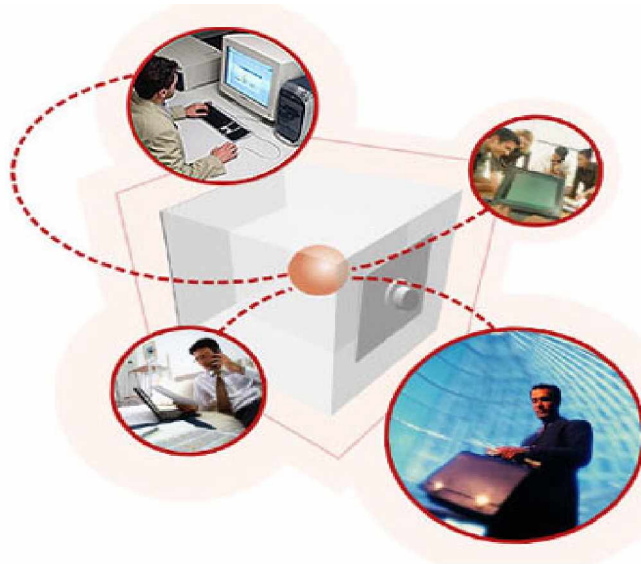


**Instituto Superior Minero Metalúrgico
“Dr. Antonio Núñez Jiménez”
Facultad Metalurgia Electromecánica.**



**Trabajo de Diploma para optar por el Título de Ingeniero en
Informática.**



Título: Sistema automatizado de control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel.

Autor: Yusimi Zaldivar Rizo

Tutor: Ing. Yodexy Mosqueda Naranjo

Consultante(s): María Caridad Sánchez Muguercia

Moa-Holguín

2010

Declaración de Autoría

Declaro que soy el único autor(a) del trabajo de diploma titulado **Sistema Automatizado de control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel (Empleni)** y que el mismo pertenece a la Empresa Empleadora del Níquel, para que hagan el uso que estime pertinente con este trabajo.

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

Firma del autor

Firma del tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título:

Autor:

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

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota 2-Desaprobado, 3-Aprobado, 4-Bien, 5-Excelente>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Nombre completo del primer tutor

<Grado científico, Categoría docente
y/o investigativa>

(Si procede)

Nombre completo del segundo tutor

<Grado científico, Categoría docente
y/o investigativa>

Fecha: _____

Pensamiento

*La verdad al cien por cien existe tan poco
como el alcohol al cien por cien.*

Sigmund Freud.

Agradecimientos

Quiero agradecer a mis padres, que me dieron la vida, la guía, el apoyo, la confianza, la voluntad y la fuerza para luchar por lograr mis sueños. Dos seres a los cuales agradezco por existir, por confiar en mí, por no cansarse, por estar siempre, por sencillamente ser como son conmigo, especiales.

*A mi hermano por ser mi mejor amigo, apoyo y guía en todo.
A mis tíos Danilo Zaldivar, Oscar Martínez, Caridad Rizo por su apoyo incondicional y por contribuir tanto en mi formación profesional.*

A mis abuelos Glaciél, Ángel, Himilce, Manuela, siempre me brindaron su apoyo cada vez que los necesité.

Al resto de mi familia por el apoyo que me han dado.

A las chicas del cuarto 143, Jacob, Adito, el negro rockero, el Negrón, Emio, Nildongo, Dayamí a todos los que tanto he molestado con mis problemas y me han soportado todos estos años.

A mi tutor por su gran esfuerzo en que todo saliera adelante y por ayudarme en todo, a Neyda, Aniubis, Virgen, Eloy.

A todos mis amigos de la vieja guardia Yoelkis Pérez, Maykel Pica, Adame, Fidel, Yurisdell, y a todo aquel que de una forma u otra me ha ayudado de manera incondicional.

A mi novio, por tanta comprensión y paciencia.

Todo lo que soy y cuanto soy se lo debo a Uds.

Muchas gracias.

Yusimi Zaldivar Rizo.

Dedicatoria

A mis padres, a quienes más quiero. Muy especial para ti Miraydis Rizo, este gran paso que he dado en mi vida te lo debo especialmente a ti, por tu comprensión, confianza, apoyo incondicional ante cualquier situación, por ser sencillamente como eres conmigo, especial.

A mi hermano, sabes que te quiero.

A mi familia.

A mis amigos.

Yusimi Zaldivar Rizo

Resumen

Como resultado del vertiginoso desarrollo de la informática, las computadoras han dejado de ser simples equipos de cómputo para convertirse en excelentes medios de intercambio de información. Estos medios son un pilar para el funcionamiento de toda institución. Las empresas cubanas del Níquel y en especial La Empleadora del Níquel no se encuentran ajenas a esto, tal es el caso, que se pretende automatizar todos los procesos que así lo permitan.

En el siguiente trabajo se pretende crear un sistema automatizado que permita almacenar las acciones correctivas y preventivas de las notas de no conformidad y los productos no conformes de las diferentes áreas de la Empresa Empleadora del Níquel "EMPLANI". Con el desarrollo del mismo se desea obtener mayor rapidez y organización a la hora de corregir los problemas encontrados en la empresa al realizarse las revisiones del Sistema de Gestión de la Calidad por la dirección, auditorías e inspecciones de procesos.

Para la realización de la investigación se hizo una revisión bibliográfica sobre las aplicaciones Web y las herramientas para la construcción de las mismas, además se presenta la metodología de Ingeniería de Software que se siguió para la construcción del software que se propone como solución de la problemática encontrada.

Abstract

Like result of dizzy development of the informatics, the computers have been left to be simple's equipments of computation for to be converted in an excellent mechanism of information.

These mechanisms are a basin for the good working of any institution.

The Nickel Cubans Company and special the Empleadora del Nickel have not been found extraneous to this.

In this work have been to pretend create an automation system to permit store to correctives and preventives actions of the not conformity notes and not conformity products of the different areas of the company Empleadora del Nickel "EMPLENI".

With the development of the same we desire to obtain major rapid and organization to the time of to correct the problems to be found in the company to realize the check of the Quality Management System for the directive, judge advocate and process inspections.

For the realization of the investigation we did a bibliographical revision about the web applications and the tools of the construction of the same, besides have been a presentation of software engineering methodology of to continued for the construction of the software.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	7
1.1 INTRODUCCIÓN.....	7
1.2 FLUJO ACTUAL DE LOS PROCESOS.....	7
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN.....	7
1.4 ANÁLISIS CRÍTICO DE LA EJECUCIÓN DE LOS PROCESOS.....	8
1.5 ESTADO DEL ARTE.....	8
1.5.1 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	8
1.6 CONCEPTOS FUNDAMENTALES.....	13
1.7 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	13
1.8 TECNOLOGÍAS A UTILIZAR.....	18
1.9 ARQUITECTURA A UTILIZAR.....	25
1.10 HERRAMIENTAS A UTILIZAR EN LA PROPUESTA DE SOLUCIÓN.....	27
1.11 METODOLOGÍA A UTILIZAR PARA LA PROPUESTA DE SOLUCIÓN.....	29
1.12 CONCLUSIONES.....	32
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.	33
2.1 INTRODUCCIÓN.....	33
2.2 MODELO DE DOMINIO.....	34
2.2.1 ¿POR QUÉ MODELO DE DOMINIO?.....	34
2.3 DEFINICIÓN DE LAS ENTIDADES Y CONCEPTOS PRINCIPALES.....	34
2.4 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.....	36
2.4.1 REQUISITOS FUNCIONALES.....	36
2.4.2 REQUISITOS NO FUNCIONALES.....	37
2.5 CONCLUSIONES.....	40
CAPÍTULO 3: ANÁLISIS Y DISEÑO	41
3.1 INTRODUCCIÓN.....	41
3.2 ACTORES DEL SISTEMA A AUTOMATIZAR.....	41
3.3 PAQUETES Y SUS RELACIONES.....	43
3.4 DIAGRAMAS DE CASOS DE USOS DEL SISTEMA A AUTOMATIZAR.....	43
3.4.1 PAQUETE NO CONFORMIDAD.....	44
3.5 DESCRIPCIÓN DEL CASO DE USO GESTIONAR NO CONFORMIDAD.....	44
3.6 DIAGRAMA DE CLASES DEL DISEÑO.....	45
3.6.1 DIAGRAMA DE CLASE WEB INSERTAR PRODUCTO.....	47
3.7 MAPA DE NAVEGACIÓN.....	47
3.8 PRINCIPIOS DE DISEÑO.....	48
3.9 INTERFAZ DE USUARIO.....	48
3.9.1 TRATAMIENTO DE ERRORES.....	48
3.10 DISEÑO DE LA BASE DE DATOS.....	48
3.10.1 MODELO FÍSICO DE DATOS.....	49
3.11 DIAGRAMA DE SECUENCIA.....	50
3.12 DIAGRAMA DE DESPLIEGUE.....	50
3.13 DIAGRAMA DE COMPONENTES.....	51
3.14 CONCLUSIONES.....	53
CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD	54
4.1 INTRODUCCIÓN.....	54
4.2 PLANIFICACIÓN POR PUNTOS DE FUNCIÓN.....	54
4.2.1 CARACTERÍSTICAS DEL PROYECTO.....	58

PODEMOS RESUMIR QUE:	65
4.3 BENEFICIOS TANGIBLES E INTANGIBLES	65
4.4 ANÁLISIS DE COSTOS Y BENEFICIOS	66
4.5 CONCLUSIONES	67
<i>CONCLUSIONES GENERALES</i>	68
<i>RECOMENDACIONES</i>	69
<i>REFERENCIAS BIBLIOGRÁFICAS</i>	70
<i>BIBLIOGRAFÍA</i>	72
<i>GLOSARIO DE TÉRMINOS</i>	74
<i>ANEXOS</i>II
ANEXO 1II
ANEXO 2	XV
ANEXO 3	XVI
ANEXO 4	XVII
ANEXO 5	XX
ANEXO 6	XXI
ANEXO 7	XXV
ANEXO 8	XXVI
ANEXO 9	XXX
ANEXO 10	XXXI
ANEXO 11	XXXVII

Índice de Tablas

TABLA 3.1 DEFINICIÓN DE ACTORES DEL SISTEMA A AUTOMATIZAR	41
TABLA 3.2. DESCRIPCIÓN DEL CU < GESTIONAR NO CONFORMIDAD >	44
TABLA 3.3 TERMINOLOGÍA DE LOS DIAGRAMAS DE CLASES DE DISEÑO.	46
TABLA 3.4 DESCRIPCIÓN TEXTUAL DE LAS CLASES PERSISTENTES	49
TABLA 4.1 ENTRADAS EXTERNAS (EI)	55
TABLA 4.2 SALIDAS EXTERNAS (EO)	56
TABLA 4.3 CONSULTAS EXTERNAS (EQ).....	57
TABLA 4.4 FICHEROS INTERNOS (ILF)	57
TABLA 4.5 PUNTOS DE FUNCIÓN DESAJUSTADOS	58
TABLA 4.6 FACTORES DE ESCALA	59
TABLA 4.7 MULTIPLICADORES DE ESFUERZO	60
TABLA 4.8 CANTIDAD DE INSTRUCCIONES FUENTES	62
TABLA 4.9 ESFUERZO DE DESARROLLO	62
TABLA 4.10 TIEMPO DE DESARROLLO	63
TABLA 4.11 CANTIDAD DE HOMBRES	63
TABLA 4.12 COSTO	64
TABLA 4.13 RESULTADOS DE LAS ESTIMACIONES DE ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO DEL PROYECTO.....	65
TABLA 3.5 DESCRIPCIÓN DEL CU <MOSTRAR REPORTE DE UNA NO CONFORMIDAD >	II
TABLA 3.6 DESCRIPCIÓN DEL CU < IMPRIMIR EL REPORTE >	III
TABLA 3.7 DESCRIPCIÓN DEL CU < NOTIFICAR POR CORREO ELECTRÓNICO >	III
TABLA 3.8 DESCRIPCIÓN DEL CU < INSERTAR ANÁLISIS DE CAUSA_RAIZ >	IV
TABLA 3.9 DESCRIPCIÓN DEL CU < INSERTAR CORRECCIONES >	IV
TABLA 3.10 DESCRIPCIÓN DEL CU < INSERTAR ACCIONES CORRECTIVAS >	V
TABLA 3.11 DESCRIPCIÓN DEL CU < INSERTAR ACCIONES PREVENTIVAS >	V
TABLA 3.12 DESCRIPCIÓN DEL CU < GESTIONAR SEGUIMIENTO>	VI
TABLA 3.13 DESCRIPCIÓN DEL CU < INSERTAR VERIFICACIÓN DEL CUMPLIMIENTO DE LAS ACCIONES>.....	VII
TABLA 3.14 DESCRIPCIÓN DEL CU < < INSERTAR VERIFICACIÓN DEL CUMPLIMIENTO DE LAS CORRECCIONES>	VII
TABLA 3.15 DESCRIPCIÓN DEL CU < CERRAR LA NO CONFORMIDAD >	VIII
TABLA 3.16 DESCRIPCIÓN DEL CU < ELIMINAR UNA NO CONFORMIDAD >.....	IX
TABLA 3.17 DESCRIPCIÓN DEL CU <MODIFICAR DESCRIPCIÓN DE LA NO CONFORMIDAD >	IX
TABLA 3.18 DESCRIPCIÓN DEL CU < MODIFICAR DESCRIPCIÓN DE LA CORRECCIÓN >	X
TABLA 3.19 DESCRIPCIÓN DEL CU < MODIFICAR DESCRIPCIÓN DE LA ACCIÓN >	XI
TABLA 3.20 DESCRIPCIÓN DEL CU < MODIFICAR ANÁLISIS DE CAUSA_RAIZ >.....	XI
TABLA 3.21 DESCRIPCIÓN DEL CU <MODIFICAR FECHA DE EJECUCIÓN DE LA CORRECCIÓN>	XII
TABLA 3.22 DESCRIPCIÓN DEL CU < FECHA DE EJECUCIÓN DE LA ACCIÓN >	XII
TABLA 3.23 DESCRIPCIÓN DEL CU < REPORTE DE ACCIONES >	XIII
TABLA 3.24 DESCRIPCIÓN DEL CU <MOSTRAR REPORTES DE ACCIONES PREVENTIVAS>	XIII
TABLA 3.25 DESCRIPCIÓN DEL CU <MOSTRAR REPORTES DE ACCIONES CORRECTIVAS>	XIV
TABLA 3.26 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_NOTAS	XXVI
TABLA 3.27 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_CAUSA	XXVI
TABLA 3.28 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_VERIFICAR	XXVII
TABLA 3.29 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_CORRECCION	XXVII
TABLA 3.30 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_PRODUCTOS	XXVII
TABLA 3.31 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE SGC_ACCIONES	XXVIII
TABLA 3.32 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE AREAS	XXVIII
TABLA 3.33 DESCRIPCIÓN TEXTUAL DE LA CLASE PERSISTENTE USUARIOS	XXIX

Índice de Figuras

FIGURA 1.1: FASES DE LA METODOLOGÍA RUP	14
FIGURA 1.2: FASES E ITERACIONES DE LA METODOLOGÍA RUP	15
FIGURA 1.3: METODOLOGÍA EXTREME PROGRAMMING	16
FIGURA 1.4: METODOLOGÍA MSF	18
FIGURA 3.1. DIAGRAMA DE PAQUETES	43
FIGURA 3.2 DIAGRAMA DE CASOS DE USO DEL PAQUETE NO CONFORMIDAD.	44
FIGURA 3.3 DIAGRAMA DE CLASE WEB INSERTAR PRODUCTO.....	47
FIGURA 3.4 DIAGRAMA DE SECUENCIA AGREGAR NO CONFORMIDAD.....	50
FIGURA 3.5 DIAGRAMA DE DESPLIEGUE	51
FIGURA 3.6 DIAGRAMA DE COMPONENTES DE SGI	52
FIGURA 3.7 DIAGRAMA DE COMPONENTES DE NO CONFORMIDAD.....	52
FIGURA 3.8 REPRESENTACIÓN DEL MODELO DE DOMINIO.....	XV
FIGURA 3.9 DIAGRAMA DE CASO DE USO DEL SISTEMA A AUTOMATIZAR	XVI
FIGURA 3.10 PAQUETE ANÁLISIS	XVII
FIGURA 3.11 PAQUETE GESTIONAR SEGUIMIENTO.....	XVIII
FIGURA 3.12 PAQUETE REPORTES DE ACCIONES.....	XIX
FIGURA 3.13 MAPA DE NAVEGACIÓN	XX
FIGURA 3.14 DIAGRAMA DE CLASE WEB INSERTAR ANÁLISIS.	XXI
FIGURA 3.15 DIAGRAMA DE CLASE WEB VERIFICAR.	XXII
FIGURA 3.16 DIAGRAMA DE CLASE WEB MOSTRAR.....	XXIII
FIGURA 3.17 DIAGRAMA DE CLASE WEB INSERTAR NOTA.....	XXIV
FIGURA 3.18 DIAGRAMA DE CLASES PERSISTENTE.	XXV
FIGURA 3.19 MODELO FÍSICO DE DATOS.....	XXX
FIGURA 3.20 DIAGRAMA DE SECUENCIA INSERTAR ACCIÓN.....	XXXI
FIGURA 3.21 DIAGRAMA DE SECUENCIA INSERTAR CORRECCIÓN	XXXI
FIGURA 3.22 DIAGRAMA DE SECUENCIA CERRAR NO CONFORMIDAD.	XXXII
FIGURA 3.23 DIAGRAMA DE SECUENCIA MODIFICAR FECHA DE EJECUCIÓN DE UNA ACCIÓN.	XXXII
FIGURA 3.24 DIAGRAMA DE SECUENCIA ELIMINAR NO CONFORMIDAD.	XXXIII
FIGURA 3.25 DIAGRAMA DE SECUENCIA MODIFICAR ANÁLISIS DE CAUSA_RAIZ.....	XXXIII
FIGURA 3.26 DIAGRAMA DE SECUENCIA MODIFICAR DESCRIPCIÓN DE LA CORRECCIÓN.	XXXIV
FIGURA 3.27 DIAGRAMA DE SECUENCIA MODIFICAR DESCRIPCIÓN DE LA NO CONFORMIDAD.....	XXXIV
FIGURA 3.28 DIAGRAMA DE SECUENCIA MODIFICAR DESCRIPCIÓN DE UNA ACCIÓN.	XXXV
FIGURA 3.29 DIAGRAMA DE SECUENCIA MODIFICAR FECHA DE EJECUCIÓN DE UNA CORRECCIÓN.	XXXV
FIGURA 3.30 DIAGRAMA DE SECUENCIA INSERTAR ANÁLISIS	XXXVI
FIGURA 3.31 DIAGRAMA DE COMPONENTES GESTIONAR SEGUIMIENTO.....	XXXVII
FIGURA 3.32 DIAGRAMA DE COMPONENTES REPORTE DE ACCIONES.....	XXXVIII
FIGURA 3.33 DIAGRAMA DE COMPONENTES ANÁLISIS.....	XXXVIII

Introducción

Con el devenir de la era digital y el auge alcanzado por las nuevas tecnologías y los productos informáticos en el mercado a nivel mundial, así como la dependencia creada por las diferentes ramas de la sociedad hacia esas tecnologías, productos y servicios, se ha hecho necesaria la creación de sistemas que mejoren y faciliten el uso de la información. Los sistemas informáticos para la gestión de la información en diferentes esferas, son los software más utilizados a nivel mundial.

Actualmente en el mundo es característico el desarrollo acelerado de las tecnologías. Ninguna empresa o entidad podrá tener éxito si no es capaz de ajustarse a los nuevos cambios aprovechando los beneficios de esta nueva era, en la que la economía global depende cada vez más de sistemas automatizados y el control de los procesos que se realizan en las empresas.

La “Gestión de la Calidad” es una forma de trabajar mediante la cual una organización asegura que se identifiquen y satisfagan las necesidades de su cliente, planificando, manteniendo y mejorando el desempeño de sus procesos de manera eficaz y eficiente, con el objeto de lograr ventajas competitivas.

Además, una buena calidad puede significar cubrir los estándares mínimos para una atención adecuada o alcanzar altos estándares de excelencia. La calidad puede referirse a la calidad técnica de la atención; a los aspectos no técnicos de la prestación de servicios, tales como el tiempo de espera del cliente y las actitudes del personal; y a los elementos programáticos, tales como la política, la infraestructura, el acceso y la administración. Es decir el concepto de gestión es un paso adelante en el camino hacia la calidad.

En el sector de gestión de la calidad en las empresas se utilizan diversos métodos y herramientas para el control y el buen funcionamiento de la misma, con esto estamos haciendo alusión a las Auditorías Internas y Externas, las Inspecciones de Procesos y la Revisión del Sistema de Gestión de la Calidad por la Dirección de la entidad. Las Auditorías constituyen la herramienta ideal para revisar y comprobar las estrategias y planes de la empresa, son realizadas por personal

capacitado ya sean externas por personal de fuera de la empresa o internas por auditores de la empresa pero pertenecientes a otras áreas. De igual forma es llevada la inspección de procesos pero como su nombre lo indica es la inspección que se realiza a los procesos de un área determinada pero por los mismos trabajadores del área, los cuales pueden ser los especialistas de áreas entre otros. La Revisión del Sistema de Gestión de la Calidad por la Dirección es la más amplia, puesto que revisa todo el Sistema de Gestión de la Calidad en la empresa y es ejecutada por el especialista en Gestión de la Calidad de la empresa en conjunto con un comité de revisión designado para realizar esta tarea.

Estas herramientas facilitan la detección de los problemas que existen en los procesos que definen el funcionamiento de la entidad, lo que permite tomar medidas para erradicar estos malos funcionamientos en los procesos, como contraparte a estos problemas surgen las notas de no conformidad y productos no conformes que es el mecanismo para dejar definido y evidenciado, cuáles son los problemas, qué procesos han sido afectados y dónde está referenciado. Las no conformidades son de gran importancia puesto que permiten a los trabajadores de la empresa realizar el análisis de causa raíz y tomar las acciones correctivas o preventivas que no son más que el análisis profundo de los problemas y las medidas que se toman para suprimir los mismos.

Nuestro país se ha visto envuelto en la necesidad de crear, desarrollar y consolidar los Comités de Calidad, como vía adecuada para estimular la iniciativa creadora de los trabajadores en el mejoramiento de la Calidad, de ahí que tenga gran importancia la constante elevación de la calidad y la eficiencia económica en la generación de productos y la prestación de servicios, al menor costo.

La Industria Cubana del Níquel, uno de los pilares más importantes en el desarrollo económico de nuestro país, en el transcurso de los años ha incursionado en el perfeccionamiento de la calidad de los procesos que se desarrollan en cada una de sus empresas, logrado esto a través de la implantación de los Sistemas de Gestión de la Calidad, lo que ha permitido un mejor desempeño de las mismas.

La Empleadora del Níquel (EMPLINI), no se encuentra ajena a esto, en la misma existe implantado un Sistema de Gestión de la Calidad que rige toda la política de dicha Empresa a través de diferentes Procedimientos. Este Sistema de Gestión de la Calidad tiene como objetivo controlar el buen funcionamiento de la entidad, lo cual es posible a través de las herramientas de control y supervisión mencionadas anteriormente.

Actualmente en la Empresa Empleadora del Níquel el proceso de levantamiento de notas de no conformidad y productos no conformes se realiza de forma manual, lo que hace el proceso más lento, no solo a la hora de su redacción, sino también en el momento de darles seguimiento a las acciones tomadas, además esto aumenta los gastos de papel y otros materiales de oficina e incrementa la morosidad del seguimiento de las mismas, puesto que no permite que la actividad sea interactiva. También se torna un poco engorroso en el momento de generar diversos reportes correspondientes a las no conformidades.

Partiendo de esta situación problemática se decidió llevar a cabo la presente investigación definiendo la siguiente pregunta:

¿Cómo agilizar y a su vez facilitar el proceso de control y seguimiento de las acciones correctivas y preventivas tomadas a partir de las no conformidades levantadas en la Empresa Empleadora del Níquel?

Como **objeto de la investigación** se tiene: El Sistema de Gestión de la Calidad y el **campo de acción**: la automatización del proceso de control y seguimiento de las acciones correctivas y preventivas en la entidad.

El **Objetivo General** de la Investigación es desarrollar un software que permita facilitar y agilizar el control y seguimiento de las acciones correctivas y preventivas tomadas a partir de las no conformidades levantadas en la Empresa Empleadora del Níquel.

