

**Universidad de las Ciencias Informáticas**

**Facultad 8**



*Análisis y diseño del Sistema para la Gestión de  
la Calidad en la Facultad 8: Módulo Revisiones  
y Auditorías.*

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

**Autor:** Roberto Rodríguez Ríos

**Tutora:** Msc. Ing. Haydée M<sup>a</sup> Cruz Torres

**Cotutora:** Ing. Lidisvey Herrero Gonzalez

Junio de 2009

“Año del 50 Aniversario de la Revolución”

## Resumen

En el presente trabajo se documentan el análisis y diseño de uno de los módulos del sistema para la Gestión de la Calidad en la Facultad 8, con el objetivo de propiciar su futura implementación y prueba. El módulo consiste en una aplicación web destinada a automatizar las actividades de control: revisiones y auditorías llevadas a cabo por el grupo de calidad. Para el desarrollo del sistema, se siguieron los pasos que propone el Rational Unified Process (RUP), para los flujos de trabajo: Modelación del Negocio, Requisitos, Análisis y Diseño.

Atendiendo a las indicaciones de la Dirección de Calidad de la UCI sobre la realización de las revisiones y auditorías, a la experiencia de la asesora de Calidad de la Facultad y otros expertos del tema, se identificaron y describieron los procesos fundamentales llevados a cabo durante las actividades de revisiones y auditorías. Se diseñó una aplicación que facilitará el control de la calidad de los proyectos de la facultad 8, a través de la orientación de pasos, artefactos y plantillas que pueden ser utilizados por los revisores y auditores del grupo de control. Además la aplicación brindará informaciones de interés que permitirán observar y medir la calidad de los proyectos de la facultad, así como el control y evaluación del personal que participará en las diferentes actividades de control.

# Índice

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>4</b>
1.1 INTRODUCCIÓN .....	4
1.2 INTRODUCCIÓN A LA CALIDAD DE SOFTWARE .....	4
1.3 CONTROL DE LA CALIDAD .....	4
1.3.1 <i>Actividades de Control de Calidad</i> .....	5
1.3.2 <i>Revisiones</i> .....	5
1.3.2.1 Tipos de Revisiones .....	6
1.3.3 <i>Auditorías</i> .....	6
1.3.3.1 Tipos de Auditorías .....	6
1.3.3.2 Etapas para la realización de auditorías .....	7
1.4 SISTEMAS SIMILARES PARA LA GESTIÓN DE LA CALIDAD .....	8
1.4.1 <i>KMKey Quality(KMKEY 2008)</i> .....	8
1.4.2 <i>ISOSYSTEM Quality Pack(SOFTEXPERT 2008)</i> .....	9
1.5 HERRAMIENTAS DE MODELADO DE SISTEMAS .....	9
1.5.1 <i>Visual Paradigm</i> .....	9
1.5.2 <i>Rational Rose</i> .....	10
1.5.3 <i>Selección de la herramienta CASE para el modelado del sistema</i> .....	10
1.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE .....	10
1.6.1 <i>El Proceso Unificado de Desarrollo (RUP)</i> .....	10
1.6.2 <i>Extreme Programing (XP)</i> .....	11
1.6.3 <i>Microsoft Solution Framework (MSF)</i> .....	12
1.6.4 <i>Selección de la Metodología a utilizar</i> .....	13
1.7 DISEÑO DE APLICACIONES WEB .....	14
1.7.1 <i>Patrones de diseño y arquitectura</i> .....	15
1.8 DESARROLLO DE APLICACIONES WEB .....	17
1.8.1 <i>Lenguajes del lado del Cliente</i> .....	17
1.8.1.1 HTML .....	17
1.8.1.2 Java Script .....	18
1.8.1.3 Selección del (los) lenguaje (s) a utilizar .....	18
1.8.2 <i>Lenguajes del lado del servidor</i> .....	19
1.8.2.1 PHP .....	19
1.8.2.2 ASP .....	19
1.8.2.3 Selección del (los) lenguaje (s) a utilizar .....	19
1.8.3 <i>Gestores de Bases de Datos</i> .....	20
1.8.3.1 MySQL .....	20

1.8.3.2	PostgreSQL .....	20
1.8.3.3	Selección del Gestor de BD a utilizar .....	20
1.9	CONCLUSIONES DEL CAPÍTULO .....	21
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA .....</b>		<b>22</b>
2.1	INTRODUCCIÓN .....	22
2.2	OBJETO DE ESTUDIO.....	22
2.2.1	<i>Problema y situación problemática</i> .....	22
2.2.2	<i>Objeto de automatización</i> .....	23
2.2.3	<i>Información que se maneja</i> .....	23
2.2.4	<i>Propuesta de sistema</i> .....	23
2.3	MODELO DE NEGOCIO .....	23
2.4	REGLAS DEL NEGOCIO A CONSIDERAR .....	24
2.5	ACTORES Y TRABAJADORES DEL NEGOCIO .....	24
2.6	DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	25
2.7	DESCRIPCIONES TEXTUALES CUN .....	26
2.8	DIAGRAMAS DE ACTIVIDADES.....	29
2.9	DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.....	31
2.10	ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA .....	32
2.10.1	<i>Requerimientos funcionales</i> .....	32
2.10.2	<i>Requerimientos no funcionales</i> .....	35
2.11	MODELACIÓN DEL SISTEMA .....	37
2.12	ACTORES DEL SISTEMA.....	37
2.13	DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	38
2.13.1	<i>Descripción textual de los casos de uso del sistema</i> .....	38
2.14	CONCLUSIONES DEL CAPÍTULO.....	58
<b>CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL MÓDULO .....</b>		<b>60</b>
2.1	INTRODUCCIÓN .....	60
2.2	MODELO DE ANÁLISIS.....	60
2.2.1	<i>Diagramas de Clases de Análisis</i> .....	60
2.3	MODELO DE DISEÑO.....	66
2.4	DIAGRAMAS DE INTERACCIÓN .....	66
2.5	DIAGRAMAS DE CLASES DEL DISEÑO .....	83
2.5.1	<i>Descripciones de las clases del diseño</i> .....	94
2.5.2	<i>Subsistema de acceso a datos</i> .....	114
2.6	DIAGRAMA DE CLASES PERSISTENTES .....	115
2.7	MODELO FÍSICO DE DATOS .....	115

2.8	DIAGRAMA DE DESPLIEGUE .....	116
2.9	CONCLUSIONES DEL CAPÍTULO .....	117
	<b>CONCLUSIONES.....</b>	<b>118</b>
	<b>RECOMENDACIONES.....</b>	<b>119</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>120</b>
	<b>GLOSARIO DE TÉRMINOS.....</b>	<b>122</b>

# Índice de Figuras

FIG. 1 - DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	26
FIG. 2 - DIAGRAMA DE ACTIVIDADES "REALIZAR AUDITORÍA" .....	30
FIG. 3 - DIAGRAMA DE ACTIVIDADES "REALIZAR REVISIÓN" .....	31
FIG. 4 - MODELO DE OBJETOS DEL NEGOCIO.....	32
FIG. 5 - DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	38
FIG. 6 - DIAGRAMA DE CLASES DE ANÁLISIS "GENERAR INFORME DE NO CONFORMIDADES" .....	61
FIG. 7 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR ACCIONES" .....	61
FIG. 8 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR ACTIVIDAD DE CONTROL" .....	62
FIG. 9 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR INFORME FINAL" .....	62
FIG. 10 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR LISTA DE CHEQUEO" .....	62
FIG. 11 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR MINUTA DE REUNIÓN" .....	63
FIG. 12 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR NO CONFORMIDAD" .....	63
FIG. 13 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR PLANTILLAS DE ACTIVIDAD DE CONTROL" .....	64
FIG. 14 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR RECOMENDACIÓN" .....	64
FIG. 15 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR TAREA" .....	65
FIG. 16 - DIAGRAMA DE CLASES DE ANÁLISIS "GESTIONAR PLAN" .....	65
FIG. 17 - DIAGRAMA DE CLASES DE ANÁLISIS "SOLICITAR REPORTES DE ACTIVIDADES DE CONTROL" .....	66
FIG. 18 - DIAGRAMA DE SECUENCIA "GENERAR DOCUMENTO DE NO CONFORMIDADES" .....	67
FIG. 19 - DIAGRAMA DE SECUENCIA "GENERAR DOCUMENTO DE NO CONFORMIDADES" .....	67
FIG. 20 - DIAGRAMA DE SECUENCIA "GENERAR DOCUMENTO DE NO CONFORMIDADES" .....	68
FIG. 21 - DIAGRAMA DE SECUENCIA "GESTIONAR ACCIONES" .....	68
FIG. 22 - DIAGRAMA DE SECUENCIA "GESTIONAR ACCIONES" .....	69
FIG. 23 - DIAGRAMA DE SECUENCIA "GESTIONAR ACCIONES" .....	69
FIG. 24 - DIAGRAMA DE SECUENCIA "GESTIONAR ACTIVIDAD DE CONTROL" .....	70
FIG. 25 - DIAGRAMA DE SECUENCIA "GESTIONAR ACTIVIDAD DE CONTROL" .....	70
FIG. 26 - DIAGRAMA DE SECUENCIA "GESTIONAR ACTIVIDAD DE CONTROL" .....	71
FIG. 27 - DIAGRAMA DE SECUENCIA "GESTIONAR ACTIVIDAD DE CONTROL" .....	71
FIG. 28 - DIAGRAMA DE SECUENCIA "GENERAR INFORME FINAL" .....	72
FIG. 29 - DIAGRAMA DE SECUENCIA "GENERAR INFORME FINAL" .....	72
FIG. 30 - DIAGRAMA DE SECUENCIA "GENERAR INFORME FINAL" .....	73
FIG. 31 - DIAGRAMA DE SECUENCIA "GESTIONAR LISTA DE RECOMENDACIONES" .....	73
FIG. 32 - DIAGRAMA DE SECUENCIA "GESTIONAR LISTA DE RECOMENDACIONES" .....	74
FIG. 33 - DIAGRAMA DE SECUENCIA "GESTIONAR LISTA DE RECOMENDACIONES" .....	74
FIG. 34 - DIAGRAMA DE SECUENCIA "GESTIONAR MINUTA DE REUNIÓN" .....	75
FIG. 35 - DIAGRAMA DE SECUENCIA "GESTIONAR MINUTA DE REUNIÓN" .....	76

FIG. 36 - DIAGRAMA DE SECUENCIA "GESTIONAR MINUTA DE REUNIÓN"	77
FIG. 37 - DIAGRAMA DE SECUENCIA "GESTIONAR NO CONFORMIDAD"	77
FIG. 38 - DIAGRAMA DE SECUENCIA "GESTIONAR NO CONFORMIDAD"	78
FIG. 39 - DIAGRAMA DE SECUENCIA "GESTIONAR NO CONFORMIDAD"	78
FIG. 40 - DIAGRAMA DE SECUENCIA "GESTIONAR NO CONFORMIDAD"	79
FIG. 41 - DIAGRAMA DE SECUENCIA "GESTIONAR TAREAS"	79
FIG. 42 - DIAGRAMA DE SECUENCIA "GESTIONAR TAREAS"	80
FIG. 43 - DIAGRAMA DE SECUENCIA "GESTIONAR TAREAS"	80
FIG. 44 - DIAGRAMA DE SECUENCIA "GESTIONAR TAREAS"	81
FIG. 45 - DIAGRAMA DE SECUENCIA "GESTIONAR PLAN"	81
FIG. 46 - DIAGRAMA DE SECUENCIA "GESTIONAR PLAN"	82
FIG. 47 - DIAGRAMA DE SECUENCIA "GESTIONAR PLAN"	82
FIG. 48 - DIAGRAMA DE SECUENCIA "GESTIONAR DATOS DE ACTIVIDADES DE CONTROL"	83
FIG. 49 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR ACTIVIDAD DE CONTROL"	84
FIG. 50 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR ACCIONES"	85
FIG. 51 - DIAGRAMA DE CLASES DEL DISEÑO "GENERAR INFORME DE NO CONFORMIDADES"	86
FIG. 52 - DIAGRAMA DE CLASES DEL DISEÑO "GENERAR INFORME FINAL"	87
FIG. 53 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR MINUTA DE REUNIÓN"	88
FIG. 54 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR NO CONFORMIDAD"	89
FIG. 55 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR PLAN"	90
FIG. 56 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR PLANTILLAS DE ACTIVIDADES DE CONTROL"	91
FIG. 57 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR RECOMENDACIONES"	92
FIG. 58 - DIAGRAMA DE CLASES DEL DISEÑO "GESTIONAR TAREAS"	93
FIG. 59 - DIAGRAMA DE CLASES DEL DISEÑO "SOLICITAR REPORTES DE ACTIVIDADES DE CONTROL"	94
FIG. 60 - DIAGRAMA DE CLASES DEL SUBSISTEMA DE ACCESO A DATOS	114
FIG. 61 - DIAGRAMA DE CLASES PERSISTENTES	115
FIG. 62 – MODELO FÍSICO DE DATOS	116
FIG. 63 - DIAGRAMA DE DESPLIEGUE	116

## Índice de Tablas

TABLA 1 – DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO .....	24
TABLA 2 - DESCRIPCIÓN DE LOS TRABAJADORES DEL NEGOCIO .....	25
TABLA 3 - DESCRIPCIÓN TEXTUAL DEL CASO DE USO “REALIZAR AUDITORÍA” .....	26
TABLA 4 - DESCRIPCIÓN TEXTUAL DEL CASO DE USO “REALIZAR REVISIÓN” .....	28
TABLA 5 - DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	37



# Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) alcanzan un desarrollo progresivo que ha posibilitado crear aplicaciones informáticas de considerable impacto. El auge de las TIC ha traído nuevos cambios en las concepciones y formas de trabajo de las organizaciones, las cuales se adaptan a las transformaciones con el fin de aprovechar los beneficios que aporta el uso de las tecnologías.

A medida que aumenta la complejidad de los sistemas de software surgen nuevos aspectos del desarrollo de aplicaciones que hasta ese momento no se habían tenido en cuenta, al menos de una forma explícita. De este modo, el proceso de desarrollo se ha ido convirtiendo gradualmente en una labor de ingeniería, en la que nuevas tareas, como la elaboración de especificaciones, el diseño del sistema, la construcción de prototipos, la integración y pruebas, la gestión de la configuración y otras muchas han ido cobrando importancia frente a las labores de programación que eran las que primaban en un inicio. La Ingeniería del Software ha ido respondiendo a esta situación con el desarrollo de nuevos modelos, notaciones, técnicas y métodos.

Los aportes tecnológicos en el campo de la informática hacen que ésta se haya convertido en una herramienta imprescindible en el manejo de la información y los servicios. Por la gran utilidad que tienen, los sistemas de gestión de información son muy utilizados en entornos académicos, corporativos y empresariales, donde se manejan grandes volúmenes de información.

La calidad no es sólo uno de los requisitos esenciales de cualquier producto, no sólo software, sino que en la actualidad es un factor estratégico clave, del que dependen la mayor parte de las organizaciones, no sólo para mantener su posición en el mercado sino incluso para asegurar su supervivencia. El desarrollo de un producto que satisfaga, en la mayor medida posible, los requerimientos del cliente, es la medida de calidad buscada en la producción. Los defectos en los productos o sistemas son el resultado de errores cometidos en etapas tempranas del desarrollo, y sólo en una tercera parte son el resultado de errores cometidos en etapas avanzadas. La idea de prevenir los defectos o detectarlos en etapas tempranas es una alternativa recomendable, si además se tiene en cuenta que medir la calidad después de elaborado el producto puede ser muy costoso en caso de que se detecten problemas derivados de imperfecciones en el diseño. La mejor forma de prevenir o contrarrestar este problema es controlar la calidad durante todas las etapas del ciclo de vida del software.

Las revisiones, al igual que las auditorías y las pruebas, constituyen algunas de las actividades que se pueden establecer para el Control de la Calidad. Su planificación, realización y seguimiento incide directamente en la calidad del producto final.

Cada una de las facultades de la Universidad de las Ciencias Informáticas (UCI) desarrolla diferentes tipos de aplicaciones informáticas referentes a un determinado perfil. En cada facultad existe un Grupo

de Calidad encargado de asesorar y controlar las actividades de aseguramiento, control y auditoría de la calidad en cada uno de los proyectos que se producen en ella. Los Grupos de Calidad de las facultades responden a una Dirección de Calidad que funciona como rectora de la producción en toda la universidad. El grupo de Calidad de la Facultad 8 desea centralizar la información de cada proyecto a través de una aplicación web compuesta por varios módulos: Capacitación, Planificación, Revisiones y Auditorías, Pruebas, Administración. El presente trabajo se refiere específicamente al desarrollo del módulo Revisiones y Auditorías.

El grupo de control de la Facultad 8 presenta varias dificultades a la hora de realizar las Auditorías y las Revisiones a los proyectos debido a que generalmente el recurso tiempo es bastante escaso y el personal no se encuentra debidamente capacitado para realizar las revisiones de ciertos artefactos. Los resultados de las inspecciones realizadas son difíciles de controlar por no existir un lugar ni una persona encargada de reunir los mismos. Los resultados pueden ser reunidos por varias personas con diversas peculiaridades, no se lograba almacenar estos resultados en un mismo formato de manera tal que permitieran fusionarse y obtener determinadas conclusiones.

Teniendo en cuenta lo expresado anteriormente se tiene como **problema a resolver** que resulta difícil en la facultad mantener actualizada la información de las revisiones y auditorías en cuanto a la detección de errores y posibles soluciones dentro de los procesos de desarrollo en los proyectos de la facultad, así como su seguimiento sistemático.

Para enfocar la investigación se trazó como **objetivo** realizar el Análisis y Diseño del Módulo Revisiones y Auditorías del Sistema para la Gestión de la Calidad de la facultad 8, de manera que facilite información para la toma de decisiones del Grupo de Calidad sobre los proyectos de la Facultad.

De este objetivo general se derivan los siguientes **objetivos específicos**:

- Facilitar la futura implementación del módulo a través de la documentación clara y precisa de todos los artefactos necesarios.
- Lograr que la aplicación modelada soporte los procesos actuales del equipo de control de la calidad de la Facultad para lograr eficiencia y calidad en estos procesos.
- Establecer la forma en que los revisores y auditores utilizarán los diferentes métodos, herramientas y manuales orientados por el asesor de calidad de la Facultad.
- Modelar una aplicación que permita comparar los proyectos de la facultad a partir de los resultados obtenidos en las diferentes actividades de control aplicadas.

El **objeto de estudio** consiste en el control de la calidad de software, por lo que el **campo de acción** se centrará en los procesos de revisiones y auditorías durante el desarrollo de software.

La investigación se desarrolló a través de las siguientes **tareas**:

- Documentar las herramientas, tendencias y tecnologías actuales relacionadas con el control de la calidad.
- Seleccionar la metodología que se utilizará para el desarrollo del sistema.
- Realizar un levantamiento de requisitos, para conocer las necesidades del cliente y lograr una familiarización con estos.
- Diseñar una aplicación que sirva de soporte a los procesos de revisiones y auditorías como parte del control de la calidad.
- Generar la documentación de la información referente al análisis y diseño del sistema.

Como **Idea a defender** se parte que si se desarrolla el análisis y el diseño del módulo de auditorías y revisiones, entonces se obtendrá un sistema que facilitará el trabajo del grupo de calidad de la facultad 8.

# Capítulo 1. Fundamentación Teórica

## 1.1 Introducción

Este capítulo está destinado a brindar una panorámica de los temas relacionados con la Calidad de Software y a partir de esto se detallan las actividades de revisiones y auditorías, sus características, procedimientos y clasificaciones, sintetizando el estado del arte de este tipo de actividades de control; además se mencionan algunas aplicaciones relacionadas con la gestión de la calidad. También se abordarán las diferentes teorías y fundamentos aplicados para el análisis y diseño del sistema, así como la descripción de los principales conceptos asociados al dominio del problema que son necesarios para entender el modelo de negocio y la propuesta de solución. En este capítulo se detallan los procesos elementales de negocio, se realiza un estudio y análisis de las tecnologías apropiadas para el desarrollo de aplicaciones Web, escogiendo la mejor opción para la construcción del sistema propuesto.

## 1.2 Introducción a la Calidad de Software

El significado de Calidad puede adquirir múltiples interpretaciones, ya que todo dependerá del nivel de satisfacción o conformidad del cliente. Sin embargo, la calidad es el resultado de un esfuerzo arduo, se trabaja de forma eficaz para poder satisfacer las necesidades del consumidor. Dependiendo de la forma en que un producto o servicio sea aceptado o rechazado por los clientes, se podría decir si éste es bueno o malo. (Calidad 2009).

El Control de Calidad dentro del Ciclo de Desarrollo de Software

Definición formal: – Software Quality Assurance (SQA) se define como un plan sistemático de evaluar la calidad y apagamiento a los procesos, políticas y estándares de un producto de software. SQA incluye el proceso de asegurar que los estándares y procedimientos sean establecidos y seguidos durante el ciclo de vida del desarrollo de software. El cumplimiento de los estándares y procedimientos acordados se evalúa mediante un proceso de monitoreo, evaluación del producto y auditorías. (ROCHE)

## 1.3 Control de la Calidad

Es el proceso de regulación a través del cual se puede medir la calidad real, compararla con las normas o las especificaciones y actuar sobre la diferencia.

Durante la investigación se encontraron diferentes significados de este concepto:

- Una parte del proceso de regulación. Por ejemplo: la inspección del producto.
- Históricamente, el nombre de un Departamento que se dedica a tiempo completo a la Función de la Calidad.

- Las herramientas, conocimientos prácticos o técnicas por medio de las cuáles se desarrollan algunas o todas las funciones.(JURAN)

La comprobación de la calidad del software debe ser una de las principales metas a trazar durante el desarrollo del mismo, esto implica definir cómo se medirá el grado de calidad que este logre alcanzar.(FERNÁNDEZ CARRASCO 2007)

### **1.3.1 Actividades de Control de Calidad**

Existen varias actividades que pueden ser ejecutadas para llevar a cabo el control de la calidad durante el proceso de desarrollo. Entre ellas se pueden encontrar las siguientes:

- Revisiones Técnicas Formales
- Inspecciones
- Auditorías de la Calidad
- Auditorías de la Gestión Documental
- Pruebas de Unidad
- Pruebas de Integración
- Pruebas de Sistema

En este trabajo se profundizará en dos grupos de actividades de control de la calidad, las revisiones y las auditorías. En los epígrafes siguientes se explican estas actividades, así como su clasificación y modo de ejecución.

### **1.3.2 Revisiones**

Actividad emprendida para asegurar la conveniencia, adecuación y eficacia del tema objeto de la revisión, para alcanzar unos objetivos establecidos.(*Norma internacional ISO 9000 2000*)

La revisión sistemática es un método de investigación desarrollado para obtener, analizar y evaluar toda la investigación relevante para una pregunta de investigación o un área de interés particular.(CORUÑA 2009)

Las revisiones de Software son un “filtro” para el proceso de Ingeniería del Software. Las revisiones se aplican en diferentes momentos del desarrollo y sirven para detectar errores y defectos que puedan ser eliminados. Sirven también para “purificar” las actividades de Ingeniería del Software que suceden como resultado del análisis, el diseño y la codificación.(PRESSMAN 2002)

De forma general se pueden describir las revisiones como actividades de control que se realizan de forma sistemática dentro del proceso de desarrollo de software para detectar errores.

### **1.3.2.1 Tipos de Revisiones**

Existen varios tipos de revisiones seguidamente se muestran dos tipos de revisiones, Revisiones formales e informales(CHAVEZ 2005):

#### **Informales**

- No hay proceso definido
- No existen roles
- Usualmente no planeadas

#### **Formales**

- Objetivos definidos
- Proceso documentado
- Roles definidos y personas entrenados en ellos
- Se utilizan listas de chequeo, reglas y métodos para encontrar defectos
- Reporte del resultado
- Recolección de datos para el control del proceso

### **1.3.3 Auditorías**

Proceso sistemático, independiente y documentado para obtener evidencias de la auditoría y evaluarla de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría.(Norma internacional ISO 9000 2000)

Una auditoría es una actividad de información, mediante la cual puede verificarse el cumplimiento del Sistema de Calidad establecido y la efectividad de dicho Sistema y, en caso contrario, evaluar la necesidad de una mejora o de una acción correctiva.(J.K 2009)

Tomando en cuenta todas las definiciones citadas anteriormente, se puede llegar a la conclusión que auditoría en la calidad no es más que un examen metódico de una situación relativa a un producto o proceso, en materia de calidad, realizado en cooperación con los interesados para verificar la concordancia de la realidad con lo preestablecido y la adecuación al objetivo buscado.

#### **1.3.3.1 Tipos de Auditorías**

Existen varias formas de clasificar las auditorías: del Sistema de Calidad, del Proceso, del Producto; que a su vez pueden ser Internas o Externas y también pueden ser De Adecuación o De Cumplimiento. (Norma internacional ISO 9000 2000; J.K 2009)

#### **Auditorías del Sistema de Calidad**

Son una actividad que se realiza para comprobar, mediante el examen y la evaluación de evidencias objetivas, que el Sistema de Calidad es adecuado y ha sido desarrollado, documentado y efectivamente implantado de acuerdo con los requisitos especificados.

#### Auditoría de Calidad del proceso

Consiste en el examen sistemático e independiente de los elementos de un proceso para determinar si las actividades y los resultados relativos a la Calidad satisfacen a las disposiciones previamente establecidas y si estas disposiciones se llevan a cabo efectivamente y son aptas para alcanzar los objetivos.

#### Auditoría de Calidad del Producto

Consiste en la estimación cuantitativa del cumplimiento de las características requeridas en el producto.

#### Auditorías Internas

Es importante para la empresa comprobar la efectividad de su Sistema de Calidad, siendo la propia empresa quien investiga sus propios sistemas, procedimientos y actividades para cerciorarse que son adecuados y que se cumplen.

Proporcionan información de si sus políticas están siendo cumplidas, de si los sistemas son tan eficientes y efectivos como deberían ser y si se precisa algún cambio. Pueden establecer una línea de comunicación por toda la compañía y convertirla en un gran elemento motivador.

Las Auditorías Internas van a ser las que se tendrán en cuenta en esta investigación pues son las que se desarrollan en la facultad.

#### **1.3.3.2 Etapas para la realización de auditorías**

La realización de una auditoría consta de los siguientes pasos:(*Norma internacional ISO 9000 2000*)

- **Planificación:** Consiste en definir los objetivos de la auditoría y su alcance. En esta etapa se elabora un Plan de Auditoría.
- **Llevar a cabo la investigación:** La auditoría se inicia con una reunión de apertura de la investigación, y se lleva a cabo mediante entrevistas y chequeos en los que se recopilan datos.
- **Analizar los datos recogidos:** El equipo de auditores debe hacer frente a cantidades considerables de datos, de entre los cuales resulta complicado seleccionar los datos relevantes, por lo que en ocasiones se utilizan técnicas de análisis estadístico. Posteriormente se realiza una evaluación en paralelo de los resultados por un grupo de evaluadores, se comparan las conclusiones obtenidas y se estudian las causas de las desviaciones significativas. Sin embargo esto se puede simplificar considerablemente definiendo con antelación los aspectos que se deben auditar. Para ello es

recomendable el uso de la plantilla definida en esta propuesta, para una mejor organización y manipulación de los datos.

- **Sugerir soluciones a los problemas encontrados y posibles mejoras:** En base al paso anterior los auditores pueden sugerir posibles soluciones a los problemas que se identificaron en la realización de la auditoría. Para esto puede convocarse una reunión, o realizarse de forma independiente.
- **Elaborar y presentar un informe de resultados:** Se deberá escoger un responsable de esta tarea, aunque en ella debe participar todo el equipo de auditores, para garantizar que los datos sean adecuadamente registrados.

## **1.4 Sistemas similares para la Gestión de la Calidad**

### **1.4.1 KMKey Quality(KMKEY 2008)**

KMKey Quality es un software de gestión de calidad ideal para la implantación y mantenimiento de un sistema de gestión de calidad (SGC) de cualquier tipo: ISO 9000 (ISO 9001:2000, ISO 14001, OHSAS 18001, etc.).

Basado en un entorno web que le proporcionará acceso universal desde cualquier ordenador autorizado, KMKey Quality incorpora un Gestor Documental donde almacenar las distintas revisiones de los documentos propios de un sistema de calidad, tales como el Manual de Calidad, la Política y los Objetivos de Calidad, el Organigrama, el Mapa de Procesos, los Procedimientos, los Formatos o las Instrucciones de Trabajo, permitiendo siempre un acceso directo a la revisión vigente. Además el gestor de expedientes le permitirá la elaboración colaborativa de documentos, gestionando la edición, la revisión y la aprobación de los mismos, integrando además un sistema de notificaciones digitales por e-mail que facilita la comunicación a los diferentes usuarios.

Desde el mismo gestor de expedientes cualquier usuario autorizado puede introducir No Conformidades, así como planificar las correspondientes Acciones Correctivas o Preventivas, repartiendo las tareas a cada uno de los responsables, y controlando plazos y acciones realizadas.

De igual modo se pueden gestionar expedientes específicos para Planes de Formación, Planes de Auditorías o Planes de Mejora Continua, el Seguimiento y Evaluación de Proveedores o la Gestión y Control de Indicadores.

En resumen, KMKey Quality es el software de gestión de calidad (SGC) ideal para la implantación y la gestión de la calidad pero tiene como inconveniente que los servidores donde se guarda la información se encuentran en otro país por lo que no cumple con las políticas de seguridad de la universidad.



