

Universidad de las Ciencias Informáticas
Facultad 7



Título: Sistema de Gestión de No conformidades.

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

Autores: Ariel Alvarez Castillo.

Yania Cedeño Oliva.

Tutores: Ing. Yadainy Betancourt Rodríguez

Ciudad de la Habana

Junio 2009

AGRADECIMIENTOS

Puede que este sea el momento más delicado de la tesis, nada se hace tan difícil como este instante. Primero quiero agradecer al Comandante en Jefe pues gracias a él y a la Revolución mi sueño de ser ingeniera hoy está bien cerca. Agradecer a toda mi familia, en especial a mi abuela que es mi vida, y que aunque no tuvo la oportunidad de estudiar, ve sus sueños realizados a través de mi, a mi mamá por ser mi mayor cómplice y la persona en quien más confío, a mi pipo a quien adoro y aunque a veces es tan exigente siempre quiere lo mejor para mi. A mis tíos y tías, en especial a Papi y Ule por ser las personas que quisiera ser. A Javier por ayudarme y quererme siempre y por estar cuando mis días se tornan oscuros. A mis amigos, en especial a Yei, Alexei, a Gretty, a Arelys, a Duni, a Romel y a la Burbuja a Nadie, Rubén, Adniel y Ania, a todos por soportarme y estar cuando los necesito. A todas las personas que en algún momento me han ayudado y a las que han marcado mi vida. Y no por último menos importante muchas gracias a Ariel, por ser no sólo mi CT sino mi ambulancia, mi camaroncito duro, mi apoyo y más importante que todo, por ser mi amigo.

A todos gracias

Yania

Agradecer a toda mi familia, en especial a mi abuela por haberme guiado por el camino correcto y estar siempre a mi lado, por su amor, su ternura y sus consejos. A mi abuelo por apoyarme siempre y estar ahí cuando lo necesito. A mis padres, por su preocupación, y dedicación. A Fidel, gracias por hacer realidad mis sueños. A Yuliet por escucharme y quererme como soy. A mis amigos, en especial a Romel, Dunieski, Osvaldo, Urko, a mis compañeros de grupo, a todo el proyecto, en especial a mi tutora y amiga Ydainy, felicidades por esa hermosa niña que pronto nacerá. Agradecer a todas esas personas que de una forma u otra hicieron posible la culminación de este trabajo, y que me ayudaron a lo largo de mi carrera. Gracias a mi compañera de tesis, Yania, por ser mi motor impulsor, por su apoyo y soportar mis malacrianzas, por pensar en mi cuando necesita algo, gracias por ser mí amiga.

Gracias a todos.

Ariel

DEDICATORIA

A mi familia, en especial a mi hermana para quien siempre quisiera ser un ejemplo.

A Javier por el amor y la paciencia.

A Fidel por la oportunidad de estudiar.

A mis amigos por el cariño de cada día.

A mis maestros por la enseñanza.

A toda la gente linda que nos ayudó en cada detalle, en especial a Yadainy.

Muy especial a Ariel por soportarme.

Yania

A mi familia, especialmente a mis abuelos.

A Yuliet por su amor.

A mis amigos.

A Yadainy.

A mis compañeros de grupo.

Y por supuesto a Yania por apoyarme en todo.

Ariel

RESUMEN

Con el objetivo de que los sistemas desarrollados en la Unidad de Compatibilización, Integración y Desarrollo tengan la calidad requerida el Centro de Calidad Estándares y Seguridad realiza diferentes pruebas a los componentes que genera cada proyecto que se desarrolla. Este proceso de pruebas no es óptimo debido a una serie de factores que influyen en él, por ello el siguiente trabajo pretende brindar una solución a la situación problemática que se enfrenta, a través de la informatización de los procesos relacionados a la gestión de las No conformidades, para desglosar el objetivo general fueron definidos una serie de objetivos y tareas, entre los que se encuentra la implementación de una herramienta que permita la gestión de los recursos del laboratorio así como los elementos a verificar y la ejecución de la prueba en sí, para de esta manera optimizar los procedimientos asociados.

Palabras claves:

Sistema, Gestión, No conformidades.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Sistemas de gestión de No conformidades existentes.....	5
1.3. Tendencias y tecnologías actuales	6
1.3.2. Lenguajes de programación para la Web	8
1.3.3. Lenguajes del lado del servidor:	9
1.3.4. Fundamentación del lenguaje de programación a utilizar.....	11
1.3.5. Servidores WEB	12
1.3.6. Fundamentación del servidor Web a utilizar	13
1.3.7. Sistema de Gestión de Base de Datos (SGBD).....	13
1.3.8. Fundamentación de la selección de la Base de Datos a utilizar	15
1.3.9. Marco de trabajo.....	16
1.3.10. Metodologías de desarrollo del software	17
1.4. Herramientas utilizadas	19
1.5. Conclusiones	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	22
2.1. Introducción	22
2.2. Objeto de estudio.....	22
2.3. Descripción de los procesos del negocio	22
2.3.1. Proceso inicial.....	22
2.3.2. Proceso de Gestión de No conformidades	23
2.4. Análisis crítico del proceso actual.....	26
2.5. Procesos objeto de automatización	27
2.6. Información que se maneja.....	27
2.7. Modelo del negocio.....	27
2.8. Actores del negocio	27
2.9. Trabajadores del negocio	28
2.10. Diagrama de casos de uso del negocio	28
2.11. Descripción de los casos de uso del negocio	29
2.12. Especificación de los requisitos de software.....	31
2.12.1. Requerimientos funcionales del sistema	31
2.12.2. Requerimientos no funcionales del sistema	34
2.13. Modelo de casos de usos del sistema	35
2.14. Definición de los actores	35
2.15. Listado de casos de uso	36

2.16.	Diagrama de casos de uso del sistema.	38
2.17.	Conclusiones.	41
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		42
3.1.	Introducción	42
3.2.	Análisis	42
3.2.1.	Diagramas de clases del análisis.....	42
3.3.	Diseño.....	43
3.3.1.	Diagrama de clases del diseño web	43
3.3.2.	Diagramas de secuencia	44
3.3.3.	Diagrama de clases persistentes.....	45
3.4.	Conclusiones	46
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA		47
4.1.	Introducción	47
4.2.	Implementación.....	47
4.2.1.	Modelo de datos	48
4.2.2.	Modelo de despliegue.....	49
4.2.3.	Ambiente de implantación.....	49
4.2.4.	Diagrama de componentes.....	50
4.3.	Prueba	52
4.3.1.	Diseños de caso de prueba	53
	DCP Gestionar proyecto.....	54
	Gestionar componente	56
	DCP Gestionar tipo de prueba.....	57
4.4.	Conclusiones	65
CONCLUSIONES.....		66
RECOMENDACIONES.....		67
BIBLIOGRAFÍA.....		68
REFERENCIAS BIBLIOGRÁFICAS		71
ANEXOS.....		73
5.1.	Anexos del Capítulo 2 “Especificación textual de los CU”	73
5.1.1.	Especificación textual del CU “Gestionar línea.”	73
5.1.2.	Especificación textual del CU “Gestionar proyecto.”	74

5.1.3.	Especificación textual del CU “Gestionar componente.”	76
5.1.4.	Especificación textual del CU “Gestionar tipo de prueba.”	78
5.1.5.	Especificación textual del CU “Gestionar elemento.”	79
5.1.6.	Especificación textual del CU “Gestionar probador.”	81
5.1.7.	Especificación textual del CU “Gestionar trabajo.”	83
5.1.8.	Especificación textual del CU “Generar parte.”	85
5.2.	Anexos del Capítulo 3 “Diagrama de clases del análisis”	86
5.2.1.	Diagrama de clases del análisis “Gestionar línea.”	86
5.2.2.	Diagrama de clases del análisis “Gestionar proyecto.”	86
5.2.3.	Diagrama de clases del análisis “Gestionar componente.”	87
5.2.4.	Diagrama de clases del análisis “Gestionar tipo de prueba.”	87
5.2.5.	Diagrama de clases del análisis “Gestionar elemento.”	88
5.2.6.	Diagrama de clases del análisis “Gestionar probador.”	88
5.2.7.	Diagrama de clases del análisis “Gestionar trabajo.”	89
5.2.8.	Diagrama de clases del análisis “Generar partes.”	89
5.3.	Anexos del Capítulo 3 “Diagrama de clases del diseño web”	90
5.3.1.	Diagrama de clases del diseño web “Gestionar línea.”	90
5.3.2.	Diagrama de clases del diseño web “Gestionar proyecto.”	90
5.3.3.	Diagrama de clases del diseño web “Gestionar componente.”	91
5.3.4.	Diagrama de clases del diseño web “Gestionar tipo de prueba.”	91
5.3.5.	Diagrama de clases del diseño web “Gestionar elemento.”	92
5.3.6.	Diagrama de clases del diseño web “Gestionar probador.”	92
5.3.7.	Diagrama de clases del diseño web “Gestionar trabajo.”	93
5.3.8.	Diagrama de clases del diseño web “Generar parte.”	93
5.4.	Anexos del Capítulo 3 “Diagrama de secuencia.”	94
5.4.1.	Diagrama de secuencia “Adicionar línea.”	94
5.4.2.	Diagrama de secuencia “Adicionar proyecto.”	94
5.4.3.	Diagrama de secuencia “Adicionar componente.”	95
5.4.4.	Diagrama de secuencia “Adicionar tipo de prueba.”	95
5.4.5.	Diagrama de secuencia “Adicionar elemento.”	96
5.4.6.	Diagrama de secuencia “Adicionar probador.”	96
5.4.7.	Diagrama de secuencia “Adicionar trabajo.”	97
5.4.8.	Diagrama de secuencia “Adicionar parte.”	97

GLOSARIO	99
-----------------	-----------

INTRODUCCIÓN

En la sociedad moderna el uso de la tecnología se ha convertido en parte fundamental de la vida cotidiana. Teléfonos celulares, robots industriales, satélites, fibra óptica y otros adelantos científico - técnicos permiten que problemas anteriormente complicados y engorrosos puedan solucionarse en poco tiempo y con gran eficiencia. Entre los más significativos avances del último siglo, la informática ocupa sin duda un lugar cimero. Con esta tecnología el hombre puede procesar un cúmulo de información de todo tipo, desde imágenes y videos hasta grandes algoritmos matemáticos. Debido a su rapidez y precisión, la aplicación de la informática se ha vuelto imprescindible para los más diversos sectores en todo el mundo.

Cuba no está exenta a este fenómeno, aún siendo una nación pequeña que por más de cinco décadas ha sido bloqueada por el imperio más cruel conocido en la historia de la humanidad. En su empeño por avanzar y alcanzar un mayor desarrollo tecnológico, el país se encuentra inmerso en un proceso de informatización de la sociedad que permitirá, en sectores tan importantes como la salud, la educación, el comercio e incluso la defensa, aumentar el nivel de vida de los cubanos. El desarrollo de productos informáticos es parte esencial para la automatización de procesos, anteriormente los software tenían que ser comprados en mercados extranjeros, acción que resultaba costosa por la influencia del bloqueo y algunos de estos productos después de adquiridos no se adecuaban a las particularidades de los diferentes renglones de la economía nacional o solo se explotaban en un porcentaje muy bajo, es por ello que resulta más económico desarrollar la industria nacional de software soportada por instituciones especializadas en la creación de aplicaciones informáticas.

Entre las principales instituciones fundadas para apoyar la creación de software, la Universidad de las Ciencias Informáticas (UCI) juega un papel decisivo ya que tiene la responsabilidad de formar profesionales preparados en la rama de la informática. Dentro de sus instalaciones se encuentra la Unidad de Compatibilización, Integración y Desarrollo (UCID), compuesta por profesionales y estudiantes que desarrollan aplicaciones para las Fuerzas Armadas Revolucionarias (FAR), posibilitando la informatización de este sector. Garantizar un rendimiento eficiente y la calidad de los productos desarrollados en este centro, constituyen premisas de esta organización.

Ello se evidencia en la existencia dentro de su estructura organizacional, de un centro que controla la calidad de los sistemas que se desarrollan en la UCID, el Centro de Calidad Estándares y Seguridad

Sistema de Gestión de No conformidades

(CCES), que se divide en cuatro grupos esenciales para su funcionamiento. Uno de ellos es el Laboratorio de Prueba, responsable de la revisión de los artefactos entregables al cliente, de verificar el cumplimiento de los requisitos acordados con el usuario y de comprobar la funcionalidad del producto en las pruebas que se realizan a la aplicación, mediante la utilización de listas de chequeos, estándares entre otras técnicas para verificar el software en condiciones extremas.

Durante las pruebas de liberación de algún componente, se chequean dos elementos principales, la documentación y la aplicación. Dentro de la primera se incluyen los Diseños de casos de uso (DCU) o Descripciones de requisitos y los Diseños de caso de prueba (DCP), plantillas que recogen en una serie de secciones la información necesaria para realizar las pruebas, igualmente se contemplan los manuales que sirven de guía al usuario para la realización de las diferentes operaciones con el software. Por su parte la aplicación contiene el código fuente estructurado de manera funcional y lógica, organizado arquitectónicamente. Una vez que son revisados estos elementos se generan un grupo de No conformidades (NC), que evidencian incumplimiento de algún requisito o estándar previamente establecido, quedando contabilizadas y clasificadas según su tipo y estado.

Durante las pruebas en el Laboratorio se han detectado una serie de problemas que no permiten la optimización del trabajo en el mismo. En el momento de asignar las tareas a los probadores no se tiene una manera eficiente de conocer la distribución realizada, provocando que la asignación no sea equitativa y ello trae consigo que el probador no aproveche mejor su tiempo por desconocer la carga de trabajo para el día. Una vez distribuidas las misiones a los probadores comienza la detección de las No conformidades, siendo propenso a errores humanos su conteo y clasificación. Por otra parte, la plantilla donde se recogen las NC no es óptima ya que tiene varios datos que se deben entrar manualmente y algunos deben repetirse en más de una ocasión. Los probadores a la hora de entregar el trabajo pueden cometer errores al colocar el mismo en las carpetas correspondientes del servidor. El especialista, no gestiona de manera eficiente su tiempo pues no puede trabajar a la par de los probadores, debe esperar a que estos terminen. Al concluir de revisar las plantillas entregadas, el especialista procede a elaborar un reporte del trabajo realizado, en la elaboración del mismo se invierte mucho tiempo contabilizando y clasificando los resultados del trabajo, igualmente resulta complicado obtener el reporte de las pruebas realizadas en una fecha anterior, ya que se dificulta la búsqueda de la información y se torna engorrosa y monótona. Otro

Sistema de Gestión de No conformidades

problema es que no existe una comunicación efectiva entre el Equipo de desarrollo y el Laboratorio de prueba pues no es posible conocer en tiempo real el estado de las No conformidades.

Debido a los problemas anteriormente reflejados se hace necesario buscar una solución para dar respuesta al **problema a resolver**:

¿Cómo gestionar el trabajo de los especialistas y probadores, la revisión, organización y contabilización de las No conformidades generadas durante las pruebas, así como la comunicación entre el Laboratorio de prueba y el equipo de desarrollo?

Se definen como **objeto de estudio** los procesos asociados a la gestión de las No conformidades, enmarcando el **campo de acción** en el proceso de gestión de las No conformidades del laboratorio de prueba del CCES, teniendo como **objetivo general** desarrollar un sistema para la gestión de los procesos asociados a las No conformidades en el laboratorio de pruebas.

Se proponen como **objetivos específicos**:

- Definir y fundamentar los elementos teóricos para el análisis del problema a resolver.
- Desarrollar el sistema para dar respuesta a las necesidades del Laboratorio de prueba del CCES.
- Caracterizar el ambiente de implantación para el correcto funcionamiento de la aplicación.
- Validar la propuesta del sistema.

Para dar cumplimiento a los objetivos planteados se hace necesaria la realización de las siguientes **tareas de la investigación**:

- Definir las tecnologías y herramientas para la implementación de la aplicación.
- Desarrollar las disciplinas de Modelamiento del Negocio, Gestión de Requerimientos, Análisis y Diseño, Implementación de la Aplicación, así como los diferentes niveles de pruebas que se llevan a cabo en cada disciplina.
- Probar la aplicación una vez concluida.
- Presentar una serie de recomendaciones para el futuro perfeccionamiento del sistema.

Sistema de Gestión de No conformidades

El contenido del Trabajo de Diploma está estructurado en los siguientes capítulos:

Capítulo 1. Fundamentación Teórica: En el presente capítulo se realiza un estudio del estado del arte del tema. Se exponen las características esenciales de los principales lenguajes de programación empleados en el desarrollo de aplicaciones Web y en ese mismo sentido se analizan las particularidades y funcionalidades de los posibles gestores de bases de datos para el desarrollo del sistema. Se seleccionan las herramientas más eficientes a emplear en la solución del problema propuesto.

Capítulo 2. Características del sistema: Se realiza la propuesta de las características que presentará la aplicación. Por otra parte se confecciona el modelo del negocio, los casos de uso del sistema así como la definición de requisitos funcionales y no funcionales que debe cumplir el mismo.

Capítulo 3. Análisis y diseño del sistema: Se plasma la elaboración de la propuesta de solución. Se modelan diagramas de clases, de datos, y se plantean los principios de diseño que se seguirán en la implementación de la aplicación.

Capítulo 4. Implementación y prueba del sistema: En el capítulo se elaboran los artefactos necesarios para desarrollar el proceso de implementación. Se modelan el diagrama de despliegue y el diagrama de componentes donde se representan los nodos en los que se distribuye la aplicación. Así como la descripción sobre el desarrollo de las pruebas al producto con el objetivo de entregar un software de alta calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo figura una breve ilustración de algunos de los sistemas automatizados que existen vinculados al campo de acción. Se recoge una descripción de las tecnologías actuales relacionadas con el objeto de estudio. Así como las herramientas y metodologías que se utilizarán para la realización del sistema propuesto.

1.2. Sistemas de gestión de No conformidades existentes

Después de realizar una investigación sobre los sistemas de gestión en la universidad y en el mundo se encontraron un grupo de aplicaciones que informatizan algunos de los procesos afines al campo de acción, entre los cuales se encuentran el Quarta Non Conformity, el GestConForm, KMKey Quality y el sistema Pruebas de Liberación Laboratorio UCI. A continuación una breve explicación de los aspectos más relevantes de cada uno.

KMKey Quality: Es un sistema para la Gestión de la calidad con un módulo específico para la gestión de las No conformidades, basado en entorno web, los usuarios con niveles de acceso específico pueden insertar las No conformidades así como la acción correctiva o preventiva para solucionar las mismas, facilitando la asignación de las tareas además del plazo de tiempo para la solución y un lugar donde se recogen las actividades que se realizan. Es usado en empresas españolas y por su fácil adaptabilidad es instalado en diversos sectores como Estaciones de radio, cajas de ahorro y empresas de consultoría.

Es un sistema con amplias facilidades para su uso pero no se adapta a las necesidades del CCES, es un software propietario cuyos requerimientos de soporte no se adecúan a los establecidos en la UCID siendo imposible su compatibilización.

GestConForm: Este sistema permite un control de las actividades desarrolladas en la detección de las No conformidades así como también la documentación de las acciones preventivas y correctivas referentes a la No conformidad. El responsable de calidad de la empresa puede acceder a la información de manera rápida y eficiente, facilitando la emisión de partes en los que se puede conocer gran cantidad de información, entre ella el estado actual de la no conformidad.

Pese a las ventajas que posee no es aplicable en el proceso de pruebas pues no cuenta con una Base de Datos con la robustez necesaria para soportar el proceso de prueba del Laboratorio, no se corresponde con la arquitectura definida por la UCID y no es un sistema multiplataforma.

Quarta Non Conformity: Sistema que proporciona facilidades en la planificación y detección de las No conformidades, brindando un análisis exhaustivo de estas permitiendo hacer una entrada de los requisitos iniciales y posteriormente al detectarse la no conformidad puede exponer el requisito que fue incumplido. Es un sistema en ambiente web, de fácil adaptación a cualquier empresa, con funcionalidades configurables, tiene la facilidad de buscar No conformidades similares que ya se hayan detectado, permite la configuración del flujo de trabajo y de la no conformidad mediante un catálogo con varios niveles.

A pesar de las facilidades de esta herramienta y de las ventajas que brinda, no es aplicable el Laboratorio de prueba porque es software propietario, por otro lado está diseñado para elementos medibles y concretos, no para componentes abstractos como los que se generan en el proceso de producción de software.

Prueba de Liberación Laboratorio: Este sistema dispone de un módulo de gestión de los recursos humanos que participan en la realización de las pruebas, además permite conocer el estado de las No conformidades y facilita el intercambio entre las parte interesadas en la realización de la prueba.

Es una herramienta creada en la universidad para la gestión de las No conformidades, no es efectiva para ser usada en la informatización del Laboratorio de prueba, ya que el proceso interno que este realiza tiene elementos muy propios que no se adaptan a las particularidades de dicha herramienta, por otra parte la arquitectura no se corresponde con la adoptada por la UCID y no es posible compatibilizarla con el marco de trabajo en el cual se desarrolla actualmente.

Por todos los elementos expuestos es que se decide la implementación de una herramienta que se adecúe a las características propias del Laboratorio de pruebas del CCES, que integre los procesos de gestión de No conformidades y seguimiento al mismo tiempo, con facilidades de asignación de tareas y generación de reportes.

1.3. Tendencias y tecnologías actuales

Teniendo en cuenta las necesidades en el Laboratorio de prueba y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tendencias actuales. A continuación se

exponen diferentes tecnologías, herramientas, lenguajes y metodologías usadas en el desarrollo de la aplicación.

1.3.1. Arquitectura Cliente / Servidor

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura.

Esta arquitectura consiste básicamente en un programa (el cliente) que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa realizarla en una red. [1]

En esta arquitectura la capacidad de procesamiento está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. [1]

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. [1]

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

Ventajas:

El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma. [2]

Un servidor da servicio a múltiples clientes en forma concurrente. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores, ya

sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final. [2]

Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un sólo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un sólo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario.

1.3.2. Lenguajes de programación para la Web

Teniendo en cuenta la arquitectura Cliente/Servidor los lenguajes de programación para las aplicaciones Web se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor.

Lenguajes del lado del cliente:

- **HTML**

HTML, siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes, también puede describir hasta cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. [3]

- **Java Script**

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, es muy utilizado principalmente en páginas web. Permite interactuar con el navegador de manera dinámica y eficaz. Sus requerimientos son relativamente sencillos, es un lenguaje cuyos códigos se interpretan en el navegador del cliente, sin tener que ir y venir del cliente al servidor actualizando la información.

- **CSS**

CSS (CSS son las siglas de Cascading Style Sheets,) es una tecnología que permite crear páginas web de una manera más exacta. Gracias a las CSS el programador es dueño de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir

márgenes, tipos de letra, fondos y colores. También se puede definir estilos propios en un archivo externo a las páginas; así, si en algún momento se desea cambiar alguno de ellos, automáticamente se actualizarán todas las páginas vinculadas al sitio. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML.

1.3.3. Lenguajes del lado del servidor:

Existe una multitud de lenguajes concebidos para el desarrollo de aplicaciones Web en el lado del servidor. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones. [4]

La versatilidad de un lenguaje está estrechamente relacionada con su complejidad. Un lenguaje complicado en su aprendizaje permite realizar un espectro de tareas más amplio y más profundo. En el dominio de la red, los lenguajes de lado del servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP y PERL.

- **PHP**

PHP (Hypertext Pre-processor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. [5]

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgre, Oracle y Microsoft SQL Server; facilitando la creación de aplicaciones web muy robustas. [6]

Ventajas:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos actuales.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

- Es libre, por lo que se presenta como una alternativa de fácil acceso.
- Permite las técnicas de Programación Orientada a Objetos.
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde php5). [7]

- **Active Server Pages (ASP)**

Es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma NET de Microsoft, de lo cual se deduce que es un lenguaje privativo y de un alto costo. Este soporta acceso a bases de datos, trabajo con archivos y carpetas, envío de email, paginación de resultados, procesado de formularios, entre otras opciones. [8]

En una página ASP se puede incluir casi todo: HTML plano, código de scripting, texto. No hay una distinción formal entre el contenido de una página y su comportamiento. Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente, hasta la implementación y la configuración de sitios. Además ASP es un sistema con nula portabilidad pues requiere necesariamente de un servidor Windows, con todas las implicaciones de alto costo, poca flexibilidad y escasa seguridad que estos equipos brindan.

- **PERL**

PERL (Practical Extracting and Reporting Language) es un lenguaje de programación de scripts que está disponible en múltiples plataformas y sistemas operativos. El desarrollo de aplicaciones en este lenguaje es muy rápido y se pueden incorporar a cualquier conjunto de instrucciones de él, una colección enorme de módulos que están favorables para su utilización.

Perl es “Software Libre”, lo que quiere decir que el código fuente está disponible para que cualquiera lo pueda ver, utilizar o modificar, brindándole de esta forma al programador mucha libertad para que haga el programa de acuerdo a sus necesidades.

Puede ser más lento que otros lenguajes haciendo lo mismo porque tiene que compilar el código fuente cada vez que corre el programa ya que no puede salvar su compilación intermedia como lo hacen Java y Python, los scripts Perl conllevan esta sobrecarga en cada ejecución utilizando muchos recursos de la máquina, lo que provoca que no sea tan ligero como otros lenguajes.

1.3.4. Fundamentación del lenguaje de programación a utilizar

Lenguaje del lado del cliente:

- **HTML, JavaScript**

HTML es el lenguaje más utilizado en el diseño de sitios Web, permite la inserción de multimedia como imágenes, sonido, videos y flash lo que resulta de agrado a la vista del usuario. Además, permite la ejecución de código JavaScript, otro lenguaje del lado del cliente que se utilizará en el desarrollo del sistema, ya que es orientado a objetos y admite un comportamiento de manera dinámica y eficaz. Además todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web.

Lenguaje del lado del servidor:

- **PHP**

Se ha seleccionado como lenguaje a utilizar en la implementación de la aplicación Web, al PHP. Teniendo en cuenta que es gratuito y multiplataforma. Debido a sus características hace posible que el cliente interactúe con una página rápida, eficiente y segura, capaz de mostrar y procesar información. Unido a lo anterior, se puede mencionar que es muy sencillo, legible y cuenta con una amplia librería de funciones que permiten hacer cualquier tipo de operación, como trabajo con archivos y carpetas, procesamiento de formularios, paginación de resultados y muchas otras opciones, lo cual concede la posibilidad de realizar un producto de acuerdo a las necesidades del usuario. Otras características de lenguaje que sustentan su selección son:

- Es software libre, lo que implica menor costo y servidores más baratos que otras alternativas.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con ella, le resultará muy fácil aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.
- PHP tiene una de las comunidades más grandes en Internet, por lo que no es complicado encontrar ayuda, documentación, artículos, noticias y más recursos.