Como **Objetivos específicos**:

- ✓ Elaborar el Marco teórico conceptual.
- ✓ Realizar un diagnóstico sobre el Sistema de Gestión de la Calidad en la Empresa Empleadora del Níquel.

- ✓ Realizar análisis, diseño e implementación del sistema.
- ✓ Realización del Estudio de Factibilidad.
- ✓ Implantación del software.

Para dar respuesta al Problema Científico se plantea como **hipótesis** que: la creación de un sistema automatizado para el control de las acciones correctivas y preventivas tomadas a partir de las no conformidades levantadas, agilizará esta actividad y además permitirá realizar el seguimiento y comprobar la eficacia de las mismas en la Empresa Empleadora del Níquel, mejorando así el desarrollo del Sistema de Gestión de la Calidad en la organización.

Para lograr los objetivos propuestos se han trazado una serie de **tareas** a realizar:

- ✓ Revisión de la literatura relacionada con el tema, observaciones en el campo del objeto de estudio y entrevistas realizadas a especialistas.
- ✓ Buscar referencias bibliográficas.
- ✓ Realizar el modelo de dominio y levantamiento de requisitos.
- ✓ Elaborar los modelos del análisis y diseño del sistema.
- ✓ Efectuar el desarrollo de la aplicación.

Para complementar estas tareas se han empleado métodos empíricos y teóricos de la investigación científica.

Los métodos empíricos ayudan en el descubrimiento de los hechos, información, procesamiento de datos y en el conocimiento de las características fundamentales del problema, los que posibilitan su estudio y explicación.

Entre los métodos empíricos usados podemos citar:

Observación: se utiliza para ver la funcionalidad de las diferentes áreas de la entidad y el comportamiento del problema.

Entrevista: se realizó una conversación planificada con el fin de obtener información individual o colectiva y determinar los principales requerimientos del sistema.

Análisis de documentos: permitió conocer cómo funcionan actualmente los procesos de control del sistema de gestión de la calidad en la entidad.

Los métodos teóricos se tuvieron en cuenta durante el transcurso de la investigación; pues crearon las condiciones para la interpretación y desarrollo de las teorías de interpretación de los datos obtenidos, o sea, para profundizar en las características fenomenológicas superficiales de la realidad.

Los **métodos teóricos** utilizados fueron:

Método Histórico _ lógico: permitió la búsqueda de los antecedentes del software así como la necesidad de su desarrollo en una forma superior.

Análisis y Síntesis: Se utilizó para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación.

Hipotético _ deductivo: para la elaboración de la hipótesis y su verificación.

Mediante la modelación se realizó el estudio del proceso de control de las acciones correctivas y preventivas en la entidad.

Esta investigación tiene como **aporte práctico** el Diseño e Implementación del Sistema Automatizado para el control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel (EMPL ENI).

El presente trabajo consta de introducción, 4 capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos:

En el Capítulo 1, Fundamentación teórica, se ofrece una breve descripción de diferentes conceptos imprescindibles que le dan base a la investigación. Así como algunos elementos importantes de las herramientas a utilizar para la implementación del sistema.

En el Capítulo 2, Características de Sistema, se abordará la descripción del modelo de dominio. Se definen las entidades y conceptos principales, además de las funcionalidades que va a concebir el sistema.

En el Capítulo 3, Análisis y Diseño, se describen en detalles los flujos de trabajos relacionados a estas etapas de diseño e implementación de la metodología utilizada, RUP.

En el Capítulo 4, Estudio de Factibilidad, se realiza un estudio de los esfuerzos requeridos para la construcción del sistema, y se valora la sostenibilidad del producto.

Para finalizar se muestran las Conclusiones a las que se arribaron, las Recomendaciones que se proponen, la Bibliografía utilizada, Glosario de Términos y Anexos con información necesaria sobre el trabajo.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

Hoy en día la informática va evolucionando considerablemente y junto con ella la producción de los diversos software. El siguiente capítulo es el resultado de una investigación acerca de las herramientas y metodologías existentes, con las cuales se le pudiera dar solución a nuestro problema planteado, se hace referencia al flujo actual de los procesos, se realiza un análisis crítico de la ejecución de los procesos, una breve referencia sobre el estado del arte del sistema, es decir un análisis de los productos de software existentes en el país y el extranjero vinculado con el campo de acción del proyecto, además de una descripción sobre las tendencias y tecnologías actuales sobre las que se apoya el sistema.

1.2 Flujo Actual de los procesos.

El Sistema de Gestión de la Calidad en las empresas es de vital importancia, por lo que debe ser interés de todos los especialistas y trabajadores de cada entidad, velar por el buen funcionamiento del mismo.

Actualmente en la Empresa Empleadora del Níquel (EMPLENI), las Acciones Correctivas y Preventivas tomadas a partir del análisis realizado a las No Conformidades levantadas a raíz de las deficiencias encontradas en las auditorías, inspecciones de procesos y revisión del Sistema de Gestión de la Calidad por la dirección, se realiza de forma manual, la misma se ejecuta a través de una planilla que sirve como guía para desarrollar todo este proceso y de esta forma almacenar las evidencias que son en concreto, el desarrollo desde el levantamiento de una No Conformidad, el seguimiento que se le realiza hasta que esta es cerrada.

1.3 Procesos objeto de automatización.

Anteriormente se veía el problema que afectaba el Sistema de Gestión de la Calidad producto a lo incómodo que se torna una vez realizadas las Auditorías, Inspecciones de Procesos y Revisiones de Sistema de Gestión de la Calidad por la dirección levantar las Notas de No Conformidad en planillas que a la hora de

ejecutar el tratamiento adecuado que lleva este proceso se hace un poco engorroso la localización de estas planillas y la revisión de las mismas. Por ello es necesario automatizar el proceso de levantamiento de No Conformidad que está incluido dentro del procedimiento de Sistema de Gestión de la Calidad y de esta forma permitir que se pueda encontrar al alcance de todos los técnicos y especialistas usuarios de la red. Como resultado de este análisis surge la necesidad de desarrollar un sistema que resuelva los problemas antes abordados.

1.4 Análisis Crítico de la Ejecución de los Procesos

Los Especialistas de Áreas, Especialista en Gestión de la Calidad y Auditores, basándose en el Sistema de Gestión de la Calidad implantado en la entidad, deben realizar periódicamente las Inspecciones de Procesos, Revisión de Sistema de Gestión de la Calidad por la dirección y Auditorías planificadas respectivamente, con el objetivo de encontrar las deficiencias que estén afectando el buen funcionamiento de la Empresa y de esta forma poder realizar un análisis profundo que les permita tomar las medidas adecuadas para erradicar estos problemas.

1.5 Estado del arte.

1.5.1 Sistemas automatizados existentes vinculados al campo de acción.

Después de una profunda búsqueda en numerosas fuentes de información y entre ellas Internet (nacional e internacional), se pudo observar que existen varios sistemas automatizados vinculados al campo de acción que le ocupa a este trabajo. Consideremos los siguientes:

KMKey (Knowledge Management Key)

Es un programa que le permite gestionar proyectos, seguir cualquier Sistema de Gestión de Calidad y gestionar las incidencias de su empresa en un entorno Web.

Se trata de software libre y por tanto no lleva asociado ningún coste por la "licencia de uso" siendo estas ilimitadas. KMKey es un producto de Earcon, S.L. Esta forma parte de la asociación de software libre CatPL.

Está desarrollado en lenguaje **Python** sobre el servidor de aplicaciones **Zope** y el gestor de contenidos **CPS**, y su principal utilidad es ofrecer una plataforma que cruza la planificación de proyectos con la gestión de contenidos y un cuadro de mando, convirtiéndolo en un entorno colaborativo tremendamente útil para cualquier empresa de servicios.

Como valor añadido, el producto tiene una larga trayectoria empresarial y se encuentra en producción en multitud de empresas y organizaciones reales. En esta nueva versión se ha mejorado considerablemente la usabilidad, el rendimiento, y las utilidades disponibles para el administrador. También se ha añadido la exportación e importación de patrones de trabajo con el fin de fomentar el intercambio de los mismos entre usuarios, y se han añadido nuevas vistas en el cuadro de mando.

Características

- **Entorno Web:** aplicación 100% programada para ser utilizada vía Internet. Se trabaja siempre mediante un navegador (Internet Explorer o Firefox) desde cualquier lugar con acceso a Internet.
- **Información accesible mediante permisos:** acceso a los expedientes donde el usuario ha sido asignado y con la posibilidad de realizar las acciones correspondientes a su perfil.
- **Software libre:** todos los sistemas operativos, bases de datos y programas necesarios para la instalación de KMKey son Open Source (licencia GPL).

No se abonan cantidades por concepto de "licencia de uso". Los costes de implantación son proporcionales a los servicios prestados.

ISOKEY

∅ ISOKEY es un sistema informático que brinda un soporte tecnológico para sistemas de gestión ISO 9000, ISO 14000, OHSAS 18000, ISO 22000, ISO-IEC 17999, ISO TS 16949, entre otros marcos normativos

internacionalmente reconocidos y aplicados. Es un producto creado por la Asociación Española de Normalización (AENOR). Facilita la implantación y mantenimiento de los requisitos de estas normas de gestión, garantizando su cumplimiento y certificación, y apoyando la mejora continua de las organizaciones.

- Ø ISOKEY permite gestionar requisitos como la estructura organizativa, la estrategia, los objetivos, los procesos, el control de la documentación, las encuestas, las auditorías, los proyectos, el mantenimiento y calibración de equipos, la capacitación del personal, las no conformidades y los incidentes, las quejas y las acciones correctivas, preventivas y de mejora.
- Ø ISOKEY maneja una amplia gama de notificaciones automáticas como forma de recordar o hacer notar a los usuarios determinados eventos del sistema. ISOKEY notifica el inicio y vencimiento de tareas relacionadas con asuntos, la eventualidad de mensajes levantados a los foros, los pendientes de revisión, aprobación y distribución de documentos, entre otros.

QACTION

Software para el control de Auditorías, Acciones Correctivas y Preventivas.

QAction es un software pensado en el usuario confeccionado por la Empresa Wilsoft. Lleva un control completo de todas las Auditorías, Quejas de clientes, Reportes de productos o servicios no conformes y de las Acciones Correctivas y Preventivas.

El sistema toma el control y da seguimiento a los problemas o no conformidades detectados, se encarga de avisarles a los responsables de los problemas así como a la persona que detectó la no conformidad cuando se haya dado respuesta con acciones correctivas.

QAction tiene el control de los tiempos para avisarles a las personas que corresponda en cada caso a través del correo electrónico. El papeleo se reducirá

considerablemente en su empresa ya que la mayor parte de la evidencia quedará registrada de forma electrónica.

Independientemente de que este software cuenta con las características relacionadas al campo de acción debemos decir que no es factible adquirirlo puesto que se comercializa con un costo que oscila entre los 7.990 y 8000 EUR.

GQUAL

GQUAL es una aplicación desarrollada con un objetivo principal, automatizar todas las labores administrativas que conlleva la obtención, gestión diaria y mantenimiento de los sistemas de gestión, basados en las principales normas de calidad, medio ambiente y prevención.

Sus características principales incluyen, entre otras muchas funcionalidades:

- **ENVIO AUTOMÁTICO DE AVISOS POR MEDIO DE CORREO ELECTRÓNICO** a todos los Usuarios que tienen que realizar alguna acción para el cumplimiento de los requisitos de las normas del sistema de gestión, en todos los módulos.
- **CUMPLIMIENTO AUTOMÁTICO DEL 100% DE LOS REQUISITOS DE LAS NORMAS ISO** en materia de gestión documental, GQual se encarga generar automáticamente todos los aspectos necesarios para el sistema de gestión, sustituyendo muchos formatos requeridos por las normas.
- **SUSTITUCIÓN DE LOS DOCUMENTOS EN PAPEL**, una vez implantado nuestros Clientes, reducen su documentación en soporte papel, en más del 90%, quedando únicamente aquellos registros que deben ir impresos.
- **POSIBILIDAD DE PARAMETRIZACIÓN DE ROLES Y ACCESOS DE CADA USUARIO**, que permite que cada persona que acceda al sistema sólo tendrá acceso a la información y acciones, para la cuál esté autorizado por los

administradores. El resto de documentos, información y botones de acciones no son visibles por los usuarios no autorizados.

Sistema automatizado para el control de la calidad de un laboratorio farmacéutico

Se desarrolla un sistema automatizado utilizando el sistema de Gestión de Bases de Datos ORACLE, versión 6.0, soportado sobre el Sistema Operativo UNIX, apoyándose en las normativas de las "Buenas practicas de fabricación" (GMP Good Manufacturing Practices).

En el laboratorio farmacéutico "Roberto Escudero Díaz" de la industria Médico Farmacéutica (IMEFA), se cuenta con un sistema automatizado para el control de las actividades de la Sub-Dirección de calidad, que forma parte del sistema para el control de la actividad del mismo.

Este sistema constituye lo que se conoce como sistema Computer Integrated Manufacturing. Además realiza de forma automática los controles de calidad a la materia prima, al material de envase, al producto en proceso y al producto terminado, consultando para esto una extensa farmacopea que especifica para cada uno de ellos, según el fabricante, los ensayos que deben realizarse y así poder garantizar si cumplen o no los límites establecidos para su aprobación. También, se automatizan los registros de las materias primas, material de envase y del producto en proceso que permiten llevar un estricto control de datos de los mismos por lotes, así como las fechas de entradas y terminación de ensayos que requieren cierto tiempo o requieren realizarse fuera del centro por distintos motivos, además de tener alguna otra información de interés que garantiza, una vez terminado el producto, emitir el vale de aprobación de producción terminada y lista para ser distribuida.

Las inspecciones que se llevan a cabo durante el proceso productivo en todo los niveles son otra de las actividades que también se encuentran automatizados dentro de este Sistema de Control de Calidad; así como la del control del expediente de Producto terminado.

El sistema además contempla la posibilidad de emitir periódicamente, como resumen de toda esta actividad, distintos reportes donde se brinda información a la IMEFA del Sistema de Inspección, análisis y decisión de materia prima, material de envase, producto en proceso, producto terminado y proceso productivo.

1.6 Conceptos Fundamentales.

La *ingeniería de software (ISW)* afecta a la economía y las sociedades de muchas maneras. Este término se refiere al área de las ciencias de la computación que trata con la construcción de sistemas de software, los cuales son tan grandes y complejos que se construyen con equipos de ingenieros. Es un proceso definido paso a paso, que facilita la especificación, el diseño, la implementación y las pruebas de una solución de software, para un conjunto de requisitos explícitos.

La *gestión de software (GS)* es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos; garantizando la calidad del producto. La GS comprende la gestión de configuración y la gestión de proyecto.

1.7 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Las técnicas indican cómo debe ser realizada una actividad técnica determinada identificada en la metodología. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. Indica cómo hay que obtener los distintos productos parciales y finales.

Aquí se describen las principales características de tres de las más famosas y conocidas metodologías de desarrollo de software: Proceso Unificado de Racional (Rational Unified Process, RUP), Programación Extrema (Extreme Programming, XP) y Microsoft Solution Framework (MSF).

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:



Figura 1.1: Fases de la Metodología RUP.

- Ø **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- Ø **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- Ø **Construcción:** En esta etapa el objetivo es obtener la capacidad operacional inicial.
- Ø **Transición:** El objetivo es llegar a obtener la versión lista para su instalación en las condiciones reales.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ø **Modelado empresarial:** Entendiendo las necesidades del negocio.
- Ø **Requerimientos:** Trasladando las necesidades del negocio a un sistema automatizado.

- ∅ **Análisis y Diseño:** Trasladando los requerimientos dentro de la arquitectura de software.
- ∅ **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ∅ **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- ∅ **Despliegue:** Describe las actividades asociadas al garantizar que el producto de software esté disponible para los usuarios.
- ∅ **Gestión de cambios y configuración:** Explica cómo controlar y sincronizar la evolución del conjunto de productos de trabajo que componen un sistema de software.
- ∅ **Gestión de proyecto:** Planificación del proyecto, la gestión del riesgo, la supervisión del progreso y la métrica.
- ∅ **Entorno:** Organiza esos elementos de método que proporcionan el entorno de desarrollo de software que da soporte al equipo de desarrollo, incluidos los procesos y las herramientas.
- ∅ **Distribución:** Hacer todo lo necesario para la salida del proyecto

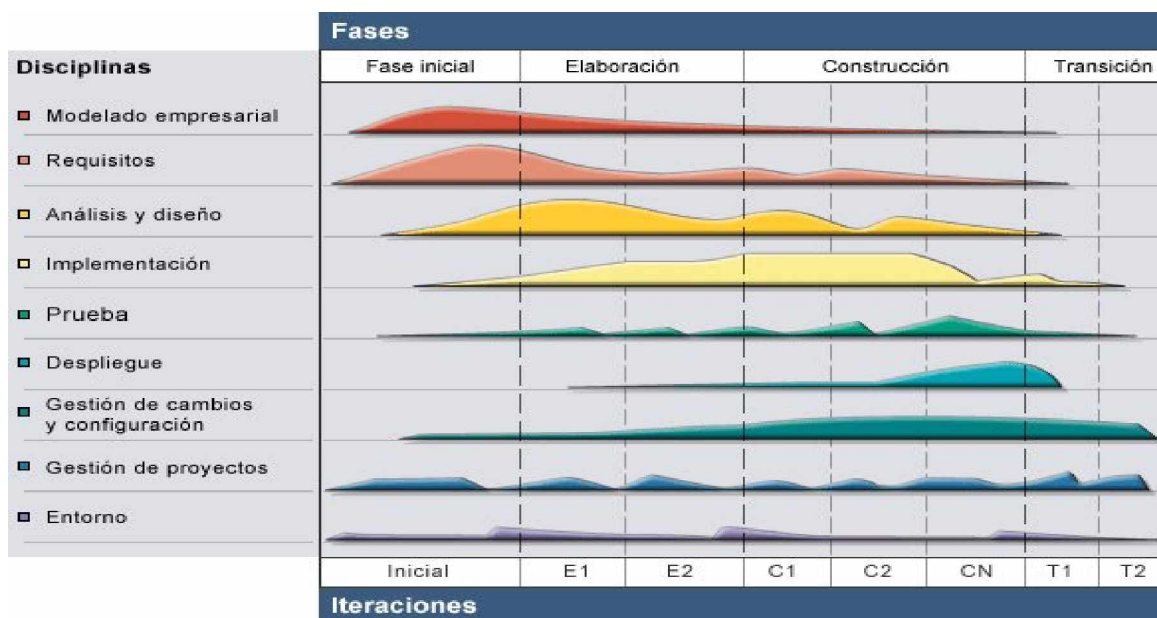


Figura 1.2: Fases e Iteraciones de la Metodología RUP.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente.

Esto trae como beneficio la retroalimentación que se tendrá en cada entregable o en cada iteración.

Los elementos del RUP son:

Actividades: Son los procesos que se llegan a determinar en cada iteración.

Trabajadores: Son las personas o entes involucrados en cada proceso.

Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

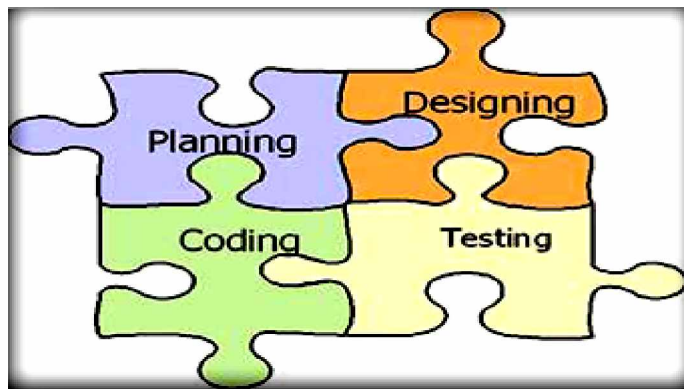


Figura 1.3: Metodología Extreme Programming.

La metodología XP se basa en:

- Ø **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, realizando pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

- Ø **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Ø **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué es lo que propone XP?

- Ø Añade funcionalidad con retroalimentación continua.
- Ø El manejo del cambio se convierte en parte sustantiva del proceso.
- Ø El costo del cambio no depende de la fase o etapa.
- Ø No introduce funcionalidades antes que sean necesarias.
- Ø El cliente o el usuario se convierte en miembro del equipo Derechos del Cliente.
- Ø Saber el estado real y el progreso del proyecto.
- Ø Añadir, cambiar o quitar requerimientos en cualquier momento.
- Ø Obtener lo máximo de cada semana de trabajo.
- Ø Decidir cómo se implementan los procesos.
- Ø Crear el sistema con la mejor calidad posible.
- Ø Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Ø Estimar el esfuerzo para implementar el sistema.
- Ø Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- Ø La comunicación entre los usuarios y los desarrolladores.
- Ø La simplicidad al desarrollar y codificar los módulos del sistema.
- Ø La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión

de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



Figura 1.4: Metodología MSF.

MSF tiene las siguientes características:

- Ø **Adaptable:** es altamente adaptable, lo que permite entregar soluciones de alta calidad y de manera rápida mientras que minimiza el riesgo asociado a todo proyecto ya que éste considera las causas más comunes que hacen fracasar los proyectos de tecnología.
- Ø **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren aproximadamente 50 personas.
- Ø **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- Ø **Tecnología Agnóstica:** puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.8 Tecnologías a utilizar.

PHP v 5.0 o superior

Las iniciales PHP significan "PHP Hypertext Pre-processor" y se trata de un lenguaje de programación que es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Es un lenguaje de programación usado generalmente para la creación de contenido para sitios o

aplicaciones Web. La versión 5.0 de PHP presenta un magnífico trabajo con el Paradigma Orientado a Objeto que permite la reutilización de código entre otras facilidades.

Como síntesis, PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas, y contiene unas 40 extensiones estables sin contar las que se están experimentando, también tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. Además de que:

- Ø Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.
- Ø Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Ø Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resultará muy fácil aprender PHP.
- Ø Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.
- Ø PHP tiene una de las comunidades más grandes en Internet, esto permite encontrar fácilmente ayuda, documentación, artículos, noticias, y otros recursos.
- Ø Permite las técnicas de Programación Orientada a Objetos.
- Ø Posibilita crear los formularios para la Web.
- Ø No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Ventajas de trabajar con PHP

- Ø Es un lenguaje multiplataforma.
- Ø Rapidez de ejecución.
- Ø Mantiene un bajo consumo de recursos de máquina.

- Ø Gran seguridad, muy poca probabilidad de corromper los datos.
- Ø Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Ø Posee una amplia documentación en Internet, incluyendo una gran variedad de ejemplos y de ayudas.

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- Ø XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Ø Document Object Model (DOM) accedido con un lenguaje script por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- Ø El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- Ø XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON(JavaScript Object Notation) y hasta EBML(Meta Lenguaje Binario Extendible).

Como el DHTML (HTML Dinámico), LAMP (se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios Web o Servidores dinámicos con un esfuerzo reducido), AJAX no

constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Ventajas:

- ∅ Recuperación asíncrona de datos, el usuario no tiene que esperar después de una petición.
- ∅ Acercamiento del ambiente de escritorio a la Web.
- ∅ No requiere plugins(es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica).

- ∅ Se reduce el tamaño de la información intercambiada.

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSIC.

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

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS.

Posee características que son implementadas únicamente por este gestor de base de datos:

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

Ventajas

- Ø Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Ø Mejores utilidades de administración (backup, recuperación de errores, etc.). Aunque se cuelgue, no suele perder información ni corromper los datos.
- Ø Mejor integración con PHP.
- Ø No hay límites en el tamaño de los registros.
- Ø Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- Ø MySQL se comporta mejor a la hora de modificar o añadir campos a una tabla "en caliente".
- Ø Es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes.
- Ø Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad.
- Ø Cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor.
- Ø Es un sistema de administración de Base de Datos. Opera en una arquitectura cliente/servidor.
- Ø Es el sistema gestor de bases de datos "Open Source" más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera.
- Ø No se necesita de un manejo complejo de la información.

- Ø El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- Ø El MySQL no tiene precio en el mercado, se adquiere libremente.

Apache

Servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente (incluso forks y productos propietarios) siempre que les reconozcas su trabajo.

Es el servidor Web más utilizado en el mundo con un 57 % de cuota de mercado, frente al 20 % de Microsoft IIS y el 7 % de Netscape. Dentro de sus puntos fuertes se encuentran:

- Ø Tiene interfaz con todos los sistemas de autenticación.
- Ø Facilita la integración como "plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Ø Tiene integración en estándar del protocolo de seguridad SSL. (más utilizado).
- Ø Provee interfaz a todas las bases de datos.