## **1.4.2 ISOSYSTEM Quality Pack(SOFTEXPERT 2008)**

Conjunto de módulos interrelacionados que dan solución integral a las necesidades de administración de la Compañía con los cuales se puede simplificar la implementación de la gestión de los Sistemas de Calidad, Gestión Completa de Proyectos y Herramientas de Análisis y Toma de Decisión en su empresa Módulos que integran este paquete: Indicadores, Procesos, Auditoría, Acciones, BI y Documentos.

Acceso al sistema 100% Web. Ambiente multiusuario con derechos de acceso configurables por usuario. Soporte e integración con sistemas de ERP. Soporte a la mayoría de los servidores de red como Windows NT/2000/XP; Netware, Linux, Unix etc. Comunicación directa con servidores de e-mail como Exchange, Lotus Notes con protocolos SMTP y MAPI para transferencia de e-mails. Compatibilidad con base de datos como: Oracle, MS SQL Server, FireBird/InterBase y DB2. Interactividad con el servicio de administración de usuarios del directorio activo de Microsoft. Compatibilidad con Windows 98, 2000, XP o superior en las estaciones de trabajo.

De forma general este sistema no presenta las funcionalidades necesarias para resolver las necesidades propias de la facultad en los procesos de auditorías y revisiones, por lo que queda descartado.

## **1.5 Herramientas de modelado de sistemas**

### **1.5.1 Visual Paradigm**

Visual Paradigm es una herramienta que sirve para el lenguaje de modelado UML y que soporta el ciclo de vida completo de desarrollo de RUP (análisis y diseño orientados a objetos, construcción, pruebas y despliegue). Tiene varias funcionalidades, permite dibujar todos los tipos de diagramas de clases, código inverso, generar código PHP desde diagramas y generar documentación. Actualmente se puede instalar tanto en Linux como en Windows. Esta herramienta tiene disponible diferentes versiones entre ellas una gratuita llamada Visual Paradigm UML Community. Facilita licencias especiales para fines académicos. Importa al Rational Rose, esta característica lo hace muy útil por la diversidad de proyectos que son modelados en el Rational.( Visual Paradigm for UML 2007)

Entre los lenguajes conocidos por Visual Paradigm para la ingeniería inversa, se encuentran los siguientes:

- Java Source.
- C++ Source.
- PHP 5.0 Source.
- XML.

### **1.5.2 Rational Rose**

Rational es actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software y sus productos están centrados en la metodología del Proceso Racional Unificado o RUP (Rational Unified Process). Entre los productos más conocidos existen: Rational Application Developer, Rational Software Architect (Posee la funcionalidad del producto antes conocido como Rational Rose) y Rational Portafolio Manager. Con esta herramienta se pueden utilizar cuatro tipos de modelos para realizar el diseño del sistema, utilizando una vista estática y otra dinámica de dichos modelos, uno lógico y otro físico. Permite crear y refinar, logrando así un modelo completo que representa el dominio del problema y el sistema de software. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades. Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa se puede obtener información sobre su diseño. Para la utilización de esta herramienta resulta algo complicado debido a que propone el empleo del proceso de desarrollo unificado, el cual es altamente complejo e incorpora algunas funcionalidades a través de otras aplicaciones pero no las integra.

### **1.5.3 Selección de la herramienta CASE para el modelado del sistema**

Después de analizar las herramientas mencionadas anteriormente se llega a la conclusión que la más adecuada es Visual Paradigm pues esta herramienta es multiplataforma, es decir, que puede ser utilizada en cualquier sistema operativo; mientras que el Rational solo se puede utilizar en Windows. Además Visual Paradigm puede soportar el ciclo de vida completo de un software, permitiendo construir aplicaciones de calidad a un menor costo y genera informes para la generación de la documentación.

## **1.6 Metodologías de desarrollo de software**

Una metodología de desarrollo establece la necesidad de considerar un diseño previo a la construcción del sistema y ofrece vías para recoger las especificaciones. Constituye un proceso para la producción organizada del software, empleando una serie de pasos, con técnicas y notaciones asociadas a cada uno. El software debe ser desarrollado con una metodología que contemple los aspectos pedagógicos y fundamentalmente hay que evaluarlo en un contexto similar al de uso.(HERNÁNDEZ 2002)

### **1.6.1 El Proceso Unificado de Desarrollo (RUP)**

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP

es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.(Jacobson 2000)

### **1.6.2 Extreme Programing (XP)**

Es la más destacada de los procesos ágiles de desarrollo de software. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos

en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.( Metodologías de desarrollo de software 2008).

Las características fundamentales del método son:

- Desarrollo iterativo e incremental, pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas, las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que calidad del código escrito de esta manera es mayor (el código es revisado y discutido mientras se escribe) esto es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario, un representante del cliente trabaja junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Obliga a realizar entregas frecuentes.

### **1.6.3 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.(Mendoza 2004).

MSF tiene las siguientes características:

- Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un lugar específico.
- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.

- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: porque 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.

- Modelo de Arquitectura del Proyecto: Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- Modelo de Equipo: Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- Modelo de Proceso: Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- Modelo de Gestión del Riesgo: Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- Modelo de Diseño del Proceso: Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- Modelo de Aplicación: Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

#### **1.6.4 Selección de la Metodología a utilizar**

Luego del estudio realizado anteriormente, donde se pudieron analizar varias metodologías y comparar sus diferentes características y particularidades de empleo en varias situaciones, se determinó cuáles no se ajustan al proceso en cuestión, en el caso de XP a pesar de su utilidad, tiene como principal objetivo

producir el software a corto plazo sin importar la documentación, está más orientada a las personas que a las herramientas y plantea que el usuario debe ser un integrante más del equipo de desarrollo. Mientras MSF, además de proponer la realización de gran cantidad de documentación no aplicable para la solución de este problema, se centra en el modelo de equipo, dejando en un segundo plano las elecciones tecnológicas.

Se seleccionó la metodología RUP como la más adecuada, ya que permite un desarrollo centrado en una arquitectura sólida. Es adaptable para proyectos a largo plazo y establece la generación de gran cantidad de artefactos los cuales serán de gran utilidad para la futura implementación del módulo.

### **1.7 Diseño de Aplicaciones Web**

Una aplicación web constituye un sistema web en el cual la acción del usuario afecta el negocio. Su arquitectura general es la de un sistema cliente/servidor donde tanto el cliente; que suelen ser estaciones de trabajo que solicitan varios servicios al servidor web y el protocolo de transferencia de hipertexto (HTTP) mediante el cual se comunican son estándares y no son creados por el desarrollador.

Comúnmente está estructurada como una aplicación en tres capas: el navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica (en este caso PHP) es la capa de en medio, y una base de datos como última capa. El navegador Web envía peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

¿Por qué desarrollar una aplicación Web?

Las aplicaciones web permiten una transmisión inmediata de contenido e información que va a permitir el desarrollo de los procesos del negocio de una mejor manera.

Entre sus ventajas se pueden mencionar las siguientes:

- Las aplicaciones web tienen la ventaja principal sobre las aplicaciones de escritorio de ser independientes de la plataforma o sistema operativo, ya que sólo utilizan el navegador de internet como medio de entrada y visualización de la información.

Distribución:

- No se instala nada en el equipo de cada usuario, solo en el servidor, lo que permite una distribución rápida a toda la red y de forma simultánea. Cualquier actualización en el servidor es visible de inmediato en todos los equipos conectados a la red y con acceso a la aplicación.
- No se necesita reiniciar el servidor después de realizar una actualización en la mayor parte de los casos.

Movilidad:

- Se puede acceder desde cualquier parte del mundo de la forma que en una Web convencional, siempre que se disponga de conexión a internet y se tengan los permisos de acceso adecuados.
- Posibilidad de hacer páginas específicas para dispositivos móviles dentro de la misma Web.
- En cuanto a su Escalabilidad, las aplicaciones Web utilizan una arquitectura de tres capas compuesta por la capa cliente, donde está únicamente el browser, la capa del Servidor Web donde está la aplicación y la capa de datos donde se encuentra la Base de Datos.

### 1.7.1 Patrones de diseño y arquitectura

Los patrones de diseño han adquirido gran popularidad entre investigadores y diseñadores de software orientado a objeto debido a que son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y aplicables a distintos problemas típicos de diseño que pueden encontrarse en diferentes contextos.

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

Los patrones son soluciones de sentido común que deberían formar parte del conocimiento de un diseñador experto. Además facilitan la comunicación entre diseñadores, pues establecen un marco de referencia.

Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución. Además, los patrones de diseño, también ayudarán a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces.

#### **Patrones de diseño de software (GRASP):**

Los patrones de asignación de responsabilidades GRASP (General Responsibility Assignment Software Patterns), describen los principios fundamentales de la asignación de responsabilidades a objetos. Constituyen el fundamento de cómo se va a diseñar el sistema finalmente. Es importante que el diseñador de software domine y aplique estos conocimientos durante la realización de un diagrama de interacción.

**Patrón Experto:** ¿Cómo asignar responsabilidades de la forma más eficiente?

Asignar una responsabilidad al experto en la información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto provee un bajo nivel de acoplamiento.

**Patrón Creador:** ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?

Las responsabilidades de crear una instancia de la **clase A** se le dará a aquella **clase B**, en los siguientes casos:

- B agrega los objetos de A
- B contiene los objetos de A
- B registra las instancias de los objetos de A.
- B tiene los datos de inicialización que serán enviados a A cuando este objeto sea creado.

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

**Patrón Controlador:** ¿Quién debería encargarse de atender un evento del sistema?

El controlador es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación. El controlador no realiza mucho trabajo por sí mismo; más bien coordina la actividad de otros objetos. Asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase que represente alguna de las siguientes opciones:

- El sistema global.
- La empresa u organización global.
- Algo activo en el mundo real que pueda participar en la tarea.
- Un manejador artificial de todos los eventos del sistema de un caso de uso (controlador de casos de uso)

**Patrón Bajo Acoplamiento:** ¿Cómo dar soporte a una mínima dependencia y a un aumento de la reutilización?

Una clase con bajo acoplamiento no depende de “muchas otras” clases. Las clases con alto acoplamiento recurren a muchas clases y no es conveniente. Son más difíciles de mantener, entender y reutilizar.

**Patrón Alta Cohesión:** Es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño.

Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.(CRAIG LARMAN 1999)

**Patrón de arquitectura**



## Modelo-Vista-Controlador (MVC)

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: el Modelo, las Vistas y los Controladores.

El MVC en una aplicación web, se representa a través de la Vista que constituyen las páginas HTML y el código que provee de datos dinámicos a la página, que permite mostrarle la información al usuario.

El Controlador que gestiona todos los eventos de entrada del usuario y es el intermediario entre el Modelo y la Vista. El Modelo contiene toda la información persistente y reglas de negocio.

El uso de este patrón es favorable; establece una separación entre los componentes de un programa permitiendo implementarlos por separado. Posibilita que si uno de los componentes tiene un mal funcionamiento puede reemplazarse sin que se vean afectados los demás. Permite que no haya ninguna dependencia directa entre la Vista y el Modelo; esto beneficia que haya un mayor soporte a los cambios en las interfaces; que tienden a cambiar con más frecuencia que la lógica del negocio. Se puede adicionar nuevas vistas sin afectar el Modelo, es decir, múltiples páginas en la aplicación web pueden emplear los mismos elementos del Modelo.( Trygve Reenskaug 1979)

### **1.8 Desarrollo de Aplicaciones Web**

En los siguientes epígrafes se abordan los diferentes lenguajes y herramientas que pueden ser utilizadas para el desarrollo de aplicaciones web. Además se realiza la selección del lenguaje o herramienta a utilizar para el desarrollo del módulo documentado en el presente trabajo. Se agrupan por los lenguajes del lado del cliente, lenguajes del lado del servidor y gestores de base de datos.

#### **1.8.1 Lenguajes del lado del Cliente**

Ejemplo de ellos son el Java Script y el Visual Basic Script. Estos lenguajes son dinámicos ya que el código se ejecuta en el navegador antes de ser enviado al servidor, su uso sin embargo se restringe principalmente a la validación de datos.

##### **1.8.1.1 HTML**

HTML, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos Esta definición se debe a que está compuesto por etiquetas que definen la estructura y el formato del documento que verá el usuario a través de la web. Esas etiquetas son leídas por el navegador o visualizador, es decir el programa que se utiliza para navegar, y que es el que ejecuta las funciones creadas en HTML permitiendo que puedan ser visibles en nuestra máquina.

En cuanto a la creación de un archivo HTML, que es un texto plano, se necesita un editor de textos que no sea complejo, como puede ser el bloc de notas. Una de las características es que, además del texto (que es el contenido en sí), este lenguaje permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes y sonidos. Actualmente existe un gran número de navegadores por lo que a veces se complica que todos puedan interpretar el código HTML de la misma manera. Por esa razón es que quienes crean las páginas, chequean que pueda ser leída por los menos por los navegadores más populares.(LANZILLOTTA. 2004).

### **1.8.1.2 Java Script**

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas Java Script van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos.(LUQUE. 2001)

Sus características más importantes son:

Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias Java Script contenidas en una página HTML y ejecutarlas adecuadamente

Es un lenguaje orientado a eventos. Cuando un usuario pincha sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante Java Script se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos.

Es un lenguaje orientado a objetos. El modelo de objetos de Java Script está reducido y simplificado, pero incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

### **1.8.1.3 Selección del (los) lenguaje (s) a utilizar**

En esta sección se expondrán las tecnologías a utilizar y se argumentarán razones de por qué se utilizarán. Primero, HTML se utiliza principalmente, por lo sencilla que se hace su programación, y además, muy importante, está concebido para que su código sea interpretado por la mayoría de los navegadores, o por lo menos los más utilizados universalmente. Segundo Java Script se utiliza para incrementar las funcionalidades de HTML, principalmente en lo que tiene que ver en la validación, además posee otras características que lo hacen muy útil, Java Script es un lenguaje interpretado, es un lenguaje orientado a eventos, es un lenguaje orientado a objetos.

## **1.8.2 Lenguajes del lado del servidor**

Las tecnologías del lado del servidor son aquellos programas o servicios que corren en un servidor remoto y que brindan funcionalidades al sistema.

### **1.8.2.1 PHP**

PHP permite la creación de aplicaciones Web muy robustas, al posibilitar la conexión a diferentes tipos de servidores de base de datos como: MySQL, Postgres, Oracle y otros. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos y puede interactuar con varios de los servidores Web más populares. Los principales usos de PHP son: la programación de páginas Web dinámicas, la programación en consola y la creación de aplicaciones gráficas independientes del navegador. La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluye al Java y Java Script, de hecho mucha de la funcionalidad del PHP se la debe al C. PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. Tiene diferentes interfaces para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTTPD35.(VÁZQUEZ 2007)

### **1.8.2.2 ASP**

Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). ASP tiene muchas ventajas(ULTRASIST 2008):

- La mayor es que reemplaza la forma tradicional de intercambiar información entre usuarios.
- Active Server Pages permiten que el código sea incrustado en un documento HTML y que corra en el servidor.
- ASP es compatible con múltiples plataformas.

Se han expuesto algunas de las ventajas de ASP, pero para utilizar esta tecnología del lado del servidor, se debe contar con servidores de bases de datos propietarios de Microsoft, como el Internet Information Server. Por lo que no se puede crear una aplicación libre usando ASP.

### **1.8.2.3 Selección del (los) lenguaje (s) a utilizar**

Como lenguaje de programación se propone el uso de PHP por las ventajas y funcionalidades que presenta para la creación de los servicios web, pues es soportado en una gran cantidad de servidores, plataformas y gestores de bases de datos, posibilitando y asegurando una migración futura por parte de los usuarios, teniendo como punto de partida la posibilidad de utilizar la capa de abstracción de datos

ADOdb, y su programación del lado del servidor, además de profundizar en las magnificas ventajas de su integración con el Apache como servidor a utilizar y el MySQL como manejador de base de datos. Tiene un gran parecido con los lenguajes estructurados como C y Perl, es compatible con aplicaciones de contenido dinámico sin tener que llegar a aprenderse nuevas funciones. Es de libre acceso a todos, contando con una gran cantidad de información relacionada y de códigos de ejemplos.

### **1.8.3 Gestores de Bases de Datos**

Es un conjunto de herramientas que proporciona a administradores, analistas, programadores, usuarios los medios necesarios para describir, recuperar y manipular los datos almacenados en la BD, manteniendo la seguridad, integridad y confidencialidad de los mismos.

#### **1.8.3.1 MySQL**

Uno de los Sistemas Gestores de Base de Datos que se presentan es MySQL. El mismo cuenta con más de seis millones de instalaciones, es multihilo, multiusuario, rápido y muy robusto. Es software libre, publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySQL AB. MySQL es muy utilizado en los sistemas Web. Tiene gran popularidad por estar grandemente ligado a PHP. Es capaz de funcionar 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.(AB. 2007)

#### **1.8.3.2 PostgreSQL**

Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Soporta tanto la programación orientada a objetos como la funcional. Las funciones tienen la particularidad de poder ejecutarse con los privilegios del usuario ejecutor o con los privilegios de un usuario definido previamente lo que supone un alto nivel de seguridad. Estas funciones son referidas en otros SGBD como procedimientos almacenados (stored procedures). PostgreSQL tiene la extraordinaria potencialidad de permitir que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos esto es posible gracias a un sistema denominado Acceso Concurrente Multiversión (MVCC).(TIENDALINUX.COM 2007)

#### **1.8.3.3 Selección del Gestor de BD a utilizar**

Aquí conviene resaltar que la elección de uno u otro SGBD vendrá dado por las necesidades específicas de la Web que se quiere implementar. A fecha de hoy, la versión libre de InterBase aún ofrece un rendimiento pobre en entornos con gran número de usuarios concurrentes por lo que las opciones principales son MySQL y PostgreSQL. Como sistema de gestión de base de datos, se seleccionó

PostgreSQL en su versión 8.1. Es software libre, multiplataforma y tiene todas las características de los SGBD modernos.

Ofrece además una potencia sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: clases, herencia, tipos y funciones. Otras características aportan potencia y flexibilidad adicional como son las restricciones (constraints), disparadores (triggers), reglas (rules) e integridad transaccional.

### **1.9 Conclusiones del capítulo**

En este capítulo se analizan y fundamentan los conceptos, tecnologías y metodología que son utilizadas para el desarrollo del sistema propuesto. Después de este análisis y la fundamentación realizada se definió RUP (Proceso de desarrollo unificado) como metodología de desarrollo, el lenguaje de programación que se utilizará es PHP y JavaScript para el control de las diferentes funciones a realizar en el cliente, como gestor de base de datos se estableció el PostgreSQL. Como herramienta CASE se usará el Visual Paradigm, y como lenguaje de modelado UML.

## **Capítulo 2. Características del sistema**

### **2.1 Introducción**

En el presente capítulo se describe la propuesta de solución a la situación problemática. Para ello se detallan los procesos del negocio mediante Diagramas de Casos de Uso del Negocio con sus correspondientes descripciones, Diagramas de Actividades y Diagramas de Clases del Modelo de Objetos.

Por otra parte, el capítulo brinda una concepción general del sistema propuesto, se exponen las funcionalidades y características, se identifican los actores que interactúan con el mismo y se presentan los Diagramas de Casos de Uso del Sistema con las correspondientes descripciones de los casos de uso.

### **2.2 Objeto de estudio**

#### **2.2.1 Problema y situación problemática**

El proyecto de calidad de la facultad está dividido en varios grupos de trabajo, entre ellos el grupo de control, que tiene la responsabilidad de realizar revisiones y auditorías a sus proyectos productivos.

El grupo de control cuenta con un equipo de 15 personas, agrupadas en 3 equipos. Un equipo se encarga de desarrollar las herramientas de apoyo como la elaboración de las entrevistas y encuestas a realizar. El resto de los equipos se especializarán en hacer revisiones y auditorías a los proyectos por polos, uno para el software educativo y otro para software de gestión y deporte.

Existe un gran volumen de información sobre cómo ejecutar el control de Calidad durante la producción de software, por lo que estos procesos se pueden realizar de diferentes formas en distintos proyectos. Generalmente el grupo de Calidad de la Facultad omite algunas actividades durante el desarrollo de las revisiones o las auditorías, o se ve en la necesidad de disminuir el nivel de rigurosidad que se requiere debido a la escasez de tiempo y personal. Por otra parte, es muy difícil lograr controlar la información obtenida a través de estas actividades cuando se han aplicado a una cantidad considerable de proyectos, por lo que muchas veces no es posible conocer el estado real de un proyecto o compararlo con otros proyectos de la facultad. Como existen diferentes formas de realizar estas actividades, los revisores y auditores pueden trabajar de manera separada con resultados difíciles de fusionar, debido a que no existe una forma fácil, intuitiva y eficiente de obtener una asesoría o recibir determinadas orientaciones durante la revisión. A pesar de que existe mucha información al respecto, los especialistas pueden sentirse un tanto "perdidos" durante la ejecución de estas actividades.

### **2.2.2 Objeto de automatización**

Los procesos que serán objeto de automatización en este trabajo son básicamente los relacionados con las auditorías y revisiones, dichos procesos incluyen actividades tales como:

- Insertar y manejar toda la información necesaria para llevar a cabo las actividades de control
- Almacenar toda la información que se genera a partir de las auditorías o las revisiones
- Insertar y asignar tareas a los miembros de los equipos de revisores o auditores.
- Tener un registro con las evaluaciones de cada uno de los miembros de los equipos de control.

### **2.2.3 Información que se maneja**

El sistema manejará información privada sobre los proyectos de la facultad. Dicha información servirá de base para medir la calidad con que se desarrollan los proyectos y facilitará la comparación y retroalimentación de las acciones ejecutadas. La información debe estar protegida de accesos no autorizados y a la vez debe estar disponible en un formato intuitivo para que pueda ser utilizada de manera efectiva. Básicamente se manejarán No Conformidades y Hallazgos encontrados en las revisiones y auditorías, así como información referente al trabajo del equipo de control de la calidad que servirá para orientar y evaluar a los revisores y auditores.

### **2.2.4 Propuesta de sistema**

Se propone un Sistema para la Gestión de la Calidad compuesto por 5 módulos:

- Administración
- Planificación
- Pruebas
- Auditorías y revisiones
- Capacitación

Específicamente en este trabajo se propone el análisis y diseño del módulo Auditorías y Revisiones. El sistema será una aplicación Web disponible desde cualquier estación de trabajo, permitirá manejar toda la información que se obtiene de las actividades de Revisiones y Auditorías.

## **2.3 Modelo de negocio**

El modelo de negocio es un artefacto de la disciplina de ingeniería del software que tiene como objetivo conocer la estructura y dinámica de la organización en el cual se va a implantar el sistema, comprender los problemas actuales e identificar las posibles mejoras. Además asegurar que el equipo de proyecto,

los usuarios finales, tengan un entendimiento común de la organización; brinda una visión de qué es lo necesario para satisfacer las exigencias de los usuarios.

#### **2.4 Reglas del Negocio a considerar**

- Un auditor solo puede realizar las tareas que le fueron asignadas.
- La información que se maneja en las actividades de control es clasificada como limitada.
- El auditor líder puede realizar cualquier tarea relacionada con la actividad de control durante la ejecución de la misma.
- Cuando un auditor o un revisor no realiza completamente las tareas que le fueron asignadas se considera que este incumplió con las mismas.
- Una vez que el proyecto a auditar es notificado el líder del mismo tiene 72 horas para no aceptar la auditoría, en ese caso debe enviar una justificación.
- El auditado debe entregar la información solicitada al equipo auditor el día planificado para el inicio de la auditoría. Cuando esta entrega no se cumple en el día estipulado se evalúa de mal al auditado y la auditoría es abortada.
- El único autorizado a publicar todo aquello que se genere de la auditoría es el jefe de la actividad o el Asesor de Calidad.
- El Informe final es clasificado como limitado además no se entrega digital.

#### **2.5 Actores y trabajadores del negocio**

Es cualquier individuo, organización, grupo o máquina que interactúa con el negocio, son los que se benefician con los procesos del negocio. A continuación se describen los actores del negocio que se aborda en el presente trabajo.(Ingeniería de Software I 2008).

**Tabla 1 – Descripción de los actores del negocio**

<b>Actor</b>	<b>Descripción</b>
<b>Asesor de calidad</b>	Es el líder del grupo de calidad de la facultad y es el que solicita realizar una actividad de



	control definiendo los parámetros de la misma.
--	--

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. (Ingeniería de Software I 2008). Los trabajadores del negocio del proceso que aborda este trabajo se describen a continuación:

**Tabla 2 - Descripción de los trabajadores del negocio**

<b>Trabajador</b>	<b>Descripción</b>
<b>Auditor Líder</b>	Persona al frente del equipo auditor.
<b>Revisor Líder</b>	Máximo responsable del Equipo de revisores que lleva a cabo la revisión.
<b>Auditor</b>	Dos o más auditores que llevan a cabo una auditoría con el apoyo de expertos técnicos. Al menos está formado por el auditor líder y un auditor.
<b>Revisor</b>	Tres o más revisores que llevan a cabo una revisión con el apoyo de expertos temáticos.

## **2.6 Diagrama de casos de uso del negocio**

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del mismo. (Ingeniería de Software I 2008).

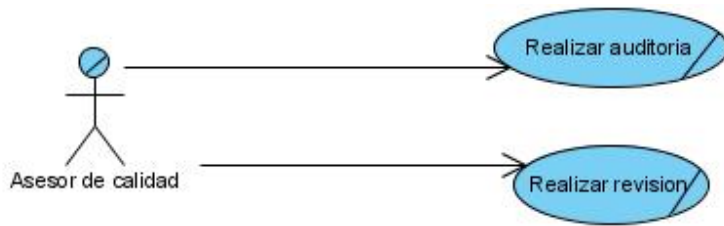


Fig. 1 - Diagrama de casos de uso del negocio

## 2.7 Descripciones textuales CUN

Tabla 3 - Descripción textual del caso de uso “Realizar auditoría”

Caso de uso del negocio	Realizar auditoría
Actores	Asesor de calidad
Resumen	El caso de uso se inicia cuando el asesor de calidad de la facultad solicita realizar una auditoría a un proyecto productivo, para lo cual se cuenta con un equipo de auditores encargados de realizar esta tarea. Durante la auditoría se evaluará el proyecto, el proceso productivo, el producto, etc. Obteniéndose finalmente una evaluación del proyecto.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
1. Define el plan, alcance, objetivos y criterios de auditoría.	2. Una vez definido el plan, alcance objetivos y los criterios de la auditoría el auditor líder se encarga de preparar los recursos asignados así como de la capacitación del equipo auditor.
	3. El equipo de auditores definen los métodos y técnicas de recopilación de información que van a utilizar durante el proceso de auditoría y preparan los documentos necesarios.
	4. Se realiza la reunión de apertura donde participan el equipo auditor en conjunto con el

	proyecto auditado.
	5. Se realizan las actividades de auditoría, revisión de documentos, entrevistas y observaciones.
	6. A partir de estas actividades se generan los hallazgos de la auditoría.
	7. Se elabora el informe preliminar así como las acciones correctivas o de mejora.
	8. Se realiza la reunión de cierre.
	9. El proyecto acepta la evaluación de la auditoría.
	10. El equipo auditor realiza el seguimiento del proyecto hasta que se resuelvan las no conformidades detectadas.
	11. El auditor líder redacta el informe final.
	12. El auditor líder evalúa el desempeño de cada uno de los miembros del equipo.
<b>Flujos alternos</b>	
9. a El proyecto no acepta la evaluación de la auditoría.	
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>
	9. a.1 El proyecto realiza una reclamación formal a la dirección de calidad de la facultad.
9.a.2 Se acepta la reclamación	
9. a.3 Se planifica otra auditoría al proyecto.	

**Tabla 4 - Descripción textual del caso de uso “Realizar revisión”**

<b>Caso de uso del negocio</b>	<b>Realizar revisión</b>
Actores	Asesor de calidad
Resumen	El caso de uso se inicia cuando el asesor de calidad de la facultad solicita realizar una revisión a un proyecto productivo, para lo cual se cuenta con un equipo de revisores encargados de realizar esta tarea. Durante la revisión se evaluará el proceso productivo y el producto. Permitiendo detectar errores en los entregables del proyecto.
Casos de uso asociados	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1. Define el plan, alcance, objetivos y guía de evaluación de la revisión.	2. Una vez definido el plan, alcance objetivos y la guía de evaluación de la revisión el revisor líder se encarga de preparar los recursos asignados así como de la capacitación del equipo revisor.
	3. El equipo de revisores definen los métodos y técnicas de recopilación de información que van a utilizar durante el proceso de revisión y preparan los documentos necesarios.
	4. Se realiza la reunión de apertura donde participan el equipo revisor en conjunto con el proyecto.
	5. Se realizan las actividades de revisión,

	revisión de documentos.
	6. A partir de la revisión de los documentos del proyecto se generan los hallazgos de la revisión.
	7. El revisor líder redacta el informe final.
	8. Se realiza la reunión de cierre.
	9. El revisor líder evalúa el desempeño de cada uno de los miembros del equipo.
<b>Flujos alternos</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>

## 2.8 Diagramas de actividades

Los casos de uso del negocio tienen secuencias de actividades que en conjunto, producen algo para el actor del negocio. La estructura del flujo de actividades se describe gráficamente con la ayuda de un diagrama de actividades, este describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Como parte importante dentro de estos diagramas se definen las actividades a automatizar, identificadas con el color más claro. (Ingeniería de Software I 2008).