1.3.5. Servidores WEB

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol), basado en arquitectura cliente/servidor. Está diseñado para transferir hipertextos, páginas Web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. [9] El servidor Web es un pilar importante en esta tecnología, puesto que es el programa que corre en una máquina central escuchando las peticiones de los clientes y respondiendo a las mismas en forma de una página que es interpretada por éste y mostrada al usuario de una manera entendible para él. Los más conocidos y utilizados en la actualidad son Apache y el IIS.

- **IIS (Internet Information Server)**

Internet Information Services (IIS) es un potente servidor Web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web sobre las versiones de Windows Server. El IIS es un software privativo que trabaja sólo sobre Windows, lo que lo deja por debajo de otros servidores Web. [10]

- **Apache**

Uno de los servidores Web más populares del mercado y el más utilizado actualmente es Apache, de código abierto y gratuito, disponible para Windows y GNU/Linux, entre otros. Apache presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. [11]

En el 2005, se convirtió en el más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado. Es un servidor que presenta un sin número de características que lo hacen estar entre los primeros en este ámbito. Es flexible, rápido y eficiente, puede ser ejecutado en cualquier plataforma y adaptado a diferentes entornos y necesidades. Se desarrolla de forma abierta y gracias a que es modular se han desarrollado diversas extensiones entre las que destaca PHP. [12]

Las principales características de Apache son:

- Funcionalidad en múltiples plataformas.
- Elaborado índice de directorios.
- Soporte del último protocolo http 1.2.
- Sencilla administración basada en la configuración de un único archivo.

- Soporte para CGI (Common Gateway Interface) y FastCGI.
- Mensajes de error altamente configurables.
- Bases de datos de autenticación y negociado de contenido.

1.3.6. Fundamentación del servidor Web a utilizar

- **Apache**

Como servidor Web se ha seleccionado el Apache, teniendo en cuenta que es gratuito, multiplataforma, eficiente y seguro. Además su selección también viene respaldada por la posibilidad de descarga a través de Internet, lo que posibilita la actualización del mismo. Adicionalmente y debido a que el código fuente está disponible, los usuarios pueden aumentar y personalizar las funciones del servidor añadiendo módulos que extienden sus capacidades. El servidor Apache es el más utilizado del mundo ofreciendo una perfecta combinación entre estabilidad y sencillez, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales.

1.3.7. Sistema de Gestión de Base de Datos (SGBD)

Los gestores de base de datos son sistemas formados por un conjunto de datos y un paquete de software para la gestión del mismo. Se controla el almacenamiento de datos redundantes. Estos resultan independientes de los programas que los usan, se guardan las relaciones entre los datos junto con éstos y se puede acceder a ellos de diversas formas. [13]

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

Entre los gestores de base de datos más comunes se encuentra el Microsoft SQL Server, el MySQL, Oracle y el PostgreSQL, a continuación se expondrán las características fundamentales de cada uno de ellos, lo que permitirá la selección del que se utilizará en la implementación del sistema propuesto.

Ventajas que ofrecen los SGBD:

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.

- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones y consultas.
- No hay duplicidad de información, comprobación de esta en el momento de introducirse.
- Integridad referencial al terminar los registros. [14]

• PostgreSQL

PostgreSQL es un servidor de base de datos relacional orientada a objetos y es el más utilizado por todos aquellos programadores que realizan aplicaciones cliente servidor, complejas o críticas. Es una alternativa económica a SQL Server, pues su costo es menor y tiene similares prestaciones. Este se puede utilizar sobre cualquier sistema operativo, característica que lo pone por encima de SQL Server y al parejo con MySQL. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. [15]

• SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Posee una gran velocidad y soporta un volumen de datos muy grande. También presenta mecanismos que le permiten realizar sentencias complicadas, lo que lo hace perfectamente adecuado para aplicaciones críticas y con cualquier grado de complejidad.

Aumenta las capacidades de los desarrolladores con un entorno de desarrollo valioso, flexible y actual para que diseñen bases de datos más seguras además de compartir datos a través de múltiples aplicaciones y dispositivos para facilitar la interconexión entre sistemas internos y externos. Posee el inconveniente que corre solamente sobre una plataforma, Windows, además de ser un software privativo con un alto costo. [15]

• MySQL

MySQL es un software libre, confiable y fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL, consume muy pocos recursos tanto del CPU como de memoria, presenta

mejoras en utilidades de administración (backup, recuperación de errores, etc.), no hay límites en el tamaño de los registros, tiene como una de sus principales ventajas la velocidad en la lectura de datos. [15]

En recientes versiones de MySQL se incluyen un conjunto de facilidades que no tenía en versiones anteriores como el uso de procedimientos almacenados, bloqueo de registros, integridad referencial, transacciones, entre otros, lo que aumenta aún más sus ventajas. [16]

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Pero en estos momentos algunas de sus partes son ofrecidas a clientes corporativos que pagan una suscripción por el sistema de base de datos, y éstos componentes sólo están disponibles como código cerrado, lo que dificulta la actualización del Gestor de Base de Datos, medida necesaria para mantener la seguridad. Además se eliminaron algunas características esenciales en los SGBD para obtener una mayor velocidad de lectura de datos. [17]

- **Oracle**

Es un sistema multiplataforma, sus características más destacadas son el soporte de transacciones, su gran estabilidad, seguridad y su escalabilidad. Su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales en general. En el desarrollo de páginas Web pasa lo mismo ya que como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL y SQL Server. [18]

Un aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma y las políticas de suministros de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. [18]

1.3.8. Fundamentación de la selección de la Base de Datos a utilizar

- **PostgreSQL**

Como Sistema de Gestión de Base de Datos se seleccionó PostgreSQL ya que es un software libre y además puede ser utilizado en cualquier plataforma. Por otro lado y comparándolo con los otros tipos de

bases de datos el más cercano es MySQL, pero actualmente es propietario. Además PostgreSQL posee una gran afinidad con PHP y es muy eficiente para aplicaciones no tan complejas como la que se pretende desarrollar. Son muchas las características de PostgreSQL que se tuvieron en cuenta a la hora de la elección, por ejemplo ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host. El tráfico de contraseñas está encriptado al conectarse a un servidor y soporta gran cantidad de datos. [19] PostgreSQL es un sistema que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

1.3.9. Marco de trabajo

- **Framework Zend Framework**

Zend Framework es un framework destinado para desarrollo de servicios y aplicaciones web con PHP, el cual ofrece soluciones en la construcción de sitios web modernos, sólidos y seguros. Tiene ventajas tales como ser de código abierto y trabajar con PHP 5, además de usar Zend Extension que fue diseñado para ser una extensión del mismo Zend Framework, contribuyendo en algunas mejoras como la memoria caché de consulta para Zend_Db_Table. Brinda también la posibilidad de instalar algún software adicional para gestionar sus aplicaciones web. Otras características que posee son:

- Trabajar con MVC (Modelo Vista Controlador).
- Contar con módulos para el manejo de archivos PDF, canales RSS y Web Services.
- Incluir objetos de las diferentes bases de datos, lo que se hace simple el trabajo para consultar su base de datos, sin tener que escribir consultas SQL.
- Poseer un buscador compatible con Lucene.
- Contiene clases robustas para la autenticación y el filtrado de entrada.

- **ExtJS Framework**

ExtJS comenzó siendo un conjunto de librerías y extensiones para YUI (Yahoo! User Interface). Poco tiempo después se convirtió en un Framework independiente y en el año 2007 se crea una compañía para comercializarlo y darle soporte. Posee dos tipos de licencias, LGPL y comercial y entre las principales características que lo distinguen se encuentran:

- Posibilidad de dividir la pantalla en paneles ajustables llamados layouts.
- Posibilita la inclusión de temas.

- Inserción de controles que soportan validación de datos.
- Poder diseñar menús drop-down y contextuales.
- Agrupar controles en pestañas.
- Utilización de elementos como combos editables, grid, ListView y listas con imágenes.
- Capacidad de crear nuevos controles así como de ampliar la funcionalidad de los ya existentes.

- **Doctrine**

Doctrine es un Mapeador Objeto-Relacional (ORM) para PHP 5.2.3 + que se halla en la cúspide de una potente capa de abstracción de bases de datos (DBAL). Entre sus principales características está la opción de escribir las preguntas en una base de datos orientada a objetos de propiedad denominada Doctrine Query Language (DQL), inspirado por Hibernate HQL. Esto provee a los desarrolladores una poderosa alternativa que mantiene la flexibilidad de SQL sin necesidad de la duplicación de código. Otra característica que posee es la posibilidad de exportar una base de datos existente a sus clases correspondientes y viceversa.

1.3.10. Metodologías de desarrollo del software

Durante el desarrollo de un software es importante aplicar una metodología adecuada para obtener clientes satisfechos con el resultado y desarrolladores orgullosos de su trabajo. No existe una metodología universal para el desarrollo del software, puesto que las características de cada proyecto exigen que el proceso sea configurable. [20]

En la actualidad existen varias metodologías orientadas a objetos (OO) basadas en UML: Rational Unified Process (RUP), OPEN, MÉTRICA 3, siendo el RUP la metodología adecuada para el desarrollo de la solución propuesta y la que se aplica actualmente en los proyectos desarrollados en la universidad.

Metodología RUP

RUP es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza Unified Model Language (UML) como lenguaje de representación visual y está basado en componentes. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

En RUP se han definido 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. Está compuesto por cuatro fases denominadas Inicio,

Elaboración, Construcción y Transición. Estas se subdividen por iteraciones que ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. En la Figura 1.1 se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

Características de RUP

- Dirigido por los casos de uso: En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. Cada iteración toma un conjunto de casos de uso o escenarios y desarrolla todo el camino a través de las distintas disciplinas.
- Centrado en la arquitectura: El Proceso Unificado, se caracteriza por el establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.
- Iterativo e incremental: RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y las cuales se definen según el nivel de madurez que alcanzan los productos que se van obteniendo con cada actividad ejecutada. La terminación de cada fase ocurre en el hito correspondiente a cada una, donde se evalúa que se hayan cumplido los objetivos de la fase en cuestión.

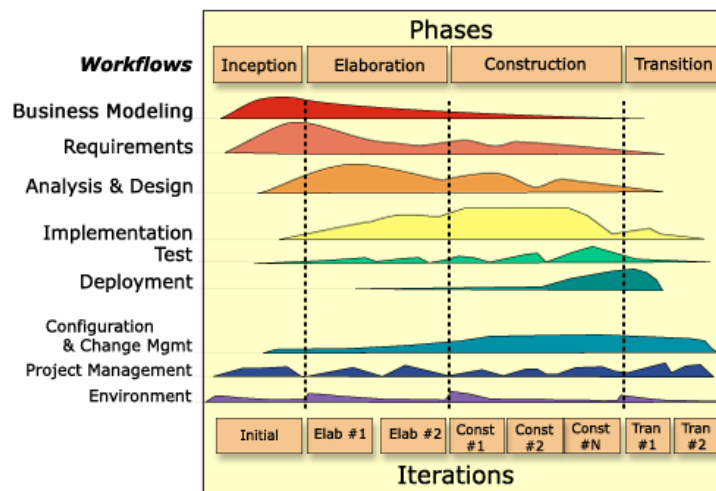


Figura 1.1 Flujos y fases de RUP

- **Lenguaje Unificado de Modelado (UML)**

Como lenguaje para el Modelamiento se seleccionó UML (Unified Modeling Language) que permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, cuya finalidad es presentar diversas perspectivas de un sistema, las cuales se conocen como modelo. [21]

Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. El UML es una herramienta que sirve de enlace entre quien tiene la idea y el desarrollador, ya que le ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien esté involucrado en su proceso de desarrollo. Este lenguaje se ha convertido en el estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación.

Las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.
- Los modelos permiten la comunicación con el cliente en todas las etapas.

1.4. Herramientas utilizadas

- **Macromedia Dreamweaver MX**

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado para la creación de aplicaciones Web. Cuenta con un amplio número de herramientas que permiten elaborar desde la más simple página Web personal hasta el sitio Web más completo y complejo. Admite la inclusión de tecnologías como PHP y MySQL haciéndolo muy conveniente para el desarrollo de la aplicación Web que automatizará las pruebas en el Laboratorio de liberación. Además posibilita la creación de una interfaz de usuario integrada, que facilita la interacción dada la potencialidad de sus componentes de trabajo. Abarca desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas, compatibles con las mejores prácticas y las tecnologías más recientes. [22]

- **Visual Paradigm para UML**

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Está disponible en varias ediciones, cada una destinada a determinadas necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Es una herramienta fácil de usar que soporta la notación UML 2.1, ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XMI, generador de informes, editor de figuras, integración con MS Visio, plug-in, integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros. Entre sus nuevas características se incluyen el modelado colaborativo con CVS y Subversión. [23]

- **Zend Studio**

Es un Integrated Development Environment (IDE) o ambiente de desarrollo integrado o disponible para desarrolladores y concentra componentes necesarios para desarrollar de aplicaciones utilizando PHP.

Zend Studio está constituido por dos partes en las que se fraccionan las funcionalidades del lado del cliente y las del servidor. Estas partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Admite además hacer depuraciones simples de scripts, pero para contar con toda la potencia de la herramienta de depuración se debe disponer de la parte del servidor, que instala el servidor Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar en conjunto de depuración. Existen versiones del producto para Windows, Linux y MacOS.

1.5. Conclusiones

En este capítulo se realizó un análisis de los sistemas existentes y se decide la implementación de una herramienta que se adecúe a las características propias del Laboratorio de pruebas del CCES, con la que se pueda integrar los procesos asociados a las No conformidades, la asignación de tareas y generación de reportes. Se investigaron las tecnologías que se adecuan para llevar a cabo el desarrollo del sistema que se pretende implementar, fundamentando la elección de las mismas. Se ha escogido Apache como servidor Web para sostener la aplicación. PHP será el lenguaje de programación producto de su eficiencia y portabilidad en diferentes plataformas y como Gestor de Base de datos se seleccionó a PostgreSQL. RUP será la metodología que guiará el proceso de desarrollo. Para la selección de las tecnologías se tuvo en cuenta que fueran software libre, en el caso del Visual Paradigm se seleccionó por ser una herramienta

que brinda muchas facilidades. Presenta una sencilla interfaz que permite al usuario trabajar con comodidad.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En el presente capítulo se realiza un análisis de las causas que originan la situación problemática y se describe el flujo actual de los procesos. Se realiza un Modelo del Negocio y se enumeran las características básicas del sistema listando los requisitos funcionales y no funcionales. Se definen además los casos de uso, actores y trabajadores del sistema.

2.2. Objeto de estudio

Se definió en el marco teórico como objeto de estudio los procesos asociados a la gestión de las No conformidades, a partir de esta definición se comienzan a estudiar los procesos que componen la gestión de No conformidades.

2.3. Descripción de los procesos del negocio

A continuación se procede a la descripción de los procesos actuales que se desarrollan como parte del negocio.

2.3.1. Proceso inicial

Las pruebas se desarrollarán en dos fases fundamentales, la planificación y la realización de la prueba propiamente, donde la segunda fase se repetirá tantas veces como sea preciso para alcanzar un producto apto y con la menor cantidad de errores posibles. Estas pruebas tienen que ser previamente solicitadas al Laboratorio de prueba por el Equipo de desarrollo para facilitar el trabajo de ambos y a su vez un uso gestionado de los recursos, para ello existen una serie de actividades entre las cuales están:

- **Decidir que tipo de pruebas se realizarán al software**

Una vez solicitada la prueba el equipo del Laboratorio define, a partir de los elementos que se desean probar, que tipo de pruebas son las más recomendables aplicar.

- **Determinar los recursos a utilizar en la prueba**

El responsable del laboratorio realiza un análisis de los recursos con los que cuenta para la realización de la prueba, es decir la cantidad de recursos humanos y tecnológicos útiles en el momento de efectuar la prueba.

- **Planificar la realización de la prueba**

En esta actividad una vez decidido el tipo de prueba a realizar, se pasa a la planificación de la prueba; se tienen presente los medios tecnológicos que se van a utilizar así como la fuerza de trabajo necesaria, se analiza si es precisa la planificación de turnos de trabajo y se estima la fecha de inicio y la fecha de fin de la prueba. En esta etapa se pactan entre el Equipo de desarrollo y el Laboratorio de prueba los elementos que serán entregados para la revisión.

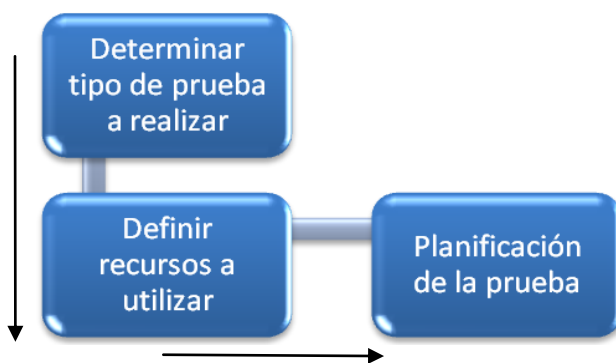


Figura 2.1 Proceso inicial.

2.3.2. Proceso de Gestión de No conformidades

Este proceso se realiza con la idea de detectar la mayor cantidad de errores en el menor tiempo posible y corregirlos de manera eficiente evitando la inserción de nuevos defectos, para ello se realizan las siguientes tareas:

- **Entrada al laboratorio**

Una vez concluido el proceso inicial se informa al especialista la realización de la prueba y se procede a la entrega que se realizará en el servidor donde cada persona tiene los privilegios necesarios según la tarea que le corresponda. El proyecto informa por vía e-mail o telefónica el momento justo de la entrega al Líder del Laboratorio de prueba. Esta contiene, según los elementos que se entregan de forma estructurada, las especificaciones necesarias para el eficiente desarrollo de la prueba. Además

incluye las No conformidades que se han detectado en iteraciones anteriores, si han ocurrido. A la vez se deben especificar en la entrega los elementos a tener en cuenta durante las comprobaciones. El Líder del Laboratorio de prueba, verifica que estén todos los artefactos previamente pactados y genera un documento de Aceptación o de No conformidades de acuerdo a la evaluación de lo recibido. De acuerdo a la valoración del Líder del Laboratorio de prueba, si se decide que no se procederá a probar, se indicaría en este documento las causas.

- **Distribución del trabajo entre los probadores**

Cuando ya se tienen todos los elementos a revisar se pasa a la asignación del trabajo a los probadores, se elabora un informe con la distribución que se irá haciendo por nivel de complejidad y en dependencia de la cantidad de personas o turnos que se hayan decidido. Los probadores deberán conocer la distribución así como la etapa de detección y en caso de que la etapa de detección sea posterior a las Pruebas Exploratorias (PE), debe existir la carpeta de Referencias para Calidad, donde se recogen las No conformidades anteriormente detectadas.

- **Realización de la prueba**

Los probadores comienzan a examinar los artefactos que pueden ser documentos o la aplicación, dentro de la documentación se encuentran los Diseños de caso de prueba (DCP), los Diseños de caso de uso (DCU), descripción de requisitos, los manuales y otros. En la aplicación se encuentra el código fuente estructurado de manera funcional y lógica organizado arquitectónicamente. Estos elementos son revisados por el probador quien irá detectando los defectos a partir de la comparación con estándares previamente establecidos. Si la etapa de detección es mayor que las de PE entonces deberá primeramente verificar que las anteriormente detectadas estén verdaderamente resultas.

- **Clasificación de las No conformidades**

Las clasificaciones de las No conformidades varían de la documentación a la aplicación aunque existen similitudes como los defectos categorizados como **Ortografía**, en esta clasificación se recoge todo lo referente a los errores ortográficos, gramaticales y de concordancia, estas son siempre de tipo significativa; **Validación** se refiere a los mensajes que deben ser mostrados en la aplicación y no aparecen recogidos en los DCP y viceversa, incluye también los campos que no están correctamente validados según la entrada de datos que deben tener y son de tipo significativa; **Funcionalidad** en la documentación recoge los errores en los pasos del flujo central que guían lo que debe hacer el usuario y aparecen en la sección Requisitos a probar del DCP, en la aplicación concierne a los

errores que no permiten que una funcionalidad determinada se ejecute, son de tipo significativa; **No correspondencia con los casos de uso o los requisitos** trata los errores cuando algún detalle no coincide entre DCU o Diseño de requerimientos (DR), aplicación y DCP, estas son de tipo significativa; la clasificación **Otros** recoge los errores relacionados con estándares de diseño o plantillas y son de tipo no significativas a excepción de los errores idiomáticos que son la mezcla de dos o más idiomas y se clasifican de tipo significativa. Propias de la aplicación son **Opciones que no funcionan** que agrupa los errores de componentes, opciones o funciones que aparecen sin desarrollar ninguna operación y son de tipo significativa. **Excepciones** por su parte, recoge los errores de excepciones controladas o no controladas por el sistema que no se corresponden con los errores cometidos y son significativa. De tipo **Interfaz** son los errores de incumplimiento con los estándares para el diseño de interfaces y son de tipo significativa. El probador cuando está efectuando las pruebas puede hacer recomendaciones sobre los aspectos que estime conveniente, pero a su vez **Recomendación** es otra clasificación, para las No conformidades que son poco relevantes y no entran en la clasificación de significativa o no significativa.

- **Generación de partes**

Una vez terminado el trabajo de los probadores, se sube al servidor para ser revisado por el Líder del laboratorio quien debe emitir un parte con las No conformidades clasificadas y contabilizadas que será recibido por el Equipo de desarrollo que al igual debe registrar el trabajo en el historial de las No conformidades. Además al final del día se crea un reporte donde se controla todo lo realizado durante la jornada.

- **Seguimiento de las No conformidades detectadas**

Este eslabón del proceso tiene gran importancia pues en él se llevan a cabo actividades que tributan a la calidad del producto final, cada vez que se ejecute este proceso será un paso más de avance en la optimización del software. Comienza cuando el Equipo de desarrollo recibe las No conformidades detectadas de ahí se emprenden las actividades que se explican a continuación:

- **Respuesta a las No conformidades**

Cuando las No conformidades que son recibidas por el Equipo de desarrollo llegan en estado Pendiente (PD), el equipo de desarrollo tiene que devolverlas en estado Resueltas (RA) o en No

procede (NP), el NP se acepta en el Laboratorio de prueba cuando viene con una justificación avalada por el Comité de control de cambios.

- **Culminación de la iteración**

La iteración a concluido cuando el Equipo de desarrollo hace otra entrega al Laboratorio de prueba, en esta prueba se verificará que las No conformidades detectadas anteriormente en Pruebas de Liberación o de Aceptación estén resueltas y que no hallan sido introducidas otras nuevas; posteriormente, de ser necesario se realiza una vez más el proceso de Gestión de las No conformidades, hasta que el producto esté listo para ser liberado.

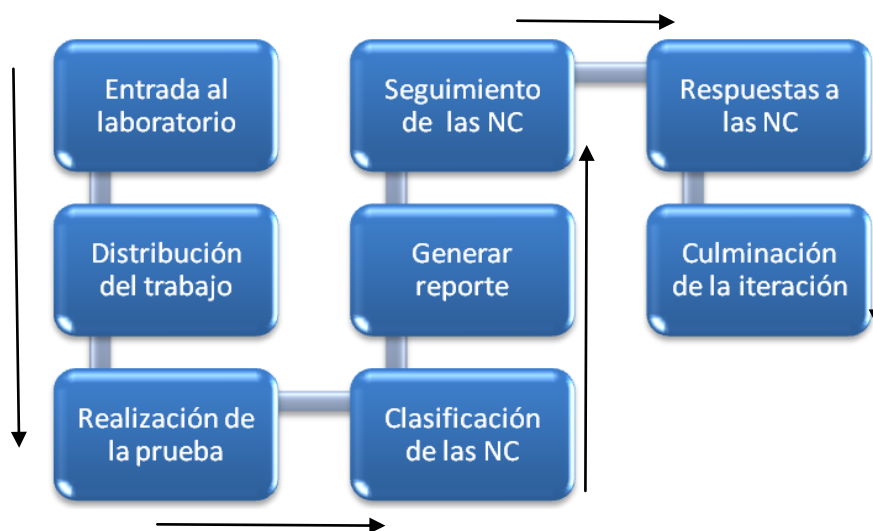


Fig. 2.2 Proceso de gestión de las No conformidades

Actualmente el proceso de gestión de las No conformidades detectadas durante las pruebas en el laboratorio se tramita por medio de plantillas elaboradas con el programa Microsoft Excel.

2.4. Análisis crítico del proceso actual

El proceso de gestión de las No conformidades que actualmente se lleva a cabo en el Laboratorio de Pruebas presenta un grupo de dificultades que conllevan a una gestión ineficiente del tiempo y los recursos. Los mayores problemas en el proceso radican en el hecho de que las tareas se realizan de forma manual. Esto provoca que a la hora de llenar las plantillas y repartir el trabajo a los probadores, el proceso se torne ineficaz y lento. Por otra parte ocurren errores a la hora de contabilizar la cantidad de No conformidades detectadas y debido a la cantidad de información que se maneja alguna información queda fuera del reporte.

2.5. Procesos objeto de automatización

Se pretende automatizar los siguientes procesos:

- Distribución del trabajo
- Realización de la prueba
- Clasificación de las No conformidades
- Generar reportes
- Seguimiento de las No conformidades
- Respuesta de las No conformidades
- Culminación de la iteración

2.6. Información que se maneja

La documentación que se maneja dentro del CCES se caracteriza por contener información sobre el desarrollo y estado de los proyectos, se debe lograr, que la manipulación de la misma sea de manera segura y confiable, que no atente contra la autenticidad e integridad de la misma. Para ello los documentos con los que se trabaja así como los datos que se gestionan en el sistema, están sujetos a reglas de seguridad por lo que, las personas que acceden a estos asumen la custodia de los mismos y no están autorizadas a divulgar las informaciones que manejan. Se define dentro del proceso de Gestión de No conformidades el término elemento para designar los artefactos que serán probados, estos podrían ser DCU, Listas de chequeo, manuales, aplicaciones, ejecutables o cualquier otro tipo de componente de software al que se le puedan aplicar pruebas.

2.7. Modelo del negocio

Con el objetivo de comprender la estructura y la dinámica del CCES así como obtener una visión de la organización se realiza la modelación del negocio. El modelo de negocio permite definir los procesos, roles y responsabilidades del centro en cuestión. Propicia además que los usuarios finales y los desarrolladores tengan un entendimiento común de la organización del sistema.

2.8. Actores del negocio

Se define como actor del negocio cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se

juega cuando se interactúa con el negocio para beneficiarse de sus resultados. En la siguiente tabla se definen los actores del negocio y su justificación.