A continuación se hace referencia de por qué este software libre goza de tanta popularidad en muchos ámbitos empresariales y tecnológicos.

- Ø Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Ø Es una tecnología gratuita de código abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.

- Ø Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
 - Ø Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
 - Ø Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte modperl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
 - Ø Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
 - Ø Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.
- Se pueden extender las características y facilidades que brinda apache como servidor Web hasta donde nuestra imaginación y conocimientos sean capaces de llegar.

Independientemente de que existen muchas herramientas para el desarrollo Web, tanto de lenguajes de programación, como de gestores de bases de datos e infraestructura de servidores, en esta investigación sólo se reflejan las características de los que a continuación les mostramos:

- Ø Lenguaje de programación, Personal Home Page (PHP).
- Ø Gestor de bases de datos, MySQL.
- Ø Infraestructura de Servidor Web, XAMPP.

Esto es debido a que en esta empresa se cuenta con el diseño de un Sistema Integrador que recoge todos los módulos o subsistemas que han sido

desarrollados, de esta forma podemos decir que es fuente de desarrollo de las nuevas necesidades y es política de la empresa la utilización de estas tecnologías.

1.9 Arquitectura a utilizar.

Arquitectura

Una arquitectura es el conjunto de decisiones significativas sobre la organización del SS, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización, estos elementos y sus interfaces, sus colaboraciones, y su composición. Ej.: los Patrones de diseño relacionados con el diseño de los objetos y frameworks de pequeña y mediana escala, que son aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura, y durante el trabajo de diseño detallado para cualquier aspecto del diseño local. También se conocen como patrones de micro-arquitectura. El patrón Fachada, que se puede utilizar para proporcionar la interfaz de una capa a la siguiente.

Arquitecturas en Capas

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo o por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente. (Garlan y Shaw) definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior.

La arquitectura por capas es un estilo de arquitectura en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, simplemente es necesario conocer la *API* que existe entre niveles.

En el diseño de sistemas informáticos actuales se suele usar las arquitecturas multinivel o **Programación por capas**. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más en boga actualmente es el diseño en tres niveles (o en tres capas).

Capas o niveles

- ∅ **Capa de presentación:** Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.
- ∅ **Capa de negocio:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de *base de datos* para almacenar o recuperar datos de él.
- ∅ **Capa de datos:** Es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

La Arquitectura en Capas o Arquitectura *n* capas es la que se ha seleccionado para darle solución a la problemática planteada anteriormente, debido a las facilidades presentadas anteriormente.

1.10 Herramientas a utilizar en la propuesta de solución.

Dreanweaver 8

Dreamweaver 8 es la opción profesional para la creación de sitios y aplicaciones Web. Proporciona una combinación potente de herramientas visuales de disposición, permite desarrollo de aplicaciones y soporte para la edición de código. Gracias a las robustas características para la integración y diseño basado en CSS. Incluye potentes controles basados en normas para asegurar un diseño de alta calidad. Un entorno de diseño construido en torno a las hojas de estilo en cascada (CSS) hace posible un desarrollo más rápido y más eficiente de sitios profesionales creados con código limpio. Es un entorno de desarrollo integrado para desarrollar sitios Web de HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP y Macromedia ColdFusion.

Rational Rose Enterprise

Rational Rose Enterprise proporciona un lenguaje de modelado común para permitir la creación más rápida de la calidad de software. Incluye Unified Modeling Language (UML) de apoyo y es una de la más completa de productos de la familia Rational Rose. Proporciona la base de datos para el modelado UML diseños, con la capacidad de representar a la integración de los datos y los requisitos de las aplicaciones a través de diseños físicos y lógicos. Permite especificar, analizar, diseñar el sistema antes de Codificar. Mantiene la consistencia de los modelos del sistema software, además de la generación de documentos automáticamente. Otra funcionalidad de la herramienta es que permite realizar la ingeniería inversa de un producto.

Xampp

Es un servidor independiente de plataforma, software libre, incluye el servidor Web Apache, los servidores de datos MySQL y SQLite, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en PEAR, el intérprete del lenguaje Perl, servidores de FTP como

ProFTPD ó FileZilla FTP Server, las estadísticas Webalizer y OpenSSL, eAccelerator, Freetype2, libjpeg, libpng, zlib, Ming, etc. entre otros. El programa esta liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios Webs y programadores testar su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.

Ventajas:

- Ø Incluye Chequeo de seguridad.
- Ø Contiene un Panel de control.
- Ø Incluye herramientas extras (Webalizer, Mercury Mail, Conmutador PHP).

Embarcadero ER/Studio 6.0.1

Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. ER/Studio se diseña para hacer más fácil de entender el estado actual de los datos de las empresas. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa.

- Ø **Potencial de ER/Studio:** Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia. Con el potencial y la facilidad de empleo de ER/Studio's que modela el ambiente, será productivo rápidamente y podrá casi demostrar resultados inmediatamente después de la instalación.

- Ø **Diagramas:** La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento, y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El Explorer Navigation facilita el trabajo hasta con los diagramas más grandes. Si se está trabajando con un modelo largo de Datos, ER/Studio ofrece un aumento en la ayuda y fácil navegación en sus modelos. La Apreciación global (overview). Se usa el browser Explorer para encontrar y seleccionar entidades. Un solo clic inmediatamente enfoca una ventana de diagrama.

Ventajas:

- Ø Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.
- Ø ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.
- Ø Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que está disponible para presentaciones profesionales.
- Ø Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

1.11 Metodología a utilizar para la propuesta de solución.

Después de realizar un análisis e investigación de algunas metodologías, se elige utilizar RUP para el desarrollo del software, ya que es la más completa y abarcadora, pues como señalan algunos autores, las otras metodologías son casos particulares de esta. Además, XP y MSF presentan algunas debilidades, lo que representa riesgos considerables, como dificultades a la hora de una buena obtención de requisitos para el sistema. RUP es adaptable, según el tipo de proyecto, así serán las características del mismo, haciéndose énfasis en aquellos

flujos de trabajo durante la vida del software, que reporten más importancia y sean indispensables. RUP permite trazarse planes de riesgos y pruebas durante el ciclo de vida. Contiene artefactos que son diseñados durante las diferentes fases, los cuales describen detalladamente las características del software desde que se realiza el análisis del problema hasta la entrega final del producto.

UML

El Proceso Unificado RUP (Racional Unified Process) utiliza el Lenguaje de Modelado UML (Unified Modeling Language) para preparar todos los esquemas de un sistema software. UML es una parte esencial del Proceso Unificado, está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

UML recomienda 9 diagramas que son modelados durante las diferentes fases de desarrollo del software, es decir, representan las siguientes vistas del sistema:

- Ø **Diagrama de Casos de Uso:** modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- Ø **Diagrama de Clases:** muestra las clases que componen el sistema y cómo se relacionan entre sí.
- Ø **Diagrama de Objetos:** muestra una serie de objetos y sus relaciones.
- Ø **Diagrama de Secuencia:** enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.
- Ø **Diagrama de Colaboración:** igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- Ø **Diagrama de Estados:** modela el comportamiento de acuerdo con eventos.
- Ø **Diagrama de Actividades:** simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.

- Ø **Diagrama de Componentes:** muestra la organización y las dependencias entre un conjunto de componentes.
- Ø **Diagrama de Despliegue:** muestra la organización y las dependencias entre un conjunto de componentes.

1.12 Conclusiones.

En este capítulo quedaron reflejados todos los conceptos relacionados con el problema, ampliando así los conocimientos para comprender mejor los diferentes procesos que ocurren en la entidad. Se hizo un análisis del estado del arte, las tecnologías y herramientas informáticas, procesos de desarrollo de software, lenguajes de programación y tendencias que fundamentan la solución propuesta.

Capítulo 2: Características del Sistema.

2.1 Introducción.

Analizando la descripción realizada de los procesos que se tienen en cuenta en esta investigación, podemos decir que independientemente de que el proceso del Sistema de Gestión de la Calidad de una empresa es bastante complejo, en el caso que nos ocupa mantiene un bajo nivel de estructuración, esto es palpable puesto que el producto final al que queremos llegar, es sólo una herramienta donde se pueda almacenar toda la información acerca de las no conformidades que se deriven de las diferentes situaciones por las que atraviesa el procedimiento del sistema, en estos casos de simplicidad, la metodología RUP propone realizar un modelo de dominio; que no es más que una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, por lo que permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Tal modelo no incluye las responsabilidades que llevan a cabo las personas, de modo que en el presente capítulo se especificará lo siguiente:

1. Definición de entidades y los conceptos principales, los cuales se definirán a partir de su símbolo, su intención, y su extensión.

Símbolo: Palabra que representa al concepto.

Intención: La definición del concepto.

Extensión: El conjunto de ejemplos a que se aplica el concepto.

2. Diagrama del Modelo del Dominio.
3. Representación de los requerimientos funcionales y no funcionales del sistema.

2.2 Modelo de Dominio

2.2.1 ¿Por qué Modelo de Dominio?

La metodología RUP propone dos variantes fundamentales para modelar los sistemas: modelo del negocio y modelo del dominio. El modelo del negocio está dirigido al funcionamiento de los procesos de negocio de las organizaciones. Y los aspectos necesarios para su definición son:

- Ø Entender la estructura y la dinámica de la organización.
- Ø Entender los problemas actuales e identificar mejoras potenciales.
- Ø Asegurarse de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización.
- Ø Derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga.

Por su parte el modelo del dominio describe los conceptos importantes del contexto como objetos del dominio, y enlaza estos objetos unos con otros. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. La identificación y la asignación de un nombre para estos objetos ayudan a desarrollar un glosario de términos que permitirá comunicarse mejor a todos los que están trabajando en el sistema. Más adelante los objetos del dominio ayudarán a identificar algunas de las clases a medida que se analiza y diseña el sistema.

Teniendo en cuenta esta explicación y haber determinado que los procesos dentro de las áreas no son de gran envergadura se llega a la conclusión de que el negocio que se está estudiando, tiene un nivel de complejidad un poco bajo, por tanto se tratará de dar un enfoque nuevo al proceso.

Ver Anexo 2.

2.3 Definición de las entidades y conceptos principales

Las Correcciones y las Acciones Correctivas forman parte del vocabulario básico de todos los Sistemas de Gestión de Calidad. La función que cumplen estos dos elementos en los sistemas de gestión, es substancialmente diferente, sin embargo son habituales las confusiones, incluso entre asesores y auditores profesionales.

Corrección: Una Corrección es el conjunto de actividades realizadas para eliminar o subsanar lo que no ha salido bien (No Conformidad). Es decir es una Acción tomada para eliminar una No Conformidad.

Acción Correctiva: Una Acción Correctiva es un conjunto de actividades emprendidas para eliminar la causa de algo que no ha salido bien. En fin una Acción Correctiva es una Acción tomada para eliminar la causa de una No Conformidad detectada u otra situación indeseable.

Las Correcciones Atacan la No Conformidad, mientras que las Acciones Correctivas atacan la causa de la no Conformidad.

Una Corrección no evita que el problema se vuelva a reproducir, lo cual es el objetivo de la Acción Correctiva.

Acciones Preventivas: Acción para eliminar las causas potenciales de una no conformidad, un defecto u otra situación potencial indeseable con vista a prevenir su ocurrencia.

Producto no Conforme: Incumplimiento de un requisito especificado. La definición comprende las desviaciones o ausencia de unas o varias características de la calidad o elementos del sistema de calidad de los requisitos especificados.

No Conformidad: Falta de cumplimiento de los requisitos especificados.

Áreas: Como su nombre lo indica son áreas de regulación, control y apoyo de la empresa y unidades empresariales de base (UEB).

Sistema: Aplicación desarrollada con el objetivo de brindar servicios, apoyar la gestión de la información y que puede ser accedida por cualquier usuario en dependencia de los privilegios que le hayan sido otorgado.

Usuarios: Todo aquel que pueda acceder al sistema, para realizar diversas funciones con el mismo, en dependencia de los privilegios que le hayan sido otorgado.

2.4 Requerimientos Funcionales y no Funcionales del Sistema

2.4.1 Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, expresando más detalladamente las responsabilidades del sistema. Con ellos se pretende determinar de manera clara y concisa lo que debe hacer el sistema siguiendo un enfoque funcional.

A continuación se listan los requerimientos funcionales:

Paquete No Conformidad

1. Mostrar el reporte de una No Conformidad.
2. Gestionar No Conformidad.
 - a). Insertar Nota de Observación.
 - b). Insertar Nota de no Conformidad.
 - c). Insertar Producto No Conforme.
3. Imprimir el Reporte de una No Conformidad.
4. Notificar por correo electrónico a la persona que le ha sido levantada la No Conformidad de un área determinada.

Paquete Análisis

5. Insertar Análisis de Causa-Raíz.
6. Insertar Correcciones.
7. Insertar Acciones Correctivas.
8. Insertar Acciones Preventivas.

Paquete Gestionar Seguimiento

9. Insertar verificación del cumplimiento de las Acciones.
10. Insertar verificación del cumplimiento de las Correcciones.
11. Cerrar la No Conformidad.
12. Eliminar una No Conformidad.
13. Modificar
 - a.) Descripción de la No conformidad
 - b.) Descripción de la Corrección
 - c.) Descripción de la Acción
 - d.) Análisis de Causa_Raíz
 - e.) Fecha de Ejecución de la Corrección
 - f.) Fecha de Ejecución de una Acción

Paquete Mostrar reporte de Acciones

14. Mostrar Reportes de Acciones Preventivas.
15. Mostrar Reportes de Acciones Correctivas.

2.4.2 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el sistema a desarrollar debe tener. Definiendo propiedades como características que hacen al producto atractivo, usable, rápido y confiable. Los requerimientos no funcionales son fundamentales en el éxito del producto.

Apariencia o interfaz externa

El sistema debe tener una interfaz sencilla, agradable, legible y de fácil uso para el usuario. El contenido será mostrado de manera comprensible y fácil de leer.

Usabilidad

Se espera que la usabilidad de este sistema sea bastante elevada, es decir, que cuente con un alto nivel de aceptación para los usuarios, debido a que constituye una forma más flexible y familiarizada de mostrar las informaciones, que en la actualidad resultan limitados a su acceso. Por su confección puede ser utilizado

por cualquier tipo de usuario en dependencia de los privilegios que le hayan sido otorgados, por lo que esto no constituye una limitación para la utilización del mismo.

Soporte

Es necesario un servidor para la base de datos. Se requiere que la base de datos sea configurable teniendo en cuenta el futuro crecimiento del sistema, al incorporársele los restantes contenidos y a su vez por nuevas opciones que se deseen incorporar.

Es importante señalar que el presente Trabajo constituye un subsistema de un Sistema de Gestión Empresarial que se encuentra funcionando en la Empresa donde, el mismo cuenta con un módulo de administración que permite realizar todo lo referente a la gestión de usuarios, de grupos, permisos, accesos, módulos, lo que facilita el trabajo, puesto que no existe la necesidad de preocuparse de la seguridad de los datos que se manipulen en el sistema.

Rendimiento

El sistema deberá ser rápido ante las solicitudes de los usuarios y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos en el cliente y en el servidor aquellas que por cuestiones de seguridad o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento y un mayor aprovechamiento de los recursos.

Portabilidad

El sistema tendrá una buena portabilidad debido a que se ejecutará sobre diferentes sistemas operativos constituyendo un sistema multiplataforma.

Seguridad

Garantizar que la información sea gestionada solo por los usuarios con derechos sobre la misma.

- Ø Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Ø Verificación sobre acciones irreversibles (eliminaciones).
- Ø Se debe identificar al usuario antes de tener derecho a realizar cualquier acción sobre el sistema.
- Ø Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

Software

El usuario utilizará Firefox Operativos Windows 98 o superior o Linux.

- Ø Se requiere de un Servidor Web Apache y un Servidor de Base de datos MySQL que podrán ser usados bajo los sistemas operativos Windows y Linux.

Hardware

El usuario debe de tener como mínimo una PC, con 128 Mb de RAM y 333 Mhz de velocidad.

- Ø Los Servidores estarán como mínimo en un Pentium IV o superior, con 512 MB de RAM, 2.5 Ghz de velocidad y 40 GB de disco duro como mínimo.

2.5 Conclusiones

En este capítulo se ha realizado una descripción sobre el funcionamiento del sistema, representado a través de un modelo de dominio y estableciendo los requerimientos funcionales y no funcionales los cuales permitirán elaborar el diagrama de caso de uso para comprender mejor el problema que la aplicación pretende solucionar.

Capítulo 3: Análisis y Diseño

3.1 Introducción

En el Flujo de Trabajo de Análisis y Diseño, específicamente en el Diseño se modela y adquiere forma el sistema, logrando obtener una mayor comprensión de los aspectos relacionados con los requisitos funcionales, no funcionales y otras restricciones del sistema, contribuyendo a definir una arquitectura estable y sólida para la futura implementación del software. El propósito del capítulo está encaminado a adquirir una comprensión de los aspectos relacionados con los requerimientos, lenguajes de programación, componentes reutilizables y tecnologías de distribución. Se presentan diferentes artefactos modelados con la herramienta Rational Rose, tales como el Diagrama de Caso de Uso del Sistema, el Modelo de Diseño, especificándose la estructura y definición de los elementos que posee, Diagramas de clases, Descripción de las clases de diseño y Diagramas de secuencias, despliegue y componentes. Se diseña la base de datos, obteniéndose el Diagrama de Clases Persistentes y el Modelo de Datos.

3.2 Actores del sistema a automatizar

Tabla 3.1 Definición de actores del sistema a automatizar.

Actores	Descripción
Especialistas en Gestión de la Calidad	<ul style="list-style-type: none"> ü Se encargan de designar un grupo de especialistas para realizar las revisiones del Sistema de Gestión de la Calidad por la Dirección de la Empresa con el objetivo de prevenir malos funcionamientos en el mismo.
Auditores	<ul style="list-style-type: none"> ü Una vez hechas las auditorías

	<p>y detectadas las deficiencias, se encargan de levantar las No Conformidades.</p> <ul style="list-style-type: none"> ü Dan seguimiento y verificación de las acciones que han levantado en diferentes áreas detectadas por ellos. ü Se encargan de cerrar las No Conformidades en caso de que se halla erradicado el problema.
Especialistas de Área	<ul style="list-style-type: none"> ü Realizan el Análisis de Causa_Raíz de cada una de las No Conformidades. ü Toman Acciones o Correcciones para así erradicar el problema. ü Generalmente reciben las No Conformidades debido a que son los responsables del área en que trabajan, por lo tanto se encargan de realizar el Análisis de Causa_Raíz de cada una de las No Conformidades. ü Realizan las Inspecciones de Procesos en sus respectivas áreas con el objetivo de detectar deficiencias reales o potenciales que los afecten en un futuro.

3.3 Paquetes y sus relaciones

Cuando un sistema contiene muchos casos de uso, es necesario un mecanismo para agruparlo y así facilitar el uso, mantenimiento y reusabilidad. Esto se logra a través de los paquetes, que son utilizados para organizar los elementos de modelado en partes mayores que se pueden manipular como un grupo. Los paquetes también ayudan a organizar los elementos en los modelos con el fin de comprenderlos más fácilmente.

UML representa los paquetes como carpetas. En el sistema se definieron los paquetes No Conformidad, Análisis, Gestionar Seguimiento y Mostrar Reportes de Acciones.

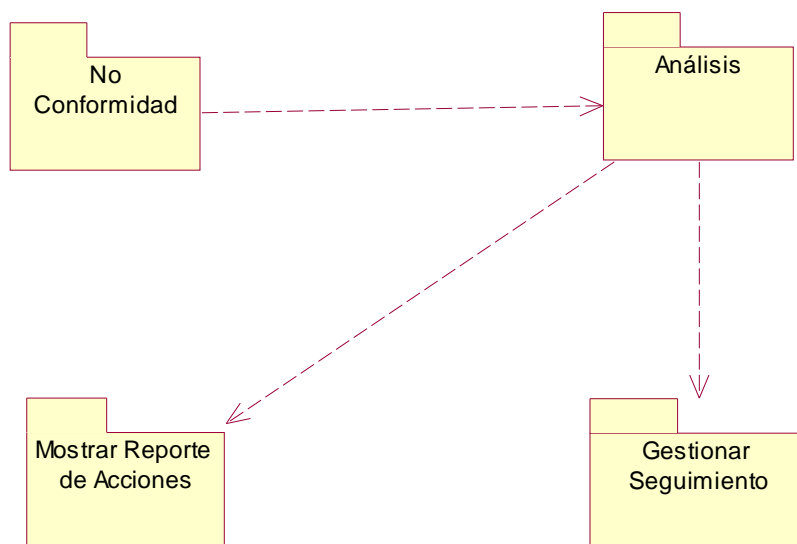


Figura 3.1. Diagrama de Paquetes

3.4 Diagramas de casos de Usos del sistema a automatizar

Utilizando las facilidades que nos brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de caso de uso. Los casos de usos modelan un diálogo entre un actor y el sistema. La colección de casos de usos para un sistema constituye todas las formas definidas en que el sistema será usado.

Un caso de uso representa una funcionalidad completa tal y como es percibida por un actor. La definición formal de un caso de uso en UML es: un caso de uso

es una secuencia de acciones realizadas por el sistema que proporciona un resultado observable de valor para un actor en particular.

Ver Anexo 3.

3.4.1 Paquete No Conformidad

La figura 3.2 muestra el Diagrama de casos de uso del Paquete Gestionar No Conformidad.

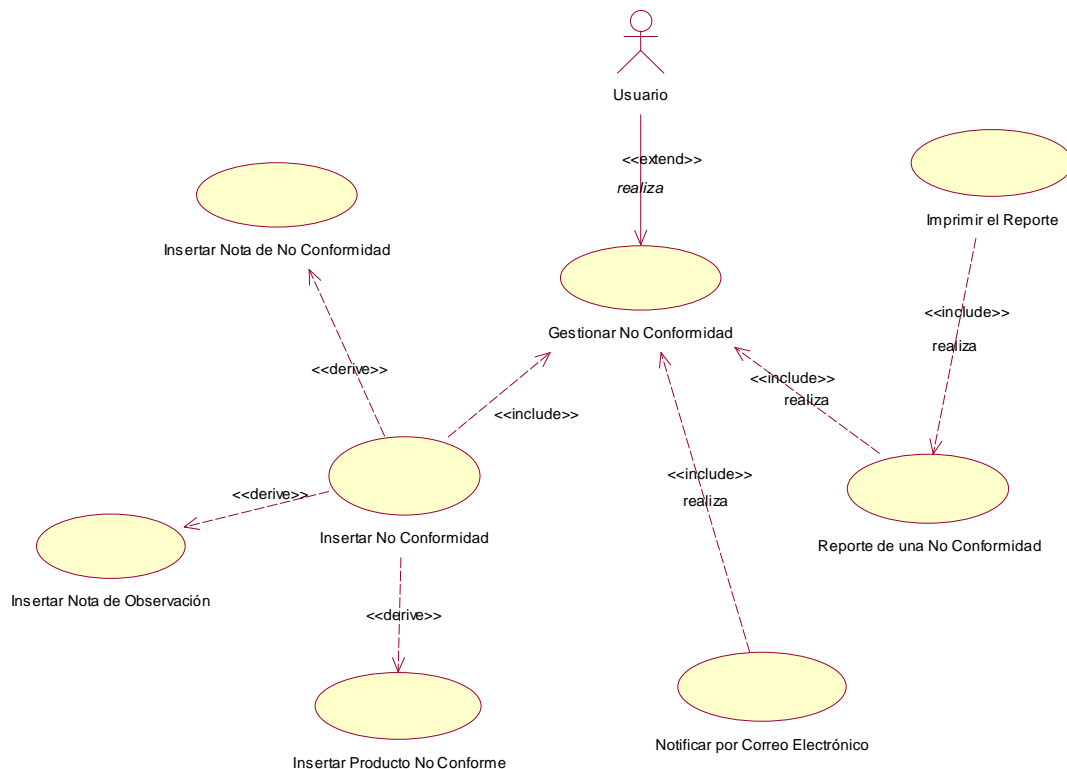


Figura 3.2 Diagrama de Casos de Uso del Paquete No Conformidad.