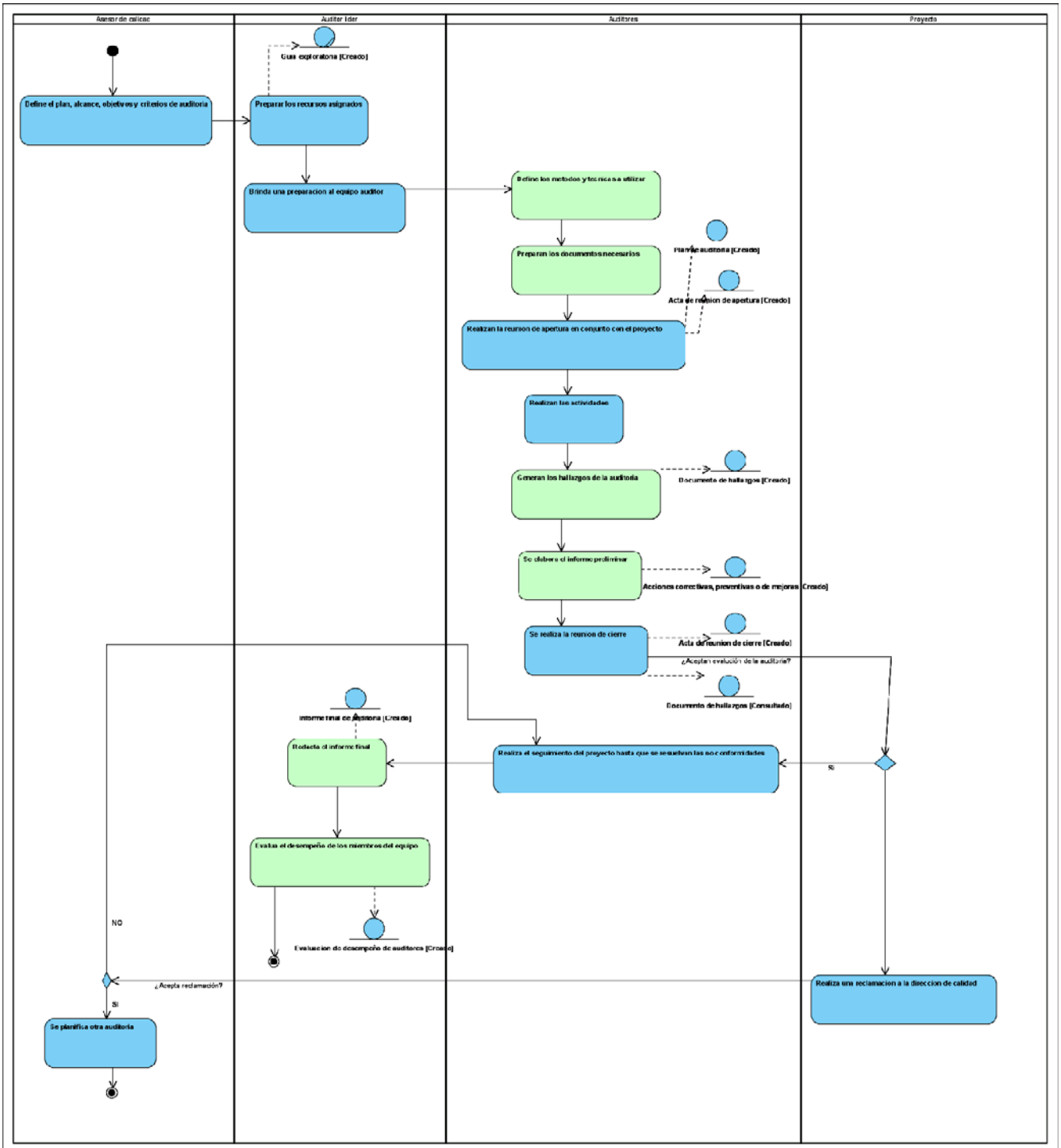


Fig. 2 - Diagrama de actividades "Realizar auditoría"

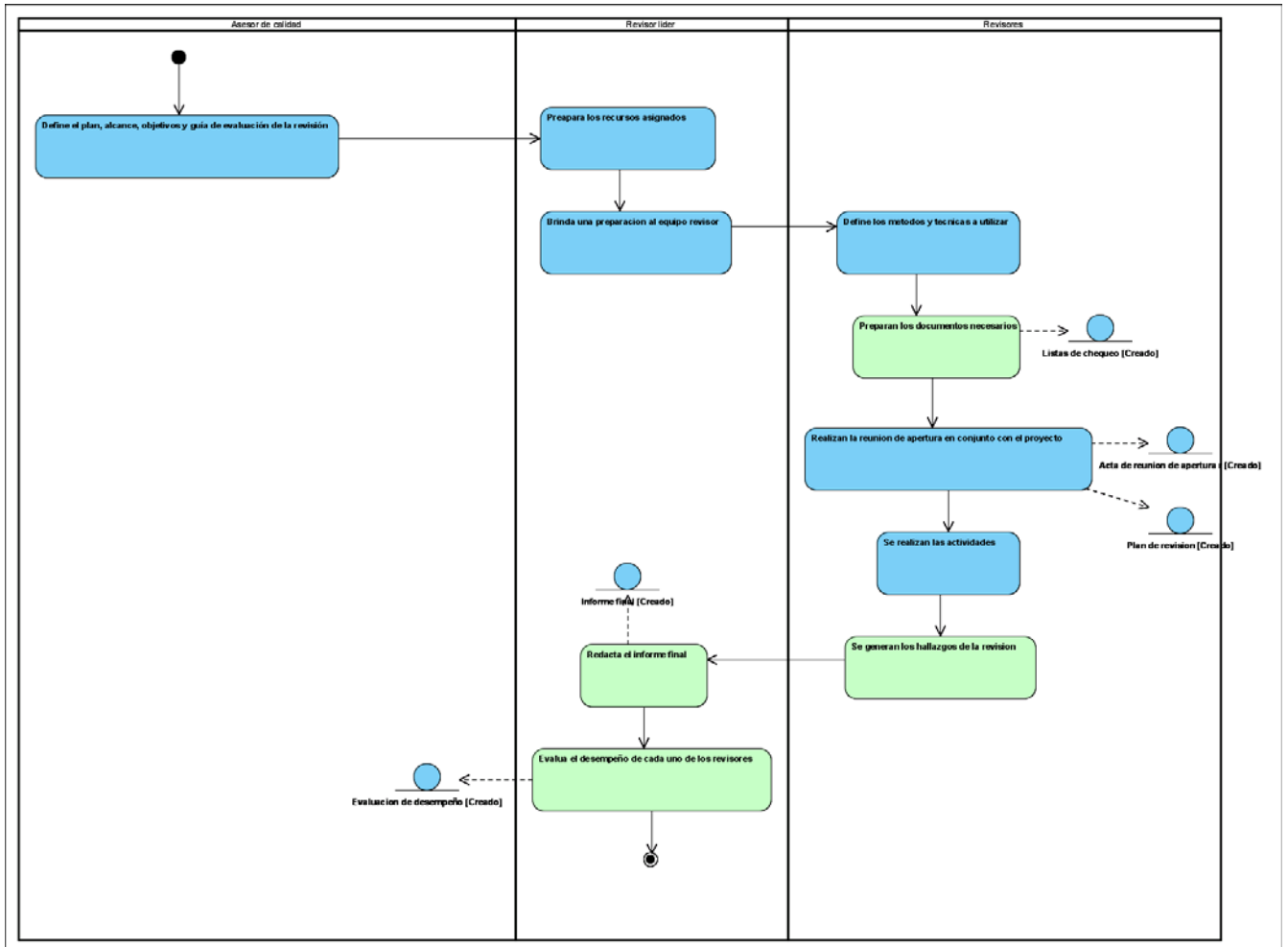
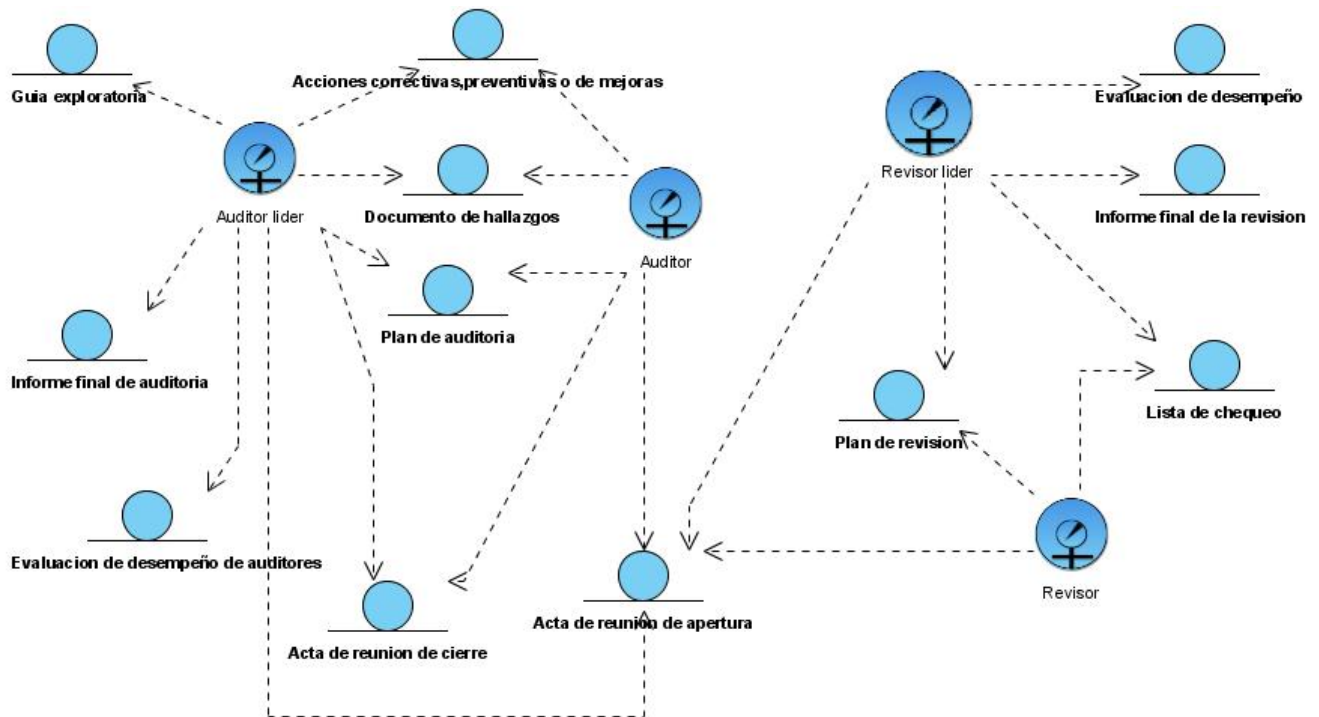


Fig. 3 - Diagrama de actividades "Realizar revisión"

## 2.9 Diagrama de clases del modelo de objetos

Este diagrama describe cómo colaboran los trabajadores y las entidades del negocio, y proporciona un acercamiento a la identificación de los futuros actores y entidades del sistema. (Ingeniería de Software I 2008).



**Fig. 4 - Modelo de objetos del negocio**

## **2.10 Especificación de los requisitos del sistema**

El flujo de trabajo de requisitos tiene como propósito fundamental guiar el desarrollo hacia el sistema correcto. Ello se puede obtener utilizando la descripción de los requisitos del sistema, o sea, condiciones y/o capacidades que el sistema debe cumplir. Las descripciones tienen que ser lo suficientemente buenas como para que se pueda llegar a acuerdos entre los clientes (entre ellos, los futuros usuarios) y los desarrolladores en cuanto a lo que debe y no debe hacer el sistema. (Ingeniería de Software I 2008).

### **2.10.1 Requerimientos funcionales**

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa. (Ingeniería de Software I 2008).

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

1. Insertar Actividad de Control
  - 1.1. Puede ser una Revisión o una Auditoría
  - 1.2. Los datos que se registran de una revisión son: Proyecto a revisar, Líder de proyecto, Asesor de calidad del proyecto, revisor líder, equipo de revisores, fecha de revisión.
  - 1.3. Los datos que se registran de una auditoría son: Proyecto a auditar, Líder de proyecto, Asesor de calidad del proyecto, auditor líder, equipo de auditores, fecha de auditoría.



2. Modificar Actividad de Control
3. Eliminar Actividad de Control
4. Insertar tarea a una Actividad de Control
  - 4.1. Los datos que se registran de una tarea son: Nombre, Responsable, fecha de cumplimiento.
  - 4.2. Una tarea puede tener asociados uno o varios recursos
5. Modificar tarea de una Actividad de Control
6. Eliminar tarea de una Actividad de Control
7. Asignar responsabilidades a los miembros del equipo de Calidad
  - 7.1 Consiste en determinar las tareas que ejecutará cada auditor o revisor
8. Ver lineamientos de calidad
9. Insertar no conformidades de una actividad de control
  - 9.1 Los datos que se registran de una no conformidad son: Lugar donde se encontró, lineamiento que se incumple, Descripción de la no conformidad, Impacto.
10. Modificar no conformidades de una actividad de control
11. Eliminar no conformidades de una actividad de control
12. Consultar no conformidades de una actividad de control
  
13. Insertar datos al Plan de actividad de control
14. Modificar datos al Plan de actividad de control
15. Imprimir/Exportar a pdf Plan de actividad de control
  
16. Insertar datos a la Minuta de reunión
17. Modificar datos a la Minuta de reunión
18. Imprimir/Exportar a pdf Minuta de reunión
  
19. Generar Documento de no conformidades
20. Modificar datos al Documento de no conformidades
21. Imprimir/Exportar a pdf Documento de no conformidades

22. Insertar Acciones correctivas, preventivas o de mejora
23. Modificar datos a las Acciones correctivas, preventivas o de mejora
24. Imprimir/Exportar a pdf Acciones correctivas, preventivas o de mejora
  
25. Insertar datos al Informe final de actividad de control
26. Modificar datos al Informe final de actividad de control
27. Imprimir/Exportar a pdf Informe final de actividad de control
28. Insertar Lista de recomendaciones
29. Modificar datos a la Lista de recomendaciones
30. Imprimir/Exportar a pdf Lista de recomendaciones
  
31. Insertar evaluación de un miembro del equipo de control
32. Modificar evaluación de un miembro del equipo de control
33. Eliminar evaluación de un miembro del equipo de control
  
34. Insertar Lista de Chequeo
35. Modificar Lista de Chequeo
36. Eliminar Lista de Chequeo
37. Insertar preguntas a una Lista de Chequeo
38. Modificar preguntas de una Lista de Chequeo
39. Eliminar preguntas de una Lista de Chequeo
40. Imprimir/exportar a pdf una Lista de Chequeo
  
41. Emitir reportes
  - 41.1. Actividades de Control por proyecto
    - 41.1.1. Debe mostrarse la evaluación de cada Actividad
    - 41.1.2. Debe permitir la posibilidad de mostrar la lista de señalamientos

- 41.2. Proyectos con mayor cantidad de señalamientos en la Facultad
  - 41.2.1. Se seleccionarán los 5 proyectos críticos que tuvieron mayor cantidad de señalamientos en las actividades de control aplicadas en el semestre.
- 41.3. Mejores proyectos de la Facultad
  - 41.3.1. Se seleccionarán los 3 proyectos con mejores resultados en las actividades de control aplicadas en el semestre.
- 41.4. Carga del equipo de Control
  - 41.4.1. Se muestra la lista de los miembros con las actividades en curso que tienen asignadas.
- 41.5. Desempeño del equipo de Control
  - 41.5.1. Se muestra la lista de los miembros con todas las actividades que se les han asignado y la evaluación que obtuvieron.
  - 41.5.2. Deben mostrarse las diferentes evaluaciones parciales.
  - 41.5.3. Debe permitir mostrar los señalamientos que se les ha realizado en las diferentes evaluaciones.

### **2.10.2 Requerimientos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros. (Ingeniería de Software I 2008).

#### **Apariencia o interfaz externa.**

El sistema debe poseer una interfaz web amigable y sencilla, fácil para la interacción del usuario. Con colores en su interfaz que distingan a la facultad, sin saturación de colores ni imágenes.

#### **Usabilidad.**

La aplicación debe cumplir con los principales principios de usabilidad, debe brindarse comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación, así como será fácil de usar para cualquier usuario del sistema con los conocimientos básicos en informática y en el uso de

aplicaciones web. La navegabilidad debe no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por los usuarios del sistema de acuerdo a nivel de acceso que este posea.

### **Rendimiento.**

- Operaciones normales en dos hilos
- Respuestas rápidas en segundos al usuario.
- Debe estar disponible las 24 horas del día.

### **Soporte.**

Se le debe dar un mantenimiento periódico a la aplicación y a la Base de Datos.

### **Portabilidad.**

El sistema debe ser Multiplataforma. Haciendo énfasis en Linux.

### **Seguridad.**

Confiabilidad: La información manejada por el sistema está protegida de acceso no autorizado, y se especifican los diferentes roles existentes para establecer el acceso de los usuarios a determinadas acciones.

Integridad: La información será protegida contra corrupción y estados inconsistentes. Disponibilidad: Los usuarios del sistema podrán acceder a la información cada vez que lo necesiten.

### **Políticos-culturales.**

La herramienta sólo podrá ser utilizada dentro de la UCI.

Las funcionalidades del sistema no deben contener palabras en otros idiomas.

Se deben poner las palabras lo más acorde a su entendimiento.

### **Legales.**

Es un deber proteger la información por parte de las personas que tienen derecho de administración u otros que pongan en peligro la integridad y seguridad del sistema.

### **Confiabilidad.**

Está prohibida la diseminación pública de la información por cualquier usuario a cualquier nivel.

### **Interfaz.**

- Las interfaces deben ser lo más intuitivas posible.
- Todas las interfaces deben tener una forma de cerrarse o retornar al paso anterior.
- Ninguna interfaz puede ir cargada de muchos colores evitando la desconcentración del usuario.

## **Ayuda y documentación en línea.**

La aplicación debe que llevar un manual de ayuda.

### **2.11 Modelación del sistema**

El modelado del sistema representa la funcionalidad completa de un sistema mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores y los casos de uso dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. (Ingeniería de Software I 2008). Esos conjuntos de funcionalidades son representados por los diferentes diagramas que darán solución a la aplicación.

### **2.12 Actores del sistema**

Estos no son más que trabajadores que participan directamente en el negocio y/o nuevos actores que surgen para ejecutar funcionalidades que aparecen debido a la presencia de requisitos especiales.

**Tabla 5 - Descripción de los actores del sistema**

<b>Actor</b>	<b>Descripción</b>
<b>Usuario</b>	Este actor puede ser revisor líder o un auditor líder. Es el encargado de gestionar las plantillas de las auditorías y las revisiones.
<b>usuarioAR</b>	Es el encargado de gestionar las no conformidades de una actividad de control. Puede ser un auditor o un revisor.
<b>Jefe del grupo de control</b>	Es el encargado de gestionar todas las actividades de control, así como los resultados de estas.
<b>Especialista en técnicas de recopilación de información</b>	Gestiona la lista de chequeo

### 2.13 Diagrama de casos de uso del sistema

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. (Ingeniería de Software I 2008).

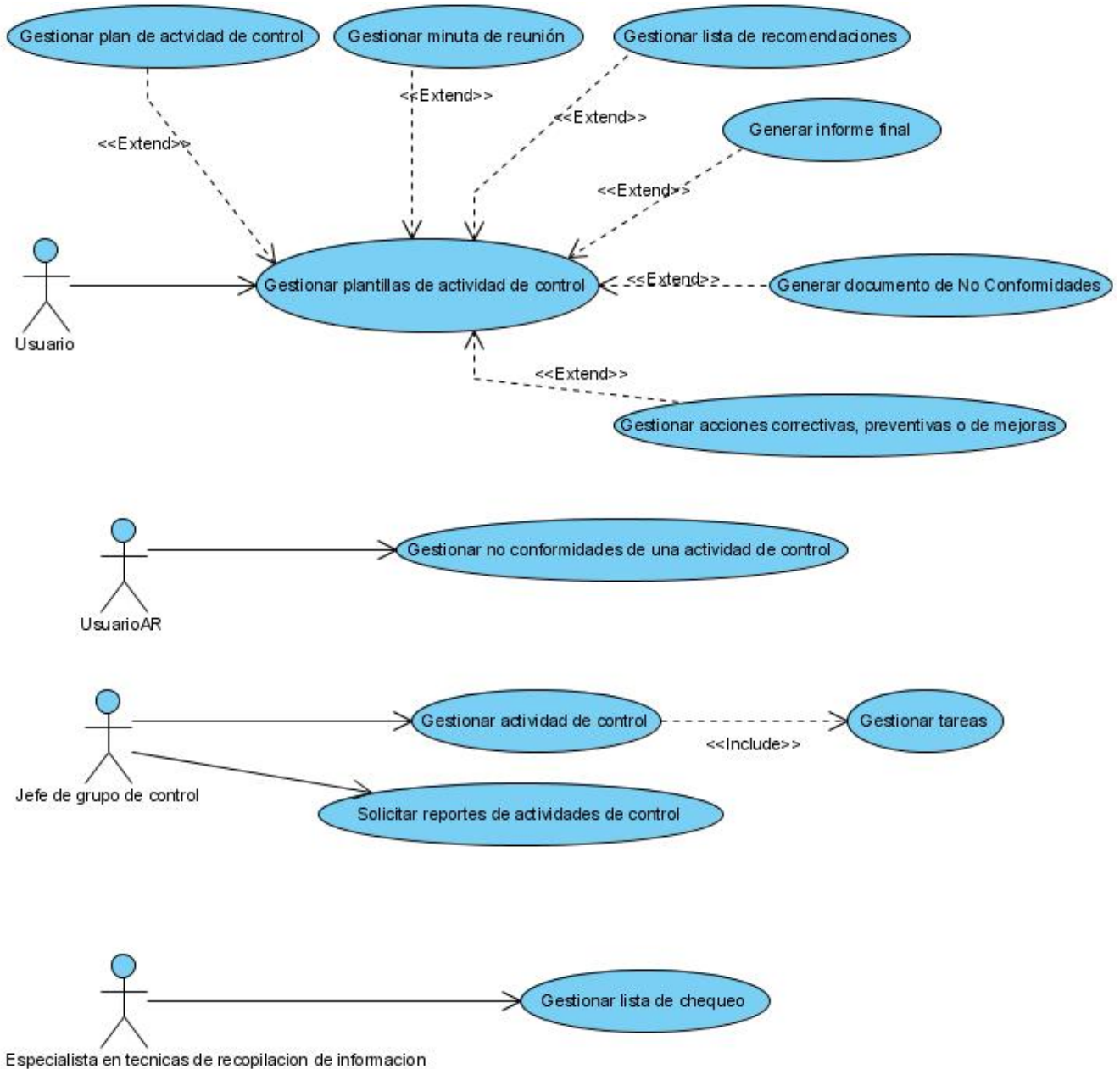


Fig. 5 - Diagrama de casos de uso del sistema

#### 2.13.1 Descripción textual de los casos de uso del sistema

Caso de Uso:	Gestionar actividad de control
--------------	--------------------------------

<b>Actores:</b>	<b>Jefe del grupo de control</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita crear, modificar o eliminar una nueva actividad de control (auditoría o revisión).
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 1,2,3</b>
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar nueva actividad de control</li> <li>• Modificar actividad de control</li> <li>• Eliminar actividad de control</li> </ul>

### *Sección "Insertar nueva actividad de control"*

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Insertar nueva actividad de control	4- El sistema muestra dos opciones Auditoría Revisión
5- El actor selecciona la opción auditoría	6- El sistema muestra una interfaz con los campos que debe llenar para insertar una auditoría.
7- El actor introduce los datos que requiere el formulario	8- Guarda los datos
	9- Termina el caso de uso

<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
5.1- El actor selecciona la opción revisión	6.1- El sistema muestra una interfaz con los campos que debe llenar para insertar una auditoría.
7.1- El actor introduce los datos que requiere el formulario	8.1- Guarda los datos
	9.1- Termina el caso de uso
<b>Sección “Modificar actividad de control”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor desea modificar una actividad de control. Selecciona el proyecto al que se le está realizando la actividad de control que desea modificar, especifica el tipo de actividad y selecciona el nombre de la actividad y presiona el botón modificar.	4- Muestra una interfaz con los datos de la actividad
5- Modifica los datos que desee de la actividad	6- Registra los campos modificados
	7- El caso de uso termina
<b>Sección “Eliminar actividad de control”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
3- El actor desea eliminar una actividad de control. Selecciona el proyecto al que se le está realizando la actividad de control que desea eliminar, especifica el tipo de actividad y selecciona el nombre de la actividad y presiona el botón eliminar.	4- Elimina la actividad seleccionada de la base de datos
	5- El caso de uso termina



<b>Caso de Uso:</b>	<b>Gestionar lista de chequeo</b>
<b>Actores:</b>	<b>Especialista en técnicas de recopilación de información</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita insertar, modificar o eliminar una lista de chequeo, o cuando se necesita insertar, modificar o eliminar preguntas a una lista de chequeo. También para imprimir o exportar dicha lista de chequeo.
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF del 34 al 40</b>
<b>Prioridad</b>	Básico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar pregunta a la lista de chequeo</li> <li>• Modificar lista de chequeo</li> <li>• Eliminar pregunta de la lista de chequeo</li> </ul>

#### *Sección "Insertar pregunta a la lista de chequeo"*

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Insertar pregunta a la lista de chequeo lista de chequeo	4- El sistema muestra una interfaz con los campos que debe llenar para insertar una pregunta a la lista de chequeo.
5- El actor introduce los datos que requiere el formulario	8- Guarda los datos

	9- Termina el caso de uso
<b>Sección “Modificar lista de chequeo”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El actor solicita modificar lista de chequeo	2- Realiza una consulta a la base de datos, y solicita la lista de chequeo existente.
	3- Muestra una interfaz con la lista de chequeo existente
4- Selecciona la pregunta que desea modificar de la lista de chequeo	5- Muestra una interfaz con los datos de la lista de chequeo
6- Modifica los datos que desee de la lista de chequeo	7- Registra los cambios realizados
	8- El caso de uso termina
<b>Sección “Eliminar pregunta de la lista de chequeo”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
1- Selecciona la opción eliminar pregunta de la lista de chequeo	2- Realiza una consulta a la base de datos, y solicita la lista de chequeo existente.
	3- Muestra una interfaz con todas la lista de chequeo existente
4- Selecciona la pregunta que desea eliminar de la lista de chequeo	5- Elimina la pregunta seleccionada de la base de datos
	6- El caso de uso termina

<b>Caso de Uso:</b>	<b>Solicitar reportes de actividades de control</b>
<b>Actores:</b>	<b>Jefe de grupo de control</b>

<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita mostrar reportes con datos de las actividades de control realizadas.
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 41</b>
<b>Prioridad</b>	<b>Básico</b>

### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Actividades de control por proyecto</li> <li>• Proyectos con mayor cantidad de señalamientos</li> <li>• Mejores proyectos de la facultad</li> <li>• Desempeño del equipo de control</li> </ul>

### *Sección “Actividades de control por proyecto”*

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción ver actividades de control de proyecto	4- Muestra una interfaz para que el usuario escoja el proyecto.
5- Selecciona el proyecto	6- Muestra las actividades de control existentes con la evaluación obtenida en cada una.
6- Selecciona la actividad de control que desea ver	7- Muestra una interfaz con todos los señalamientos encontrados en esa actividad
	8- Termina el caso de uso

**Sección “Proyectos con mayor cantidad de señalamientos”**

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor selecciona la opción ver proyectos con mayor cantidad de señalamientos	4- Solicita la cantidad de señalamiento de cada proyecto.
	5- Muestra una interfaz con los cinco proyectos con mayor cantidad de señalamientos
	6- El caso de uso termina

**Sección “Mejores proyectos de la facultad”**

<b>Acción del actor</b>	<b>Acción del sistema</b>
3- El actor selecciona la opción ver mejores proyectos de la facultad	4- Solicita la cantidad de señalamiento de cada proyecto.
	5- Muestra una interfaz con los cinco proyectos con menor cantidad de señalamientos
	6- El caso de uso termina

**Sección “Desempeño del equipo de control”**

<b>Acción del actor</b>	<b>Acción del sistema</b>
3- El actor selecciona la opción ver desempeño del equipo de control	4- Solicita la tareas asignadas a cada miembro del equipo de control así como la evaluación obtenida en cada tarea
	5- Muestra una interfaz con las tareas realizadas por cada miembro del equipo auditor y la evaluación obtenida
	6- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar no conformidades de una actividad de control</b>
---------------------	---

<b>Actores:</b>	<b>Auditor, Revisor</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita insertar, modificar o eliminar una no conformidad encontrado en una actividad de control.
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF del 9 al 12</b>
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar no conformidad</li> <li>• Modificar no conformidad</li> <li>• Eliminar no conformidad</li> </ul>

### *Sección "Insertar no conformidad"*

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Insertar no conformidad	4- El sistema muestra una interfaz con los campos que debe llenar para insertar una no conformidad.
5- El actor introduce los datos que requiere el formulario	6- Guarda los datos
	7- Termina el caso de uso

### *Sección "Modificar no conformidad"*

Acción del actor	Respuesta del sistema
------------------	-----------------------

3- El actor solicita modificar las no conformidades	4- Solicita las no conformidades existentes dentro de una actividad de control.
	5- Muestra una interfaz con todos las no conformidades existentes
6- Selecciona la no conformidad que desea modificar	7- Muestra una interfaz con los datos de la no conformidad que desea modificar
8- Modifica los datos	9- Actualiza los cambios realizados
	10- El caso de uso termina

### **Sección “Eliminar no conformidad”**

<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción eliminar no conformidad	4- Solicita las no conformidades existentes dentro de una actividad de control.
	5- Muestra una interfaz con todas las no conformidades.
6- Selecciona la no conformidad que desea eliminar	7- Elimina la no conformidad seleccionada
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar plan de una actividad de control</b>
<b>Actores:</b>	<b>Revisor líder, Auditor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita gestionar los datos de un plan de una actividad de control.
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 13,14,15</b>

<b>Prioridad</b>	Básico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar plan</li> <li>• Modificar plan</li> <li>• Exportar plan</li> </ul>
<b>Sección "Insertar plan"</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
3- El actor selecciona la opción insertar plan	4- Muestra una interfaz con un formulario.
5- Llena los campos del formulario	6- Guarda los datos del plan.
	7- Termina el caso de uso
<b>Sección "Modificar Plan"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor solicita modificar los datos del plan	4- Solicita los datos del plan de la actividad de control.
	5- Muestra una interfaz con los datos de esta
6- Modifica los datos que desee del plan	7- Actualiza los cambios realizados
	8- El caso de uso termina
<b>Sección "Exportar plan"</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar plan	4- Solicita los datos del plan de la actividad de

	control.
	5- Muestra una interfaz con los datos y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar la plantilla plan de actividad de control	7- Exporta la plantilla
	8- El caso de uso termina
Flujos alternos	
Acción del actor	Acción del sistema
6- cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar minuta de reunión</b>
<b>Actores:</b>	<b>Revisor líder, Auditor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita gestionar la plantilla minuta de reunión
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 16,17,18</b>
<b>Prioridad</b>	Básico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar.



