



Universidad de las Ciencias Informáticas

Facultad 6.

**Herramienta Informática para automatizar los procesos en el Laboratorio
de Calidad de Software: Módulo Gestión de las No Conformidades.**



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

Autores:

Martha Nieves Borrero.

Asnier Góngora Rodríguez.

Tutores:

Ing. Ana M. Abreu Sáez.

Ing. Ramsés Delgado Martínez.

Ciudad de La Habana, Cuba.

Julio 2007.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Dirección de Calidad de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Martha Nieves Borrero

Asnier Góngora Rodríguez

Firma del primer Autor

Firma del segundo Autor

Ing. Ana Margarita Abreu Sáez

Ing. Ramsés Delgado Martínez

Firma del primer Tutor

Firma del segundo Tutor

Agradecimientos

A nuestros padres por su inmenso apoyo y dedicación.

A Aylín Febles por darnos la oportunidad de trabajar en este proyecto.

A Michael e Irina porque a ellos le debemos mucho de nuestros conocimientos.

A Ana Margarita por ser la más preocupada y atenta.

A Delvis y Roig por ser los mejores amigos y compañeros de proyecto.

A Maikel por su inmensa ayuda y dedicación en todo momento.

A Yanet por su apoyo incondicional y su perfeccionismo.

A Liusmila y Guillermo por su constante ayuda y apoyo.

A Dailien, Yetel y Yoendris por su amistad incondicional.

A la Revolución por brindarnos esta gran oportunidad y confiar tanto en nosotros.

A nuestros profesores de la UCI que han influido en nuestra preparación profesional.

Dedicatoria

Martha

A todas las personas que contribuyeron a que este sueño se hiciera realidad, en especial:

A mi mamá Martha Borrero y mi papá Fidelio Nieves por ser mi razón de ser.

A mi hermana Milena por su constante apoyo y dedicación.

A mi abuela Tata por ser mi fuente de inspiración y mi gran motivación.

A Lety, Yusef, Maritza y Kaky por su gran ayuda y apoyo en todo momento.

A Luisito, Bryan y Vida por ser los niños más lindos del mundo y la lucecita que me ilumina todos los días.

A Haymée y Haylín por enseñarme a ser una mejor persona.

A Rosa y Betty por llevarme siempre por el buen camino.

A Didiana por quererme tanto y ser mi mejor amiga.

A Dani y Alty por su inmensa ayuda en estos cinco años.

A Ana Margarita por ser más que mi tutora.

A Yanni y Brend que tanto esperaron a que llegara este día.

A los que con su ejemplo nos animan cada día a ser mejores.

Asnier

A mi madre Isabel Rodríguez por estar siempre presente cuando la necesito, por brindarme su amor y cariño todos estos años. Por confiar siempre en mí y por lo mucho que la quiero.

A mi padre Felipe Góngora por brindarme su apoyo en todo momento y hacer posible que este momento ocurra. Por su confianza y esfuerzo todos estos años.

A mi hermano Félix Góngora por brindarme su ayuda en todo momento.

A mis abuelos Lilia Marrero, Enrique Rodríguez y Luz María que con su amor siempre me han dado fuerzas para continuar.

Resumen

La investigación está asociada al surgimiento del Laboratorio Central de Certificación de Software en la Universidad de las Ciencias Informáticas, que se espera sea el Centro Nacional de Calidad de Software Cubano. Se pretende dar respuesta a la siguiente interrogante: ¿Cómo definir y automatizar el proceso de gestión y reportes de las No Conformidades en el Laboratorio de Calidad de software de la UCI? Para el logro de este objetivo se implementó una herramienta informática que automatice el proceso de Gestión de las No Conformidades diseñando una aplicación Web que sea capaz de tramitar todos los elementos de las No Conformidades. Esta aplicación les permitirá, a los Asesores de Calidad y al personal autorizado, interactuar a través de la red con la información almacenada en la aplicación así como hacer reportes de seguimiento para llevar un control minucioso de todo lo referente a las pruebas de calidad de software. Para garantizar la portabilidad de la aplicación se desarrollará bajo ambientes multiplataforma, fundamentalmente con herramientas de software libre.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica.	6
1.1 Calidad del Software.	6
1.1.1 Estado Actual.	8
1.2 Pruebas de Software. No Conformidades.	9
1.3 ¿Qué es una No Conformidad?.....	10
1.4 Definición del Proceso de Gestión de las No Conformidades.....	10
1.5 Aplicaciones desarrolladas para la Gestión de No Conformidades. Limitaciones.	25
1.5.1 SIGNO - Sistema para la Gestión de No Conformidades.	25
1.6 Tendencias y tecnologías actuales. Metodología utilizada.	26
1.6.1 Metodología Utilizada	26
1.6.2 Visual Paradigm.	26
1.6.3 Dreamweaver.....	27
1.6.4 ¿ASP o PHP?	28
1.6.5 ¿Por qué MySQL?.....	29
1.6.6 Java	30
1.6.7 ¿Servidor Apache?.....	30
1.6.8 WAMP 5.....	32
1.6.9 Arquitectura Cliente-Servidor.....	32
Capítulo 2. Características del sistema.	36
2.1 Objeto de Estudio.....	36
2.1.1 Situación Problémica.....	36
2.1.2 Objeto de automatización.	36
2.1.3 Información que se maneja.	37
2.1.4 Propuesta de sistema.....	37
2.2 Modelo de negocio.....	39
2.2.1 Actores del negocio.....	39
2.2.2 Trabajadores del negocio.	40
2.2.3 Diagrama de Casos de Uso del Negocio.	40
2.2.4 Casos de Uso del Negocio	41
2.2.5 Diagramas de Actividades.....	42
2.3 Definición de Requisitos Funcionales.....	44
2.4 Definición de requisitos no funcionales.	51
2.5 Actores del sistema a automatizar	53
2.6 Listado de casos de uso.....	54
2.7 Paquetes y sus relaciones	56
2.8 Diagrama de casos de uso del sistema.....	59
2.8.1 El paquete Asignar trabajo.	59
2.8.1.1 Descripción de los casos de uso del Paquete Asignar trabajo.....	60
2.8.2 El paquete Gestionar No Conformidades.	65
2.8.3 El paquete Insertar interrupción.....	71
2.8.4 El paquete Asignar tiempo de trabajo.....	72
2.8.5 El paquete Insertar respuestas a las No Conformidades.	74

2.8.6 El paquete Mostrar estados.....	75
2.8.7 El paquete Mostrar reportes.....	77
2.8.8 El paquete Administrar usuario.....	79
2.8.9 El paquete Autenticar usuario.....	82
2.9 Casos de uso por ciclo.....	83
Capítulo 3. Análisis y diseño del sistema.....	85
3.1 Modelo de Análisis.....	85
3.1.1 Diagrama de clases de Análisis.....	85
3.2 Modelo de diseño.....	87
3.2.1 Arquitectura Web de tres niveles.....	88
3.2.2 Diagramas de secuencia del diseño.....	89
3.2.3 Diagrama de clase del Diseño.....	93
3.3 Diseño de la Base de Datos.....	97
3.3.1 Descripción de las tablas.....	97
3.4 Principios de diseño.....	97
3.5 Tratamiento de errores.....	98
3.6 Seguridad.....	100
3.7 Concepción de la ayuda.....	100
Capítulo 4. Implementación y prueba.....	102
4.1 Implementación.....	102
4.1.1 Diagrama de despliegue.....	102
4.1.2 Diagramas de componentes.....	102
4.2 Modelo de prueba.....	105
Conclusiones.....	107
Recomendaciones.....	108
Referencia Bibliográfica.....	109
Bibliografía.....	110
Anexo 1: Diagramas de clases persistentes.....	112
Anexo 2. Descripción de las Tablas de la Base de Datos.....	113
Anexo 3. Modelo de Caso de Prueba.....	119

Índice de tablas

Tabla 1.1	Actividades del proceso Gestión las no conformidades	12
Tabla 2.1	Descripción de los actores del negocio	40
Tabla 2.2	Descripción de los trabajadores del negocio	40
Tabla 2.3	Descripción de los actores del sistema	54
Tabla 2.4	Caso de uso "Insertar elemento de prueba"	54
Tabla 2.5	Caso de uso "Buscar elemento de prueba"	54
Tabla 2.6	Caso de uso "Asignar trabajo"	54
Tabla 2.7	Caso de uso "Mostrar datos de los elementos de prueba"	55
Tabla 2.8	Caso de uso "Gestionar las No Conformidades"	55
Tabla 2.9	Caso de uso "Gestionar reportes"	55
Tabla 2.10	Caso de uso "Asignar tiempo de realización de un elemento de prueba"	55
Tabla 2.11	Caso de uso "Insertar interrupción"	55
Tabla 2.12	Caso de uso "Insertar respuestas a las No Conformidades2"	55
Tabla 2.13	Caso de uso "Mostrar estados"	56
Tabla 2.14	Caso de uso "Mostrar reportes de las No Conformidades"	56
Tabla 2.15	Caso de uso "Gestionar usuario"	56
Tabla 2.16	Caso de uso "Autenticar usuario"	56
Tabla 2.17	Descripción del caso de uso "Insertar elemento de prueba"	62
Tabla 2.18	Descripción del caso de uso "Asignar Trabajo"	64
Tabla 2.19	Descripción del caso de uso "Buscar elemento de prueba"	65
Tabla 2.20	Descripción del caso de uso "Gestionar las No Conformidades"	69
Tabla 2.21	Descripción del caso de uso "Mostrar datos de los elementos de prueba"	70
Tabla 2.22	Descripción del caso de uso "Gestionar reportes"	71
Tabla 2.23	Descripción del caso de uso "Insertar interrupción"	72
Tabla 2.24	Descripción del caso de uso "Asignar tiempo de realización de un elemento de prueba"	73
Tabla 2.25	Descripción del caso de uso "Insertar respuestas a las No Conformidades"	75
Tabla 2.26	Descripción del caso de uso "Mostrar estados"	76
Tabla 2.27	Descripción del caso de uso "Mostrar reportes de las No Conformidades"	79
Tabla 2.28	Descripción del caso de uso "Gestionar usuario"	82
Tabla 2.29	Descripción del caso de uso "Autenticar usuario"	83
Tabla 2.30	Casos de uso por ciclo	84

Índice de figuras

Figura 1.1 Esquema del proceso	13
Figura 2.1 Diagrama del Modelo del Negocio.	41
Figura 2.2 Diagrama de actividades: Brindar Reporte de las No Conformidades.	42
Figura 2.3 Diagrama de actividades: Ejecutar Prueba	43
Figura 2.4 Diagrama de clases del modelo de objetos	44
Figura 2.5 Diagrama de Paquetes	58
Figura 2.6 Diagrama de Actores del Sistema.	59
Figura 2.7 Diagrama de paquete “Asignar Trabajo”.	60
Figura 2.8 Diagrama de paquete “Gestionar No Conformidades”	66
Figura 2.9 Diagrama de paquete “Insertar Interrupción”	71
Figura 2.10 Diagrama de paquete “Asignar tiempo de Trabajo”	73
Figura 2.11 Diagrama de paquete “Insertar respuesta a las No Conformidades”	74
Figura 2.12 Diagrama de paquete “Mostrar Estados”	75
Figura 2.13 Diagrama de paquete “Mostrar reportes”.	77
Figura 2.14 Diagrama de paquete “Administrar usuario”	79
Figura 2.15 Diagrama de paquete “Autenticar usuario”	82
Figura.3.1 Diagrama de Clases de Análisis Asignar Trabajo.....	86
Figura.3.2 Diagrama de Clases de Análisis Gestionar No Conformidades.....	86
Figura.3.3 Diagrama de Clases de Análisis Mostrar Reporte de las No Conformidades.	86
Figura 3.4 Arquitectura Web de 3 niveles.	89
Figura.3.5 Diagrama de secuencia del diseño Asignar Trabajo.	89
Figura.3.6 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Insertar No Conformidades).....	90
Figura.3.7 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Modificar No Conformidades).	91
Figura.3.8 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Eliminar No Conformidades).....	92
Figura.3.9 Diagrama de secuencia del diseño Reporte de No Conformidades.....	93
Figura.3.10 Diagrama de clase del diseño Asignar Trabajo.	94
Figura.3.11 Diagrama de clase del diseño Gestionar No Conformidades.	95
Figura.3.12 Diagrama de clase del diseño Mostrar reportes de las No Conformidades.....	96
Figura.3.13 Modelo Entidad Relación.	97
Figura 3.14 Mensaje de verificación. Eliminar Elemento de prueba.	99
Figura 3.15 Mensaje de validación. Insertar no conformidad.	99
Figura 3.16 Mensaje de validación. Campo Nombre del Elemento de prueba.	99
Figura 3.17 Mensaje de validación. Insertar Elemento de prueba.	99
Figura 3.19 Mensaje de verificación. Campo Contraseñas.	100
Figura 3.20 Interfaz de autenticación.	100
Figura 4.1 Diagrama de despliegue.	102
Figura 4.2 Diagrama de componentes Asignar trabajo.	103
Figura 4.3 Diagrama de componentes Gestionar No Conformidades.....	104
Figura 4.4 Diagrama de componentes Mostrar reportes de las No Conformidades.....	105

Introducción

La calidad en el desarrollo y mantenimiento de las aplicaciones del Software se han convertido, actualmente, en uno de los principales objetivos estratégicos de las entidades que se dedican a la producción debido a que cada vez más, los procesos principales de dichas empresas dependen de los sistemas informáticos para su buen funcionamiento.

En la evolución experimentada por la calidad del software se ha transitado de un tratamiento centrado fundamentalmente en la inspección y detección de errores, a una aproximación más sistemática, dada la importancia que ha adquirido la calidad en la ingeniería del software. [1]

En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora tanto de productos como de procesos de software. Todo ello ha influido de forma significativa en el papel que actualmente tiene la calidad en las organizaciones, convirtiéndose en una filosofía y una cultura que afecta a toda la organización.

El desarrollo acelerado de la ciencia y las tecnologías de la información en Cuba, así como la velocidad de cambio en el manejo de los negocios, ha traído como consecuencia que las empresas informáticas enfrenten cada día, un reto para brindar una respuesta rápida, eficaz y con calidad a los clientes, que cada vez se vuelven mas exigentes, no sólo en cuanto al precio sino también en la confiabilidad que deben brindar los productos de software.

Cualquier organización que se dedique a la producción o comercialización de software, debe tener en cuenta que la organización de sus procesos juega un papel determinante en la competitividad de la misma.

En el marco de estas problemáticas, surge en el 2002 la Universidad de las Ciencias Informáticas con dos objetivos fundamentales: Formar profesionales, comprometidos con su Patria altamente calificados en la

rama de la informática y producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

¿Cómo producir con calidad y disciplina industrial en un entorno esencialmente docente?

Toda Industria de Software trae consigo un Laboratorio de Calidad donde se certifican con un sello de calidad los productos terminados que están listos para salir al mercado y conjuntamente se mide la calidad de los productos que se están produciendo en la misma. La UCI se ha convertido en una industria productora de Software, ya que se han creado varios productos que han sido comercializados internacionalmente. Por esta razón surgió el Laboratorio Central de Certificación de Software en la UCI, aunque no está oficialmente constituido a nivel nacional se espera que sea el Centro Nacional de Calidad de Software Cubano.

En estos momentos todo el trabajo que se realiza en el Laboratorio de Calidad, - que debe quedar almacenado para su uso y conocimiento posteriormente por parte de los estudiantes y demás personal relacionado con el tema - se realiza de forma manual, por lo que los métodos que se están usando no son los más eficientes para estos fines, pues las informaciones que se almacenan son cada vez más grandes y trae consigo las siguientes consecuencias:

- Pérdida de tiempo y derroche de los recursos. Debido a esto no se puede hacer una buena gestión de las No Conformidades y de los reportes de las mismas.
- La accesibilidad, por parte de los usuarios, de la información de los reportes de las No Conformidades de manera rápida es insuficiente pues no existe una publicación de la misma en la red, ni un buen intercambio de información entre los asesores de calidad y el equipo de desarrollo.
- No hay seguridad ni integridad de la documentación de entrega por parte del equipo de desarrollo, pues la misma no cuenta con ninguna protección.

Por todo lo antes expuesto se plantea como **problema científico** de esta investigación:

¿Cómo contribuir al desarrollo de los procesos de calidad de software?

Para dar respuesta a este problema, se decidió diseñar una herramienta informática que automatice el proceso de Gestión de las No Conformidades creando una aplicación que sea capaz de tramitar todos los elementos de las No Conformidades que le permitirán a los Asesores y al personal autorizado interactuar a través de la red con la información almacenada en la aplicación, así como hacer reportes de seguimiento para llevar un control minucioso de todo lo referente a las pruebas de calidad de software.

Por tanto el **objeto de estudio** es: El desarrollo de los procesos de calidad de software.

El **campo de acción** esta determinado por: El proceso de Gestión de las No Conformidades para las actividades de un laboratorio de pruebas y certificación en la UCI.

Este trabajo tiene como **objetivo general**: Definir el proceso de Gestión de las No Conformidades y desarrollar una aplicación Web que permita elevar la eficiencia del proceso de pruebas del laboratorio de certificación de la UCI.

Para cumplir el objetivo general se trazaron los siguientes **objetivos específicos**

- Definir el proceso de Gestión de las No Conformidades para el laboratorio de calidad.
- Implementar una aplicación web a partir del proceso definido.

Se plantean las siguientes **tareas** para poder dar cumplimiento de forma exitosa a los objetivos planteados:

- Realización de entrevistas a especialistas de Calidad para definir los procesos a partir de la experiencia adquirida.
- Definición del proceso de Gestión de las No Conformidades en un laboratorio de calidad.
- Realización del modelado del negocio de nuestra herramienta informática.
- Realización de entrevistas con el cliente para efectuar el levantamiento de requisitos.
- Realización del flujo de trabajo de análisis y diseño.
- Implementación y prueba de la herramienta que proponemos.

Los **aportes prácticos** que se esperan obtener son:

- Centralización de los reportes de No Conformidades de todos los proyectos de la universidad, permitiendo la trazabilidad con los casos de uso y los requerimientos que le dan origen a las mismas.
- Rapidez en la comunicación entre el laboratorio de calidad y el equipo de desarrollo.
- Rapidez en la búsqueda de información de las No Conformidades y las respuestas de las mismas.

La investigación esta estructurada por: Introducción, 4 capítulos, conclusiones, recomendaciones, bibliografías y anexos.

En el **Capítulo 1 Fundamentación Teórica**: se abordan los objetivos estratégicos de la organización, una breve valoración del flujo actual de los procesos y la definición de los mismos, su análisis crítico, así como las tendencias y tecnologías actuales relacionadas con el objeto de estudio.

En el **Capítulo 2 Características del Sistema**: se refleja un análisis detallado del flujo actual de los procesos, los trabajadores que desarrollan dichos procesos y otras personas involucradas. Se define el objeto de automatización, la información que se maneja y la propuesta del sistema. Se identifican los requisitos funcionales y no funcionales, los casos de uso con sus descripciones, los diagramas de actividades y el modelo de objeto.

En el **Capítulo 3 Análisis y Diseño del Sistema**: se analizan los casos de uso del sistema para diseñar las clases que se implementarán, se representan los diagramas de secuencia del diseño, el diagrama de las clases diseñadas con sus relaciones, los principios utilizados para el diseño de dichas clases, el diagrama de clases persistentes, el diagrama entidad relación y las descripción de las tablas de la base de datos.

En el **Capítulo 4 Implementación y Prueba**: se reflejan los diferentes diagramas correspondientes a este flujo de trabajo, donde se implementara la propuesta realizada en el análisis y diseño, se representa el diagrama de despliegue, los de componentes y el modelo de prueba correspondiente a la aplicación desarrollada.

Capítulo 1: Fundamentación Teórica.

Introducción.

En este capítulo se abordarán los objetivos estratégicos de la organización, una breve valoración del flujo actual de los procesos y la definición de los mismos, su análisis crítico, así como las tendencias y tecnologías actuales relacionadas con el objeto de estudio.

1.1 Calidad del Software.

La garantía de calidad es una actividad esencial en cualquier empresa que cree productos que van a ser usados por otros. Antes del siglo XX, la garantía de calidad era responsabilidad única de la persona que hacía el producto. La primera función de control y de garantía de calidad formal fue introducida por los laboratorios Bell en 1916 y se extendió rápidamente por todo el mundo de las manufacturas. Hoy en día, cada empresa tiene un mecanismo que asegura la calidad de sus productos realmente, durante la pasada década, se han usado ampliamente como tácticas de mercado la declaración explícita de mensajes que ponían de manifiesto la calidad ofrecida por las empresas.

La historia de la garantía de calidad en el desarrollo de software ha sido paralela a la historia en la fabricación de hardware. Durante los primeros años de información (los 50 y los 60), la calidad era responsabilidad únicamente del programador.

Durante los años 70 se introdujeron estándares de garantía de calidad para el software en los contratos militares de desarrollo de software y se han extendido rápidamente en el desarrollo del software del mundo comercial.

Los factores de calidad del software se centran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad de soportar los cambios y su adaptabilidad a nuevos entornos; y estos factores son los siguientes:

- 1 **Corrección.** El grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el cliente.

- 2 **Fiabilidad.** El grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida. Esta puede ser medida o estimada por datos históricos o estadísticos.
- 3 **Eficiencia.** La cantidad de recursos de computadora y de código requeridos por un programa para llevar a cabo sus funciones.
- 4 **Integridad.** El grado en que puede controlarse el acceso al software o a los datos, por personal no autorizado.
- 5 **Facilidad de uso.** El esfuerzo requerido para aprender un programa, trabajar con él, preparar su entrada e interpretar su salida.
- 6 **Facilidad de Mantenimiento.** El esfuerzo requerido para localizar y arreglar un error de un programa.
- 7 **Flexibilidad.** El esfuerzo requerido para modificar un programa operativo.
- 8 **Facilidad de prueba.** El esfuerzo requerido para probar un programa de manera que se asegure que realiza su función requerida.
- 9 **Portabilidad.** El esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistemas de software a otro.
- 10 **Reusabilidad.** El grado en que un programa o partes de un programa se puede reusar en otras aplicaciones. Esto va relacionado con el empaquetamiento y el alcance de las funciones que realiza el programa.
- 11 **Facilidad de interoperación.** El esfuerzo requerido para acoplar un sistema a otro.

En nuestro país contamos con una naciente Industria del Software que se ha desarrollado en los últimos años con el objetivo de producir software para las empresas y entidades nacionales con el fin de automatizar la sociedad cubana. DESOFT es la empresa encargada del desarrollo del Software en Cuba. Existen otras empresas como Softel que se desempeña generalmente en productos para la salud. Con estos nuevos retos que han surgido en nuestro país, las empresas y organizaciones se han visto en la necesidad de brindar una respuesta rápida a los clientes que cada vez se vuelven más exigentes, no solo con el costo, sino con la confiabilidad, funcionalidad, integridad, portabilidad y eficiencia del producto. En consecuencia a todas estas exigencias las empresas se ven en la obligación de garantizar la calidad del producto para satisfacer las expectativas del cliente. En Cuba no contamos con un laboratorio que pruebe y certifique los productos desarrollados por las diversas empresas solo contamos con departamentos de control de la calidad internos en cada empresa.

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. [1]

1.1.1 Estado Actual.

En la actualidad la informática ha avanzado en el mundo, cada día se hacen más software y los clientes u organizaciones tienen más necesidad y requisitos que quieren automatizar. Con este avance de la informática los clientes quieren que el software que ellos compran sean más avanzados y con menos defectos. Los desarrolladores necesitan que alguien evalúe los productos terminados que compran para tener una garantía de que el software tiene calidad. Por ese motivo surgen modelos, técnicas y normas de calidad que establecen como hacer un software con calidad desde los procesos que lo inician hasta el producto terminado.

En estos momentos la calidad crece conjuntamente con la Industria del Software ha sido un impacto estratégico, una oportunidad de ventaja competitiva una filosofía, una cultura, una estrategia, un estilo de gerencia de las empresas.

¿Cuba queda exenta?

En los últimos años en Cuba ha avanzado en gran medida con relación a este tema, y es necesario que estos productos se certifiquen para que poder ser insertados en el mercado nacional e internacional.

Existen en el mundo varias empresas y organizaciones que se dedican aplicar pruebas de software para poder certificar un producto con la calidad y la excelencia requerida.

En nuestro país en estos momentos no existen herramientas que gestionen las No Conformidades ya que no contamos con una Empresa de Calidad de Software que certifique estos productos terminados. Se están tomando una serie de medidas en el ámbito nacional para definir procesos y crear una entidad que retome algunos trabajos realizados por el Laboratorio de Calidad de la UCI y exista como tal el Centro Nacional de Calidad de Software. La Universidad de las Ciencias Informáticas surgida hace 4 años es un ejemplo de empresa y organización que se dedica a la producción de software. A raíz de este incremento en el desarrollo del software a nivel nacional se creó un Laboratorio de Calidad en el cual se revisan los productos terminados de la universidad y que posteriormente se comercializarán tanto en nuestro país como a nivel internacional.

La gestión de las No Conformidades es uno de los procesos que se desarrollan en un Laboratorio de Calidad, con la automatización de estos se logra más rapidez en los reportes y eficiencia en el trabajo que se realiza en el Laboratorio de Calidad referentes a las No Conformidades. Es por esta razón que surge la necesidad de agrupar toda la información que interviene en el proceso de revisión de un determinado producto para ser tratada y almacenada en Bases de Datos para posteriores revisiones y por ser información clasificada de extrema seguridad.

Para dar solución a este problema hemos decidido diseñar una aplicación Web que integre la planificación de los recursos humanos y materiales, que gestione las No Conformidades, haga seguimiento a dichos elementos y guarde toda la información de las pruebas realizadas a los diferentes productos terminados.

1.2 Pruebas de Software. No Conformidades.

Las pruebas de software son una técnica clásica para la verificación y validación del software, junto a otras alternativas de evaluación como revisiones, auditorías, métricas, entre otros.

Entendemos las pruebas de software como “la ejecución de un programa con el fin de detectar defectos” [2]

El diseño de las pruebas se basa en la creación de casos de prueba cuya ejecución permitirá observar posibles síntomas de defectos. Se puede definir un caso de prueba como “el conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito” [2]

Aplicando las normas, técnicas y modelos (CMMI, ISO, CMM, entre otros) al software se le realizan varios tipos de pruebas para poder decidir la calidad que tiene el producto final. Toda la capacidad de detección de defectos de las pruebas se basa en la consideración de cualquier tipo de discrepancia entre la salida obtenida y la salida esperada como síntoma de un problema en el software.

1.3 ¿Qué es una No Conformidad?

De acuerdo a la definición en la norma ISO 9000: 2000 (3.6.2), una No Conformidad es: “un no cumplimiento a un requisito.” [3]

1.4 Definición del Proceso de Gestión de las No Conformidades.

La definición del proceso de Gestión de las No Conformidades que a continuación se muestra es el resultado de año y medio de trabajo en un laboratorio de pruebas de software de los autores de este trabajo y la experiencia de varios especialistas de calidad, en él se reflejan las principales actividades que involucra el proceso así como los responsables y participantes del mismo, este proceso que se describe forma parte de las actividades del laboratorio de calidad en la UCI.

Descripción de procesos	
Código	01.01
Proceso	Gestión de las No Conformidades.
Responsable	Especialista de calidad.
Participantes	Jefe de prueba, Probadores, Desarrolladores.
Clientes	Dirección de Calidad.

I- Misión

Garantizar que se realizan las pruebas a un software y obtener los reportes correspondientes.

II- Resultados a obtener

Las no conformidades a un software así como las respuestas a las mismas y los reportes correspondientes a ellas.

III- Funciones

Comunicar la revisión del software.

- Se le comunica el jefe de prueba que tiene que revisar un software por parte del Especialista de calidad, comunicándole el nombre del proyecto y entregándole la documentación necesaria para la revisión de un software.
- Repartir el trabajo entre los probadores.
- El Jefe de prueba reparte los casos de uso entre los probadores de su equipo de trabajo.
- Los Probadores elaboran los diseños de caso de prueba.
- Revisión del software.
- Los Probadores comienzan a revisar el software y elaboran un informe con las no conformidades encontradas.
- Se le comunica a los Desarrolladores que se terminó una iteración de la revisión del software y se le entrega las no conformidades encontradas.
- Responder las no conformidades.
- Los desarrolladores elaboran un informe con las respuestas a las no conformidades encontradas en la revisión.
- El Jefe de prueba comunica al especialista de calidad que se culminó la revisión del software sin ninguna no conformidad encontrada y este le comunica al solicitante.
- Comunicar la culminación de la revisión del software al solicitante.