3.5 Descripción del Caso de Uso Gestionar No Conformidad

Tabla 3.2. Descripción del CU < Gestionar No Conformidad >

Nombre del caso de Uso	Gestionar No Conformidad
Actores	Todos(Inician)
Resumen	El Caso de Uso incluye los casos de usos: Insertar Nota de Observación, Insertar Nota de No Conformidad,

	<p>Insertar Producto No Conforme, Reporte de una No Conformidad, Imprimir Reporte, Notificar por Correo Electrónico.</p> <p>El Caso de Uso inicia cuando los Usuarios (Especialistas de Áreas, Especialista en Gestión de la Calidad o Auditores) a través de las inspecciones de procesos, revisiones del sistema de gestión de la calidad por la dirección o auditorías realizadas a las diferentes áreas respectivamente, detectan deficiencias y proceden a levantar las No Conformidades.</p>
Referencias	<i>R1,R2 a), b), c),R3,R4</i>
Precondiciones	Debe ser detectada al menos una deficiencia.
Poscondiciones	Debe ser notificada por Correo electrónico la No Conformidad a la persona que la recibe.
Requisitos especiales	

Para más detalles *ver Anexo 1 y 4.*

3.6 Diagrama de clases del diseño

Los diagramas de clases del diseño resumen la definición de las clases (e interfaces) que se implementan en el software. En los diagramas siguientes se muestra la interacción entre las clases del diseño definidas para cada caso de uso en particular.

A continuación se muestra la tabla 3.3 con la terminología que se usará en los diagramas para un mayor entendimiento de los mismos.

Tabla 3.3 Terminología de los diagramas de clases de diseño.

Clases	Descripción
sv	Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.
cl	Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script, que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.
form	Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. Estos se comunican con las páginas servidores mediante submit.
<<build>>	Significa que una página servidor construye a una página cliente.
<<submit>>	Significa que los formularios envían sus datos al código servidor para ser

	procesados los pedidos.
<<link>>	Significa una relación común entre las páginas Web que es el hipervínculo.

3.6.1 Diagrama de clase Web Insertar Producto.

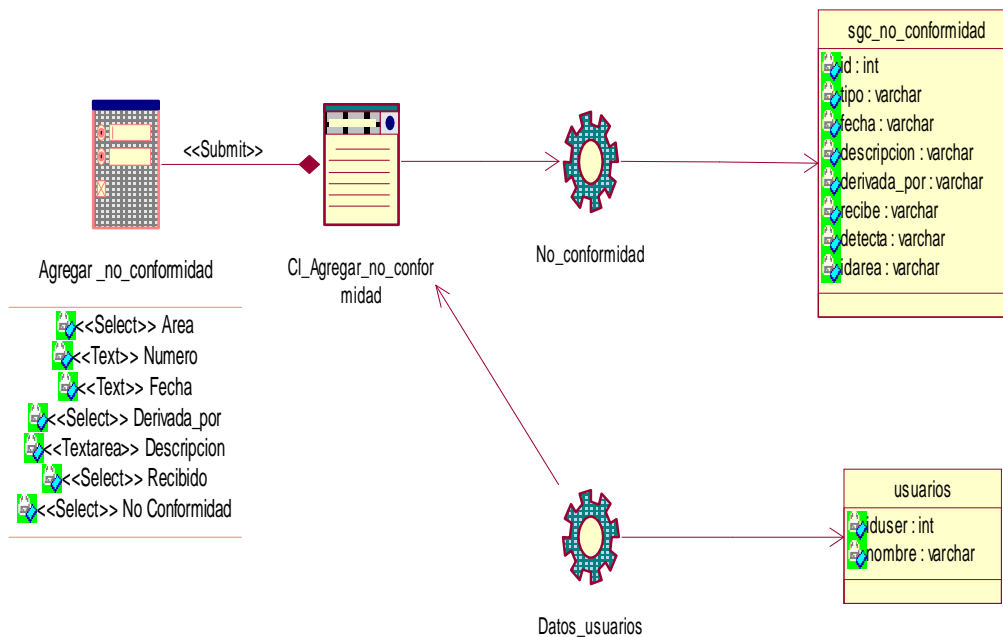


Figura 3.3 Diagrama de Clase Web Insertar Producto.
Ver anexo 6

3.7 Mapa de Navegación

Un Mapa de navegación define la estructura jerárquica de páginas lógicas de la aplicación y los niveles de los usuarios en la navegación. Es una forma de representar la navegación que se realiza por todo el sitio.

Ver Anexo 5

3.8 Principios de diseño

El diseño de las interfaces de la aplicación es un elemento de vital importancia en el desarrollo de un sistema, pues la calidad en este aspecto puede ser un elemento determinante en el éxito o fracaso del sistema.

Las interfaces diseñadas cumplen con el patrón de diseño establecido por la Intranet y las aplicaciones Web desarrolladas en Empresa Empleadora del Níquel (EMPLENI). Son legibles, con una combinación de colores agradables y poco llamativos de forma que el usuario no pierda concentración mientras interactúa con la aplicación.

La aplicación consta de una página principal la cual tiene un menú desde el que se puede acceder a todas las informaciones.

El menú está estructurado para la fácil comprensión del usuario y brinda acceso rápido a todas las opciones. Los mensajes que se muestran, ya sean de error o informativos, son breves y concisos.

3.9 Interfaz de usuario

La página principal de la aplicación es común para todos los usuarios, una vez que hayan accedido tendrán acceso solo a los menús a los cuales se les haya dado permiso. El sistema le brinda la posibilidad al administrador de darles permisos a los usuarios en dependencia de sus necesidades y obligaciones. Los menús se encuentran en la cabecera de todas las páginas del sistema.

3.9.1 Tratamiento de errores

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano. Por último, se utilizan errores en forma de mensajes de texto en la misma página de presentación donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar.

3.10 Diseño de la Base de Datos

Para diseñar la base de datos del sistema, se utilizó el diagrama de clases persistentes. Algunas de las clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son los que pueden

modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos.

Ver Anexo 7.

Tabla 3.4 Descripción textual de las clases persistentes

Nombre de la clase	sgc_no_conformidad
Propósito	Eliminar las deficiencias.
Atributos	Descripción de Atributos
id	Identificador
tipo	El tipo de no conformidad existente que puede ser un producto.
fecha	Fecha.
descripcion	Una breve descripción acerca de la no conformidad.
derivada _ por	Evento del que ha sido derivada la no conformidad
Recibe	Usuario que recibe la no conformidad.
Detecta	Usuario detecta la no conformidad.
Diarrea	Identificador del área.

Ver Anexo 8

3.10.1 Modelo Físico de Datos

Describe la representación física y lógica de los datos constantes utilizados por la aplicación. Usualmente describe los diferentes componentes de la estructura de una base de datos relacional. Se usa para definir la relación entre las clases del diseño y las estructuras de datos.

Ver Anexo 9.

3.11 Diagrama de Secuencia

Con la idea de dar una visión gráfica de las interacciones de los actores con el sistema, se utilizan los diagramas de secuencia (DSS), los cuales muestran qué hace el sistema ante el medio, sin explicar el cómo.

Forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto. Es útil para observar la vida de los objetos, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de invocaciones entre los componentes.

Muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

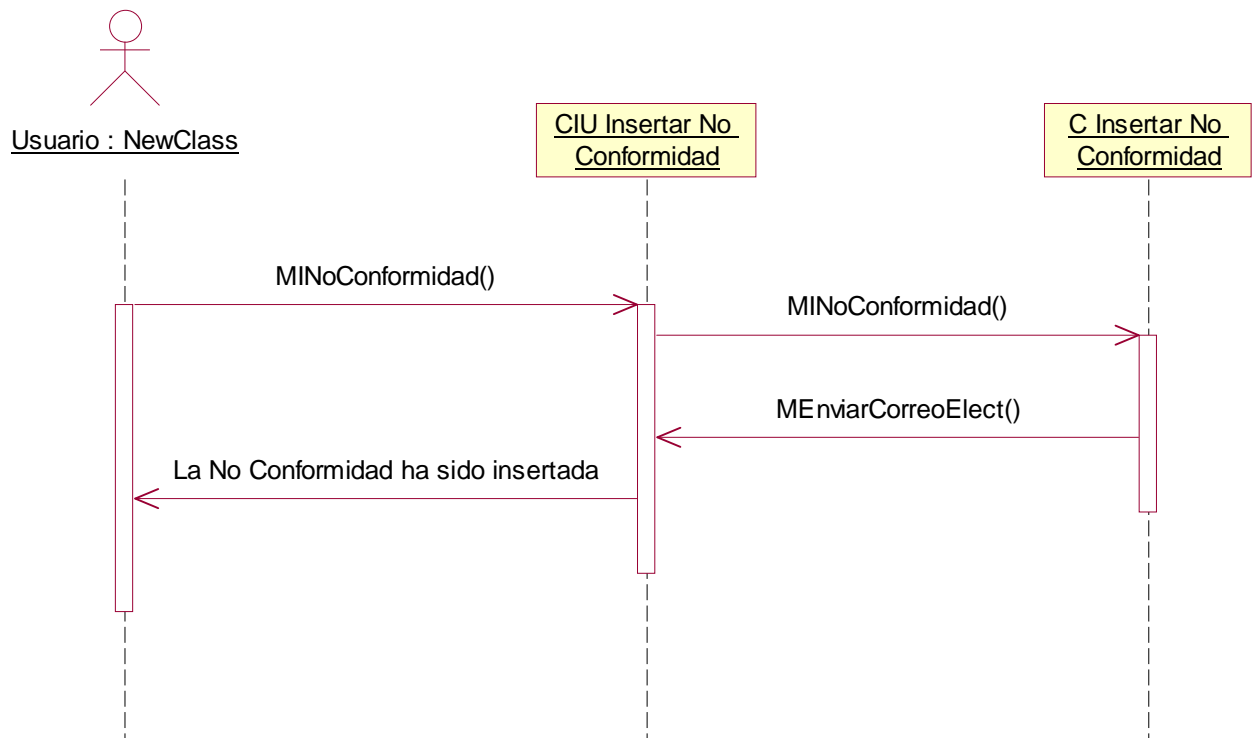


Figura 3.4 Diagrama de Secuencia Agregar No Conformidad.

Ver Anexo 10

3.12 Diagrama de despliegue

El diagrama de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos. Los nodos representan

recursos de cómputos: procesadores o dispositivos de hardware. La distribución de la aplicación quedará de la siguiente manera:

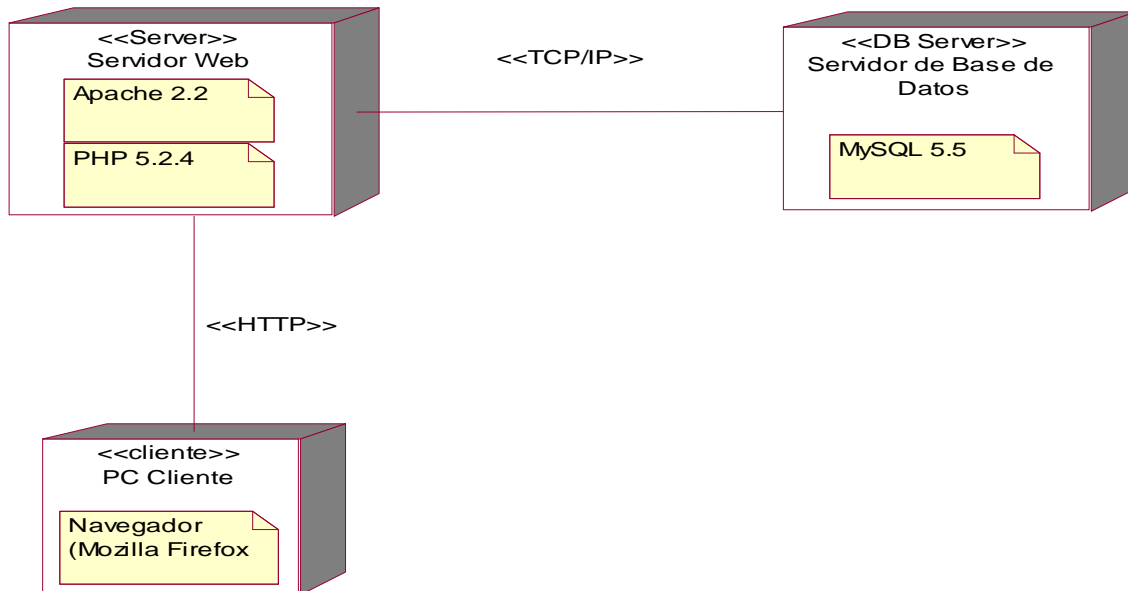


Figura 3.5 Diagrama de despliegue

En el Servidor de Aplicación se encuentra montado el Sistema, específicamente la Capa de Presentación y la Capa de Negocio, junto con el servidor Web Apache que posibilita su publicación. En el Servidor de Base de Datos, se encuentra la Capa de Datos de la aplicación, en la cual se almacena la información que se gestiona a través del sistema. En el nodo que representa la PC_Usuario, se encuentra el navegador Web, a través del cual se puede ejecutar la aplicación.

3.13 Diagrama de Componentes

Con el objetivo de lograr una mejor visión de los componentes que forman el sistema, se presentan los Diagramas de Componentes que describen los elementos del sistema y sus relaciones.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el Modelo del Diseño. Los componentes pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente.

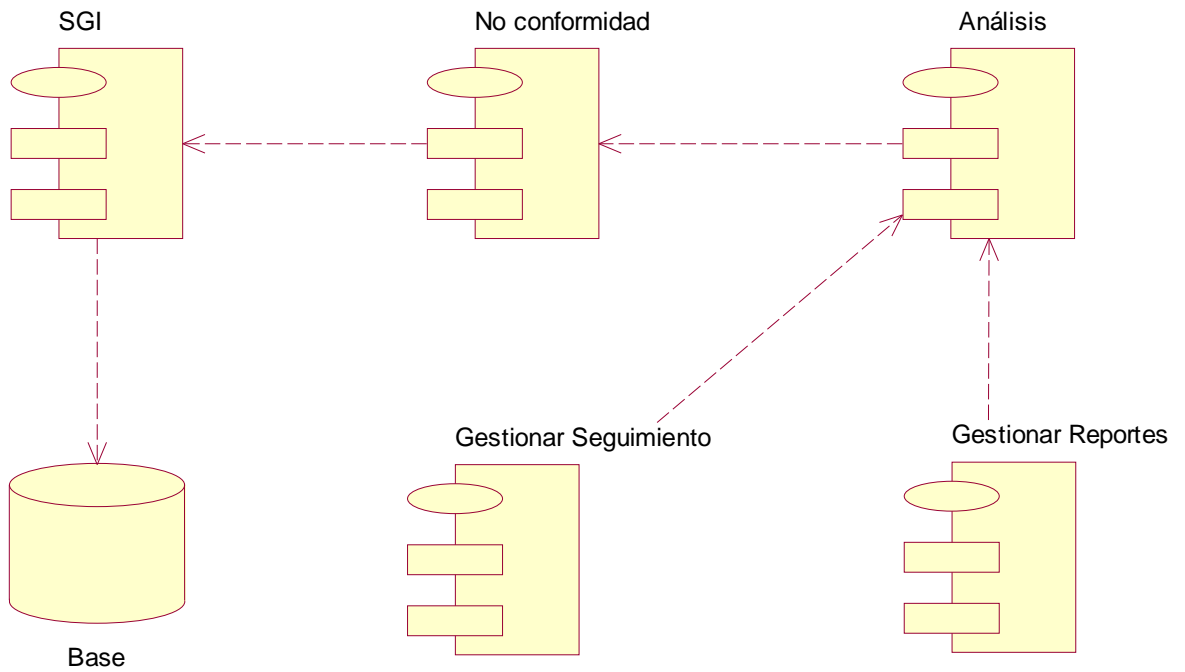


Figura 3.6 Diagrama de Componentes de SGI

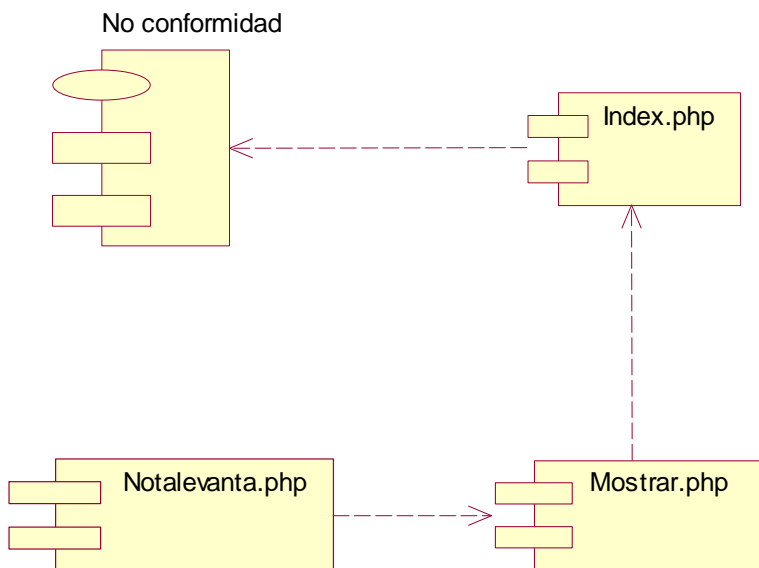


Figura 3.7 Diagrama de Componentes de no conformidad

Ver Anexo 11

3.14 Conclusiones

Se finaliza la etapa de diseño del sistema pudiéndose obtener un grupo numeroso de diagramas que reflejan la estructura del sistema y sin ellos resulta difícil obtener resultados satisfactorios en el desarrollo de la aplicación.

El diseño de la base de datos es de vital importancia en cualquier herramienta y en este capítulo que concluye se analizan los diagramas de clases persistentes y el modelo de datos. Esto permite conocer los campos de las tablas y de qué forma se establece las relaciones entre estas.

De modo general, se puede decir que en este capítulo se modelan los diagramas más importantes y los que más detallan las partes de la aplicación y la forma en que están distribuidas en esta.

Capítulo 4: Estudio de Factibilidad

4.1 Introducción

Para lograr con eficiencia y eficacia la implementación de un proyecto, es de suma importancia para todos los implicados en el mismo, efectuar y evaluar la factibilidad antes de su elaboración, para poder lograr definir con acierto si es conveniente llevar a cabo dicho proyecto. Por lo que el presente capítulo de la investigación, está dedicado a realizar el estudio de la factibilidad, los beneficios y coste del sistema propuesto.

4.2 Planificación por puntos de función

- ✓ Planificación por puntos de función
- ✓ Pasos para calcular usando COCOMO II
- ✓ Obtener los puntos de función (UFP)
- ✓ Identificación de las características
- ✓ Clasificación
- ✓ Ponderación aplicando pesos
- ✓ Estimar la cantidad de instrucciones fuentes (SLOC)
- ✓ Utilización de tabla de lenguaje
- ✓ Aplicar las fórmulas
- ✓ Obtener Esfuerzo
- ✓ Obtener Tiempo
- ✓ Costo del proyecto.
- ✓ Obtención de los puntos de función (UFP)
- ✓ Identificar las características
- ✓ Entradas Externas (EI)
- ✓ Salidas Externas (EO)
- ✓ Ficheros Lógicos Internos (ILF)
- ✓ Consultas (Peticiones) Externas (EQ)

Vamos a apoyarnos de los Requerimientos Funcionales del Sistema, que son los datos más cercanos que van a manipular en el sistema.

Entradas Externas (EI):

Entradas externas (EI): hacen referencia a los tratamientos que procesan datos o información de control introducidos en la aplicación desde fuera de sus límites. Son las entradas a la aplicación.

Tabla 4.1 Entradas externas (EI)

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación (Alta, media Baja)
Insertar Nota de No Conformidad	2	9	Medio
Insertar Nota de Observación	2	9	Medio
Insertar Producto no Conforme	2	6	Medio
Modificar Descripción de la No Conformidad	3	3	Bajo
Modificar Descripción de la Corrección	3	3	Bajo
Modificar Descripción de la de la Acción	3	4	Bajo
Modificar Análisis de Causa_Raíz	3	4	Bajo
Modificar Fecha de ejecución de la Corrección	3	2	Bajo
Modificar Fecha de ejecución de la Acción	3	2	Bajo
Cantidad de ficheros			
Bajo	Medio	Alta	
6	3	0	

Salidas Externas (EO)

Salidas externas (EO): cualquier proceso elemental que genere datos o información de control que salga de los límites de la aplicación. Equivale a las salidas que genera la aplicación.

Tabla 4.2 Salidas externas (EO)

Nombre de la salida externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación (Alta, media Baja)
Mostrar el reporte de una No Conformidad.	2	3	Bajo
Mostrar Reportes de Acciones Preventivas.	3	6	Medio
Mostrar Reportes de Acciones Correctivas.	3	6	Medio
Imprimir el Reporte de una No Conformidad.	2	4	Bajo
Cantidad de ficheros			
Baja	Medio	Alta	
2	2	0	

Consultas externas (EQ)

Consultas externas (EQ): Todo aquel proceso que esté formado por una combinación Entrada/Salida (E/O) produciendo una consulta a los datos.

El flujo de datos deberá de tener dos direcciones. Como consecuencia de una consulta no se modifican los datos del sistema. La complejidad de las consultas viene dada por la mayor entre la entrada y la salida.

Tabla 4.3 Consultas externas (EQ)

Nombre de las Consultas Externas	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación (Alta, media Baja)
Eliminar una no Conformidad	1	9	Bajo
Cerrar una no Conformidad	3	5	Bajo
Cantidad de ficheros			
Baja	Medio	Alta	
2	0	0	

Ficheros lógicos internos (ILF):

Ficheros lógicos internos (ILF): grupo de datos interrelacionados lógicamente y que pueden ser identificados claramente y siempre se mantienen dentro de los límites de la aplicación, son internos y propios de la aplicación. Son los Archivos de la aplicación.

Tabla 4.4 Ficheros internos (ILF)

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Bajo, Media y Alto)
areas	1+	2	Baja
sgc_no_conformidad	1+	8	Baja
sgc_verificar	1+	7	Baja
usuarios	1+	2	Baja
sgc_notas	1+	5	Baja
sgc_causa	1+	4	Baja
sgc_correccion	1+	6	Baja
sgc_productos	1+	4	Baja
sgc_acciones	1+	8	Baja
Cantidad de records			
Bajo	Medio	Alto	
9	0	0	

Puntos de Función desajustados:

Tabla 4.5 Puntos de Función desajustados

Elementos	Bajas		Medios		Altas		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	9	7	0	10	0	15	63
Entradas externas	6	3	3	4	0	6	30
Salidas externas	2	4	2	5	0	7	18
Consultas externas	2	3	0	4	0	6	6
Total	19		5		0		117

4.2.1 Características del proyecto

Cálculo de la cantidad de instrucciones fuentes.

Para el cálculo de las instrucciones fuentes (SLOC) se utilizó la fórmula siguiente:

$$\text{SLOC} = \text{UFP} * \text{ratio}$$

$$\text{SLOC} = 117 * 69$$

$$\text{SLOC} = 8073$$

$$\text{KSLOC} = 8,073 \text{ (Miles de líneas de código)}$$

Donde UFP es el total de puntos de función desajustados, y ratio es una constante para las SLOC de cada lenguaje de programación en este caso tiene un valor para PHP de 69.

Factores de escala:

Tabla 4.6 Factores de escala

Factor de escala	Valor	Justificación
PREC (Precedencia)	3.72	Posee aspectos novedosos para el desarrollador.
FLEX (Flexibilidad)	2.03	Existió cierto acuerdo de forma general en cuanto a las interfaces de diseño y los requisitos del software.
RESL (Riesgos)	2.83	La herramienta tiene altas posibilidades de hacer frente a los riesgos.
TEAM (Cohesión del equipo)	0.00	La herramienta tiene altas posibilidades de hacer frente a los riesgos.
PMAT(Madurez del proceso)	4.68	Posee un nivel de madurez medio.

Multiplicadores de Esfuerzo:

Tabla 4.7 Multiplicadores de Esfuerzo

Multiplicador(+)	Valor	Justificación
RCPX (Fiabilidad y complejidad del producto)	1.00	El grado de confiabilidad es nominal, la complejidad está en el nivel de moderada. La base de datos es pequeña y la confianza entre documento y desarrollo del sistema es nominal. A partir de estos resultados el indicador concluye en nominal.
RUSE (Requerimientos de reusabilidad)	1.07	En la implementación del sistema existe una alta reusabilidad de códigos.
PDIF (Dificultad de la plataforma)	0.87	El sistema se considera estable y las restricciones de almacenamiento y tiempo son de un 50% por lo que el valor general es nominal.
PERS (Capacidad del personal)	0.83	Los desarrolladores tienen gran

		conocimiento en la programación del sistema.
PREX (Experiencia del personal)	1.00	Basta experiencia en cuanto al lenguaje, se conoce el tipo de software y herramientas para el desarrollo de aplicaciones de este tipo. Por tanto se valora como nominal.
FCIL (Facilidades para el desarrollo)	0.73	Se emplearon herramientas modernas de programación. Así como para la documentación se utilizó la notación UML y para su modelado visual se empleó la herramienta Rational Rose.
SCED (Esfuerzo del calendario)	1.00	Se estima que el producto se hará en el tiempo establecido.