Actores del Negocio	Justificación
Solicitante	Líder de proyecto que solicita a la dirección de CCES la entrada de los componentes a probar. Es quien inicia las operaciones que dan comienzo al proceso de negocio.

Tabla 2.1: Actores del negocio

2.9. Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

Trabajadores del negocio	Justificación
Líder del laboratorio	Máximo responsable del laboratorio de prueba.
Jefe de turno	Responsable de repartir el trabajo a los probadores y enviar los reportes de las pruebas.
Probador	Persona que revisa los elementos que se encuentran en el proceso de prueba y registra las No conformidades detectadas.

Tabla 2.2: Justificación de los trabajadores del negocio

2.10. Diagrama de casos de uso del negocio

Describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio y sus relaciones, que se corresponden con los procesos del negocio y los clientes, respectivamente.

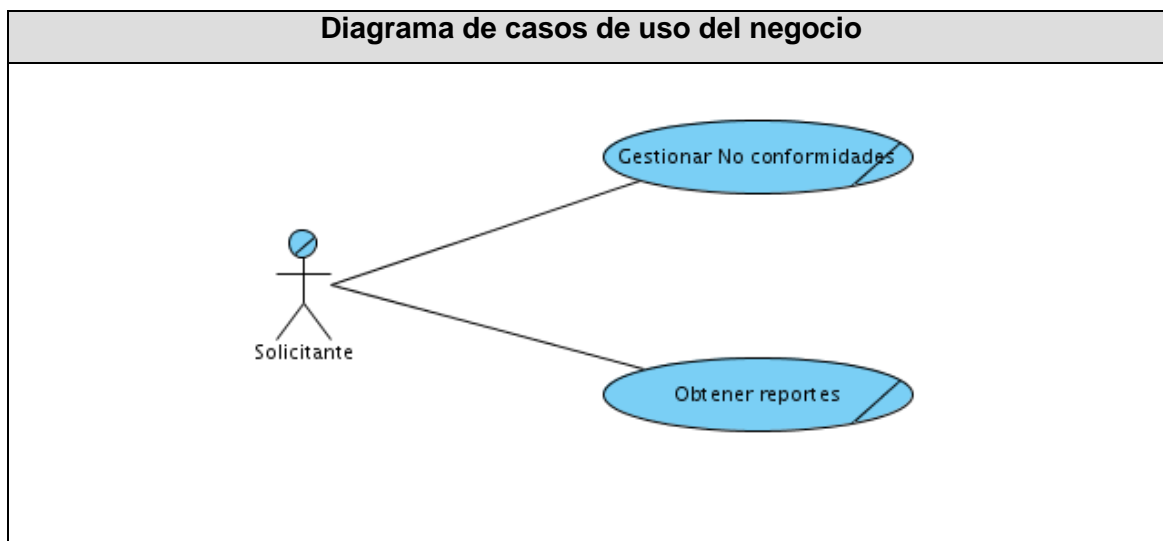


Figura 2.3: Diagrama de Casos de Uso del Negocio.

2.11. Descripción de los casos de uso del negocio

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio.

Especificación textual del Caso de Uso “Gestionar No conformidades”

Nombre del caso de uso del negocio:	Gestionar No conformidades
Actores del negocio:	Solicitante
Trabajadores del negocio	
Propósito:	En este Caso de Uso se describe como se detectan las No conformidades en el proceso del negocio.
Resumen:	
El solicitante inicia el caso de uso al informar al Líder del laboratorio que los elementos a probar se encuentran en el lugar de la entrega. El líder del laboratorio informa al jefe de turno que ya se encuentra el contenido de trabajo actualizado y este lo reparte entre los probadores, estos entregan al jefe de turno las No conformidades que encuentran, el caso de uso finaliza cuando el Jefe de turno realiza la entrega de las No conformidades detectadas al solicitante.	

Casos de uso asociados:	
Flujo de trabajo	
Acción del actor	Respuesta del negocio
1. El Solicitante informa al Líder del laboratorio la entrega de los artefactos.	<p>2. El Líder del laboratorio revisa la información y verifica que la estructura de los elementos sea la pactada.</p> <p>3. El Líder del laboratorio informa al Jefe de turno la entrada de trabajo.</p> <p>4. El jefe de turno recibe el trabajo y lo distribuye entre los probadores.</p> <p>5. Los probadores detectan las No conformidades en los artefactos entregados.</p> <p>6. Los probadores entregan al Jefe de turno las inconformidades encontradas.</p> <p>7. El Jefe de turno revisa las No conformidades</p> <p>8. El Jefe de turno elabora los partes con la información recolectada.</p> <p>7. El caso de uso finaliza cuando el Jefe de turno entrega al Solicitante las No conformidades halladas.</p>
Prioridad:	Alta
Mejoras:	Será más fácil la comunicación entre los trabajadores y se disminuirá la cantidad de errores a la hora de contabilizar el trabajo.
Cursos alternos:	
Si el Líder del Laboratorio al revisar los artefactos determina que no son correctos o no	

se corresponde con lo pactado con el solicitante, no se recibe y por tanto no se realiza la prueba.

Tabla 2.3: Descripción textual del CU Gestionar No conformidades

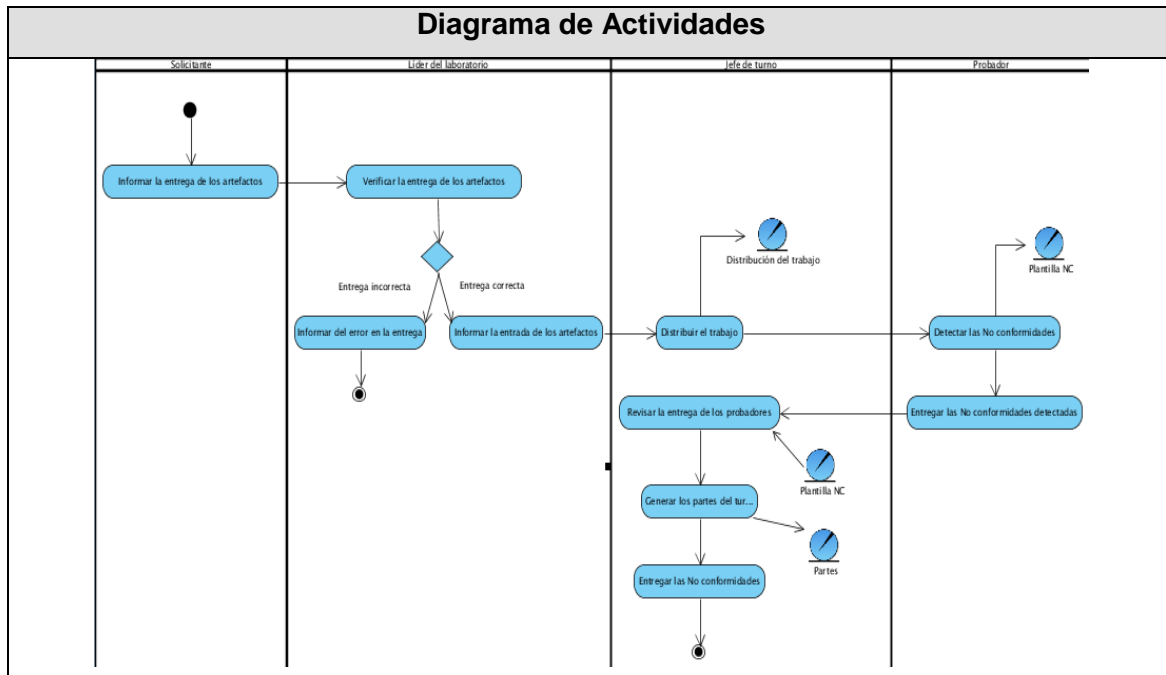


Figura 2.4: Diagrama de Actividades CU No conformidades

2.12. Especificación de los requisitos de software

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

2.12.1. Requerimientos funcionales del sistema

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

RF1. Gestionar línea

RF1.1 Adicionar línea

RF1.2 Modificar línea

RF1.3 Eliminar línea

RF2. Gestionar proyecto

RF2.1 Adicionar proyectos

RF2.2 Modificar proyectos

RF2.3 Eliminar proyectos

RF3. Gestionar componente

RF3.1 Adicionar componente

RF3.2 Modificar componente

RF3.3 Eliminar componente

RF4. Gestionar tipo de prueba

RF4.1 Adicionar tipo de prueba

RF4.2 Modificar tipo de prueba

RF4.3 Eliminar tipo de prueba

RF5. Gestionar elementos

RF5.1 Adicionar elemento

RF5.2 Modificar elemento

RF5. 3 Eliminar elemento

RF6. Gestionar probador

RF6.1 Adicionar probadores

RF6.2 Modificar probadores

RF6. Eliminar probadores

RF7. Gestionar trabajo

RF7.1 Adicionar trabajo

RF7.2 Modificar trabajo

RF7.3 Eliminar trabajo

RF7.4 Repartir trabajo

RF8. Gestionar No conformidades.

RF8.1 Adicionar No conformidades

RF8.2 Modificar No conformidades

RF8.3 Eliminar No conformidades

RF8.4 Revisar No conformidades

RF8.5 Aprobar No conformidades

RF8.6 Responder No conformidades

RF9. Generar partes

RF9.1 Generar parte de producción

RF9.2 Obtener reporte

RF9.2.1 Reporte diario

RF9.2.2 Reporte semanal

RF9.2.3 Reporte por elemento

RF9.2.4 Reporte por componente

2.12.2. Requerimientos no funcionales del sistema

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

- **Apariencia o interfaz externa:**

- Diseño sencillo, de manera que se pueda utilizar de forma fácil el sistema.
- Diseño para la resolución 800x600, aunque podrá verse en otras resoluciones.

- **Usabilidad**

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

- **Rendimiento**

- Rápido acceso de búsqueda a la información en tiempos relativamente cortos.
- El sistema deberá de ser lo más estable y confiable posible.

- **Soporte:**

- Se requiere un navegador que interprete Java Script.
- Versión de PHP 5.0 o superior.
- Se requiere un servidor de bases de datos que soporte grandes volúmenes de información y velocidad de procesamiento.
- Se requiere la instalación del navegador Mozilla Firefox 3.0.

- **Portabilidad**

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

- **Seguridad**

- Establecer permisos para garantizar que sólo se acceda a la información quien tenga permiso.

- Garantizar la protección ante acciones no autorizadas.
- Verificación sobre acciones irreversibles (eliminación, modificación).
- **Confiabilidad**
 - Garantizar un tratamiento adecuado de excepciones y validaciones de las entradas de los usuarios.
 - Garantizar la recuperación del sistema ante fallos y errores.

2.13. Modelo de casos de usos del sistema

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

2.14. Definición de los actores

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

Actores	Justificación
Líder del laboratorio	Máximo responsable del laboratorio de prueba, es además quien recibe la entrada de los artefactos a probar.
Jefe de turno	Responsable de repartir el trabajo a los probadores y enviar los reportes de las pruebas.
Probador	Persona que revisa los elementos que se encuentran en el proceso de prueba y registra las No conformidades detectadas.
Solicitante	Interactúa con el sistema cuando envía los artefactos a revisar, también cuando recibe las No conformidades, y por último cuando desea ver algún reporte.
Trabajador	Es la generalización de Probador y Jefe de turno.
Calidad central	Es el usuario que puede acceder a los reportes para saber el estado del Laboratorio de prueba.

Solicitador de reportes	Es la generalización de Calidad central, Líder de laboratorio, Jefe de turno y Solicitante.
Usuario común	Es la generalización de Líder de laboratorio y Jefe de turno.
Usuario laboratorio	Es la generalización del Jefe de turno, Solicitante y Probador.

Tabla 2.5 Actores del sistema

2.15. Listado de casos de uso

CU-1	Gestionar línea
Actor	Jefe de turno
Descripción	Se gestionan los datos de las líneas de los proyectos.
Referencia	RF1

CU-2	Gestionar proyecto
Actor	Jefe de turno
Descripción	Se gestiona toda la información sobre los proyectos que entran al laboratorio de prueba.
Referencia	RF2

CU-3	Gestionar componentes
Actor	Jefe de turno
Descripción	Permite la gestión de los diferentes componentes de los proyectos que se probarán.
Referencia	RF3

CU-4	Gestionar tipo de prueba
Actor	Jefe de turno
Descripción	Se gestiona toda la información sobre los tipos de prueba.
Referencia	RF4

CU-5	Gestionar elemento
Actor	Jefe de turno

Descripción	Se gestionan los elementos que se probarán.
Referencia	RF5

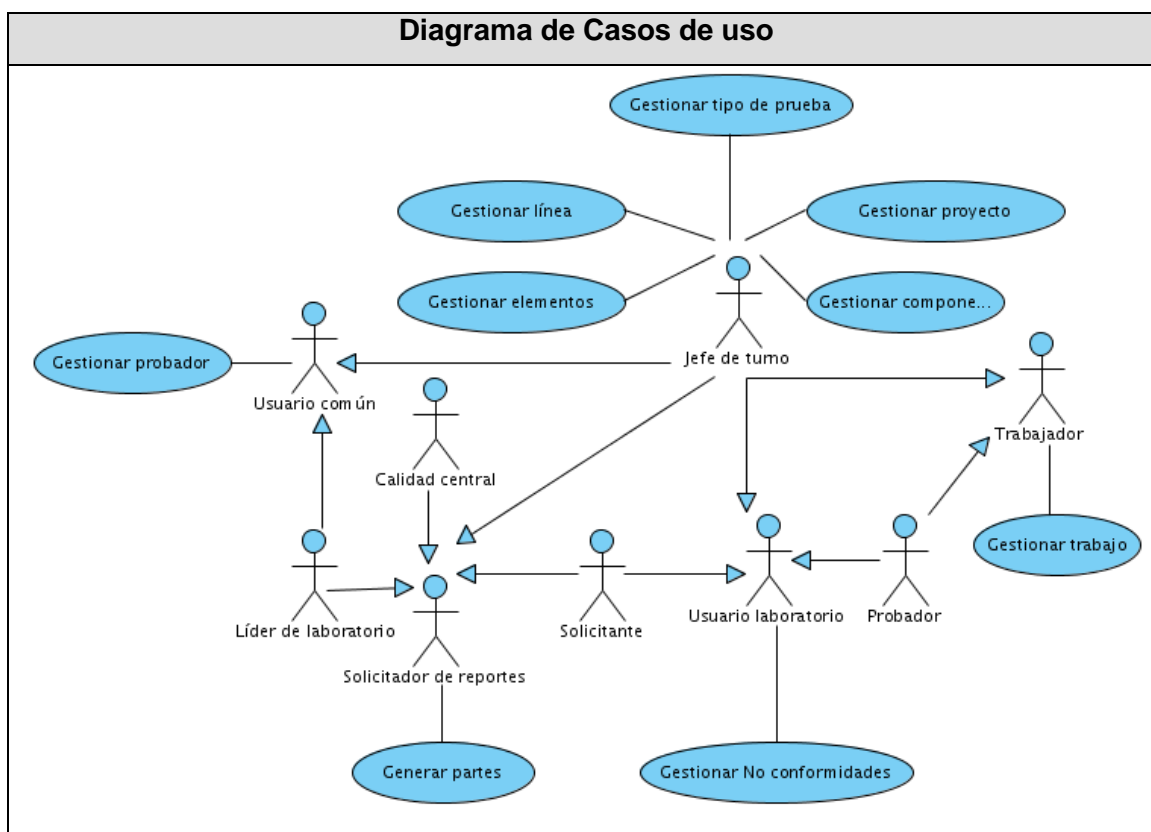
CU-6	Gestionar probador
Actor	Usuario común
Descripción	Se gestionan todos los datos de los probadores pertenecientes al CCES.
Referencia	RF6

CU-7	Gestionar trabajo
Actor	Trabajador
Descripción	Se asigna el trabajo entre los probadores.
Referencia	RF7

CU-8	Gestionar No conformidades
Actor	Usuario laboratorio
Descripción	Se gestionan las No conformidades que el probador detecta durante las pruebas. El Jefe de turno las revisa y aprueba.
Referencia	RF8

CU-9	Generar partes
Actor	Solicitador de reporte
Descripción	Genera el parte de producción del día y los reportes que se precisen.
Referencia	RF9

2.16. Diagrama de casos de uso del sistema.



Una vez definidos los casos de uso del sistema se pasa a la descripción textual de los mismos, mediante la descripción textual se explica de forma detallada el funcionamiento y el desarrollo de los casos de uso, así como las interrelaciones que existen entre ellos. Por la existencia de nueve casos de uso, se incluye en este capítulo sólo la descripción textual del caso de uso “Gestionar No conformidades” por ser el caso de uso principal del sistema, además de tener gran amplitud y complejidad, el resto de las descripciones textuales se encuentran en la sección de anexos del documento, bajo el subtítulo Anexos del Capítulo 2 “Descripción textual de los casos de uso del sistema”.

Especificación textual del Caso de Uso “Gestionar No conformidades”

Caso de Uso	Gestionar No conformidades
Actores	Trabajador
Propósito	Permitir al usuario insertar, modificar y eliminar no conformidad, así como revisar, aprobar y dar respuestas a las mismas.
Resumen	Los probadores insertan las No conformidades, posteriormente las

	revisan, haciendo cambios o eliminando algunas, el jefe de turno las revisa y aprueba, el equipo de desarrollo les da respuesta.
CU asociados	
Precondiciones	El actor debe estar autenticado y debe tener los permisos necesarios para realizar la funcionalidad. Deben existir líneas, proyectos, componentes y elementos insertados en el sistema.
Poscondiciones	El sistema contiene No conformidades insertadas, modificadas, revisadas y aprobadas.
Referencias	RF8.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona en el menú Inicio/Qualitás/Gestionar No conformidades.	2. El sistema muestra la interfaz de Gestionar No conformidades.
3. El actor selecciona la opción Adicionar. - -Si el actor selecciona la opción Modificar (Ver Sección I) - -Si el actor selecciona la opción Eliminar (Ver Sección II) - -Si el actor selecciona la opción Revisar (Ver Sección III) - -Si el actor selecciona la opción Respuesta a las No conformidades (Ver Sección IV)	4. El sistema muestra una interfaz con los campos necesarios para adicionar una nueva no conformidad.
5. El actor llena los campos con los datos correspondientes y presiona el botón Guardar.	6. El sistema muestra un mensaje indicando que la No conformidad ha sido guardada correctamente y que puede añadir una nueva. 7. El sistema actualiza la lista de las

	NC.
Cursos Alternos	
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Línea 5: El actor presiona el botón Aplicar , sistema actualiza el listado de las NC y mantiene la interfaz abierta por si el usuario desea insertar una nueva NC.	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Modificar de la NC que desea variar.	2. El sistema muestra una interfaz con los datos que posee la NC.
3. El actor modifica la NC y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que la NC fue modificada satisfactoriamente. 5. El sistema actualiza la lista de las NC.
Cursos Alternos	
Línea 3: En caso de que el actor desee cancelar la operación podrá presionar el botón Cancelar y el sistema cierra la interfaz.	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción eliminar de la NC que desea quitar.	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar .	4. El sistema muestra un mensaje indicando que el elemento fue borrado satisfactoriamente 5. El sistema actualiza la lista de las NC.
Cursos alternos	
Línea 3: En caso de que el actor desee cancelar la operación podrá presionar el botón Cancelar y el sistema cierra la interfaz.	
Sección (III) Revisar	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la NC que desea revisar y presiona el botón Revisar .	2. El sistema muestra una interfaz con los datos de la NC.
3. El usuario modifica los datos de haber	5. El sistema muestra un mensaje

algún error.	indicando que la NC se guardó
4. El usuario presiona el botón Aceptar .	correctamente.
Cursos alternos	
Línea4: El actor presiona el botón Cancelar , y sistema cierra la interfaz.	
Sección (VI) Aprobar	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la NC que desea aprobar y presiona el botón Aprobar.	2. El sistema muestra un mensaje indicando que se aprobó correctamente la NC.
Cursos Alternos	
Línea1: El actor presiona el botón Cancelar , y sistema cierra la interfaz.	
Sección (V) Respuesta a las No conformidades	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el menú Inicio/Qualitás/Respuesta a las No conformidades .	2. El sistema muestra una ventana todas las No conformidades pendientes a respuestas.
3. El usuario selecciona la No conformidad que desea responder y selecciona el botón Respuesta a las No conformidades .	5. El sistema cambia el estado de la NC.
4. El actor cambia el estado de la NC y presiona el botón Aceptar .	6. El sistema muestra un mensaje indicando que la operación fue satisfactoria.
	7. El sistema actualiza el listado de las No conformidades.
Cursos Alternos	
Línea 4: El usuario presiona el botón Cancelar y el sistema cierra la interfaz.	

2.17. Conclusiones.

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

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

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

3.2. Análisis

El análisis juega un papel principal en la comprensión del sistema que se pretende construir, mediante esta etapa definida por RUP, se hace más fácil la visualización de cómo quedará el sistema, pues mediante los diagramas de clases se podrá conocer de manera precisa, la disposición de los elementos que conformarán el producto final.

3.2.1. Diagramas de clases del análisis

Los diagramas de clases están compuestos por las clases interfaz, las clases controladoras y las clases entidad, mediante estos diagramas es posible conocer como quedará el flujo de la información, así como la distribución de las interfaces que poseerá el sistema y la relación entre ellas.

En consecuencia con el capítulo anterior se adjuntará sólo el diagrama de clases del análisis del caso de uso “Gestionar No conformidades”, el resto de los diagramas se encuentran en los Anexos del trabajo bajo el subtítulo Anexos del Capítulo 3 “Diagrama de clases del análisis.”

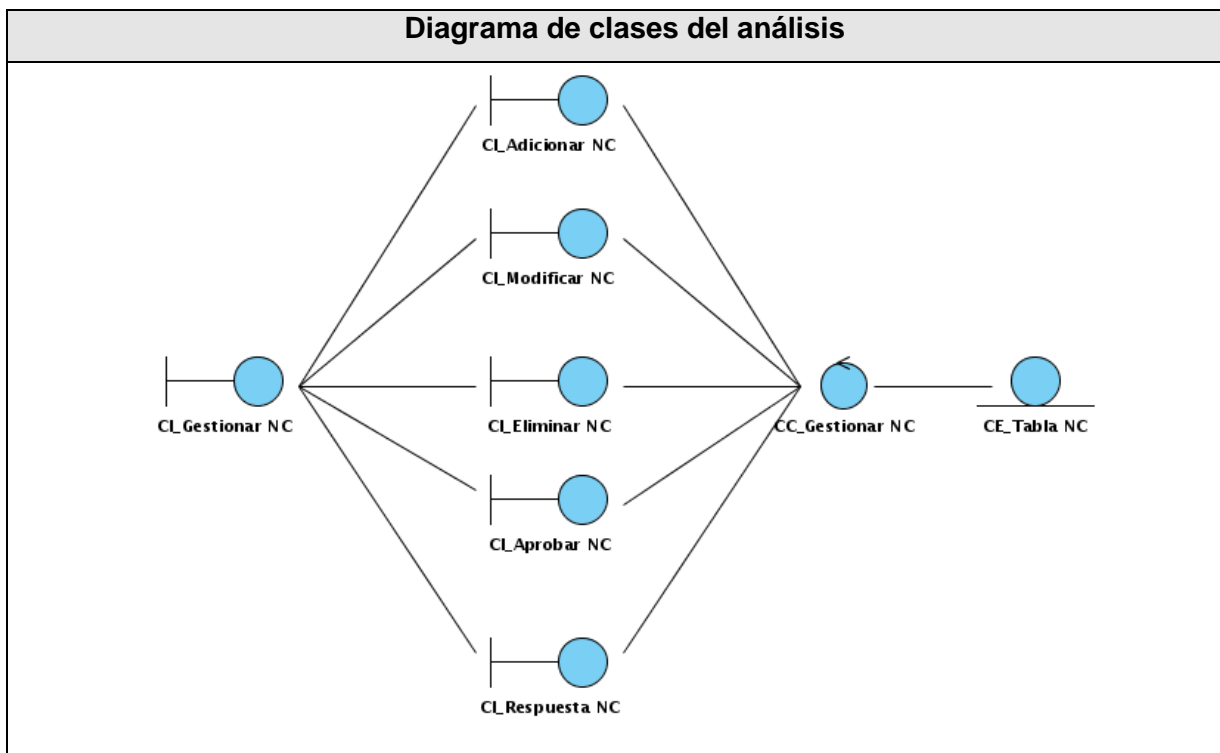


Figura 3.1 Diagrama de clases del análisis "Gestionar No conformidades"

3.3. Diseño

El diseño es parte fundamental en la construcción del sistema que se pretende crear, pues es una abstracción de cómo quedará el producto final. A partir del diseño se obtendrá un modelo cercano a la versión final del software que se construye.

3.3.1. Diagrama de clases del diseño web

Mediante los diagramas de clases de diseño web es posible obtener una abstracción de cómo quedará la implementación del sistema, los diagramas especifican la relación entre los distintos elementos lo que hace que sea más fácil la implementación, facilitan y proveen una abstracción de lo que será el producto final. En esta sección se adjunta el diagrama de clases del diseño web del caso de uso "Gestionar No conformidades", el resto de los diagramas de este tipo se encuentran en los Anexos del trabajo bajo el subtítulo Anexos del Capítulo 3 "Diagrama de clases del diseño web."