IV- Relación de actividades del proceso

Código	Actividad	Resp.	Part.	Resultados a obtener
01.01	Comunicar la revisión del software.	Especialista de calidad	Especialista de calidad.	Que el Jefe de prueba sepa que tiene que comenzar la revisión de un software.
01.02	Repartir trabajo entre los probadores.	Jefe de prueba	Jefe de prueba	Que se elabore un informe con el trabajo repartido entre los Probadores.

01.03	Confección de los diseños de caso de prueba.	Probador	Probador	Que quede elaborado los diseños de caso de prueba para cada caso de uso.
01.04	Revisión del software.	Jefe de prueba	Jefe de prueba y Probador	Que se elabore un documento con las no conformidades encontradas en la revisión del software.
01.05	Comunicar a los desarrolladores la culminación de una iteración de la revisión del software.	Jefe de prueba	Jefe de prueba	Que los Desarrolladores sepan que se terminó la revisión del software y tengan las no conformidades encontradas en un documento.
01.06	Responder las no conformidades.	Desarrollador	Desarrollador	Que quede un informe con las respuestas a las no conformidades encontradas.
01.07	Comunicar la culminación de la revisión del software al solicitante.	Especialista de calidad	Especialista de calidad y Jefe de prueba	Que el solicitante sepa que se terminó la revisión al software.

Tabla 1.1 Actividades del proceso Gestión las no conformidades

V- Esquema del proceso

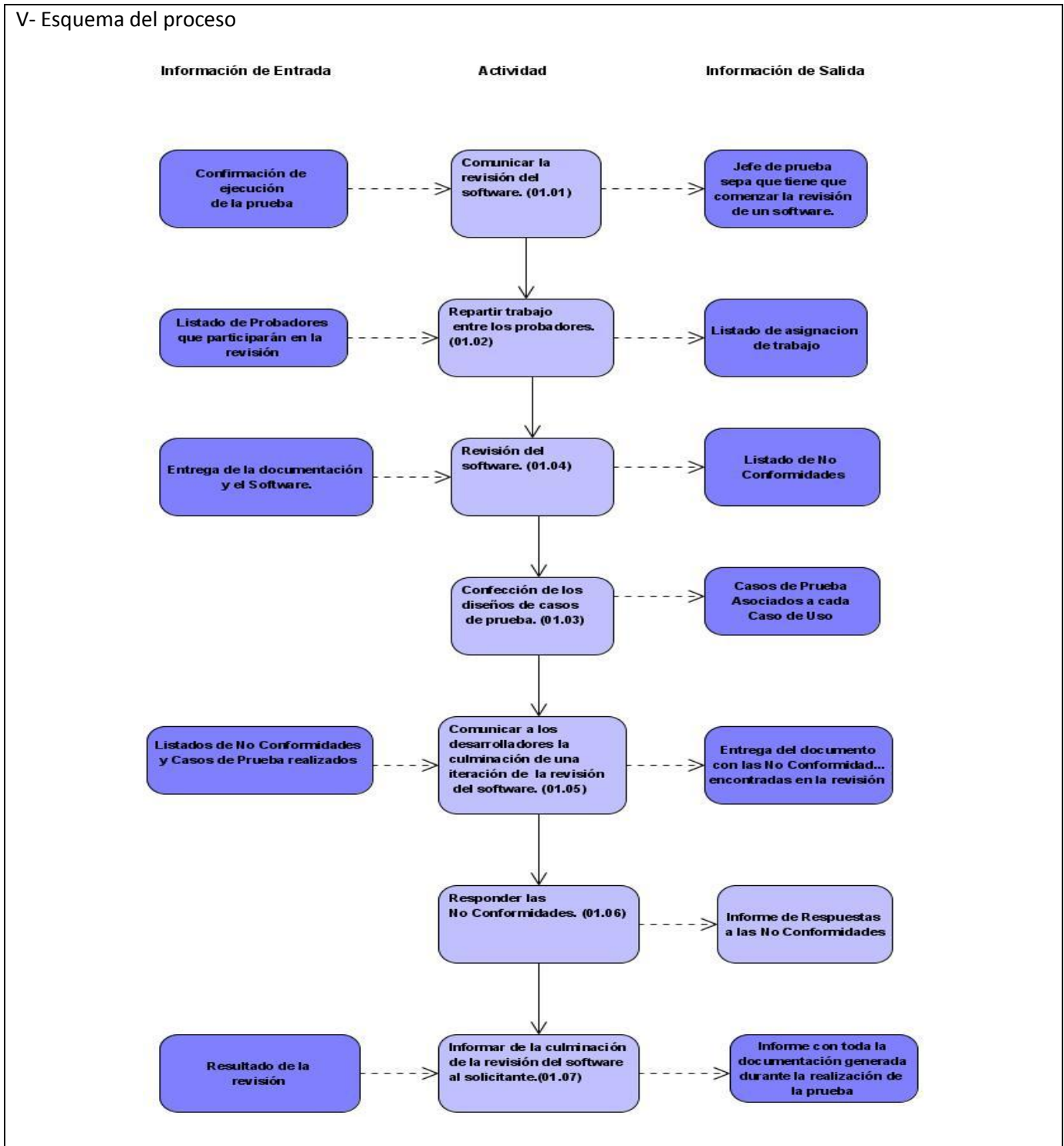


Figura 1.1 Esquema de proceso

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.01
Actividad	Comunicar la revisión del software.
Proceso	Gestión de las No Conformidades.
Responsable	Especialista de calidad.
Participantes	Especialista de calidad.

I- Resultados a obtener

Que el Jefe de prueba sepa que tiene que comenzar la revisión de un software y tenga toda la documentación necesaria para ello.

II- Descripción del procedimiento

Después de haberse decidido empezar la revisión del software por la dirección de calidad y los Especialistas de calidad, este selecciona un Jefe de prueba y le comunica al mismo que tiene que comenzar la revisión de un software. El Especialista de calidad le entrega al Jefe de prueba toda la documentación que hace falta para la revisión del software en cuestión, entre los cuales, los más importantes se encuentran:

- Manual de usuario
- Manual de instalación
- Descripción de los caso de uso
- Requisitos funcionales

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.02
Actividad	Repartir trabajo entre los probadores.
Proceso	Gestión de las No Conformidades.
Responsable	Jefe de prueba.
Participantes	Jefe de prueba.

I- Resultados a obtener

Que quede repartido el trabajo entre los Probadores.

II- Descripción del procedimiento

Una vez que el Jefe de prueba tenga toda la documentación que se necesita para la revisión del software, este comienza con la distribución del trabajo entre los probadores. Se comenzará a repartir los casos de uso entre los probadores en dependencia de la complejidad (Alta, Media y Baja) del caso de uso y de la cantidad de turnos de trabajo con que se cuente, de la misma forma se repartirá los capítulos del manual de usuario. Así queda elaborado un informe con la asignación del trabajo de los probadores.

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.03
Actividad	Confección de los diseños de caso de prueba.
Proceso	Gestión de las No Conformidades.
Responsable	Probador.
Participantes	Probador.

I- Resultados a obtener

Que quede elaborado los diseños de caso de prueba por cada caso de uso.

II- Descripción del procedimiento

Cuando el probador sepa los casos de uso que tiene que revisar, este comienza la confección del diseño de caso de prueba que ayudará, para encontrar las no conformidades al software. Se confeccionará un diseño de caso de prueba por cada caso de uso y que el mismo tenga las especificaciones, estructura y formato predeterminados en el documento “Plantilla Diseño de CP” aprobado por la Dirección de Calidad de la UCI.

III- Documentos normativos de referencia.

Código	Denominación	Emisor
C-1.1	Plantilla Diseño de CP	Dirección de calidad

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.04
Actividad	Revisión del software.
Proceso	Gestión de las No Conformidades.
Responsable	Jefe de prueba.
Participantes	Probador y Jefe de prueba.

I- Resultados a obtener

Que se encuentren las no conformidades del software y se elabore un informe con las mismas.

II- Descripción del procedimiento

El Probador con el diseño de caso de prueba terminado procede a la revisión del software. En la revisión del software se irán encontrando no conformidades, con las cuales se confeccionará un documento de las no conformidades encontradas al software, este documento debe tener las especificaciones, estructura y formato predeterminados en el documento "Plantilla No Conformidades" aprobado por la Dirección de Calidad de la UCI. Una vez que el Probador termine la revisión el Jefe de prueba revisará las no conformidades encontradas para dar culminación a esta iteración de la revisión del software.

III- Documentos normativos de referencia.

Código	Denominación	Emisor
C-1.2	Plantilla No Conformidades	Dirección de Calidad

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades

Código	01.01.05
Actividad	Comunicar a los desarrolladores la culminación de una iteración de la revisión del software.
Proceso	Gestión de las No Conformidades.
Responsable	Jefe de prueba.
Participantes	Jefe de prueba.

I- Resultados a obtener

Que los Desarrolladores sepan que se terminó la revisión de una iteración del software y tengan el documento con las no conformidades encontradas.

II- Descripción del procedimiento

El Jefe de prueba le comunica al Especialista de calidad que se terminó una iteración de la revisión del software y al mismo tiempo se lo comunica a los Desarrolladores, entregándoles las no conformidades encontradas de esta revisión en un informe de no conformidades.

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.06
Actividad	Responder las no conformidades.
Proceso	Gestión de las No Conformidades.
Responsable	Desarrollador.
Participantes	Desarrollador.

I- Resultados a obtener

Que los Desarrolladores elaboren un informe con las respuestas a las no conformidades.

II- Descripción del procedimiento

Los Desarrolladores leen las no conformidades encontradas por los Probadores y comienzan a elaborar en documento dándole respuestas a estas no conformidades. El documento con las respuestas a las no conformidades debe tener las especificaciones, estructura y formato predeterminados en el documento “Plantilla Respuesta a No Conformidades” aprobado por la Dirección de Calidad de la UCI. Una vez que terminan de darle repuestas a las no conformidades le entregan este documento al Jefe de prueba que esté al frente de la revisión del software.

III- Documentos normativos de referencia.		
Código	Denominación	Emisor
C-1.3	Plantilla Respuesta a No Conformidades	Dirección de Calidad

Fecha de actualización	19/05/2007
------------------------	------------

Procedimiento de ejecución de las actividades	
Código	01.01.07
Actividad	Comunicar la culminación de la revisión del software al solicitante.
Proceso	Gestión de las No Conformidades.
Responsable	Especialista de calidad.
Participantes	Especialista de calidad y Jefe de prueba.

I- Resultados a obtener

Que el solicitante sepa que se terminó la revisión del software.

II- Descripción del procedimiento

Cuando el Jefe de prueba le informe al Especialista de calidad que en una iteración de la revisión del software no se encontraron no conformidades, este elabora un informe de culminación de la revisión del software y se lo informa el solicitante, dando por terminado la revisión del software.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de contenidos de trabajo por áreas
--

Código	A01
Área	Dirección de Calidad
Jerarquía	
Subordinada a:	Vicerrectoría primera de la UCI.

I- Resultados por los que responde

- Selección del Jefe de prueba que realizará la revisión al software.
- Comunicar al solicitante que se termino la revisión al software.

IV- Relación de actividades que ejecuta

Código	Actividad
01.01	Comunicar la revisión del software.
01.07	Comunicar la culminación de la revisión del software al solicitante.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de contenidos de trabajo por áreas

Código	A02
Área	Laboratorio de Calidad de las facultades.
Jerarquía	
Subordinada a:	Vice-decanato de producción de las facultades.

I- Resultados por los que responde

- Asignación de trabajo a los probadores.

- Realización de los diseños de caso de prueba.
- Revisión del software.

Informar de la culminación de una iteración de la revisión del software y entregar el informe con las no conformidades.

IV- Relación de actividades que ejecuta	
Código	Actividad
01.02	Repartir trabajo entre los probadores.
01.03	Confección de los diseños de caso de prueba.
01.04	Revisión del software.
01.05	Comunicar a los desarrolladores la culminación de una iteración de la revisión del software.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de contenidos de trabajo por áreas	
Código	A03
Área	Laboratorio de producción de las facultades.
Jerarquía	
Subordinada a:	Vice-decanato de producción de las facultades.

I- Resultados por los que responde

Elaborar un informe con las respuestas a las no conformidades.

IV- Relación de actividades que ejecuta

Código	Actividad
01.06	Responder las no conformidades.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de puestos de trabajo	
Código	PT01
Denominación	Especialista de calidad
Área	A01(Dirección de Calidad)
Categoría	

I- Resultados por los que responde

- Selección del Jefe de prueba que realizará la revisión al software.
- Comunicar al solicitante que se termino la revisión al software.

IV- Relación de actividades que ejecuta	
Código	Actividad
01.01	Comunicar la revisión del software.
01.07	Comunicar la culminación de la revisión del software al solicitante.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de puestos de trabajo

Código	PT02
Denominación	Jefe de prueba
Área	A02(Laboratorio de Calidad de las facultades)
Categoría	

I- Resultados por los que responde

- Asignación de trabajo a los probadores.
- Revisión del software.
- Informar de la culminación de una iteración de la revisión del software y entregar el informe con las no conformidades.

IV- Relación de actividades que ejecuta

Código	Actividad
01.02	Repartir trabajo entre los probadores.
01.04	Revisión del software.
01.05	Comunicar a los desarrolladores la culminación de una iteración de la revisión del software.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de puestos de trabajo	
Código	PT03
Denominación	Probador
Área	A02(Laboratorio de Calidad de las facultades)

Categoría	
-----------	--

I- Resultados por los que responde

- Realización de los diseños de caso de prueba.
- Revisión del software.

IV- Relación de actividades que ejecuta

Código	Actividad
01.03	Confección de los diseños de caso de prueba.
01.04	Revisión del software.

Fecha de actualización	19/05/2007
------------------------	------------

Descripción de puestos de trabajo	
-----------------------------------	--

Código	PT04
Denominación	Desarrollador
Área	A03(Laboratorio de producción de las facultades)
Categoría	

I- Resultados por los que responde

Elaborar un informe con las respuestas a las no conformidades.

IV- Relación de actividades que ejecuta

Código	Actividad
01.06	Responder las no conformidades.

Fecha de actualización	19/05/2007
------------------------	------------

1.5 Aplicaciones desarrolladas para la Gestión de No Conformidades. Limitaciones.

Existen varias herramientas que gestionan las No Conformidades en todo el mundo utilizándose en otros Laboratorios de Calidad, pero cada una de estos hace su gestión de forma peculiar.

1.5.1 SIGNO - Sistema para la Gestión de No Conformidades.

Este sistema es desarrollado por PROFITek S.A. en Colombia en el año 2005 y es la versión en java 2.1. PROFITek S.A. Ha desarrollado una valiosa herramienta informática de gestión que permite hacer el registro y la gestión total de las no-conformidades, tanto internas como externas, para un sistema de gestión de la calidad ISO 9000:2000.

SIGNO permite registrar toda la información de una no-conformidad, asignar responsables (automática o manualmente), reportar la respuesta, reportar el problema y sus causas, definir el plan de acción (acciones correctivas/preventivas), visualizar mediante semáforos la posible demora en ejecutar acciones, reportar la ejecución de acciones, reportar la auditoría realizada, cerrar no-conformidades y realizar consultas específicas por diferentes criterios. Adicionalmente permite todo el manejo de usuarios y motivos para hacer una fácil adaptación a las necesidades de su organización. [4]

Limitaciones: El costo, que va más allá de nuestras posibilidades, no es configurable a la hora de decidir que pruebas hacer y que elementos peculiares de estas definir. Las pruebas realizadas en la Universidad son peculiares a la hora de hacer alguna revisión en dependencia del cronograma de trabajo y las necesidades del cliente y estas no estarían definidas ni configuradas en este Software. Con nuestra herramienta este problema de configuración estaría al alcance de todos y las No Conformidades detectadas serían guardadas para darle un seguimiento y posteriormente crear un Sistema Inteligente que genere los diseños de casos de pruebas automáticamente y detecte posibles No Conformidades en dependencia del tipo de producto a probar.

1.6 Tendencias y tecnologías actuales. Metodología utilizada.

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

1.6.1 Metodología Utilizada

La Programación Extrema (XP) es una metodología ligera de software que se basa en la simplicidad, la comunicación, realimentación o reutilización del código desarrollado. Es una de las metodologías de desarrollo de software utilizada para proyectos de corto plazo.

El Proceso Unificado (RUP - Rational Unified Process) es un marco genérico de trabajo que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Utiliza UML para modelar todas las especificaciones de un sistema de software. Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. RUP tiene como objetivo: Asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. El Proceso Unificado está dirigido por casos de uso, es centrado en la arquitectura y es iterativo e incremental. [5]

Se ha decidido utilizar RUP porque es para proyectos complejos y de larga duración como el que se propone.

1.6.2 Visual Paradigm.

Se escoge Visual Paradigm, herramienta CASE (Computer-Aided Systems Engineering) que utiliza “UML” como lenguaje de modelaje. Esta herramienta es multiplataforma, soporta los últimos estándares de anotaciones de JAVA y UML, provee soporte para la generación de código PHP y para la realización de ingeniería inversa para Java, reduciendo el tiempo de desarrollo de software. Además, Visual Paradigm -

UML se integra con Eclipse, Borland, JBuilder, Oracle y otras para soportar las fases de implementación en el desarrollo de software.

Rasgos:

- Soporte de Planificación de Caso de Uso.
- Soporte de Análisis Textual.
- Modelo de apoyo entre los diagramas.
- Soporte para la importación de proyecto desde Rational Rose.
- Generación de reportes HTML.
- Generación de reportes PDF. [6]

Se decidió utilizar el Visual Paradigm porque es una herramienta CASE de software libre, de fácil integración con PHP que es el lenguaje que se utilizará para desarrollar la aplicación Web.

1.6.3 Dreamweaver

Dreamweaver es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Java script para crear efectos e interactividades.
- Inserción de archivos multimedia.

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas. Es compatible con las principales tecnologías de servidor como, por ejemplo, ColdFusion, PHP, ASP, ASP.NET y JSP, para que los desarrolladores, más o menos expertos, puedan dar vida a sus diseños. [7]

1.6.4 ¿ASP o PHP?

Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). ASP realiza muchas tareas apoyándose en objetos que deben ser comprados a ciertas empresas especializadas, o en algunos casos programados por el servidor. La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante.

PHP es un lenguaje del lado del servidor, que se utiliza básicamente para generar páginas dinámicas y que facilita de forma sencilla el acceso a diferentes bases de datos. También se utiliza para generar imágenes, generar pdf. PHP lo desarrollan programadores Web introduciendo nuevas funciones en las versiones mejoradas del lenguaje.

PHP tiene muchas ventajas algunas de ellas son:

- Libre y Abierto (Código fuente disponible, es gratuito)
- Multiplataforma: inicialmente fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo, pero es perfectamente compatible con Windows. ASP, sólo está orientado a sistemas Windows
- Soporte para varios servidores Web
- Fácil acceso a Bases de Datos
- Mucha documentación (Ejemplos, manuales)
- Presenta una integración perfecta entre Apache-PHP-MySQL
- Posee una sintaxis bastante clara
- Fácil aprendizaje
- Seguro
- Además PHP está orientado a objetos [8]

Como ASP es un lenguaje que ha sido comercializado con IIS decidimos utilizar PHP porque presenta una compatibilidad total con Apache que es el servidor que vamos a utilizar.

1.6.5 ¿Por qué MySQL?

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

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Características Distintivas:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Clúster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Tipos de compilación del servidor

- Estándar: Los binarios estándares de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.

- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento. [9]

1.6.6 Java

La tecnología JAVA, una tecnología madura, extremadamente eficaz y sorprendentemente versátil, se ha convertido en un recurso inestimable ya que permite a los desarrolladores:

- Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma
- Crear programas para que funcionen en un navegador Web y en servicios Web
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios basados en la tecnología JAVA para crear servicios o aplicaciones totalmente personalizados
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier dispositivo digital.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores con diferentes sistemas operativos. Con Java podemos programar páginas Web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que deseemos desarrollar con acceso a través Web se puede hacer utilizando JAVA. [10]

1.6.7 ¿Servidor Apache?

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- Módulos Base: Módulo con las funciones básicas del Apache.
- Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTP de 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Características:

- Funciona sobre muchas plataformas (muchos sabores de Unix, Linux, Vms, Win32, OS2)
- Módulos cargados dinámicamente.
- CGI, Perl (ejemplo: Formularios, diccionarios en línea, entre otros) Ejemplo CGI Ejemplo Perl.
- Php3 + Bases de datos.
- SSL: transacciones seguras.
- Alto desempeño. [11]

1.6.8 WAMP 5

W.A.M.P. son las siglas de Windows + Apache + MySQL + PHP, es decir, un instalador paso a paso que incluye en un solo paquete estas tres tecnologías para montar tu propio servidor bajo Windows.

El software que se instala con WAMP5 contiene los siguientes servidores y programas:

Apache 1.3.31. El servidor de páginas web más extendido del mercado. Aunque la última versión de este servidor es Apache 2, se instala una versión anterior que resulta más estable. Existe un Add-on que permite sustituir la versión 1.3.31 de Apache por la última versión.

- PHP5. El motor renovado del lenguaje.
- MySQL. La base de datos más extendida para utilizar con PHP.
- PHPmyadmin. Un software que permite administrar una base de datos a través de una interfaz web.
- SQLitemanager. Un sistema para administrar una base de datos a partir de sentencias SQL.

[12]

1.6.9 Arquitectura Cliente-Servidor

Con la proliferación de ordenadores personales de bajo coste en el mercado, los recursos de sistemas de información existentes en cualquier organización se pueden distribuir entre ordenadores de diferentes tipos: ordenadores personales de gama baja, media y alta, estaciones de trabajo, miniordenadores o incluso grandes ordenadores.

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspasa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar

información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.
- Por su parte los servidores realizan, entre otras, las siguientes funciones:
- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

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

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas

- Aumento de la productividad:
 - Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos.
 - Mediante la integración de las aplicaciones cliente/servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
 - Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.

- Menores costes de operación:
 - Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes permite un mejor rendimiento del conjunto.
 - Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
 - El movimiento de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costes de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.

- Mejora en el rendimiento de la red:
 - Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.

- Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones.
- En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.
- Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieran aislados. [13]

Conclusiones del capítulo.

En este capítulo se demuestra la necesidad de crear una herramienta informática que permita gestionar las No Conformidades en un Laboratorio de Calidad, así como dar seguimiento a estos proceso clave durante las pruebas a un producto terminado en un Laboratorio de Calidad. Guardar en una potente Base de Datos todos los elementos relacionados con dichas pruebas para posteriormente crear un Sistema Inteligente que genere los Casos de Pruebas automáticamente y sea capaz de interpretar a partir de los datos guardados la complejidad y correspondencia de las No Conformidades detectadas.

Capítulo 2. Características del sistema.

Introducción.

En este capítulo se refleja un análisis detallado del flujo actual de los procesos, los trabajadores que desarrollan dichos procesos y otras personas involucradas. Se define el objeto de automatización, la información que se maneja y la propuesta del sistema. Se identifican los requisitos funcionales y no funcionales, los casos de uso con sus descripciones, los diagramas de actividades y el modelo de objeto.

2.1 Objeto de Estudio

2.1.1 Situación Problémica.

La Dirección de Calidad de la UCI se ha trazados varios objetivos con el fin de controlar la calidad de los productos de software elaborados en la universidad ya sean con el fin de ser exportados o comercializados en el mercado nacional. Así como formar estudiantes con el perfil de calidad de software. La actividad de la gestión de las No Conformidades comienza en el Laboratorio de Calidad cuando el asesor asigna el trabajo al grupo de probadores que van a realizar las pruebas a determinado producto de software. El probador comienza a revisar y a medida que va encontrando errores o dificultades ya sea en la documentación o la aplicación va insertando en un documento estas No Conformidades para luego ser procesadas. Cuando el probador termina de insertar todas las No Conformidades el asesor envía estas No Conformidades al grupo de desarrollo, estos revisan y arreglan y envían las respuestas de las No Conformidades al grupo de prueba.

Este proceso de gestión de las No Conformidades se hace de forma manual no hay control de la documentación ni seguridad para la misma, los responsables de las actividades no siempre son los que la realizan y no hay una comunicación fluida con el equipo de desarrollo. Todo esto conlleva a un mal funcionamiento del grupo de calidad, mal uso de la documentación y pérdida de información por lo que no se hace una buena gestión de las No Conformidades.

2.1.2 Objeto de automatización.

Laboratorio de calidad

1. Control de la gestión de las No Conformidades encontradas al software.
2. Control del cumplimiento de los requisitos.
3. Control de los reportes de las No Conformidades.
4. Control de los reportes de la cantidad de No Conformidades encontradas.
5. Control de los diseños de caso de prueba.
6. Control de la información.

Equipo de desarrollo

7. Control de la gestión de las respuestas de las No Conformidades.
8. Control de los reportes de las respuestas de las No Conformidades.

2.1.3 Información que se maneja.

1. Especificación de Casos de Uso.
 - Listado de casos de usos involucrados en el proyecto con sus requisitos asociados.
2. Manuales de Instalación.
 - Información por pasos de la instalación.
3. Manuales de Usuarios.
 - Información detallada de las interfaces del software.
4. Documento de Integración
 - Información de integración de los módulos de un proyecto.
5. Documento de Respuesta a las No Conformidades.
 - Respuesta a las No Conformidades del grupo de desarrollo.

2.1.4 Propuesta de sistema.

La herramienta que se propone desarrollar **SoftlaQ** automatizará 3 de los procesos que se llevan a cabo en el Laboratorio Central de Calidad de Software de la UCI. Para una mayor eficiencia en la implementación se organizó en tres módulos fundamentales.

- El módulo de planificación:

El principal objetivo de este módulo es la planificación y estimación de los recursos, para poder realizar esta actividad ya anteriormente en el sistema deben ser insertadas, las pruebas y sus métricas correspondientes. Estas planificaciones serán adicionadas en el sistema, y se podrán acceder a ellas mediante reportes que serán solicitados por personas interesadas. Estos reportes además de exportarlos e imprimirlos se pueden gestionar o sea, buscar, mostrar, seleccionar, modificar y eliminar. Las pruebas y las métricas también podrán ser gestionadas.

- El módulo de No Conformidades:

En este módulo se gestionarán todas las No Conformidades encontradas durante el proceso de revisión, para realizar esta actividad se debe tener previamente toda la documentación necesaria en dependencia del tipo de prueba que se realizará y las condiciones requeridas para el proceso de prueba. Las No Conformidades serán insertadas en el sistema y estas se podrán modificar, eliminar y se le darán respuesta por parte del equipo de desarrollo. Además el sistema permitirá exportar e imprimir los diferentes reportes de No Conformidades, asignar el trabajo a los probadores y se visualizará el estado de los elementos de prueba, los módulos y los proyectos y a la vez permite dar un reporte de toda la información relacionada con este proceso.

- El módulo de seguimiento.

El módulo de seguimiento es el encargado de administrar todos los recursos utilizados para la realización de las diferentes pruebas, permitiendo conocer el estado real de estos y a la vez brindar un conjunto de reportes que posibilitan mantener actualizados a los directivos de la universidad. Como resultado de este proceso se obtendrá un histórico con toda la información relacionada con los diferentes tipos de recursos.

Esta herramienta se construirá sobre la tecnología PHP, utilizando las amplias ventajas que el mismo ofrece y que fueron descritas en el capítulo 1, además utilizará como gestor de base de datos el MySQL por ser compatible con el lenguaje seleccionado y brindar diferentes funcionalidades descritas también en el capítulo 1.

El sistema constará de políticas de seguridad otorgando a cada usuario los derechos que le corresponden. Existirán 5 tipos de usuarios: administradores, especialista de calidad, jefes de prueba, probadores y desarrolladores. Cada usuario que entra al sistema debe autenticarse antes de realizar alguna acción.