Tabla 4.8 Cantidad de Instrucciones Fuentes

Características	
Puntos de función desajustados	117
Ratio (PHP)	69
SLOC = UFP * Ratio	8073
KSLOC	8.073

Esfuerzo de Desarrollo:

Tabla 4.9 Esfuerzo de desarrollo

Cálculo de:	Justificación
Esfuerzo de Desarrollo (PM)	<p>El esfuerzo se representa mediante la fórmula de Bohem y se expresa en hombre mes:</p> $E = B + 0.01 * \Sigma SF$ $= 0.91 + 0.01 * 13.26$ $= 1.0426$ $PM = A * (Size)^E * \pi EMI$ $= 2,94 * (8.073)^{1.0426} * 0,56640323$ $PM = 14.6942591$ $PM \approx 14.69 \text{ Hombre/Mes}$ <p>(Aproximadamente se necesitan de 15 personas para realizar el software en un mes).</p>

Tiempo de desarrollo

Tabla 4.10 Tiempo de desarrollo

Cálculo de:	Justificación
<p>Tiempo de desarrollo (TDEV)</p>	<p>El tiempo de desarrollo en meses viene dado por la fórmula:</p> <p>TDEV = C * (PM)^F (meses)</p> <p>donde:</p> <p>C = 3.67</p> <p>D = 0.28</p> <p>B = 0.91</p> <p>F = D + 0.2 * (E - B)</p> <p>= 0.28 + 0.2 * (1.0426 - 0.91)</p> <p>= 0.30652</p> <p>TDEV = C * (PM)^F</p> <p>= 3,67 * (14.69)^0.30652</p> <p>= 8.36337529</p> <p>TDEV ≈ 8 meses</p> <p>(Aproximadamente 8 meses)</p>

Cantidad de hombres:

Tabla 4.11 Cantidad de hombres

Cálculo de:	Justificación
<p>Cantidad de hombres (CH)</p>	<p>La cantidad de hombres es el resultado de la división del Esfuerzo entre el Tiempo de Desarrollo.</p> <p>CH = PM/TDEV</p> <p>= 14.69/8.3</p> <p>= 1.76987952</p> <p>CH ≈ 2 Hombres</p>

	<p>Los valores obtenidos indican que el proyecto necesitaría 2 hombres para su realización en aproximadamente 8 meses.</p> <p>No obstante como la cantidad de desarrolladores es 1, el tiempo de duración real es de 15 meses.</p> <p>TDEV_{real} = PM / CDes Real</p> <p>= 14.69 / 1</p> <p>= 14.69 meses</p> <p>TDEV_{real} ≈ 15 meses</p>
--	--

Costo:

Tabla 4.12 Costo

Cálculo de:	Justificación
Costo (C)	<p>CHM: Costo por hombres mes.</p> <p>SP: Salario promedio.</p> <p>C = CHM * PM</p> <p>Donde:</p> <p>CHM = 1 * SP</p> <p>CHM = 1 * 365 = \$ 365</p> <p>C = 365 * 14.69</p> <p>C = \$ 5361.85</p> <p>Dando como resultado \$5361.85 (CUP)</p>

Podemos resumir que:

Tabla 4.13 Resultados de las estimaciones de esfuerzo, tiempo de desarrollo, cantidad de hombres y costo del proyecto

Cálculo de:	Valor	Justificación
Esfuerzo	14.69 hombres-mes	Cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto.
Tiempo de desarrollo	8 meses	Cantidad de meses para terminar el proyecto.
Cantidad de personas	2 Hombres	Cantidad de personas necesarias para terminar el proyecto en 9 meses.
Costo	\$5361.85	Cantidad de dinero que cuesta el proyecto después de terminado.
Salario medio	\$ 365.00	Salario básico de un ingeniero.

4.3 Beneficios tangibles e intangibles

Entre los principales beneficios que se logran con el Sistema Automatizado para el control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel podemos decir que el mismo:

- √ Protege los recursos de la organización, buscando su adecuada administración ante riesgos potenciales y reales que los pueden afectar.

- ✓ Garantiza la eficacia, eficiencia y economía en todas las operaciones de la organización, promoviendo y facilitando la correcta ejecución de las funciones y actividades establecidas.
- ✓ Garantiza que todas las actividades y recursos de la organización estén dirigidos al cumplimiento de los objetivos previstos.
- ✓ Asegura que todas las acciones institucionales de la empresa se desarrollen en el marco de las normas constitucionales, legales y reglamentarias.

Con él se logra una aplicación Web que anteriormente no estaba elaborada y a la vez un sistema de gestión de la información para perfeccionar la gestión que realizan las áreas y los trabajadores en ella. Le brinda al sistema administrativo de la entidad una mayor confiabilidad en la información y en sus operaciones, por los materiales y documentación que ofrece este sistema. La realización de este proyecto resulta muy útil a la empresa, ya que constituirá una herramienta de trabajo diario, brindando facilidades como ahorro de tiempo y recursos.

Beneficiará a los usuarios, para los que fue confeccionado el sistema, pues los contenidos que se exponen se harán más visibles y comprensibles, habrá una contribución al uso de las nuevas tecnologías.

4.4 Análisis de costos y beneficios

La puesta en funcionamiento de este sistema representa un aporte al desarrollo del plan de informatización de la sociedad y el empleo eficiente de las TIC, lo que contribuye con la batalla de ideas que se lleva a cabo en nuestro país.

Es por eso que partiendo de los costos obtenidos que demuestran que la empresa posee los recursos necesarios para llevar a cabo el desarrollo del sistema y en concordancia con los beneficios tangibles e intangibles el sistema se considera factible.

4.5 Conclusiones

El estudio de factibilidad constituye, sin lugar a dudas, una herramienta de gran importancia que permite estimar los costos. Mediante el análisis de la factibilidad del proyecto se pueden determinar los indicadores principales para conocer con exactitud los beneficios reales que aporta el sistema a la empresa o entidad donde se desarrolla este. El Sistema automatizado de control de acciones correctivas y preventivas de la Empresa Empleadora del Níquel después de haber transitado por las fases del cálculo resultó ser factible, sobre todo después del análisis entre los costos y los beneficios. Estos beneficios quedaron plasmados y clasificados en tangibles e intangibles y permitieron determinar los valores de costos, tiempo de desarrollo y recursos requeridos.

Conclusiones Generales.

El problema de la Gestión de la información, preocupa a todos hoy en día, puesto que cada vez los volúmenes son más grandes y se requieren formas más eficientes y seguras para guardarla y manejarla. La Empleni es un ejemplo claro de lo que está ocurriendo en este sentido. Es por ello que surge este proyecto en aras de resolver en parte esta situación.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos, para esto se siguieron los siguientes pasos:

- Ø Se realizó un estudio de los servicios que se prestan en la UEB para tener un mejor conocimiento, además de investigar cómo se guarda el producto final.
- Ø Se desarrolló un estudio de las diferentes tecnologías y herramientas para la confección del sistema, realizando una selección de acuerdo con las especificaciones del cliente.
- Ø Se elaboró un modelo de dominio y levantamiento de requisitos.
- Ø Se efectuó el análisis, diseño e implementación con lo que se resolvió las anteriores dificultades:
 1. El cliente ya cuenta con un sistema capaz de guardar grandes volúmenes de información de forma segura, eliminando los inconvenientes de que no se pierdan o deterioren ni que se acumulen de forma excesiva.
 2. La empresa ya cuenta con la disponibilidad de todos los servicios prestados, con mayor seguridad y confiabilidad.

Por todo lo anterior se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos satisfactoriamente.

Recomendaciones.

A modo general los objetivos trazados al inicio de este trabajo han sido logrados, pero al mismo tiempo, a lo largo del proceso de desarrollo, ha quedado claro que la propuesta es sólo la primera fase de un proyecto que puede ser mucho más ambicioso. Por tanto se hacen las siguientes recomendaciones:

- Ø Continuar trabajando en el sistema con el objetivo de mejorar los reportes de forma tal que sean más detallados y específicos.
- Ø Implementar nuevas funcionalidades de Sistema de gestión de la Calidad con aras de aumentar la potencialidad del mismo.

Referencias Bibliográficas:

1. *Getting Started with UML*. January 02, 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.omg.org/UML/>.
2. Sanchez, M.A.M., et al., *Metodologías De Desarrollo De Software*. 2005: Perú S.A.C.
3. *Open Workbench*. 2007 [cited; Available from: http://www.puntoorg.org/open_workbench_gesti_n_de_proyectos_para_win_dows .
4. *SGBD*. [cited 2008 23 de Septiembre]; Available from: <http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema5/tema5.htm>.
5. *Microsoft SQL Server*. [cited 2008 23 de Septiembre]; Available from: http://es.wikipedia.org/wiki/Microsoft_SQL_Server.
6. *Oracle* [cited 2008 23 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/Oracle>.
7. *MySQL* [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/MySQL>.
8. *HTML y XHTML*. [cited 2008 24 de Septiembre]; Available from: http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html.
9. *CSS*. [cited 2008 24 de Septiembre]; Available from: http://www.librosweb.es/css/capitulo1/que_es_css.html.
10. *Javascript*. [cited 2008 23 de Septiembre]; Available from: <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
11. *ASP*. [cited 2008 23 de Septiembre]; Available from: <http://www.maestrosdelweb.com/editorial/aspintro/>.
12. *PERL*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/541.php>.
13. *JSP*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/831.php>.

14. *Python*. [cited 2008 23 de Septiembre]; Available from: <http://www.desarrolloweb.com/articulos/1325.php>.
15. *Internet Information Services* [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/IIS>.
16. *Zope*. [cited 2008 19 de Septiembre]; Available from: <http://es.wikipedia.org/wiki/Zope>.
17. [cited 2008 24 de Septiembre]; Available from: <http://www.phpizza.com/es/>.
18. Company, T.P. *Zend Studio 5 LAS SOLUCIONES MÁS COMPLETAS PARA EL DESARROLLO DE PHP*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.zend.com/en/products/studio/>.
19. Company, T.P. *Performance, Management, Integration, and Enterprise Scalability*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.zend.com/en/products/platform/>.
20. *Acerca de Dreamweaver*. [cited 2008 24 de Septiembre]; Available from: http://www.tci.cl/cursos/acerca_de.php.
21. Incorporated, A.S. *Información sobre la integración de las funciones de Studio 8*. 2007 [cited 2008 19 de Septiembre]; Available from: <http://www.adobe.com/es/products/studio/integration/>.
22. *Rational Rose Enterprise*. [cited 2008 24 de Septiembre]; Available from: <http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html>.
23. *XAMPP*. [cited 2008 5 de Octubre]; Available from: <http://es.wikipedia.org/wiki/XAMPP>.
24. *XAMMP*. [cited 2008 5 de Octubre]; Available from: <http://www.genbeta.com/2005/09/27-xampp-servidor-web-correo-base-de-datos-y-mas-extras-en-pocos-minutos>.
25. *Embarcadero ER/Studio* [cited 2008 10 de Octubre]; Available from: http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755965.

26. *ER/Studio*. [cited 2008 10 de Octubre]; Available from: <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml>

Bibliografía

1. <http://softlibre.barrapunto.com/article.pl?sid=09/12/03/1218258&from=rss>
2. <http://kmkey-es.blogspot.com/2009/12/como-instalarse-kmkey-makhno.html>
3. <http://www.kmkey.com/>
4. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0034-75152006000300009
5. <http://www.portalcalidad.com>
6. <http://www.widman.biz> (ANALISIS CAUSA RAIZ, UNA HERRAMIENTA INVALUABLE PARA EL DIAGNOSTICO DE FALLAS --por Ing. Omar Linares, Presidente de la Asociación Boliviana de Ingeniería de Mantenimiento (ASBOMAN) de Santa Cruz.) (pdf).
7. <http://www.extrategus.com.ar> (Acciones en Búsqueda de la Excelencia. Compromiso Total con la Calidad) (pdf).
8. El Análisis de Causa Raíz, como herramienta en la mejora de la Confiabilidad. (pdf).
Autora: Carolina Altmann
9. ISO-DOCUMENT: Auditorías y Acciones Correctivas-Preventivas(ppt).
10. LAS HERRAMIENTAS DE LA CALIDAD, DIPLOMADO IV INFORMATICA MEDICA, DR. E. COSME SUAREZ ORTIZ.
11. <http://www.monografias.com>
12. <http://www.inghenia.com>
13. <http://hederaconsultores.blogspot.com>
14. <http://www.megaupload.com/?d=EP9DOO7Y>
15. Babel, un Sistema Automatizado de Gestión de Información para los servicios de traducción e interpretación. autor: Elda Jeny Báez Álvarez, Joviael Rodríguez Cepero y Boris Luis Orduñez Centro de Información de

- ETECSA. Calle. 3era, Centro de Negocios Miramar , entre 76 y 78, Playa.
Ciudad de La Habana.
16. Sistema Automatizado para la Gestión de la Seguridad Radiológica en un
Centro de Producción de Radio fármacos y Compuestos Marcados.
17. <http://www.hotfrog.es>
18. <http://www.wilsoft-la.com> Qaction
19. <http://www.marna.com.ve/PlanAcciones.htm> **ISO-Document Software.**
Control de no conformidades y Acciones correctivas y preventivas.
20. [http://www.portalcalidad.com/etiquetas/173-
Acciones_correctivas_y_preventivas](http://www.portalcalidad.com/etiquetas/173-Acciones_correctivas_y_preventivas)

Glosario de Términos

Listado de términos ordenados alfabéticamente, que son usados en el trabajo. Se indica para cada término su significado.

Apache: Servidor de páginas web de código abierto para diferentes plataformas (UNIX, Windows, etc.).

GPL (General Public License): Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

Open source: Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como free software.

PHP (PHP Hypertext Pre-processor): Lenguaje de programación para el desarrollo de Web dinámicas, con sintaxis parecida a la de C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

Corrección: Una Corrección es el conjunto de actividades realizadas para eliminar o subsanar lo que no ha salido bien (No Conformidad). Es decir es una Acción tomada para eliminar una No Conformidad.

Acción Correctiva: Una Acción Correctiva es un conjunto de actividades emprendidas para eliminar la causa de algo que no ha salido bien. En fin una Acción Correctiva es una Acción tomada para eliminar la causa de una No Conformidad detectada u otra situación indeseable.

Acciones Preventivas: Acción para eliminar las causas potenciales de una no conformidad, un defecto u otra situación potencial indeseable con vista a prevenir su ocurrencia.

Producto no Conforme: Incumplimiento de un requisito especificado. La definición comprende las desviaciones o ausencia de unas o varias características de la calidad o elementos del sistema de calidad de los requisitos especificados.

No Conformidad: Falta de cumplimiento de los requisitos especificados.

Áreas: Como su nombre lo indica son áreas de regulación, control y apoyo de la empresa y unidades empresariales de base (UEB).

Sistema: Aplicación desarrollada con el objetivo de brindar servicios, apoyar la gestión de la información y que puede ser accedida por cualquier usuario en dependencia de los privilegios que le hayan sido otorgado.

Usuarios: Todo aquel que pueda acceder al sistema, para realizar diversas funciones con el mismo, en dependencia de los privilegios que le hayan sido otorgado.

ANEXOS

ANEXO 1

Descripción de los Casos de Uso

Tabla 3.5 Descripción del CU <Mostrar Reporte de una No Conformidad >

Nombre del caso de Uso	Mostrar Reporte de una No Conformidad.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialistas de Área, Especialista en Gestión de la Calidad y Auditores) a través de las inspecciones de procesos, revisiones del sistema de gestión de la calidad por la dirección o auditorías realizadas a las diferentes áreas respectivamente se realiza el Análisis de Causa_Raíz, Fecha en que se realiza la misma.
Referencias	R1
Precondiciones	Debe haber sido gestionada al menos una No Conformidad.
Poscondiciones	Deben mostrar el reporte, los Usuarios (Especialista de área, Especialista en Gestión de la Calidad o Auditores) a las diferentes áreas donde fueron detectadas las deficiencias.
Requisitos especiales	

Tabla 3.6 Descripción del CU < Imprimir el Reporte >

Nombre del caso de Uso	Imprimir el Reporte.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores), generan un reporte de una no conformidad.
Referencias	R3
Precondiciones	Debe mostrar el Reporte de una No Conformidad.
Poscondiciones	
Requisitos especiales	Debe existir configurada al menos una impresora.

Tabla 3.7 Descripción del CU < Notificar por correo electrónico >

Nombre del caso de Uso	Notificar por correo electrónico.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) a través de las inspecciones de procesos, revisiones del sistema de gestión de la calidad por la dirección o auditorías realizadas, encuentran problemas y deficiencias en el área y proceden a levantar la No Conformidad, el sistema automáticamente notifica por correo electrónico a la persona que recibe la Nota de no Conformidad.

Referencias	<i>R4</i>
Precondiciones	Debe agregarse al menos una No Conformidad.
Poscondiciones	
Requisitos especiales	Debe estar configurado el puerto de salida y servidor de correo.

Tabla 3.8 Descripción del CU < Insertar Análisis de Causa_Raíz >

Nombre del caso de Uso	Insertar Análisis de Causa_Raíz
Actores	Especialista de Área (Inicia)
Resumen	El Caso de Uso inicia cuando el Usuario (Especialista de Área) a través de las inspecciones de procesos, realizan los diferentes análisis donde se realiza un estudio profundo de los motivos, causas y razones de las deficiencias encontradas en las diferentes áreas.
Referencias	<i>R5</i>
Precondiciones	Debe haber una gestión de Análisis
Poscondiciones	
Requisitos especiales	Sólo deben Insertar el análisis de Causa_Raíz el Especialista de Área.

Tabla 3.9 Descripción del CU < Insertar Correcciones >

Nombre del caso de Uso	Insertar Correcciones
Actores	Especialista de Área (Inicia)
Resumen	El Caso de Uso inicia cuando el Usuario (Especialista de Área) a través de las inspecciones de procesos, realizan los diferentes análisis luego insertan las Correcciones pertinentes.

Referencias	<i>R6</i>
Precondiciones	Debe haber una gestión de Análisis
Poscondiciones	
Requisitos especiales	Sólo deben insertar las Correcciones el Especialista de Área.

Tabla 3.10 Descripción del CU < Insertar Acciones Correctivas >

Nombre del caso de Uso	Insertar Acciones Correctivas
Actores	Especialista de Área (Inicia)
Resumen	El Caso de Uso inicia cuando el Usuario (Especialista de Área) inserta el análisis de causa raíz a las no conformidades reales levantadas, y luego insertan las Acciones Correctivas pertinentes.
Referencias	<i>R7</i>
Precondiciones	Se debe realizar el análisis de causa raíz.
Poscondiciones	Deben quedar definidas las acciones.
Requisitos especiales	

Tabla 3.11 Descripción del CU < Insertar Acciones Preventivas >

Nombre del caso de Uso	Insertar Acciones Preventivas
Actores	Especialista Área (Inicia)
Resumen	El Caso de Uso inicia cuando el Usuario (Especialista de Área) realiza el análisis de causa raíz a las no conformidades potenciales levantadas, y luego inserta las Acciones Preventivas pertinentes.
Referencias	<i>R8</i>
Precondiciones	Se debe realizar análisis de causa raíz.

Poscondiciones	Deben quedar definidas las acciones.
Requisitos especiales	

Tabla 3.12 Descripción del CU < Gestionar Seguimiento>

Nombre del caso de Uso	Gestionar Seguimiento.
Actores	Todos (Inician)
Resumen	<p>El Caso de Uso incluye los casos de uso: Insertar Verificación del cumplimiento de las de las acciones, Insertar Verificación del cumplimiento de las de las correcciones, Cerrar la no conformidad, Eliminar una no conformidad, Modificar: Descripción de la No conformidad, Descripción de la corrección, Descripción de la acción, Análisis de causa_raíz, Fecha de Ejecución de la No Conformidad, Fecha de Ejecución de una Acción.</p> <p>El caso de uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) realizan cada una de las funciones antes mencionadas a las no conformidades levantadas, según el privilegio que tengan.</p>
Referencias	R9, R10, R11, R12, R13 a), b), c), d), e), f)
Precondiciones	Debe haberse hecho un seguimiento a las diferentes Notas de No Conformidad levantadas.
Poscondiciones	
Requisitos especiales	

Tabla 3.13 Descripción del CU < Insertar verificación del cumplimiento de las Acciones>

Nombre del caso de Uso	Insertar verificación del cumplimiento de las Acciones.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) dan seguimiento a las diferentes no Conformidades levantadas y luego insertan la verificación del cumplimiento de las Acciones.
Referencias	<i>R9</i>
Precondiciones	Debe haberse hecho un seguimiento a las diferentes Notas de No Conformidad levantadas.
Poscondiciones	
Requisitos especiales	

Tabla 3.14 Descripción del CU < < Insertar verificación del cumplimiento de las Correcciones>

Nombre del caso de Uso	Insertar verificación del cumplimiento de las Correcciones.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) dan seguimiento a las diferentes no Conformidades levantadas y luego verifican el cumplimiento de las Correcciones.
Referencias	<i>R10</i>

Precondiciones	Debe haberse hecho un seguimiento a las diferentes No Conformidades levantadas.
Poscondiciones	
Requisitos especiales	

Tabla 3.15 Descripción del CU < Cerrar la No Conformidad >

Nombre del caso de Uso	Cerrar la No Conformidad
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) verifican que las correcciones o acciones tomadas erradicaron el problema y proceden a cerrar las No Conformidades levantadas.
Referencias	R11
Precondiciones	Insertan la verificación del cumplimiento de las acciones.
Poscondiciones	Erradicado el problema.
Requisitos especiales	

Tabla 3.16 Descripción del CU < Eliminar una No Conformidad >

Nombre del caso de Uso	Eliminar una No Conformidad
Actores	Especialista en Gestión de la Calidad (Inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad da seguimiento a las diferentes No Conformidades levantadas y determina que alguna de estas no tiene relevancia, seguidamente tiene la facultad de eliminarla, previa consulta con la persona que detecta la desviación.
Referencias	R12
Precondiciones	
Poscondiciones	
Requisitos especiales	Sólo puede Eliminar la No Conformidad el Especialista en Gestión de la Calidad.

Tabla 3.17 Descripción del CU <Modificar Descripción de la No Conformidad >

Nombre del caso de Uso	Modificar Descripción de la No Conformidad.
Actores	Especialista en Gestión de la Calidad (Inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad da seguimiento a las diferentes No Conformidades levantadas y determina que alguna de estas no está bien elaborada en

	correspondencia con la desviación, en este caso tiene la facultad de reelaborar la descripción de la misma.
Referencias	<i>R13_ a)</i>
Precondiciones	Debe haber sido levantada una Nota de No Conformidad.
Poscondiciones	
Requisitos especiales	Sólo puede Modificar la Descripción de la No Conformidad el Especialista en Gestión de la Calidad.

Tabla 3.18 Descripción del CU < Modificar Descripción de la Corrección >

Nombre del caso de Uso	Modificar Descripción de la Corrección
Actores	Especialista en Gestión de la Calidad (Inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad da seguimiento a las diferentes No Conformidades levantadas y determina que la corrección tomada no se encuentra acorde al problema detectado, tiene la facultad de redefinir la descripción de la Corrección tomada.
Referencias	<i>R13_ b)</i>
Precondiciones	Debe haber sido levantada una Nota de No Conformidad.
Poscondiciones	Debe notificar a los implicados en la corrección.
Requisitos especiales	Sólo puede Modificar la Descripción de la Corrección el Especialista en Gestión de la Calidad.

Tabla 3.19 Descripción del CU < Modificar Descripción de la Acción >

Nombre del caso de Uso	Modificar Descripción la Acción
Actores	Especialista en Gestión de la Calidad (inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad da seguimiento a las diferentes No Conformidades levantadas y hace una Descripción de la Acción tomada.
Referencias	R13_ c)
Precondiciones	Debe haber sido levantada una No Conformidad.
Poscondiciones	
Requisitos especiales	Sólo puede Modificar la Descripción de la Acción el Especialista en Gestión de la Calidad.