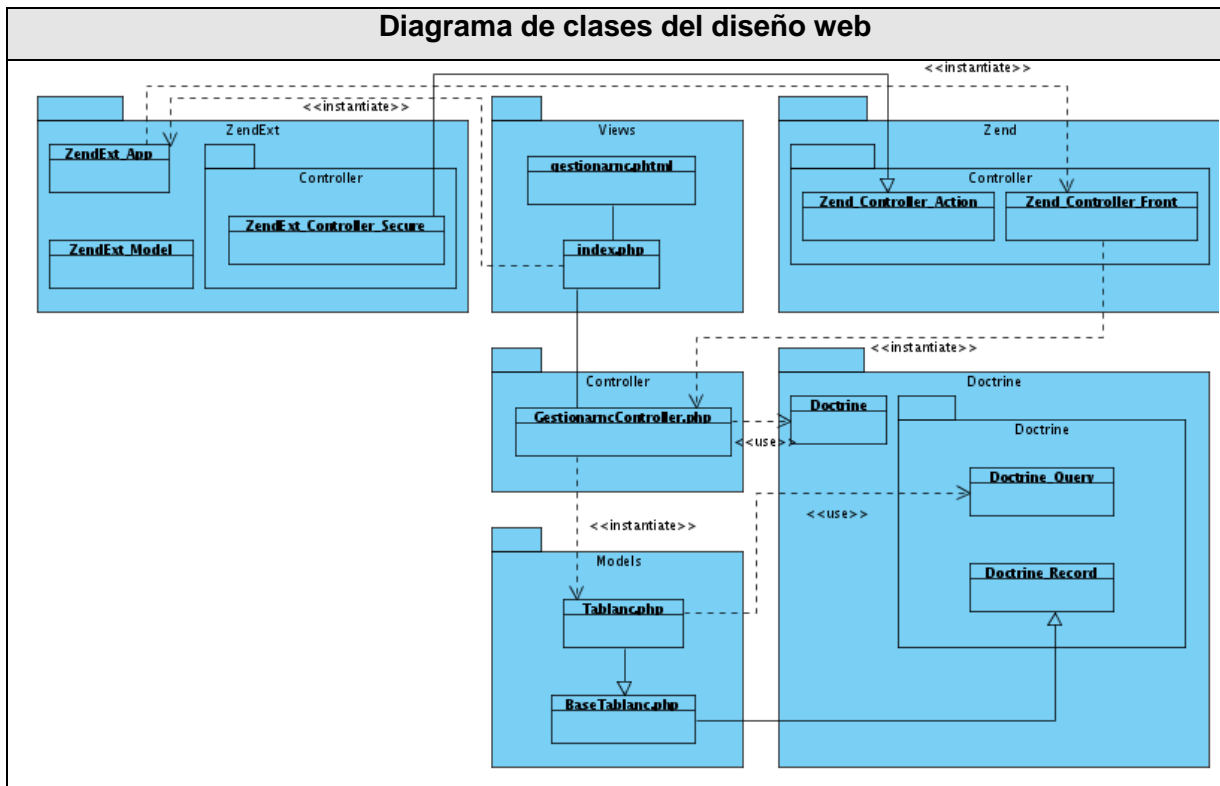


Figura 3.2 Diagrama de clases del diseño “Gestionar No conformidades”

3.3.2. Diagramas de secuencia

Los diagramas de interacción se dividen en dos tipos, los diagramas de secuencia y los diagramas de colaboración, a continuación se presentan los diagramas de secuencia correspondientes a los casos de uso que se identificaron en el sistema. Para continuar con el ejemplo ilustrativo de los epígrafes anteriores se adjunta el diagrama de secuencia del caso de uso “Adicionar No conformidades”, por la extensión de los mismos se adjuntaron solamente los de la función Adicionar de todos los CU del sistema, en caso del CU Generar parte que no posee una función Adicionar se adjunta el generar parte. Los diagramas mencionados se encuentran en los Anexos, del trabajo bajo el subtítulo Anexos del Capítulo 3 “Diagrama de secuencia.” El resto de las funcionalidades de los CU se encuentran en el material que se adjunta al presente trabajo de diploma.

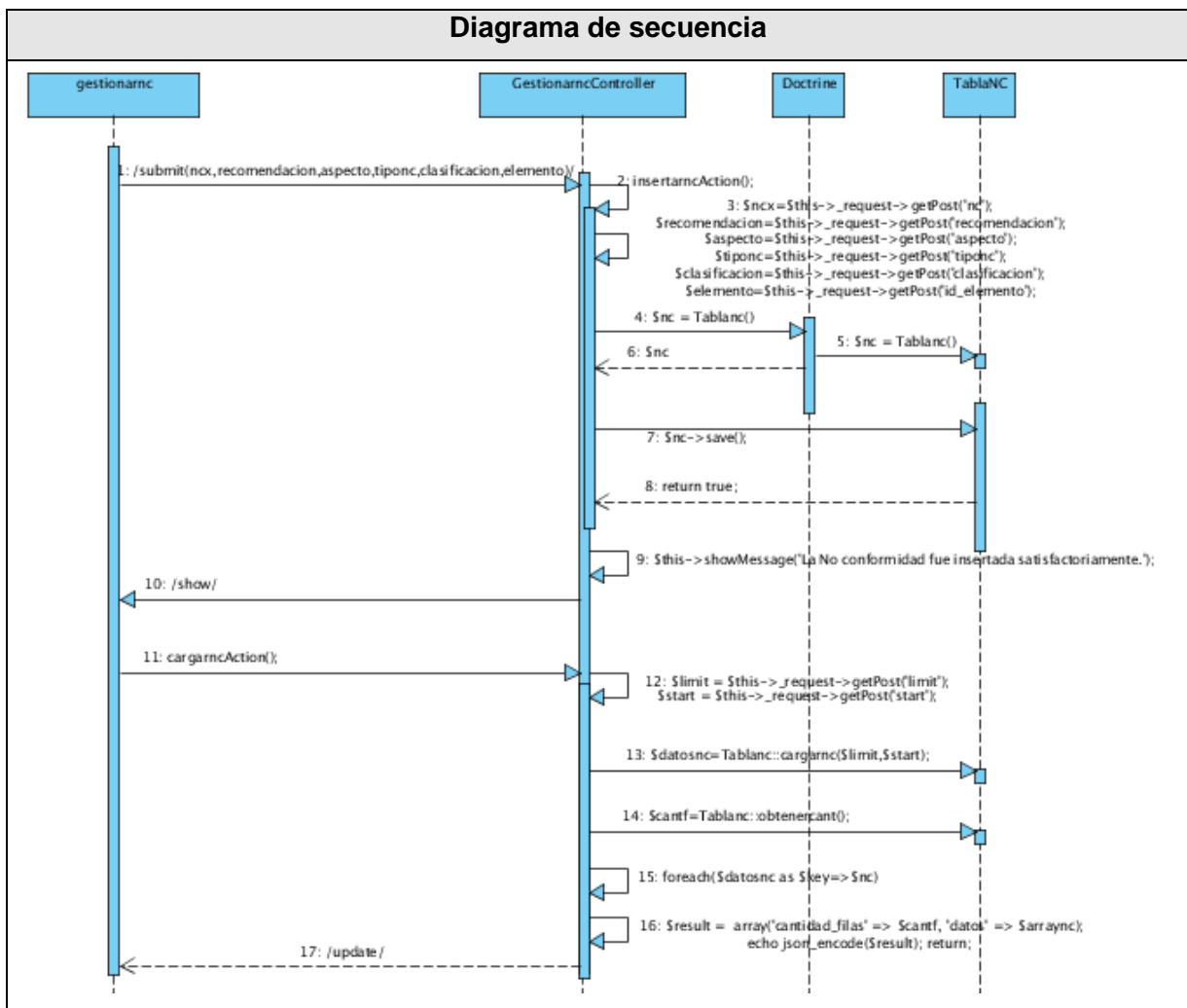


Figura 3.3 Diagrama de secuencia “Gestionar No conformidades”

3.3.3. Diagrama de clases persistentes

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

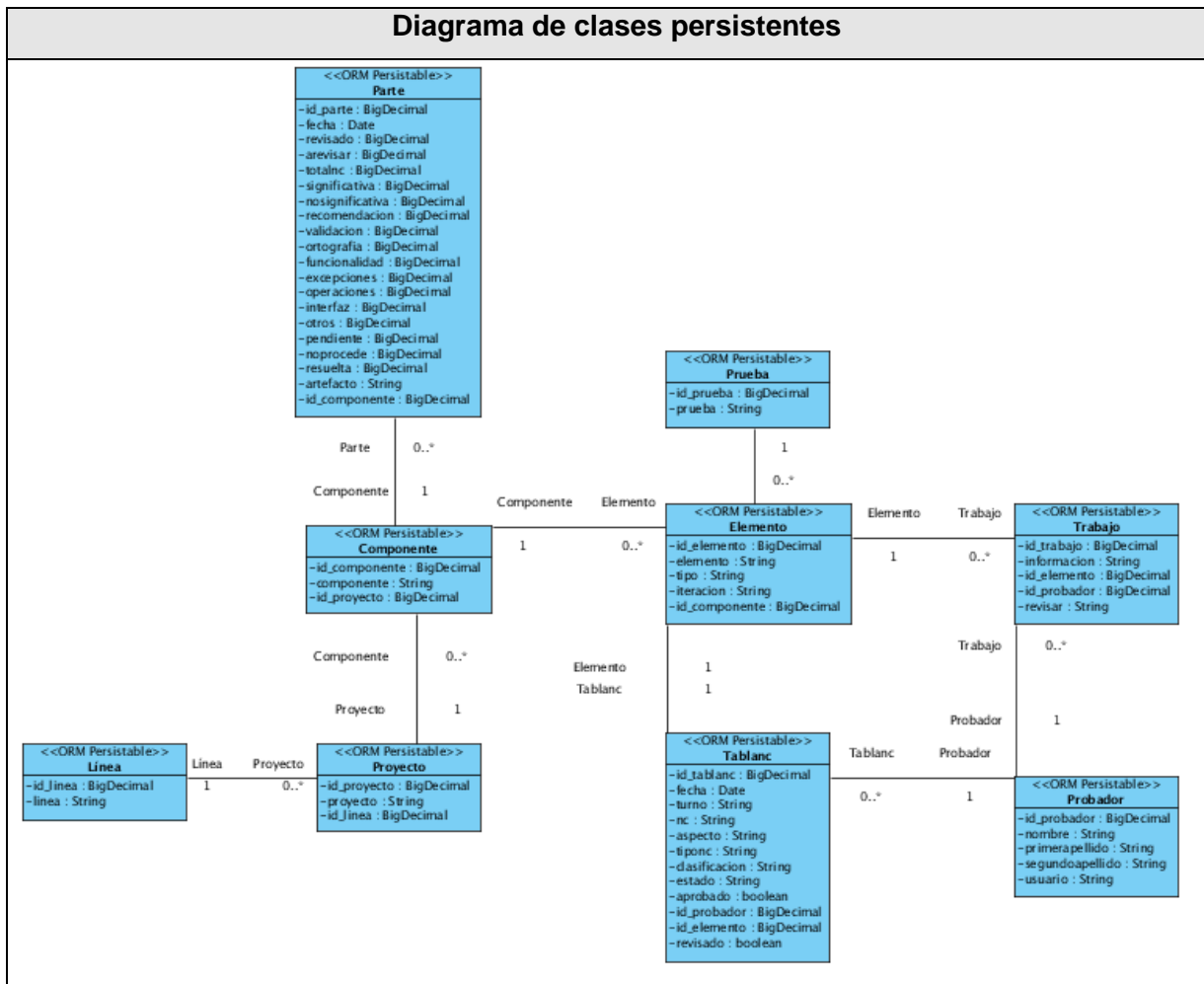


Figura 3.4 Diagrama de clases persistentes

3.4. Conclusiones

Se realizó el proceso de Modelamiento de las fases de Análisis y Diseño propuestas por RUP, se generaron los Diagramas de clases del análisis, los Diagramas de clases del diseño con estereotipos web, los Diagramas de secuencia y el Diagrama de clases persistentes. Con la elaboración de estos artefactos, se concluye el capítulo con una visión más detallada y precisa de lo que será el sistema que se llevará a cabo en la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

Posterior a la fase de Análisis y Diseño se sucede la fase de Implementación del sistema. Dentro de esta fase existen una serie de artefactos que tienen como elementos de entrada para su confección, los desarrollados en la fase anterior, con el objetivo de ir conformando el proyecto como un sistema completo y con un acabado a la altura de los requerimientos previamente definidos.

Los objetivos de este capítulo en la fase de Implementación son, representar el Modelo de datos y los Diagramas despliegue y componente. En la fase de Prueba, se abordará sobre el tipo de prueba que se realizarán a la aplicación.

4.2. Implementación



Figura 4.1 Patrón Modelo - Vista - Controlador.

Para llevar a cabo la implementación del sistema se trabajó con el patrón de arquitectura Modelo - Vista - Controlador.

El Modelo - Vista - Controlador se divide en tres partes. En el Modelo se representa la información con la que se trabaja, es decir, son una serie de datos de los cuales se pueden derivar datos nuevos a través de operaciones. La Vista es la representación del Modelo en términos óptimos para la interacción con el usuario, usualmente como la interfaz de usuario. La parte del Controlador, responde a acciones o eventos solicitados por el usuario generalmente en la Vista y que invocan cambios o modificaciones en el Modelo.

4.2.1. Modelo de datos

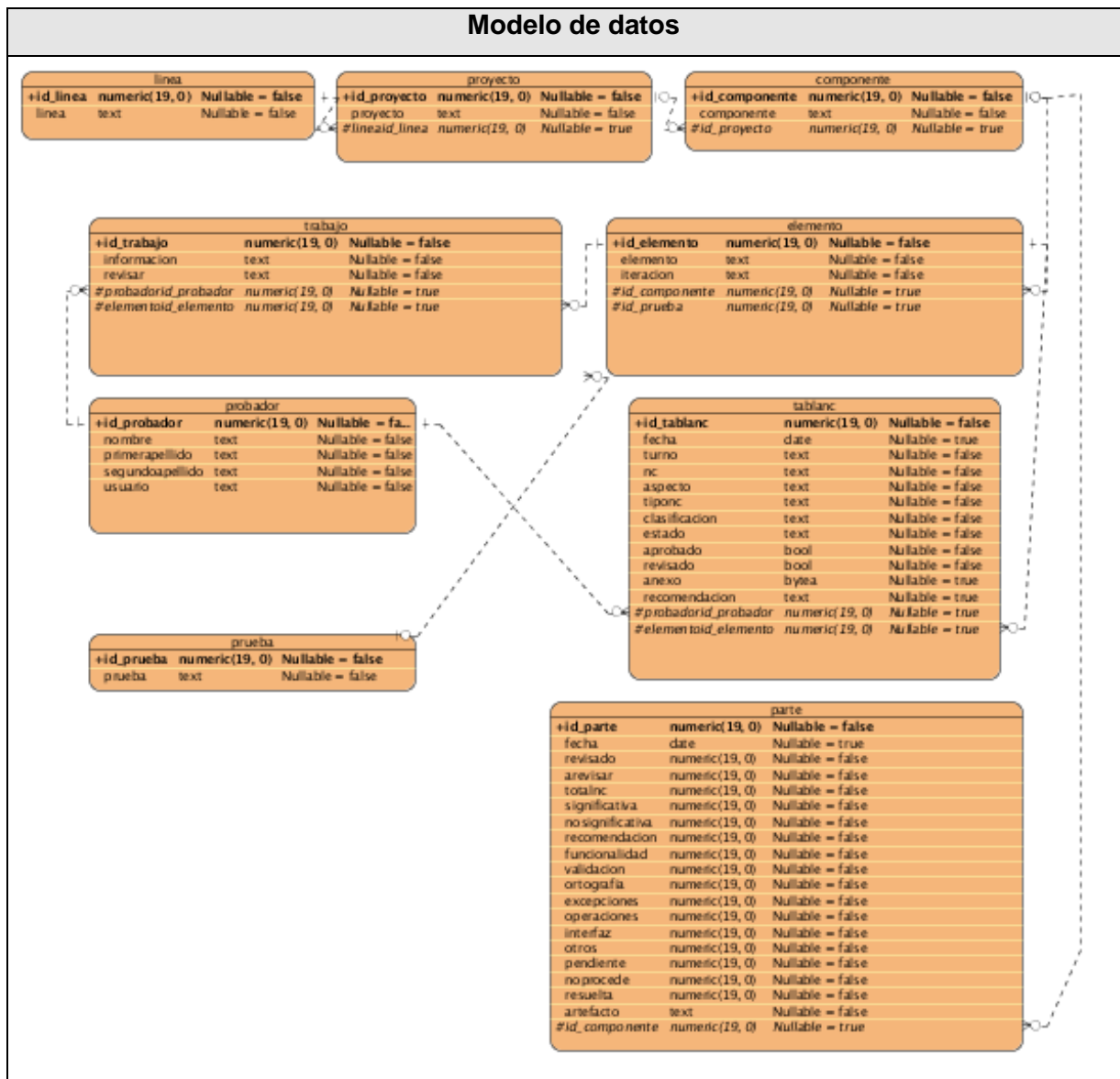


Figura 4.2 Modelo de datos.

4.2.2. Modelo de despliegue

El diagrama de despliegue ilustra la representación física de los elementos con los que deberá contar el sistema, los componentes de este diagrama son los nodos y las relaciones entre ellos, estas relaciones pueden ser de diferentes tipos dependiendo de lo que represente el nodo.

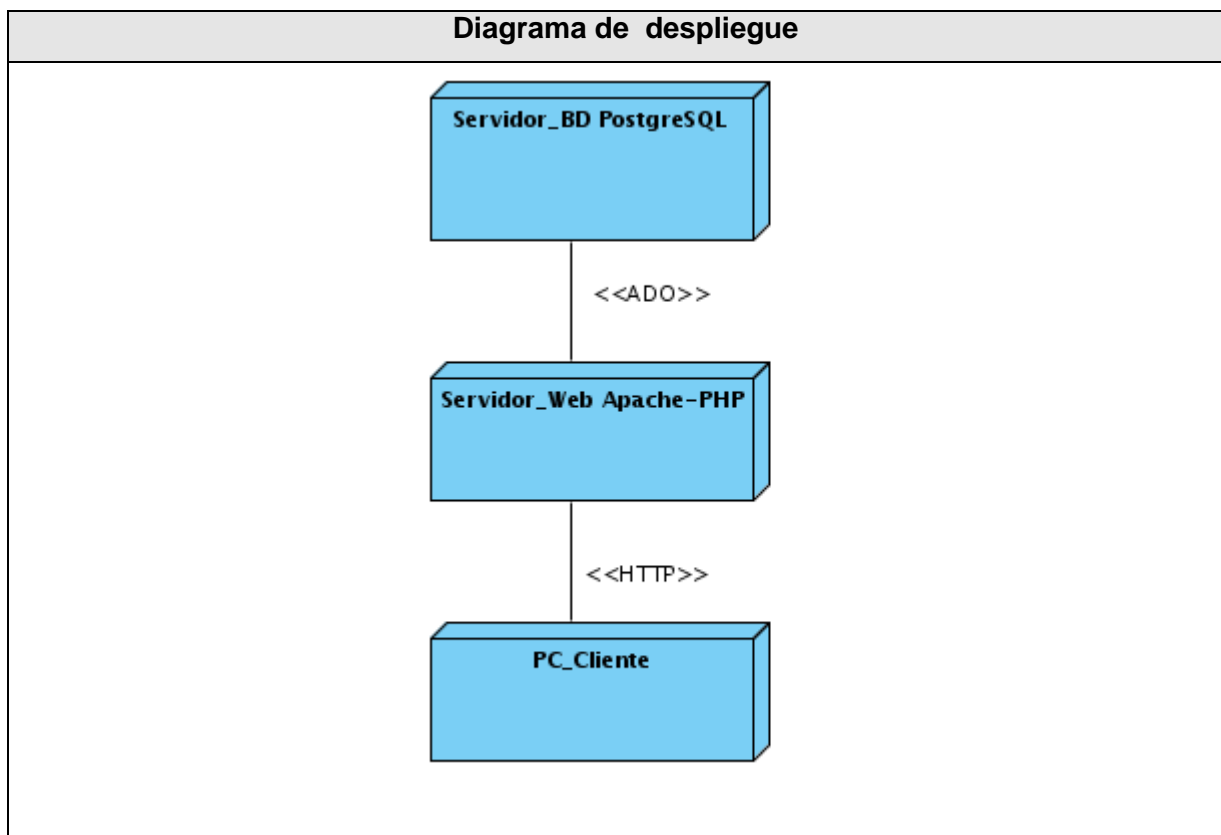


Figura 4.3 Diagrama de despliegue.

4.2.3. Ambiente de implantación

Para el correcto funcionamiento de la aplicación son necesarias una serie de características para el ambiente donde será implantada, a continuación se precisan las particularidades:

Para instalar el **Qualitás Sistema de Gestión de No conformidades** se precisa una Computadora Personal (PC) servidor de 512 MB de memoria RAM y un procesador de 3.00GHZ, con Sistema Operativo Linux o Microsoft Windows Server 2003, o superior. En la PC

servidor se necesita además la instalación del Gestor de Base de Datos PostgreSQL en su versión 8.3 y el Servidor Web Apache.

La PC clientes para un buen rendimiento se aconseja tener como mínimo 1.3 GHZ de velocidad de procesamiento y 256 MB de RAM, podrán tener cualquier Sistema Operativo siempre y cuando se tenga instalado y configurado el navegador web Mozilla Firefox en la versión 3.0.

Se recomienda para la utilización del sistema por parte de la administración, personas con conocimientos informáticos, así como familiarizadas con el proceso de Prueba que se realiza en el CCES. Los clientes del sistema por su parte deberán tener conocimientos básicos sobre ambientes Web así como de la importancia de las prueba dentro del proceso de desarrollo de software.

4.2.4. Diagrama de componentes

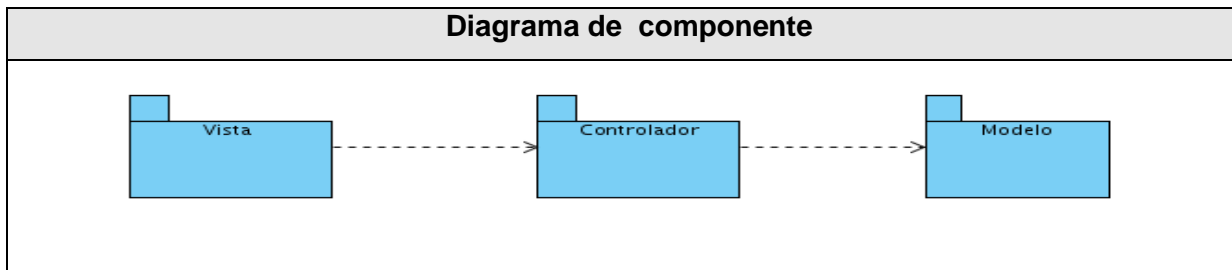


Figura 4.4 Vista general diagrama de componentes.



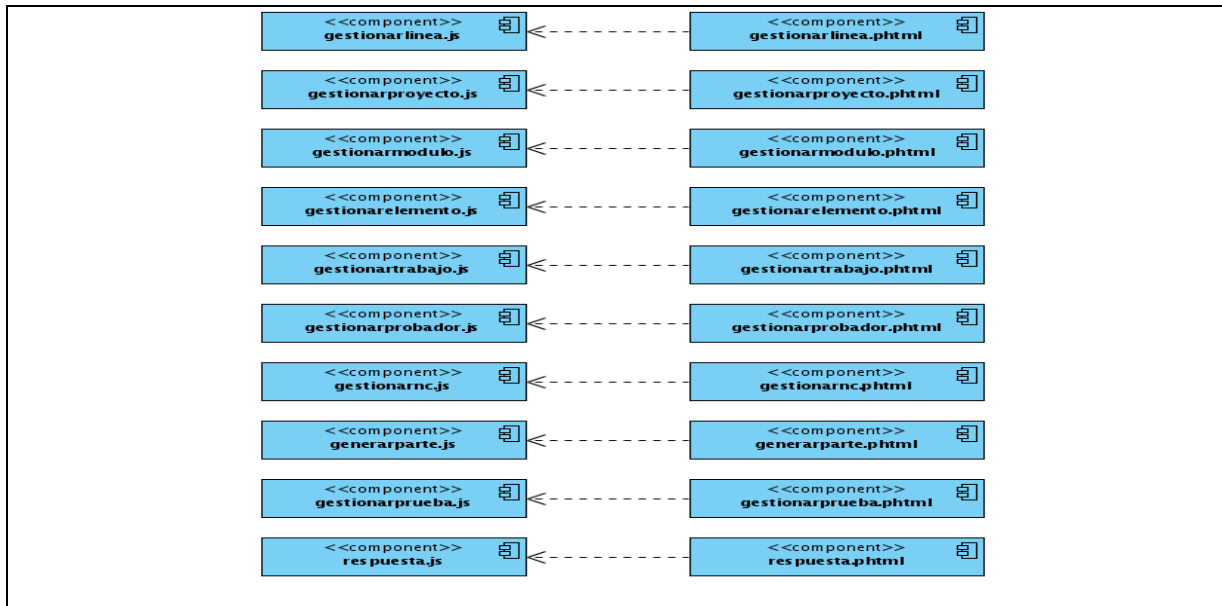


Figura 4.5 Detalles de la Vista.

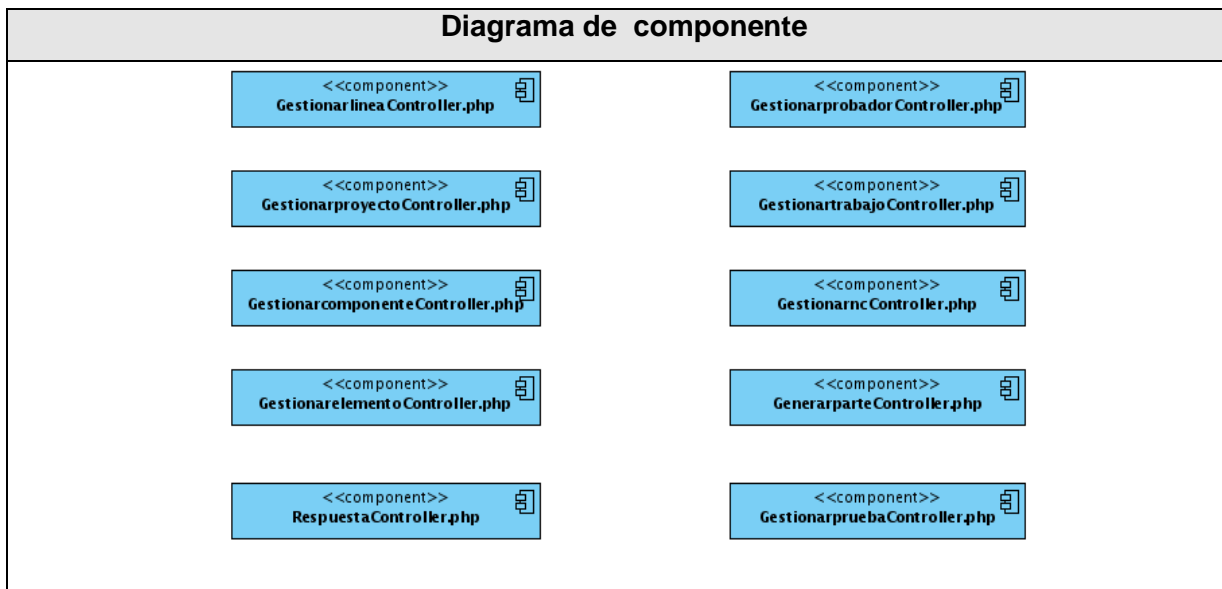


Figura 4.6 Detalles del Controlador.