- Insertar minuta de reunión
- Modificar minuta de reunión
- Exportar minuta de reunión

**Sección “Insertar minuta de reunión ”**

<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
3- El actor selecciona la opción insertar minuta de reunión	4- Muestra una interfaz con un formulario.
5- Llena el formulario	5- Guarda los datos
	6- Termina el caso de uso

**Sección “Modificar minuta de reunión ”**

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor solicita modificar los datos del acta minuta de reunión	4- Solicita la plantilla minuta de reunión
	5- Muestra una interfaz con los datos de esta
6- Modifica los datos que desee de la minuta de reunión	7- Actualiza los cambios realizados
	8- El caso de uso termina

**Sección “Exportar minuta de reunión”**

<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar minuta de reunión	4- Solicita la plantilla minuta de reunión
	5- Muestra una interfaz con los datos de esta plantilla y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar la plantilla minuta de	7- Exporta la plantilla minuta de reunión

reunión	
	8- El caso de uso termina
Flujos alternos	
Acción del actor	Acción del sistema
6- Cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar lista de recomendaciones</b>
<b>Actores:</b>	<b>Revisor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita generar la plantilla lista de recomendaciones
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 28,29,30</b>
<b>Prioridad</b>	Básico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	<p>2- El sistema muestra las opciones que se pueden realizar.</p> <ul style="list-style-type: none"> <li>• Insertar recomendaciones</li> <li>• Modificar lista de recomendaciones</li> <li>• Exportar lista de recomendaciones</li> </ul>
Sección "Insertar recomendaciones"	

<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
3- El actor selecciona la opción Insertar recomendaciones	4- Muestra una interfaz con un formulario.
5- Llena el formulario	6- Guarda los datos
	7- Termina el caso de uso
<b>Sección “Modificar lista de recomendaciones”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor solicita modificar los datos de la lista de recomendaciones	4- Solicita la lista de recomendaciones.
	5- Muestra una interfaz con los datos de esta
6- Modifica los datos que desee de la lista de recomendaciones	7- Actualiza los cambios realizados en la base de datos
	8- El caso de uso termina
<b>Sección “Exportar lista de recomendaciones ”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar lista de recomendaciones	4- Solicita la plantilla lista de recomendaciones.
	5- Muestra una interfaz con los datos de esta plantilla y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar la plantilla lista de recomendaciones	7- Exporta la plantilla lista de recomendaciones
	8- El caso de uso termina
<b>Flujos alternos</b>	

Acción del actor	Acción del sistema
6- Cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Generar informe final de actividad de control</b>
<b>Actores:</b>	<b>Revisor líder, Auditor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita gestionar el informe final de una actividad de control
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 25,26,27</b>
<b>Prioridad</b>	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar informe final</li> <li>• Modificar informe final</li> <li>• Exportar informe final</li> </ul>

#### Sección "Generar informe final"

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Insertar informe final	4- El sistema muestra una interfaz con un formulario
5- Llena el formulario	6- Guarda los datos

	7- Termina el caso de uso
<b>Sección “Modificar informe final”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3- El actor solicita modificar los datos del informe final	4- Solicita informe final.
	5- Muestra una interfaz con los datos de esta
6- Modifica los datos que desee del informe final	7- Actualiza los cambios realizados
	8- El caso de uso termina
<b>Sección “Exportar informe final”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar informe final	4- Solicita la plantilla informe final.
	5- Muestra una interfaz con los datos de esta plantilla y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar la plantilla informe final	7- Exporta la plantilla informe final
	8- El caso de uso termina
<b>Flujos alternos</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
6- cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Generar documento de no conformidades</b>
<b>Actores:</b>	<b>Auditor líder, revisor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita generar la plantilla documento de no conformidades
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF 19,20,21</b>
<b>Prioridad</b>	Básico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Generar documento de no conformidades</li> <li>• Modificar documento de no conformidades</li> <li>• Exportar documento de no conformidades</li> </ul>

#### Sección “Generar documento de no conformidades ”

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Generar documento de no conformidades	4- Solicita todos los datos necesarios para crear esta plantilla
	5- Muestra la plantilla en una interfaz y la guarda en la base de datos
	6- Termina el caso de uso

#### Sección “Modificar documento de no conformidades ”

Acción del actor	Respuesta del sistema

3- El actor solicita modificar los datos del documento de no conformidades	4- Solicita las no conformidades.
	5- Muestra una interfaz con los datos
6- Modifica los datos que desee	7- Actualiza los cambios realizados
	8- El caso de uso termina

**Sección “Exportar documento de no conformidades ”**

<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar documento de no conformidades	4- Solicita la plantilla documento de no conformidades.
	5- Muestra una interfaz con los datos de esta plantilla y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar esta plantilla	7- Exporta la plantilla
	8- El caso de uso termina

**Flujos alternos**

<b>Acción del actor</b>	<b>Acción del sistema</b>
6- cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar acciones correctivas, preventivas y de mejoras</b>
<b>Actores:</b>	<b>Auditor líder, Revisor líder</b>
<b>Resumen:</b>	Este caso de uso se inicia cuando se necesita gestionar la plantilla acciones correctivas, preventivas y de mejoras

<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	RF 22,23,24
<b>Prioridad</b>	Básico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del sistema
1- El actor accede al sistema.	2- El sistema muestra las opciones que se pueden realizar. <ul style="list-style-type: none"> <li>• Insertar acciones correctivas, preventivas y de mejoras</li> <li>• Modificar acciones correctivas, preventivas y de mejoras</li> <li>• Exportar acciones correctivas, preventivas y de mejoras</li> </ul>

#### **Sección “Insertar acciones correctivas, preventivas y de mejoras”**

Acción del Actor	Respuesta del sistema
3- El actor selecciona la opción Insertar acciones correctivas, preventivas y de mejoras	4- El sistema muestra una interfaz con un formulario
5- Llena el formulario	6- Guarda los datos
	7- Termina el caso de uso

#### **Sección “Modificar acciones correctivas, preventivas y de mejoras”**

Acción del actor	Respuesta del sistema
3- El actor solicita modificar los datos de las acciones correctivas, preventivas y de mejoras	4- Solicita las acciones correctivas, preventivas y de mejoras.
	5- Muestra una interfaz con los datos



6- Modifica los datos que desee	7- Actualiza los cambios realizados
	8- El caso de uso termina
<b>Sección “Exportar acciones correctivas, preventivas y de mejoras”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
3- Selecciona la opción exportar acciones correctivas, preventivas y de mejoras	4- Solicita la plantilla acciones correctivas, preventivas y de mejoras.
	5- Muestra una interfaz con los datos de esta plantilla y solicita confirmar si realmente desea exportar
6- Confirma que desea exportar esta plantilla	7- Exporta la plantilla
	8- El caso de uso termina
<b>Flujos alternos</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
6- cancela la acción de exportar	7- Cierra la interfaz
	8- El caso de uso termina

<b>Caso de Uso:</b>	<b>Gestionar Plantillas de una actividad de control</b>
<b>Actores:</b>	<b>Auditor líder , Revisor líder</b>
<b>Resumen:</b>	El CU comienza cuando el actor solicita gestionar las plantillas de una actividad de control.
<b>Precondiciones:</b>	El usuario debe autenticarse y tener los privilegios necesarios para realizar esta acción.
<b>Referencias</b>	<b>RF del 13 al 30</b>

<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1- El actor accede al sistema.	2- El sistema solicita llenar un formulario con el nombre del proyecto, el tipo de actividad que se le realizó (Revisión o Auditoría) y el nombre de la Actividad
3- El actor especifica los datos	4- El sistema le muestra el conjunto de plantillas que corresponda en dependencia del tipo de actividad e indica cuáles de ellas ya fueron generadas anteriormente
5- El actor selecciona la plantilla	6- El sistema muestra la plantilla y muestra una opción para si el actor desea bloquearla
7- El actor bloquea la plantilla	8- El sistema registra esta acción
	9- El caso de uso termina
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
5- El actor no bloquea la plantilla	6- El caso de uso termina

### **2.14 Conclusiones del capítulo**

En este capítulo se ha abordado la situación problemática y las características del sistema. Se ha realizado el estudio de los objetivos y los procesos del negocio que los soportan, así como un análisis crítico de todos los procesos vinculados en el campo de acción. Se identificaron quiénes son los trabajadores y los actores que intervienen en el mismo y con cuáles actividades y entidades interactúan estos. Mediante el modelado del negocio se entendieron mejor los métodos de trabajo con los diagramas de actividades por casos de uso del negocio y con el levantamiento de requerimientos funcionales se definieron los casos de uso críticos a automatizar. Se muestran además, los requerimientos no

funcionales. Se describieron los actores que interactúan con el sistema, los casos de uso a automatizar y se presentan los diagramas de casos de uso del sistema y el diagrama de clases del modelo de objetos

## Capítulo 3. Análisis y diseño del módulo

### 2.1 Introducción

En el presente capítulo se expone el análisis y diseño propuesto para la solución de la aplicación, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. También se presentarán los diagramas de interacción, los diagramas de clases del diseño del sistema, el diagrama de clases persistente y el modelo de datos con que contará el sistema.

### 2.2 Modelo de análisis

El modelo de análisis es la primera representación técnica del sistema, por lo que puede considerarse como una primera aproximación al modelo de diseño, y es por tanto una entrada fundamental para las actividades de diseño e implementación subsiguientes. A partir de su realización se refinan y estructuran los requisitos obtenidos con anterioridad proporcionando una visión general del sistema. El modelo de análisis debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueda validar una vez que se construye el software. (Ingeniería de Software I 2008).

#### 2.2.1 Diagramas de Clases de Análisis

El diagrama de clases de análisis es un artefacto en el que se representa los conceptos fundamentales en un dominio del problema. Los diagramas de clases de análisis, representan las definiciones y relaciones entre las clases. Las clases del análisis se clasifican en Interfaz, de Control o Entidad.

- Clase de Interfaz: Modelan la interacción entre el sistema y sus actores.
- Clase Entidad: Modelan información que posee una larga vida.
- Clase de Control: Representan coordinación, secuencia, transacciones, y control de otros objetos y a menudo encapsula a un caso de uso en concreto.

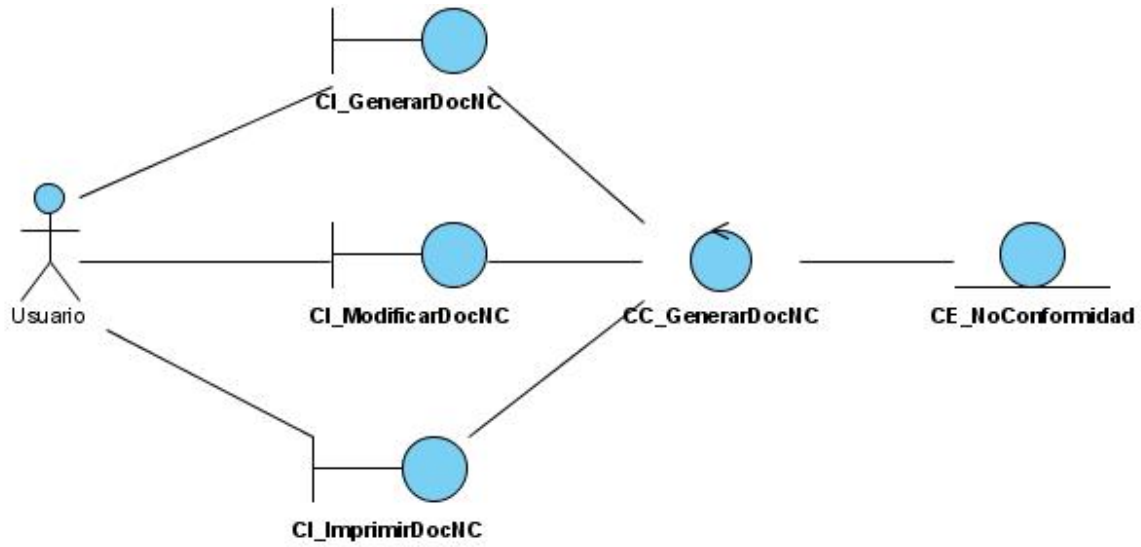


Fig. 6 - Diagrama de Clases de Análisis “Generar Informe de No Conformidades”

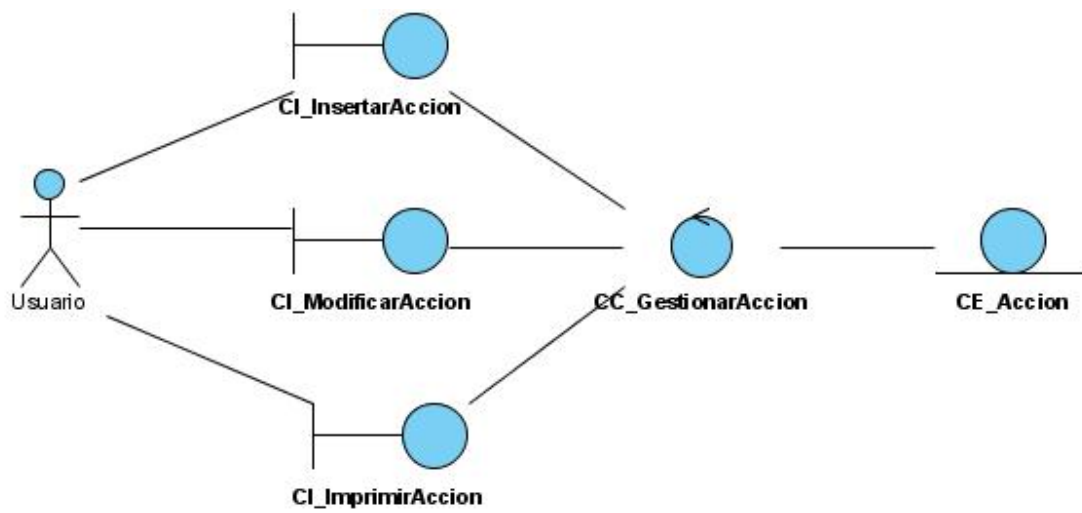


Fig. 7 - Diagrama de Clases de Análisis “Gestionar Acciones”

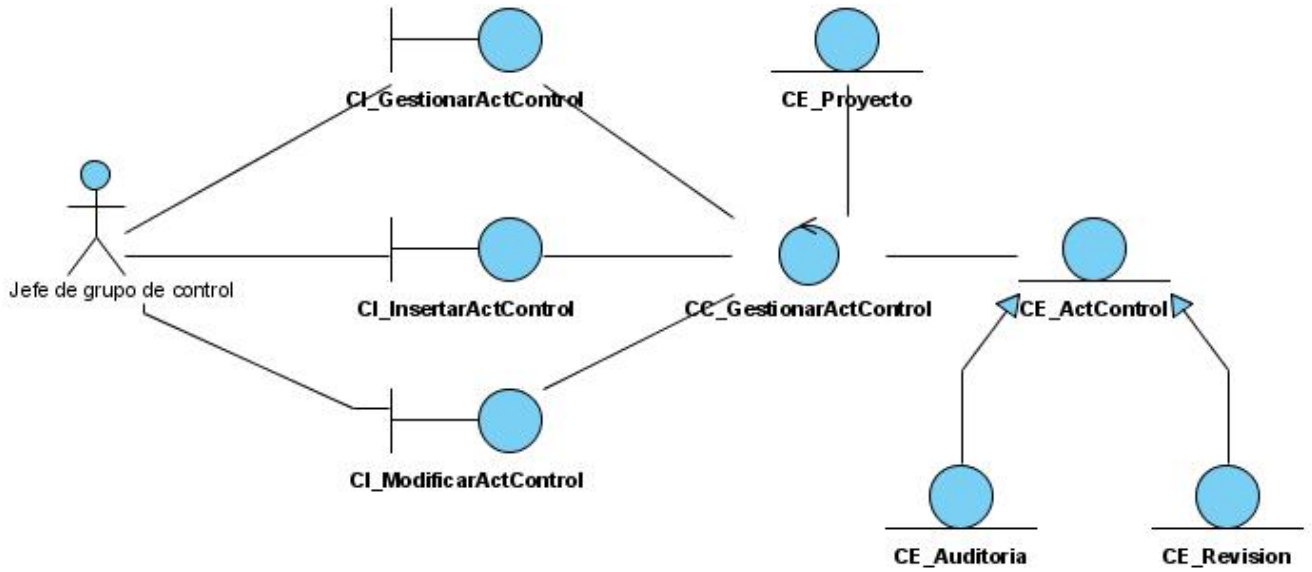


Fig. 8 - Diagrama de Clases de Análisis “Gestionar Actividad de control”

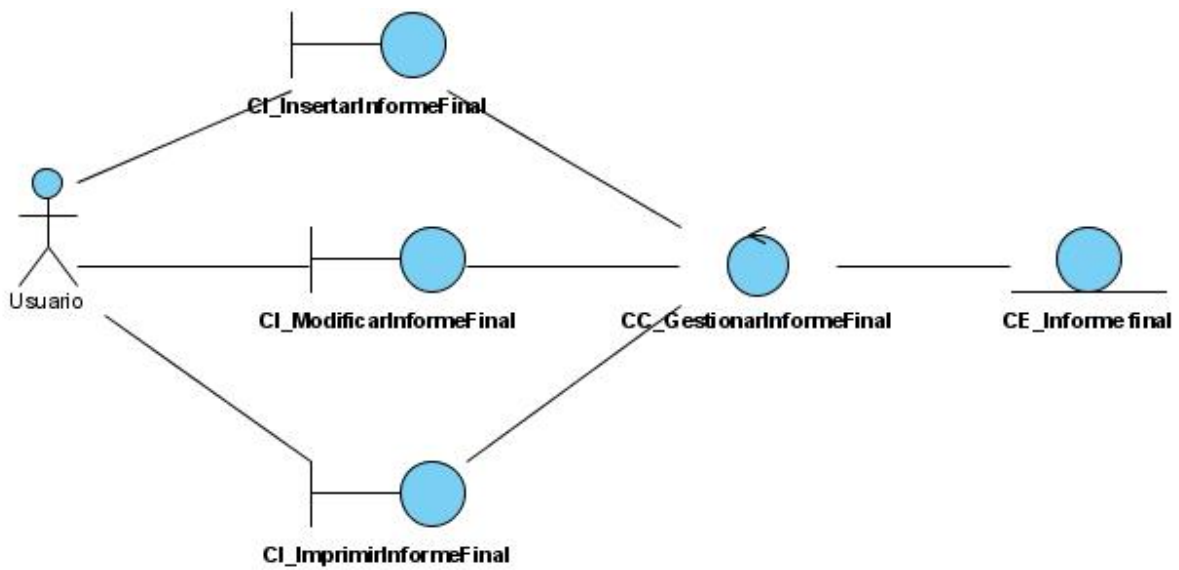


Fig. 9 - Diagrama de Clases de Análisis “Gestionar Informe Final”

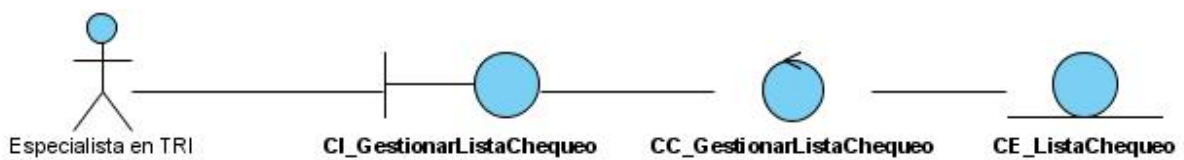


Fig. 10 - Diagrama de Clases de Análisis “Gestionar Lista de chequeo”

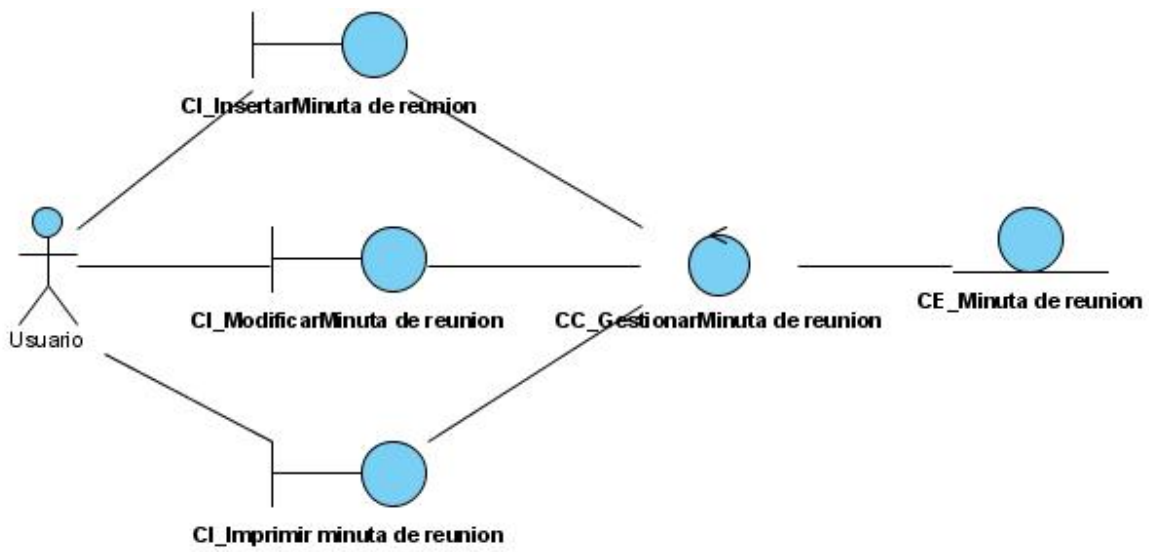


Fig. 11 - Diagrama de Clases de Análisis “Gestionar Minuta de Reunión”

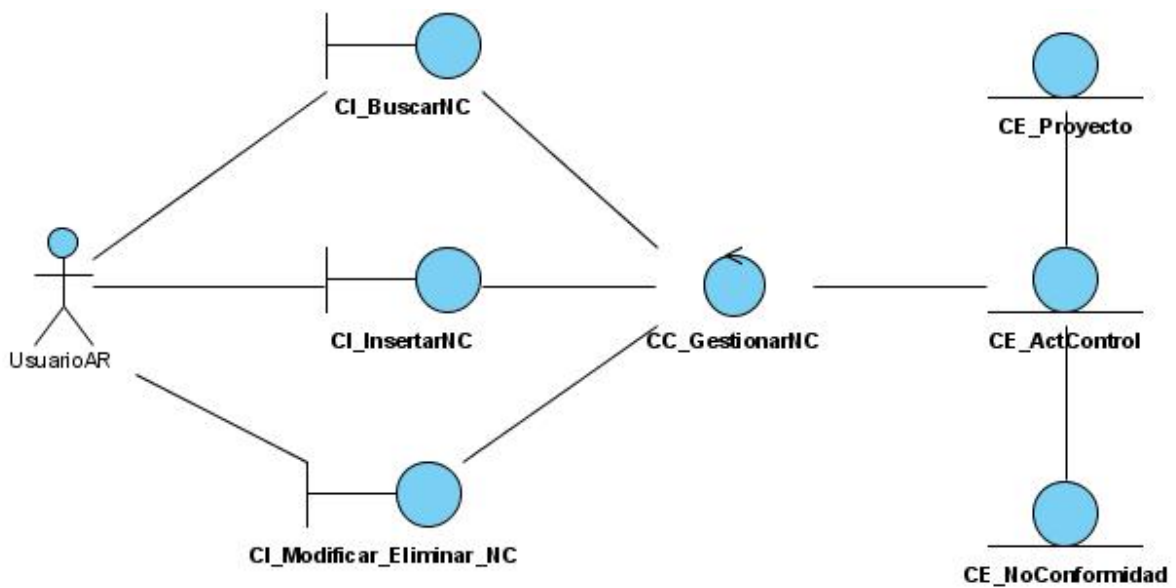


Fig. 12 - Diagrama de Clases de Análisis “Gestionar No Conformidad”

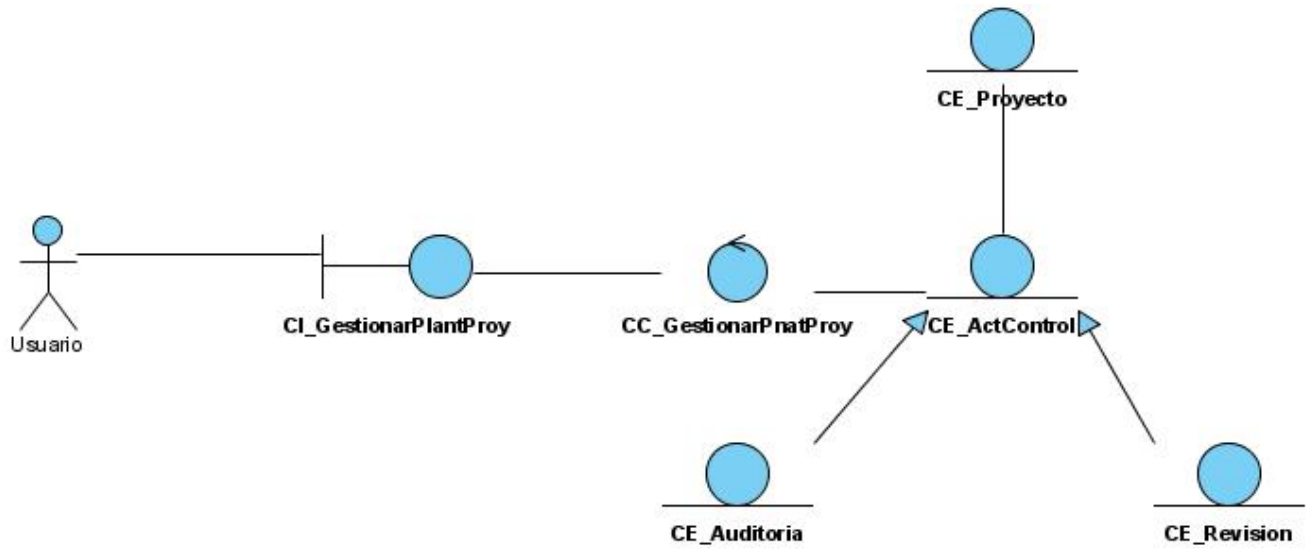


Fig. 13 - Diagrama de Clases de Análisis “Gestionar Plantillas de Actividad de Control”

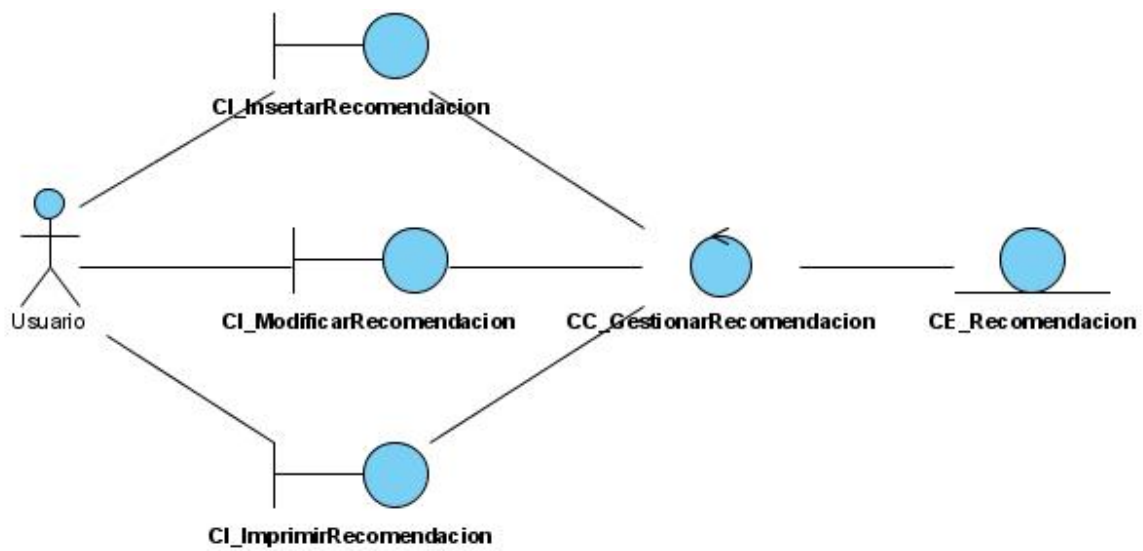


Fig. 14 - Diagrama de Clases de Análisis “Gestionar Recomendación”