Tabla 3.20 Descripción del CU < Modificar Análisis de Causa_Raíz >

Nombre del caso de Uso	Modificar Análisis de Causa_Raíz
Actores	Especialista en Gestión de la Calidad (inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad da seguimiento a las diferentes no Conformidades levantadas y determina que no se realizó un análisis minucioso de las deficiencias existentes en un área determinada y prosigue a reelaborar el análisis.
Referencias	R13_ d)
Precondiciones	Debe haberse hecho un análisis de causa raíz.
Poscondiciones	

Requisitos especiales	Sólo puede Modificar el Análisis de Causa_Raíz el Especialista en Gestión de la Calidad.
------------------------------	--

Tabla 3.21 Descripción del CU <Modificar Fecha de Ejecución de la Corrección>

Nombre del caso de Uso	Modificar Fecha de Ejecución de la Corrección.
Actores	Especialista en Gestión de la Calidad (Inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad define la Fecha de Ejecución de la Corrección.
Referencias	R13_ e)
Precondiciones	Debe haber sido levantada una Nota de No Conformidad.
Poscondiciones	
Requisitos especiales	Sólo puede Modificar la Fecha de Ejecución de la Corrección el Especialista en Gestión de la Calidad.

Tabla 3.22 Descripción del CU < Fecha de Ejecución de la Acción >

Nombre del caso de Uso	Fecha de Ejecución de la Acción.
Actores	Especialista en Gestión de la Calidad (Inicia)
Resumen	El Caso de Uso inicia cuando Especialista en Gestión de la Calidad define la Fecha de Ejecución de la Acción.
Referencias	R13_ f)
Precondiciones	Debe haber sido levantada una Nota de No Conformidad.

Poscondiciones	
Requisitos especiales	Sólo puede Modificar la Fecha de Ejecución de la Acción el Especialista en Gestión de la Calidad

Tabla 3.23 Descripción del CU < Reporte de Acciones >

Nombre del caso de Uso	Reporte de Acciones
Actores	Todos (Inician)
Resumen	El Caso de Uso incluye los casos de usos: Reporte de Acciones Correctivas, Reporte de Acciones Preventivas. El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) muestran los reportes.
Referencias	<i>R14, R15</i>
Precondiciones	Deben haber sido tomadas las Acciones.
Poscondiciones	
Requisitos especiales	

Tabla 3.24 Descripción del CU <Mostrar Reportes de Acciones Preventivas>

Nombre del caso de Uso	Reporte de Acciones Preventivas
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) luego de realizar el análisis correspondiente a cada dificultad existente, hacen los Reportes de Acciones Preventivas

	pertinentes.
Referencias	<i>R14</i>
Precondiciones	Deben haber sido tomadas las Acciones.
Poscondiciones	
Requisitos especiales	

Tabla 3.25 Descripción del CU <Mostrar Reportes de Acciones Correctivas>

Nombre del caso de Uso	Mostrar Reportes de Acciones Correctivas.
Actores	Todos (Inician)
Resumen	El Caso de Uso inicia cuando los Usuarios (Especialista de Área, Especialista en Gestión de la Calidad o Auditores) luego de realizar el análisis correspondiente a cada dificultad existente, hacen los Reportes de Acciones Correctivas pertinentes.
Referencias	<i>R15</i>
Precondiciones	Deben haber sido tomadas las Acciones.
Poscondiciones	
Requisitos especiales	

ANEXO 2

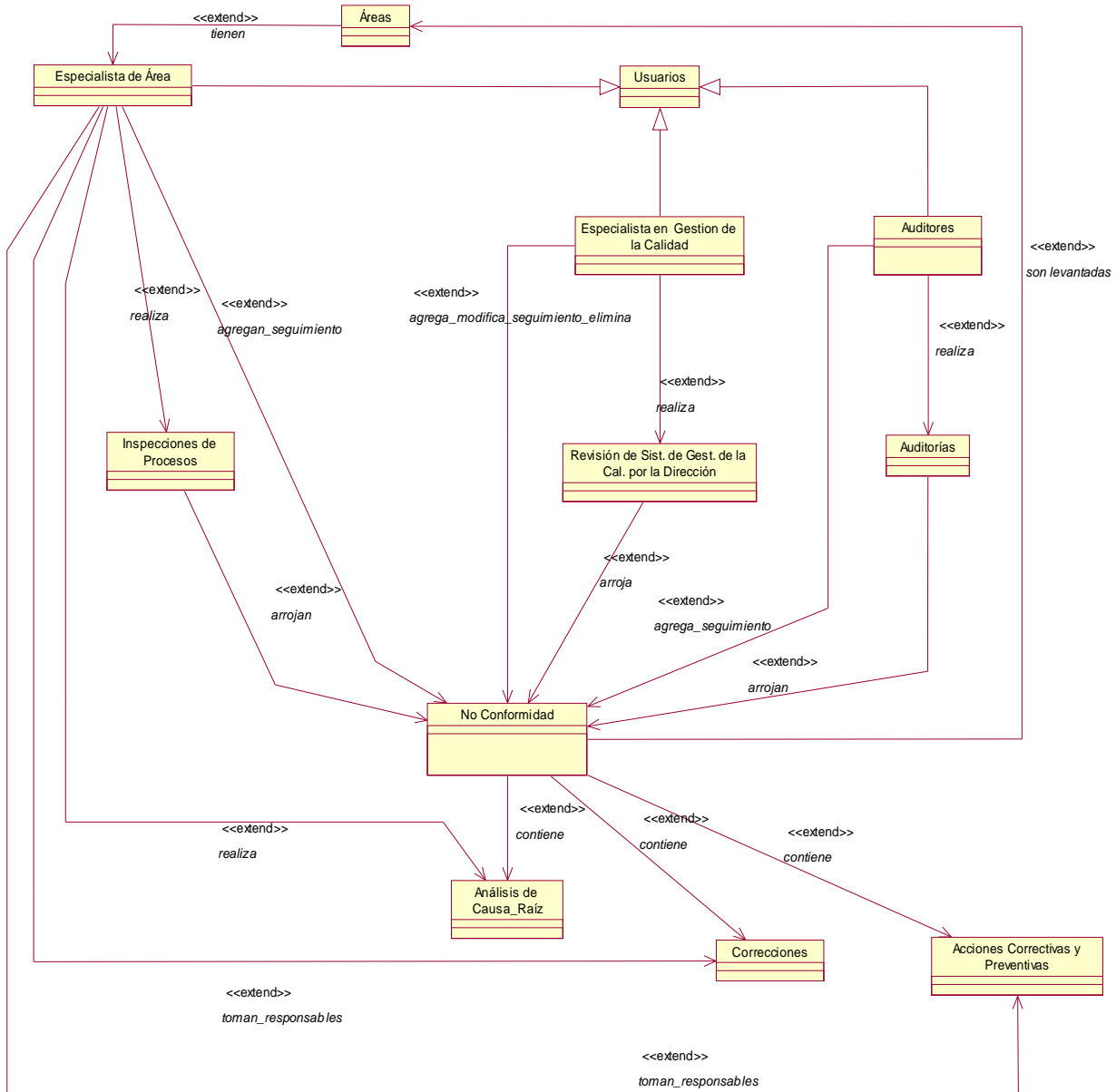


Figura 3.8 Representación del Modelo de Dominio.

ANEXO 3

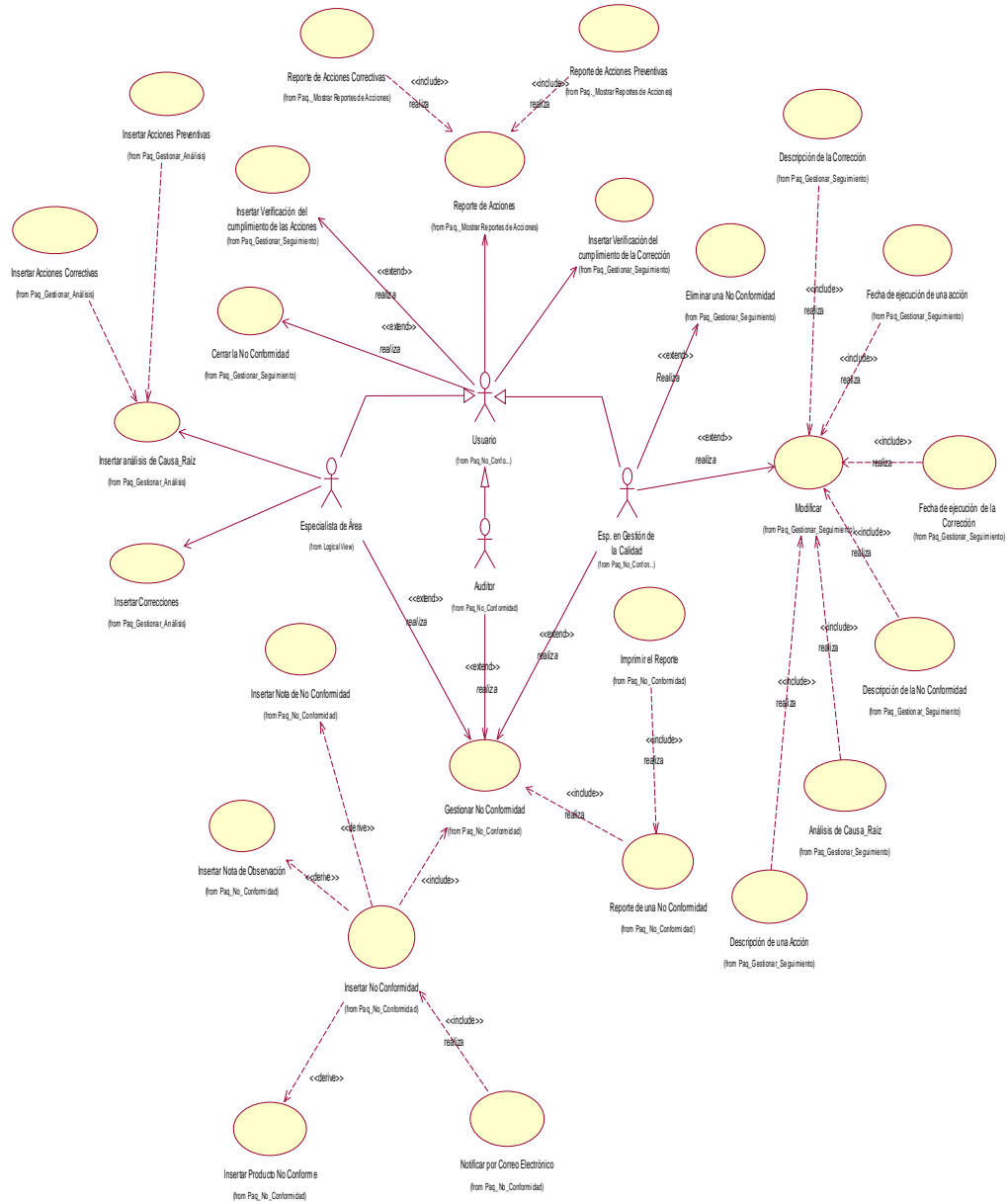


Figura 3.9 Diagrama de caso de Uso del sistema a automatizar

ANEXO 4

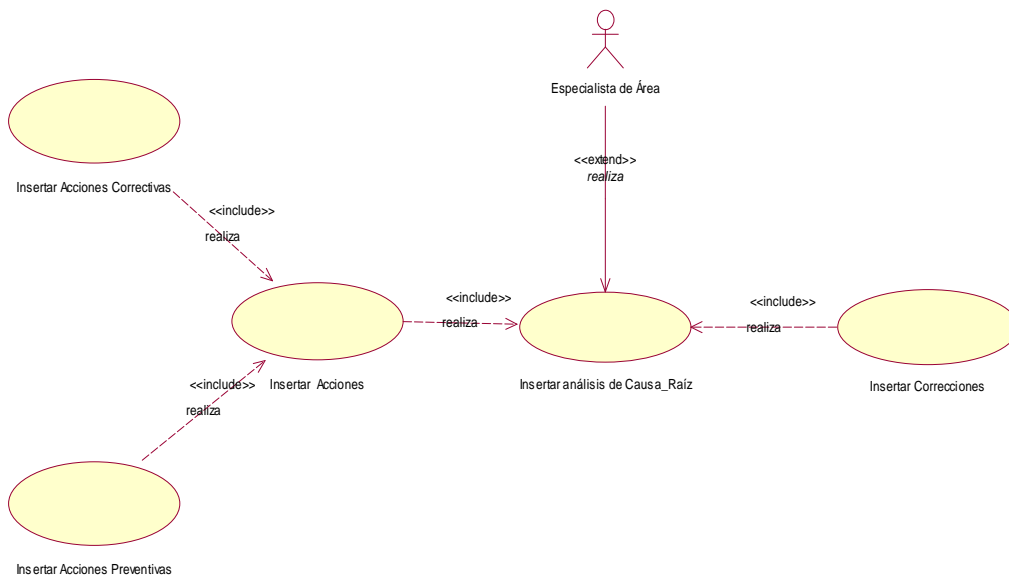


Figura 3.10 Paquete Análisis

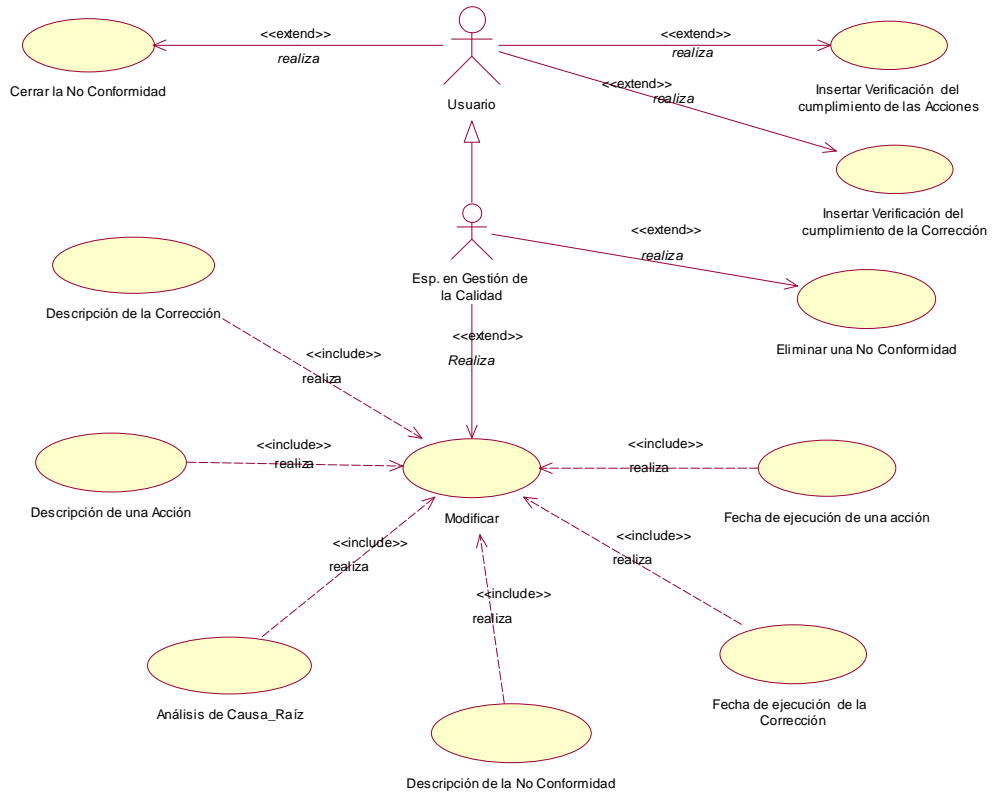


Figura 3.11 Paquete Gestionar Seguimiento

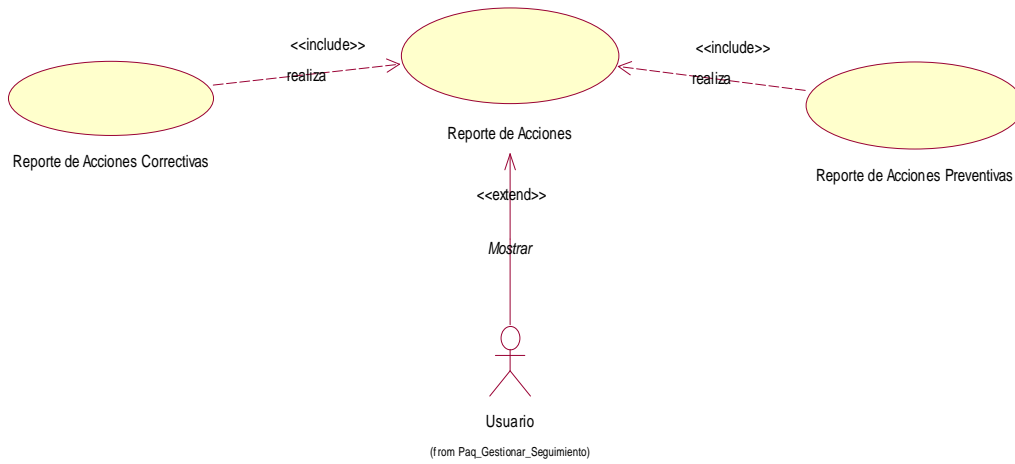


Figura 3.12 Paquete Reportes de Acciones

Anexo 5

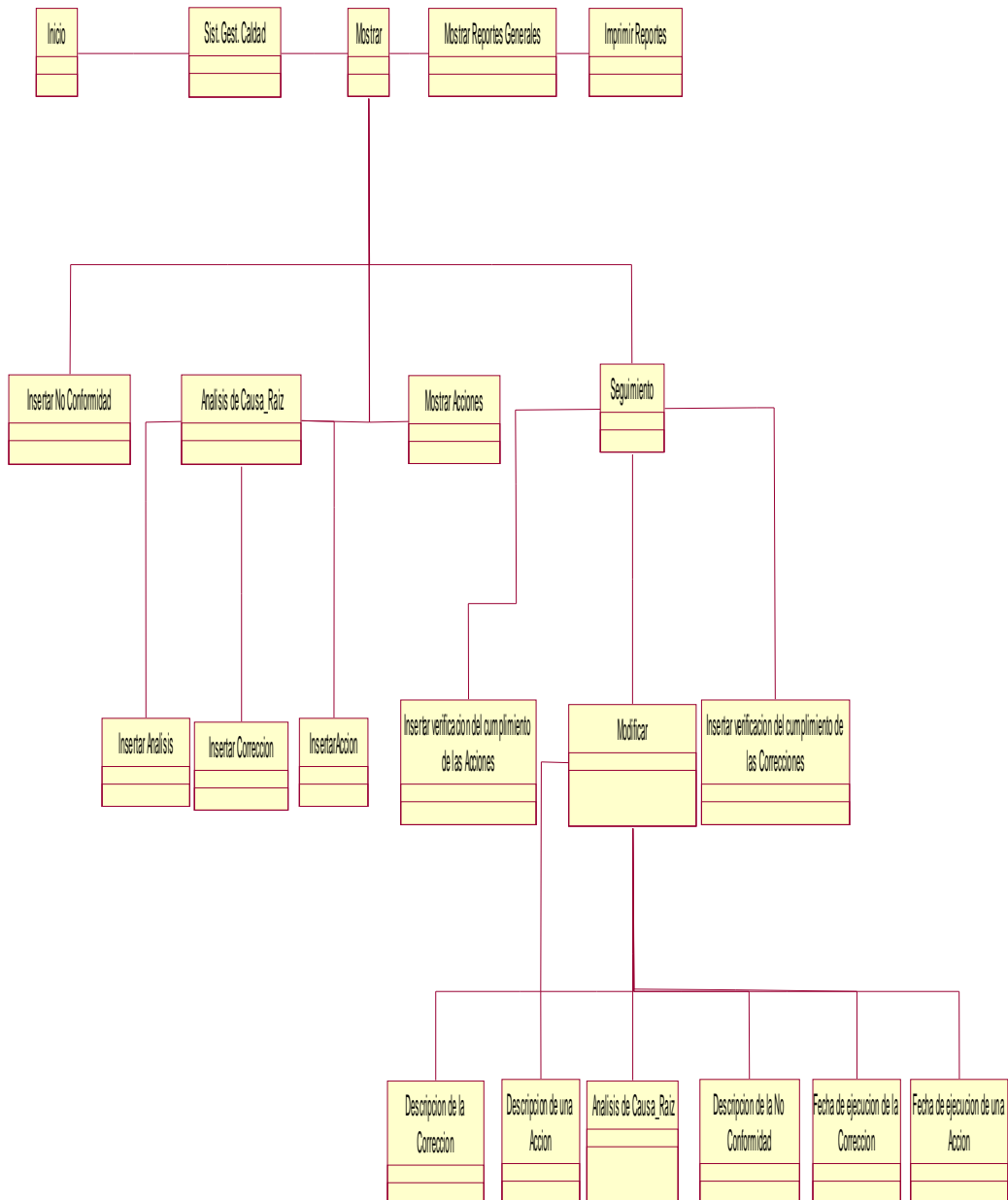


Figura 3.13 Mapa de navegación

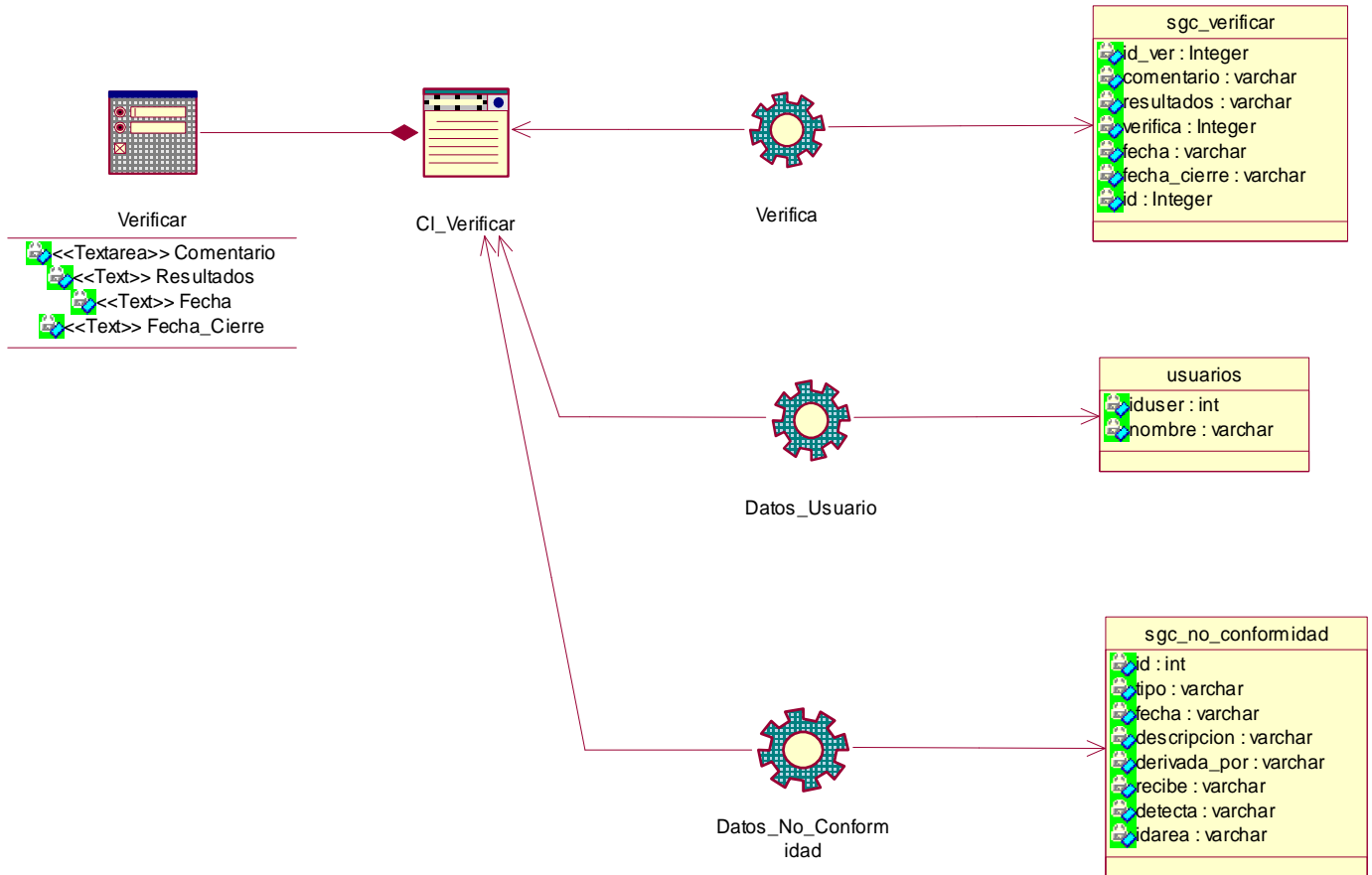


Figura 3.15 Diagrama de clase Web Verificar.