2.2 Modelo de negocio

El modelo de negocio que se va a describir responde a un conjunto de actividades realizadas dentro de los procesos que se desarrollan en un laboratorio de calidad de software en la UCI. Inicialmente el solicitante (vicedecanos de producción, asesor de calidad de cada facultad) realiza la solicitud de revisión de sus productos a la dirección de calidad de la UCI. Posteriormente se procede a la revisión del software que incluye tres actividades fundamentales, la revisión de documentación, planificación de las pruebas y ejecución de pruebas.

En la revisión de la documentación se verifica que la documentación del proyecto cumpla con los lineamientos básicos que exige la dirección de calidad, de no cumplir se le informa la negación de la revisión del software, en caso contrario se comienza con la planificación y estimación de las pruebas, luego se realiza la ejecución de las mismas incluyendo la realización de reportes de las No Conformidades obtenidas. Paralelamente a esto se efectúa la administración de los recursos.

Los directivos de la universidad (rector, vicerrectores, directivos de calidad, decanos, vicedecanos) pueden solicitar reportes en dependencia de su interés.

2.2.1 Actores del negocio.

Actores del negocio	Justificación
Solicitante	Asesor de calidad de la facultad, vicedecano de producción ó toda aquella persona que solicite la revisión del software.
Directivo	Persona en la UCI como rector, vicedecanos, directivos de calidad, decanos, vicedecanos que estén interesados en

	realizar solicitudes de reportes.
--	-----------------------------------

Tabla 2.1 Descripción de los actores del negocio

2.2.2 Trabajadores del negocio.

Nombre del actor	Descripción
Jefe de prueba	Encargado en cada facultad de controlar los recursos en el laboratorio de calidad. Planifica los recursos del laboratorio, repartir el trabajo, supervisar la gestión de las no conformidades y realiza los reportes de seguimiento de los recursos.
Especialista de calidad	Encargado de insertar las pruebas a realizar y las métricas.
Probador	Persona que realiza las pruebas de software.
Desarrollado	Ve las no conformidades emitidas y las responde.

Tabla 2.2 Descripción de los trabajadores del negocio

2.2.3 Diagrama de Casos de Uso del Negocio.

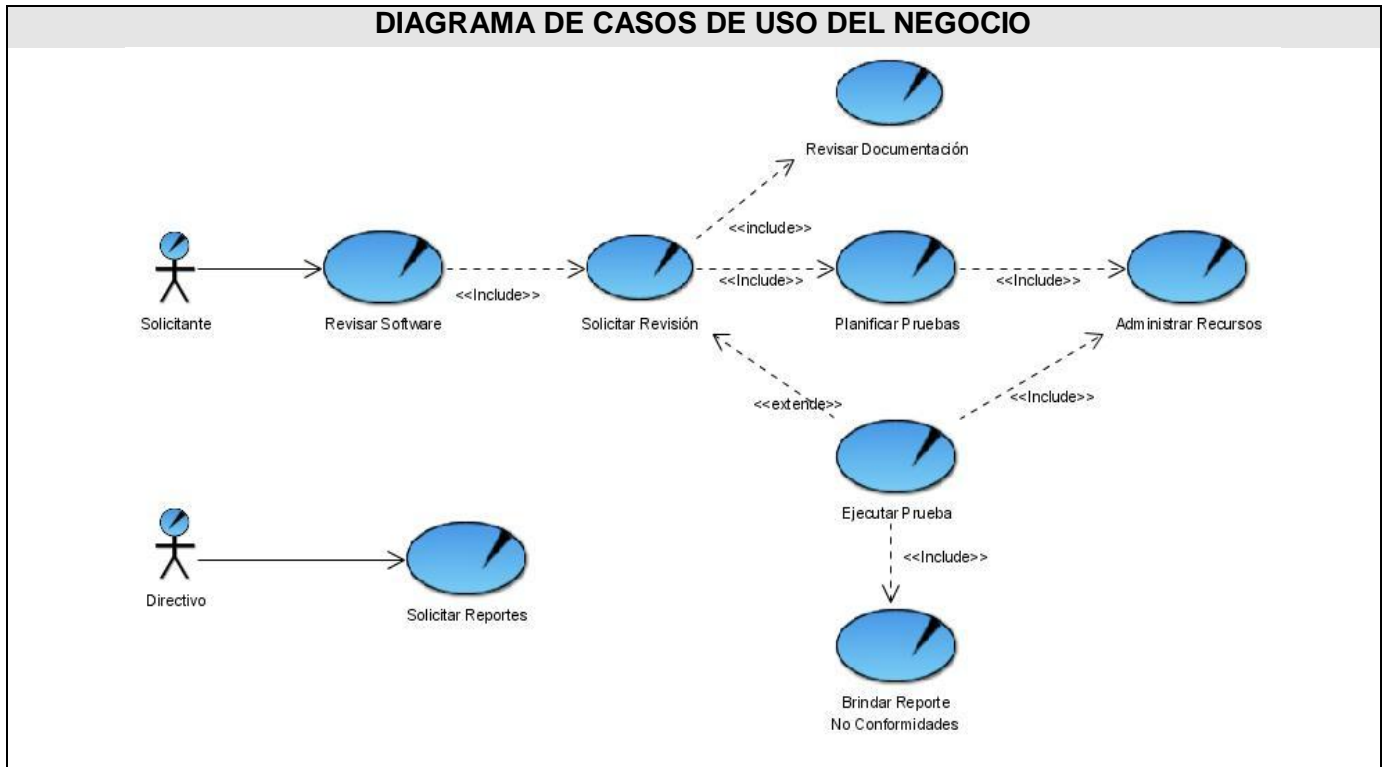


Figura 2.1 Diagrama del Modelo del Negocio.

2.2.4 Casos de Uso del Negocio

- Ejecutar Prueba
- Brindar Reporte No Conformidades.

Ejecutar Prueba: El caso de uso es iniciado cuando el especialista de calidad informa la realización de las pruebas al jefe de prueba, este recibe la documentación del trabajo y asigna a las pruebas el probador, este hace el diseño de caso de prueba y comienza a revisar, si no encuentra No Conformidades informa al jefe de prueba mediante el documento de diseño de caso de prueba que no encontró No Conformidades, el jefe de prueba recibe el aviso y elabora un informe del proyecto revisado y se lo envía al especialista de calidad y este revisa el informe y finaliza el caso de uso. En caso de que el probador encuentre No Conformidades elaborará un informe con las No Conformidades encontradas y se lo envía al jefe de prueba, este recibe el informe y se lo envía al desarrollador, este recibe el informe y elabora un

documento de respuestas a las No Conformidades, se lo envía al jefe de prueba y este lo recibe y asigna el trabajo nuevamente hasta que no se encuentren No Conformidades, esto demuestra que el software esta listo para ser liberado finalizando así el caso de uso.

Brindar reporte de las No Conformidades: El caso de uso es iniciado cuando el Jefe de pruebas selecciona del documento de diseño de caso de uso todas las No Conformidades encontradas y elabora un reporte de las No Conformidades y se lo envía al especialista de calidad.

2.2.5 Diagramas de Actividades

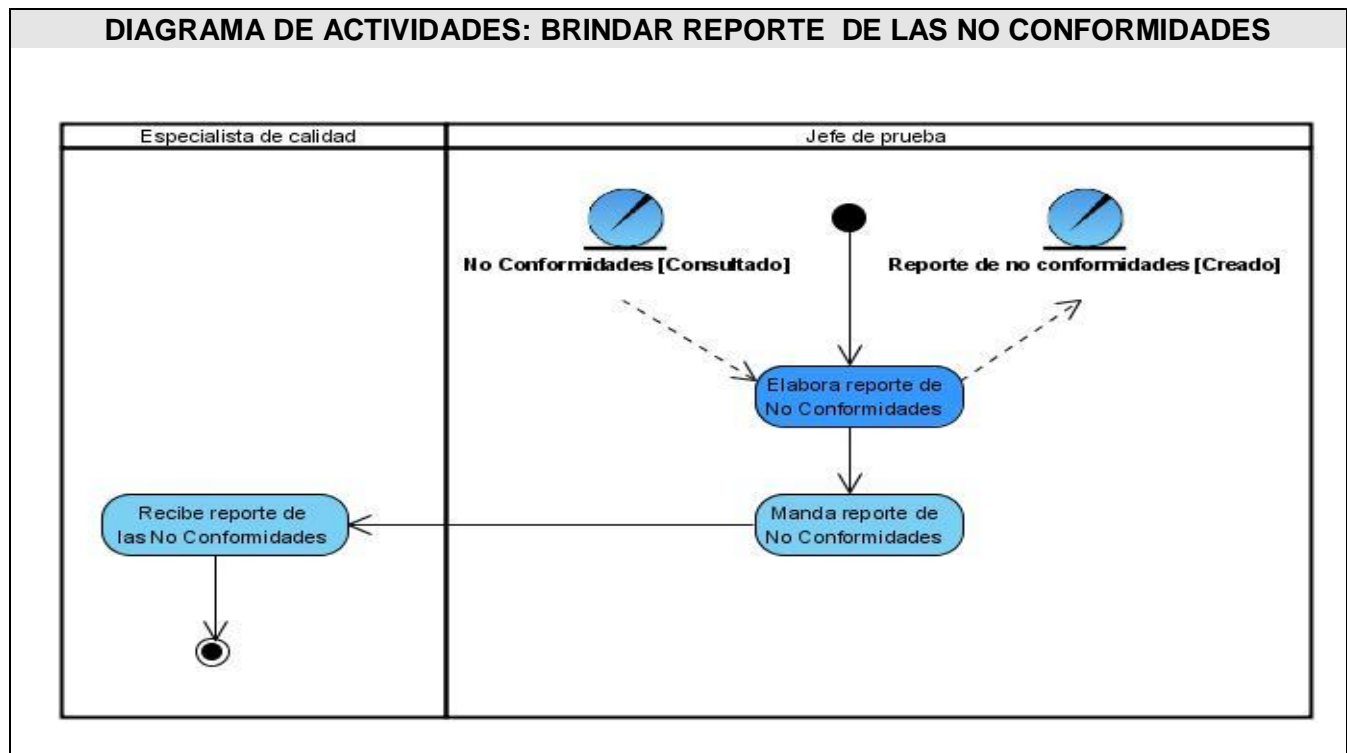


Figura 2.2 Diagrama de actividades: Brindar Reporte de las No Conformidades.

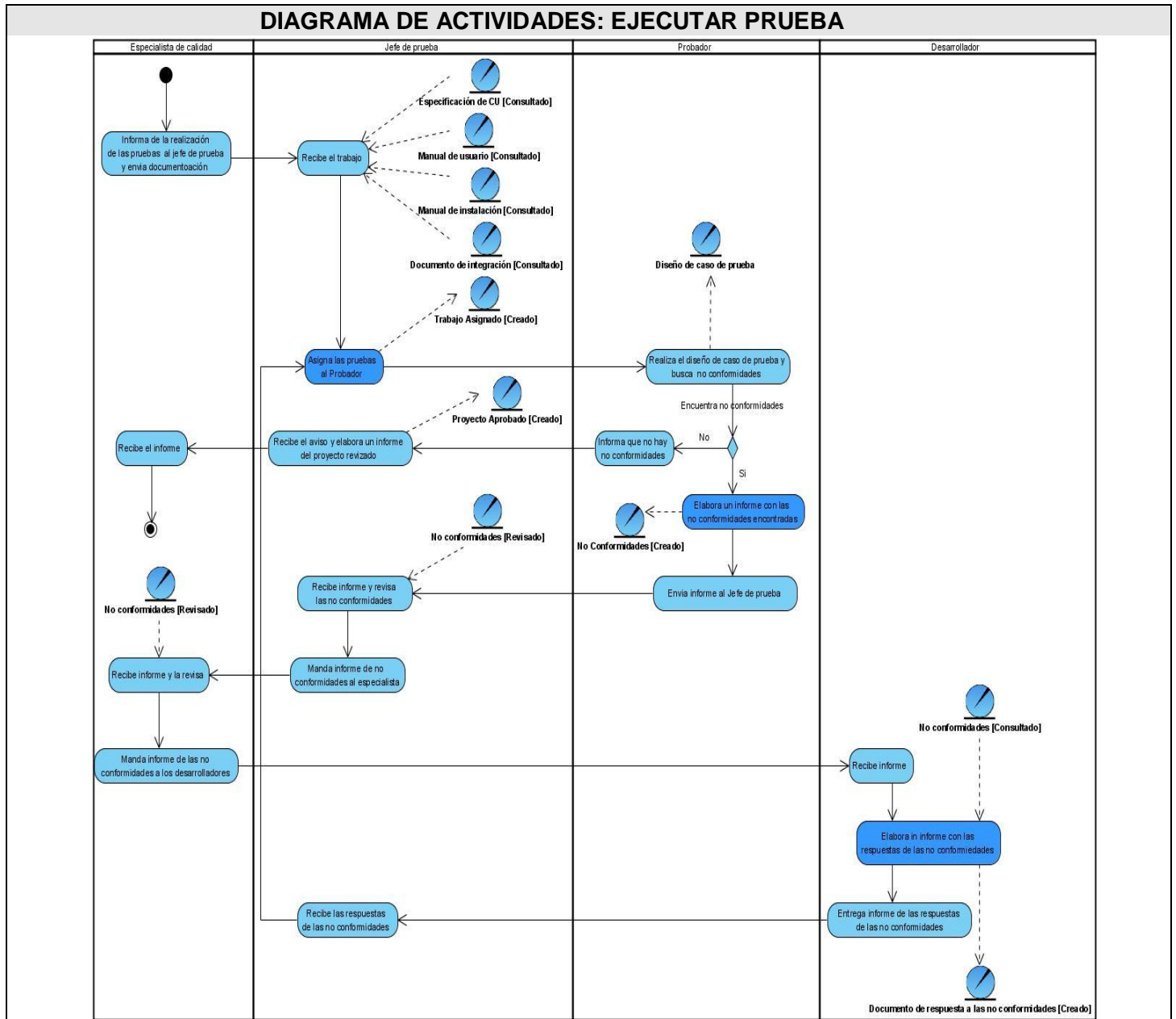


Figura 2.3 Diagrama de actividades: Ejecutar Prueba

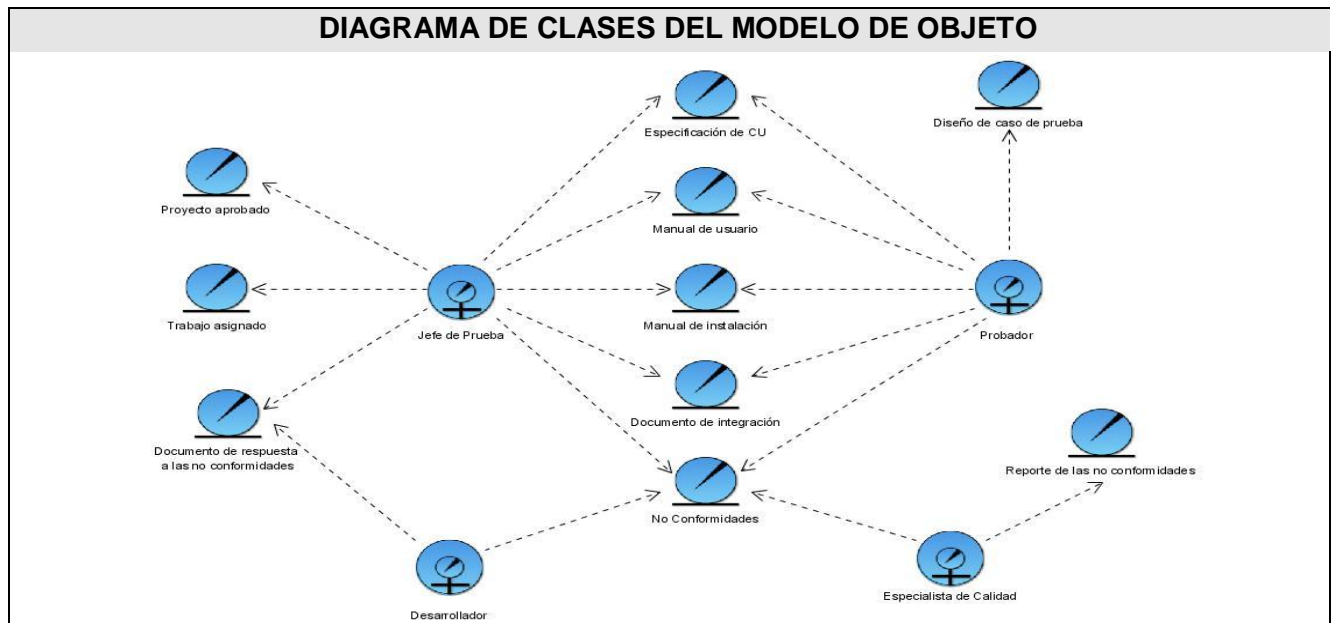


Figura 2.4 Diagrama de clases del modelo de objetos.

2.3 Definición de Requisitos Funcionales

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

RF 1 Buscar tipo de prueba

- 1.1 Buscar Módulo.
 - 1.1.2 Nombre Proyecto.
- 1.2 Mostar Módulo.
 - 1.2.1 Nombre del Proyecto.
 - 1.2.2 Módulo.
 - 1.2.3 Iteración.
- 1.3 Seleccionar Módulo.
- 1.4 Seleccionar un Tipo de prueba.
 - 1.4.1 Tipo de prueba.
- 1.5 Mostrar Tipo de prueba.

- 1.5.1 Nombre del Proyecto.
- 1.5.2 Módulo.
- 1.5.3 Iteración.
- 1.5.4 Tipo de prueba.

RF 2 Buscar elemento de prueba

- 2.1 Buscar elemento de prueba.
 - 2.1.1 Nombre del proyecto.
 - 2.1.2 Módulo.
 - 2.1.3 Iteración.
 - 2.1.4 Tipo de prueba.
- 2.2 Seleccionar elemento de prueba.

RF 3 Gestionar asignación de trabajo

- 3.1 Asignar elemento de prueba al probador.
- 3.2 Mostrar trabajo asignado.
 - 3.2.1 Nombre y Apellidos.
 - 3.2.2 Facultad.
 - 3.2.3 Organización.
 - 3.2.4 Elemento de prueba.
 - 3.2.5 Complejidad.
- 3.3 Eliminar asignación de trabajo.

RF 4 Gestionar Probador

- 4.1 Buscar probador.
 - 4.1.1 Nombre.
 - 4.1.2 Sexo.
 - 4.1.3 Si es estudiante:
 - 4.1.3.1 Facultad.
 - 4.1.4 Si es trabajador:
 - 4.1.4.1 Organización.
- 4.2 Mostrar probador.
 - 4.2.1 Nombre.

- 4.2.2 Apellido.
- 4.2.3 Sexo.
- 4.2.4 Si es estudiante:
 - 4.2.4.1 Facultad.
- 4.2.5 Si es trabajador:
 - 4.2.5.1 Organización.

4.3 Seleccionar probador.

RF 5 Gestionar elemento de prueba

- 5.1 Insertar Elemento de prueba.
 - 5.1.1 Nombre del elemento.
 - 5.1.2 Complejidad.
 - 5.1.3 Clasificación.
- 5.2 Mostrar Elemento de prueba.
 - 5.2.1 Nombre del elemento.
 - 5.2.2 Complejidad.
 - 5.2.3 Clasificación.
- 5.3 Eliminar un elemento de prueba.

RF 6 Mostrar datos de los elementos de prueba

- 6.1 Mostrar datos del elemento de prueba.
 - 6.1.1 Nombre.
 - 6.1.2 Complejidad.
 - 6.1.3 Clasificación.
- 6.2 Mostrar fecha de culminación de un elemento de prueba, si se terminó.
- 6.3 Mostrar Probador asignado al elemento de prueba.
 - 6.3.1 Nombre.
 - 6.3.2 Si es estudiante:
 - 6.3.2.1 Facultad.
 - 6.3.3 Si es trabajador:
 - 6.3.3.1 Organización.

RF 7 Gestionar No Conformidades

- 7.1 Insertar no conformidad.
 - 7.1.1 Elemento.
 - 7.1.2 Número de la no conformidad.
 - 7.1.3 No Conformidad.
 - 7.1.4 Aspecto correspondiente.
 - 7.1.5 Etapa de detección.
 - 7.1.6 Clasificación
- 7.2 Mostrar las No Conformidades
 - 7.2.1 Elemento.
 - 7.2.2 Número de la no conformidad.
 - 7.2.3 No Conformidad.
 - 7.2.4 Aspecto correspondiente.
 - 7.2.5 Etapa de detección.
 - 7.2.6 Clasificación.
- 7.3 Eliminar no conformidad.
- 7.4 Modificar no conformidad.
 - 7.4.1 Elemento.
 - 7.4.2 Número de la no conformidad.
 - 7.4.3 No Conformidad.
 - 7.4.4 Aspecto correspondiente.
 - 7.4.5 Etapa de detección.
 - 7.4.6 Clasificación.
- 7.5 Asignar fecha de culminación de un elemento de pruebas.

RF 8 Gestionar las respuestas de las No Conformidades

- 8.1 Buscar no conformidades
 - 8.1.1 Proyecto.
 - 8.1.2 Módulo.
 - 8.1.3 Iteración.
 - 8.1.4 Tipo de prueba.
- 8.2 Mostrar las no conformidades.

8.2.2 Elemento de prueba.

8.2.2 Número.

8.2.3 No Conformidad

8.2.4 Aspecto correspondiente

8.2.5 Etapa de detección.

8.2.6 Clasificación

8.3 Insertar la respuesta de las No Conformidades.

8.4 Mostrar la respuesta a las No Conformidades.

RF 9 Gestionar Interrupción del trabajo

9.1 Insertar tiempo de interrupción.

RF 10 Gestionar estados.

10.1 Buscar estado Módulo y elemento de prueba.

10.1.1 Proyecto.

10.1.2 Módulo.

10.1.3 Iteración.

10.1.4 Tipo de prueba.

10.2 Mostrar estado de la Módulo.

10.3 Mostrar estado de los elementos de prueba.

RF 11 Gestionar Reporte

11.1 Exportar reporte.

11.1.1 Se exporta en la parte de la PC que se desee en formato pdf.

11.2 Imprimir reporte.

RF 12 Gestionar la visualización de las No Conformidades

12.1 Buscar Módulo de un Proyecto.

12.1.1 Nombre Proyecto.

12.2 Mostar Módulo de un Proyecto.

12.2.1 Nombre del Proyecto.

12.2.2 Módulo.

12.2.3 Iteración.

12.3 Seleccionar Módulo de un Proyecto.

- 12.4 Seleccionar Tipo de prueba.
- 12.5 Buscar Tipo de prueba.
- 12.6 Mostrar Tipo de prueba.
 - 12.6.1 Nombre del Proyecto.
 - 12.6.2 Módulo.
 - 12.6.3 Iteración.
 - 12.6.4 Tipo de prueba.
- 12.7 Seleccionar Elemento de prueba.
 - 12.7.1 Mostrar las no conformidades.
 - 12.7.2 Número.
 - 12.7.3 Elemento.
 - 12.7.4 No Conformidad.
 - 12.7.5 Aspecto correspondiente.
 - 12.7.6 Etapa de detección.
 - 12.7.7 Clasificación.

RF 13 Gestionar reporte general

- 13.1 Mostrar reportes Cantidad de No Conformidades por prueba en un determinado proyecto, Módulo, iteración y tipo de prueba.
 - 13.1.1 Proyecto
 - 13.1.2 Módulo
 - 13.1.3 Iteración
 - 13.1.4 Tipo de prueba
 - 13.1.5 Elemento de prueba
 - 13.1.6 Número de no conformidad
 - 13.1.7 Complejidad del elemento de prueba
 - 13.1.8 Fecha de revisión
- 13.2 Las No Conformidades encontradas en un Proyecto, Módulo, Iteración, Tipo de prueba y elemento de prueba.
 - 13.2.1 Proyecto.
 - 13.2.2 Módulo.

- 13.2.3 Iteración.
- 13.2.4 Tipo de prueba.
- 13.2.5 Elemento de prueba.
- 13.2.6 Número.
- 13.2.7 Elemento.
- 13.2.8 No Conformidad.
- 13.2.9 Aspecto correspondiente.
- 13.2.10 Etapa de detección.
- 13.2.11 Clasificación.

RF 14 Gestionar tiempo de realización de un elemento de prueba

- 14.1 Asignar tiempo de inicio y fin de un elemento de prueba.

RF 15 Gestionar usuario

- 15.1 Crear usuario del sistema.
 - 15.1.1 Usuario
 - 15.1.2 Contraseña
 - 15.1.3 Rol.
- 15.2 Eliminar usuario del sistema.
- 15.3 Editar Datos del usuario.
 - 15.3.1 Usuario
 - 15.3.2 Contraseña
- 15.4 Asignar a los usuarios los roles:
 - 15.4.1 Especialista
 - 15.4.1.1 Mostrar estados de las pruebas
 - 15.4.1.2 Gestionar las No Conformidades
 - 15.4.1.3 Mostrar datos de elemento de prueba
 - 15.4.1.4 Reportes
 - 15.4.1.5 Autenticar Usuario.
 - 15.4.2 Jefe de prueba
 - 15.4.2.1 Asignar trabajo
 - 15.4.2.2 Mostrar estados de las pruebas

- 15.4.2.3 Gestionar las No Conformidades
- 15.4.2.4 Insertar nuevo elemento de prueba
- 15.4.2.5 Mostrar datos de elemento de prueba
- 15.4.2.6 Reportes
- 15.4.2.7 Autenticar Usuario.
- 15.4.3 Probador
 - 15.4.3.1 Gestionar las No Conformidades
 - 15.4.3.2 Mostrar datos de elemento de prueba
 - 15.4.3.3 Reporte de No Conformidades
 - 15.4.3.4 Autenticar Usuario.
- 15.4.4 Desarrollador
 - 15.4.4.1 Insertar respuestas a las No Conformidades
 - 15.4.4.2 Autenticar Usuario.

RF 16 Autenticar Usuario

- 16.1 Autenticar Usuario.
 - 16.1.1 Usuario.
 - 16.1.2 Contraseña.

RF 17 Enviar confirmación de culminación del trabajo

- 17.1 Enviar confirmación de culminación de la revisión de un elemento de prueba de un probador a un jefe de prueba
- 17.2 Enviar confirmación de culminación de la revisión de un modulo de un jefe de prueba a un especialista y a los desarrolladores.

2.4 Definición de requisitos no funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto.

A continuación se muestran los requerimientos no funcionales:

1. Apariencia o interfaz externa.

El sistema deberá poseer una interfaz web sencilla, amigable, lo más atractiva y clara posible para el usuario, además su funcionamiento debe ser fácil comprensión.

2. Usabilidad.

El sistema solo podrá ser utilizado por aquellas personas que de una u otra forma se encuentren relacionadas con el control y aseguramiento de la calidad en los proyectos productivos de la Universidad de las Ciencias Informáticas.

3. Rendimiento.

Debido a que se trata de una aplicación cliente/servidor debe ser eficiente, con capacidad adecuada de procesamiento y cálculo, así como requiere de un tiempo de respuesta relativamente pequeño.

4. Portabilidad.

El sistema debe ser multiplataforma.

5. Seguridad.

Para poder acceder al sistema el usuario deberá estar registrado en la aplicación. Los usuarios serán creados en dependencia del rol que desempeñen. Se debe garantizar que solo posean acceso a la información con derecho a ver o manipular.

6. Software.

En el lado del Cliente debe existir un navegador que soporte Java Script y Adobe Reader 5.

En el lado del Servidor debe estar instalado: PHP 5.0 o superior y gestor de base de datos MySQL 4.1.8 o superior.

7. Hardware.

Para el desarrollo y ejecución de esta aplicación se necesita conexión a la red local, por lo que se requiere tarjeta de red. Además es necesario contar con una impresora para poder imprimir los diferentes tipos de reportes. Pentium IV, 239 GHz, 249 MB de RAM.

2.5 Actores del sistema a automatizar

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. En este caso los actores que interactúan con el sistema se definen a continuación en la tabla.