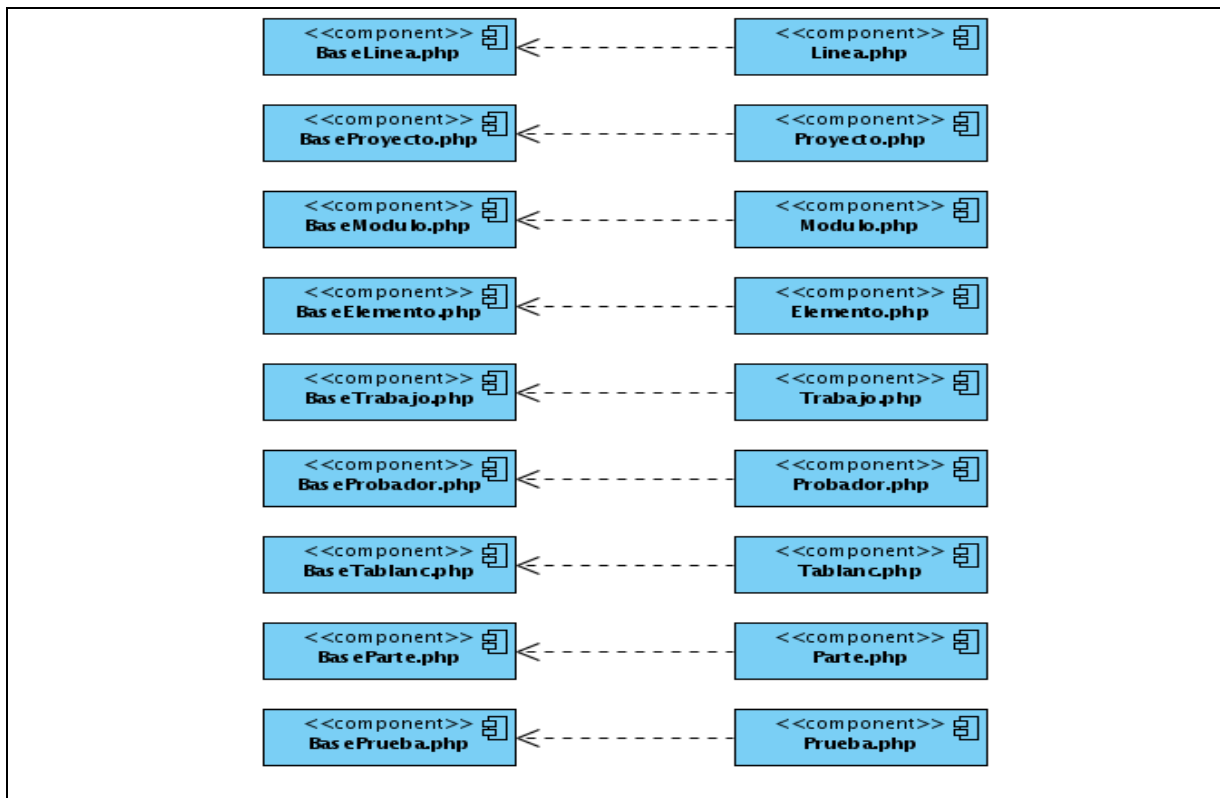


Figura 4.7 Detalles del Modelo.

4.3. Prueba

Existen en la actualidad varios tipos de pruebas de software, todas están destinadas a encontrar la mayor cantidad de defectos para ofrecer un producto de óptima calidad, las pruebas no pueden probar la ausencia de defectos, sino detectar la existencia de estos. Entre los principales métodos de prueba se encuentran, pruebas de **Caja Blanca** y de **Caja Negra**, las primeras se centran en la revisión del código fuente del software y requieren conocimiento del funcionamiento interno del sistema. Las últimas por su parte se refieren a las pruebas que se realizan a la interfaz del sistema, son conocidas también como pruebas de comportamiento y se basan en los requerimientos funcionales del sistema.

Para la comprobación final de **Qualitás, Sistema de Gestión de No conformidades** se realizaron las siguientes pruebas, a través de las cuales se verificaron todos los requisitos funcionales del sistema:

- **Pruebas unitarias** por parte del equipo de desarrollo.
- **Pruebas de liberación** con un pequeño despliegue en un escenario controlado.
- **Pruebas de aceptación** del cliente y despliegue piloto en un escenario controlado.
- **Pruebas de aceptación** con usuarios finales (piloto) en un escenario real.

Las **Prueba de Aceptación** son el conjunto de actividades desarrolladas en un entorno controlado, para verificar el cumplimiento y calidad en la implementación de las funcionalidades pactadas previamente de las **Aplicaciones Informáticas (AI)**.

Las **Pruebas Piloto** son el conjunto de actividades, desarrolladas en un entorno real, para verificar el cumplimiento y calidad en la implantación de las funcionalidades pactadas en la **Solución** para cada etapa.

4.3.1. Diseños de caso de prueba

Los diseños de casos de prueba son una técnica para probar el sistema mediante el método de Caja Negra, se pueden elaborar por CU o por requisitos y su objetivo fundamental es ofrecer al usuario una guía detallada de cómo realizar la prueba, mostrando los pasos a seguir y los datos a introducir con la finalidad de obtener el resultado esperado.

A continuación se adjunta una breve descripción de los DCP utilizados durante las pruebas de aceptación. Los DCP completos se encuentran en el material adjunto al trabajo de diploma.

Gestionar línea

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar línea
Propósito	Probar que se puede adicionar una nueva línea de trabajo al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar línea .
Entrada	Se intenta adicionar una nueva línea de trabajo con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona la línea de trabajo, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar línea
Propósito	Probar que se puede eliminar una línea de trabajo del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar línea .
Entrada	Se intenta eliminar una línea de trabajo.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina la línea, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar línea
Propósito	Comprobar que se puede modificar una línea de trabajo en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar línea .
Entrada	Se intenta modificar una línea de trabajo con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica la línea de trabajo, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

DCP Gestionar proyecto

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar proyecto
Propósito	Probar que se puede adicionar un nuevo proyecto al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar proyecto .
Entrada	Se intenta adicionar un nuevo proyecto con datos válidos e inválidos.

Resultado esperado	Si las entradas son válidas el sistema adiciona el proyecto, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar proyecto
Propósito	Probar que se puede eliminar un proyecto del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar proyecto.
Entrada	Se intenta eliminar un proyecto.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el proyecto, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar proyecto
Propósito	Comprobar que se puede modificar un proyecto en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar proyecto.
Entrada	Se intenta modificar un proyecto con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el proyecto, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Gestionar componente

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar componente
Propósito	Probar que se puede adicionar un nuevo componente al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar componente .
Entrada	Se intenta adicionar un nuevo componente con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona el componente, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar componente
Propósito	Probar que se puede eliminar un componente del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar componente .
Entrada	Se intenta eliminar un componente.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el componente, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar componente
Propósito	Comprobar que se puede modificar un componente en el sistema.

Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar componente .
Entrada	Se intenta modificar un componente con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el componente, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

DCP Gestionar tipo de prueba

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar tipo de prueba
Propósito	Probar que se puede adicionar un nuevo tipo de prueba al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar tipo de prueba .
Entrada	Se intenta adicionar un nuevo tipo de prueba con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona el tipo de prueba, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar el tipo de prueba
Propósito	Probar que se puede eliminar un tipo de prueba del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar tipo de prueba .
Entrada	Se intenta eliminar un componente.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el tipo de prueba, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar tipo de prueba
Propósito	Comprobar que se puede modificar un tipo de prueba en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar tipo de prueba.
Entrada	Se intenta modificar un tipo de prueba con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el tipo de prueba, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

DCP Gestionar elemento

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar elemento
Propósito	Probar que se puede adicionar un nuevo elemento al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar elemento.
Entrada	Se intenta adicionar un nuevo elemento con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona el elemento, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar elemento
Propósito	Probar que se puede eliminar un elemento del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar elemento.
Entrada	Se intenta eliminar un elemento.

Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el elemento, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar elemento
Propósito	Comprobar que se puede modificar un elemento en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar elemento.
Entrada	Se intenta modificar un elemento con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el elemento, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

DCP Gestionar probador

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar probador
Propósito	Probar que se puede insertar un nuevo probador al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Calidad/ Gestionar probador.
Entrada	Se intenta adicionar un nuevo probador con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona el probador, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar probador

Propósito	Probar que se puede eliminar un probador del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar probador.
Entrada	Se intenta eliminar un probador.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el probador, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar probador
Propósito	Comprobar que se puede modificar un probador en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar probador.
Entrada	Se intenta modificar un probador con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el probador, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Gestionar trabajo

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Adicionar trabajo
Propósito	Probar que se puede insertar un nuevo trabajo al sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar trabajo.
Entrada	Se intenta adicionar un nuevo trabajo con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema adiciona el trabajo, si las entradas son inválidas el sistema subraya el campo inválido de rojo.

Resultado de la prueba	
-------------------------------	--

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar trabajo
Propósito	Probar que se puede eliminar un trabajo del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar trabajo.
Entrada	Se intenta eliminar un trabajo.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina el trabajo, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar trabajo
Propósito	Comprobar que se puede modificar un trabajo en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar trabajo.
Entrada	Se intenta modificar un trabajo con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica el trabajo, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Repartir trabajo
Propósito	Comprobar que se puede repartir el trabajo entre los probadores.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar trabajo.
Entrada	Se intenta repartir un trabajo con datos válidos e inválidos.

Resultado esperado	Si las entradas son válidas el sistema reparte el trabajo correctamente, si las entradas son inválidas el sistema subraya de rojo el campo inválido.
Resultado de la prueba	

DCP Gestionar No conformidades

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Eliminar No conformidades
Propósito	Probar que se puede eliminar No conformidades del sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar No conformidades.
Entrada	Se intenta eliminar una No conformidad.
Resultado esperado	Si se presiona el botón Aceptar el sistema elimina la No conformidad, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Modificar No conformidades
Propósito	Comprobar que se puede modificar una No conformidad en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar No conformidades.
Entrada	Se intenta modificar una línea de trabajo con datos válidos e inválidos.
Resultado esperado	Si las entradas son válidas el sistema modifica la No conformidad, si las entradas son inválidas el sistema subraya el campo inválido de rojo.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Revisar No conformidades
Propósito	Comprobar que se puede marcar como revisada una No conformidad en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar No conformidades.
Entrada	Se intenta revisar una No conformidad en el sistema.
Resultado esperado	Si se presiona el botón Aceptar el sistema marca como revisada la No conformidad, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Revisar No conformidades
Propósito	Comprobar que se puede marcar como revisada una No conformidad en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar No conformidades.
Entrada	Se intenta revisar una No conformidad en el sistema.
Resultado esperado	Si se presiona el botón Aceptar el sistema marca como revisada la No conformidad, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Aprobar No conformidades
Propósito	Comprobar que se puede marcar como aprobada una No conformidad en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Gestionar No conformidades.
Entrada	Se intenta aprobar una No conformidad en el sistema.

Resultado esperado	Si se presiona el botón Aceptar el sistema marca como aprobada la No conformidad, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Respuesta a la No conformidades
Propósito	Comprobar que se puede dar respuesta a una No conformidad.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Respuesta a las No conformidades.
Entrada	Se intenta dar respuesta a una No conformidad.
Resultado esperado	Si se presiona el botón Aceptar el sistema marca con un estado diferente a la No conformidad, si se presiona el botón Cancelar el sistema cancela la operación.
Resultado de la prueba	

Generar partes

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Generar parte
Propósito	Comprobar que se puede generar un parte en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Generar partes.
Entrada	Se intenta generar un parte.
Resultado esperado	El sistema muestra el parte.
Resultado de la prueba	

Pruebas de aceptación: Diseño de casos de prueba.	
Requisito	Obtener reporte
Propósito	Comprobar que se puede obtener un reporte en el sistema.
Condiciones de ejecución	El usuario debe tener los permisos necesarios para ejecutar el requisito. Se debe seleccionar el menú Inicio/ Qualitás/ Generar partes.
Entrada	Se intenta obtener un reporte.
Resultado esperado	El sistema abre la aplicación PADTSI y crea el reporte.
Resultado de la prueba	

4.4. Conclusiones

En el capítulo se desarrollaron las fases de Implementación y Prueba, se realizaron los artefactos correspondientes de cada una, se definió el ambiente de implantación para la aplicación y se efectuaron los DCP a aplicar en las pruebas piloto al sistema, las cuales resultaron satisfactorias, teniendo muy buena aceptación el sistema por parte de los usuarios finales.

CONCLUSIONES

Posterior al estudio que se realizó del estado del arte referente a las herramientas de gestión de No conformidades existentes en Cuba y en el mundo, se llevó a cabo la implementación de una aplicación web que ofrece una serie de funcionalidades con el objetivo de facilitar los procesos del Laboratorio de pruebas. Para ello se realizó también:

- Un análisis de las principales tecnologías para el desarrollo de aplicaciones web que permitió seleccionar las herramientas y metodologías para desarrollar el sistema.
- Un estudio de los procesos existentes dentro del Laboratorio de prueba del CCES que propició amplio conocimiento del negocio y de los procesos a informatizar.
- El desarrollo del análisis, el diseño y la implementación de una aplicación web que permite la gestión de las No conformidades que sean detectadas en el proceso de prueba del CCES.
- La aprobación de la solución propuesta por parte del Cliente, proceso en el cual surgieron una serie de recomendaciones para propiciar el perfeccionamiento futuro del sistema.

De acuerdo a las consideraciones expuestas anteriormente es posible afirmar que se alcanzó de manera satisfactoria el objetivo propuesto: informatizar los procesos asociados a la gestión de No conformidades en el laboratorio de pruebas.

RECOMENDACIONES

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

- Implantar el sistema en el laboratorio de pruebas de la Subdirección de Calidad del ERP.
- Valorar la inclusión de nuevos componentes a la aplicación, según las necesidades y a partir del estudio de las aplicaciones descritas en el estado del arte, a fin de elevar las prestaciones del sistema propuesto.
- Se sugiere actualizar y ofrecer soporte a la aplicación en consecuencia a la evolución del marco de trabajo hacia nuevas tecnologías y herramientas.

BIBLIOGRAFÍA

1. **Telemática A.** [En línea] [Citado el: 28 de abril de 2009.]
http://www.fiec.espol.edu.ec/resources/download/revista/Telematica_A%C3%B1oVII_No15.pdf.
2. [En línea] [Citado el: 2009 de abril de 12.] http://pooteeweet.org/files/ipc08spring/doctrine_orm.pdf.
3. [En línea] [Citado el: 10 de marzo de 2009.] <http://extjs.com/>.
4. [En línea] [Citado el: 25 de abril de 2009.] <http://www.doctrine-project.org/>.
5. [En línea] [Citado el: 6 de marzo de 2009.] <http://extjsref.blogspot.com/2008/07/01-introduccion.html>.
6. **Emilia Reggio.** [En línea] <http://www.blulink.com/en/software/quality-system/quarta-non-conformity.html>.
7. *Doctrine.* [En línea] [Citado el: 25 de abril de 2009.]
http://www.doctrine-project.org/documentation/manual/1_0/en/introduction.
8. Especificación de Requerimientos de Software (SRS). [En línea] [Citado el: 6 de marzo de 2009.]
<http://readysset.tigris.org/nonav/es/templates/srs.html>.
9. KMKey. [En línea] <http://www.kmkey.com/>.
10. **Godino Bueno, Cristina.** [En línea] 2006. <http://www.lcc.uma.es/pfc/412.pdf>.
11. **Cesar García.** [En línea] 22 de julio de 2008.
<http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript/>.
12. **Zulueta, Delmys Pozo y Palenzuela, Nilien Cruz.** Herramienta para informatizar los procesos en el Laboratorio de Pruebas de Liberación. Ciudad de la Habana : s.n., 2008.
13. **Rodríguez, Yadainy Betancourt.** Postgrado Metodología de la Investigación Sistema para la Gestión de Opiniones. Ciudad de la Habana : s.n., 2008.
14. **López, Leyvis Luis Valdes y Ortiz, Andry Leal.** Sistema de Gestión y Control de Métricas. Ciudad de la Habana : s.n., 2008.
15. **Roman S. Borschel; Guilherme Blanco; Jonathan H. Wage.** [En línea] 2009.
<http://www.quizzpot.com/2009/01/ext-js-framework/>.
16. [En línea] 2009. [Citado el: 7 de mayo de 2009.]
<http://swik.net/json/del.icio.us%2Ftag%2Fjson/Getting+started+with+ExtJS+and+Zend+Framework/b8t6i>.
17. Naneau. [En línea] 2006 . <http://naneau.nl/playground/grid/>.
18. Infojobs. [En línea] [Citado el: 12 de abril de 2009.] <http://madrid.oferta.infojobs.net/profesional-php-zend-framework-extjs-jquery/of-i491536354330084152054951107045>.
19. Omdir. [En línea] <http://osdir.com/ml/php.eaccelerator.user/2007-01/msg00000.html>.
20. [En línea] <http://wareseeker.com/Software-Development/zend-extension-0.4.0-beta.zip/318530>.
21. [En línea] <http://www.devpapers.com/>.
22. Softpedia. [En línea]

<http://www.softpedia.com/get/Programming/Components-Libraries/Zend-Extension.shtml>.

23. Zend Technologies. [En línea] 1999-2009 .

<http://www.mistersoft.org/freelancing/getafreelancer/2008/12/PHP-System-Admin-adding-a-zend-extension-in-php-ini-file-357886.html>.

24. Simfony.es. [En línea] <http://www.symfony.es/categoria/plugins/page/2/>.

25. Slideshare. [En línea] <http://www.slideshare.net/atix/revista-de-software-libre-atix-numero-8>.

26. [En línea] http://www.oc.ccn.cni.es/PO-002-noconformidades_es.html.

27. [En línea] <http://www.redanagos.es/doctecnicos/punto5.2.1.5-Gu%C3%ADa%20Calidad.pdf>.

28. Maestros del web. [En línea] <http://www.maestrosdelweb.com/editorial/herramientas-adecuadas-para-el-diseno-y-desarrollo-de-un-sitio-web/>.

29. Cristalab. [En línea] <http://www.cristalab.com/tips/herramientas-para-diseno-web-en-linea-c23204/>.

30. [En línea] www.deepinphp.com.

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

32. [En línea] <http://www.desarrolloweb.com/articulos/243.php>.

33. **Lic. Ingrid Riquelme Ramírez.** [En línea]

http://www.astraph.com/udl/biblioteca/antologias/desarrollo_pag_web.pdf.

34. [En línea] http://www.ecicomputacion.com.ar/php___bases_de_datos.html.

35. [En línea] <http://www.softwarelibre.venezuela.net.ve/index.php>.

36. [En línea] <http://www.monografias.com/trabajos66/introduccion-php/introduccion-php2.shtml>.

37. [En línea] <http://150.185.75.30/atiwiki/index.php/.NE>.

38. [En línea] http://www.ciberaula.com/curso/masterphp/que_es/.

39. [En línea] <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.

40. LA PUBLICACIÓN WEB PER A LA GESTIÓN DE FORMACIÓN INFORMÁTICA A TERRASSA. [En línea] http://www.recercat.net/bitstream/2072/8966/1/PFC_LuqueRubio.pdf.

41. Curso básico de PHP. [En línea] http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos_PHP/Introduccion.htm.

42. Base de Datos. [En línea] <http://www.dspace.espol.edu.ec/bitstream/123456789/1416/1/2751.pdf>.

43. Sistema de Gestión de Base de Datos. [En línea] <http://www.monografias.com/trabajos56/sistemas-bases-de-datos/sistemas-bases-de-datos2.shtml>.

44. PostgreSQL Vs My SQL. [En línea] <http://wolfkira.wordpress.com/2008/02/28/postgresql-vs-mysql>.

45. Estudio de diseño web y diseño gráfico. [En línea]

http://www.ideocentro.com/recursos.php?id=13&estudio_diseno_web_ideocentro=Mysql.

46. Grupo Usuario Linux de Nicaragua . [En línea]

http://www.linux.org.ni/modules.php?name=News&new_topic=4.

47. Qué es Oracle. [En línea] <http://www.desarrolloweb.com/articulos/840.php>.
48. El CoDiGo K » Oracle. [En línea] <http://www.elcodigok.com.ar/category/bases-de-datos/oracle/>.
49. GESTOR DE CV ONLINE. [En línea]
<http://upcommons.upc.edu/pfc/bitstream/2099.1/5180/1/memoria%20-%20GESTOR%20CV%20online.pd>.
50. Introducción a UML . [En línea] http://www.vico.org/aRecursosPrivats/TRAD_introUML.pdf.
51. Cursos de Dreamweaver . [En línea] <http://ar.emagister.com/cursos-dreamweaver-kwes-290.htm>.
52. Sección de paradigma visual . [En línea]
http://www.freedownloadmanager.org/es/downloads/paradigma_visual_libre/.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Modelo Cliente Servidor [Disponible en: <<http://www.deepinphp.com/2007/06/17/el-modelo-cliente-servidor>> [Fecha de consulta: 15 enero 2009]
- [2] Definición arquitectura cliente servidor [Disponible en: <<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>> [Fecha de consulta: 15 enero 2009]
- [3] Desarrollo de Páginas Web. Lic. Ingrid Riquelme Ramírez [Disponible en: <http://www.astraph.com/udl/biblioteca/antologias/desarrollo_pag_web.pdf> [Fecha de consulta: 18 enero 2009]
- [4] Lenguajes de lado servidor [Disponible en: <<http://www.desarrolloweb.com/articulos/243.php>> [Fecha de consulta: 18 enero 2009]
- [5] Programación Web: PHP + Base de Datos [Disponible en: <http://www.ecicomputacion.com.ar/php___bases_de_datos.html> [Fecha de consulta: 18 enero 2009]
- [6] PHP [Disponible en: <<http://www.softwarelibre.venezuela.net.ve/index.php>> [Fecha de consulta: 18 enero 2009]
- [7] Breve historia de PHP [Disponible en: <<http://www.monografias.com/trabajos66/introduccion-php/introduccion-php2.shtml>> [Fecha de consulta: 18 enero 2009]
- [8] .NET [Disponible en: <<http://150.185.75.30/atiwiki/index.php/.NET>> [Fecha de consulta: 19 enero 2009]
- [9] Master en PHP (PHP, MySQL, XML) [Disponible en: <http://www.ciberaula.com/curso/masterphp/que_es/> [Fecha de consulta: 19 enero 2009]
- [10] Internet Information Services [Disponible en: <<http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>> [Fecha de consulta: 19 enero 2009]
- [11] A PUBLICACIÓN WEB PER A LA GESTIÓN DE FORMACIÓN INFORMÁTICA A TERRASSA [Disponible en: <<http://www.recercat.net/bitstream/2072/8966/1/PFCLuqueRubio.pdf>> [Fecha de consulta: 20 enero 2009]
- [12] Curso básico de PHP [Disponible en: <http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos_PHP/Introduccion.htm> [Fecha de consulta: 20 enero 2009]
- [13] Bases de Datos [Disponible en: <<http://www.dspace.espol.edu.ec/bitstream/123456789/1416/1/2751.pdf>> [Fecha de consulta: 20 enero 2009]

- [14] Sistemas de gestión de bases de datos [Disponible en:
<<http://www.monografias.com/trabajos56/sistemas-bases-de-datos/sistemas-bases-de-datos2.shtml>>
[Fecha de consulta: 20 enero 2009]
- [15] PostGreSQL vs. MySQL [Disponible en: <<http://wolfkira.wordpress.com/2008/02/28/postgresql-vs-mysql>> [Fecha de consulta: 20 enero 2009]
- [16] Estudio de diseño web y diseño gráfico [Disponible en:
<http://www.ideocentro.com/recursos.php?id=13&estudio_diseno_web_ideocentro=Mysql> [Fecha de consulta: 20 enero 2009]
- [17] Grupo Usuario Linux de Nicaragua [Disponible en:
<http://www.linux.org.ni/modules.php?name=News&new_topic=4> [Fecha de consulta: 21 enero 2009]
- [18] Qué es Oracle [Disponible en: <<http://www.desarrolloweb.com/articulos/840.php>> [Fecha de consulta: 21 enero 2009]
- [19] El CoDiGo K » Oracle [Disponible en: <<http://www.elcodigok.com.ar/category/bases-de-datos/oracle/>>
[Fecha de consulta: 21 enero 2009]
- [20] GESTOR DE CV ONLINE [Disponible en:
<<http://upcommons.upc.edu/pfc/bitstream/2099.1/5180/1/memoria%20-20GESTOR%20CV%20online.pdf>>
[Fecha de consulta: 21 enero 2009]
- [21] Introducción a UML [Disponible en: <http://www.vico.org/aRecursosPrivats/TRAD_introUML.pdf>
[Fecha de consulta: 21 enero 2009]
- [22] Cursos de Dreamweaver [Disponible en: <<http://ar.emagister.com/cursos-dreamweaver-kwes-290.htm>>
[Fecha de consulta: 21 enero 2009]
- [23] Sección de paradigma visual [Disponible en:
<http://www.freedownloadmanager.org/es/downloads/paradigma_visual_libre/> [Fecha de consulta: 21 enero 2009]

ANEXOS

5.1. Anexos del Capítulo 2 “Especificación textual de los CU”

5.1.1. Especificación textual del CU “Gestionar línea.”

Asociación de Uso	Gestionar línea	
Actores	Jefe de turno	
Propósito	Permitir al usuario insertar, modificar o eliminar una línea.	
Resumen	El usuario inserta, elimina o modifica una línea.	
CU asociados		
Precondiciones	El actor debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	El sistema contiene líneas insertadas o modificadas.	
Referencias	RF1.	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el menú Inicio/Qualitás , la opción a Gestionar línea .	2. El sistema muestra la interfaz línea.	
3. El actor selecciona el botón Adicionar . -Si el actor selecciona la opción Modificar (Ver Sección I) -Si el actor selecciona la opción Eliminar (Ver Sección II)	4. El sistema muestra una interfaz con los campos necesarios para adicionar una nueva línea.	
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	6. El sistema valida los datos que se introducen.	
	7. El sistema muestra un mensaje indicando que la línea ha sido insertada correctamente.	
	8. El sistema actualiza el listado de las líneas.	
Cursos Alternos		
Línea 5: El actor presiona el botón Aplicar , el sistema actualiza el listado de las líneas existentes y mantiene la interfaz abierta (Ver Sección I).		

Línea 5: El actor presiona el botón Cancelar y sistema cierra la interfaz.	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre la línea que desea modificar y selecciona el botón Modificar .	2. El sistema muestra una interfaz con los datos que posee la línea.
3. El actor modifica los datos de la línea y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que los datos de la línea fueron actualizados correctamente.
	5. El sistema actualiza la lista de probadores.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y sistema cierra la interfaz.	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la línea que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar del mensaje de confirmación.	4. El sistema muestra un mensaje indicando que la línea fue eliminada de manera correcta.
	5. El sistema actualiza la lista de las líneas existentes.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y sistema cierra la interfaz.	