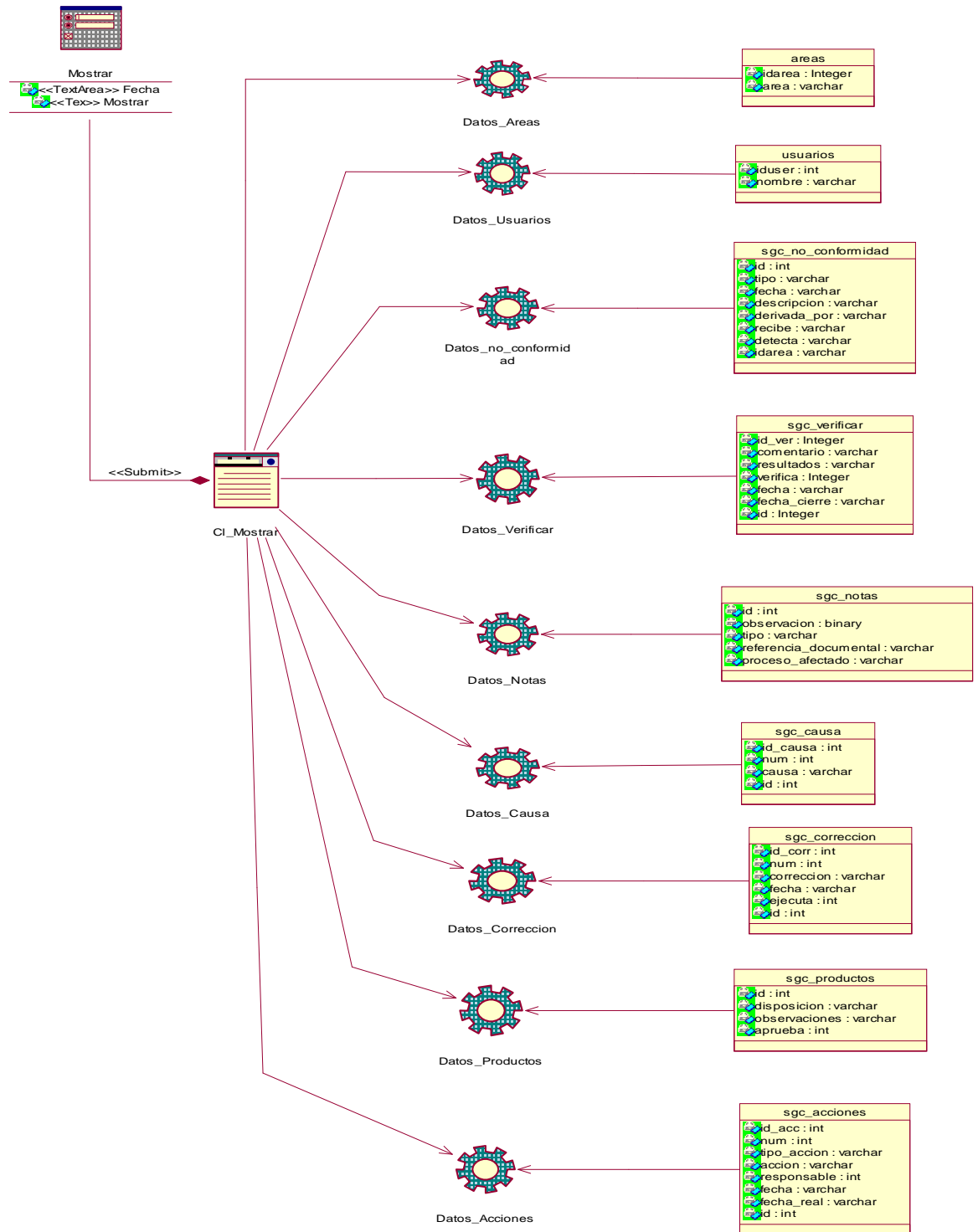


Figura 3.16 Diagrama de clase Web Mostrar.

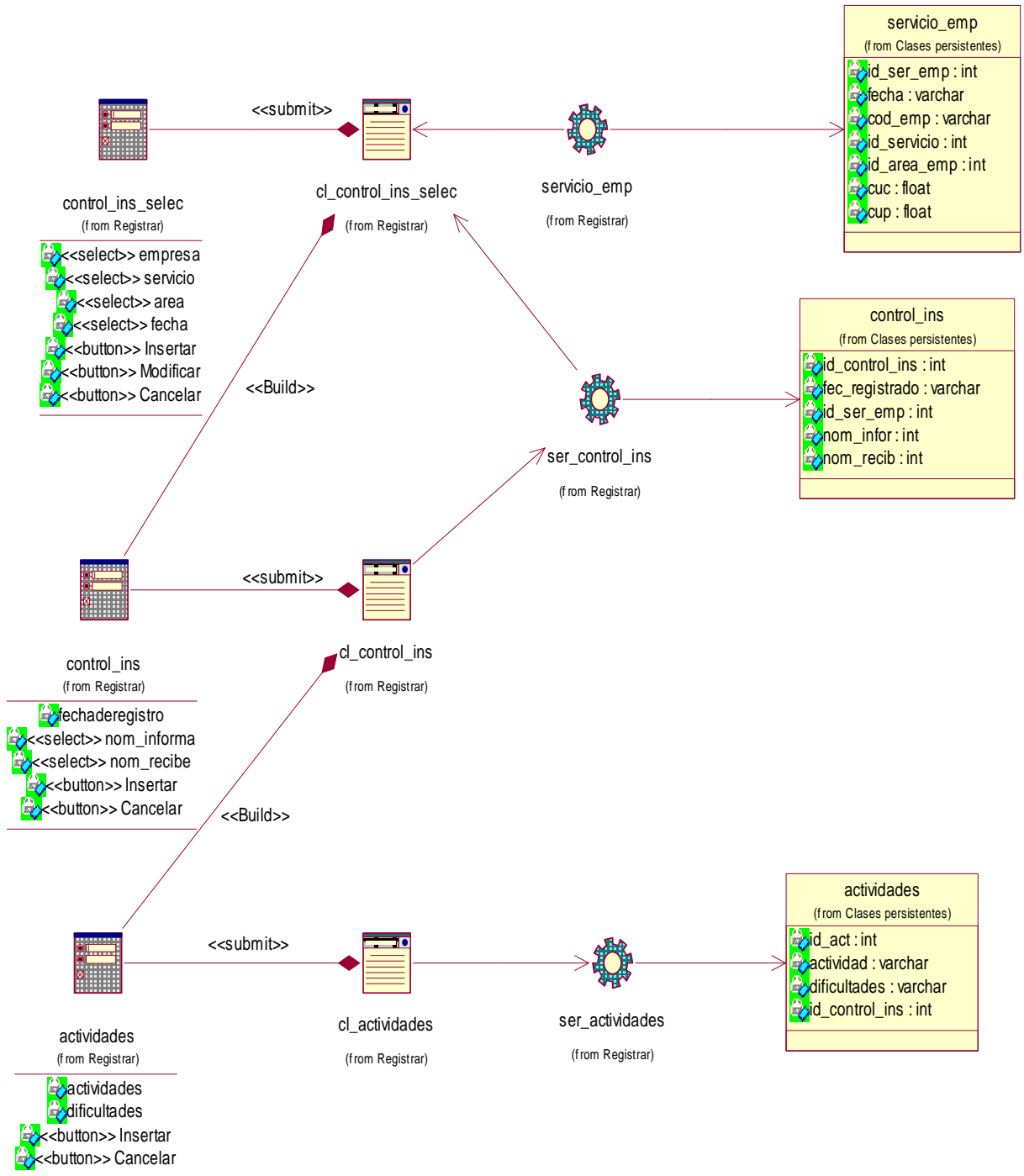


Figura 3.17 Diagrama de clase Web Insertar Nota.

Anexo 7

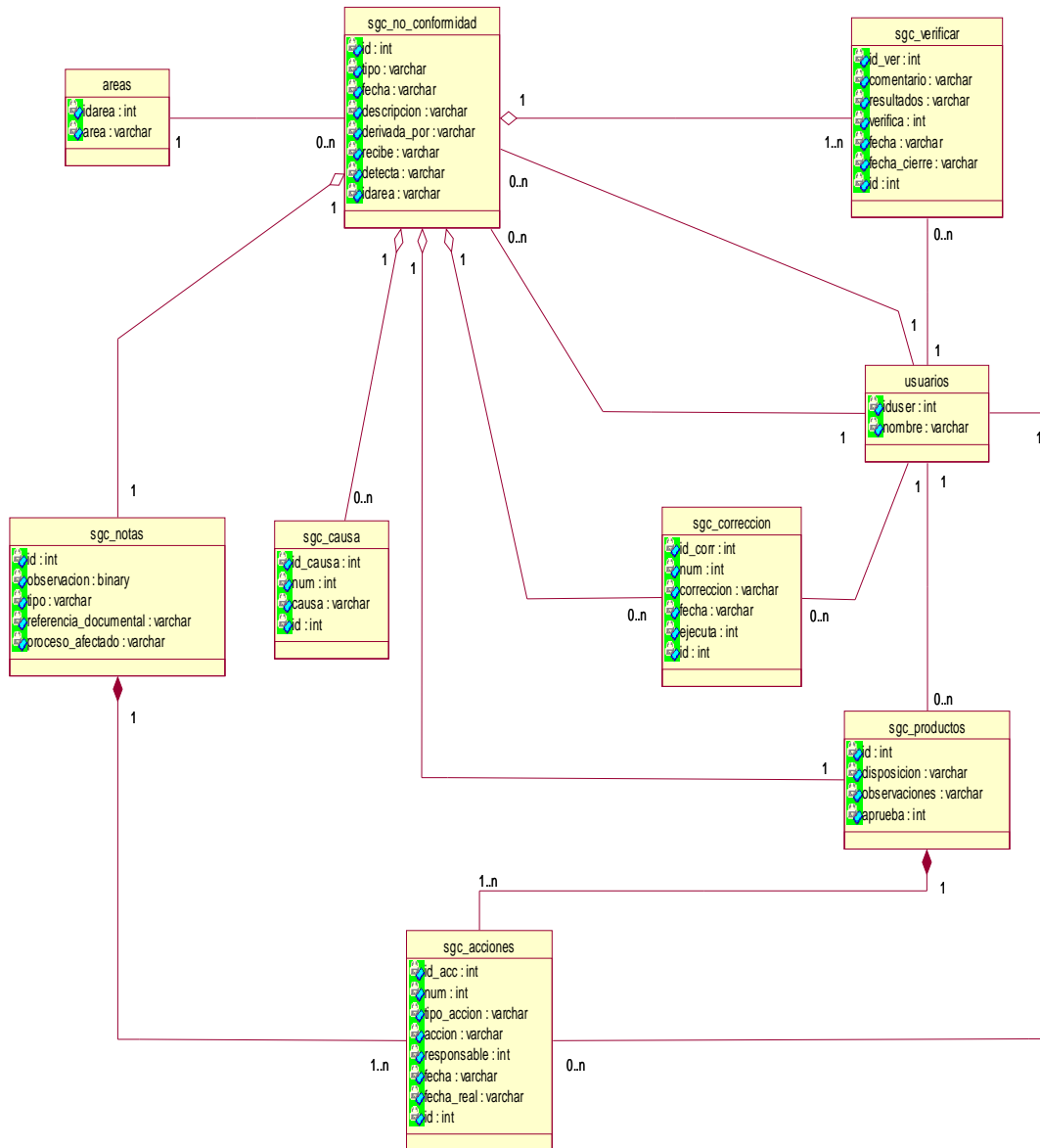


Figura 3.18 Diagrama de Clases Persistente.

Anexo 8

Descripción textual de las clases persistentes

Tabla 3.26 Descripción textual de la clase persistente *sgc_notas*

Nombre de la clase	sgc_notas
Propósito	Estas pueden ser las notas de observación, de no conformidad dejan plasmada la deficiencia existente.
Atributos	Descripción de Atributos
Id	Identificador.
Observación	Si es de observación.
Tipo	Si es real o potencial la nota de no conformidad.
referencia_documental	Referencia el lugar donde está planteado como se debe llevar el proceso.
proceso _ afectado	El proceso donde se encuentra la desviación.

Tabla 3.27 Descripción textual de la clase persistente *sgc_causa*

Nombre de la clase	sgc_causa
Propósito	Se da a conocer las causas existentes.
Atributos	Descripción de Atributos
id_causa	Identificador de la Causa.
num	Número.
causa	Descripción de la causa.
id	Identificador.

Tabla 3.28 Descripción textual de la clase persistente sgc_verificar

Nombre de la clase	sgc_verificar
Propósito	Verificar las acciones tomadas
Atributos	Descripción de Atributos
Id_ver	Identificador verificar
Resultados	Descripción de los resultados.
Verifica	Usuario.
Fecha	Fecha.
fecha_cierre	Fecha de cierre.
Id	Identificador.
Comentario	Breve comentario acerca de la verificación hecha.

Tabla 3.29 Descripción textual de la clase persistente sgc_correccion

Nombre de la clase	sgc_corrección
Propósito	Tomar correcciones dado un problema existente
Atributos	Descripción de Atributos
id_corr	Identificador de la Corrección.
Num	Número
Corrección	Descripción de la corrección.
Fecha	Fecha.
Ejecuta	Usuario
Id	Identificador.

Tabla 3.30 Descripción textual de la clase persistente sgc_productos

Nombre de la clase	sgc_productos
Propósito	De esta manera se conoce cuando el producto presenta problemas y que destino se le puede dar.
Atributos	Descripción de Atributos
Id	Identificador.

Disposición	Lo que se va hacer con el producto defectuoso.
Observaciones	Las distintas observaciones que se tienen en cuenta para realizar las correcciones pertinentes.
Aprueba	Usuario

Tabla 3.31 Descripción textual de la clase persistente sgc_acciones

Nombre de la clase	sgc_acciones
Propósito	Se toman diferentes acciones (correctivas o preventivas) para eliminar el problema.
Atributos	Descripción de Atributos
id_acc	Identificador de la acción.
Num	Número.
tipo_accion	Si es una acción correctiva o preventiva.
Acción	Las acciones tomadas
Responsable	Usuario.
Fecha	Fecha.
fecha _ real	Fecha real.
Id	Identificador.

Tabla 3.32 Descripción textual de la clase persistente Areas

Nombre de la clase	Areas
Propósito	Locales en donde se realizan las diferentes inspecciones y se toman las No Conformidades.
Atributos	Descripción de Atributos
Idarea	Identificador de área.
Area	Las áreas de la entidad.

Tabla 3.33 Descripción textual de la clase persistente Usuarios

Nombre de la clase	Usuarios
Propósito	Son los encargados de realizar las diferentes tareas que le son asignadas
Atributos	Descripción de Atributos
Iduser	Identificador del usuario.
Nombre	Nombre del usuario.

Anexo 9

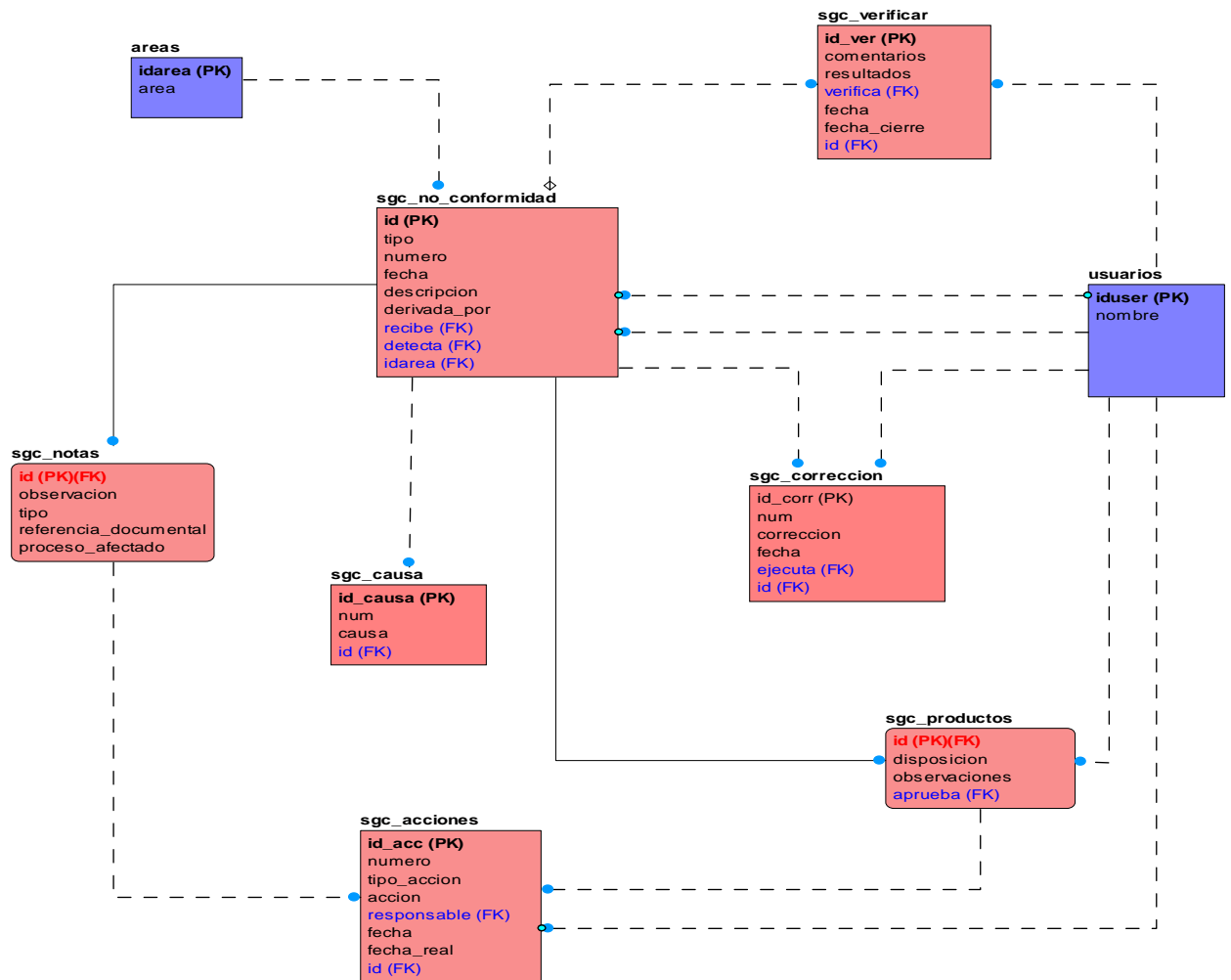


Figura 3.19 Modelo físico de datos.

Anexo 10

Diagrama de Secuencia.

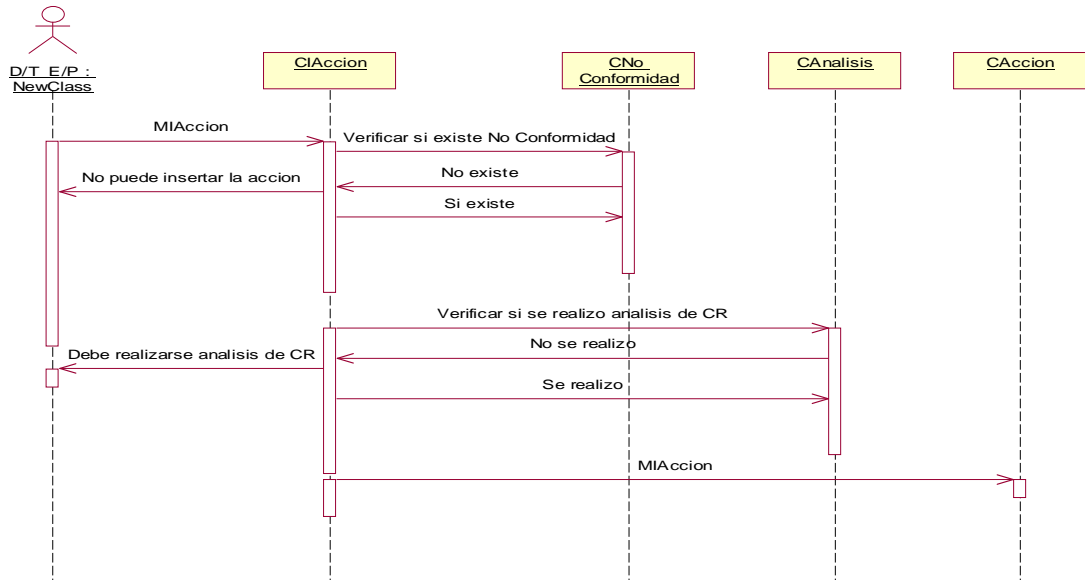


Figura 3.20 Diagrama de Secuencia Insertar Acción

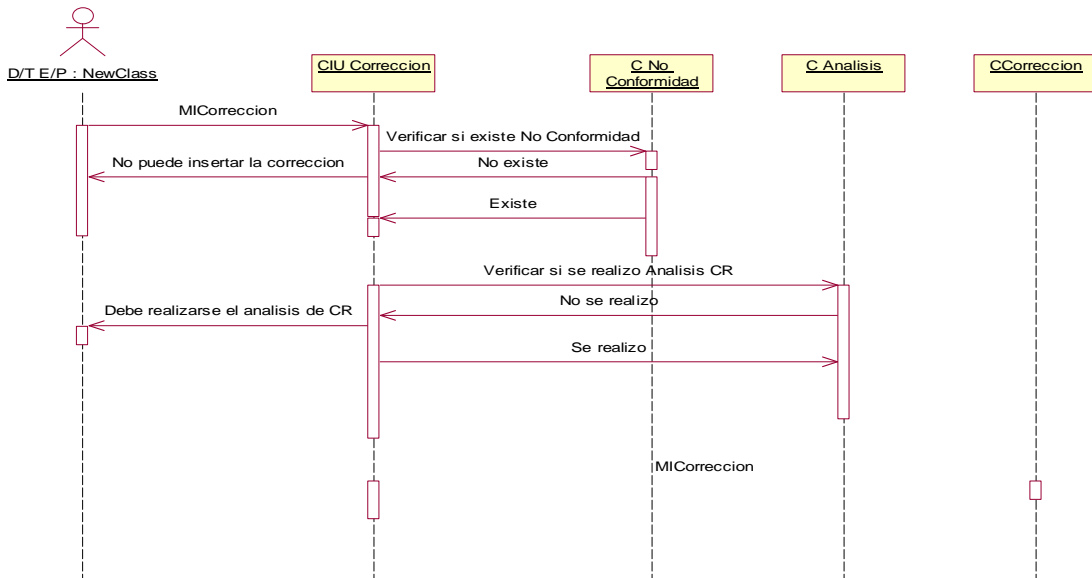


Figura 3.21 Diagrama de Secuencia Insertar Corrección

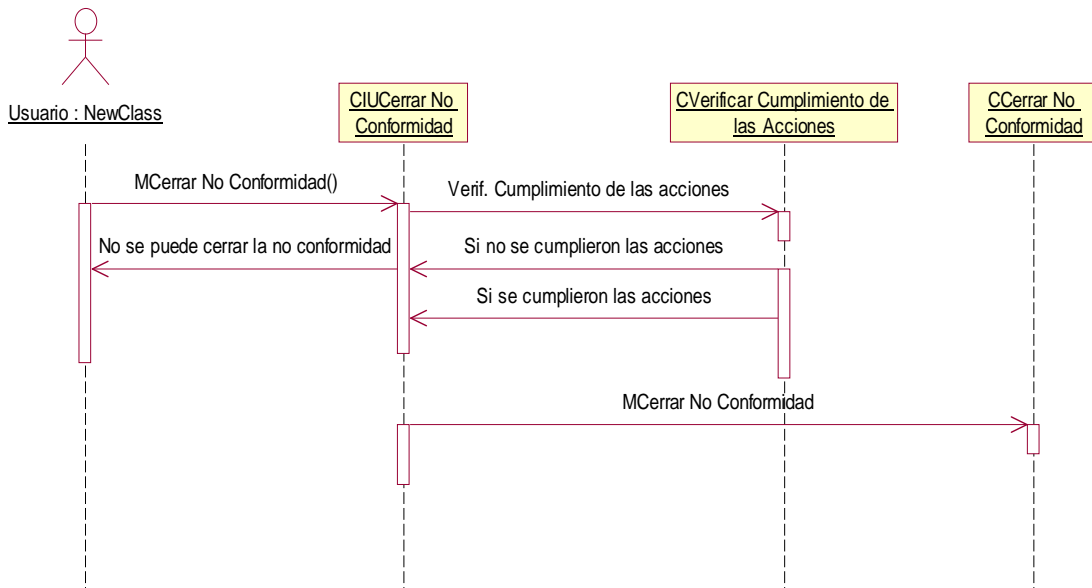


Figura 3.22 Diagrama de Secuencia Cerrar No Conformidad.