Nombre del actor	Descripción
Usuario del sistema	Rol que representa a los usuarios del sistema que se han autenticado y pueden acceder a los recursos que le son permitidos.
Administrador	Este rol constituye una generalización del Usuario del sistema. Encargado del mantenimiento del sistema así como de gestionar todo el proceso de permisos a los usuarios que acceden al mismo.
Especialista de calidad.	Este rol constituye una generalización del Comunicador de estados. Rol que recoge a todos los especialistas de calidad. Además puede ver los Reportes generales que brinda la aplicación.
Jefe de prueba	Este rol constituye una generalización del Comunicador de estados. Rol que recoge a todos los encargados de las revisiones. Encargado de repartir el trabajo a los probadores, inserta los Elementos de prueba y emite reportes de No Conformidades.
Comunicador de estados	Este rol constituye una generalización del Gestor de No Conformidades. Encargado de mostrar el estado de los Proyectos, Módulos y todos sus Elementos de prueba.
Probador	Este rol constituye una generalización del Gestor de No Conformidades. Rol que recoge a todos los estudiantes ó trabajadores que revisaran el software. Son los encargados de emitir las No Conformidades

	encontradas.
Gestor de Conformidades	No Este rol constituye una generalización del Usuario del sistema. Encargado de gestionar las No Conformidades, puede ver los datos de los elementos de prueba y gestiona las No Conformidades.
Desarrollador	Este rol constituye una generalización del Usuario del sistema. Rol que recoge a todos los que trabajan en la confección del software. Son los encargados de emitir las respuestas de las No Conformidades.
Reloj	Encargado de recoger el tiempo de inicio y fin de la revisión de un Elemento de prueba.

Tabla 2.3 Descripción de los actores del sistema

2.6 Listado de casos de uso.

CU-1	Insertar elemento de prueba
Actor	Jefe de prueba.
Descripción	En este caso de uso se busca un Proyecto, Módulo, Iteración y Tipo de prueba y se le insertan los Elementos de pruebas relacionados con ellos.
Referencia	RF 1, RF 5

Tabla 2.4 Caso de uso "Insertar elemento de prueba"

CU-2	Buscar elemento de prueba
Actor	Jefe de prueba, Especialista de calidad, Comunicador de estados, Probador, Gestor de No Conformidades y Desarrollador.
Descripción	En este caso de uso se buscan los Elementos de prueba que tiene un Tipo de prueba, Iteración, Módulo y Proyecto.
Referencia	RF 2

Tabla 2.5 Caso de uso "Buscar elemento de prueba"

CU-3	Asignar trabajo
Actor	Jefe de prueba.
Descripción	En este caso de uso de asigna el trabajo a los probadores, se buscan los elementos de prueba y se reparten entre los probadores.
Referencia	RF 3, RF 4

Tabla 2.6 Caso de uso "Asignar trabajo"

CU-4	Mostrar datos de los elementos de prueba
Actor	Jefe de prueba, Especialista de calidad, Comunicador de estados, Probador y Gestor de No Conformidades.
Descripción	En este caso de uso se muestra los datos de los elementos de pruebas, así

	como el probador que lo tiene asignado, su complejidad y si se terminó de revisar la fecha de culminación.
Referencia	RF 6

Tabla 2.7 Caso de uso “Mostrar datos de los elementos de prueba”

CU-5	Gestionar las No Conformidades
Actor	Jefe de prueba, Especialista de calidad, Comunicador de estados, Probador y Gestor de No Conformidades.
Descripción	En este caso de uso de insertan, modifican y eliminan los No Conformidades asociadas a un Proyecto, Módulo, Iteración y Tipo de prueba.
Referencia	RF 7, RF 12

Tabla 2.8 Caso de uso “Gestionar las No Conformidades”

CU-6	Gestionar reportes
Actor	Jefe de prueba, Especialista de calidad, Comunicador de estados, Probador, Gestor de No Conformidades, Visualizador de No Conformidades y Desarrollador.
Descripción	En este caso de uso de puede imprimir y/o exportar los reportes que brinda el sistema.
Referencia	RF 11

Tabla 2.9 Caso de uso “Gestionar reportes”

CU-7	Asignar tiempo de realización de un elemento de prueba
Actor	Reloj.
Descripción	En este caso de uso se recoge el tiempo de inicio y de culminación de la revisión de un elemento de prueba.
Referencia	RF 14

Tabla 2.10 Caso de uso “Asignar tiempo de realización de un elemento de prueba”

CU-8	Insertar interrupción
Actor	Probador.
Descripción	En este caso de uso de inserta el tiempo de interrupción de una revisión de un Elemento de prueba.
Referencia	RF 9

Tabla 2.11 Caso de uso “Insertar interrupción”

CU-9	Insertar respuestas a las No Conformidades
Actor	Desarrollador.
Descripción	En este caso de uso de insertan las respuestas de las No Conformidades las cuales están asociadas a un Proyecto, Módulo, Iteración y Tipo de prueba.
Referencia	RF 8, RF 12

Tabla 2.12 Caso de uso “Insertar respuestas a las No Conformidades2

CU-10	Mostrar estados
Actor	Jefe de prueba, Especialista de calidad y Comunicador de estados.
Descripción	En este caso de uso se muestran los estados en que se encuentran las revisiones de los Proyectos, Módulos y Elementos de prueba.
Referencia	RF 10

Tabla 2.13 Caso de uso “Mostrar estados”

CU-11	Mostrar reportes de las No Conformidades
Actor	Especialista de calidad.
Descripción	Es este caso de uso se muestran los repostes generales de las No Conformidades.
Referencia	RF 13

Tabla 2.14 Caso de uso “Mostrar reportes de las No Conformidades”

CU-12	Gestionar usuario
Actor	Administrador.
Descripción	En este caso de uso se insertan, modifican y eliminan los usuarios del sistema, así como asignarle los roles que existen en la aplicación.
Referencia	RF 15

Tabla 2.15 Caso de uso “Gestionar usuario”

CU-13	Autenticar usuario
Actor	Usuario del sistema.
Descripción	En este caso de uso los usuarios del sistema pueden autenticarse en el sistema.
Referencia	RF 16

Tabla 2.16 Caso de uso “Autenticar usuario”

2.7 Paquetes y sus relaciones

Un sistema grande se debe dividir en unidades más pequeñas, de modo que pueda ser entendido por las personas que necesiten consultarlo. Dado el número de casos de uso y actores, se introducen paquetes en el modelo de casos de uso para tratar su tamaño, organizar los elementos en grupos y hacerlo más comprensibles. Se decidió dividir el sistema en 9 paquetes:

- 1- El paquete Asignar trabajo contiene los casos de usos relacionados con la asignación de trabajo.
- 2- El paquete Gestionar No Conformidades contiene los casos de usos relacionados con la gestión de las No Conformidades.
- 3- El paquete Insertar interrupción contiene los casos de usos relacionados con la interrupción en el trabajo de un probador.

- 4- El paquete Asignar tiempo de trabajo contiene los casos de usos relacionados con la asignación de tiempo de trabajo a un probador.
- 5- El paquete Insertar respuestas a las No Conformidades contiene los casos de usos relacionados con las respuestas de las No Conformidades.
- 6- El paquete Mostrar estados contiene los casos de usos relacionados con mostrar los estados.
- 7- El paquete Mostrar reportes contiene los casos de usos relacionados con los reportes generales.
- 8- El paquete Administrar usuario contienen los casos de usos relacionados con la gestión de los usuarios.
- 9- El paquete Autenticar usuario contiene los casos de usos relacionados con la autenticación de un usuario.

A continuación en la figura se representa el diagrama de paquetes y sus relaciones.

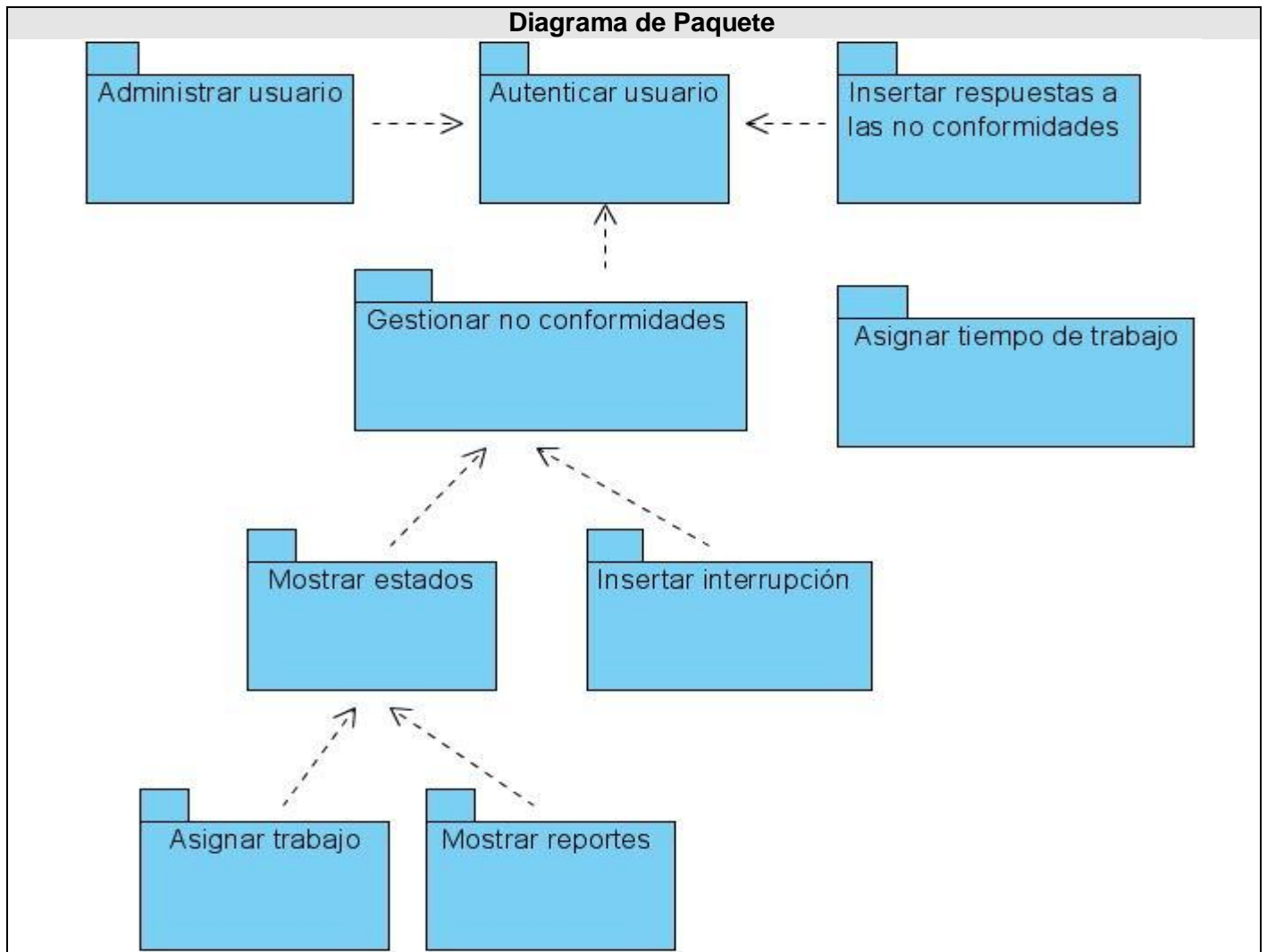


Figura 2.5 Diagrama de Paquetes

A continuación en la figura se representa el diagrama de Actores del sistema.

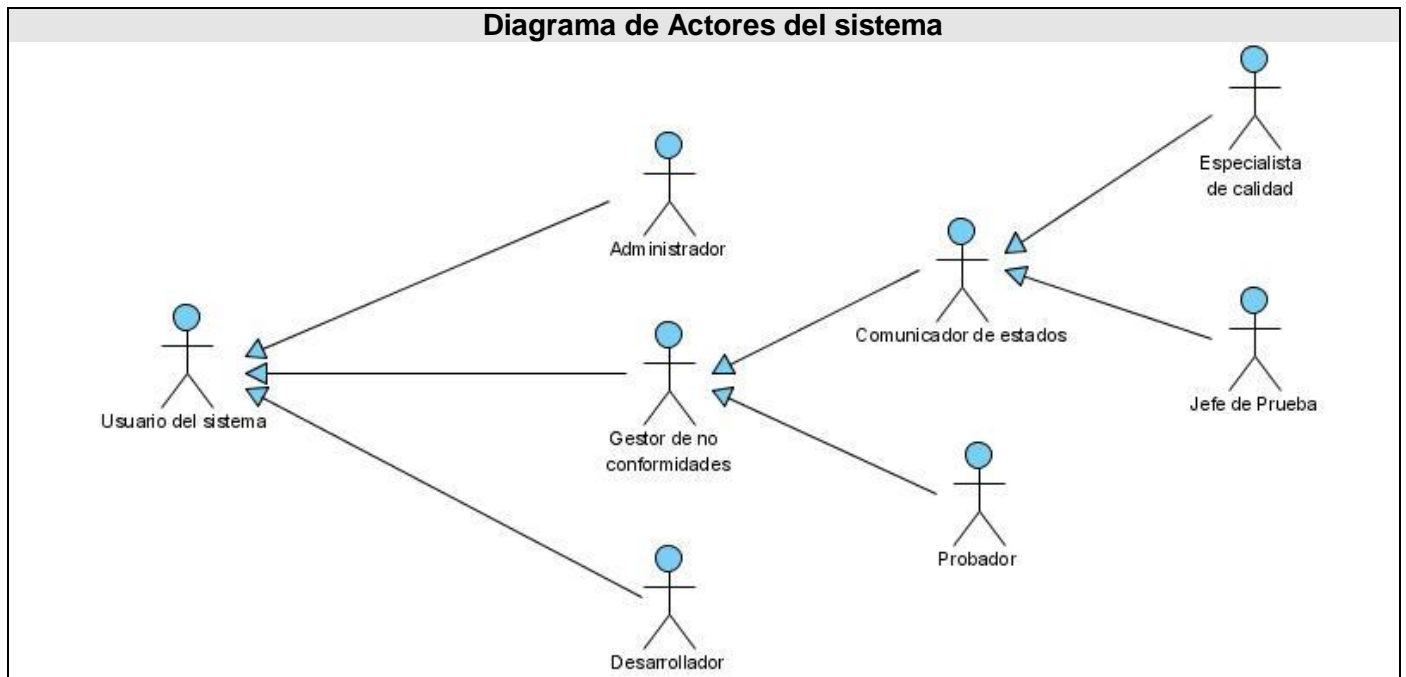


Figura 2.6 Diagrama de Actores del Sistema.

2.8 Diagrama de casos de uso del sistema.

Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia.

2.8.1 El paquete Asignar trabajo.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

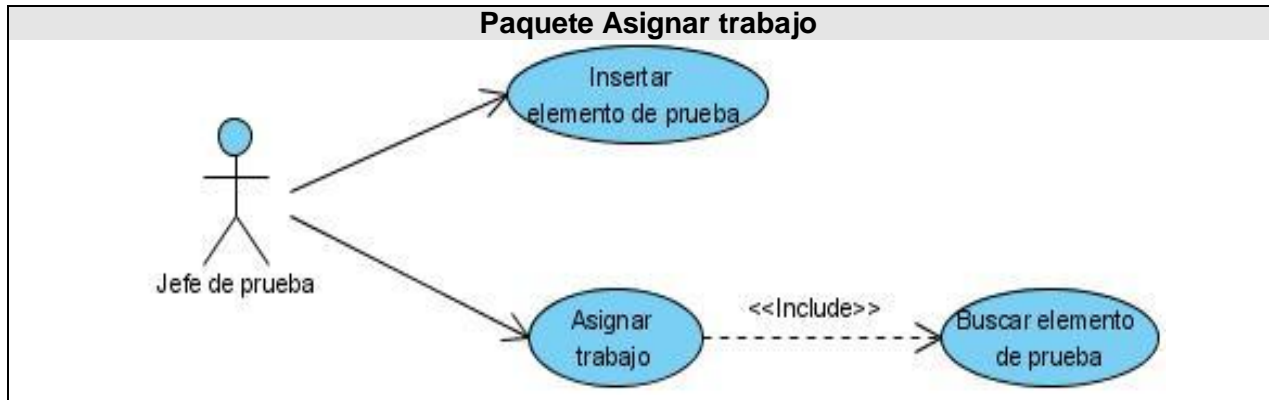


Figura 2.7 Diagrama de paquete “Asignar Trabajo”.

2.8.1.1 Descripción de los casos de uso del Paquete Asignar trabajo.

La descripción de los casos de uso del sistema representa las acciones y reacciones que muestran el comportamiento del sistema desde el punto de vista del usuario, detalla el flujo de trabajo que se produce en la interacción entre actores y casos de uso. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

Caso de uso: Insertar elemento de prueba
Actores: Jefe de prueba
Propósito
Insertar los elementos de prueba a los tipos de pruebas.
Resumen
El caso de uso se inicia cuando el Jefe de prueba desea insertar un elemento de prueba a un Proyecto, Módulo, Iteración y Tipo de prueba. El Jefe de prueba inserta los elemento de prueba con los siguientes datos: Nombre del elemento de prueba, Complejidad y Clasificación.
Precondiciones
Debe haber un proyecto revisándolo.
Referencias: RF 1, RF 5
Casos de uso asociados:
Poscondiciones
Quedan insertados los elementos de prueba a un Tipo de prueba.
Requerimientos especiales:
Prototipo:

Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de prueba desea insertar elementos de prueba.	2. El sistema muestra la interfaz de búsqueda de un Módulo seleccionando un proyecto.
3. El Jefe de prueba selecciona un proyecto y presiona buscar.	4. El sistema muestra los Módulos del proyecto seleccionado con los siguientes datos: <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración Y brinda la posibilidad de seleccionar un tipo de prueba.
5. El Jefe de prueba selecciona el tipo de prueba y presiona buscar.	6. El sistema muestra el Tipo de prueba con los siguientes datos: <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración • Tipo de Prueba
	7. El sistema brinda la posibilidad de insertar un elemento de prueba con los siguientes datos: <ul style="list-style-type: none"> • Nombre del elemento de prueba • Complejidad • Clasificación
8. El Jefe de prueba introduce los datos de los elementos de prueba y presiona buscar.	9. El sistema muestra lo elementos insertados.
	10. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Eliminar un elemento de prueba ver sección Eliminar
	11. El sistema finaliza la ejecución del caso de uso.
Sección : Eliminar	
1. El Jefe de prueba selecciona un elemento de prueba a eliminar.	2. El sistema selecciona el elemento de prueba a eliminar.

3. El Jefe de prueba elimina el elemento de prueba.	4. El sistema muestra un mensaje de confirmación "Desea eliminar el elemento de prueba".
5. Si el Jefe de prueba desea eliminar el elemento de prueba y presiona Aceptar. En caso contrario ver Curso alterno	6. El sistema elimina el elemento de prueba.
Cursos alternos	
5.1 Si el Jefe de prueba no desea eliminar la asignación y presiona Cancelar. Proveniente del flujo básico 1	

Tabla 2.17 Descripción del caso de uso "Insertar elemento de prueba"

Caso de uso: Asignar Trabajo	
Actores: Jefe de prueba	
Propósito	
Asignarle los elementos de prueba a los Probadores.	
Resumen	
El caso de uso comienza cuando el jefe desea asignarle trabajo a los probadores. El Jefe de prueba busca los elementos de prueba según el Proyecto, Módulo, Iteración y Tipo de prueba, y además busca los Probador y le asigna a los Probadores uno ó varios elementos de prueba. El sistema le muestra la asignación de trabajo al Jefe de Prueba.	
Precondiciones	
Deben existir Probadores en el sistema. Deben existir elementos de prueba insertados en el Tipo de prueba, Iteración, Módulo y Proyecto seleccionado. Debe haber un proyecto revisándose.	
Referencias: RF 3, RF 4	
Casos de uso asociados:	
Buscar elementos de prueba.	
Poscondiciones	
Quedan asignados los elementos de prueba a los Probadores.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de prueba necesita repartir el trabajo a los Probadores.	2. El sistema muestra la interfaz para asignar el trabajo a los Probadores.

<p>3. El Jefe de prueba busca los elementos de prueba Ver caso de uso asociado Buscar elemento de prueba</p>	<p>4. El sistema brinda la posibilidad de buscar un Probador por los siguientes criterios de búsqueda.</p> <ul style="list-style-type: none"> • Nombre • Apellidos • Sexo <p>Además</p> <ul style="list-style-type: none"> • Si es trabajador por la organización a la que pertenece. • Si es estudiantes por facultad y grupo.
<p>5. El Jefe de prueba introduce los criterios de búsquedas por los que desee buscar a los Probadores y presiona buscar.</p>	<p>6. El sistema busca a los Probadores según los criterios de búsquedas especificados.</p>
	<p>7. El sistema muestra todos los probadores que cumplan con los criterios de búsquedas especificados, con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre • Apellidos • Sexo • Teléfono • Email <p>Además</p> <ul style="list-style-type: none"> • Si es trabajador muestra la organización a la que pertenece • Si es estudiante muestra la facultad y el grupo al que pertenece. <p>En caso de no encontrarse probadores ver curso alterno 7.1</p>
<p>8. El Jefe de prueba selecciona un Probador y el ó los elementos de prueba que desea asignarle al probador, presiona asignar.</p>	<p>9. El sistema marca al Probador y el ó los elementos de prueba seleccionado por el Jefe de prueba.</p>
<p>10. El Jefe de prueba asigna el ó los elementos de prueba al probador.</p>	<p>11. El sistema muestra la asignación de trabajo.</p>

	12. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Eliminar asignación ver sección Eliminar. Ver sección Eliminar
	13. El sistema finaliza la ejecución del caso de uso.
Cursos alternos	
	7.1 Si no encuentra ningún Probador el sistema muestra el siguiente mensaje de información “No se encontraron Probadores”. Proveniente del flujo básico 4
Sección : Eliminar	
1. El Jefe de prueba selecciona una asignación.	2. El sistema selecciona la asignación que el Jefe de prueba haya seleccionado.
3. El Jefe de prueba elimina la asignación	4. El sistema muestra un mensaje de confirmación “Desea eliminar la asignación”.
5. Si el Jefe de prueba desea eliminar la asignación y presiona Aceptar. En caso contrario ver Curso alternativo 5.1	6. El sistema elimina la asignación
Cursos alternos	
5.1 Si el Jefe de prueba no desea eliminar la asignación y presiona Cancelar. Proveniente del flujo básico 1	

Tabla 2.18 Descripción del caso de uso “Asignar Trabajo”

Caso de uso: Buscar elemento de prueba
Actores: Jefe de prueba.
Propósito
Buscar los elementos de prueba de un Tipo de prueba, Iteración, Módulo y Proyecto.
Resumen
El caso de uso se inicia cuando el Jefe de prueba desea busca los elementos de prueba. El Jefe de prueba selecciona el Proyecto, Módulo, Iteración y Tipo de prueba y el sistema muestra los elementos de prueba relacionados con los parámetros seleccionados.
Precondiciones
Deben existir elementos de prueba insertados en el Tipo de prueba, Iteración, Módulo y Proyecto seleccionado.
Referencias: RF 2

Casos de uso asociados:	
Poscondiciones	
Se mostraron todos los elementos de pruebas que tenga el Tipo de prueba, Iteración, Módulo, Proyecto seleccionado.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de prueba necesita buscar los elementos de prueba.	2. El sistema muestra la interfaz para buscar los elementos de prueba.
	3. El sistema brinda la posibilidad de seleccionar los siguientes criterios de búsquedas: <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración • Tipo de prueba
4. El Jefe de prueba introduce los criterios de búsquedas y presiona buscar.	5. El sistema busca los elementos de prueba según los criterios de búsqueda seleccionados.
	6. El sistema muestra todos los elementos de prueba según los criterios de búsquedas especificados. En caso que no se encuentren elementos de prueba ver curso alterno 6.1
7. El Jefe de prueba selecciona un elemento de prueba.	8. El sistema selecciona el elemento de prueba que el Visualizador de No Conformidades haya seleccionado.
	9. El sistema finaliza la ejecución de caso de uso.
Cursos alternos	
	6.1 El sistema muestra el siguiente mensaje de información "No se encontraron elementos de prueba". Proveniente del flujo básico 3

Tabla 2.19 Descripción del caso de uso "Buscar elemento de prueba"

2.8.2 El paquete Gestionar No Conformidades.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

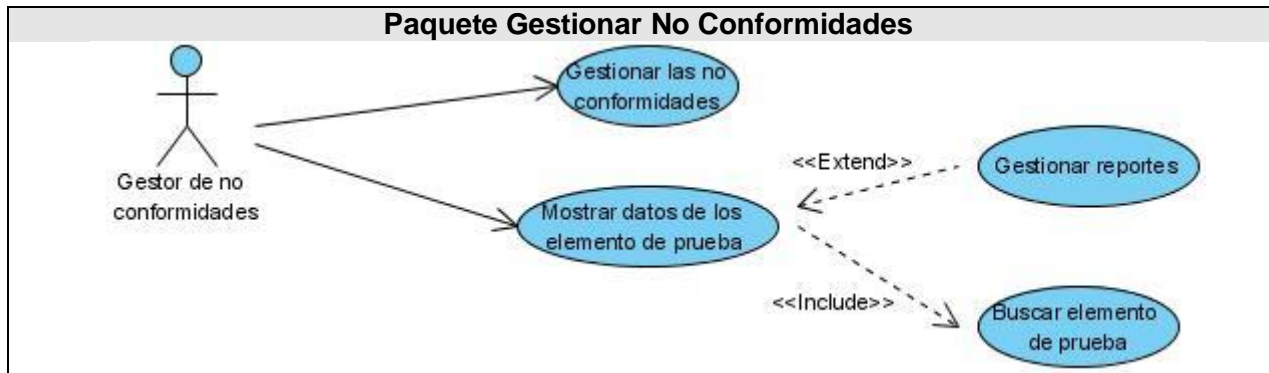


Figura 2.8 Diagrama de paquete “Gestionar No Conformidades”.

Descripción de los casos de uso del Paquete Gestión de las No Conformidades.