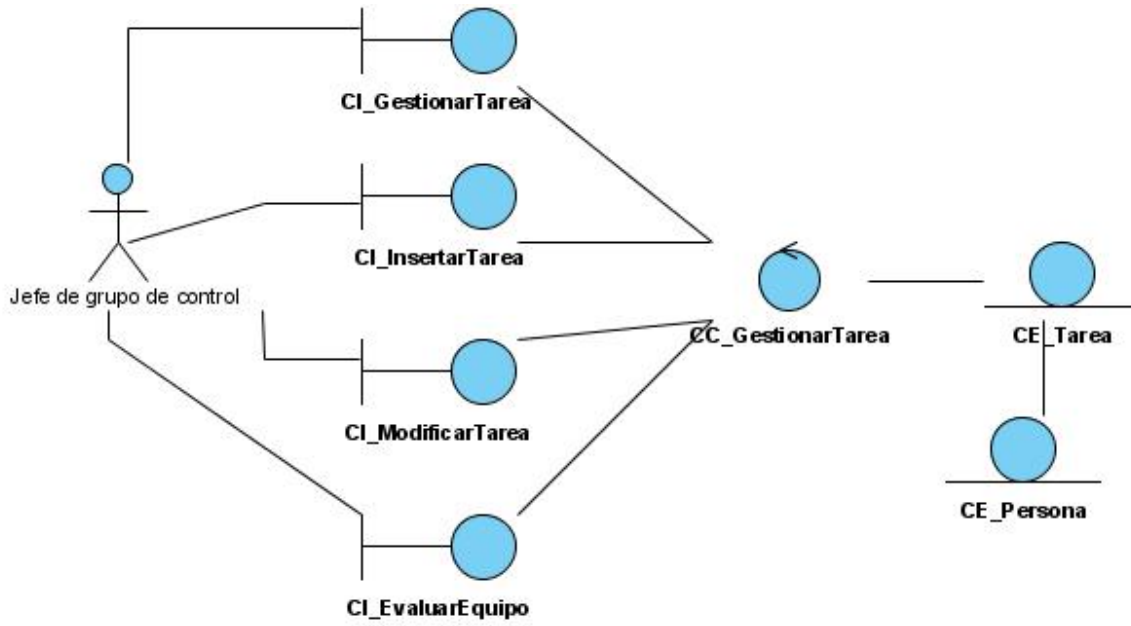


Fig. 15 - Diagrama de Clases de Análisis "Gestionar Tarea"

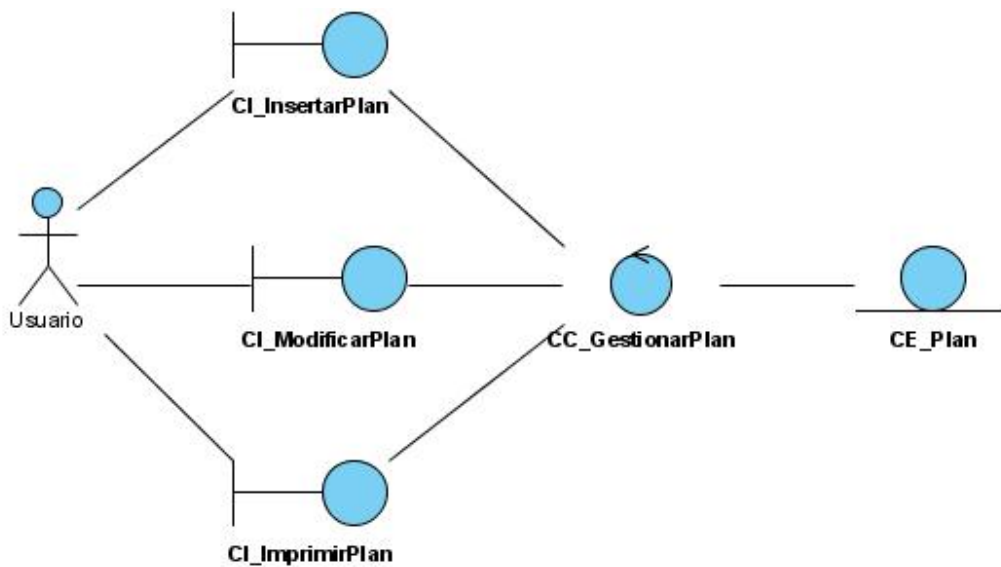


Fig. 16 - Diagrama de Clases de Análisis "Gestionar Plan"

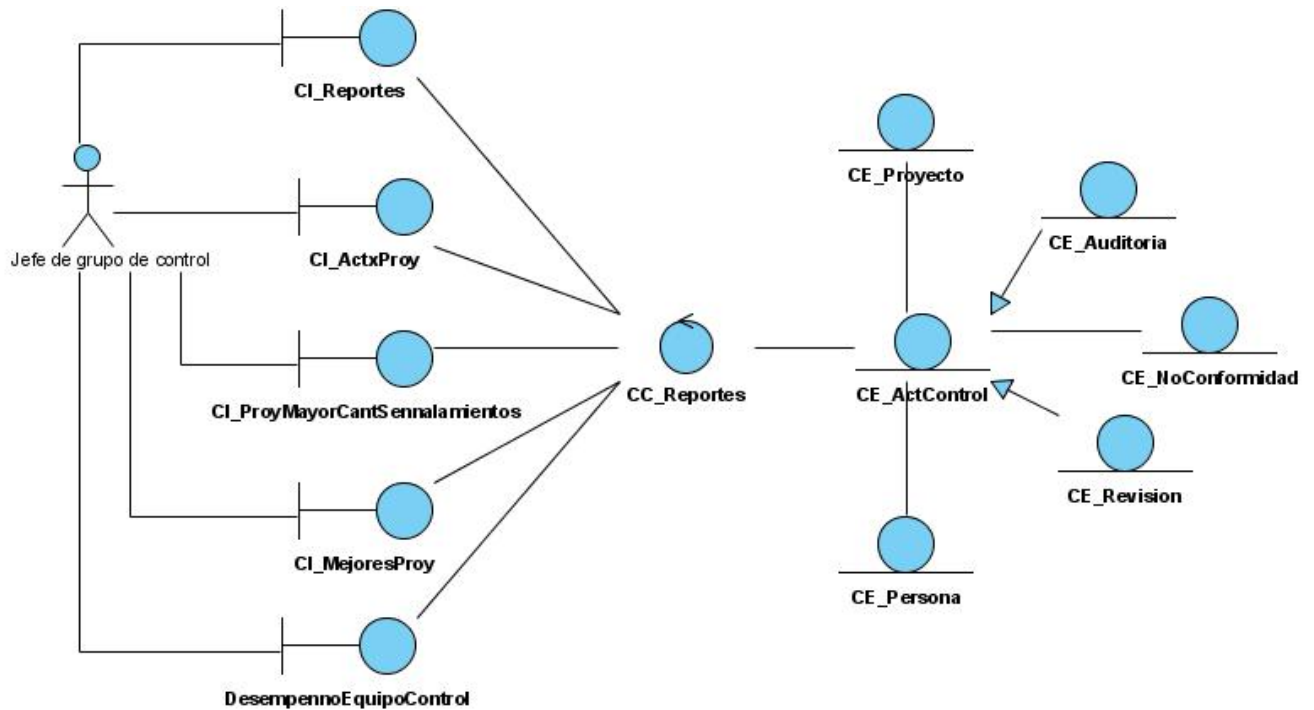


Fig. 17 - Diagrama de Clases de Análisis “Solicitar Reportes de Actividades de control”

### 2.3 Modelo de Diseño

El diseño es el centro de atención final de la fase de elaboración y el comienzo de las iteraciones de construcción, modela el sistema y se le da forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Además impone una estructura que se debe conservar lo más fielmente posible cuando se le dé forma al sistema. (Ingeniería de Software I 2008).

### 2.4 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Pueden servir para visualizar, especificar, construir y documentar los aspectos dinámicos de una sociedad particular de objetos, o pueden ser usados para modelar un flujo particular de control de un caso de uso. (Ingeniería de Software I 2008).

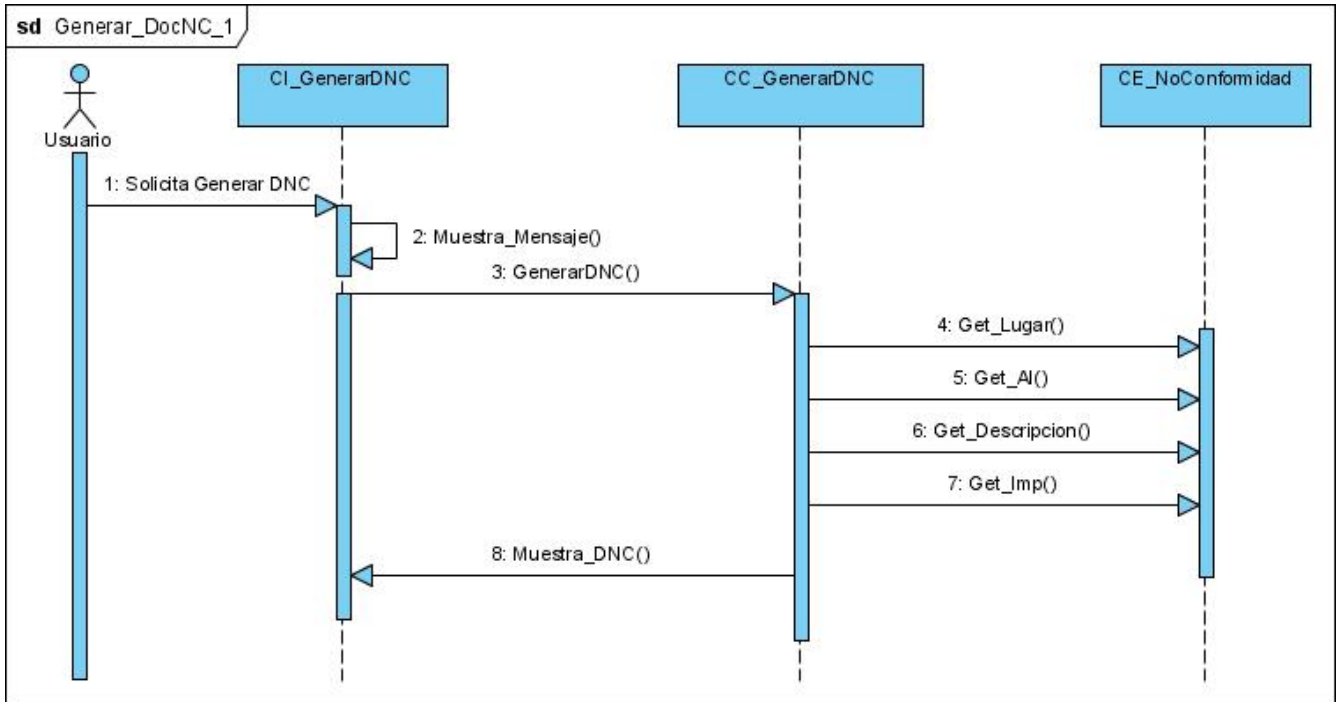


Fig. 18 - Diagrama de Secuencia “Generar documento de No Conformidades”

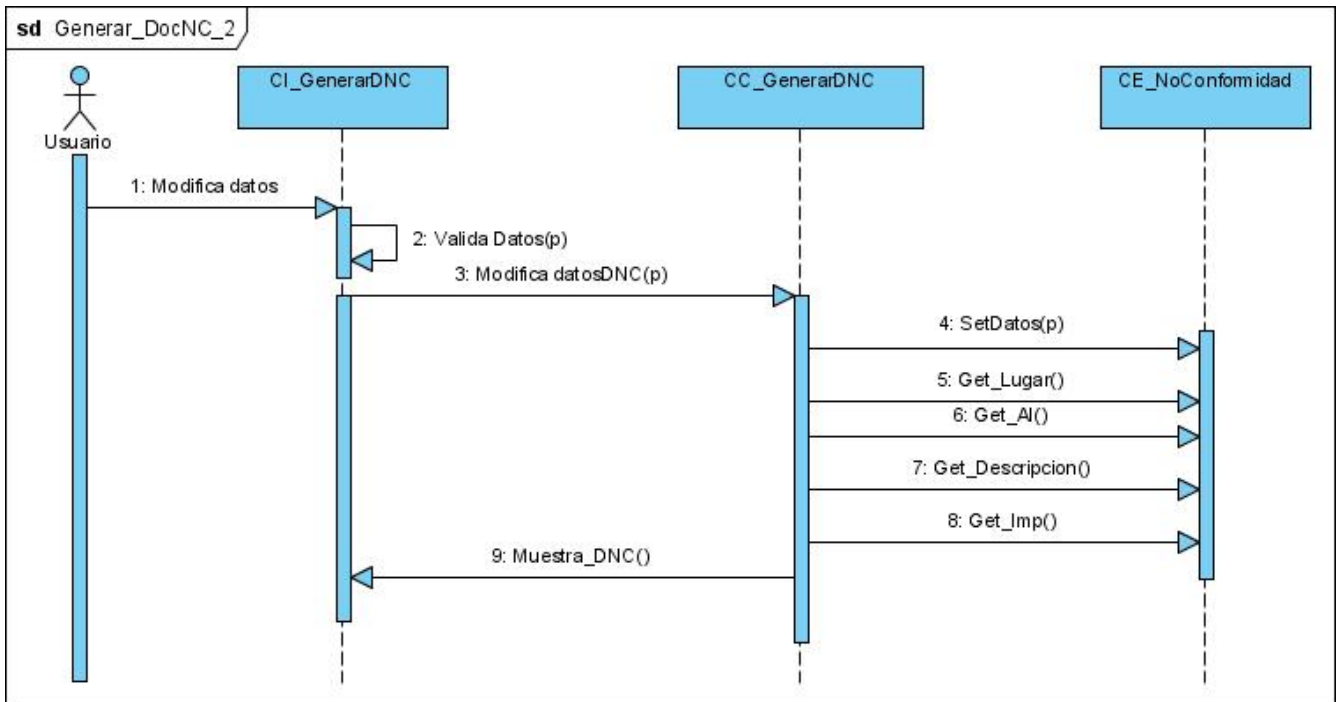


Fig. 19 - Diagrama de Secuencia “Generar documento de No Conformidades”

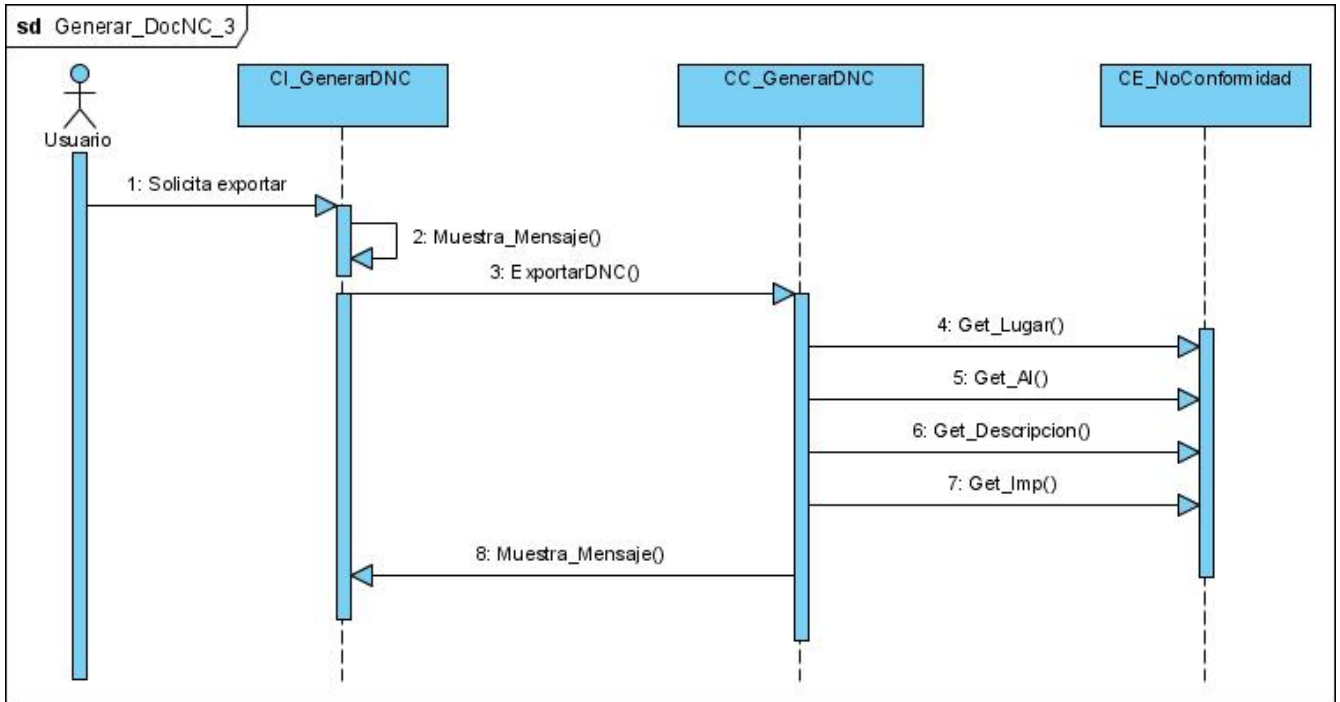


Fig. 20 - Diagrama de Secuencia “Generar documento de No Conformidades”

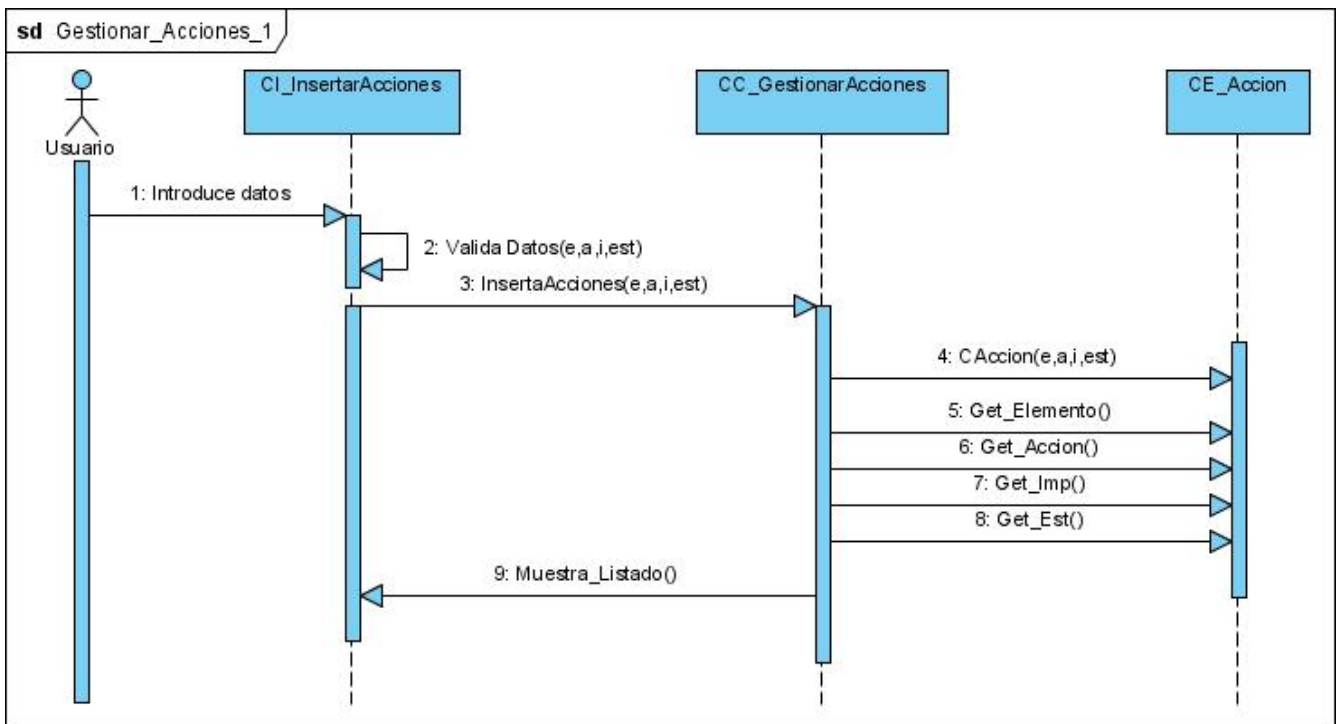
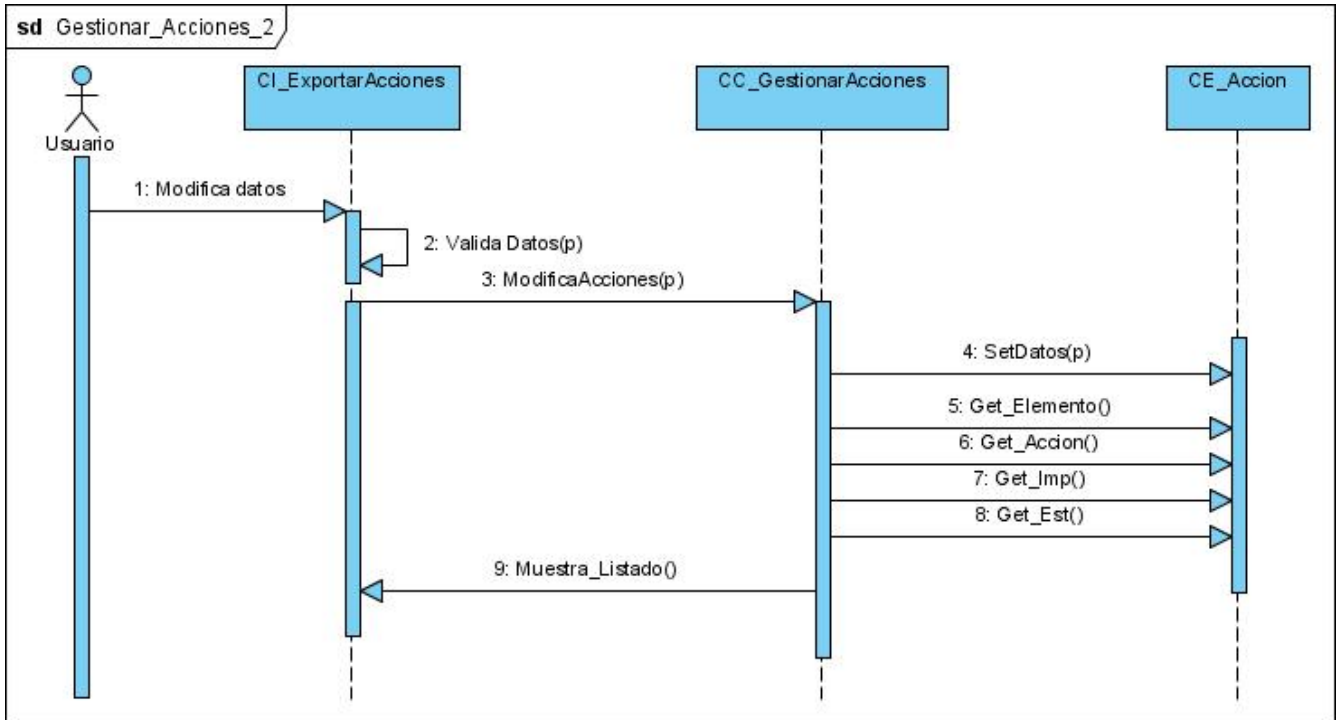
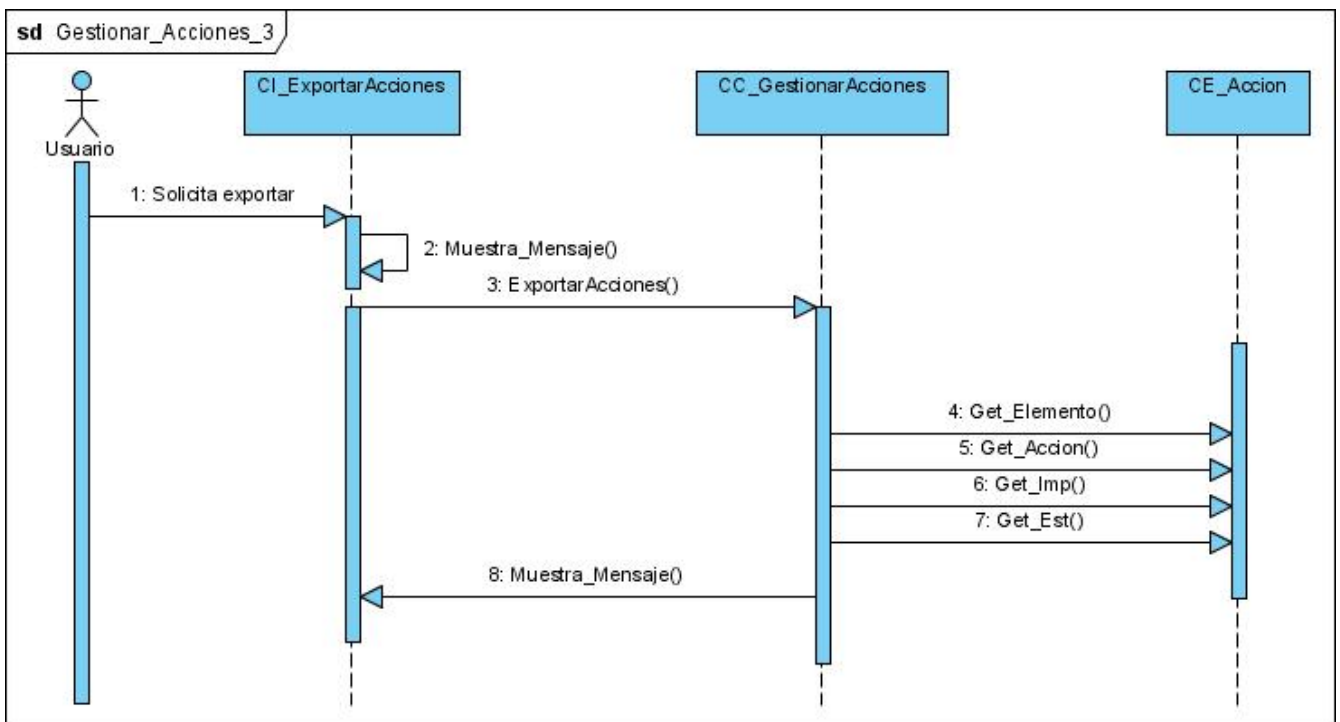


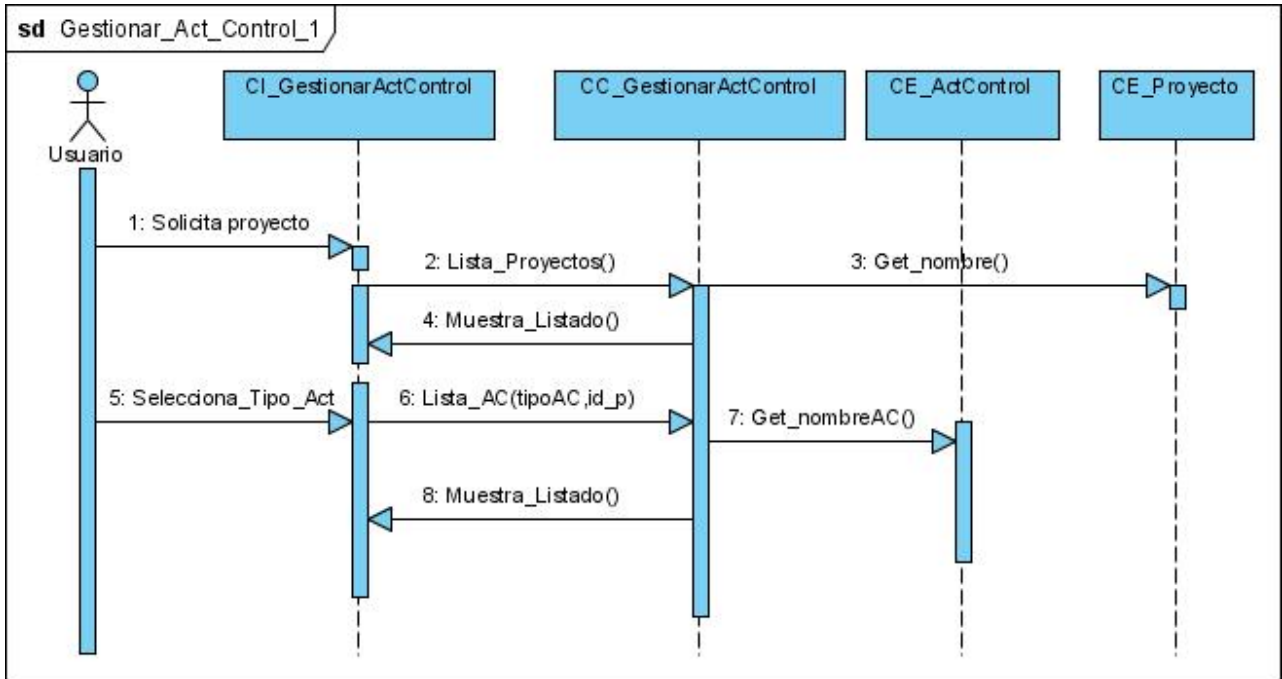
Fig. 21 - Diagrama de Secuencia “Gestionar acciones”



