de prueba que se realizan seguido de la implementación. Donde se verificó el resultado de esta probando cada construcción, incluyendo las versiones finales del sistema.

Se utilizaron diagramas de componentes para representar a través de un grafo los componentes de software unidos por medio de relaciones de dependencia; con los cuales se modeló la vista estática de un



CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA.

sistema. Además sirvieron para mostrar la organización y las dependencias lógicas entre un conjunto de componentes software. En este momento, ya se tiene el producto de software.

CONCLUSIONES

Al término del presente proyecto:

1. Se logró una especificación de los requisitos fundamentales para el diseño del Sistema.
2. Se definieron los casos de uso que satisfacen los requisitos especificados.
3. Se obtuvo un diseño de la base de datos.
4. Se documentó, de acuerdo con RUP, lo desarrollado.
5. Se implementó lo que se ha definido para esta etapa de desarrollo.

Este trabajo ha permitido al autor:

1. Familiarizarse con la tecnología .Net y los servicios Web.
2. Profundizar en el conocimiento del Proceso Unificado de Desarrollo de Software.
3. Realizar un pequeño aporte en el Proyecto de Sistema de Gestión Penitenciaria.

RECOMENDACIONES

- Hacer la modelación e implementación de los módulos de solicitud de cambio, listas de chequeo, casos de prueba.
- Hacer las modificaciones para que los implementadores haciendo uso de este sistema puedan notificar cuando han resuelto las no conformidades generadas.

BIBLIOGRAFÍA REFENRECIADA

1. **Fernández Carrasco, Oscar, García León, Delba y Beltrán Benavides, Alfa.** Un enfoque actual sobre la calidad del software. *Un enfoque actual sobre la calidad del software*. [En línea] Septiembre-Diciembre de 1995. [Citado el: 15 de Marzo de 2007.] http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
2. Prueba de Software. *Prueba de Software*. [En línea] e-Quality Corp. [Citado el: 25 de Marzo de 2007.] <http://www.e-quality.net/definiciones.php>.
3. IBM. *IBM*. [En línea] IBM. [Citado el: 28 de Abril de 2007.] http://www-306.ibm.com/software/info/ecatalog/es_ES/?&S_TACT=none&S_CMP=none.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education. S.A, 2000.

BIBLIOGRAFÍA

1. *Calidad del Software*. **Cueva Lovelle, Juan Manuel**. España : s.n., 1999.
2. *Estimación del Coste de la Calidad a través de la Simulación del Proceso de Desarrollo*. **Ruiz Carreira, Mercedes y Ramos Roman, Isabel**. 1, Colombia : s.n., Vol. 2.
3. **Guzmán Cortés, Oscar Hernando**. [En línea] 20 de Abril de 2004. [Citado el: 02 de Abril de 2007.] http://www.willydev.net/descargas/oguzman-diseno_pruebas.pdf.
4. **Gracia, Joaquin**. [En línea] 26 de Noviembre de 2006. [Citado el: 04 de Abril de 2007.] <http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>.
5. **Guzmán-Arenas, Adolfo**. [En línea] Octubre de 2003. [Citado el: 07 de 04 de 2007.] <http://www.cic.ipn.mx/aguzman/papers/165%20mitos,%20creencias%20y%20superst%20sobre%20la%20calidad%20del%20software.pdf>.
6. Intelligent Solutions. *Intelligent Solutions*. [En línea] I-Sol, 2000-2006. [Citado el: 10 de Abril de 2007.] <http://www.i-sol.com.ar/pg005.html>.
7. **Pérez Ortega, Joaquín, de la Cruz Fernández, Gerardo y Ramírez García, Arturo**. [En línea] Mayo-Junio de 2002. [Citado el: 17 de Abril de 2007.] <http://www.iie.org.mx/bolAT02/aplica1.pdf>.
8. **L. Guerini, Mario, y otros**. [En línea] [Citado el: 21 de Abril de 2007.] <http://www.itba.edu.ar/capis/webcapis/RGMITBA/comunicacionesrgm/CACIC-2006-Articulo-692.pdf>.
9. Software Expert. *Software Expert*. [En línea] SE. [Citado el: 25 de Abril de 2007.] <http://www.sequalitysoftware.com/estandares-atendidos.php>.
10. **S. Pressman, Roger**. *Ingeniería de Software. Un Enfoque Práctico*. Madrid : s.n., 2001.

GLOSARIO DE TÉRMINOS.