Caso de uso: Gestionar las No Conformidades	
Actores: Gestor de No Conformidades	
Propósito	
Insertar, eliminar y modificar las No Conformidades encontradas en las pruebas.	
Resumen	
El caso de uso se inicia cuando el Gestor de No Conformidades desea insertar las No Conformidades. El Gestor de No Conformidades selecciona el Proyecto, Módulo, Iteración, Tipo de prueba y buscar los elementos de prueba, y después le inserta las No Conformidades encontradas en las pruebas, también puede modificarlas ó eliminarlas.	
Precondiciones	
Deben de existir elementos de pruebas.	
Referencias: RF 7, RF 12, RF 17	
Casos de uso asociados:	
Poscondiciones	
Quedan insertadas, modificadas ó eliminadas las No Conformidades.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema

<p>1. El Gestor de No Conformidades desea gestionas las No Conformidades.</p>	<p>2. El sistema muestra la interfaz para buscar los módulos de un Proyecto, seleccionado el nombre de un Proyecto.</p>
<p>3. El Gestor de No Conformidades selecciona el nombre del Proyecto y presiona buscar</p>	<p>4. El sistema muestra los módulos con los siguientes datos:</p> <ul style="list-style-type: none"> • Módulo • Iteración <p>Y brinda la posibilidad de seleccionar el Tipo de prueba.</p>
<p>5. El Gestor de No Conformidades selecciona el Tipo de prueba y presiona buscar.</p>	<p>6. El sistema muestra el tipo de prueba con los siguientes datos:</p> <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración • Tipo de prueba <p>Y brinda la posibilidad de seleccionar un elemento de prueba.</p>
<p>7. El Gestor de no conformidades selecciona el elemento de prueba y presiona Mostrar</p>	<p>8. El sistema muestra las no conformidades.</p> <ul style="list-style-type: none"> • Elemento • Número • No Conformidad • Etapa de detección • Clasificación <p>El sistema brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Si el usuario desea insertar un no conformidad ver sección Insertar • Si el usuario desea eliminar un no conformidad ver sección Eliminar • Si el usuario desea modificar una no conformidad ver sección Modificar.
	<p>9. El sistema brinda la posibilidad de mandar confirmación de trabajo terminado.</p>
<p>10. El Gestor de no conformidades presiona enviar confirmación.</p>	<p>11. El sistema abre el correo Outlook y brinda la posibilidad de escoger el usuario.</p>
<p>12. El Gestor de no conformidades introduce el usuario y presiona enviar.</p>	<p>13. El sistema envía el correo de aviso.</p>
	<p>14. El sistema finaliza la ejecución del caso de uso.</p>

Sección : Insertar	
	<p>1. El sistema muestra una interfaz para insertar las No Conformidades con los siguientes datos:</p> <ul style="list-style-type: none"> • Elemento • Número • No conformidad • Aspectos correspondientes • Etapa de detección • Importancia • Recomendaciones
<p>2. El Gestor de No Conformidades llena los campos de la no conformidad. En caso de no llenar todos los campos ver Curso alterno 2.1</p>	<p>3. El sistema inserta la no conformidad y las muestra con los siguientes datos:</p> <ul style="list-style-type: none"> • Elemento • Número • No conformidad • Aspectos correspondientes • Etapa de detección • Importancia • Recomendaciones
Cursos alternos	
	<p>2.1 El sistema muestra un mensaje de información “Debe llenar todos los campos”. Proveniente del flujo básico 1</p>
Sección : Modificar	
<p>1. El Gestor de No Conformidades selecciona modificar la no conformidad.</p>	<p>2. El sistema muestra una ventana para modificar la no conformidad para modificar los siguientes datos:</p> <ul style="list-style-type: none"> • Elemento • Número • No conformidad • Aspectos correspondientes • Etapa de detección • Importancia • Recomendaciones
<p>3. El Gestor de No Conformidades modifica cualquiera de los datos de la no conformidad:</p> <ul style="list-style-type: none"> • Elemento • Número 	<p>4. El sistema modifica los datos de los elementos de prueba y muestra los siguientes datos:</p> <ul style="list-style-type: none"> • Elemento • Número • No conformidad

<ul style="list-style-type: none"> • No conformidad • Aspectos correspondientes • Etapa de detección • Importancia • Recomendaciones <p>Y presiona modificar.</p>	<ul style="list-style-type: none"> • Aspectos correspondientes • Etapa de detección • Importancia • Recomendaciones
Sección : Eliminar	
1. El Gestor de No Conformidades solicita eliminar la no conformidad.	2. El sistema muestra un mensaje de confirmación “Desea eliminar la no conformidad”.
3. Si el Gestor de No Conformidades desea eliminar la no conformidad presiona Aceptar. En caso contrario ver Curso alterno 3.1	4. El sistema elimina la no conformidad
Cursos alternos	
3.1 Si el Jefe de prueba no desea eliminar una no conformidad presiona Cancelar. Proveniente del flujo básico 1	

Tabla 2.20 Descripción del caso de uso “Gestionar las No Conformidades”

Caso de uso: Mostrar datos de los elementos de prueba
Actores: Gestor de No Conformidades
Propósito
Mostrar los elementos de prueba, con su estado, complejidad y el probador asignado a el.
Resumen
El caso de uso se inicia cuando el Gestor de No Conformidades desea ver la asignación de trabajo y los datos del elemento de prueba. El Gestor de No Conformidades introduce el Proyecto, Módulo, Iteración y Tipo de prueba para que se muestren los elementos de pruebas con sus datos: Nombre del elemento de prueba, Clasificación, Complejidad, Probador asignado y Fecha de culminación.
Precondiciones
Deben existir elementos de prueba insertados.
Referencias: RF 6
Casos de uso asociados:
Buscar elementos de prueba.
Poscondiciones
Se mostraron los datos de los elementos de prueba: Nombre del elemento de prueba, Clasificación, Complejidad, Probador asignado y Fecha de culminación.
Requerimientos especiales:

Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Gestor de No Conformidades desea ver los datos de los elementos de prueba.	2. El sistema muestra la interfaz para mostrar los datos de los elementos de prueba.
3. El Gestor de No Conformidades busca el elemento de prueba (ver caso de uso asociado Buscar elementos de prueba)	
4. El Gestor de No Conformidades selecciona mostrar los datos.	5. El sistema muestra los datos del elementos de prueba: <ul style="list-style-type: none"> • Nombre del elemento de prueba • Complejidad • Clasificación • Fecha de culminación • Probador asignado
	6. El sistema finaliza la ejecución del caso de uso.

Tabla 2.21 Descripción del caso de uso “Mostrar datos de los elementos de prueba”

Caso de uso: Gestionar reportes
Actores: Gestor de No Conformidades
Propósito
Poder imprimir y/o exportar un reporte
Resumen
El caso de uso de inicia cuando el Gestor de No Conformidades después de tener un reporte puede solicitar imprimirlo y/o exportarlo.
Precondiciones
Debe tener un reporte.
Referencias: RF 11
Casos de uso asociados:
Poscondiciones
Imprime y/o exporta un reporte.
Requerimientos especiales:
Prototipo:

Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El caso de uso comienza cuando el Gestor de No Conformidades desea imprimir y/o exportar un reporte. El Gestor de no conformidades Presiona Vista de Impresión.	2. El sistema muestra una vista del reporte. Y brinda la posibilidad de <ul style="list-style-type: none"> • Si desea imprimir un reporte ver Sección Imprimir. • Si desea exportar un reporte ver Sección Exportar.
	3. El sistema finaliza la ejecución del caso de uso.
Sección : Imprimir	
1. El Gestor de No Conformidades solicita imprimir un reporte.	2. El sistema muestra una ventana para seleccionar la configuración de la impresión.
3. El Gestor manda a imprimir el reporte.	4. El sistema imprime el reporte.
Sección : Exportar	
1. El Gestor de No Conformidades solicita exportar un reporte.	2. El sistema muestra una ventana para seleccionar la dirección donde se desea guardar.
3. El Gestor de No Conformidades manda a exportar el reporte.	4. El sistema exporta el reporte.

Tabla 2.22 Descripción del caso de uso “Gestionar reportes”

2.8.3 El paquete Insertar interrupción.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

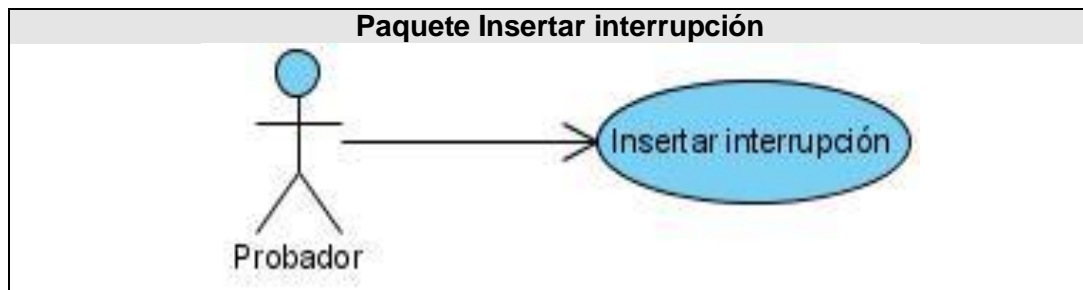


Figura 2.9 Diagrama de paquete “Insertar Interrupción”.

Descripción de los casos de uso del Paquete Interrupción.

Caso de uso: Insertar interrupción	
Actores: Probador	
Propósito	
Insertar la interrupción en el turno de trabajo.	
Resumen	
El caso de uso se inicia cuando el probador inserta el tiempo de interrupción en su turno de trabajo.	
Precondiciones	
Debe de estar gestionando las No Conformidades	
Referencias: RF 9	
Casos de uso asociados:	
Poscondiciones	
Quedan insertada la interrupción del turno de trabajo.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Probador desea insertar el tiempo de interrupción de un elemento de prueba.	2. El sistema muestra la interfaz para insertar el tiempo de Interrupción.
	3. El sistema brinda la posibilidad de insertar el tiempo de interrupción de un elemento de prueba: <ul style="list-style-type: none"> • Tiempo de interrupción.
4. El Probador escribe el tiempo y selecciona insertar tiempo de interrupción.	5. El sistema inserta el tiempo de interrupción y muestra un mensaje de información “Se inserto el tiempo de interrupción correctamente”
	6. El sistema finaliza la ejecución del caso de uso.

Tabla 2.23 Descripción del caso de uso “Insertar interrupción”

2.8.4 El paquete Asignar tiempo de trabajo.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.



Figura 2.10 Diagrama de paquete “Asignar tiempo de Trabajo”.

Descripción de los casos de uso del Paquete Tiempo de trabajo

Caso de uso: Asignar tiempo de realización de un elemento de prueba	
Actores: Reloj	
Propósito	
Recoger el tiempo que se tarda un probado para encontrar las No Conformidades.	
Resumen	
El caso de uso se inicia cuando el usuario con rol de Probador entre a la sección de él en el sistema, y seleccione el elemento de prueba al cual le buscara las No Conformidades, el sistema tomara la hora de la maquina como tiempo de inicio de la prueba y cuando el usuario termina de insertar todas las No Conformidades que haya encontrado de un elemento de prueba el sistema toma la hora de la maquina como tiempo de fin de la prueba.	
Precondiciones	
El usuario que inicia su sección en el sistema debe tener rol de Probador El usuario con rol de probador debe de haber hacho una prueba.	
Referencias: RF 14	
Casos de uso asociados:	
Poscondiciones	
Quedan registrados los tiempos de inicio y fin de un Elemento de prueba.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Probador selecciona un elemento de prueba para insertarle las No Conformidades.	2. El sistema recoge la hora de la maquina en ese momento como tiempo de inicio de un elemento de prueba.
3. Cuando el probador termine de insertar todas las No Conformidades que presione insertar.	4. El sistema recoge la hora de la maquina en ese momento como tiempo de fin de un elemento de prueba.

Tabla 2.24 Descripción del caso de uso “Asignar tiempo de realización de un elemento de prueba”

2.8.5 El paquete Insertar respuestas a las No Conformidades.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.



Figura 2.11 Diagrama de paquete “Insertar respuesta a las No Conformidades”.

Descripción de los casos de uso del Paquete Respuestas a las No Conformidades.

Caso de uso: Insertar respuestas a las No Conformidades	
Actores: Desarrollador	
Propósito	
Insertar las respuestas de las No Conformidades.	
Resumen	
El caso de uso se inicia cuando el Desarrollador busca las No Conformidades para insertarle las respuestas.	
Precondiciones	
Deben existir No Conformidades insertadas en su proyecto.	
Referencias: RF 8,	
Casos de uso asociados:	
Buscar elemento de prueba.	
Poscondiciones	
Quedan insertadas las respuestas de las No Conformidades.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Desarrollador desea gestionar las No Conformidades.	2. El sistema muestra la interfaz para buscar las no conformidades por los siguientes criterios de búsqueda. <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración

	<ul style="list-style-type: none"> Tipo de prueba
3. El Desarrollador inserta los criterios de búsquedas y presiona Buscar.	<p>4. El sistema las no conformidades con los siguientes parámetros:</p> <ul style="list-style-type: none"> Elemento Número No conformidad Aspectos correspondientes Etapas de detección Clasificación <p>El sistema brinda la posibilidad de insertar la respuesta a la no conformidad</p>
5. El Desarrollador introduce la respuesta a la no conformidad y presiona insertar.	<p>6. El sistema inserta y muestra la respuesta a la no conformidad con los siguientes datos:</p> <ul style="list-style-type: none"> Elemento Número No conformidad Aspectos correspondientes Etapas de detección Importancia Recomendaciones Respuesta
	7. El sistema finaliza la ejecución de caso de uso.

Tabla 2.25 Descripción del caso de uso “Insertar respuestas a las No Conformidades”

2.8.6 El paquete Mostrar estados.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

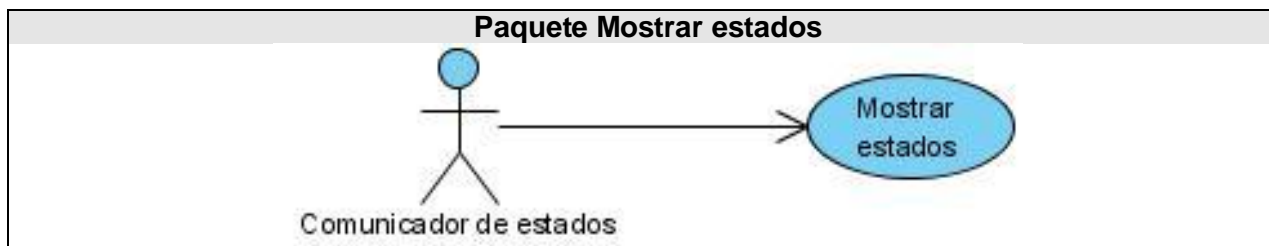


Figura 2.12 Diagrama de paquete “Mostrar Estados”.

Descripción de los casos de uso del Paquete Mostrar estados.

Caso de uso: Mostrar estados	
Actores: Comunicador de estado	
Propósito	
Mostrar los estados de los elementos de prueba, módulos y proyectos	
Resumen	
El caso de uso se inicia cuando el Comunicador de estados selecciona un Proyecto, Módulos, Iteración y elemento de prueba para mostrar el estado en que se encuentran los mismos.	
Precondiciones	
Deben de existir un Proyecto insertado	
Referencias: RF 10,	
Casos de uso asociados:	
Poscondiciones	
Se mostraron los estados de un Proyecto, Módulo y Elementos de prueba.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Comunicador de estados desea ver el estado de un Proyecto, Módulo y sus Elementos de prueba.	2. El sistema muestra la interfaz para buscar los elementos de pruebas por los siguientes criterios de búsquedas: <ul style="list-style-type: none"> • Proyecto • Módulo • Iteración • Tipo de prueba
3. El Comunicador de estado introduce los criterios de búsquedas y presiona buscar.	4. El sistema muestra los siguientes datos: <ul style="list-style-type: none"> • Nombre de Proyecto • Nombre del Módulo • Estado del Módulo • Iteración • Nombre de los elementos de Pruebas • Estados de los elementos de prueba
	5. El sistema Finaliza la ejecución del caso de uso.

Tabla 2.26 Descripción del caso de uso “Mostrar estados”

2.8.7 El paquete Mostrar reportes.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.



Figura 2.13 Diagrama de paquete "Mostrar reportes".

Descripción de los casos de uso del Paquete Reportes.

Caso de uso: Mostrar reportes de las No Conformidades	
Actores: Especialista de calidad	
Propósito	
Mostrar varios reportes de las No Conformidades encontradas.	
Resumen	
El caso de uso se inicia cuando el especialista de calidad selecciona el tipo de reporte que quiere ver, que puede ser: Reporte de las No Conformidades encontradas en una Iteración, Módulo y Proyecto, Reporte de la cantidad de No Conformidades encontradas en una Iteración, Módulo y Proyecto.	
Precondiciones	
Haber terminado la revisión de una Iteración de un Módulo ó Proyecto.	
Referencias: RF 13	
Casos de uso asociados:	
Gestionar Reporte	
Poscondiciones	
Quedan mostrados los diferentes reportes de las No Conformidades.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema

<p>1. El especialista de calidad desea ver un reporte general de las No Conformidades de una iteración de un Módulo o Proyecto.</p>	<p>2. El sistema muestra una interfaz para mostrar el reporte general de las No Conformidades.</p>
	<p>3. El sistema brinda la posibilidad de mostrar los siguientes reportes:</p> <ul style="list-style-type: none"> • En caso de querer mostrar Reporte de las No Conformidades encontradas en un Módulo ó Proyecto ver sección Reporte de No Conformidades encontradas. • En caso de querer mostrar Reporte de la cantidad de No Conformidades encontradas en una Iteración, Módulo y Proyecto ver Sección Reporte de cantidad de No Conformidades encontradas.
	<p>4. El sistema brinda la posibilidad de imprimir y/o exportar el reporte (Ver caso de uso asociado Gestionar Reporte)</p>
	<p>5. El sistema finaliza la ejecución del caso de uso.</p>
<p>Sección: Reporte de No Conformidades encontradas.</p>	
	<p>1. El sistema brinda la posibilidad de seleccionar:</p> <ul style="list-style-type: none"> • Proyecto. • Módulo. • Iteración. • Tipo de prueba.
<p>2. El Especialista de calidad introduce los criterios de búsqueda.</p>	<p>3. El sistema muestra el reporte de las No Conformidades encontradas.</p>
<p>Sección: Reporte de cantidad de No Conformidades encontradas.</p>	
	<p>4. El sistema brinda la posibilidad de seleccionar:</p> <ul style="list-style-type: none"> • Proyecto. • Módulo. • Iteración. • Tipo de prueba.

5. El Especialista de calidad introduce los criterios de búsqueda.	6. El sistema muestra el reporte de cantidad de No Conformidades encontradas.
--	---

Tabla 2.27 Descripción del caso de uso “Mostrar reportes de las No Conformidades”

2.8.8 El paquete Administrar usuario.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

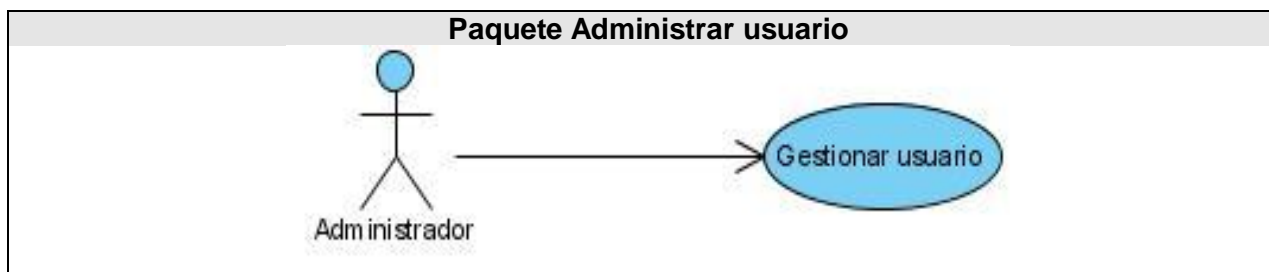


Figura 2.14 Diagrama de paquete “Administrar usuario”.

Descripción de los casos de uso del Paquete Administrar usuario.

Caso de uso: Gestionar usuario
Actores: Administrador
Propósito
Crear un nuevo usuario, editar datos de usuario, asignar roles a usuarios, restringir roles a usuarios y eliminar usuarios
Resumen
El caso de uso se inicia cuando el Administrador va a realizar alguna de las siguientes operaciones: <ul style="list-style-type: none"> • Crear un nuevo usuario. • Editar datos de usuario. • Buscar y visualizar datos de usuario. • Asignar roles a usuarios. • Restringir roles a usuarios. • Eliminar usuario.
El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.
Precondiciones
El usuario debe ser administrador.
Referencias: RF 15
Casos de uso asociados:

Poscondiciones	
Queda creado, editado los datos, asignado el rol, restringido el rol ó eliminado un usuario.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador quiere realizar alguna de las operaciones siguientes: <ul style="list-style-type: none"> • Crear un nuevo usuario, • Editar datos de usuario. • Asignar roles a usuarios. • Restringir roles a usuarios. • Eliminar usuario. 	2. El sistema en dependencia de la operación que solicita el Administrador hace lo siguiente: <ul style="list-style-type: none"> • Si desea Crear un nuevo usuario ir a la Sección “Crear un nuevo usuario”. • Si desea Editar datos de usuario ir a la Sección “Editar datos de usuario”. • Si desea Asignar roles a usuarios ir a la Sección “Asignar roles a usuarios”. • Si desea Eliminar usuario ir a la Sección “Eliminar usuario”.
	3. El sistema finaliza la ejecución del caso de uso
Sección: Crear un nuevo usuario.	
	1. Muestra la interfaz correspondiente para la creación de un nuevo usuario.
	2. Brinda la posibilidad de crear un nuevo usuario con los siguientes datos: <ul style="list-style-type: none"> • Usuario • Contraseña • Asignar rol al usuario
3. Provee los datos necesarios para crear un nuevo usuario.	4. Crea un nuevo usuario.
	5. Da la posibilidad de crear otro usuario.
Cursos alternos	
	4.1 Si falta algún dato necesario para la creación del nuevo usuario, emite un mensaje de error “Debe de llenar todos los datos del usuario” y no crea el usuario.. Proveniente del flujo básico 2
Sección: Editar datos de usuario.	

	1. Muestra la interfaz correspondiente para la edición de datos de un usuario.
	2. El sistema brinda la posibilidad de buscar un usuario según: <ul style="list-style-type: none"> • Usuario. • Contraseña.
3. El administrador introduce uno de los criterios de búsqueda.	4. El sistema busca y muestra los usuarios con los criterios de búsquedas especificados.
	5. El sistema brinda posibilidad de editar los datos del usuario: <ul style="list-style-type: none"> • Usuario • Contraseña
6. EL usuario edita los datos del usuario: <ul style="list-style-type: none"> • Usuario • Contraseña 	7. El sistema cambia los datos del usuario.
	8. Da la posibilidad de editar los datos de otro usuario.
Cursos alternos	
	4.1 En caso de no existir el usuario especificado se muestra un mensaje de información "No se encontró el usuario" Proveniente del flujo básico 2
Sección: Asignar roles a usuarios.	
	1. Ejecuta los pasos del 1 al 4 de la sección "Editar datos de usuario" .
	2. El sistema brinda la posibilidad de asignar los siguientes roles a los usuarios: <ul style="list-style-type: none"> • Jefe de Prueba • Especialista • Desarrollador • Probador • Administrador

3. El Administrador selecciona el rol que va asignar.	4. El sistema asigna el rol especificado.
Sección: Eliminar usuarios.	
	1. Ejecuta los pasos del 1 al 4 de la sección “Asignar roles a usuarios” .
2. El administrador selecciona el usuario que desea borrar.	3. Muestra un mensaje de ratificación “Esta seguro que sede a eliminar el usuario”
4. Acepta el mensaje.	5. El sistema elimina el usuario seleccionado.
Cursos alternos	
4.1 Cancelar el mensaje. Retorna al 2	

Tabla 2.28 Descripción del caso de uso “Gestionar usuario”

2.8.9 El paquete Autenticar usuario.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete.

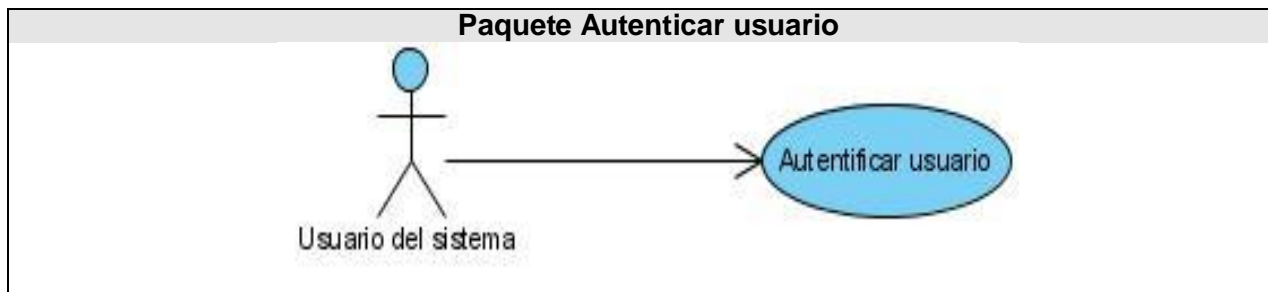


Figura 2.15 Diagrama de paquete “Autenticar usuario”.

Descripción de los casos de uso del Paquete Usuario.

Caso de uso: Autenticar usuario
Actores: Usuario del sistema
Propósito
Permitir autenticarse.
Resumen
El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.

Precondiciones	
El usuario debe estar registrado.	
Referencias: RF 16	
Casos de uso asociados:	
Poscondiciones	
Se habilitan las funcionalidades según los privilegios.	
Requerimientos especiales:	
Prototipo:	
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario introduce nombre de usuario y contraseña.	2. El sistema encripta la contraseña.
	3. El sistema desencripta la contraseña.
	4. Busca el usuario y compara la contraseña.
	5. En caso de ser correcta la contraseña, se le asignan los permisos al usuario y este entra al sistema.
	6. Termina el caso de uso Autenticarse.
Cursos alternos	
	5.1 En caso de no coincidir la contraseña se le envía un mensaje de: "Contraseña incorrecta". Proveniente del flujo básico 1.

Tabla 2.29 Descripción del caso de uso "Autenticar usuario"

2.9 Casos de uso por ciclo.

Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Insertar elemento de prueba	Asignar trabajo	Se insertan los elementos de prueba que mas tarde se revisarán.
CU-2	Buscar elemento de prueba	Asignar trabajo	Se buscan los elementos de prueba que se van asignar a los Probadores, y para realizar otras funcionalidades que el sistema realiza.
CU-3	Asignar trabajo	Asignar trabajo	Se buscan los Probadores y se le asignan los elementos de prueba buscados con

			anterioridad.
CU-4	Mostrar datos de los elementos de prueba	Gestionar No Conformidades	Se muestran los datos de los elementos de prueba con el Probador asignado a cada elemento de prueba.
CU-5	Gestionar las No Conformidades	Gestionar No Conformidades	Se insertan, modifican y eliminan las No Conformidades encontradas en la revisión del software.
CU-6	Gestionar reportes	Gestionar No Conformidades	Se muestran varios reportes relacionados con las No Conformidades encontradas en las revisiones.
CU-7	Asignar tiempo de realización de un elemento de prueba	Asignar tiempo de trabajo	Se asigna el tiempo que se tarda un Probador en revisar un elemento de prueba.
CU-8	Insertar interrupción	Insertar interrupción	Se inserta el tiempo de interrupción de un probador cuando esta revisando un elemento de prueba.
CU-9	Insertar respuestas a las No Conformidades	Insertar respuestas a las No Conformidades	Los Desarrolladores insertan las respuestas a las No Conformidades encontradas en las revisiones.
CU-10	Mostrar estados	Mostrar estados	Se muestran los estados en que se encuentran los Elementos de pruebas, Módulos y Proyectos.
CU-11	Mostar reportes de las No Conformidades	Mostar reportes	Se muestran los distintos reportes relacionados con las No Conformidades encontradas en las revisiones.
CU-12	Gestionar usuario	Administrar usuario	Se crear los usuarios y asignar los roles que deben tener cada uno de ellos.
CU-13	Autenticar usuario	Autenticar usuario	Después que es creado el usuario, este se podrá autenticar en el sistema.

Tabla 2.30 Casos de uso por ciclo

Conclusiones del capítulo.

Como se puede apreciar en este capítulo describimos explícitamente la idea del sistema que pretendemos realizar. Con una explicación clara del negocio, sus requisitos y descripciones de sus casos de uso. Además de conocer y describir las funciones de los actores y trabajadores involucrados en el sistema.

Capítulo 3. Análisis y diseño del sistema.

Introducción.

En este capítulo se analizan los casos de uso del sistema para diseñar las clases que se implementarán, se representan los diagramas de secuencia del diseño, el diagrama de las clases diseñadas con sus relaciones, los principios utilizados para el diseño de dichas clases, el diagrama de clases persistentes, el diagrama entidad relación y las descripciones de las tablas de la base de datos.

3.1 Modelo de Análisis.

El Modelo de análisis es una jerarquía de paquetes del análisis que contienen clases del análisis y realizaciones de casos de uso, además ofrece una especificación más precisa de los requisitos, sin dejar de mencionar que en el modelo de análisis se describe utilizando el lenguaje de los desarrolladores, puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

El modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y en general su mantenimiento; además se puede considerar como la primera aproximación al modelo de diseño.

3.1.1 Diagrama de clases de Análisis.

Una clase de análisis representa una abstracción de una y/o subsistemas del diseño del sistema. Las clases del análisis siempre encajan en uno de los tres estereotipos básicos: de interfaz, de control o entidad. Cada estereotipo implica una semántica específica, lo cual constituye un método potente y consistente de identificar y describir las clases del análisis.