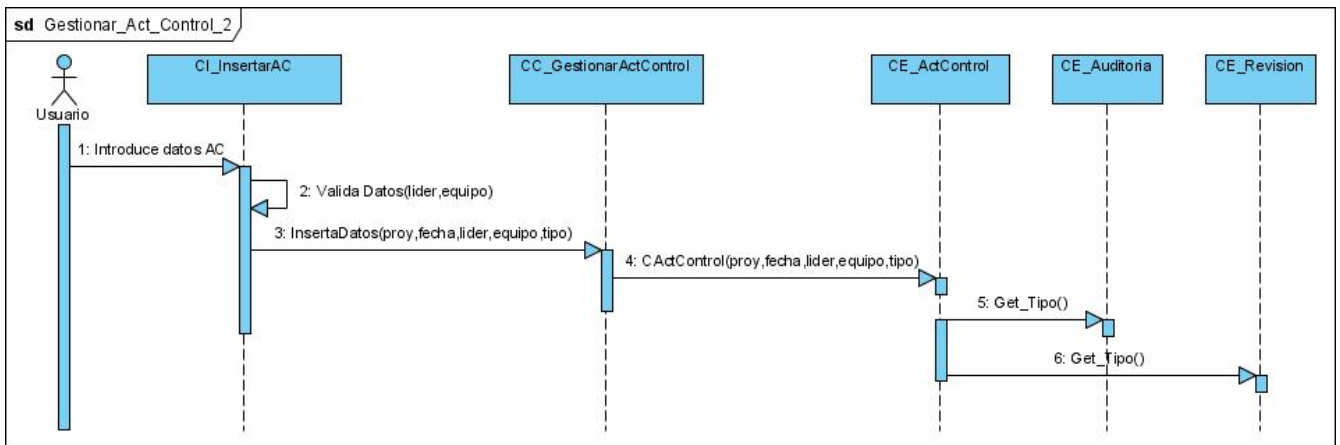
**Fig. 22 - Diagrama de Secuencia "Gestionar acciones"**



**Fig. 23 - Diagrama de Secuencia "Gestionar acciones"**



**Fig. 24 - Diagrama de Secuencia “Gestionar actividad de control”**



**Fig. 25 - Diagrama de Secuencia “Gestionar actividad de control”**

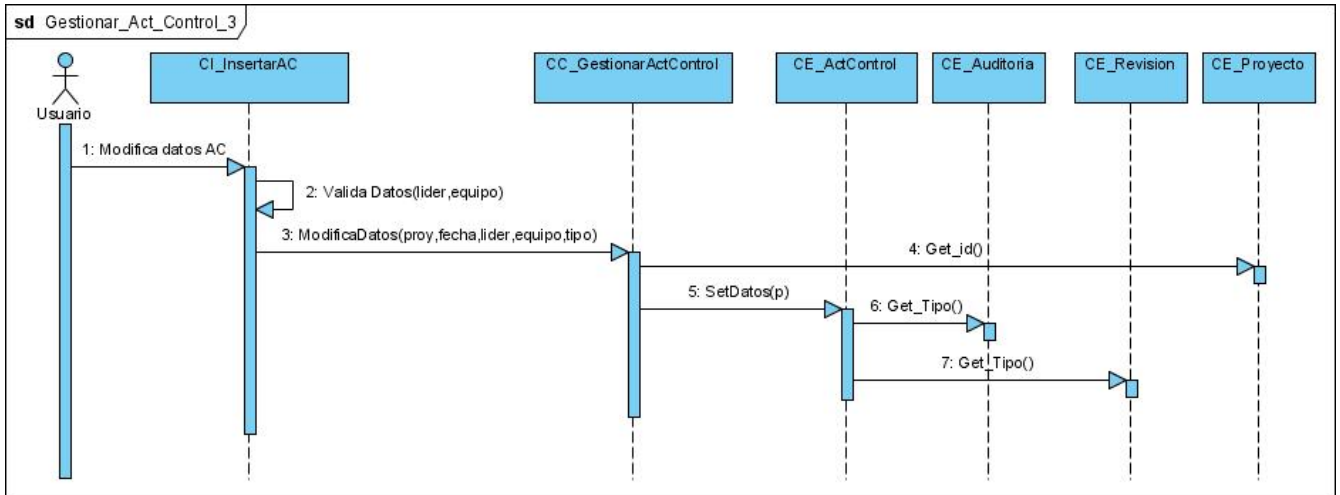


Fig. 26 - Diagrama de Secuencia "Gestionar actividad de control"

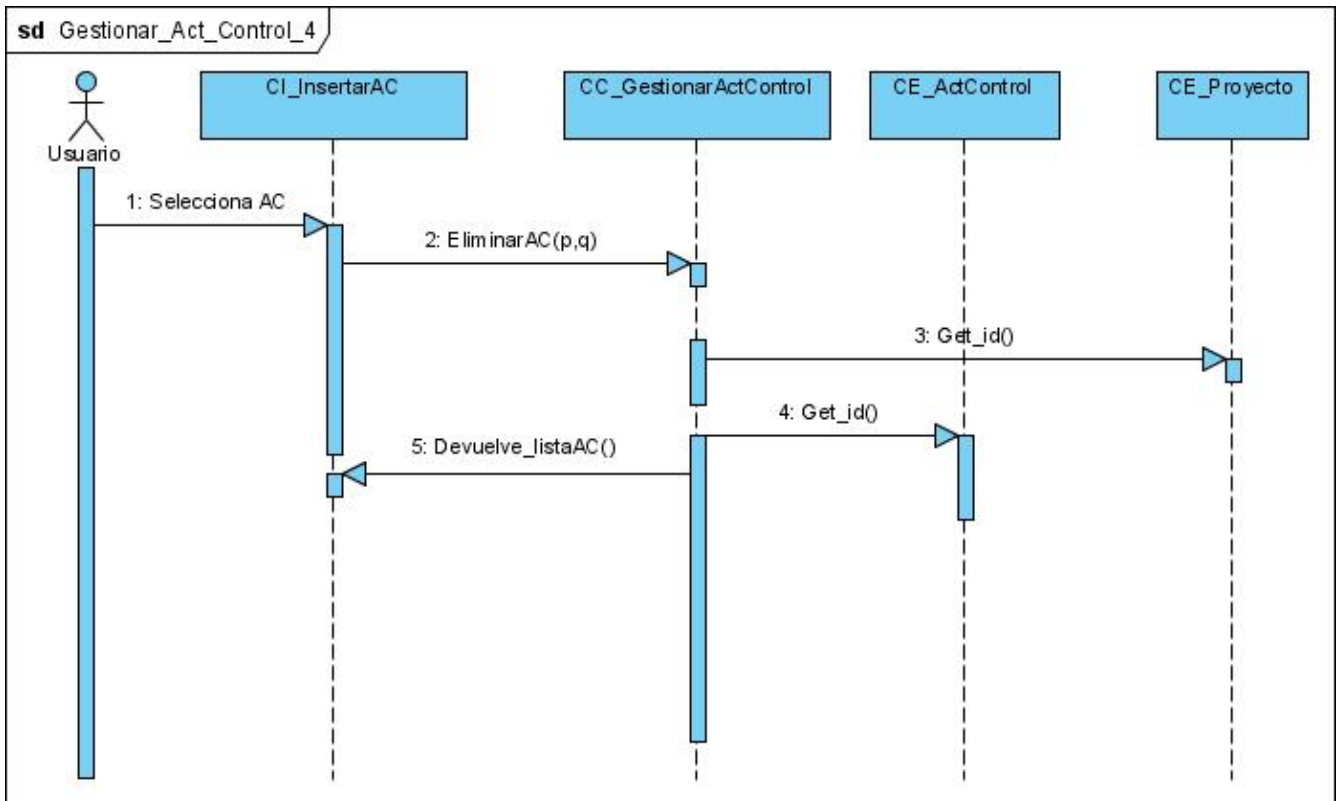
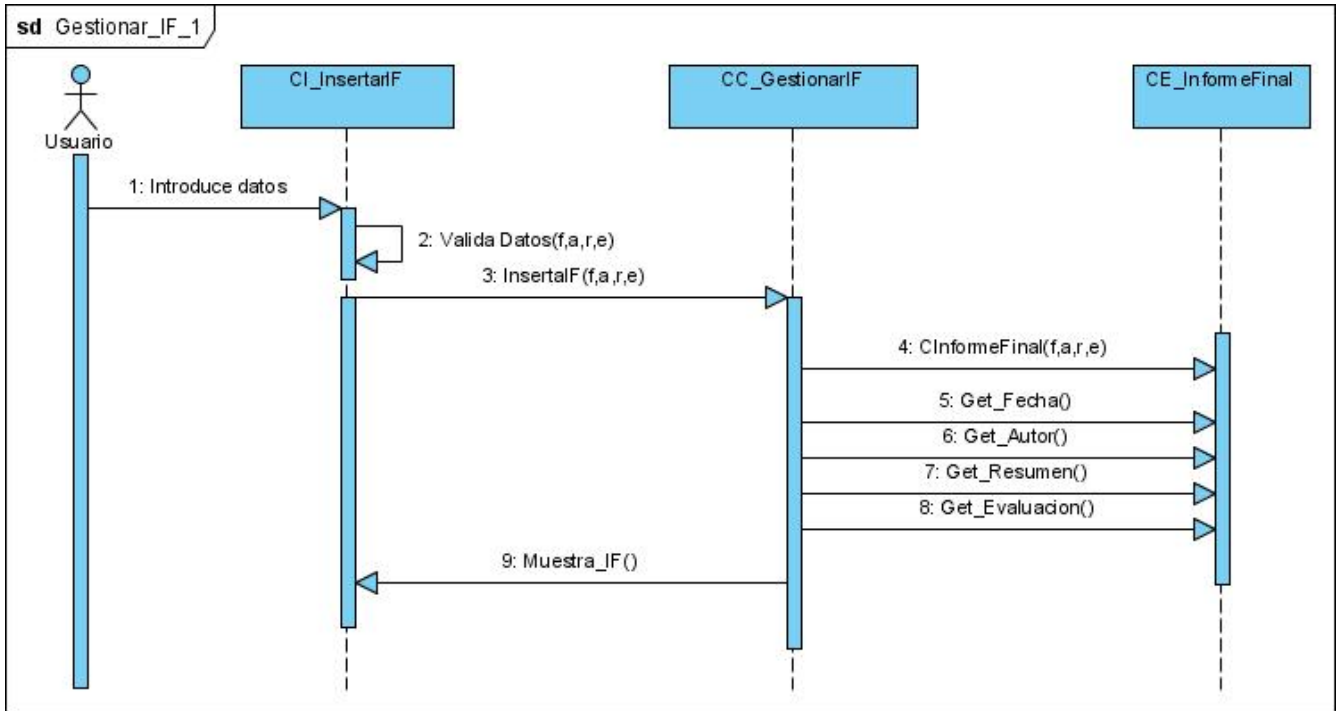
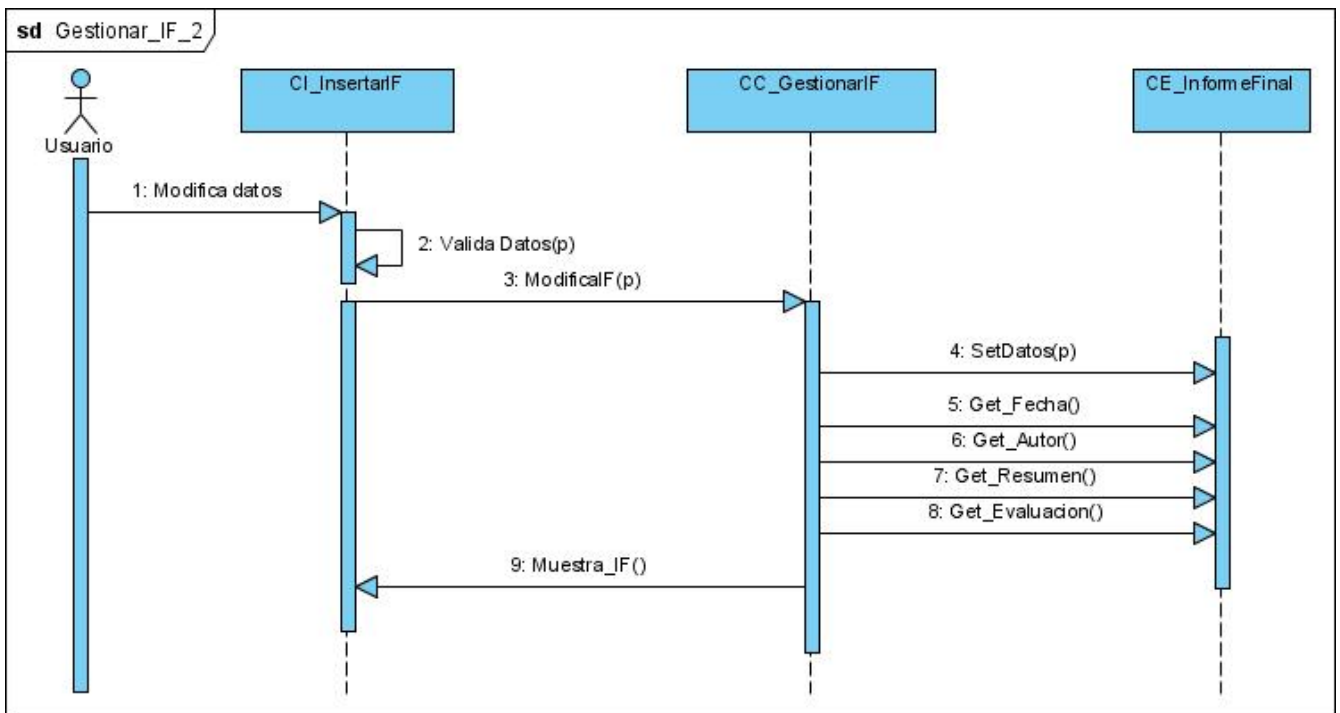


Fig. 27 - Diagrama de Secuencia "Gestionar actividad de control"



**Fig. 28 - Diagrama de Secuencia "Generar informe final"**



**Fig. 29 - Diagrama de Secuencia "Generar informe final"**



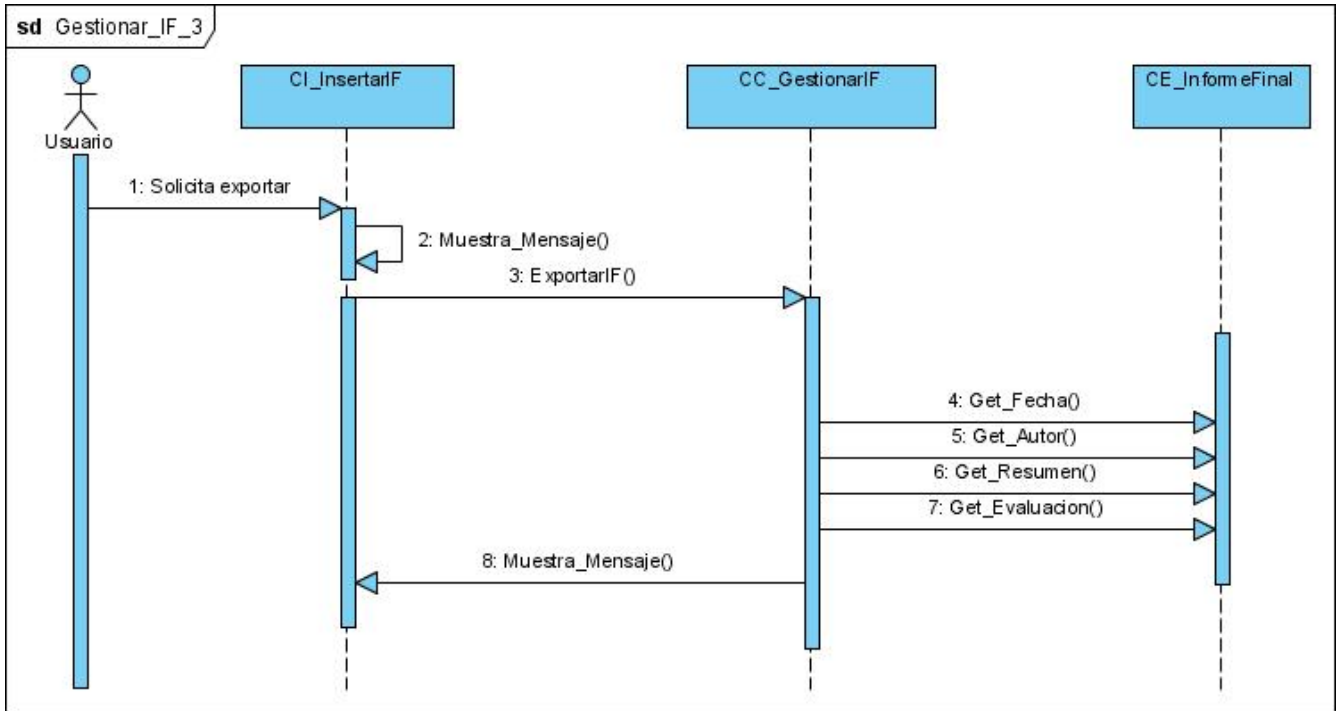


Fig. 30 - Diagrama de Secuencia "Generar informe final"

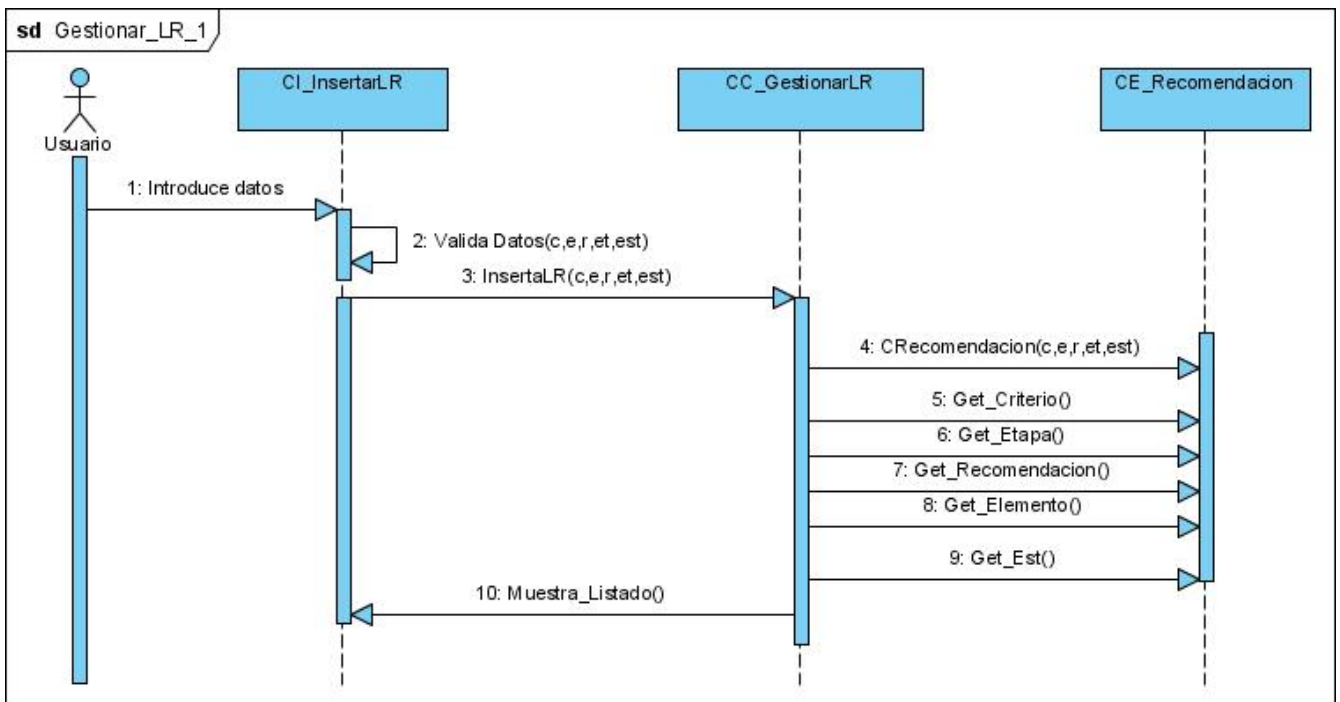
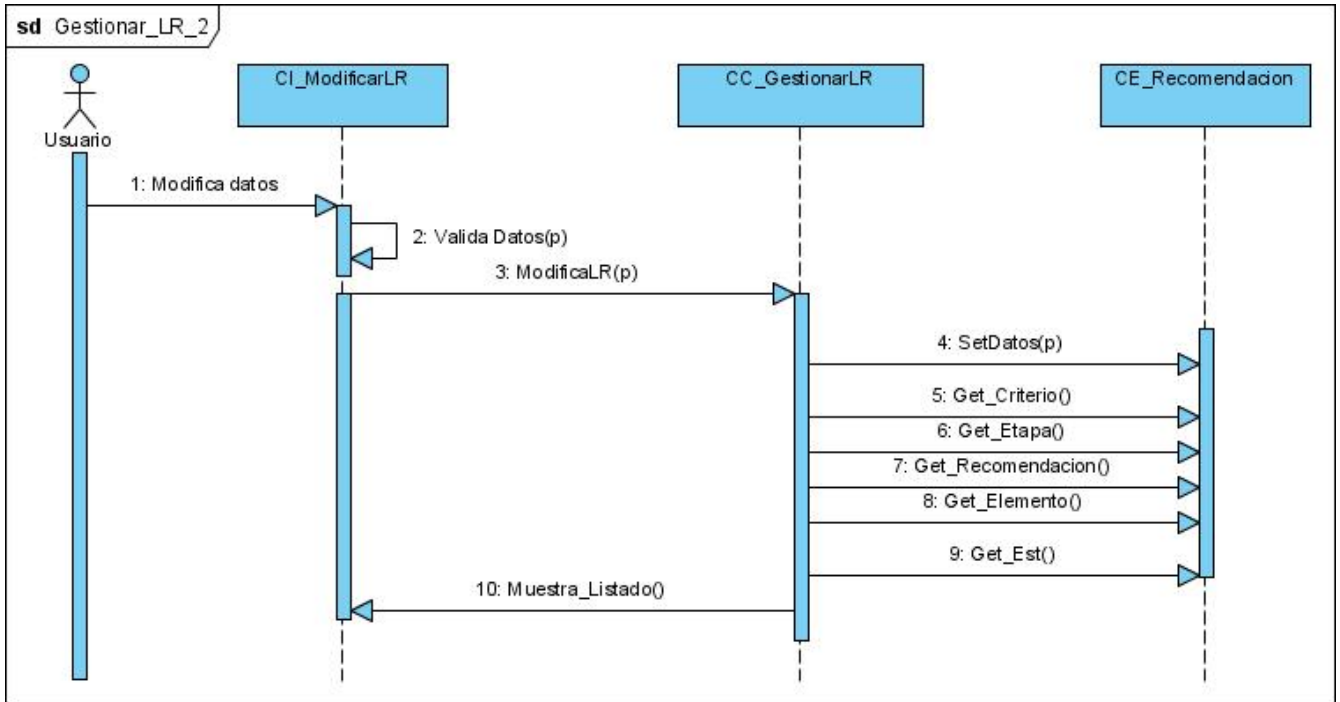
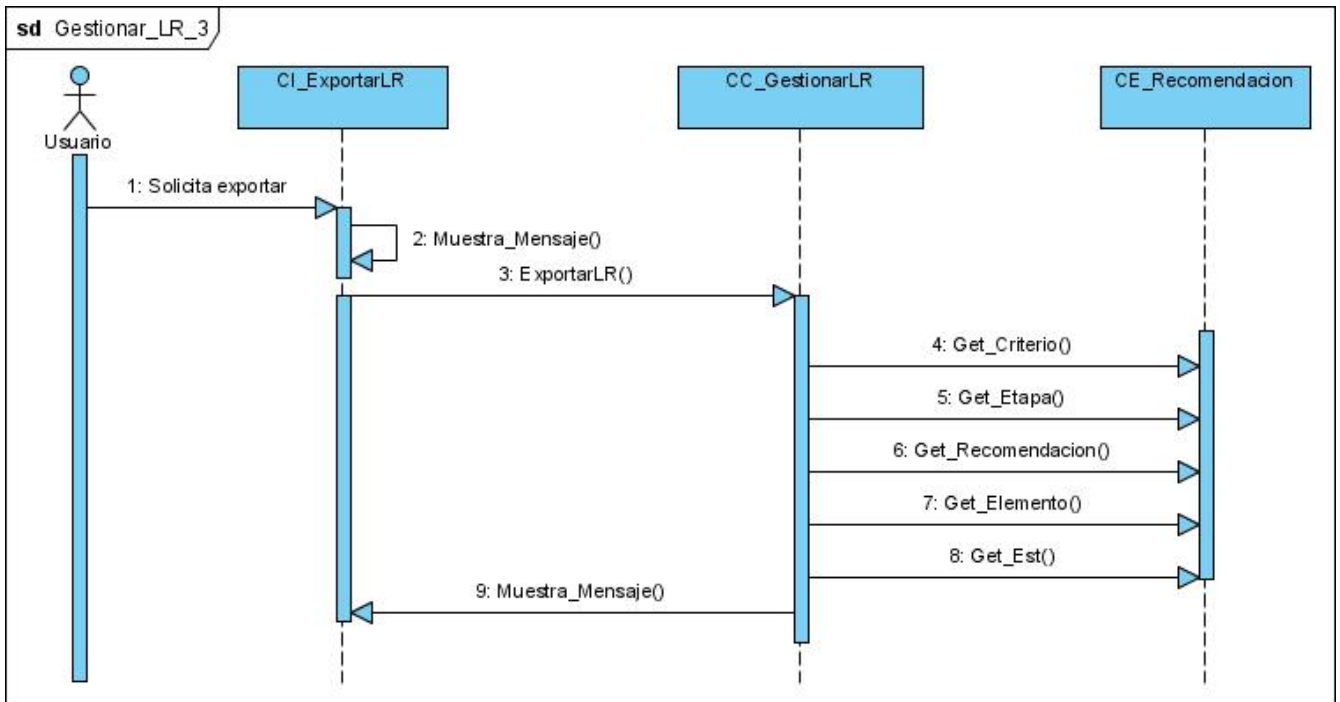


Fig. 31 - Diagrama de Secuencia "Gestionar lista de recomendaciones"



**Fig. 32 - Diagrama de Secuencia "Gestionar lista de recomendaciones"**



**Fig. 33 - Diagrama de Secuencia "Gestionar lista de recomendaciones"**

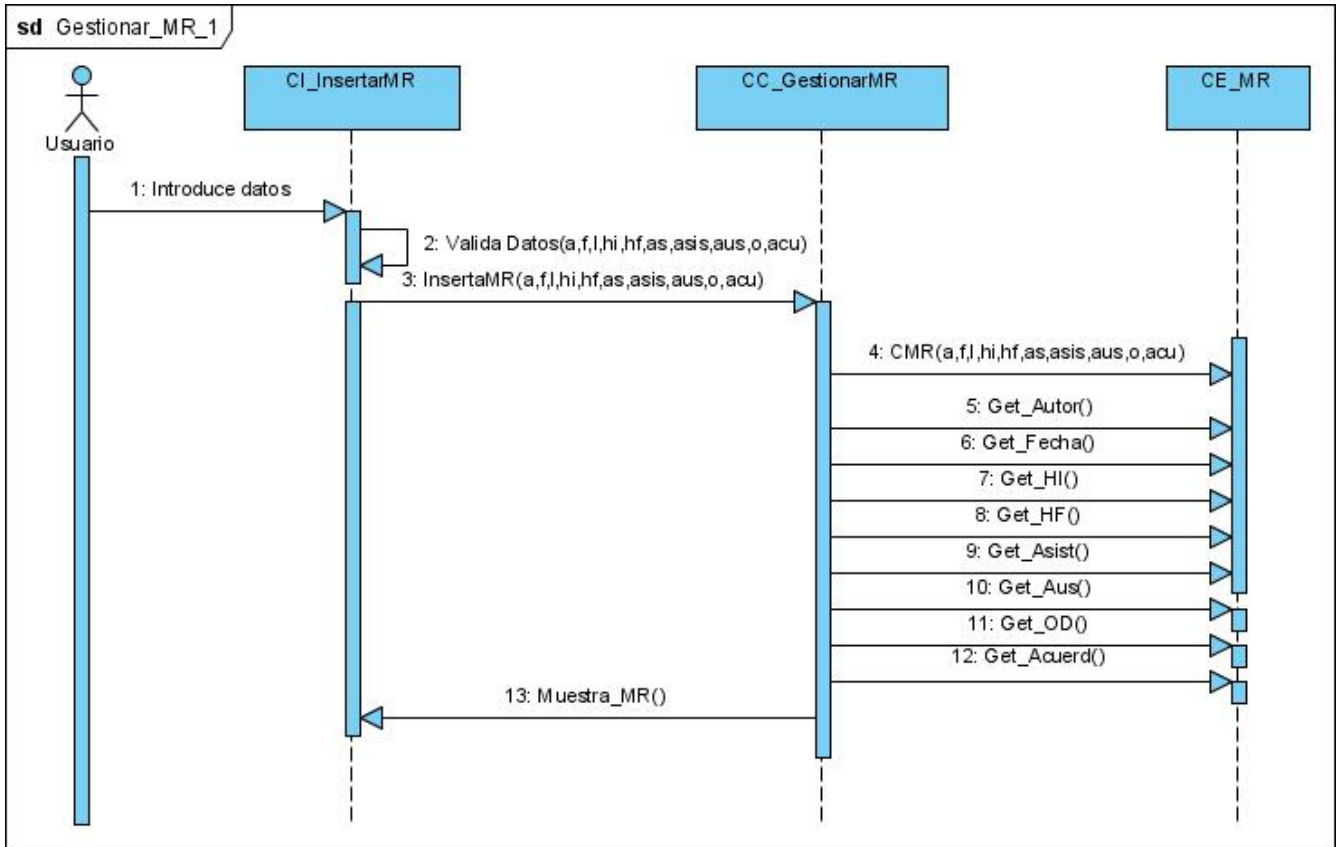


Fig. 34 - Diagrama de Secuencia "Gestionar minuta de reunión"