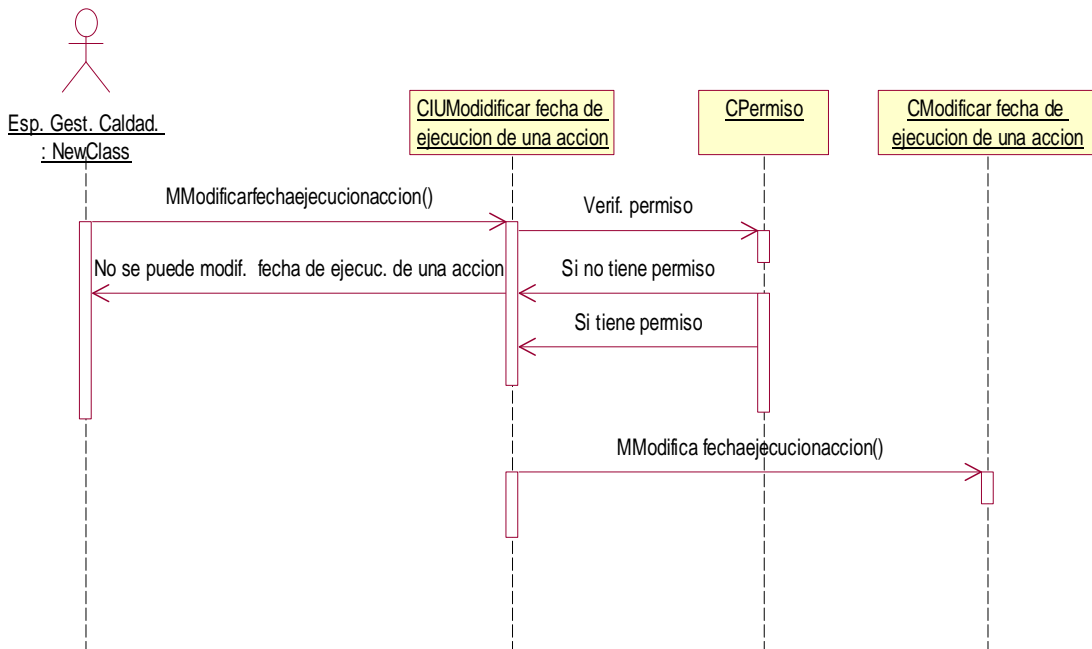


Figura 3.23 Diagrama de Secuencia Modificar fecha de ejecución de una Acción.

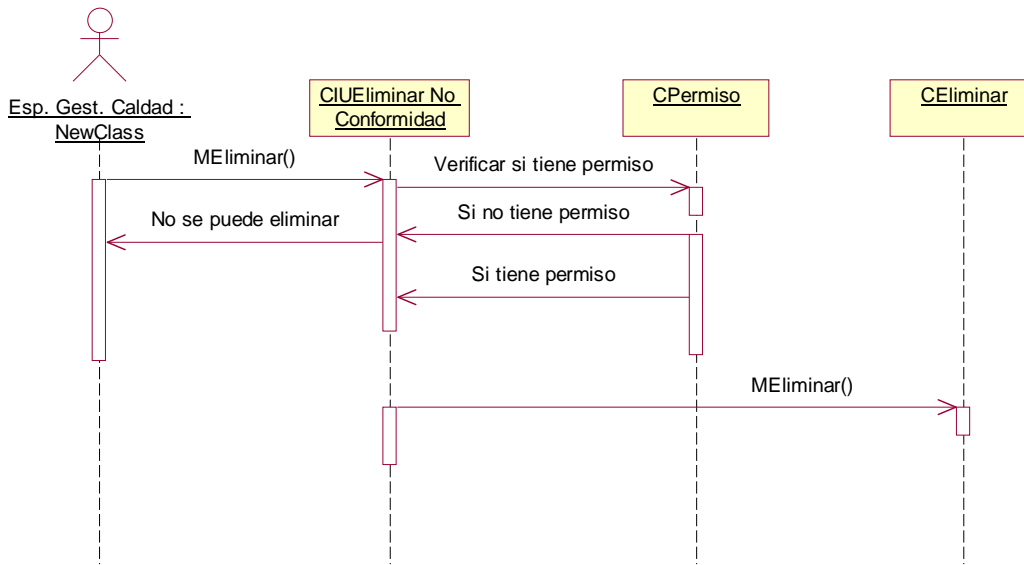


Figura 3.24 Diagrama de Secuencia Eliminar No Conformidad.

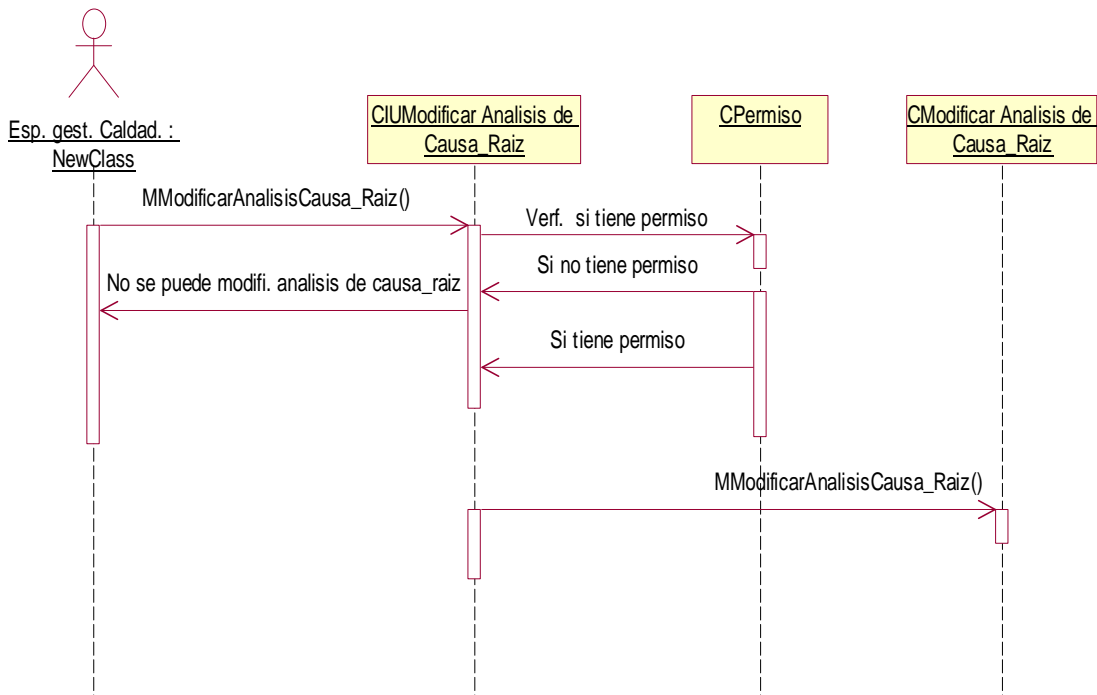


Figura 3.25 Diagrama de Secuencia Modificar análisis de Causa Raíz.

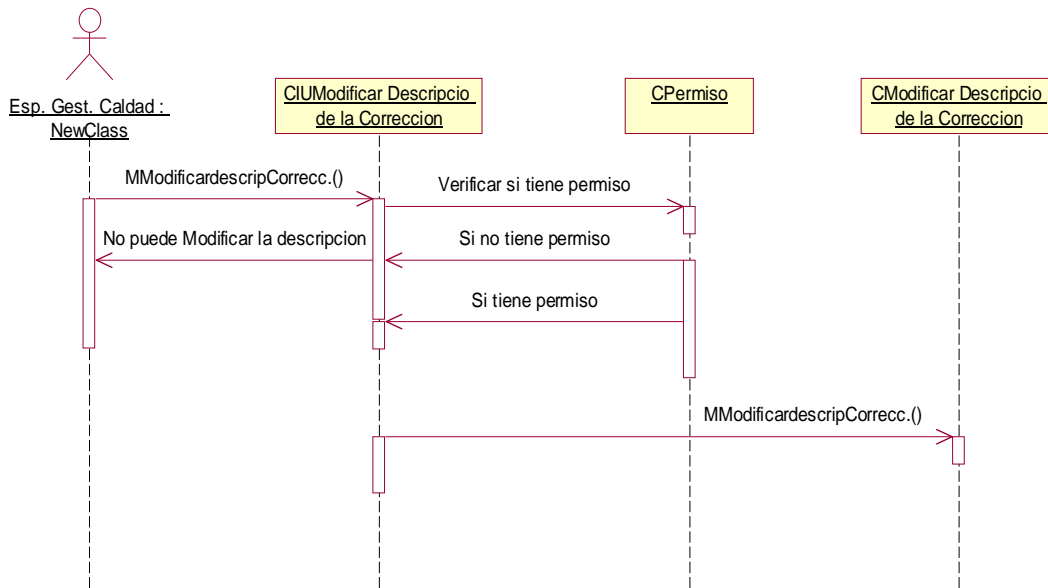


Figura 3.26 Diagrama de Secuencia Modificar Descripción de la Corrección.

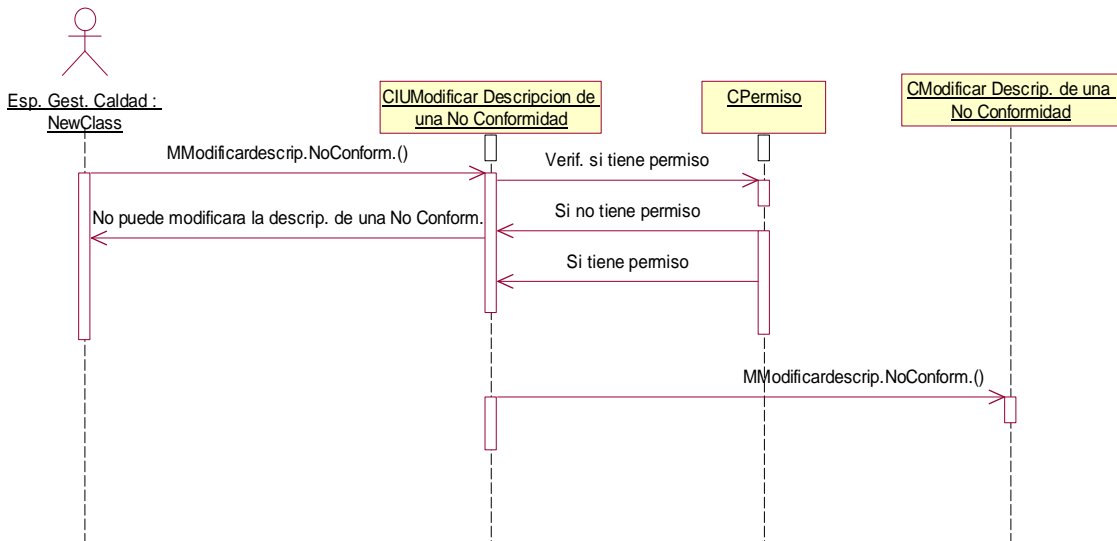


Figura 3.27 Diagrama de Secuencia Modificar Descripción de la No Conformidad.

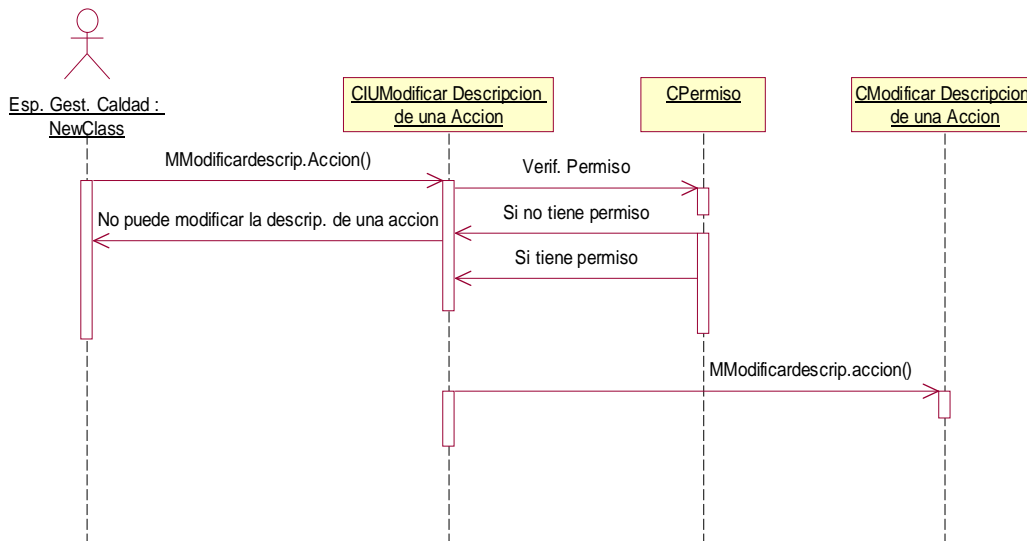


Figura 3.28 Diagrama de Secuencia Modificar Descripción de una Acción.

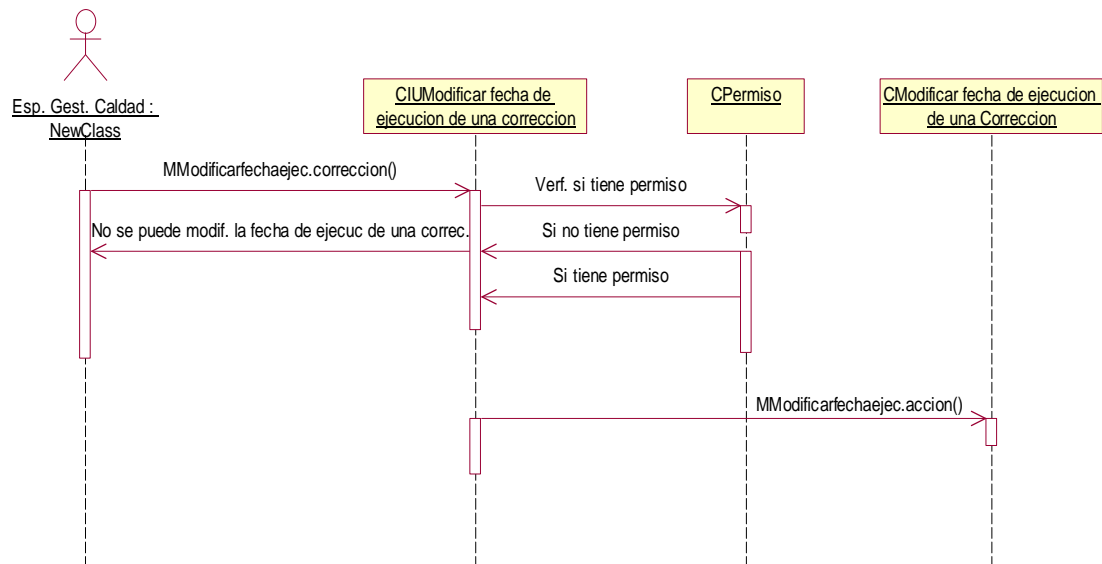


Figura 3.29 Diagrama de Secuencia Modificar fecha de ejecución de una Corrección.

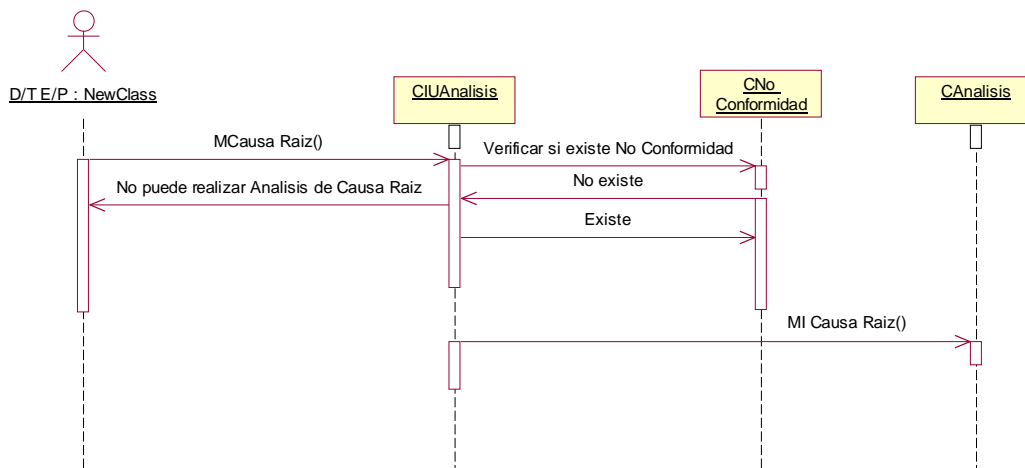


Figura 3.30 Diagrama de Secuencia Insertar Análisis

Anexo 11

Diagrama de Componentes

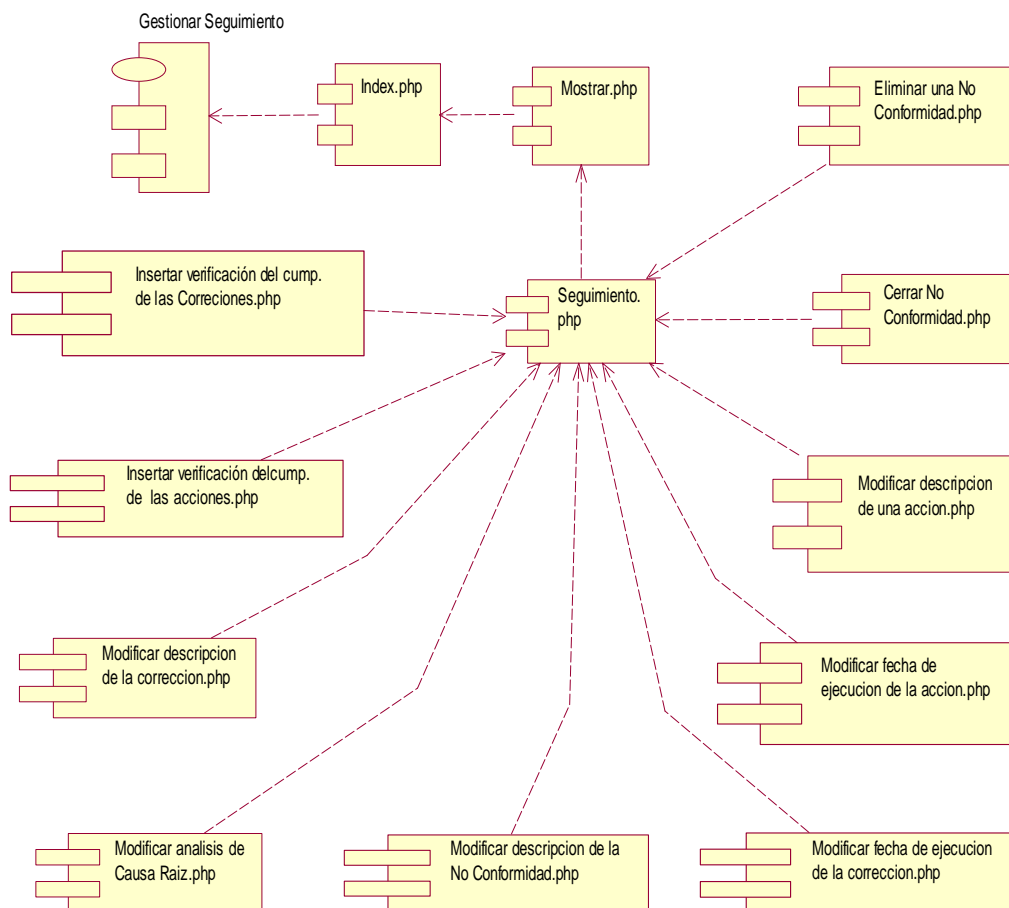


Figura 3.31 Diagrama de Componentes Gestionar Seguimiento

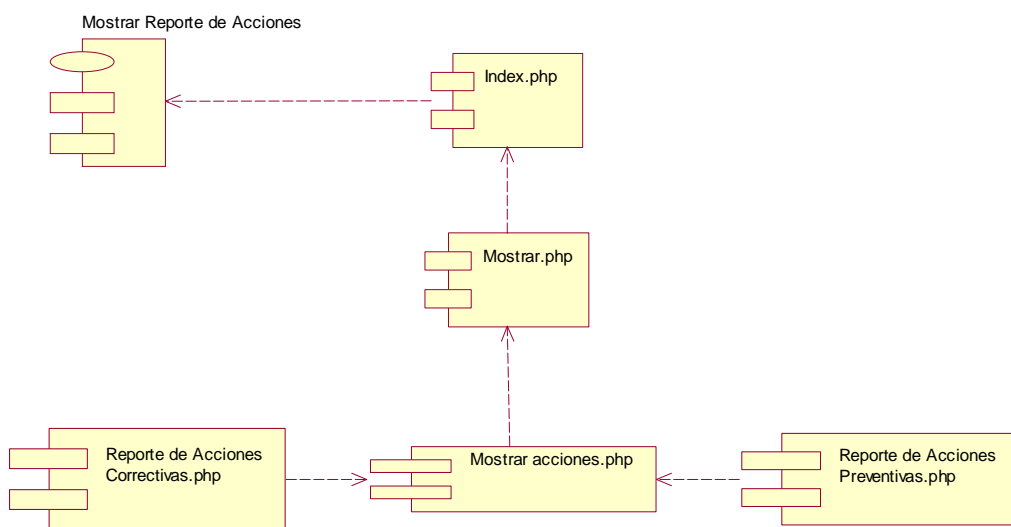


Figura 3.32 Diagrama de Componentes Reporte de Acciones

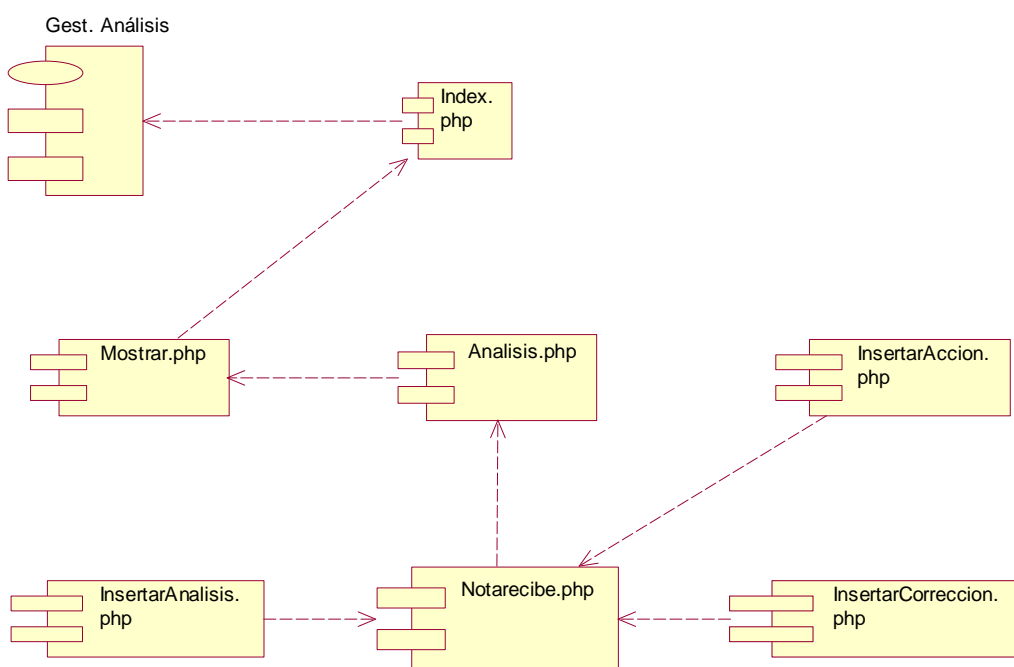


Figura 3.33 Diagrama de Componentes Análisis