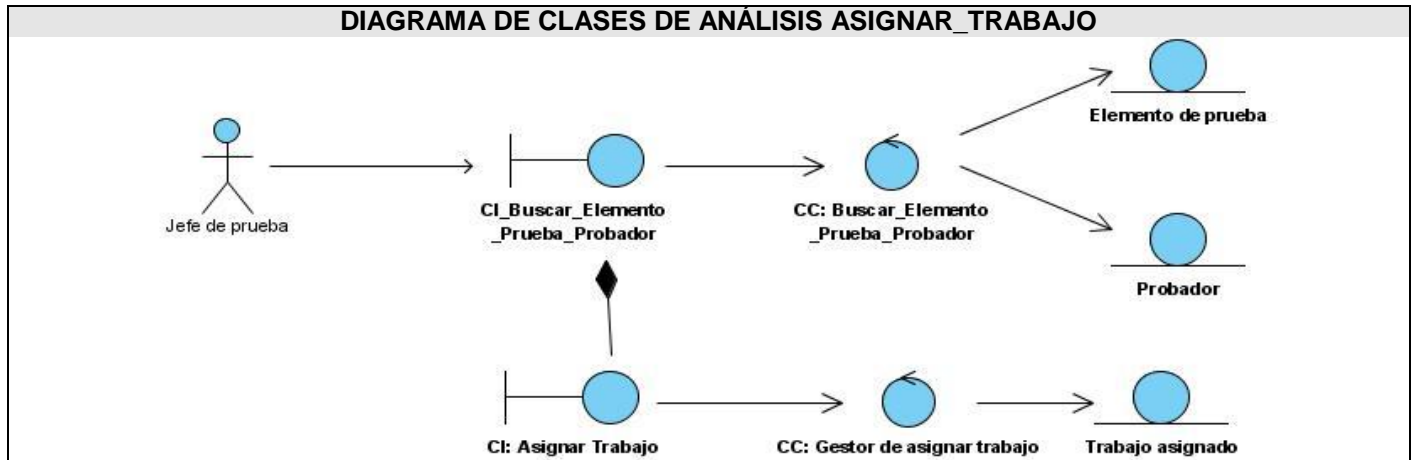


Figura.3.1 Diagrama de Clases de Análisis Asignar Trabajo.

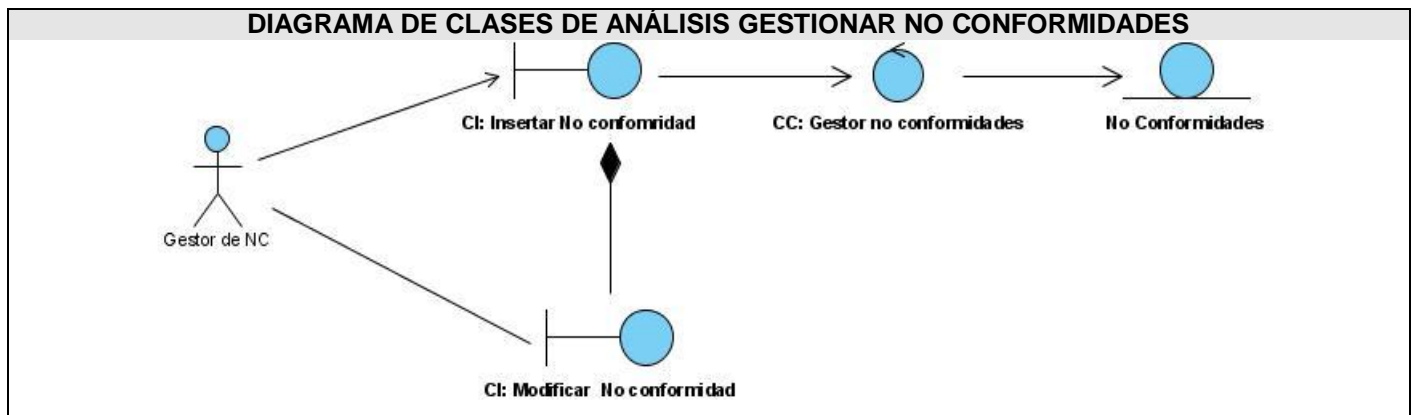


Figura.3.2 Diagrama de Clases de Análisis Gestionar No Conformidades.

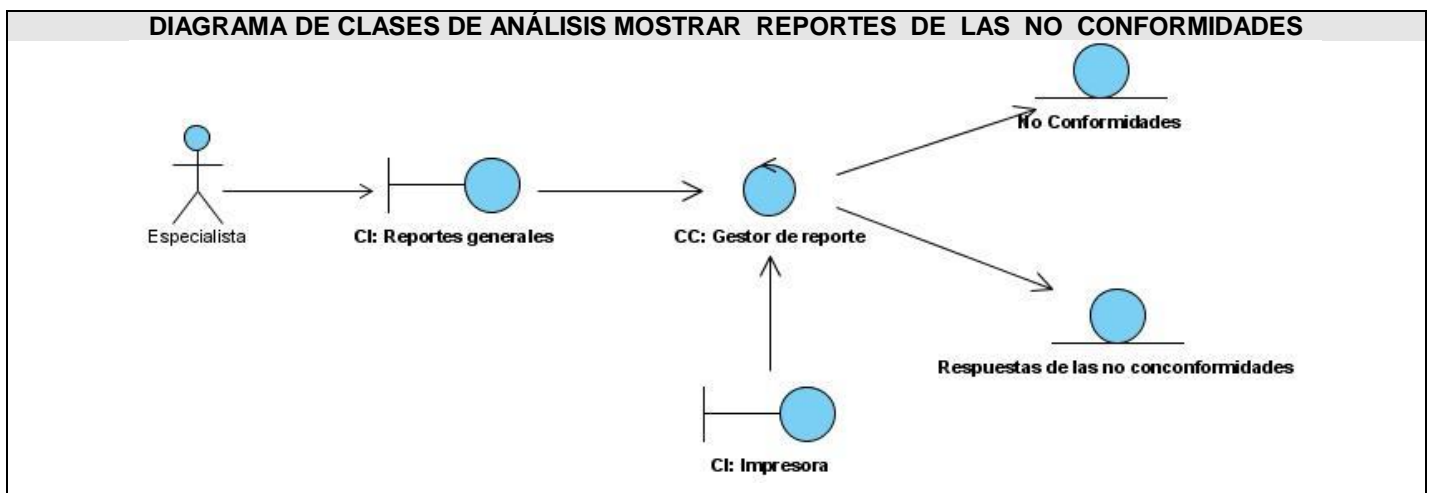


Figura.3.3 Diagrama de Clases de Análisis Mostrar Reporte de las No Conformidades.

3.2 Modelo de diseño.

El modelo de diseño 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 con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Una clase de diseño es una abstracción sin costuras de una clase o construcción similar en la implementación del sistema, debido a que:

- El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación. Esto significa que las operaciones, parámetros, atributos, tipos y demás son especificados utilizando sintaxis del lenguaje de programación elegido.
- Las relaciones de aquellas clases del diseño implicadas con otras clases, a menudo tienen un significado directo cuando la clase es implementada. Por ejemplo, la generalización o algún estereotipo de generalización tiene una semántica que se corresponde con el significado de generalización (o herencia) en el lenguaje de programación. Esto es, las asociaciones y agregaciones a menudo se corresponden con variables (atributos) de clases en la implementación para proporcionar referencias entre objetos.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases (esto es, en el código). Si los métodos se especifican en el diseño, se suelen especificar en lenguaje natural, o en pseudo código, y por eso pueden ser reutilizados como comentarios en las implementaciones del método. Esto es una de las principales abstracciones entre diseño e implementación y es raramente necesario por lo que se recomienda que el mismo desarrollador diseñe e implemente una clase.

UML posee una extensión para el modelado de aplicaciones Web, dicha extensión es usada para el diseño de las clases. Los estereotipos que usa esta extensión son:



<<Server page>>Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

Esta clase sólo puede tener relaciones con objetos en el servidor, una relación 1:1 con un fichero en el servidor. En las aplicaciones en PHP se corresponde con un fichero **.php**.



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML; mezcla de datos, presentación y lógica. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor.



<<Form>> Grupo de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y HiddenFields).

3.2.1 Arquitectura Web de tres niveles.

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no solo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



Figura 3.4 Arquitectura Web de 3 niveles.

3.2.2 Diagramas de secuencia del diseño.

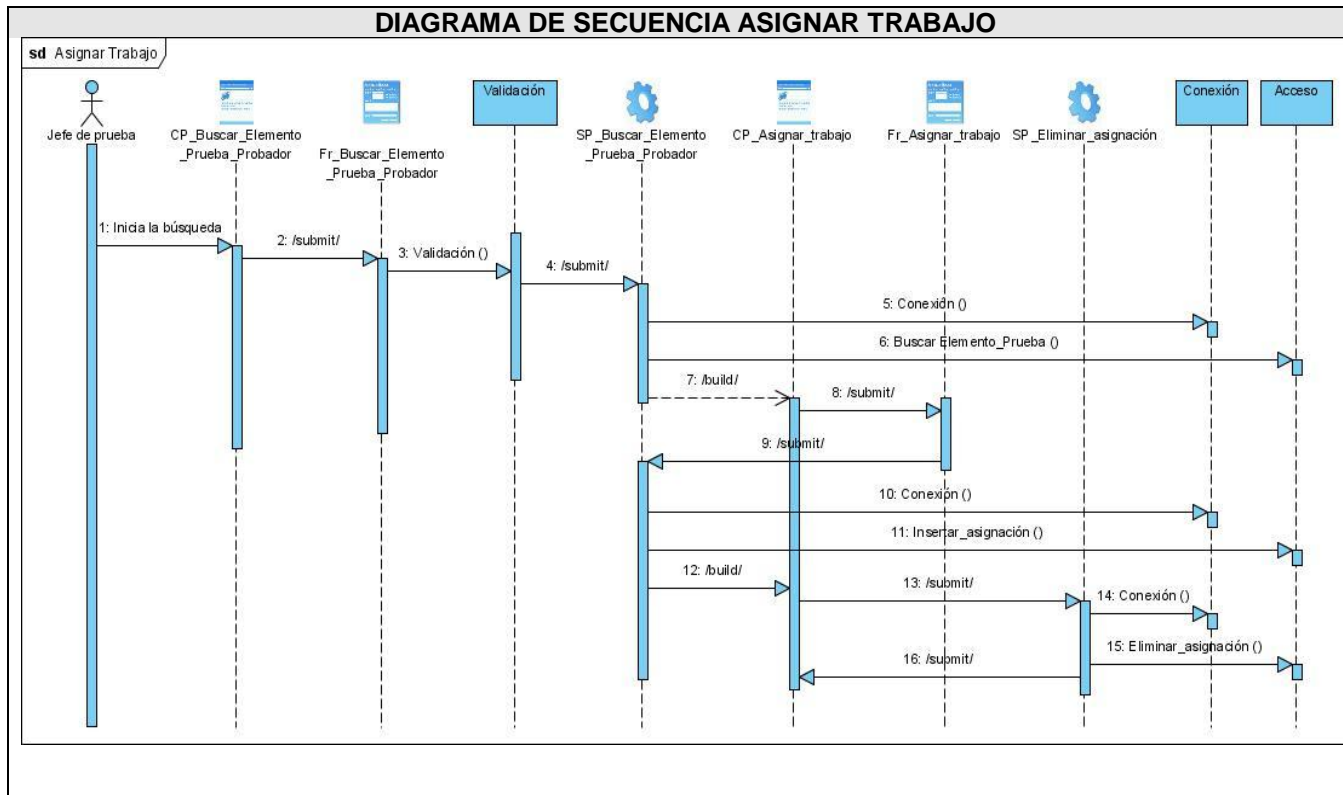


Figura.3.5 Diagrama de secuencia del diseño Asignar Trabajo.

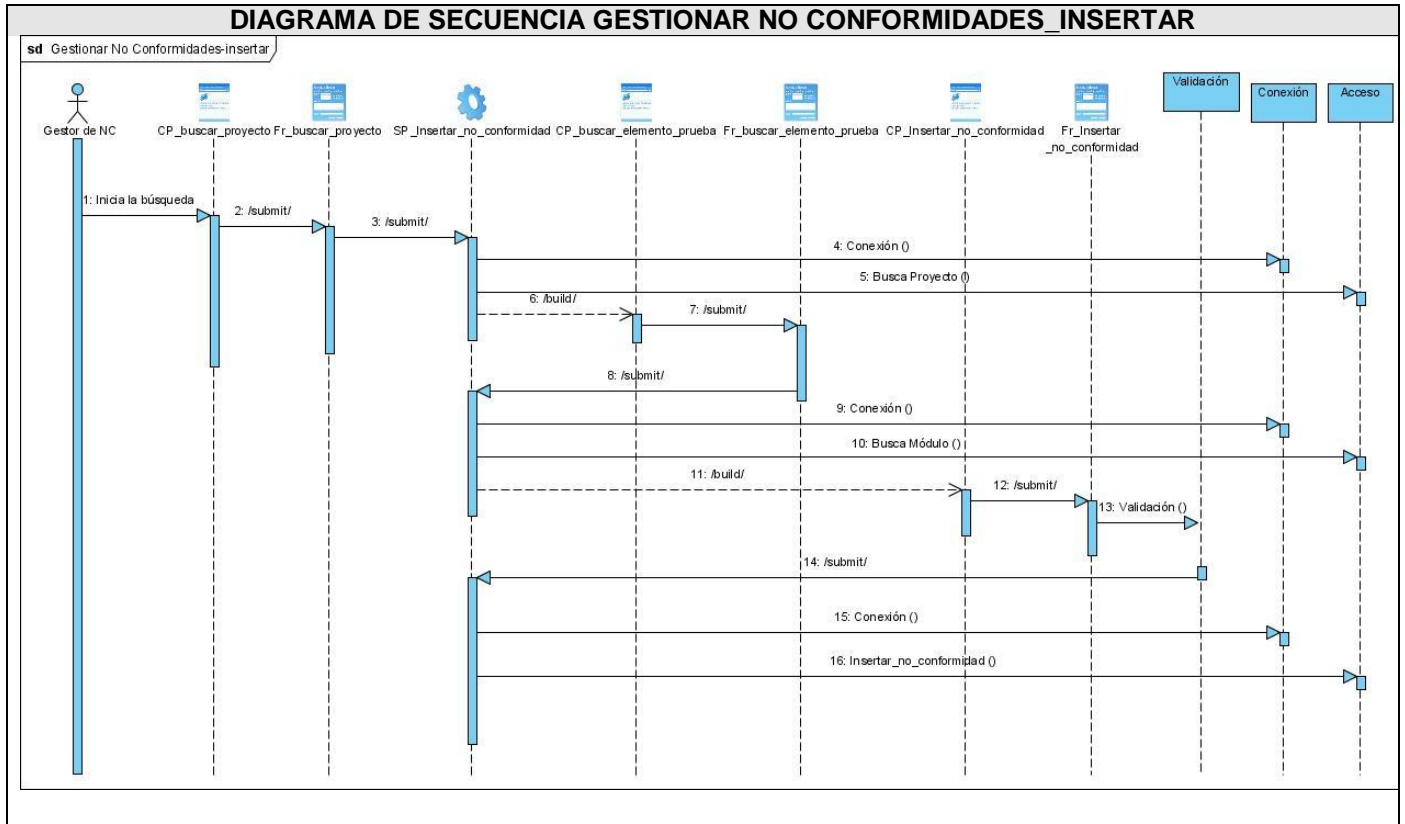


Figura.3.6 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Insertar No Conformidades).

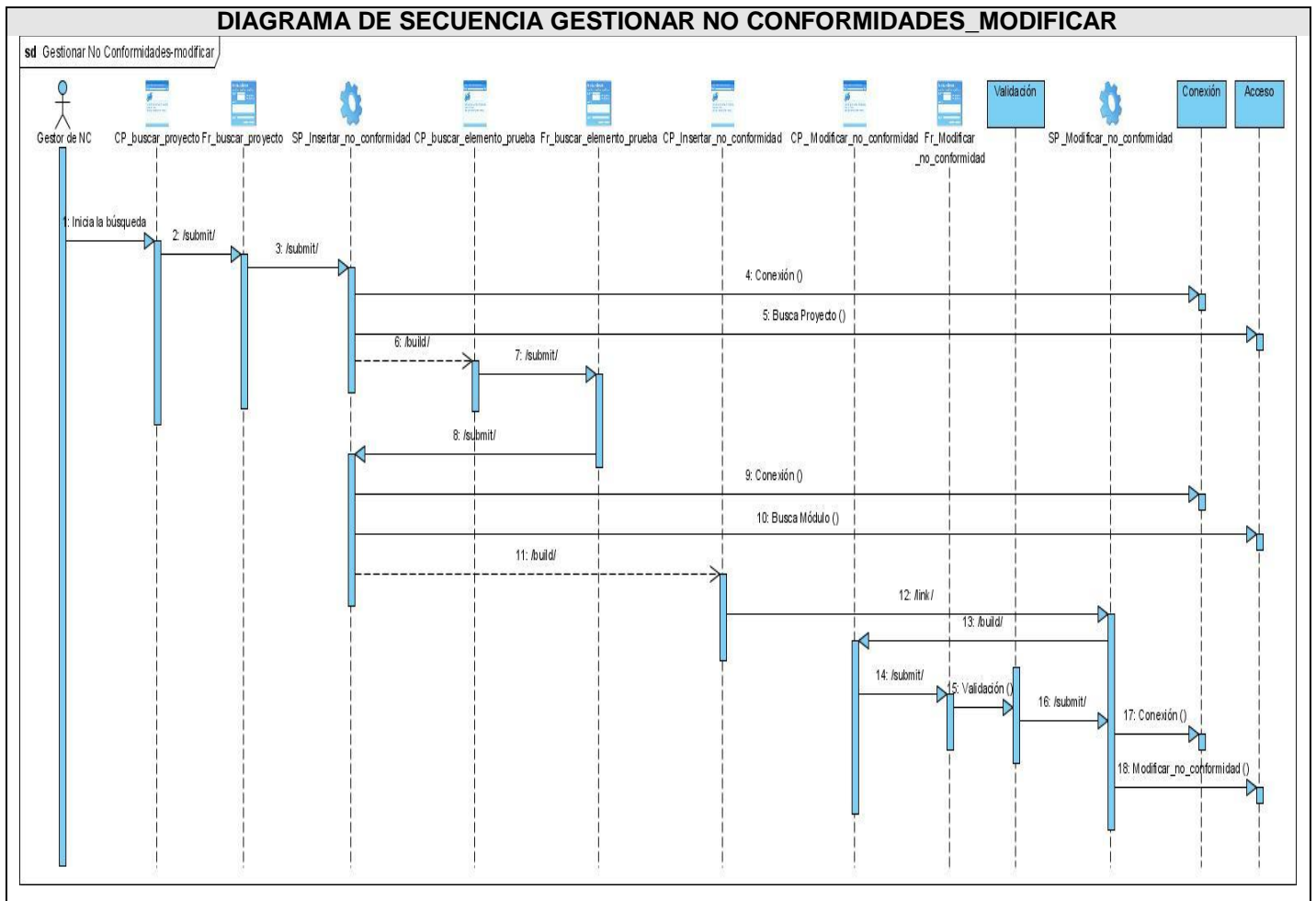


Figura.3.7 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Modificar No Conformidades).

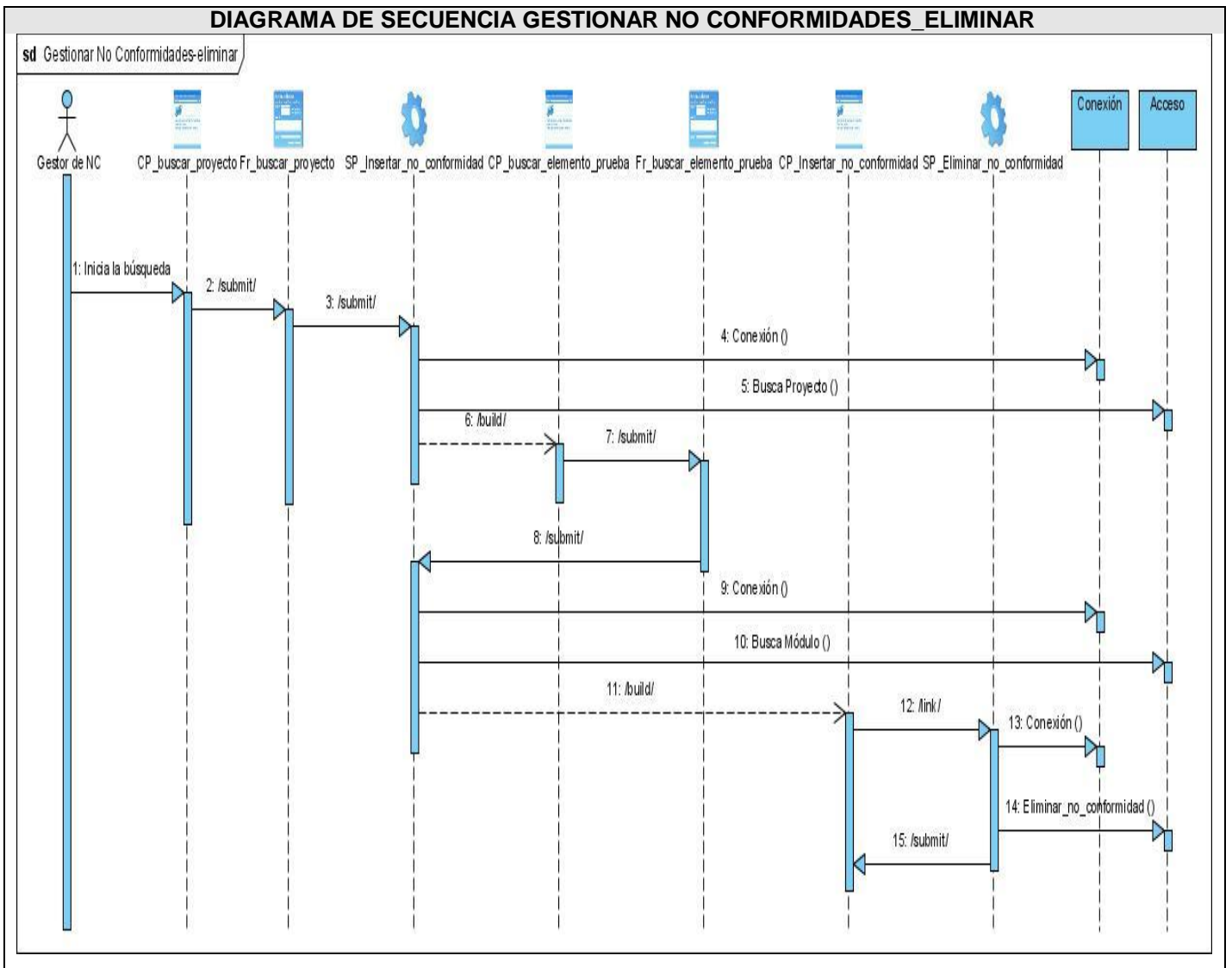


Figura.3.8 Diagrama de secuencia del diseño, escenario del caso de uso Gestionar No Conformidades (Eliminar No Conformidades).

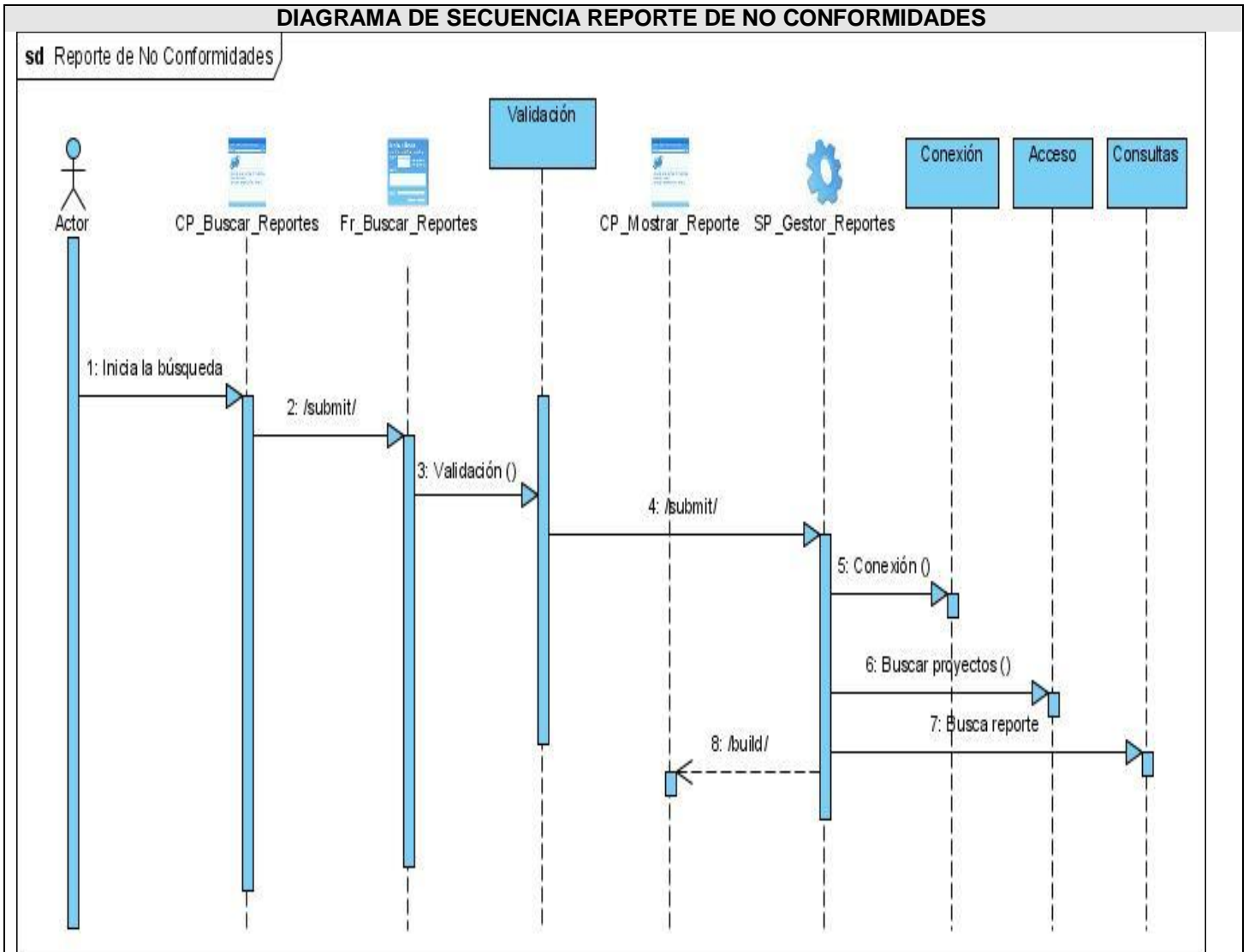


Figura.3.9 Diagrama de secuencia del diseño Reporte de No Conformidades.

3.2.3 Diagrama de clase del Diseño

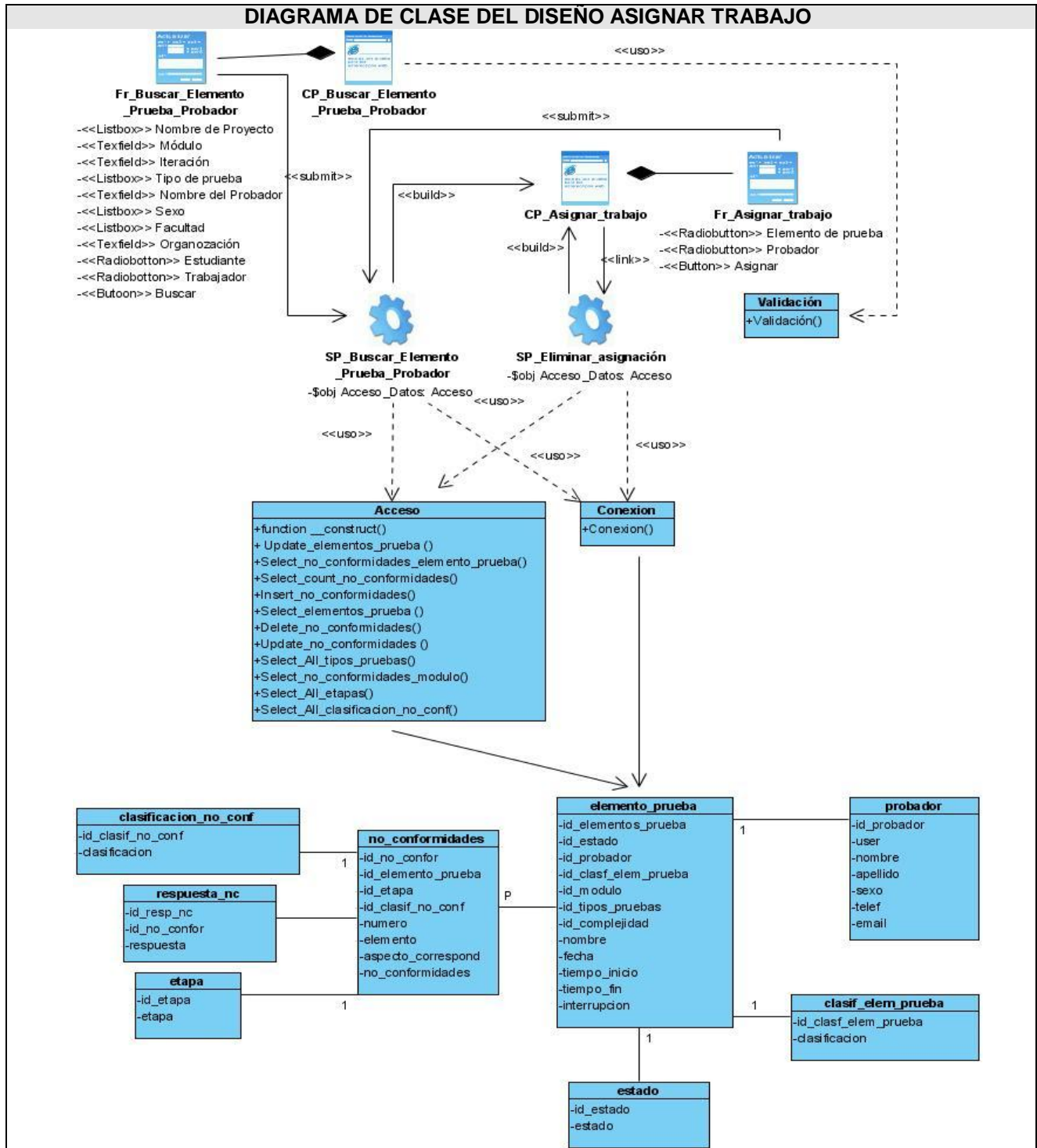


Figura.3.10 Diagrama de clase del diseño Asignar Trabajo.