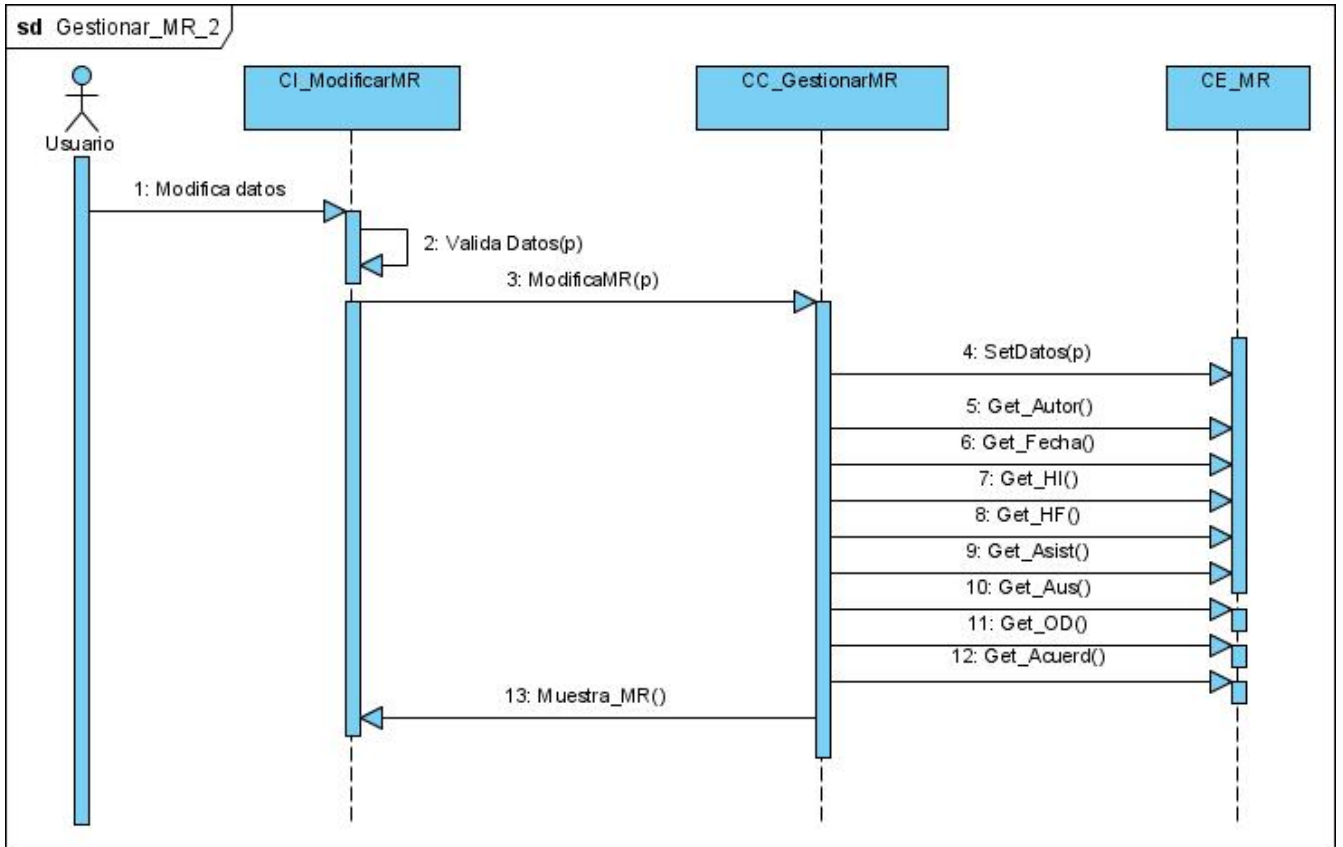
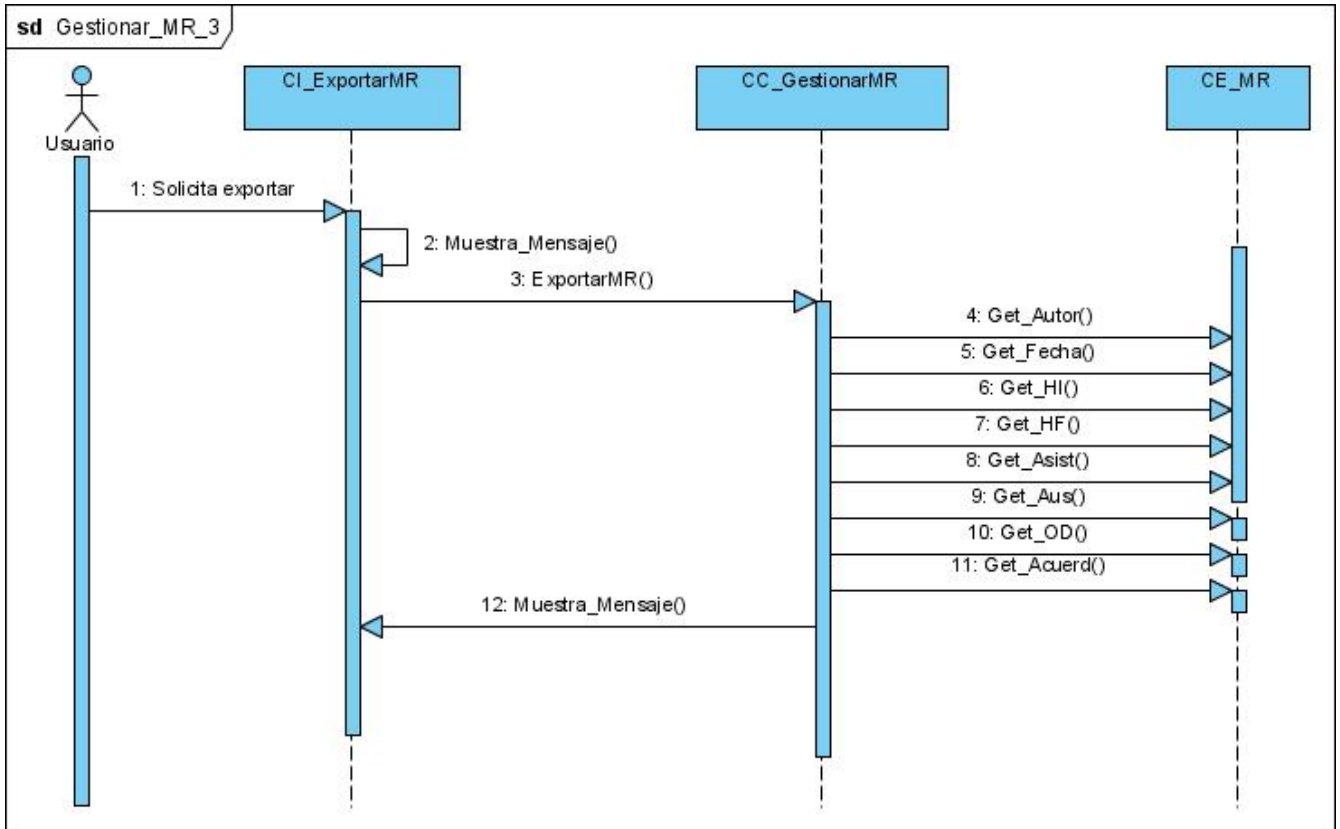
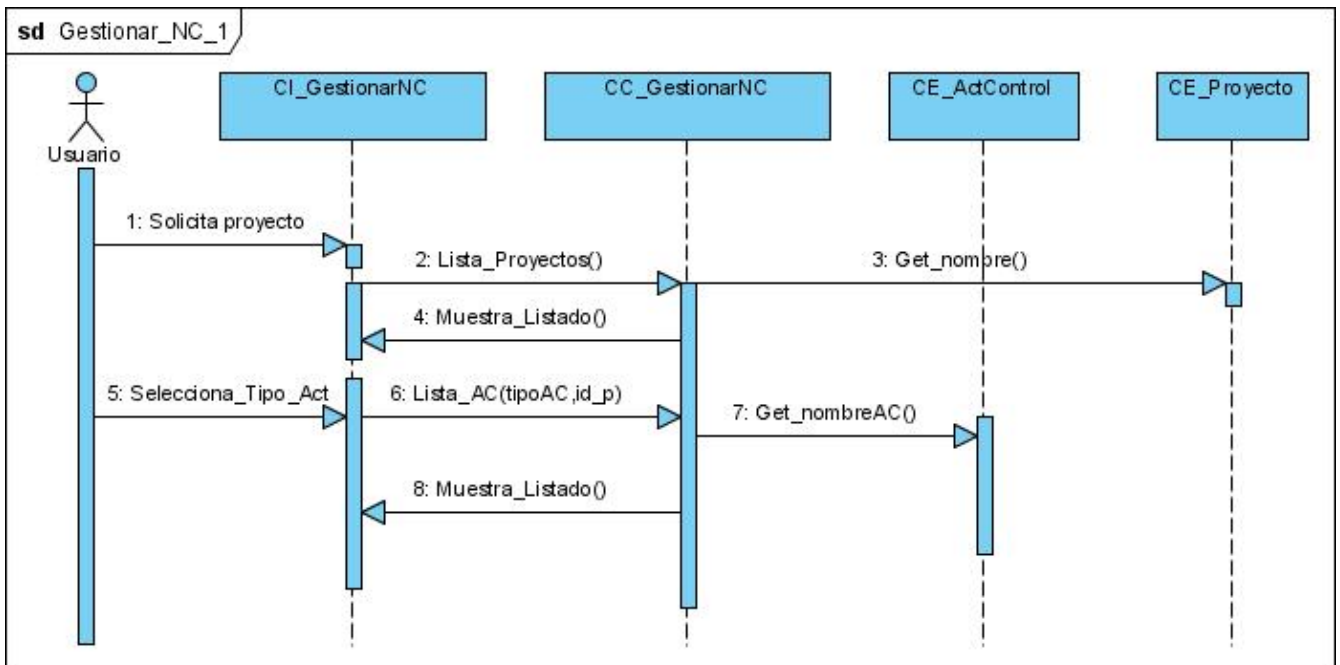


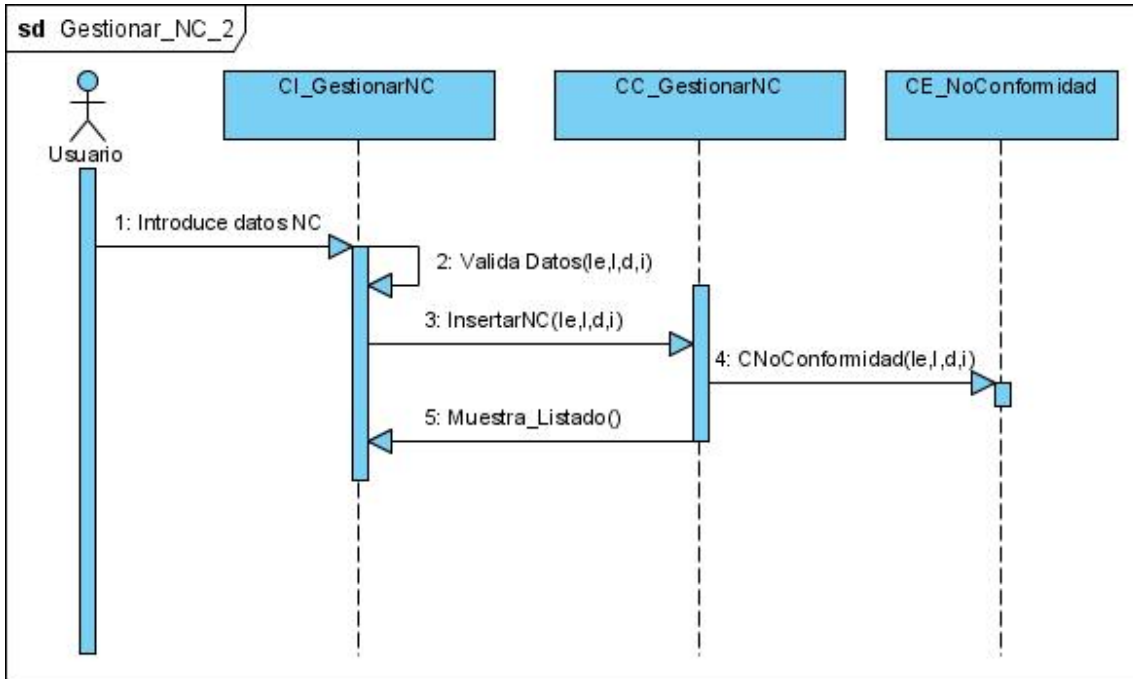
Fig. 35 - Diagrama de Secuencia "Gestionar minuta de reunión"



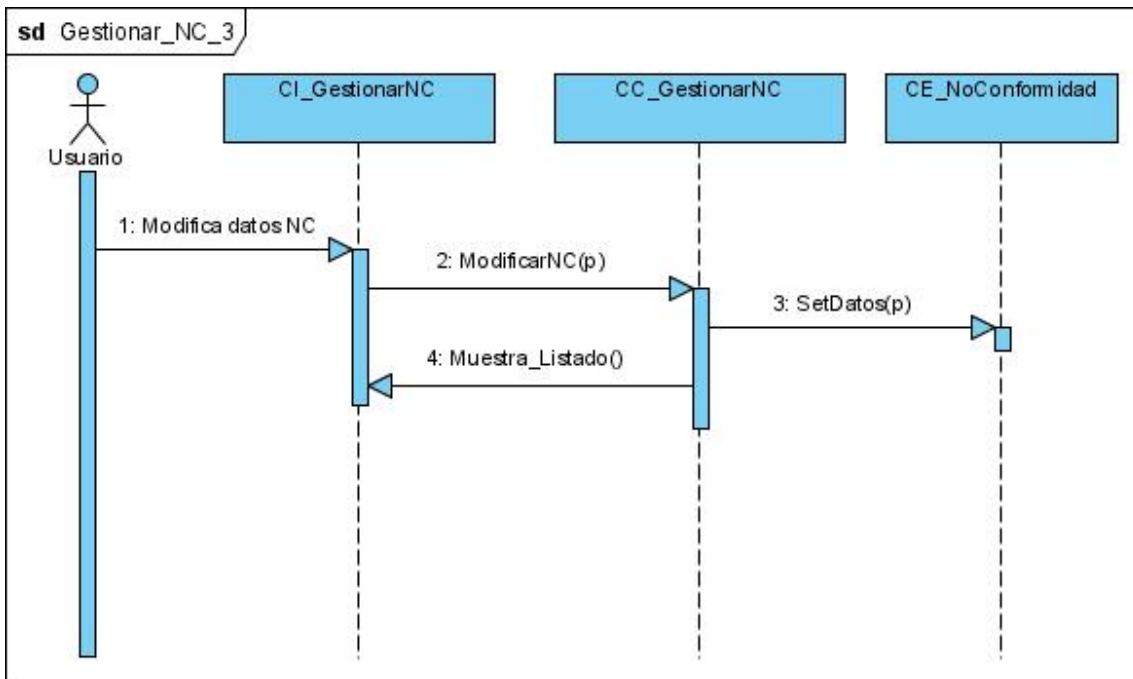
**Fig. 36 - Diagrama de Secuencia "Gestionar minuta de reunión"**



**Fig. 37 - Diagrama de Secuencia "Gestionar No conformidad"**



**Fig. 38 - Diagrama de Secuencia “Gestionar No conformidad”**



**Fig. 39 - Diagrama de Secuencia “Gestionar No conformidad”**

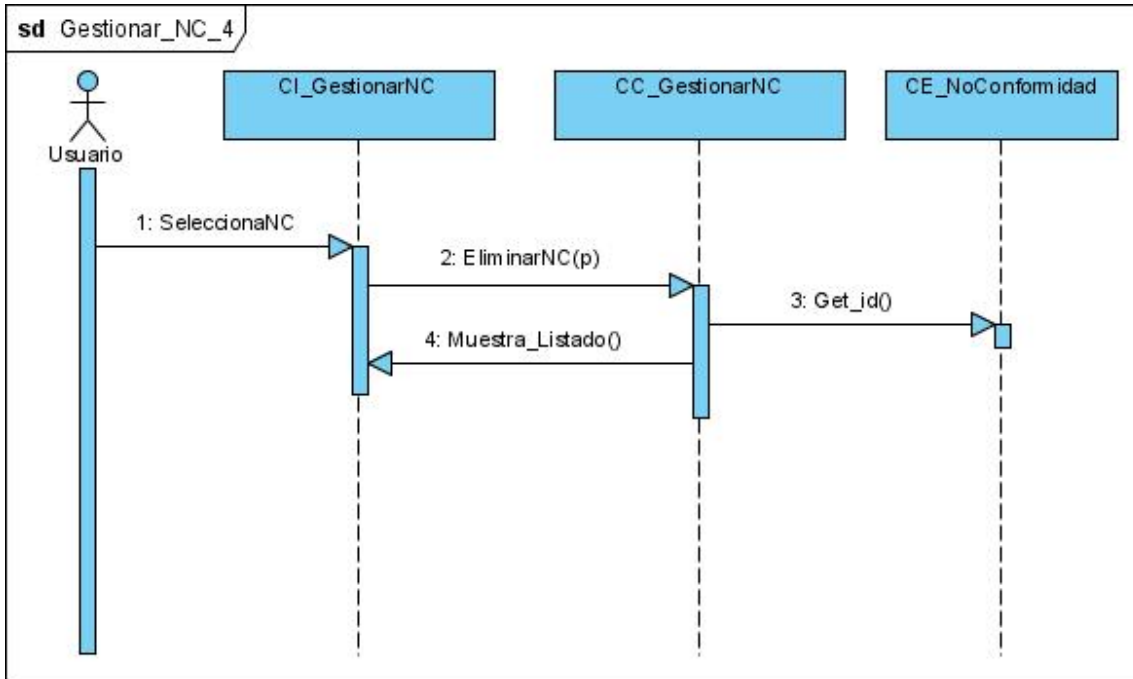


Fig. 40 - Diagrama de Secuencia "Gestionar No conformidad"

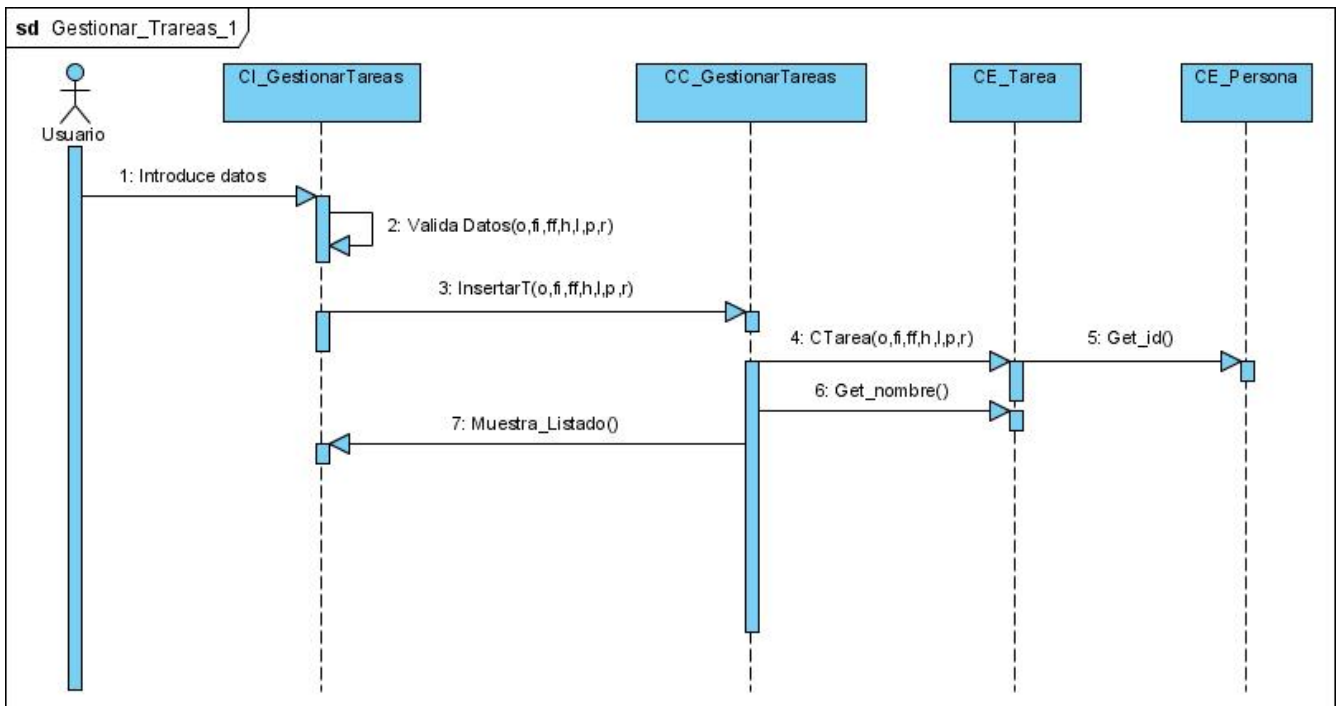
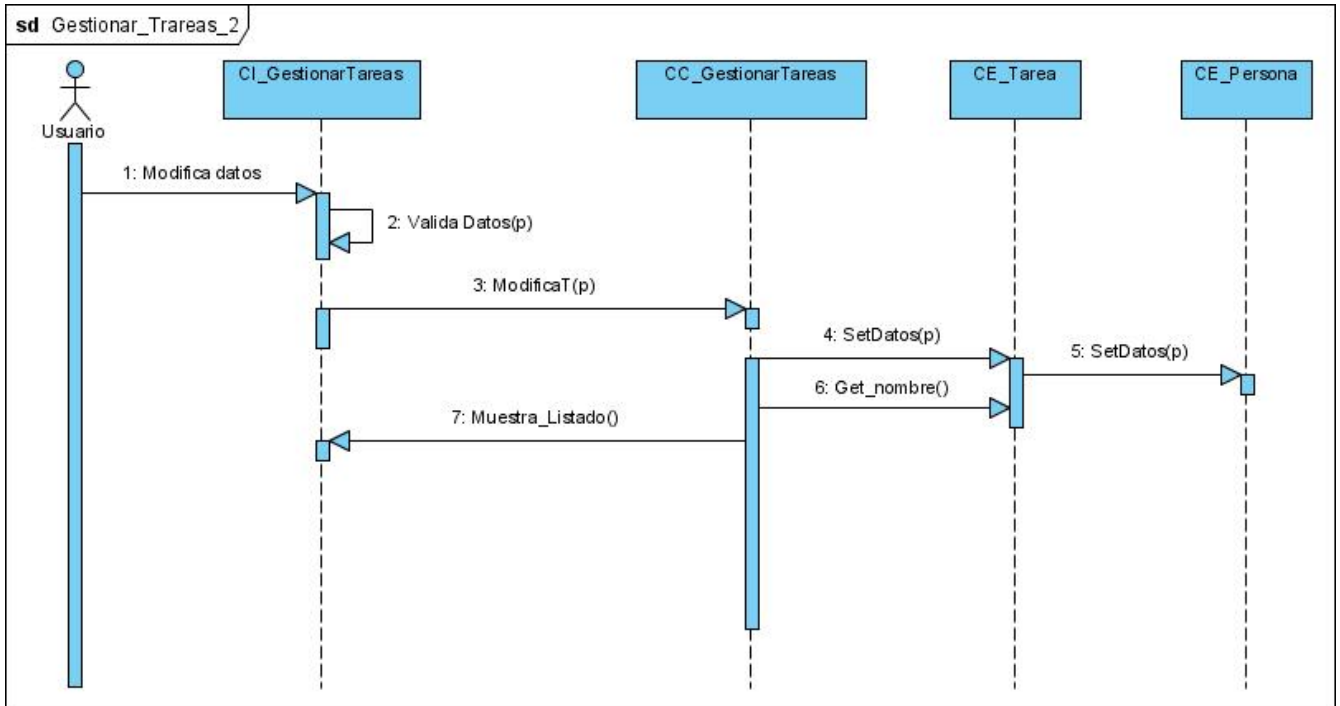
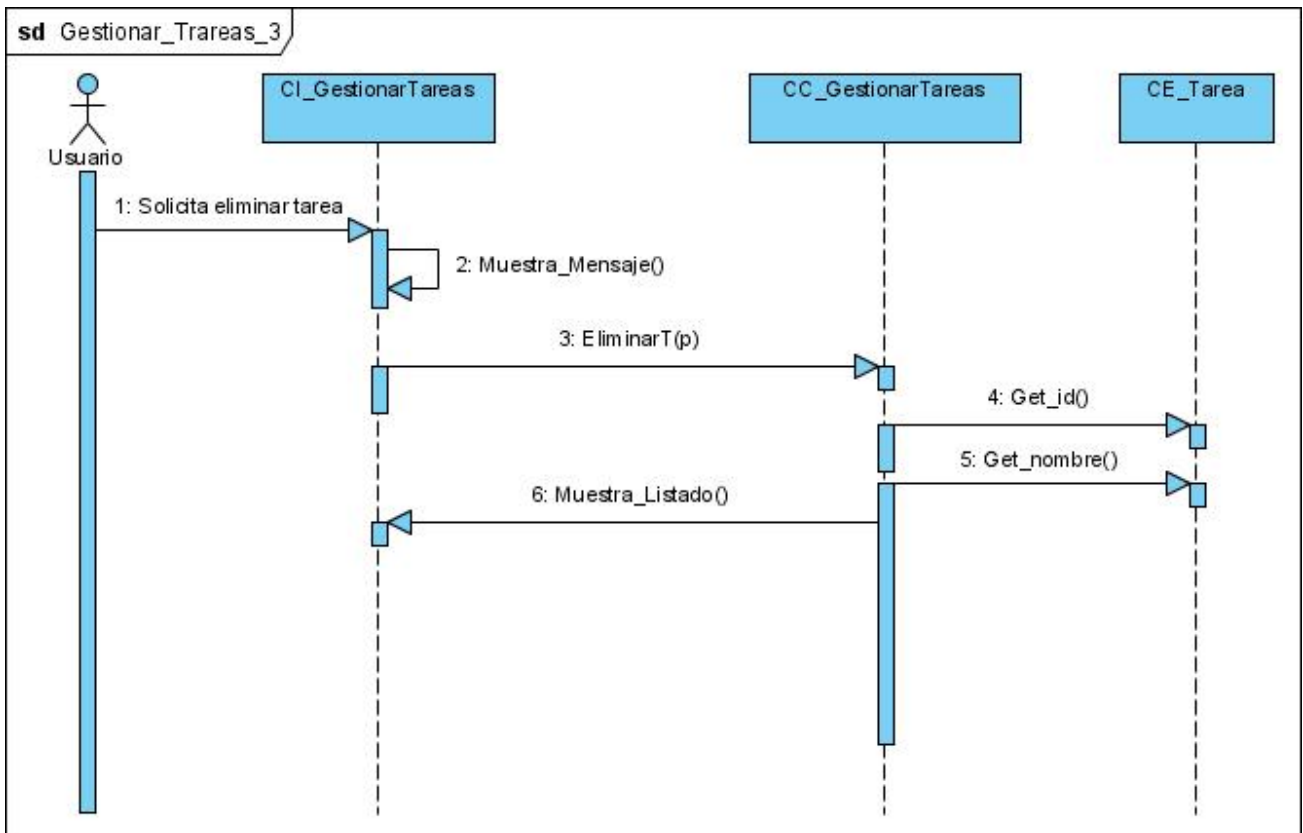


Fig. 41 - Diagrama de Secuencia "Gestionar Tareas"



**Fig. 42 - Diagrama de Secuencia "Gestionar Tareas"**



**Fig. 43 - Diagrama de Secuencia "Gestionar Tareas"**



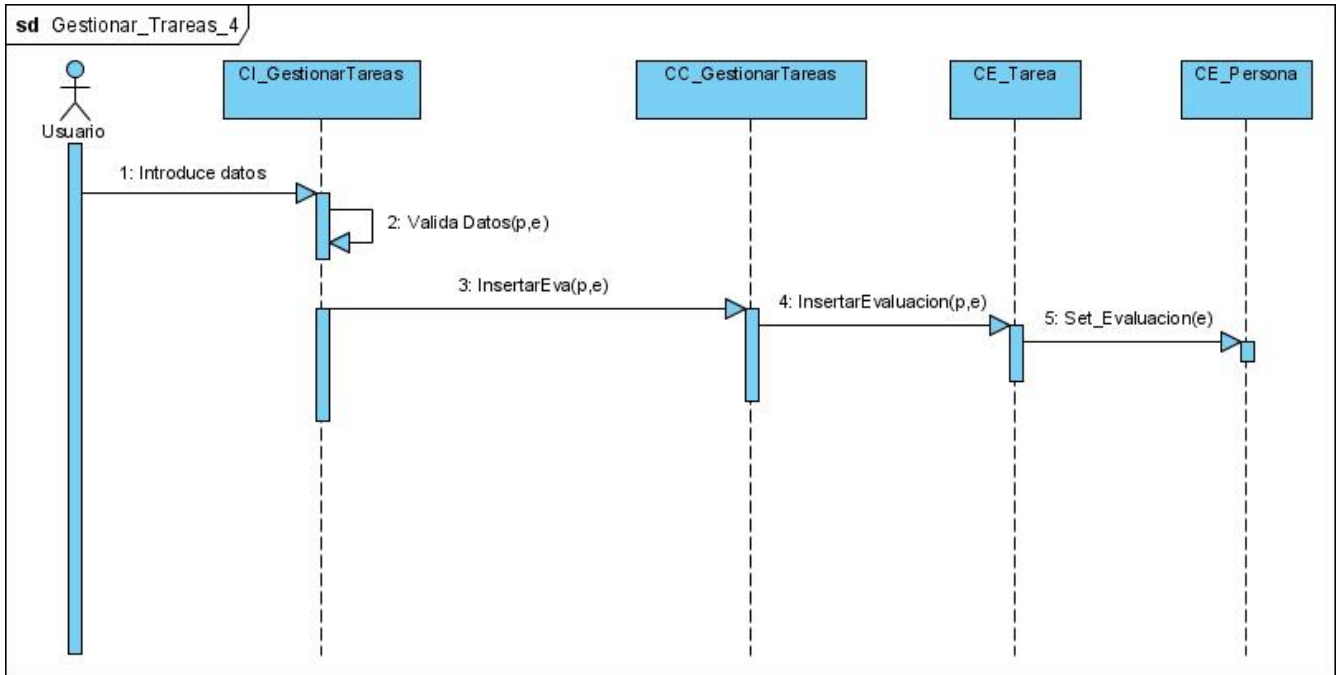


Fig. 44 - Diagrama de Secuencia "Gestionar Tareas"