1 Fiabilidad: Software confiable, pocos errores. Se mide a través del número de mensajes de error, mientras más haya, contiene menos errores este software.

2 Flexibilidad o Adapatibilidad: Capacidad de adaptación a diferentes tipos y ambientes de uso. Se mide viendo a cuántos estándares este software se adhiere.

3 Robustez: Pocas fallas catastróficas. Se mide a través de pruebas y uso prolongado.

4 Comprensión: Capacidad de entender lo que el sistema hace. Se mide viendo la cantidad de comentarios que posee el software, y la extensión de sus manuales.

5 Modularidad: Se mide contando el número de módulos que forman un programa.

6 Complejidad: Se refiere a qué tan fácil es entender el código. Se mide contando el número de anidaciones en expresiones o postulados (“complejidad ciclomática”).

7 Portabilidad: Es la facilidad con que se eche a andar en otro sistema operativo distinto al que fue creado. Se mide preguntando a usuarios que han hecho estos trabajos.

8 Usabilidad: Es alta cuando el programa aporta gran valor agregado a nuestro trabajo. “Es indispensable contar con él”. Se mide viendo qué porcentaje de nuestras necesidades cubre ese programa.

9 Reutilización: Se mide por la cantidad de veces que partes del programa se han reutilizado en otros proyectos de desarrollo de software.

10 Eficiencia: Se mide en segundos, es la rapidez en su ejecución. (Esta medición no tiene objeción, se mide lo que se quiere medir).

11 Principio Tecnológico: Define las técnicas a utilizar en el proceso de desarrollo del software.

12 Principio Administrativo: Contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.

13 Principio Ergonómico: Define la interfaz entre el usuario y el ambiente automatizado.

14 ISO: Organización Internacional para la Estandarización. Plantea qué hacer para responder a los requerimientos de un mercado cada vez más competitivo y cómo deben responder los proveedores y compradores respecto a la calidad de los bienes o servicios intercambiados.

15 IEEE: Instituto de Ingenieros Eléctricos y Electrónicos. Asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación e ingenieros en telecomunicación. Se creó en 1884.

16 CMM: Modelo de Madurez y Capacidad del Proceso de Software. Es un esquema de diagnóstico y de evaluación de la madurez del proceso de desarrollo de software. Ayuda a las organizaciones para producir de manera consistente y predecible productos de calidad superior. Su principal objetivo es el de producir productos que cumplan los requerimientos del usuario.

17 UML: Lenguaje Unificado de Modelación. Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

18 Programación Multicapas: Se definen tres capas: Presentación, Reglas del Negocio y Acceso a Datos. Presentación: En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el sistema. Reglas del Negocio: Contiene todas las subrutinas creadas con el propósito de regular alguna acción del usuario. Acceso a Datos: En esta capa se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación.

19 Servicios Web: Son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet.

20 Protocolos Web: Definen reglas de comunicación.

21 XML: Lenguaje de Marca Extensible. Es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas. Es la base de los Servicios Web, y a pesar de su sencillez aparente, está transformando completamente la creación y el uso de software. Es la solución a un problema de comunicación entre programas de ordenador, pues la información generalmente queda fuertemente ligada al programa con el cual fue creada,

y es así como se pierde mucho tiempo en pasar de un formato de definición a otro. El contenido almacenado en un documento XML se puede transferir fácilmente a través de la red.

22 SOAP: Protocolo de acceso simple a objetos. Es un protocolo de mensaje liviano basado en XML, usado para codificar los mensajes de Web Services antes de enviarlos por la red. Los mensajes SOAP son independientes de cualquier sistema operativo y protocolo, y pueden ser transportados usando una variedad de protocolos de internet, incluyendo HTTP, SMTP y MIME. Permite que programas que corren en diferentes sistemas operativos se comuniquen.

23 UDDI: Es un directorio universal de Servicios Web basado en XML, que permite publicar, localizar y utilizar servicios Web.

24 WSDL: Lenguaje de descripción de servicios web. Es un lenguaje en formato XML que define las operaciones que proporciona un servicio, desarrollado conjuntamente por Microsoft e IBM.

25 Plataforma.Net: Se conoce por infraestructura .NET al conjunto de todas las tecnologías que conforman el nuevo entorno para desarrollar (tanto servicios Web como aplicaciones tradicionales, aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), y ejecutar aplicaciones escalables y distribuidas. La parte de .NET que nos permite desarrollar estas aplicaciones es el Framework .NET.