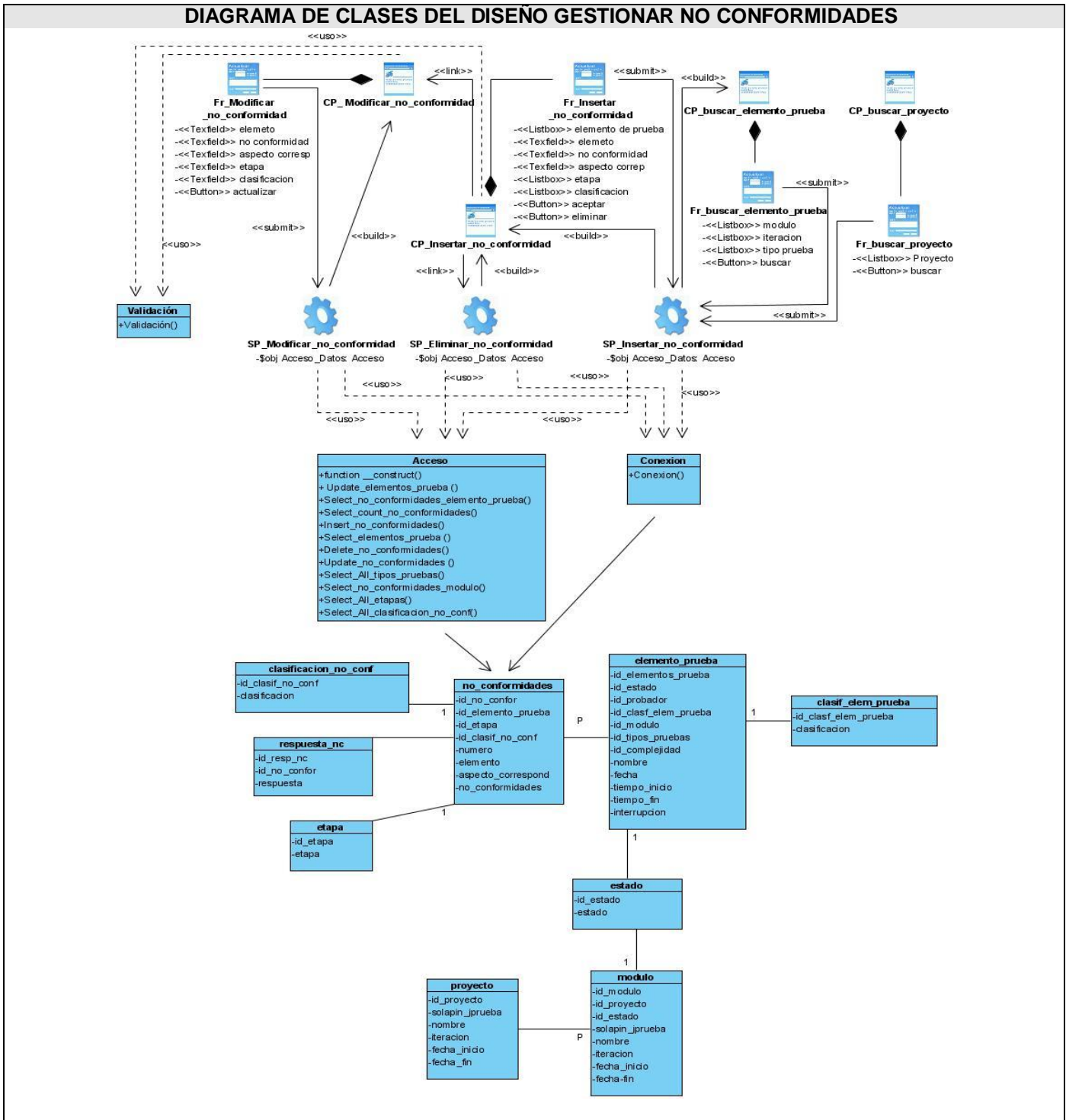


Figura.3.11 Diagrama de clase del diseño Gestionar No Conformidades.

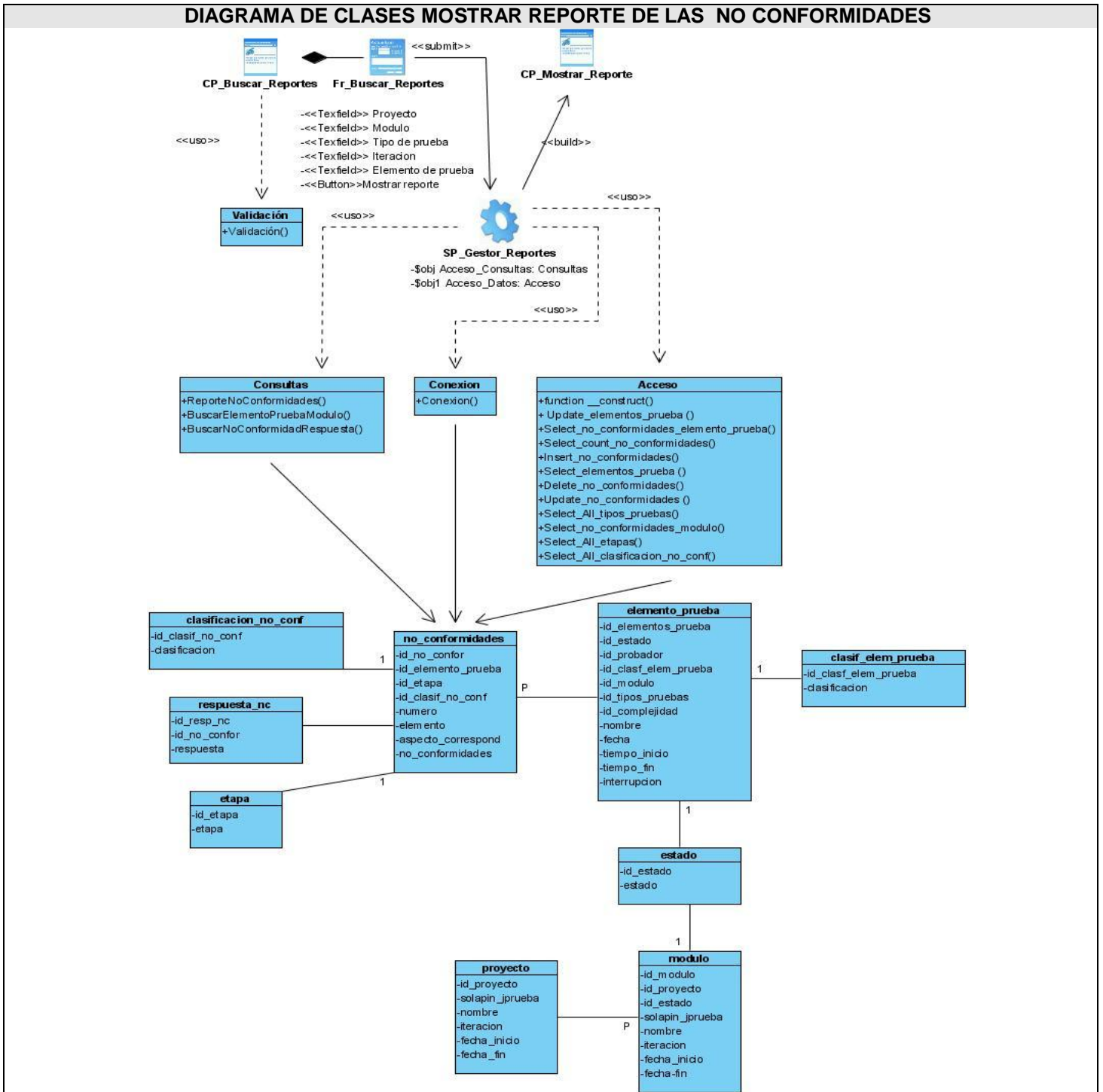


Figura.3.12 Diagrama de clase del diseño Mostrar reportes de las No Conformidades.

3.3 Diseño de la Base de Datos

El diagrama de clases persistentes se representa en el [Anexo 1](#).



Figura.3.13 Modelo Entidad Relación.

3.3.1 Descripción de las tablas.

Se representa la descripción de todas las tablas de la Base de Datos en el [Anexo 2](#).


3.4 Principios de diseño.

Para garantizar un vínculo entre el usuario y el sistema realizado se ha hecho un esfuerzo por garantizar la máxima claridad posible en la interfaz procurando que sea intuitiva y que garantice un empleo óptimo del sistema y una comodidad en el trabajo de los usuarios finales.


Se trabajó con formularios Web, para lograr una mayor visualización de la información, se utilizó un mismo color en todas las páginas y un mismo tipo y tamaño de letra con textos claros, el color predominante es el azul.

El envío de la información debe ser lo mas rápida posible por lo que no se utilizaron muchas imágenes ni funciones que atenten contra esto y que posibiliten también una navegación rápida y eficiente.

Cada página está representada por un título acorde a su funcionalidad. Los botones principales están adornados con iconos que sirven de referencia a la función de los mismos. A continuación se presentan algunos de estos íconos:

Eliminar 

Ayuda 

Modificar 

3.5 Tratamiento de errores.

Se contará con un sistema de tratamiento de errores para disminuir la posibilidad de cometerlos. Para esto se contará con la validación de la información introducida en el sistema por la validación de los formularios utilizando funciones JavaScripts.

Mediante la interfaz Web se evitará que el usuario asuma un papel activo en la introducción de la información, para esto se contará con cuadros de opción, menú de selección lo cual facilitará la entrada de datos. La información que requiera ser adicionada por el usuario se validará mediante funciones que garanticen que sea válida y que el cuadro de texto no esté vacío si es obligatorio llenarlo. Si hay un error en la información le saldrá al usuario un mensaje en pantalla indicándole el error, al oprimir Aceptar el mensaje desaparece y el usuario podrá seguir introduciendo los datos en el formulario.

También se validarán las opciones correspondientes a la extracción o modificación de datos del servidor de base datos. Si se desea eliminar algún elemento de la BD se preguntará al usuario si está seguro de realizar dicha acción, al igual que cuando desee modificar alguna información, antes de actualizarla se le preguntará si desea realizarla o no. Así se logra que se realicen las operaciones que se desean y que se rectifique al cometer un error.

A continuación se muestran algunos de estos mensajes:

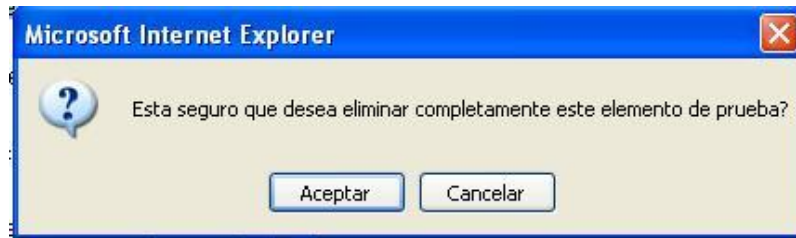


Figura 3.14 Mensaje de verificación. Eliminar Elemento de prueba.



Figura 3.15 Mensaje de validación. Insertar no conformidad.



Figura 3.16 Mensaje de validación. Campo Nombre del Elemento de prueba.



Figura 3.17 Mensaje de validación. Insertar Elemento de prueba.



Figura 3.18 Mensaje de validación. Campo Interrupción.

3.6 Seguridad.

La seguridad en el sitio está implementada a través del servidor de base de datos MySQL y el uso de variables de sesión para restringir el acceso de los usuarios a determinadas páginas. Así como también un Sub_ Módulo de autenticación que verifica que el usuario existe en la aplicación y que los datos que introduce sean correctos, en caso de no ser así no tendrá acceso a la aplicación.

Para garantizar la seguridad de la información se crearon varios niveles de seguridad, definidos como tipos de usuario, que pueden ser: Especialista de Calidad, Jefe de Prueba, Probador, Desarrollador y Administrador. Este último es el encargado del buen funcionamiento del sistema por lo que tendrá derecho al control total del mismo. Los demás usuarios no tendrán acceso a la información restringida para ellos, para esto se trabaja con variables de sesión de forma tal que siempre se sabe qué usuario intenta visitar dichas páginas y estas se muestran sólo para aquellos que pueden tener acceso a ellas.

A continuación se muestran algunos de estos mensajes:



Figura 3.19 Mensaje de verificación. Campo Contraseñas.

Autenticarse	
Usuario :	<input type="text"/>
Contraseña:	<input type="password"/>
Error: Password ó Usuario no existe	
<input type="button" value="Entrar"/>	

Figura 3.20 Interfaz de autenticación.

3.7 Concepción de la ayuda.

El sistema consta de una ayuda que permite a los usuarios del mismo conocer aspectos de su funcionamiento, el vínculo para acceder a la ayuda se encuentra en la plantilla del sistema por lo que se

podrá consultar desde cualquier interfaz, en ella se explica de forma clara y amena todas las funcionalidades del sistema por niveles de permiso.

Conclusiones del capítulo.

En el capítulo se realizaron las definiciones del Modelo de Análisis, Modelo de clases de análisis así como los diagramas de análisis. Y se realizaron los diferentes diagramas de diseño, diagramas de secuencia, se describieron las clases utilizadas en el diseño, así como el modelo entidad relación de la base de datos y las descripciones de las tablas de la base de datos. También se explicó la definición del diseño que se aplicó, la forma de tratar los errores, se definió la seguridad del sistema y la concepción de la ayuda.

Capítulo 4. Implementación y prueba.

Introducción.

En este capítulo se reflejan los diferentes diagramas correspondientes a este flujo de trabajo, donde se implementara la propuesta realizada en el análisis y diseño, se representa el diagrama de despliegue, los de componentes y el modelo de prueba correspondiente a la aplicación desarrollada.

4.1 Implementación

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

4.1.1 Diagrama de despliegue

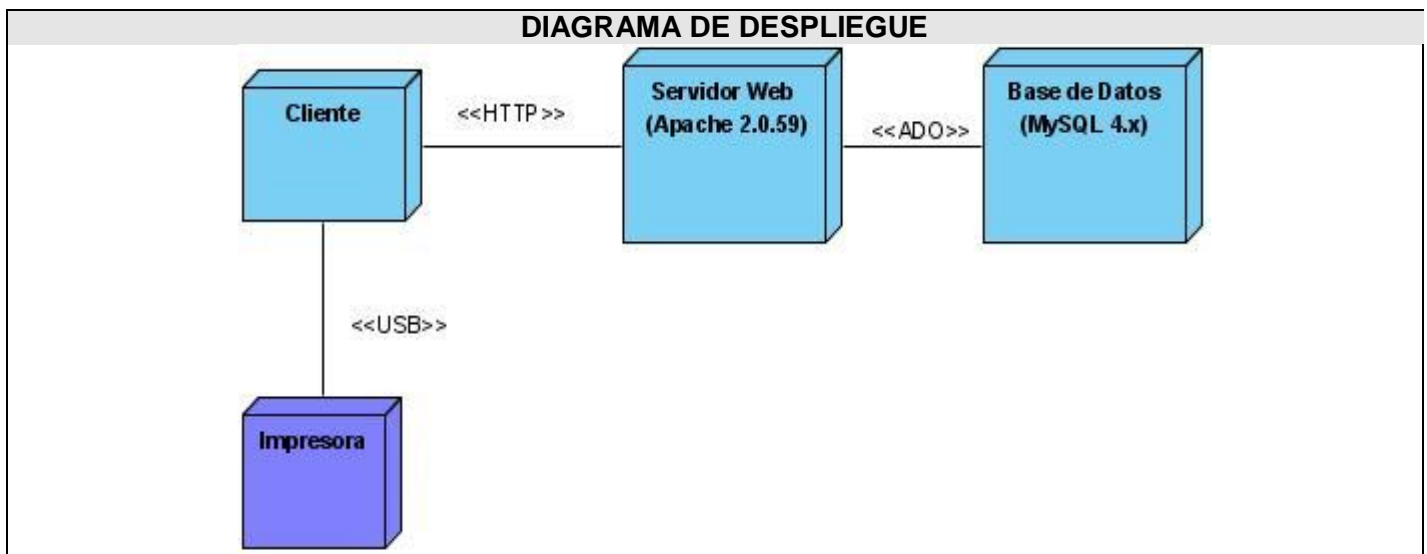


Figura 4.1 Diagrama de despliegue.

4.1.2 Diagramas de componentes

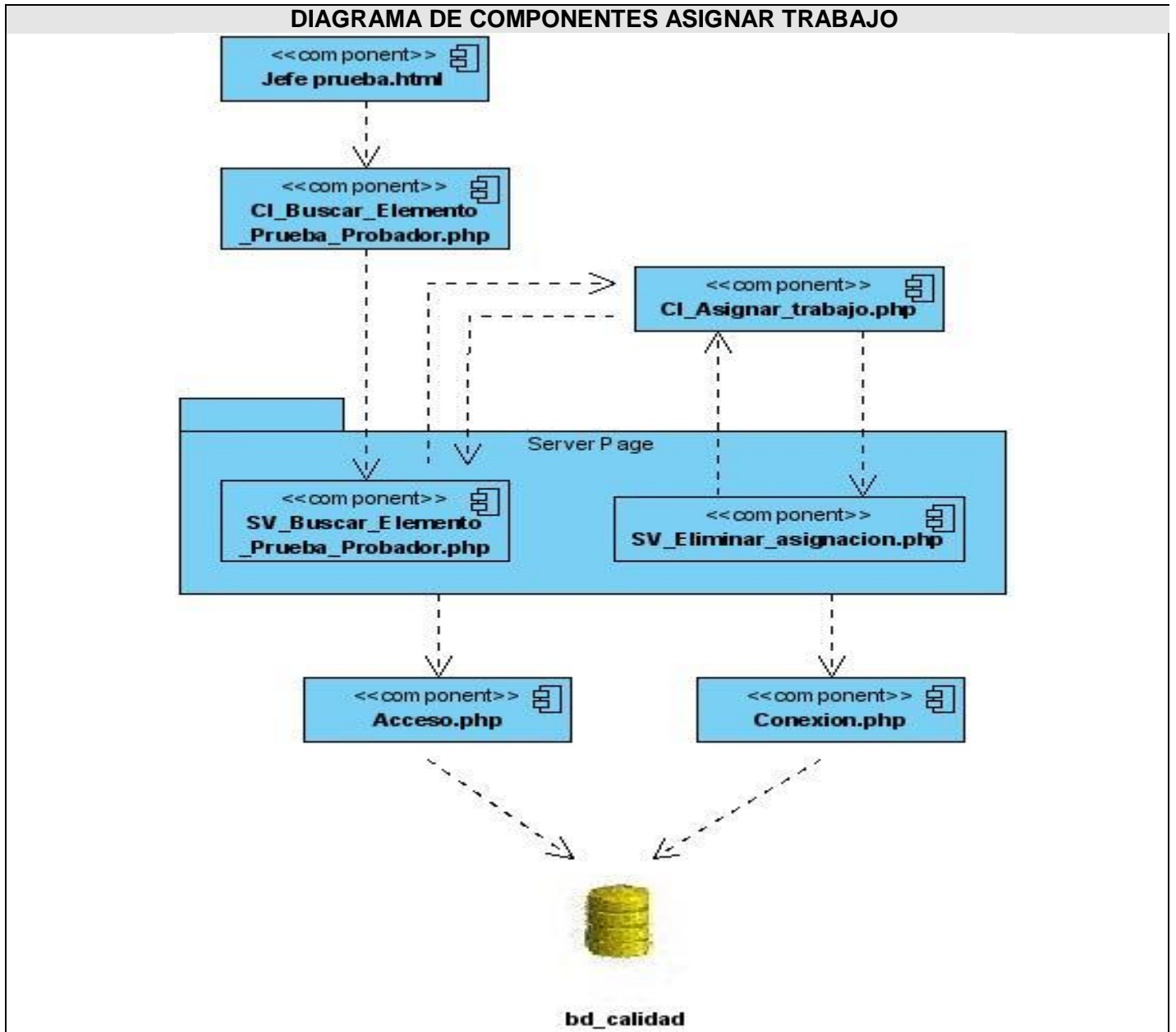


Figura 4.2 Diagrama de componentes Asignar trabajo.

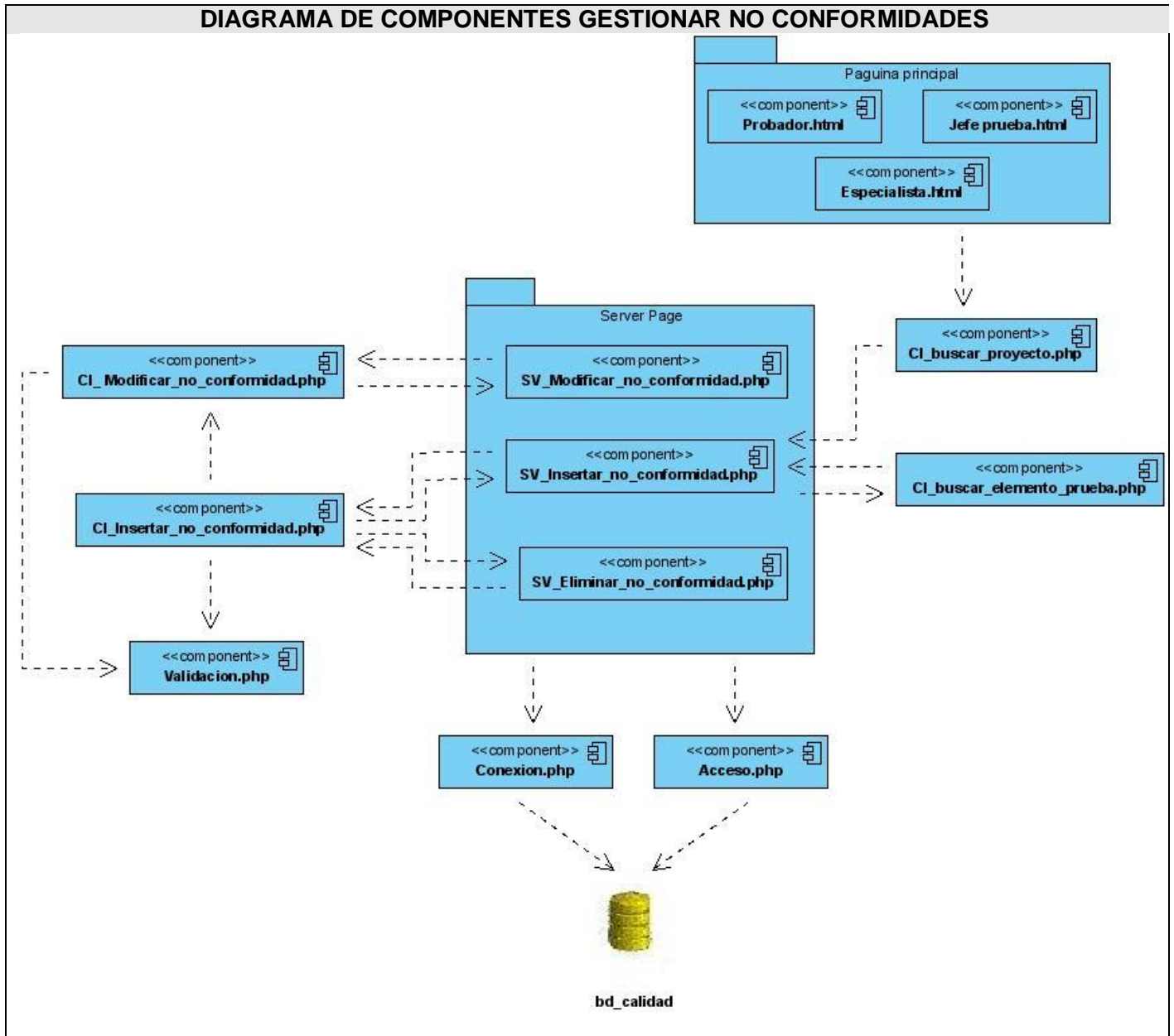


Figura 4.3 Diagrama de componentes Gestionar No Conformidades.

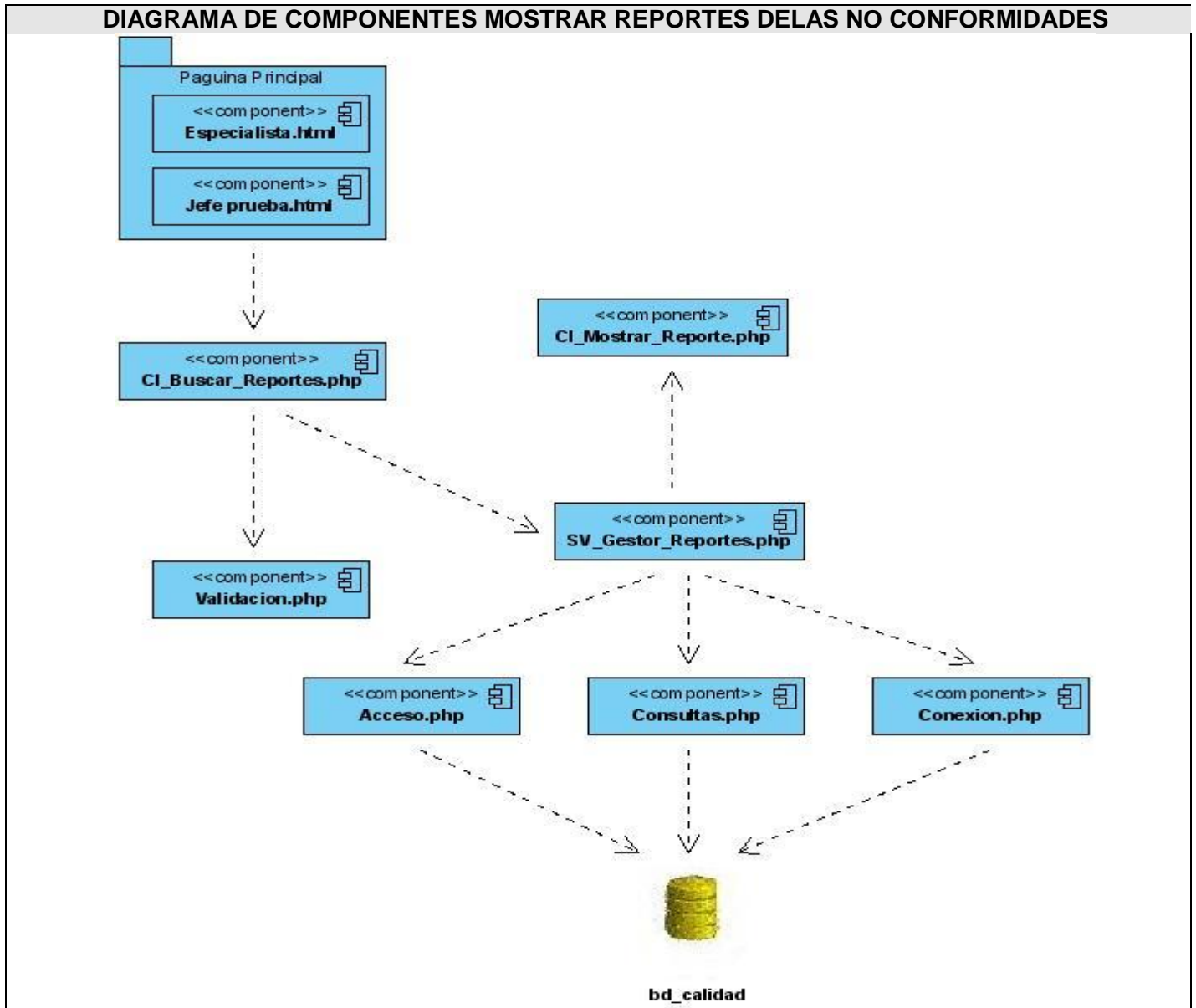


Figura 4.4 Diagrama de componentes Mostrar reportes de las No Conformidades.