5.1.2. Especificación textual del CU “Gestionar proyecto.”

Caso de Uso	Gestionar proyecto
Actores	Jefe de turno
Propósito	Permitir al usuario insertar, modificar o eliminar un proyecto.
Resumen	El usuario inserta, elimina o modifica un proyecto.
CU asociados:	Gestionar línea, Gestionar componente.
Precondiciones	El actor debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.
Poscondiciones	El sistema contiene proyectos insertados o modificados.

Referencias	RF2.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el menú Inicio/Qualitás , la opción a Gestionar proyecto .	2. El sistema muestra la interfaz Gestionar proyecto.
3. El actor selecciona el botón Adicionar . - Si el actor selecciona la opción Modificar (Ver Sección I). - Si el actor selecciona la opción Eliminar (Ver Sección II).	4. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo proyecto.
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	6. El sistema valida los datos que se introducen. 7. El sistema muestra un mensaje indicando que el proyecto ha sido guardado correctamente. 8. El sistema actualiza el listado de proyectos.
Cursos Alternos	
Línea 5: El actor presiona el botón Aplicar , el sistema actualiza el listado de proyectos y mantiene la interfaz abierta. (Ver Sección I).	
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el proyecto que desea modificar y selecciona el botón Modificar .	2. El sistema muestra una interfaz con los datos que posee el proyecto.
3. El actor modifica los datos del proyecto y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que los datos del proyecto fueron actualizados correctamente. 5. El sistema actualiza la lista de proyectos.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (II) Eliminar	

Acción del Actor	Respuesta del Sistema
1. El actor selecciona el proyecto que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar del mensaje de confirmación.	4. El sistema muestra un mensaje indicando que el proyecto fue eliminado de manera correcta. 5. El sistema actualiza la lista de proyecto.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

5.1.3. Especificación textual del CU “Gestionar componente.”

Caso de Uso	Gestionar componentes
Actores	Jefe de turno
Propósito	Permitir al usuario insertar, modificar o eliminar componentes.
Resumen	El usuario inserta, elimina o modifica componentes.
CU asociados	Gestionar línea, Gestionar proyecto.
Precondiciones	El actor se debe haber autenticado y debe tener los permisos necesarios para realizar la funcionalidad, debe haber líneas y proyectos insertados en el sistema.
Poscondiciones	El sistema contiene componentes insertados o modificados.
Referencias	RF3.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el menú Inicio/Qualitás , la opción a Gestionar componentes .	2. El sistema muestra la interfaz Gestionar componentes.

<p>3. El actor selecciona el botón Adicionar.</p> <p>-Si el actor selecciona la opción Modificar (Ver Sección I).</p> <p>-Si el actor selecciona la opción Eliminar (Ver Sección II).</p>	<p>4. El sistema muestra una interfaz con los campos necesarios para adicionar nuevos componentes.</p>
<p>5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar.</p>	<p>6. El sistema valida los datos que se introducen.</p> <p>7. El sistema muestra un mensaje indicando que el componente ha sido guardado correctamente.</p> <p>8. El sistema actualiza el listado de componentes.</p>
Cursos Alternos	
<p>Línea 5: El actor presiona el botón Aplicar, el sistema actualiza el listado de componentes y mantiene la interfaz abierta (Ver Sección I).</p>	
<p>Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.</p>	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
<p>1. El actor da clic sobre el componentes que desea modificar y selecciona el botón Modificar.</p>	<p>2. El sistema muestra una interfaz con los datos que posee el componente.</p>
<p>3. El actor modifica los datos del componentes y presiona el botón Guardar.</p>	<p>4. El sistema muestra un mensaje donde indica que los datos del componente fueron actualizados correctamente.</p> <p>5. El sistema actualiza la lista de componentes.</p>
Cursos Alternos	
<p>Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.</p>	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
<p>1. El actor selecciona el componentes que desea eliminar y presiona el botón Eliminar.</p>	<p>2. El sistema muestra un mensaje de confirmación.</p>
<p>3. El actor presiona el botón Aceptar del mensaje de confirmación.</p>	<p>4. El sistema muestra un mensaje indicando que el componente fue eliminado de manera correcta.</p> <p>5. El sistema actualiza la lista de componentes.</p>

Cursos Alternos
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.

5.1.4. Especificación textual del CU “Gestionar tipo de prueba.”

Caso de Uso	Gestionar tipo de prueba	
Actores	Jefe de turno	
Propósito	Permitir al usuario insertar, modificar y eliminar tipos de pruebas.	
Resumen	El actor inserta el tipo de pruebas posteriormente puede hacer modificaciones a los datos, o eliminar algún tipo de prueba en caso de ser necesario.	
CU asociados	Gestionar elemento.	
Precondiciones	El actor se debe haber autenticado y debe tener los permisos necesarios para realizar la operación.	
Poscondiciones	El sistema contiene tipo de pruebas insertados, modificados.	
Referencias	RF4.	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el menú Inicio/Qualitäts , la opción a Gestionar tipo de prueba.	2. El sistema muestra la interfaz Gestionar tipo de prueba.	
3. El actor selecciona el botón Adicionar. -Si el actor selecciona la opción Modificar (Ver Sección I) -Si el actor selecciona la opción Eliminar (Ver Sección II)	4. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo tipo de prueba.	
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar.	6. El sistema valida los datos que se introducen. 7. El sistema muestra un mensaje indicando que el tipo de prueba ha sido guardado correctamente. 8. El sistema actualiza el listado de tipo de	

pruebas.	
Cursos Alternos	
Línea 5: El actor presiona el botón Aplicar y el sistema actualiza el listado de tipo de pruebas y mantiene la interfaz abierta (Ver Sección I).	
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el tipo de prueba que desea modificar.	2. El sistema muestra una interfaz con los datos que posee el tipo de prueba.
3. El actor modifica el tipo de prueba y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que el tipo de prueba fue actualizado correctamente.
	5. El sistema actualiza la lista de tipo de pruebas.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el tipo de prueba que desea eliminar y da clic en el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
	3. El actor presiona el botón Aceptar del mensaje.
	5. El sistema actualiza la lista de tipo de pruebas.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

5.1.5. Especificación textual del CU “Gestionar elemento.”

Caso de Uso	Gestionar elemento
--------------------	--------------------

Actores	Jefe de turno	
Propósito	Permitir al usuario insertar, modificar y eliminar elementos.	
Resumen	El actor inserta los elementos posteriormente puede hacer modificaciones a los datos, o eliminar algún elemento en caso de ser necesario.	
CU asociados	Gestionar componente, Gestionar proyecto.	
Precondiciones	El actor se debe haber autenticado y debe tener los permisos necesarios para realizar la operación.	
Poscondiciones	El sistema contiene elementos insertados, modificados.	
Referencias	RF5.	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el menú Inicio/Qualitäts , la opción a Gestionar elemento .	2. El sistema muestra la interfaz Gestionar elemento.	
3. El actor selecciona el botón Adicionar. -Si el actor selecciona la opción Modificar (Ver Sección I) -Si el actor selecciona la opción Eliminar (Ver Sección II)	4. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo elemento.	
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar	6. El sistema valida los datos que se introducen.	
	7. El sistema muestra un mensaje indicando que el elemento ha sido guardado correctamente.	
	8. El sistema actualiza el listado de elementos.	
Cursos Alternos		

Línea 5: El actor presiona el botón Aplicar y el sistema actualiza el listado de elementos y mantiene la interfaz abierta (Ver Sección I).	
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el elemento que desea modificar.	2. El sistema muestra una interfaz con los datos que posee el elemento.
3. El actor modifica el elemento y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que el elemento fue actualizado correctamente.
	5. El sistema actualiza la lista de elementos.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el elemento que desea eliminar y da clic en el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar del mensaje.	4. El sistema muestra un mensaje indicando que el elemento fue eliminado de manera correcta.
	5. El sistema actualiza la lista de elementos.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

5.1.6. Especificación textual del CU “Gestionar probador.”

Caso de Uso	Gestionar probador
Actores	Usuario común
Propósito	Permitir al usuario insertar, modificar o eliminar un probador.

Resumen	El usuario inserta, elimina o modifica un probador.	
CU asociados		
Precondiciones	El actor debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.	
Poscondiciones	El sistema contiene probadores insertados o modificados.	
Referencias	RF6.	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona el menú Inicio/Qualitäts , la opción a Gestionar probador .	2. El sistema muestra la interfaz Gestionar probador.	
3. El actor selecciona el botón Adicionar . -Si el actor selecciona la opción Modificar (Ver Sección I) -Si el actor selecciona la opción Eliminar (Ver Sección II)	4. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo probador.	
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	6. El sistema valida los datos que se introducen. 7. El sistema muestra un mensaje indicando que la operación se realizó de manera correcta. 8. El sistema actualiza el listado de probadores.	
Cursos Alternos		
Línea 5: El actor presiona el botón Aplicar y el sistema actualiza el listado de probadores y mantiene la interfaz abierta (Ver Sección I).		
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz.		
Sección (I) Modificar		
Acción del Actor	Respuesta del Sistema	

1. El actor da clic sobre el probador que desea modificar y selecciona el botón Modificar .	2. El sistema muestra una interfaz con los datos que posee el probador.
3. El actor modifica los datos del probador y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que los datos del probador fueron actualizados correctamente. 5. El sistema actualiza la lista de probadores.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	
Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el probador que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar del mensaje de confirmación.	4. El sistema muestra un mensaje indicando que el probador fue eliminado de manera correcta. 5. El sistema actualiza la lista de probadores.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

5.1.7. Especificación textual del CU “Gestionar trabajo.”

Caso de Uso	Gestionar trabajo
Actores	Trabajador
Propósito	Permitir al usuario insertar, modificar o eliminar trabajo.
Resumen	El usuario inserta, elimina o modifica un probador.
CU asociados	Gestionar probadores, Gestionar elemento.
Precondiciones	El actor debe haberse autenticado y debe tener los permisos necesarios para realizar la funcionalidad.
Poscondiciones	El sistema contiene trabajo insertados, modificados o repartidos.
Referencias	RF8.
Descripción	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el menú Inicio/Qualitás , la opción a Gestionar probador .	2. El sistema muestra la interfaz Gestionar probador.
3. El actor selecciona el botón Adicionar . -Si el actor selecciona la opción Modificar (Ver Sección I). -Si el actor selecciona la opción Eliminar (Ver Sección II).	4. El sistema muestra una interfaz con los campos necesarios para adicionar un nuevo trabajo.
5. El actor llena los campos con los datos correspondientes y presiona el botón Aceptar .	6. El sistema valida los datos que se introducen.
	7. El sistema muestra un mensaje indicando que el trabajo se ha guardado correctamente.
	8. El sistema actualiza el listado de probadores.
Cursos Alternos	
Línea 5: El actor presiona el botón Aplicar , el sistema actualiza el listado de trabajos y mantiene la interfaz abierta (Ver Sección I).	
Línea 5: El actor presiona el botón Cancelar y el sistema cierra la interfaz	
Sección (I) Modificar	
Acción del Actor	Respuesta del Sistema
1. El actor da clic sobre el trabajo que desea modificar y selecciona el botón Modificar .	2. El sistema muestra una interfaz con los datos que posee el trabajo.
3. El actor modifica los datos del trabajo y presiona el botón Guardar .	4. El sistema muestra un mensaje donde indica que los datos del trabajo fueron actualizados correctamente.
	5. El sistema actualiza la lista de trabajos.
Cursos Alternos	
Línea 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

Sección (II) Eliminar	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona el trabajo que desea eliminar y presiona el botón Eliminar .	2. El sistema muestra un mensaje de confirmación.
3. El actor presiona el botón Aceptar del mensaje de confirmación.	4. El sistema muestra un mensaje indicando que el trabajo fue eliminado de manera correcta.
	5. El sistema actualiza la lista de trabajos.
Cursos Alternos	
Línea: 3: El actor presiona el botón Cancelar y el sistema cierra la interfaz.	

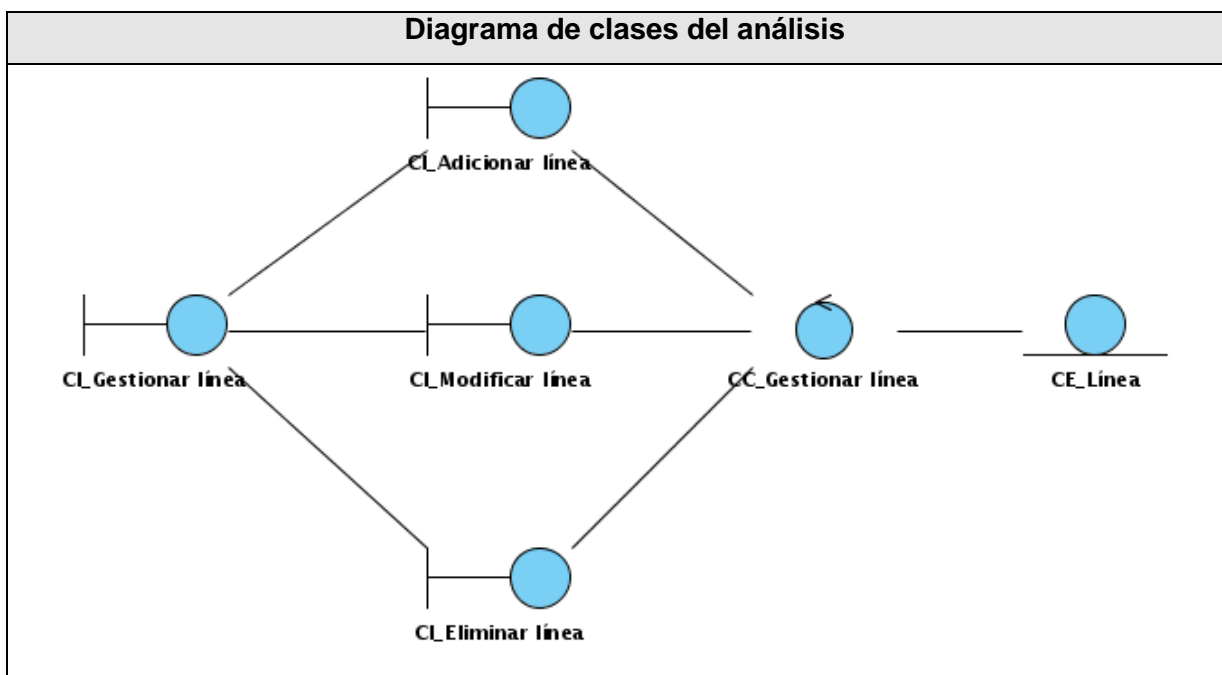
5.1.8. Especificación textual del CU “Generar parte.”

Caso de Uso	Generar partes
Actores	Solicitador de reporte
Propósito	Permite al usuario visualizar los partes diarios y crear reportes.
Resumen	El usuario selecciona la opción y obtiene los partes diarios.
CU asociados	Gestionar No conformidades
Precondiciones	El actor se debe haber autenticado y debe tener los permisos necesarios para realizar la operación.
Poscondiciones	El sistema permite la visualizar partes y crear reportes.
Referencias	RF9.
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona en el menú Inicio/Qualitás la opción Generar parte .	2. El sistema muestra una interfaz con los campos del parte.
3. El actor selecciona el botón Generar parte . -Si el actor selecciona la opción Reporte (Ver Sección I).	4. El sistema muestra los partes existentes.
5. El actor cierra la interfaz.	
Cursos Alternos	
Línea 2: El sistema no tiene ningún parte y el actor cierra la interfaz.	

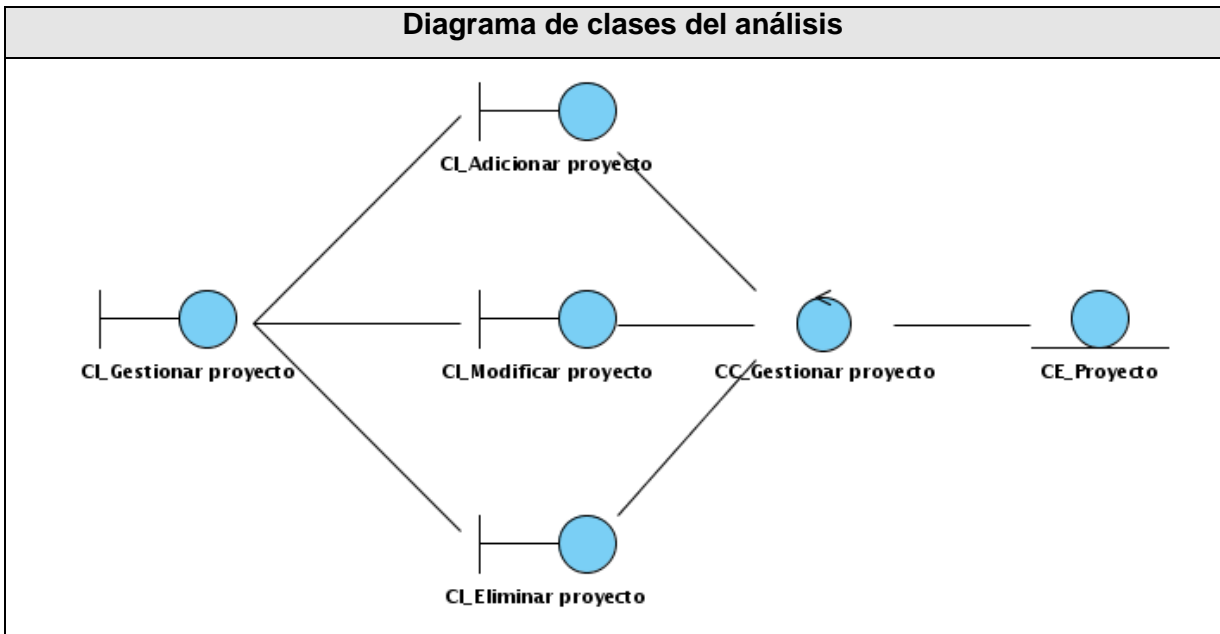
Sección (I) Reporte	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Reporte .	2. El sistema abre la aplicación PATDSI.
Cursos Alternos	
Línea 2: El actor cierra la interfaz.	

5.2. Anexos del Capítulo 3 “Diagrama de clases del análisis”

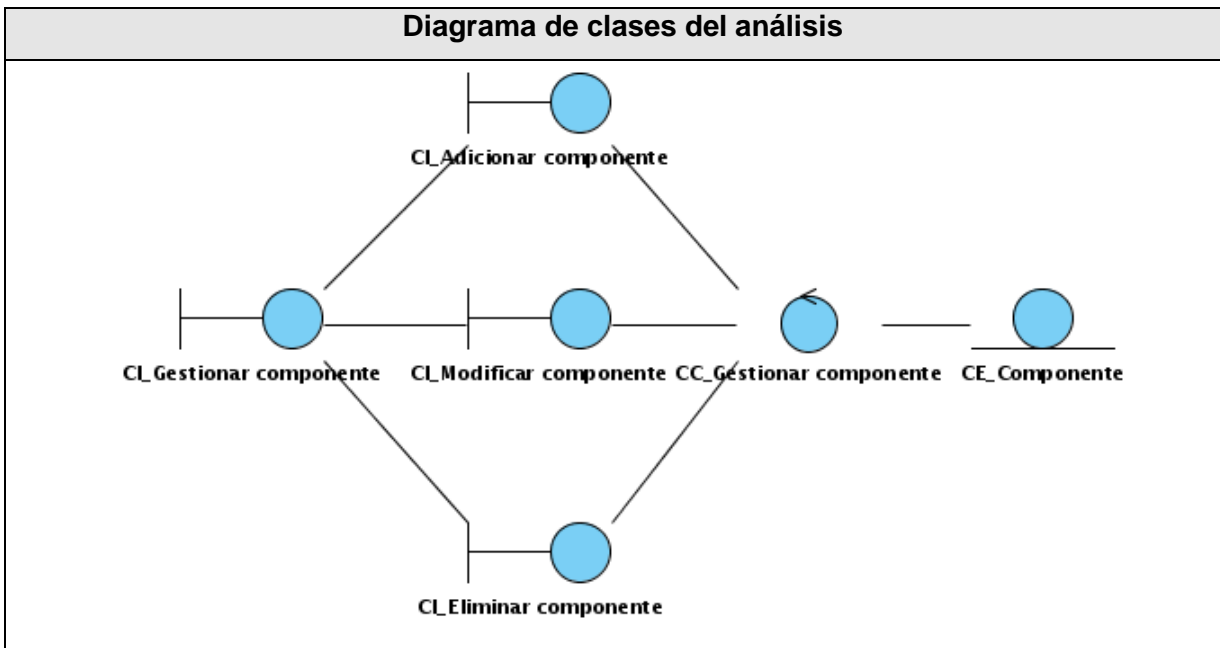
5.2.1. Diagrama de clases del análisis “Gestionar línea.”



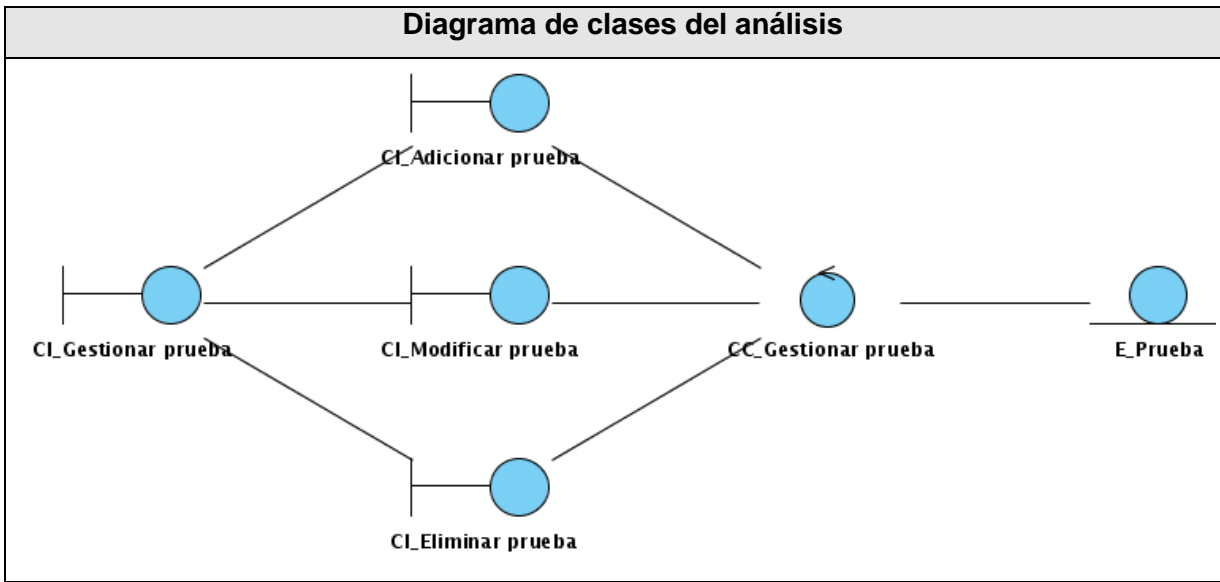
5.2.2. Diagrama de clases del análisis “Gestionar proyecto.”



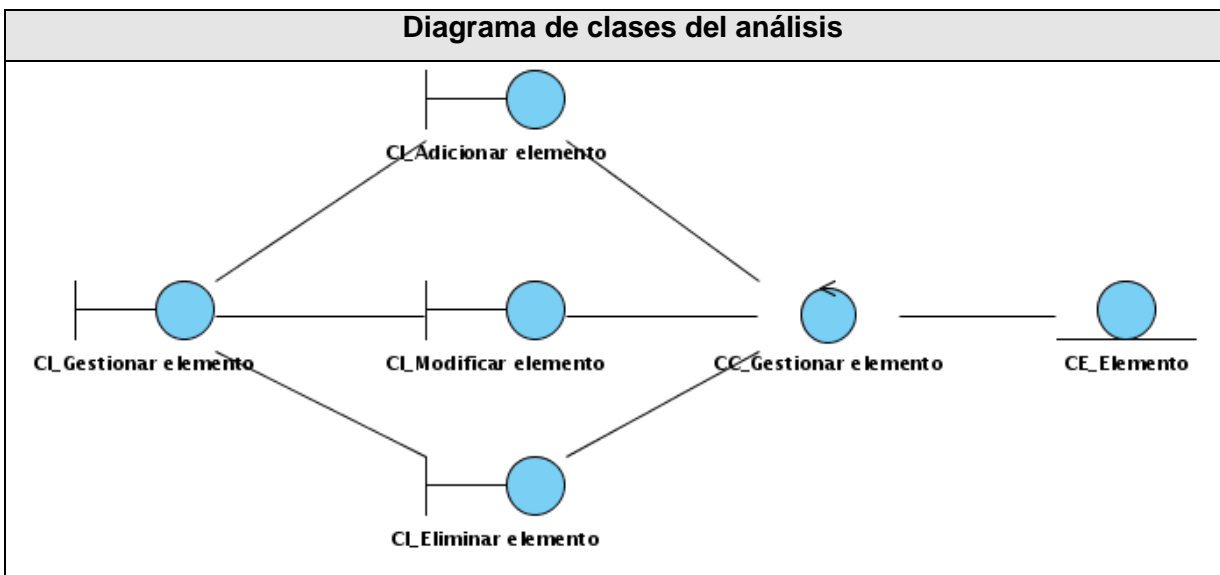
5.2.3. Diagrama de clases del análisis “Gestionar componente.”



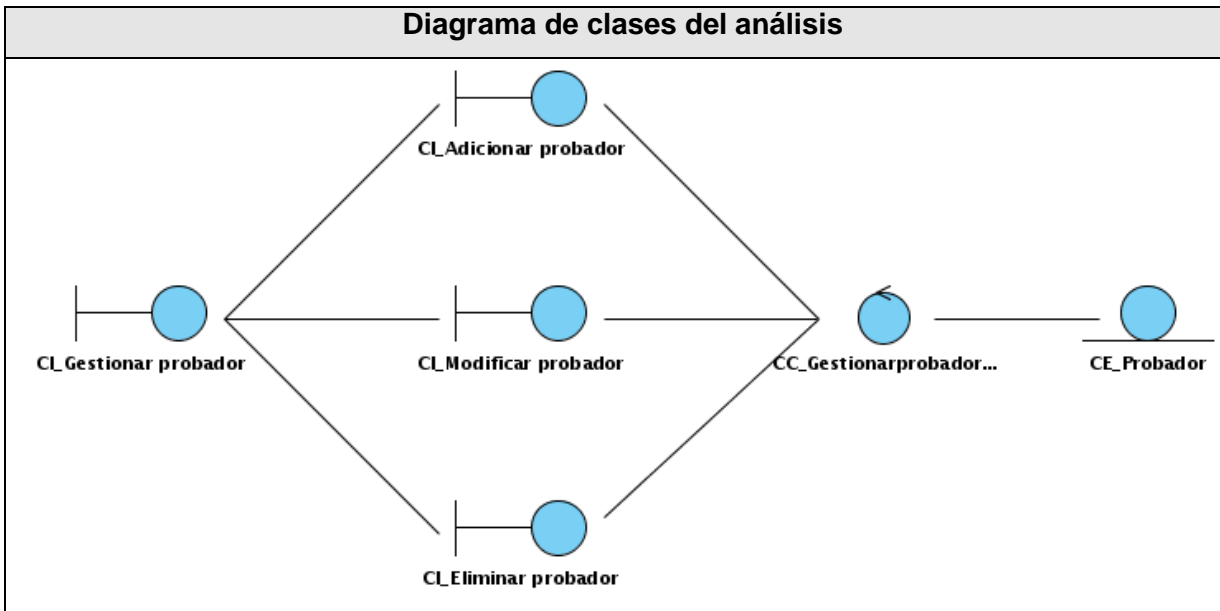
5.2.4. Diagrama de clases del análisis “Gestionar tipo de prueba.”



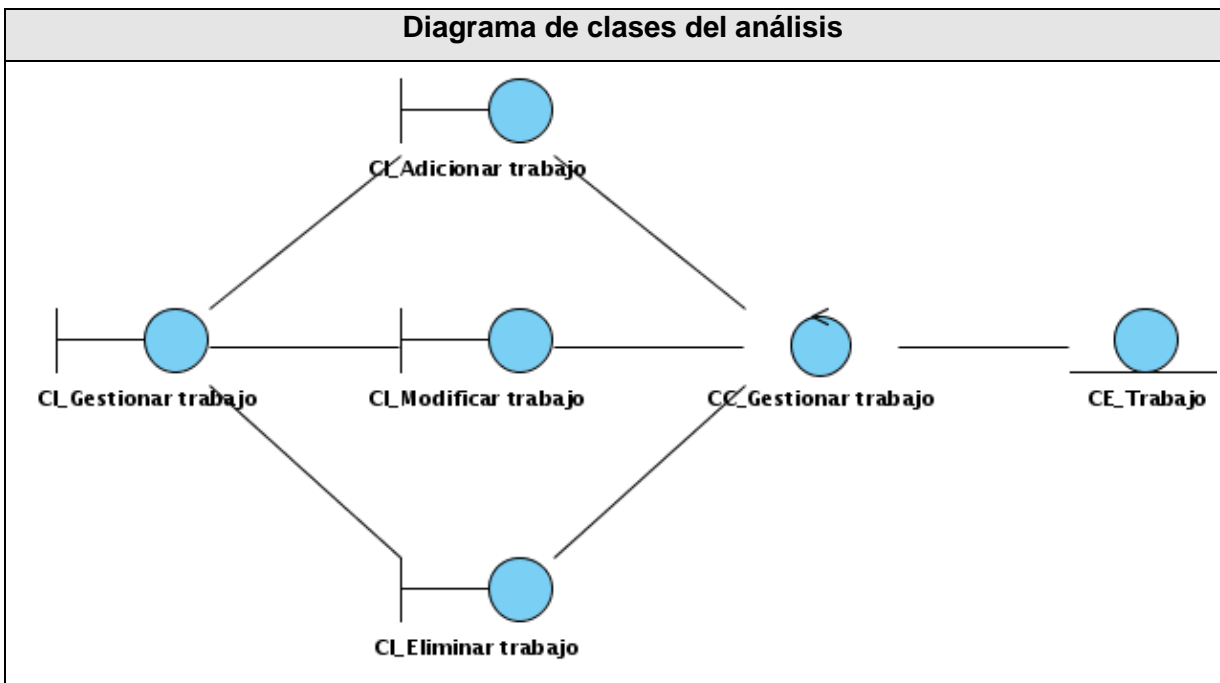
5.2.5. Diagrama de clases del análisis “Gestionar elemento.”



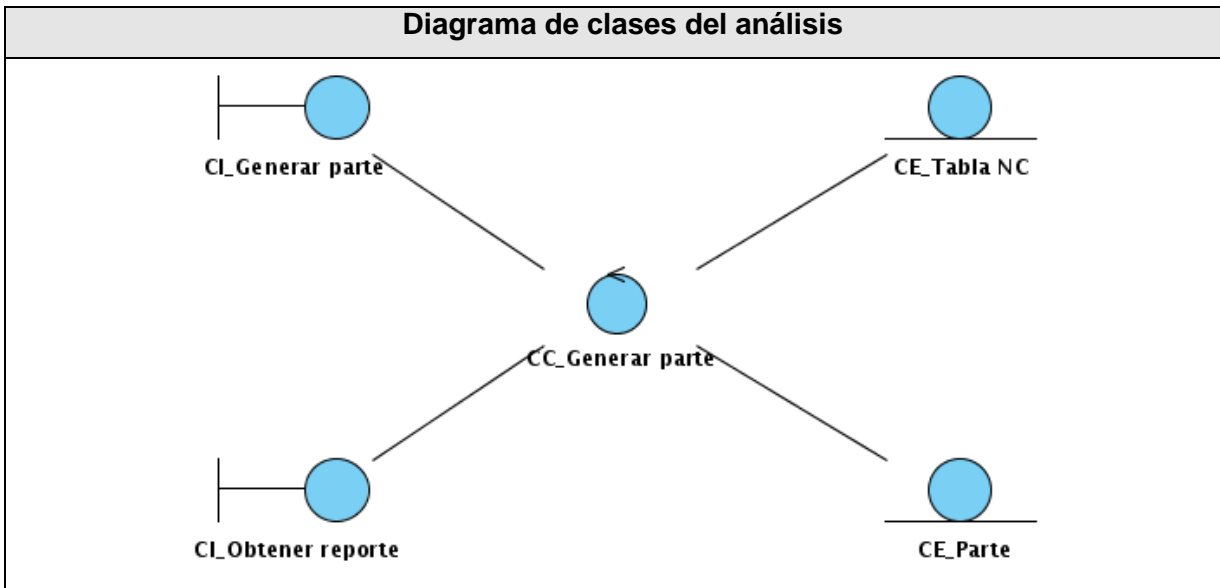
5.2.6. Diagrama de clases del análisis “Gestionar probador.”



5.2.7. Diagrama de clases del análisis “Gestionar trabajo.”

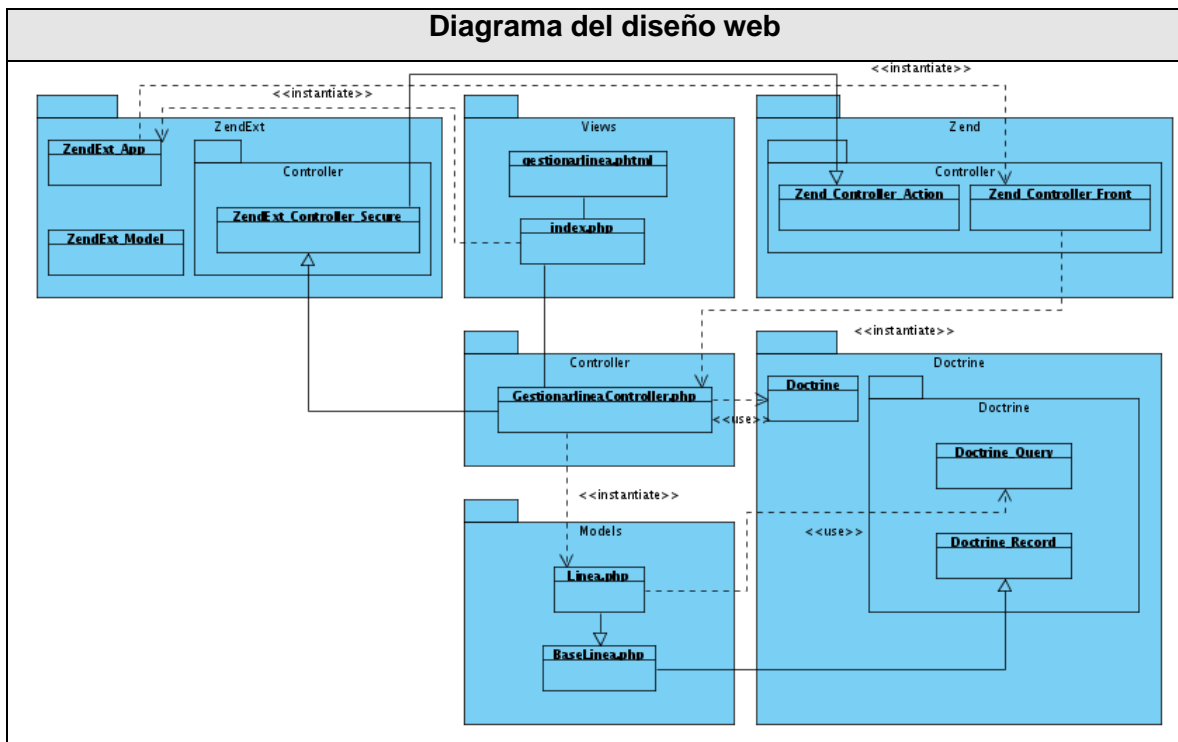


5.2.8. Diagrama de clases del análisis “Generar partes.”

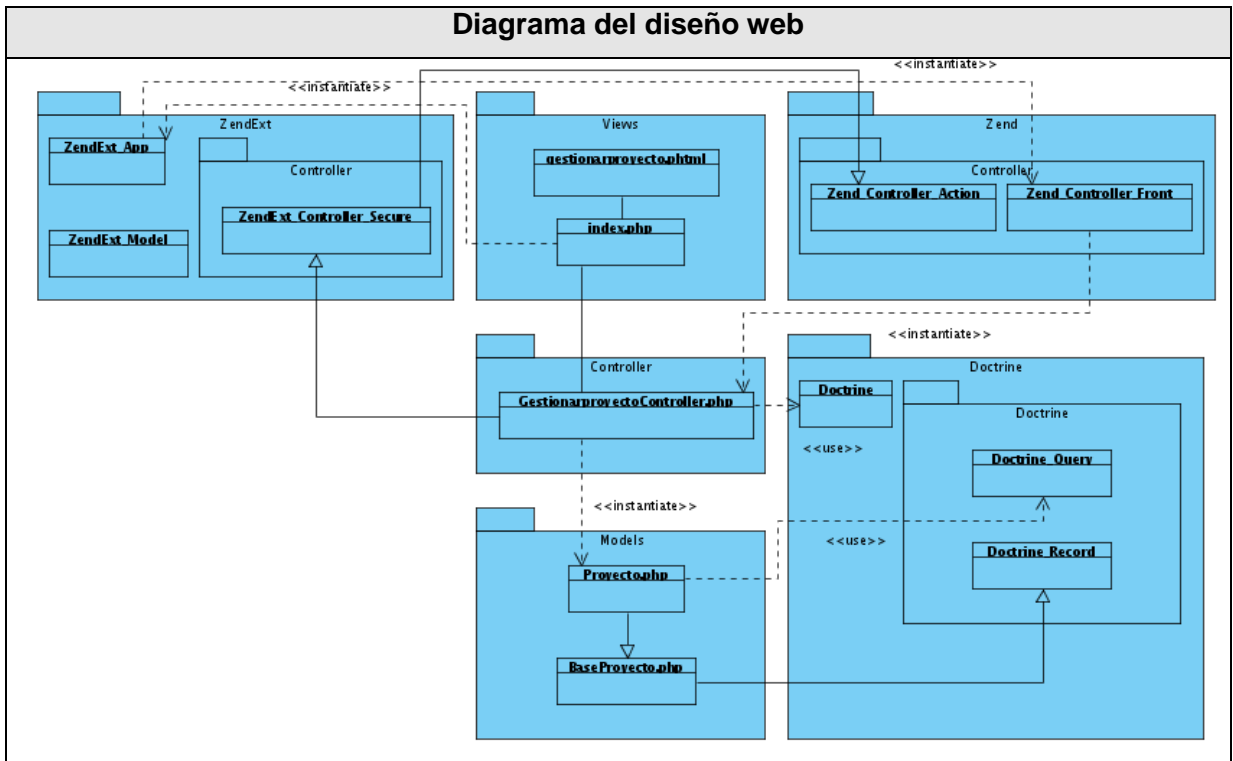


5.3. Anexos del Capítulo 3 “Diagrama de clases del diseño web”

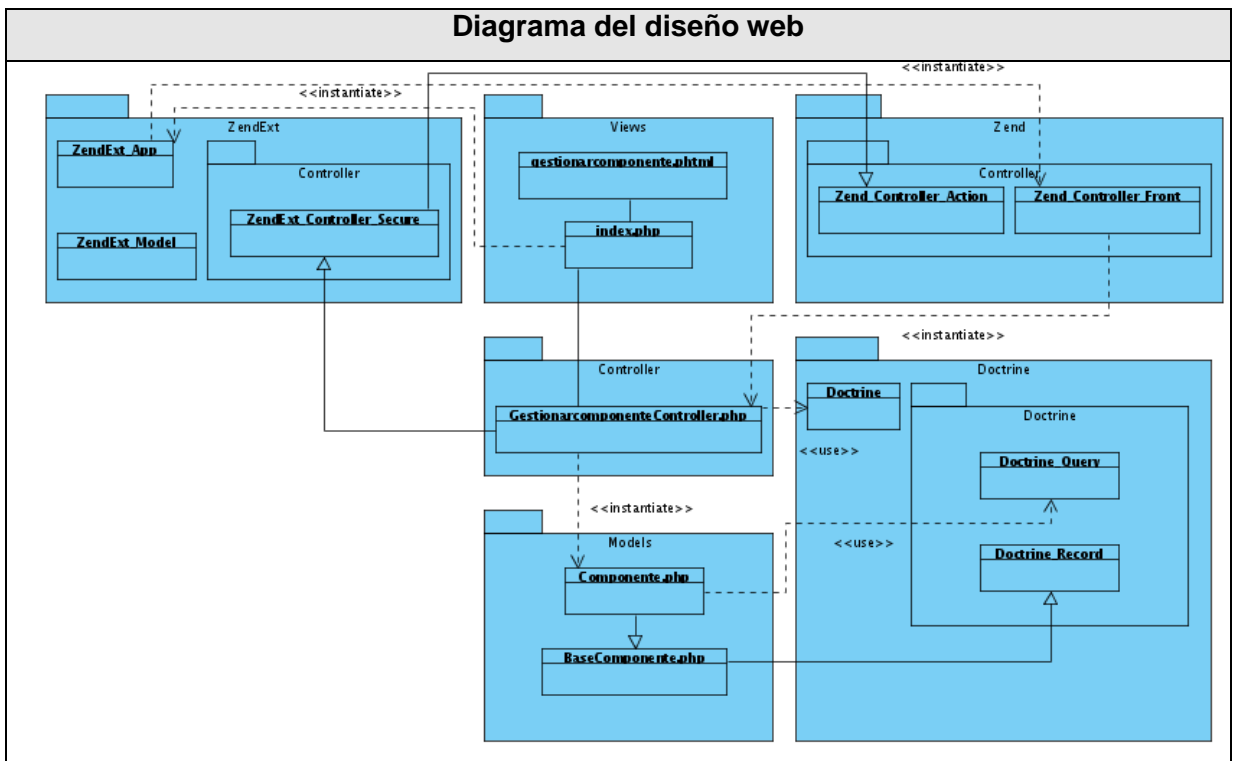
5.3.1. Diagrama de clases del diseño web “Gestionar línea.”



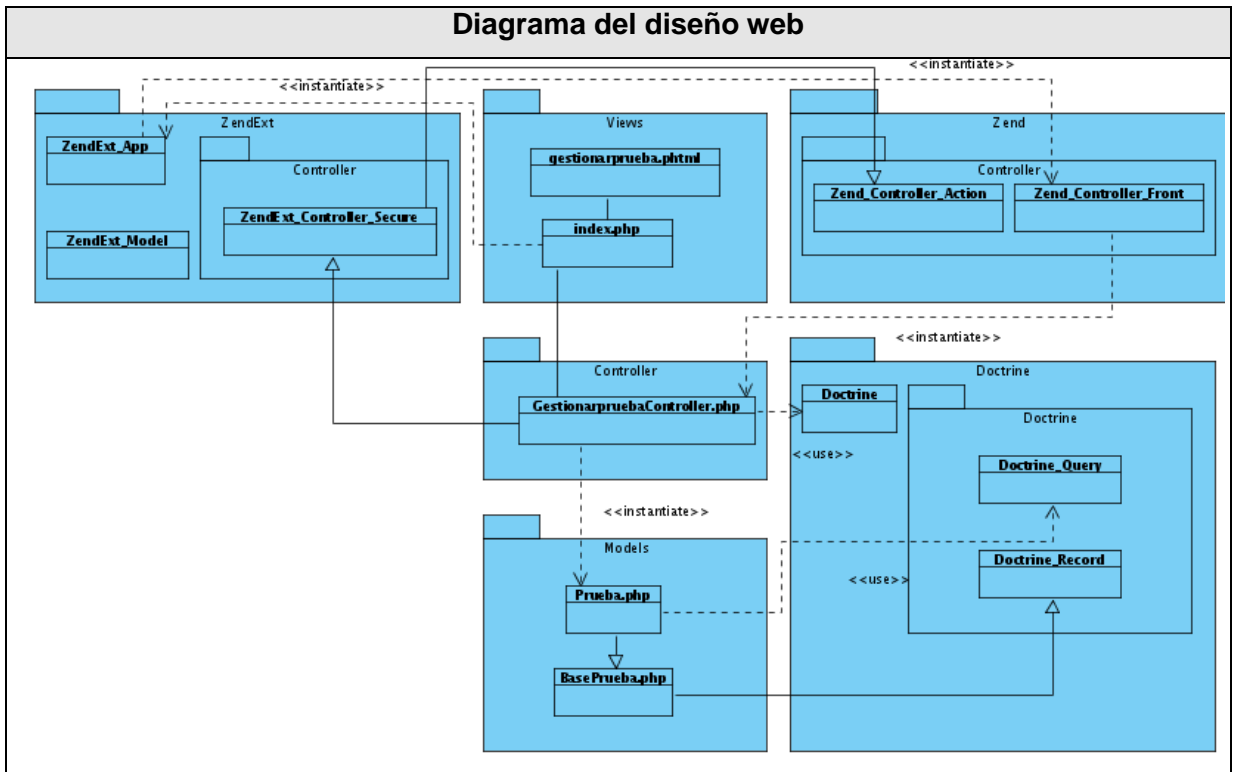
5.3.2. Diagrama de clases del diseño web “Gestionar proyecto.”



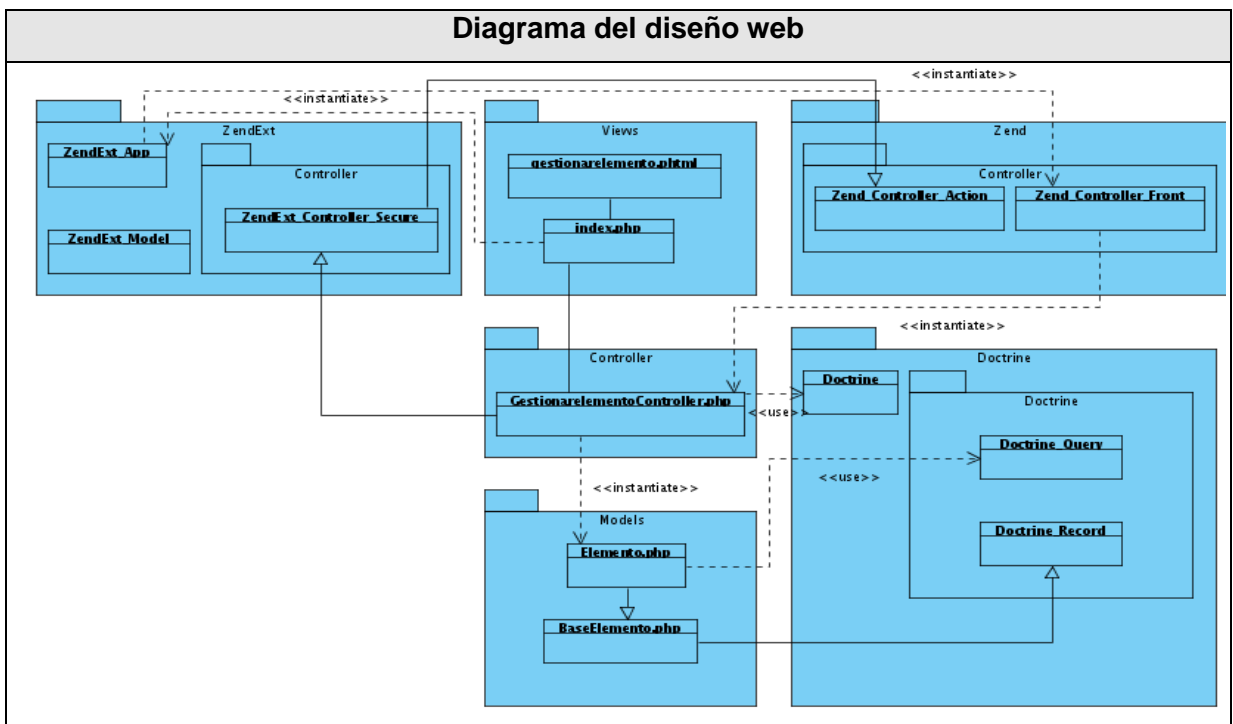
5.3.3. Diagrama de clases del diseño web “Gestionar componente.”



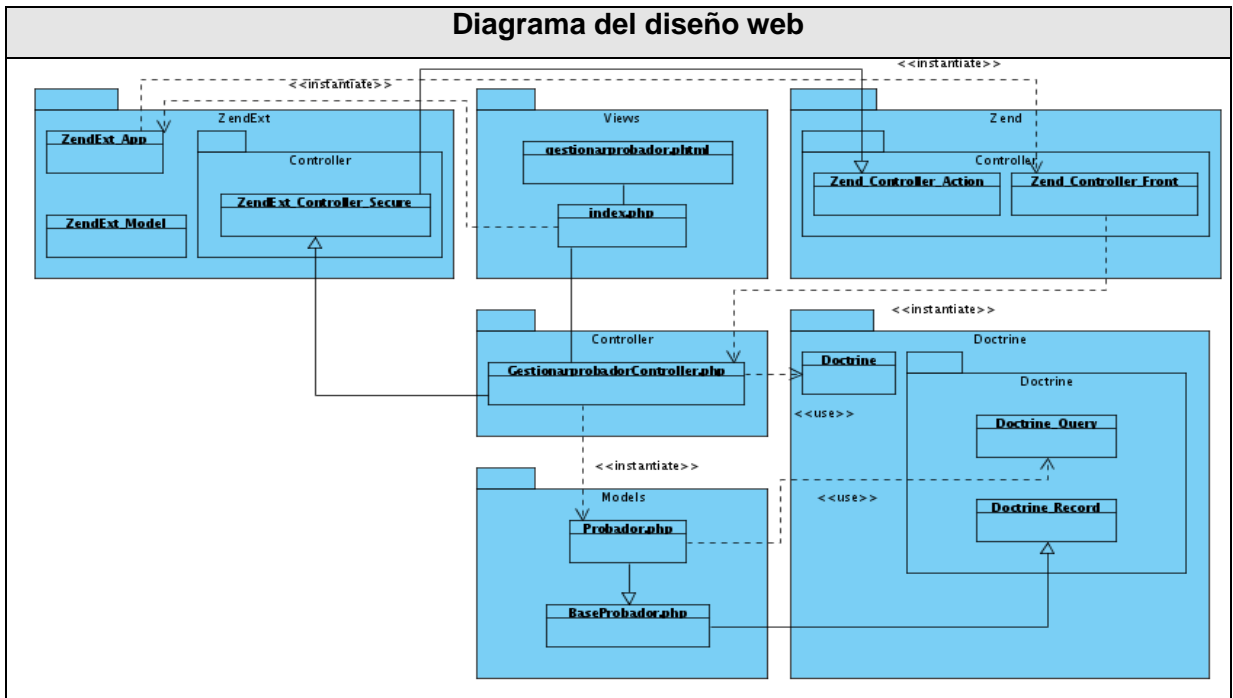
5.3.4. Diagrama de clases del diseño web “Gestionar tipo de prueba.”



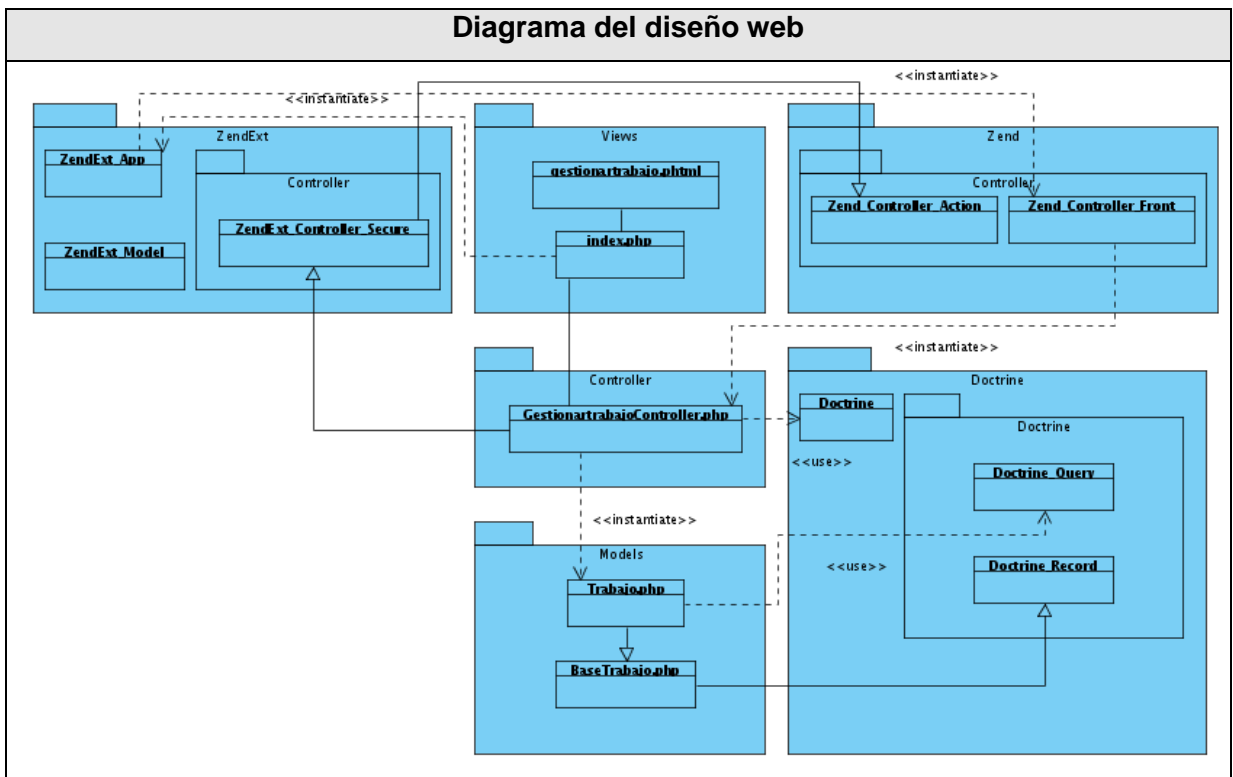
5.3.5. Diagrama de clases del diseño web “Gestionar elemento.”