4.2 Modelo de prueba.

Se representa el modelo de prueba propuesto para nuestro sistema en el [Anexo 3](#).

Conclusiones del capítulo.

En este capítulo se realizó toda la implementación del sistema y se mostraron los diagramas de despliegue y componentes, con los casos de pruebas elaborados se comprobó la integración de los módulos del sistema.

Conclusiones

- Con la definición del proceso anteriormente expuesto se garantizará una excelente gestión de las No Conformidades de un laboratorio de calidad.
- La confección de la herramienta informática, implementada a partir de la necesidad de automatizar los procesos que se llevan a cabo en el Laboratorio Central de Calidad de Software de la UCI, permitirá elevar la eficiencia en el proceso de revisión de los productos, posibilitará dar respuesta a las solicitudes de los clientes en el menor tiempo posible además se obtendrá como resultado la centralización de todas las No Conformidades de todas las pruebas realizadas a un software en la Universidad de Ciencias Informáticas.

Recomendaciones

- Continuar desarrollando la línea de investigación relacionada con la calidad de software; para ampliar y actualizar los conocimientos acordes con las últimas tendencias existentes a nivel mundial y poder así hacer más eficiente este proceso en la Universidad.
- Profundizar en el estudio de las tendencias y tecnologías actuales para determinar con mayor exactitud el basamento teórico de las nuevas herramientas a desarrollar.
- Aplicar esta herramienta en otros laboratorios de calidad de software con el objetivo de generalizar la propuesta para elevar la eficiencia de estos procesos y evaluar las mejoras posibles para una nueva versión.
- Diseñar a partir del sistema desarrollado y utilizando los datos que se almacenen durante un tiempo, un sistema inteligente que permita la generación automática de algunos datos y ayude a la toma de decisiones.

Referencia Bibliográfica.

- [1]. Computer dictionary, IEEE Std. 610. 1990.
- [2]. **G.J.Myers**. The art of software testing, John Wiley & sons. 1979.
- [3]. **ISO 9000**. 2000.
- [4]. Profitek. [Online] 02 01, 2007. <http://www.profitek.com.co/productos/signo.htm>.
- [5]. Microsoft. [Online] 02 01, 2007.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp>.
- [6]. Visual Paradigm. (2007, 2 16). Retrieved from www.visual-paradigm.com/product/vpuml/
- [7]. Desarrollo Desarrollo Web. (2007, 2 14). <http://www.desarrolloweb.com/articulos/332.php>.
- [8]. PHP. [Online] 2 17, 2007. <http://php.ciberaula.com/articulo/PHPoASP>.
- [9]. My Sql. [Online] 3 2 ,2007. www.mysql.com/products/mysqlcc/.
- [10]. Zona Clic. [Online] 10 4 ,2007. <http://clic.xtec.es/es/jclic/java.htm>.
- [11]. Desarrollo Web. [Online] 23 1 ,2007. <http://www.desarrolloweb.com/articulos/25.php>.
- [12]. Desarrollo Web. [Online] 3 5, 2007. <http://www.desarrolloweb.com/articulos/1598.php>.
- [13]. CSI. [Online] 3 2, 2007. <http://www.csi.map.es/csi/silice/Global71.html> .

Bibliografía

Computer dictionary, IEEE Std. 610. 1990.

CSI. [Online] 3 2, 2007. <http://www.csi.map.es/csi/silice/Global71.html> .

Desarrollo Desarrollo Web. (2007, 2 14). <http://www.desarrolloweb.com/articulos/332.php>.

Desarrollo Web. [Online] 23 1 ,2007. <http://www.desarrolloweb.com/articulos/25.php>.

Desarrollo Web. [Online] 3 5, 2007. <http://www.desarrolloweb.com/articulos/1598.php>.

El proceso Unificado de Desarrollo de Software. Grady Booch, James Rumbaugh Ivar Jacobson (enero 2007)

G.J.Myers. The art of software testing, John Wiley & sons. 1979.

ISO 9000. 2000.

Microsoft. [Online] 02 01, 2007.

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp>.

My Sql. [Online] 3 2 ,2007. www.mysql.com/products/mysqlcc/.

PHP. [Online] 2 17, 2007. <http://php.ciberaula.com/articulo/PHPoASP>.

Profitek. [Online] 02 01, 2007. <http://www.profitek.com.co/productos/signo.htm>.

Visual Paradigm. (2007, 2 16). Retrieved from www.visual-paradigm.com/product/vpuml/

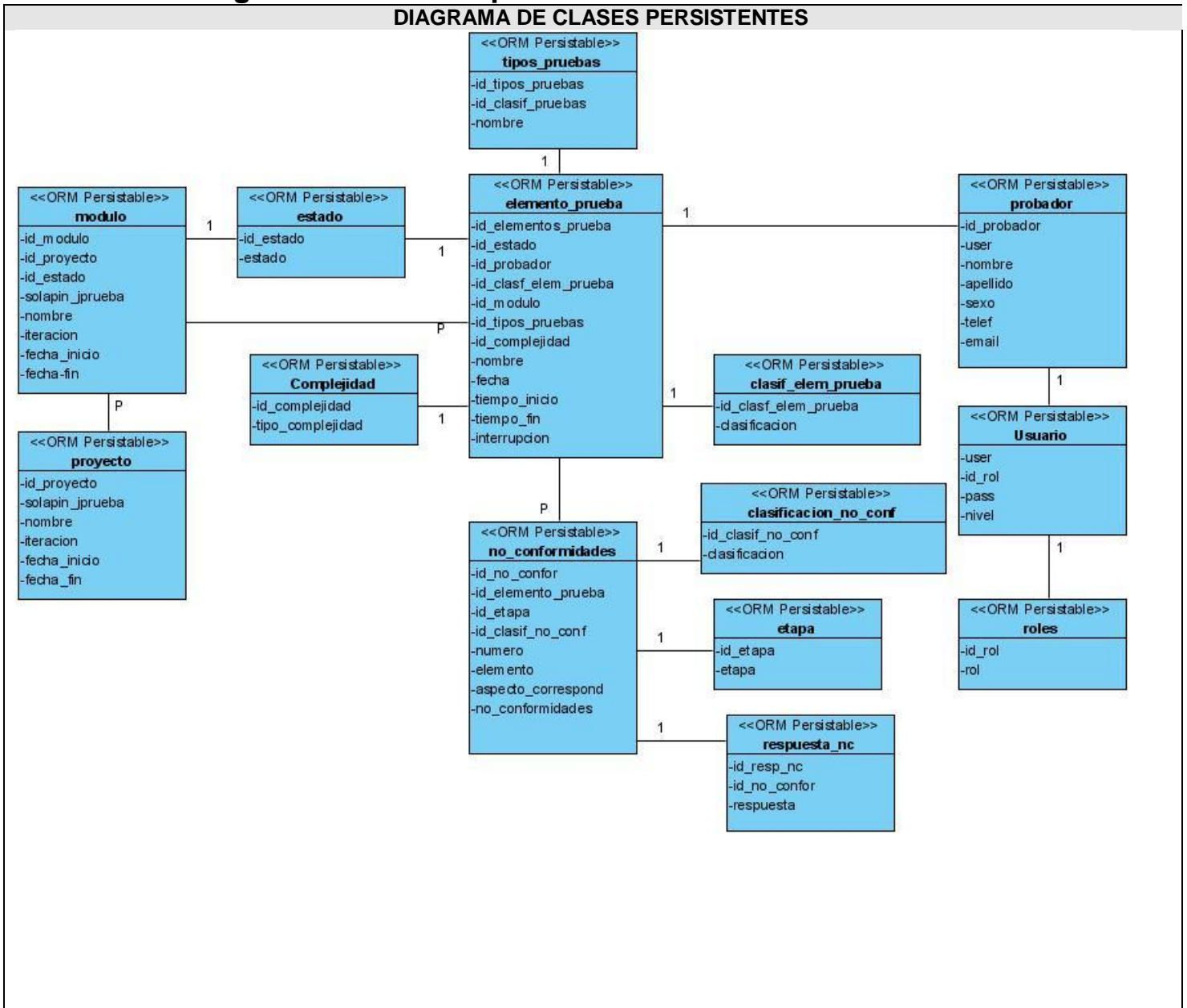
Zona Clic. [Online] 10 4 ,2007. <http://clic.xtec.es/es/jclic/java.htm>.

Glosario de Términos

- **CMMI**: Capability Maturity Model Integration. Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon.
- **CUN**: Casos de uso del Negocio.
- **CUS**: Casos de uso del Sistema.
- **IEEE**: Corresponde a las siglas de The Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros de telecomunicaciones, ingenieros electrónicos, Ingenieros en informática.
- **ISO**: Organización Internacional para la Estandarización (International Organization for Standardization). Es una organización que ha definido un conjunto de protocolos diferentes, llamados protocolos ISO/OSI. Esta organización de carácter voluntario fue fundada en 1946 y es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones.
- **RF**: Requisitos Funcionales.
- **RNF**: Requisitos No Funcionales.
- **Herramientas CASE**: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **SoftlaQ**: Software para la automatización de los procesos de Laboratorios de Calidad de Software:
 - Planificación y estimación de los recursos humanos y tecnológicos.
 - Gestiona de las No Conformidades.
 - Seguimiento de los recursos humanos y tecnológicos.

Anexos

Anexo 1: Diagramas de clases persistentes.



Anexo 2. Descripción de las Tablas de la Base de Datos.

Nombre: elemento_prueba		
Descripción: Se almacenan los datos correspondientes a los elementos de prueba que se van a revisar.		
Atributo	Tipo	Descripción
id_elemento_prueba	Integer	Es la llave primaria y almacena el identificador de un elemento de prueba.
id_clasif_elem_prueba	Integer	Es una llave extranjera y almacena el identificador de la clasificación de un elemento de prueba
id_probador	Integer	Es una llave extranjera y almacena el identificador que distingue al probador.
id_tipo_pruebas	Integer	Es una llave extranjera y almacena el identificador del tipo de prueba al que pertenece un elemento de prueba.
id_estado	Integer	Es una llave extranjera y almacena el identificador del estado en que se encuentra el elemento de prueba.
Id_Módulo	Integer	Es una llave extranjera y almacena el identificador del Módulo al que pertenece el elemento de prueba.
Id_complejidad	Integer	Es una llave extranjera y almacena el identificador de la complejidad que tiene el elemento de prueba.
nombre	String	Almacena el nombre que lleva el elemento de prueba.
fecha	String	Almacena la fecha en que se reviso el elemento de prueba.
tiempo_inicio	Integer	Almacena el tiempo en que se comienza a revisar el elemento de prueba.
tiempo_fin	Integer	Almacena el tiempo en que se termina de revisar el elemento de prueba.
interrupción	Integer	Almacena el tiempo de interrupción que puede tener un probador al estar revisando el elemento de prueba.

Tabla 1 Descripción de la tabla de la base de dato “elemento_prueba”

Nombre: No_conformidades		
Descripción: Se almacenan los datos correspondientes a las no conformidades encontradas en las revisiones.		
Atributo	Tipo	Descripción
id_no_confor	Char	Es la llave primaria y almacena el identificador de una no conformidad.
id_elemento_prueba	Integer	Es una llave extranjera y almacena el identificador de un elemento de prueba.
id_etapa	Integer	Es una llave extranjera y almacena el identificador de la etapa en que se encuentra la revisión.
id_clasif_no_conf	Integer	Es una llave extranjera y almacena el identificador de la clasificación de la no conformidad.
Número	Integer	Almacena el Número consecutivo de las no conformidades encontradas
elemento	String	Almacena el elemento de la no conformidad encontrada.
aspecto_correspond	String	Almacena el donde fue que se encontró la no conformidad.
no_conformidades	String	Almacena la no conformidad encontrada.

Tabla 2 Descripción de la tabla de la base de dato “No_conformidades”

Nombre: clasif_elem_prueba		
Descripción: Se almacena la clasificación de las no conformidades.		
Atributo	Tipo	Descripción
id_clasf_elem_prueba	Integer	Es la llave primaria y almacena el identificador de la clasificación de los elementos de prueba.
clasificación	String	Almacena la clasificación de un elemento de prueba.

Tabla 3 Descripción de la tabla de la base de dato “clasif_elem_prueba”

Nombre: clasificación_no_conf		
Descripción: Se almacena la clasificación de las no conformidades.		
Atributo	Tipo	Descripción
id_clasif_no_conf	Integer	Es la llave primaria y almacena el identificador de la clasificación de las no conformidades.
clasificación	Char	Almacena la clasificación de una no conformidad.

Tabla 4 Descripción de la tabla de la base de dato “clasificación_no_conf”

Nombre: Respuesta_NC		
Descripción: Se almacena las respuestas de las no conformidades encontradas.		
Atributo	Tipo	Descripción
id_resp_nc	Integer	Es la llave primaria y almacena el identificador de una respuesta a una no conformidad.
id_no_confor	Char	Es una llave extranjera y almacena el identificador de la no conformidad a la que le da respuesta.
respuesta	String	Almacena la respuesta de a las no conformidades.

Tabla 5 Descripción de la tabla de la base de dato “Respuesta_NC”

Nombre: etapas		
Descripción: Se almacena la etapa en que se encontró la no conformidad.		
Atributo	Tipo	Descripción
Id_etapa	Integer	Es la llave primaria y almacena el identificador de la etapa en que se encuentra una no conformidad.
Etapa	String	Almacena la etapa en que se encuentra una no conformidad.

Tabla 6 Descripción de la tabla de la base de dato “etapas”

Nombre: probador		
Descripción: Se almacena los datos que coinciden de los estudiantes y los trabajadores, los probadores del sistema.		
Atributo	Tipo	Descripción
Id_probador	Integer	Es llave primaria y almacena el identificador que distingue al probador.
User	Char	Es una llave extranjera y almacena el nombre de usuario relacionado con los probadores.
Nombre	String	Almacena el nombre del probador.
Apellidos	String	Almacena los apellidos del probador.
Sexo	Char	Almacena el sexo del probador.
Telef	String	Almacena el teléfono del probador.
email	String	Almacena el correo electrónico del probador.

Tabla 7 Descripción de la tabla de la base de dato “probador”

Nombre: estados		
Descripción: Se almacena el estado en que se encuentra los elementos de prueba y el Módulo.		
Atributo	Tipo	Descripción
id_estado	Integer	Es la llave primaria y almacena el identificador de un estado.
estado	String	Almacena el estado en que se encuentra un elemento de prueba y un Módulo.

Tabla 8 Descripción de la tabla de la base de dato “estados”

Nombre: Usuario		
Descripción: Se almacena los datos de los usuarios del sistema.		

Atributo	Tipo	Descripción
User	Char	Es la llave primaria y almacena el nombre de usuario relacionado con los probadores y jefes de pruebas.
Id_rols	Integer	Es una llave extranjera y almacena el identificador del rol que desempeña el usuario.
Pass	String	Almacena la contraseña que tiene el usuario.
Nivel	Integer	Almacena

Tabla 9 Descripción de la tabla de la base de dato "Usuario"

Nombre: roles		
Descripción: Se almacena el rol que desempeñan los usuarios del sistema.		
Atributo	Tipo	Descripción
Id_rol	Integer	Es la llave primaria y almacena el identificador de un rol que puede desempeñar un usuario.
rol	String	Almacena un rol que puede desempeñar un usuario del sistema.

Tabla 10 Descripción de la tabla de la base de dato "roles"

Nombre: tipos _ pruebas		
Descripción: En esta tabla se almacenan todos los tipos de prueba.		
Atributo	Tipo	Descripción
id_tipos_pruebas	Integer	Este campo es llave primaria y almacena el identificador que distingue a los tipos de prueba.
id_clasif_pruebas	Integer	Este campo es llave extranjera y almacena el identificador que distingue la clasificación de la pruebas.
nombre	Char	Este campo almacena los nombres de los tipos de pruebas.

Tabla 11 Descripción de la tabla de la base de dato “tipos _ pruebas”

Nombre: Proyecto		
Descripción: En esta tabla se almacena los datos del proyecto.		
Atributo	Tipo	Descripción
id _ proyecto	String	Este campo es llave primaria y almacena el identificador que distingue a los proyectos.
solapin_jprueba	Integer	Este campo es llave extranjera y almacena el número de Solapín del jefe de prueba.
nombre	String	Este campo almacena los nombres de los proyectos.
iteración	Integer	Este campo almacena las iteraciones de pruebas de un proyecto.
fecha _ inicio	String	Este campo almacena la fecha en que el proyecto entra en revisión.
fecha _ fin	String	Este campo almacena la fecha en que el proyecto termino de ser revisado.

Tabla 12 Descripción de la tabla de la base de dato “Proyecto”

Nombre: Módulo		
Descripción: En esta tabla se almacenan los datos de los módulos de un proyecto.		
Atributo	Tipo	Descripción
id _ Módulo	Integer	Este campo es llave primaria y almacena el identificador que distingue a los módulos.
id _ proyecto	String	Este campo es llave extranjera y almacena el identificador que distingue a los proyectos.
solapin_jprueba	Integer	Este campo es llave extranjera y almacena el número de Solapín del jefe de prueba.
id _ estado	Integer	Este campo es llave extranjera y almacena el

		identificador que distingue el estado en el que se encuentra un módulo.
nombre	String	Este campo almacena los nombres de los módulos.
iteración	Integer	Este campo almacena las iteraciones de pruebas de un módulo.
fecha _ inicio	String	Este campo almacena la fecha en que el Módulo entra en revisión.
fecha _ fin	String	Este campo almacena la fecha en que el Módulo termino de ser revisado.

Tabla 13 Descripción de la tabla de la base de dato “Módulo”

Anexo 3. Modelo de Caso de Prueba.

Módulo Planificación.

Entrada	Resultados	Condiciones
<ul style="list-style-type: none"> • Se escoge el Jefe de prueba. • Se escoge el proyecto. 	<ul style="list-style-type: none"> • Se muestra una interfaz con el nombre y apellidos del jefe de prueba seleccionados y muestra el listado de los módulos del proyecto. 	<ul style="list-style-type: none"> • En el módulo de Seguimiento de debió haber insertado con anterioridad el jefe de prueba. • En el módulo de Planificación se debió haber insertado con anterioridad el proyecto.
<ul style="list-style-type: none"> • Se selecciona la prueba a planificar. 	<ul style="list-style-type: none"> • Aparece una interfaz donde se insertan los recursos a planificar. 	<ul style="list-style-type: none"> • En el módulo de Planificación se debió haber insertado con anterioridad la prueba a planificar.
<ul style="list-style-type: none"> • Se selecciona insertar una prueba. 	<ul style="list-style-type: none"> • El sistema inserta nuevas pruebas. 	<ul style="list-style-type: none"> • Se debe introducir en el módulo de planificación: <ul style="list-style-type: none"> • Nombre de la prueba • Clasificación

		<ul style="list-style-type: none"> Métricas asociadas a la prueba.
<ul style="list-style-type: none"> Se selecciona insertar un proyecto. 	<ul style="list-style-type: none"> El sistema inserta nuevos proyectos y muestra la interfaz para insertar sus módulos. 	<ul style="list-style-type: none"> Se debe introducir en el módulo de planificación: <ul style="list-style-type: none"> Nombre del proyecto Iteración del proyecto. Fecha Jefe de Prueba.
<ul style="list-style-type: none"> Un usuario se autentica por el Rol de Especialista. 	<ul style="list-style-type: none"> Se muestra la interfaz del especialista con sus permisos. <p><i>Planificación:</i></p> <ul style="list-style-type: none"> Gestionar Datos del Proyecto. Gestionar Métricas. Gestionar Planificaciones. Gestionar Pruebas Insertar Pruebas. <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> Mostrar estados de las pruebas. Gestionar las No Conformidades. Datos de los elementos. Reporte de todos los proyectos. <p><i>Seguimiento:</i></p>	<ul style="list-style-type: none"> El administrador debió haber asignado el rol de especialista al usuario.

	<ul style="list-style-type: none"> • Gestionar datos de Jefe de prueba. • Generar reportes. • Gestionar Cursos optativos. • Verificar cumplimiento de métricas. 	
<ul style="list-style-type: none"> • Un usuario se autentica por el Rol de Jefe de Prueba. 	<ul style="list-style-type: none"> • Se muestra la interfaz del Jefe de prueba con sus permisos. <p><i>Planificación:</i></p> <ul style="list-style-type: none"> • Gestionar Datos del Proyecto. • Gestionar Planificaciones. • Insertar datos del proyecto. • Planificar Pruebas. <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> • Asignar trabajo. • Mostrar estado de las pruebas. • Gestionar las No Conformidades. • Insertar nuevo elemento de prueba. • Datos de los elementos de prueba. • Reportes de los proyectos en los que ha trabajado. <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> • Gestionar datos de Probador. 	<ul style="list-style-type: none"> • El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.

	<ul style="list-style-type: none"> • Evaluar Probador. • Gestionar PC 	
--	---	--

Tabla 14 Modelo de caso de prueba “Módulo Planificación”

No conformidades.

Entrada	Resultados	Condiciones
• Seleccionar proyecto.	• Se muestra la lista de módulos correspondientes al proyecto.	• Se debe de haber insertado el proyecto en el modulo de planificación.
• Se selecciona el módulo al cual se le van a hacer las pruebas.	• Se muestra la interfaz para insertar las no conformidades del modulo probado.	• Se debe de haber insertado los elementos de prueba en el modulo de No Conformidades.
• Se selecciona insertar elementos de prueba.	• Se muestra una interfaz para escoger el modulo al cual se le van a insertar los elementos de prueba.	• Se debe de haber insertado el proyecto en el módulo de planificación.
• Se selecciona asignar trabajo.	• Se muestra una interfaz con los probadores y elementos de prueba.	• Se deben de haber insertado los probadores en el módulo de Seguimiento y los elementos de prueba en el módulo No Conformidades.
• Se selecciona insertar No Conformidades.	• Se muestra la interfaz para insertar las no conformidades.	• Se debe de haber insertado el proyecto en el módulo de planificación y los elementos de prueba en el módulo de No Conformidades.
• Se selecciona insertar respuestas de las no conformidades.	• Se muestra la interfaz para insertar respuesta a las no conformidades.	• Se debe de haber insertado no conformidades al sistema en el módulo No Conformidades.
• Un usuario se autentica por el Rol de Especialista.	• Se muestra la interfaz del especialista con sus permisos.	• El administrador debió haber asignado el rol de especialista al

	<ul style="list-style-type: none"> • <p><i>Planificación:</i></p> <ul style="list-style-type: none"> • Gestionar Datos del Proyecto. • Gestionar Métricas. • Gestionar Planificaciones. • Gestionar Pruebas • Insertar Pruebas. <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> • Mostrar estados de las pruebas. • Gestionar las No Conformidades. • Datos de los elementos. • Reporte de todos los proyectos. <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> • Gestionar datos de Jefe de prueba. • Generar reportes. • Gestionar Cursos optativos. • Verificar cumplimiento de métricas. 	<p>usuario.</p>
<ul style="list-style-type: none"> • Un usuario se autentica por el Rol de Jefe de Prueba. 	<ul style="list-style-type: none"> • Se muestra la interfaz del Jefe de prueba con sus permisos. <p><i>Planificación:</i></p> <ul style="list-style-type: none"> • Gestionar Datos del 	<ul style="list-style-type: none"> • El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.

	<p>Proyecto.</p> <ul style="list-style-type: none"> • Gestionar Planificaciones. • Insertar datos del proyecto. • Planificar Pruebas. <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> • Asignar trabajo. • Mostrar estado de las pruebas. • Gestionar las No Conformidades. • Insertar nuevo elemento de prueba. • Datos de los elementos de prueba. • Reportes de los proyectos en los que ha trabajado. <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> • Gestionar datos de Probador. • Evaluar Probador. • Gestionar PC. 	
--	---	--

Tabla 15 Modelo de caso de prueba “No conformidades”

Seguimiento.

Entrada	Resultados	Condiciones
<ul style="list-style-type: none"> • Se selecciona insertar Probadores. 	<ul style="list-style-type: none"> • El sistema inserta nuevos probadores. 	<ul style="list-style-type: none"> • Se debe introducir en el módulo de Seguimiento los siguientes datos: <ul style="list-style-type: none"> • Se debe introducir en el módulo de Seguimiento.

		<ul style="list-style-type: none"> • Nombre • Apellidos • Sexo • Teléfono • Correo • Apartamento • Facultad • Grupo • Solapín
<p>•Se selecciona insertar un Jefe de prueba.</p>	<p>•El sistema inserta nuevos Jefes de prueba.</p>	<p>•Se debe introducir en el módulo de Seguimiento los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre • Apellidos • Departamento • Título • Grado Científico • Facultad • Experiencia • Dirección • Teléfono • Correo • Desempeño laboral • Solapín
<p>•Se selecciona verificar la veracidad de las métricas aplicadas a una prueba.</p>	<p>•Muestra las métricas aplicadas y el tiempo real en que se realizaron las pruebas.</p>	<p>•Se debe de haber insertado en el modulo planificación las pruebas con sus métricas y el sistema debió de haber registrado el tiempo real de las pruebas en el módulo No Conformidades.</p>

<ul style="list-style-type: none"> •Se selecciona evaluar probador. 	<ul style="list-style-type: none"> •Muestra la interfaz para evaluar el probador. 	<ul style="list-style-type: none"> •Se debe de haber insertado el probador en el modulo de Seguimiento y revisar el trabajo que hizo en el modulo No Conformidades.
<p>Se selecciona gestionar PC.</p>	<p>Muestra la interfaz para seleccionar la planificación.</p>	<p>Se debe de haber realizado planificación en el modulo Planificación.</p>
<ul style="list-style-type: none"> •Un usuario se autentica por el Rol de Especialista. 	<ul style="list-style-type: none"> •Se muestra la interfaz del especialista con sus permisos. <p><i>Planificación:</i></p> <ul style="list-style-type: none"> • Gestionar Datos del Proyecto. • Gestionar Métricas. • Gestionar Planificaciones. • Gestionar Pruebas • Insertar Pruebas. <p><i>No Conformidades:</i></p> <ul style="list-style-type: none"> • Mostrar estados de las pruebas. • Gestionar las No Conformidades. • Datos de los elementos. • Reporte de todos los proyectos. <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> • Gestionar datos de Jefe de prueba. • Generar reportes. • Gestionar Cursos optativos. 	<ul style="list-style-type: none"> •El administrador debió haber asignado el rol de especialista al usuario.

	<ul style="list-style-type: none"> • Verificar cumplimiento de métricas. 	
<ul style="list-style-type: none"> •Un usuario se autentica por el Rol de Jefe de Prueba. 	<ul style="list-style-type: none"> •Se muestra la interfaz del Jefe de prueba con sus permisos. <p><i>Planificación:</i></p> <ul style="list-style-type: none"> • Gestionar Datos del Proyecto. • Gestionar Planificaciones. • Insertar datos del proyecto. • Planificar Pruebas. <p><i>No conformidades:</i></p> <ul style="list-style-type: none"> • Asignar trabajo. • Mostrar estado de las pruebas. • Gestionar las No Conformidades. • Insertar nuevo elemento de prueba. • Datos de los elementos de prueba. • Reportes de los proyectos en los que ha trabajado. <p><i>Seguimiento:</i></p> <ul style="list-style-type: none"> • Gestionar datos de Probador. 	<ul style="list-style-type: none"> •El administrador o el Especialista debieron haber asignado el rol de Jefe de prueba al usuario.

	<ul style="list-style-type: none">• Evaluar Probador.• Gestionar PC.	
--	---	--

Tabla 16 Modelo de caso de prueba "Seguimiento"