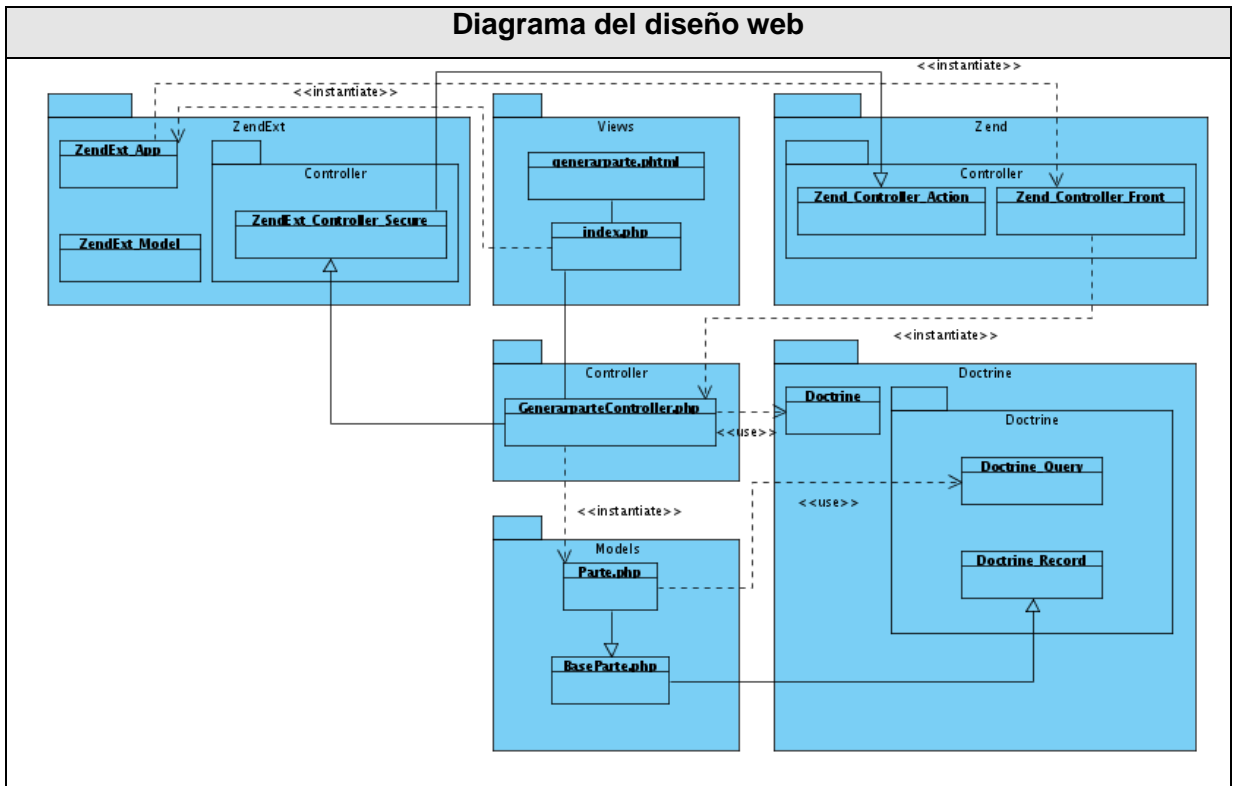
5.3.6. Diagrama de clases del diseño web “Gestionar probador.”



5.3.7. Diagrama de clases del diseño web “Gestionar trabajo.”

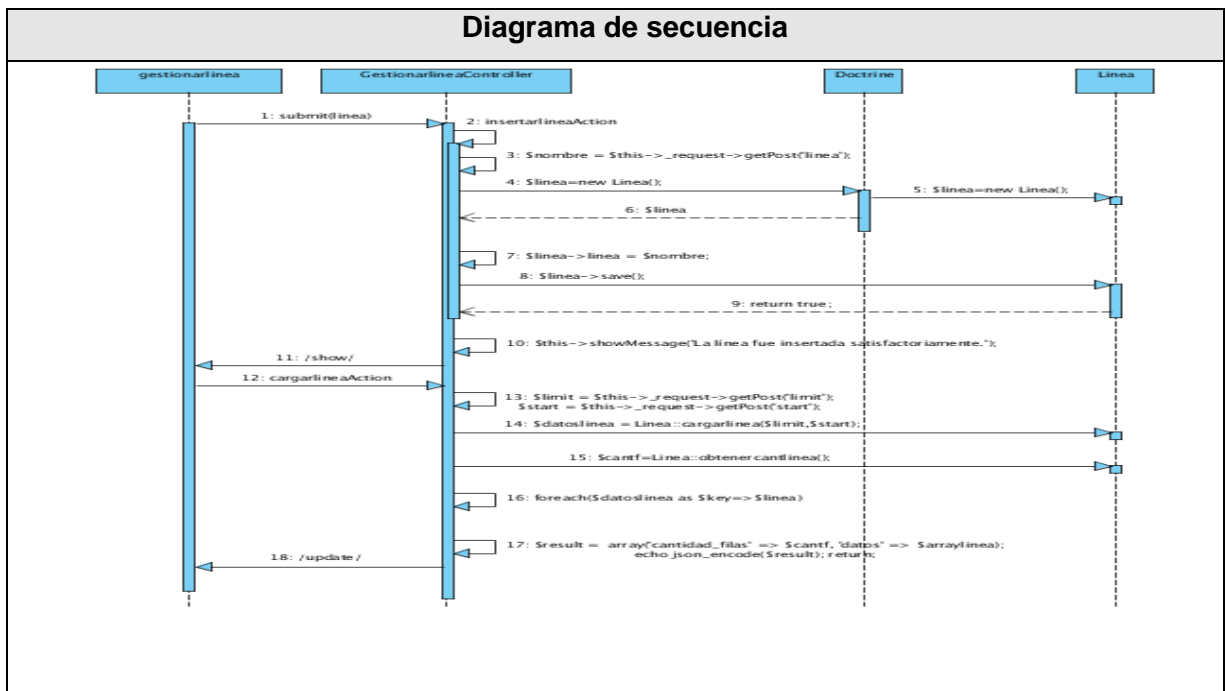


5.3.8. Diagrama de clases del diseño web “Generar parte.”

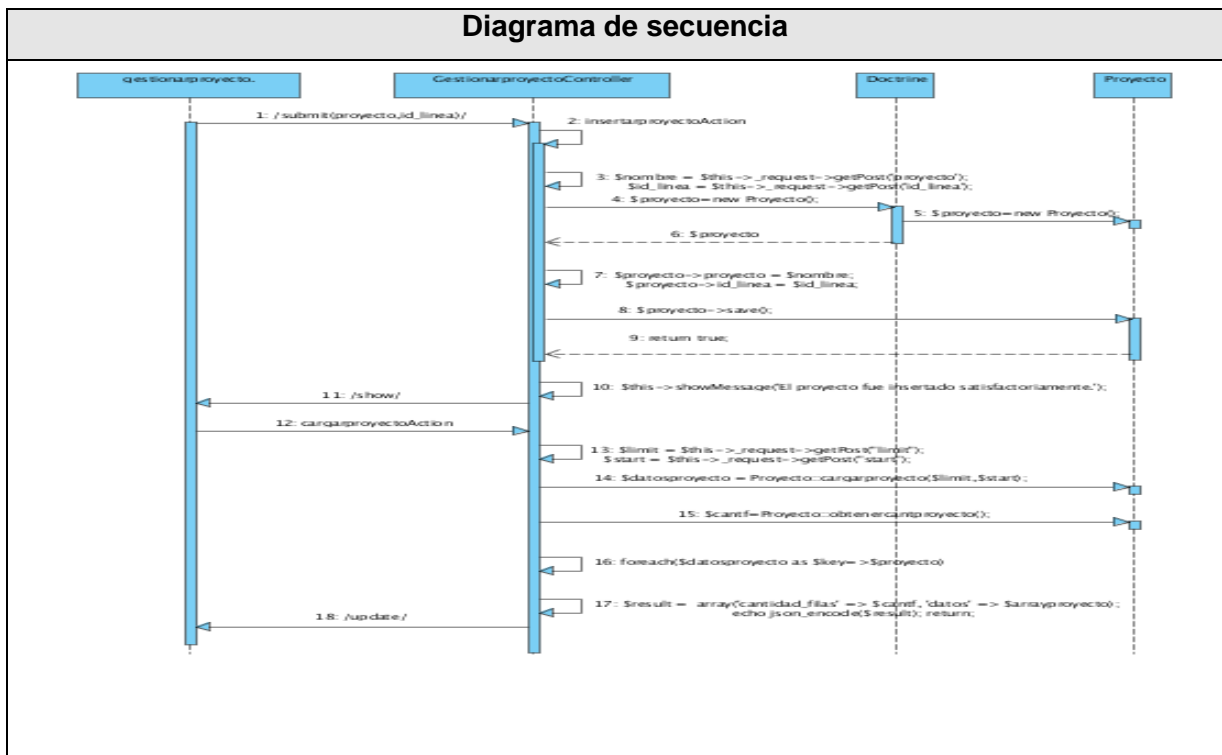


5.4. Anexos del Capítulo 3 “Diagrama de secuencia.”

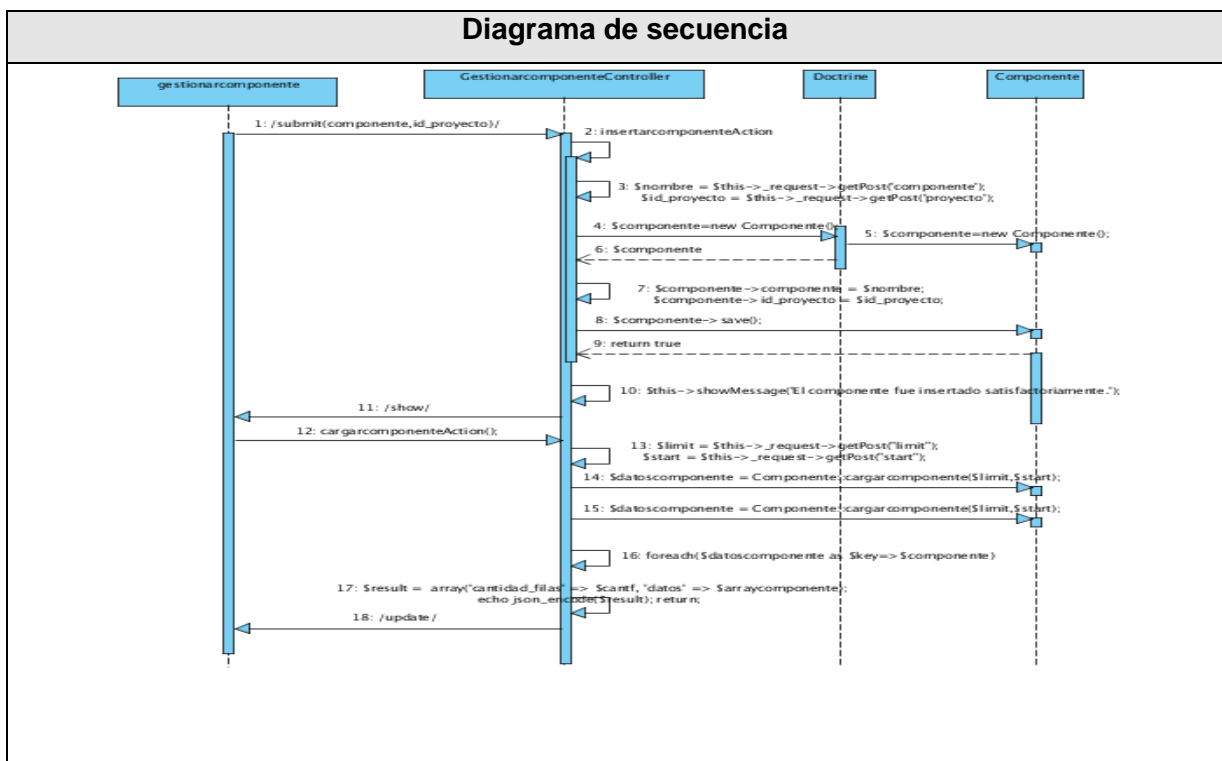
5.4.1. Diagrama de secuencia “Adicionar línea.”



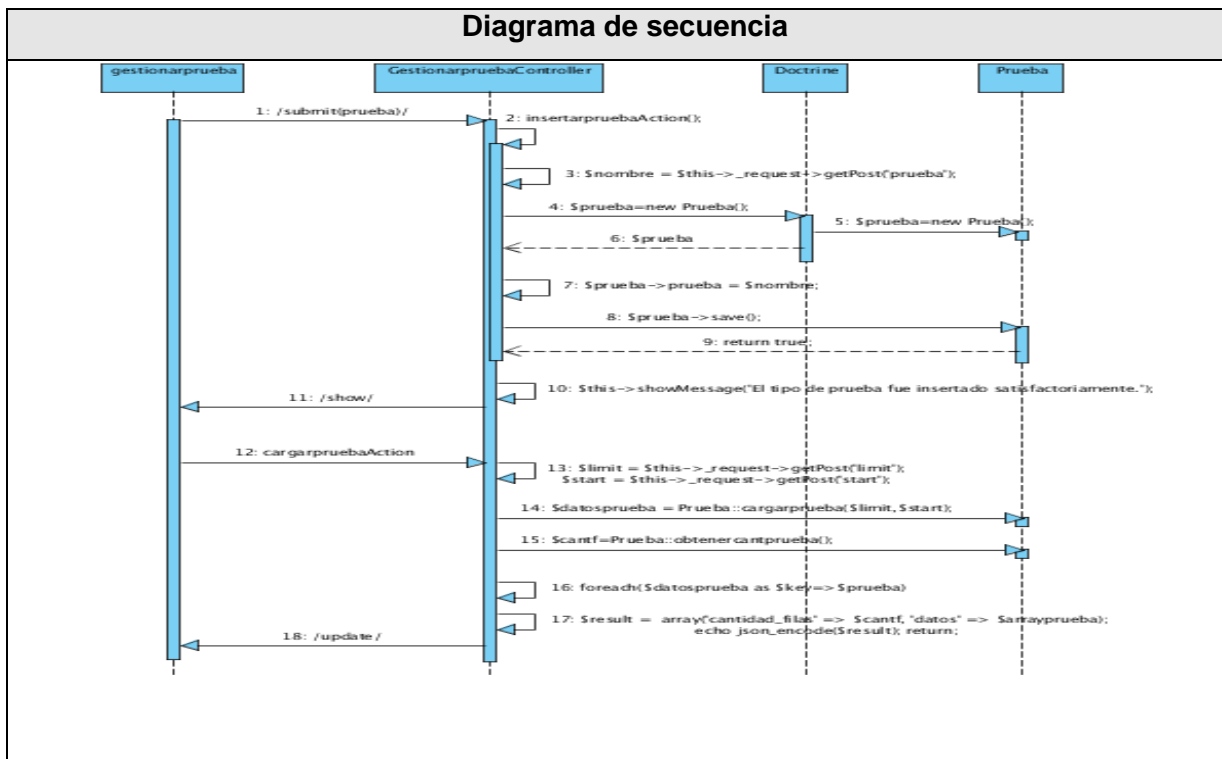
5.4.2. Diagrama de secuencia “Adicionar proyecto.”



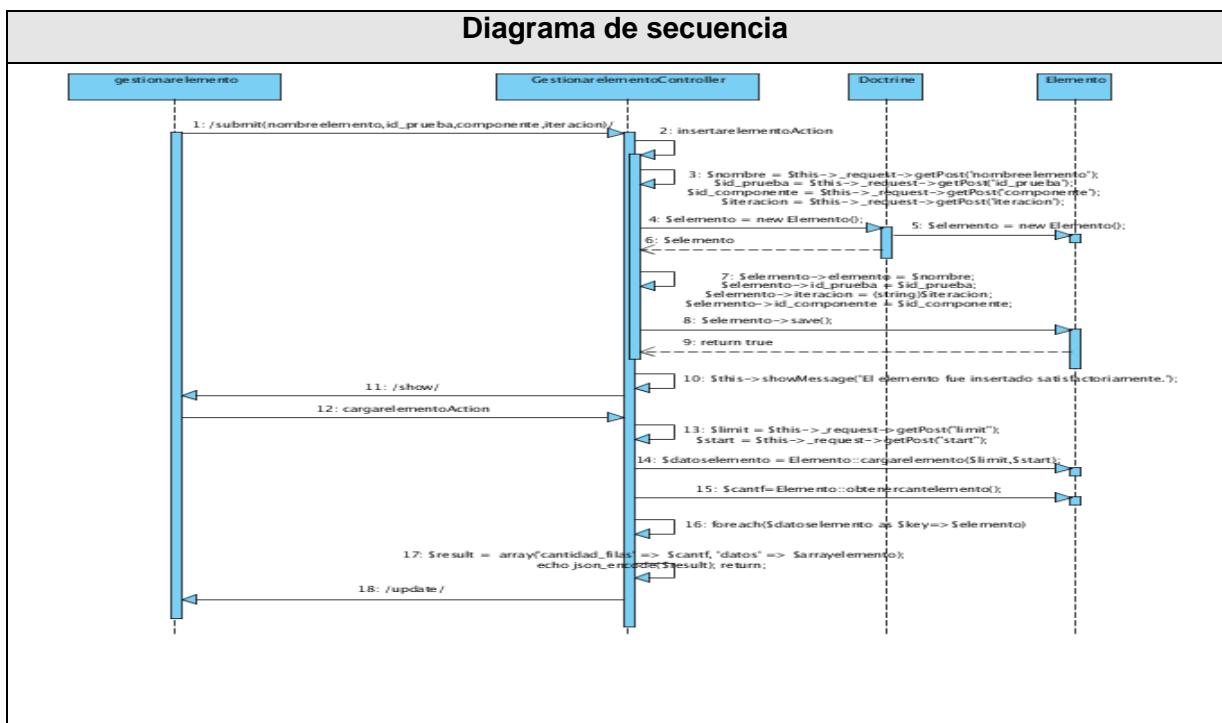
5.4.3. Diagrama de secuencia “Adicionar componente.”



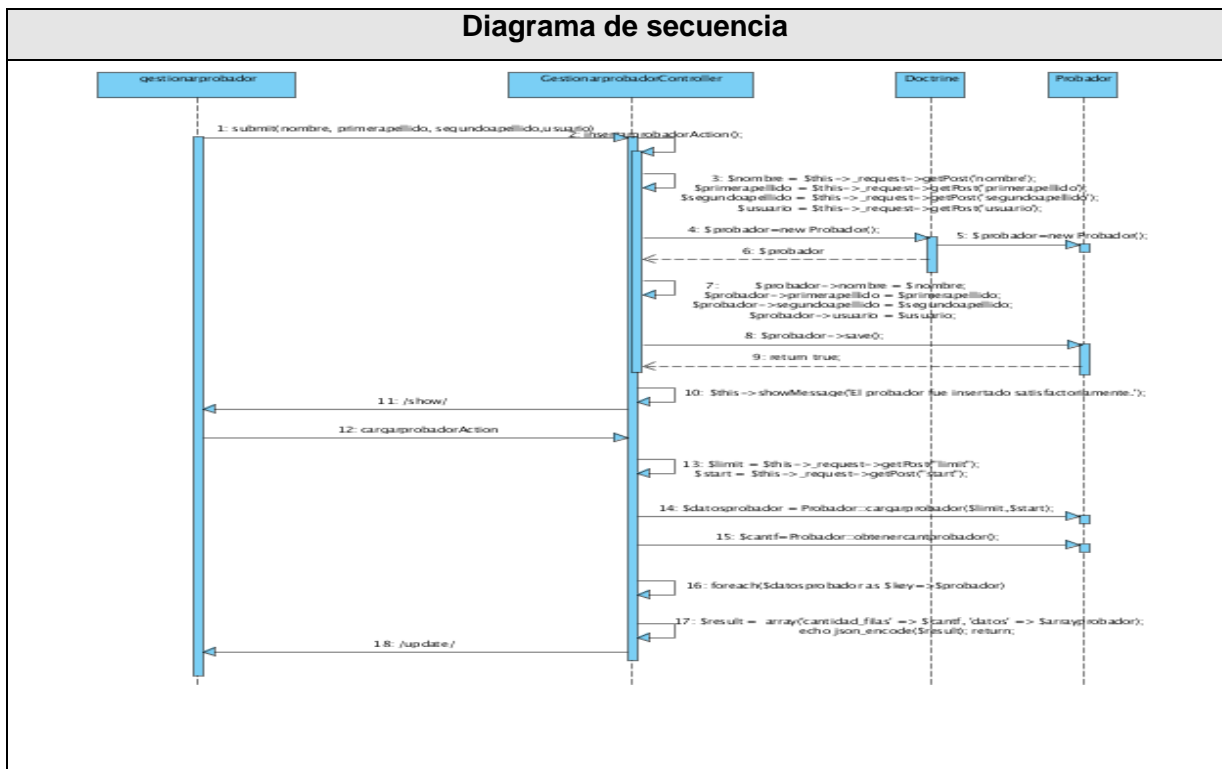
5.4.4. Diagrama de secuencia “Adicionar tipo de prueba.”



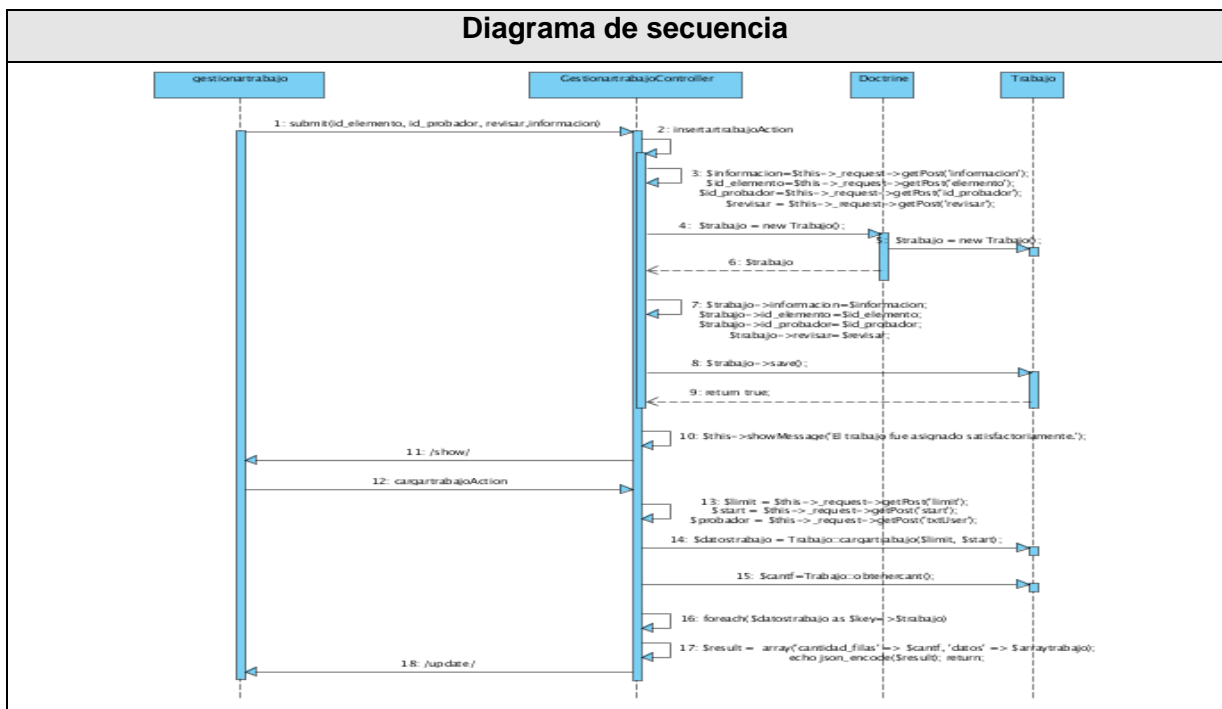
5.4.5. Diagrama de secuencia “Adicionar elemento.”



5.4.6. Diagrama de secuencia “Adicionar probador.”



5.4.7. Diagrama de secuencia “Adicionar trabajo.”



5.4.8. Diagrama de secuencia “Adicionar parte.”

GLOSARIO

ASP: *Active Server Pages*. 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). Con ASP se pueden combinar páginas HTML, scripts y objetos COM.

CGI: *Common Gateway Interface*. Estándar para transferir datos entre el cliente y el programa.

CSS: *Cascading Style Sheets*. Tecnología que permite la creación páginas web.

DCP: Diseño de Caso de prueba.

DCU: Diseño de Caso de uso.

DR: Diseño de requerimientos.

GPL: Licencia Pública de GNU.

HTML: *Hypertext Markup Language*. Lenguaje de marcas usado para escribir documentos para servidores World Wide Web. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: *Hypertext Transfer Protocol*. Es un protocolo diseñado para transferir hipertextos, páginas Web o páginas HTML.

IDE: *Integrated Development Environment*. Ambiente de desarrollo integrado

IIS: *Internet Information Services*. Es un servidor Web para aplicaciones Web sobre versiones de Windows.

Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems.

MA: Medición y Análisis. Área de proceso del nivel 2 CMMI.

MSSQL: *Microsoft SQL Server*. Es un sistema de gestión de bases de datos relacionales.

MVC: *Modelo Vista Controlador*. Patrón de arquitectura de software.

MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

NC: No conformidades.

Oracle: Sistema de gestión de base de datos relacional desarrollado por Oracle Corporation.

ORM: Mapeador Objeto-Relacional

PERL: *Practical Extraction and Report Language*. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.

PHP: *Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts.

PostgreSQL: Es un SGBD Objeto-Relacionales (ORDBMS) libre.

RUP: *Rational Unified Process*. En español: Proceso Unificado de Desarrollo. Metodología para el desarrollo de Software.

SGBD: Sistemas de gestión de base de datos.

UCI: Universidad de las Ciencias Informáticas.

UCID: Unidad de Compatibilización Integración y Desarrollo.

UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.