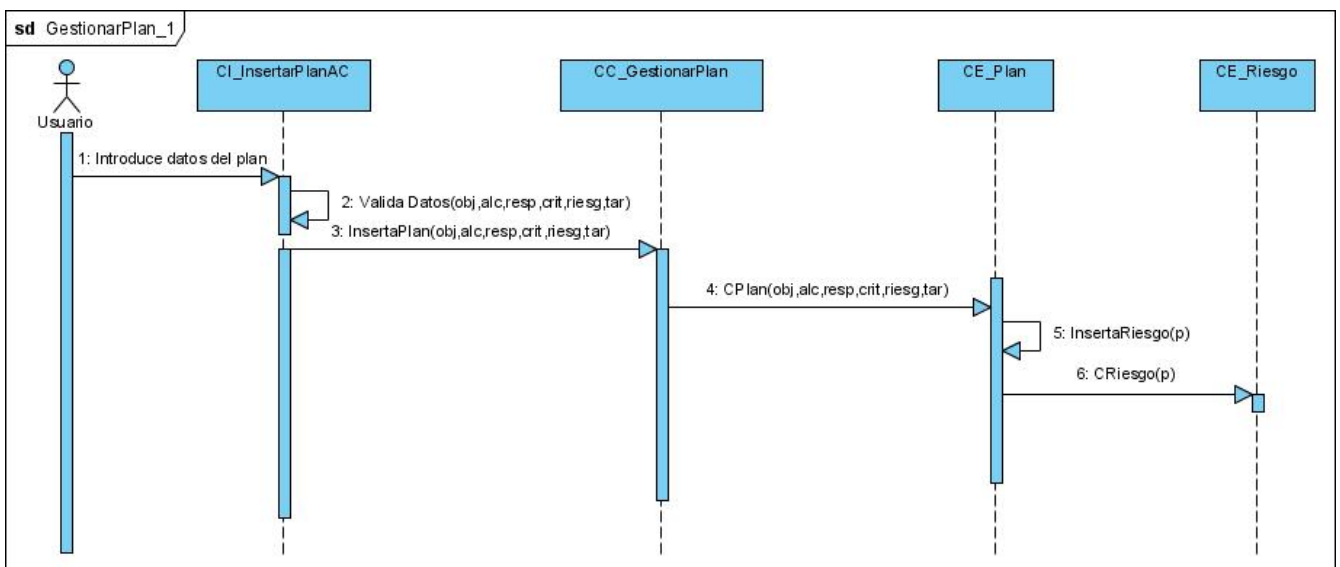


Fig. 45 - Diagrama de Secuencia "Gestionar Plan"

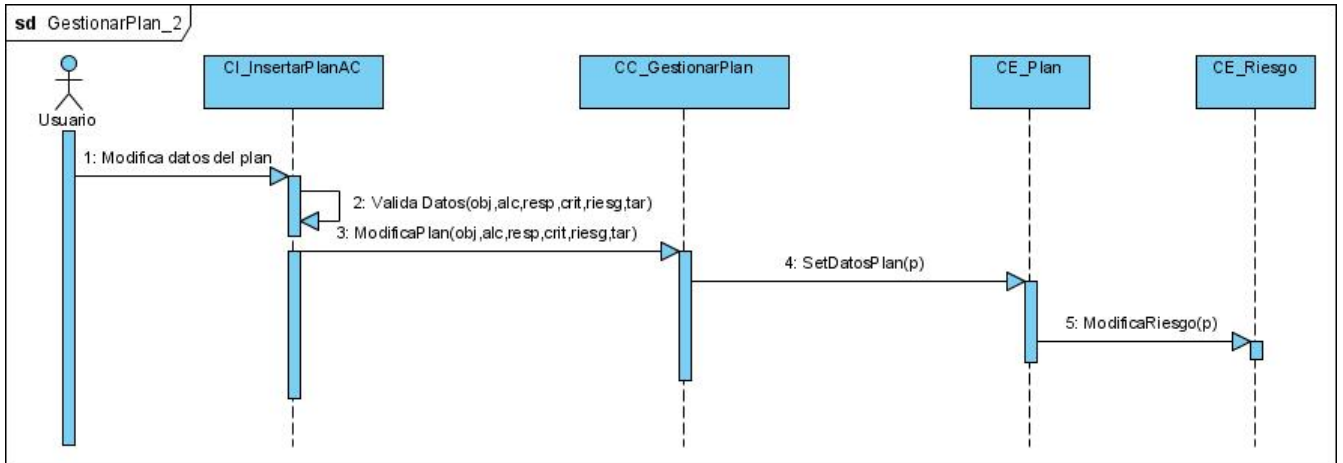


Fig. 46 - Diagrama de Secuencia "Gestionar Plan"

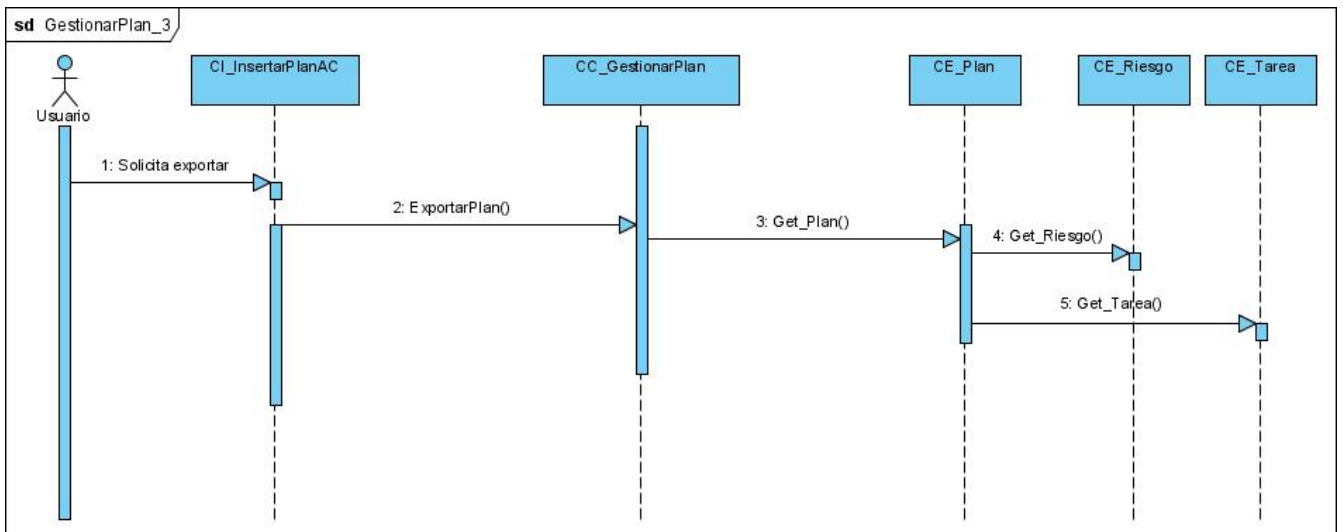
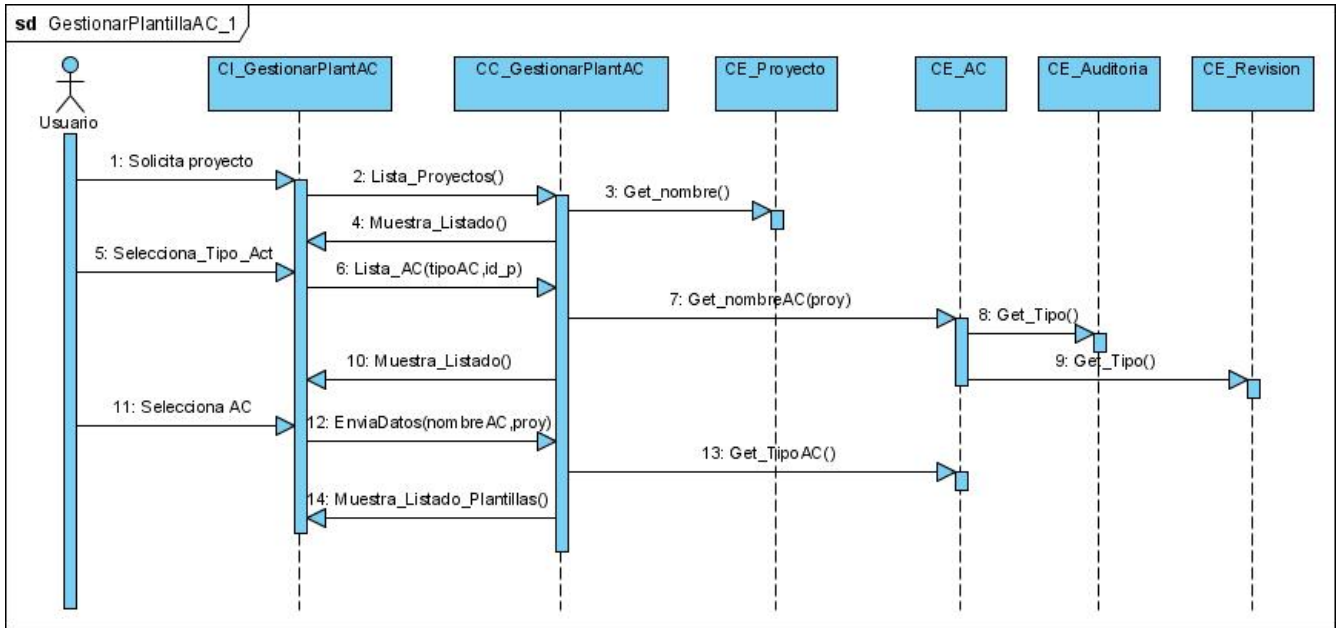


Fig. 47 - Diagrama de Secuencia "Gestionar Plan"



**Fig. 48 - Diagrama de Secuencia “Gestionar Datos de Actividades de Control”**

## 2.5 Diagramas de Clases del Diseño

Una clase de diseño es una construcción similar en la implementación del sistema. Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Los diagramas de clases de diseño del sistema está separado en casos de usos, es decir, cada caso de uso cuenta con un diagrama de clases, para hacer más fácil la comprensión. (Ingeniería de Software I 2008).

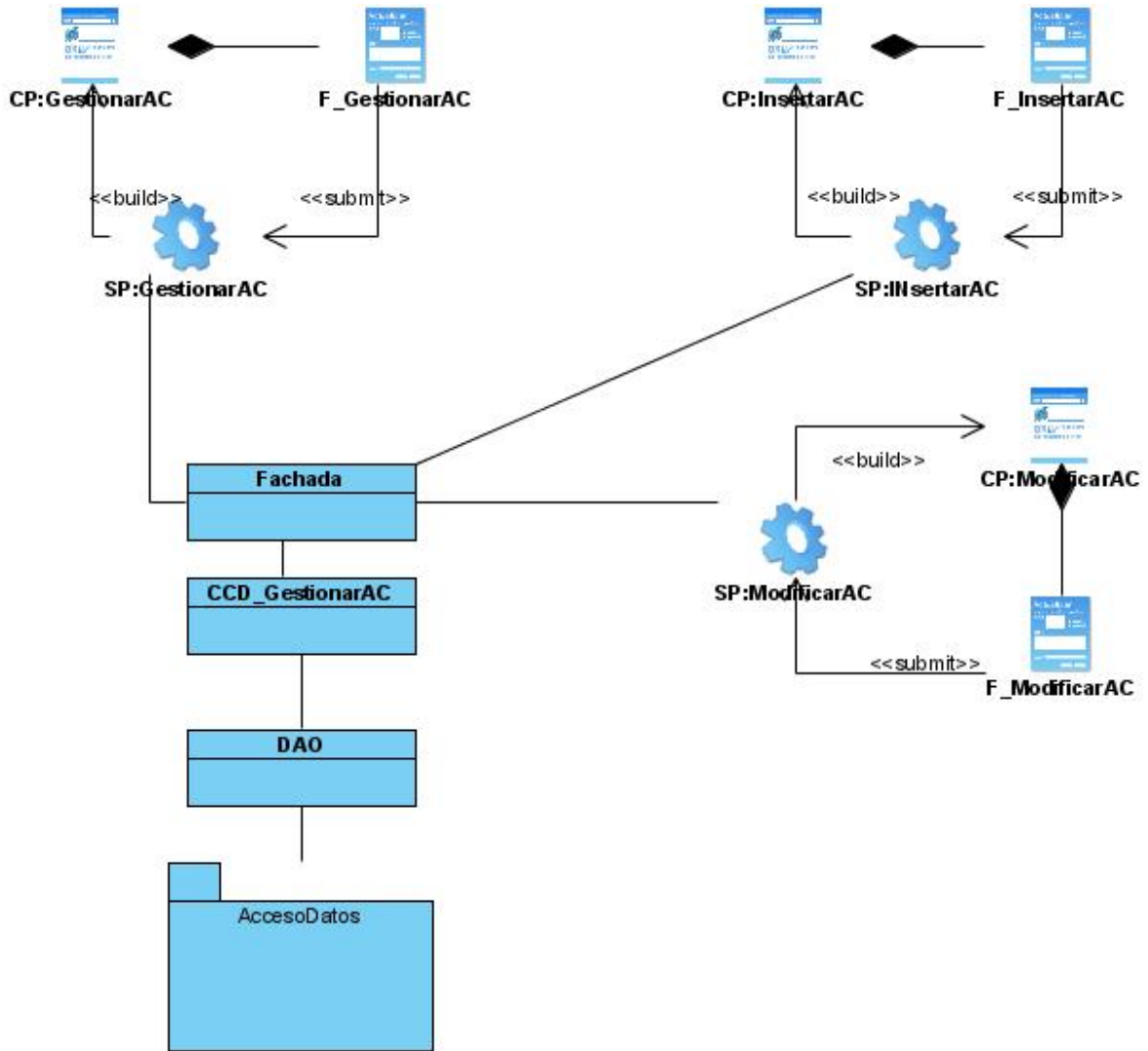


Fig. 49 - Diagrama de Clases del Diseño "Gestionar Actividad de Control"

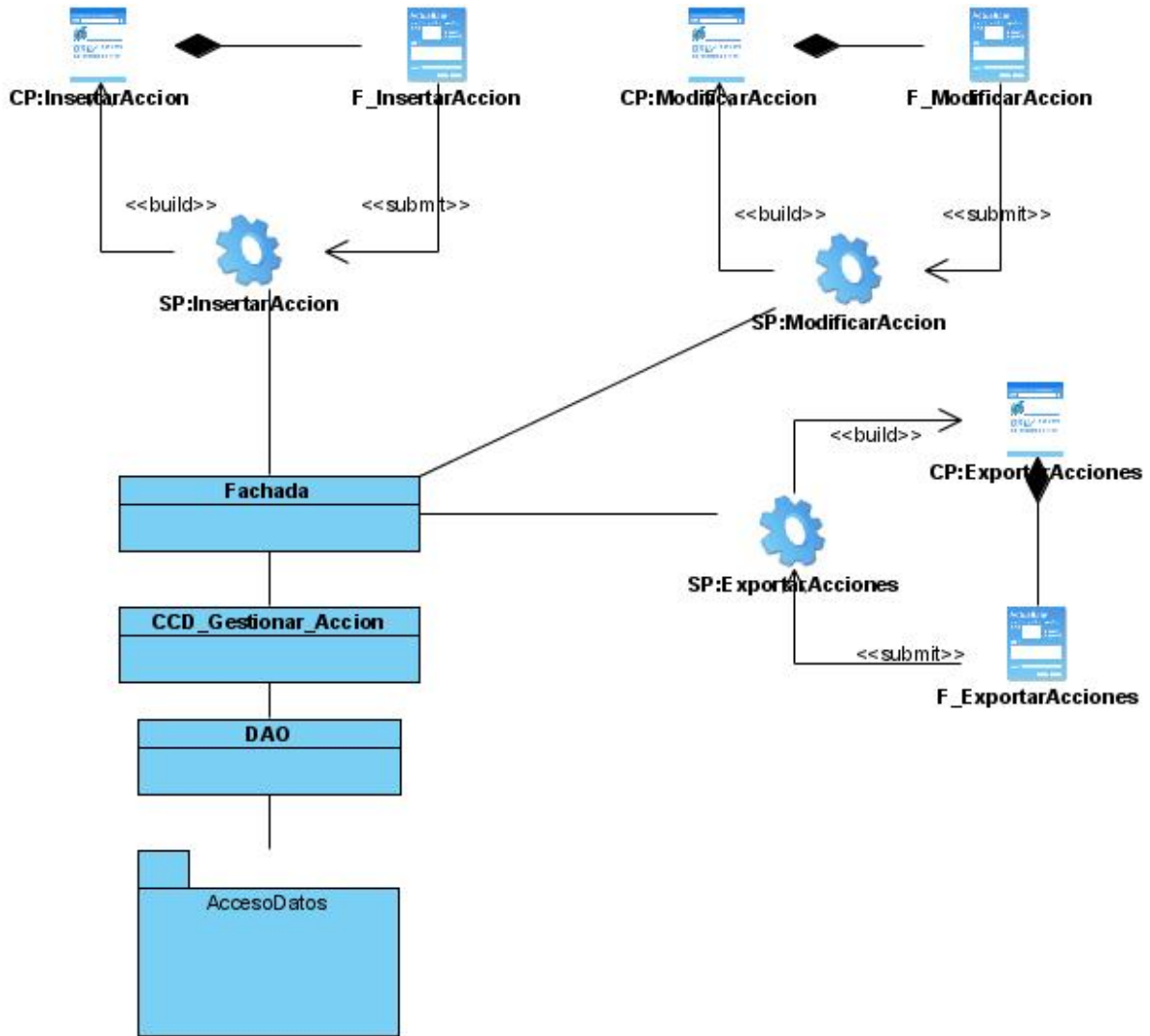


Fig. 50 - Diagrama de Clases del Diseño "Gestionar Acciones"

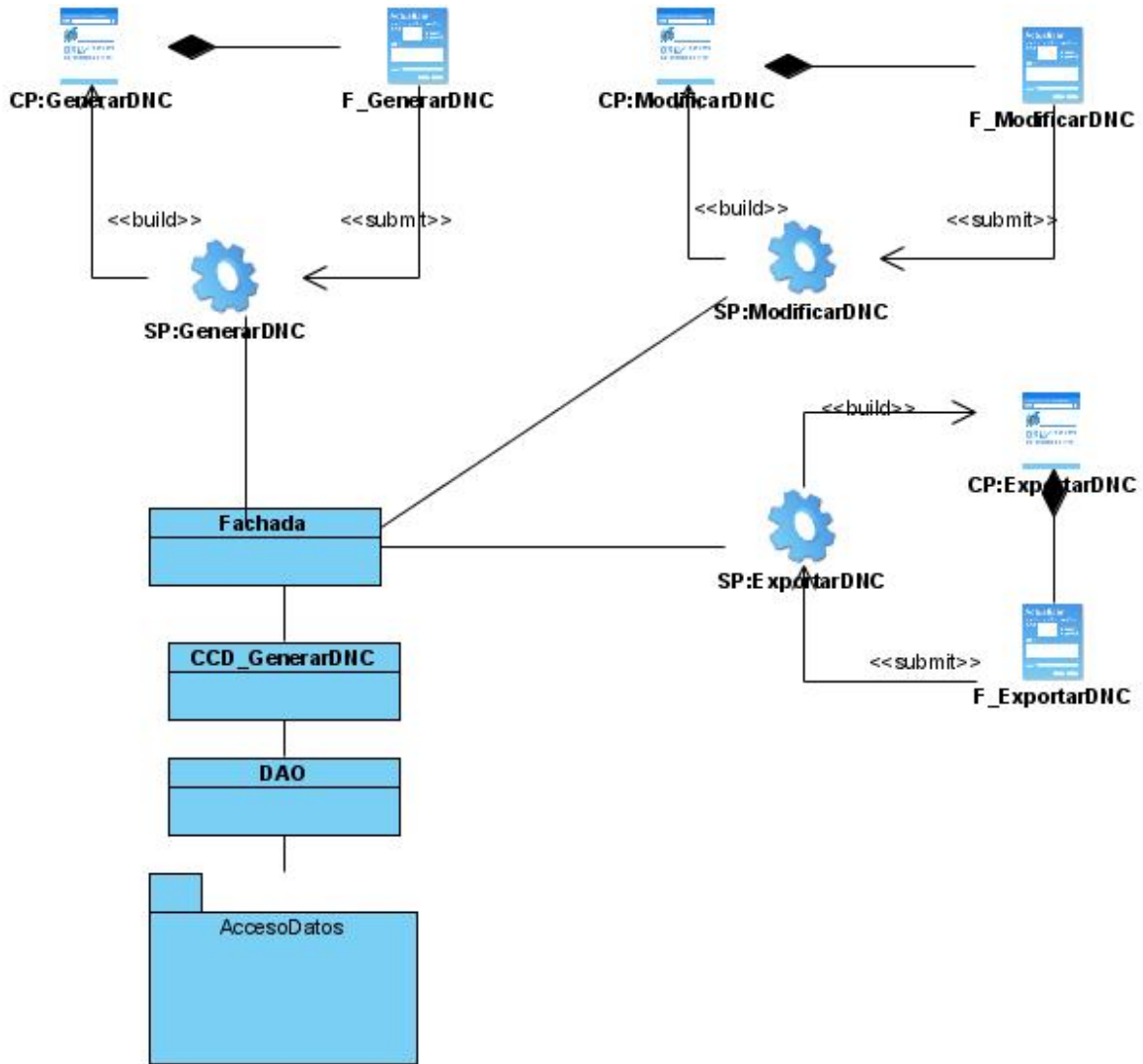


Fig. 51 - Diagrama de Clases del Diseño "Generar Informe de No Conformidades"

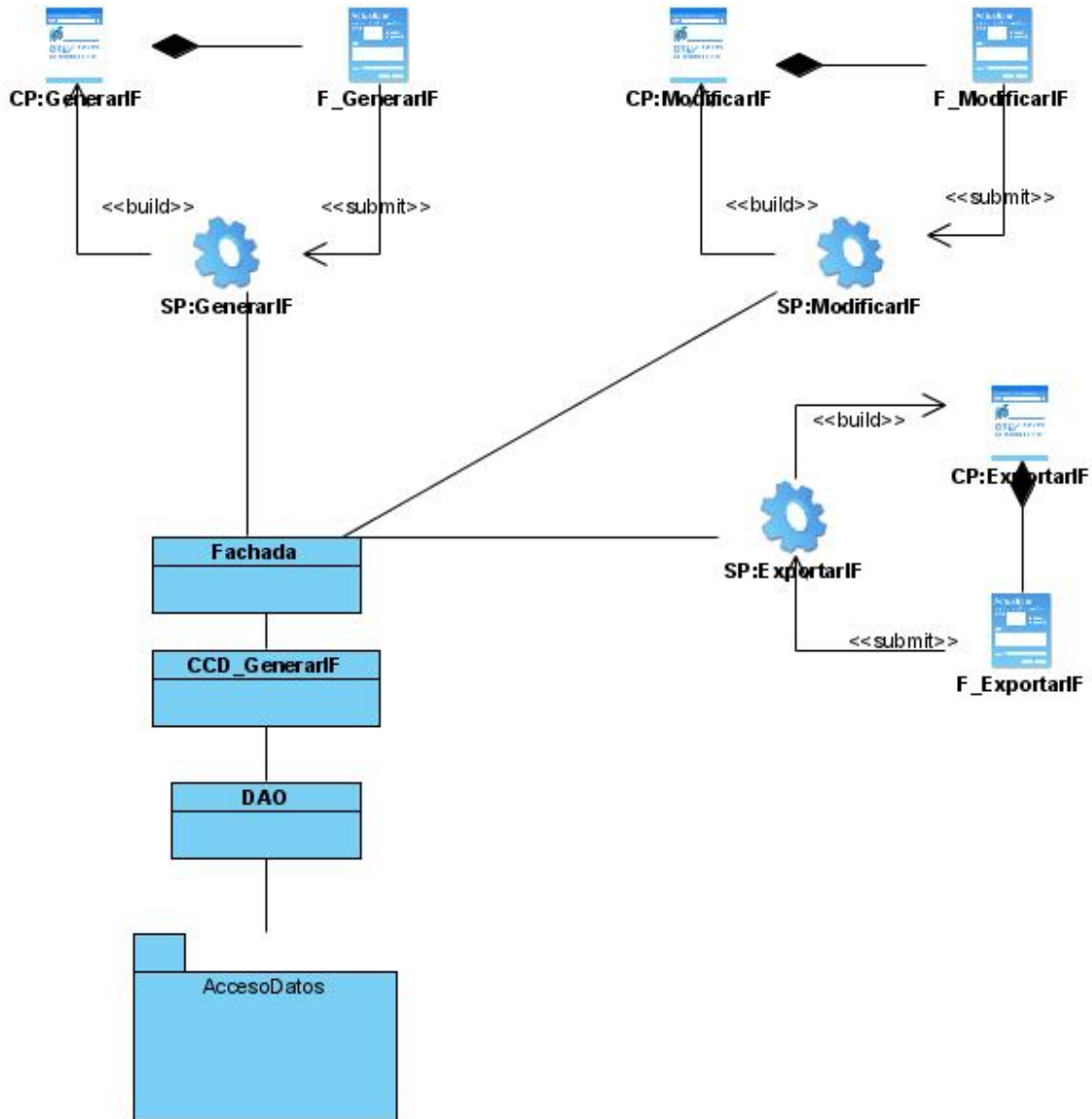


Fig. 52 - Diagrama de Clases del Diseño "Generar Informe Final"

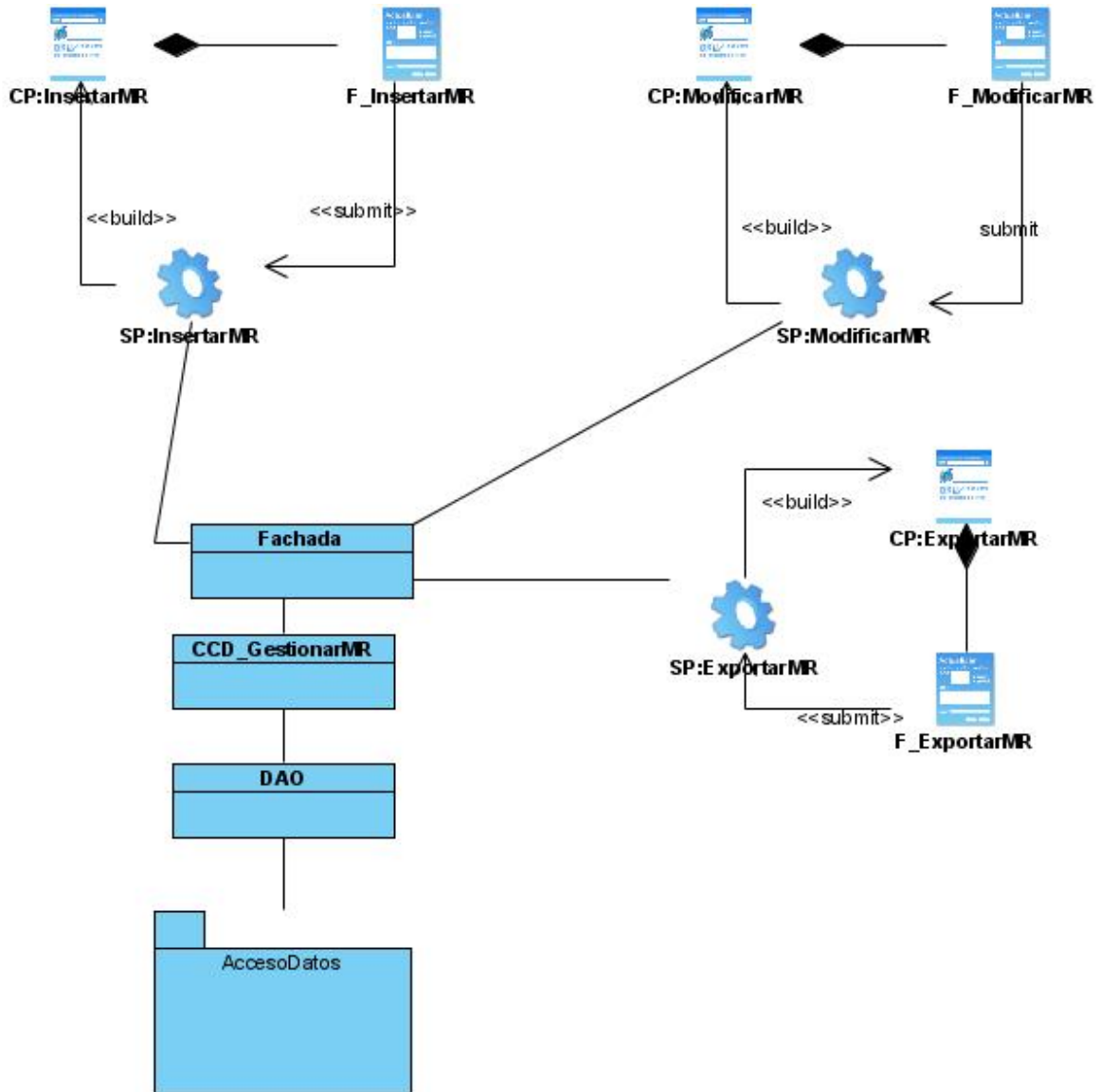


Fig. 53 - Diagrama de Clases del Diseño "Gestionar Minuta de Reunión"



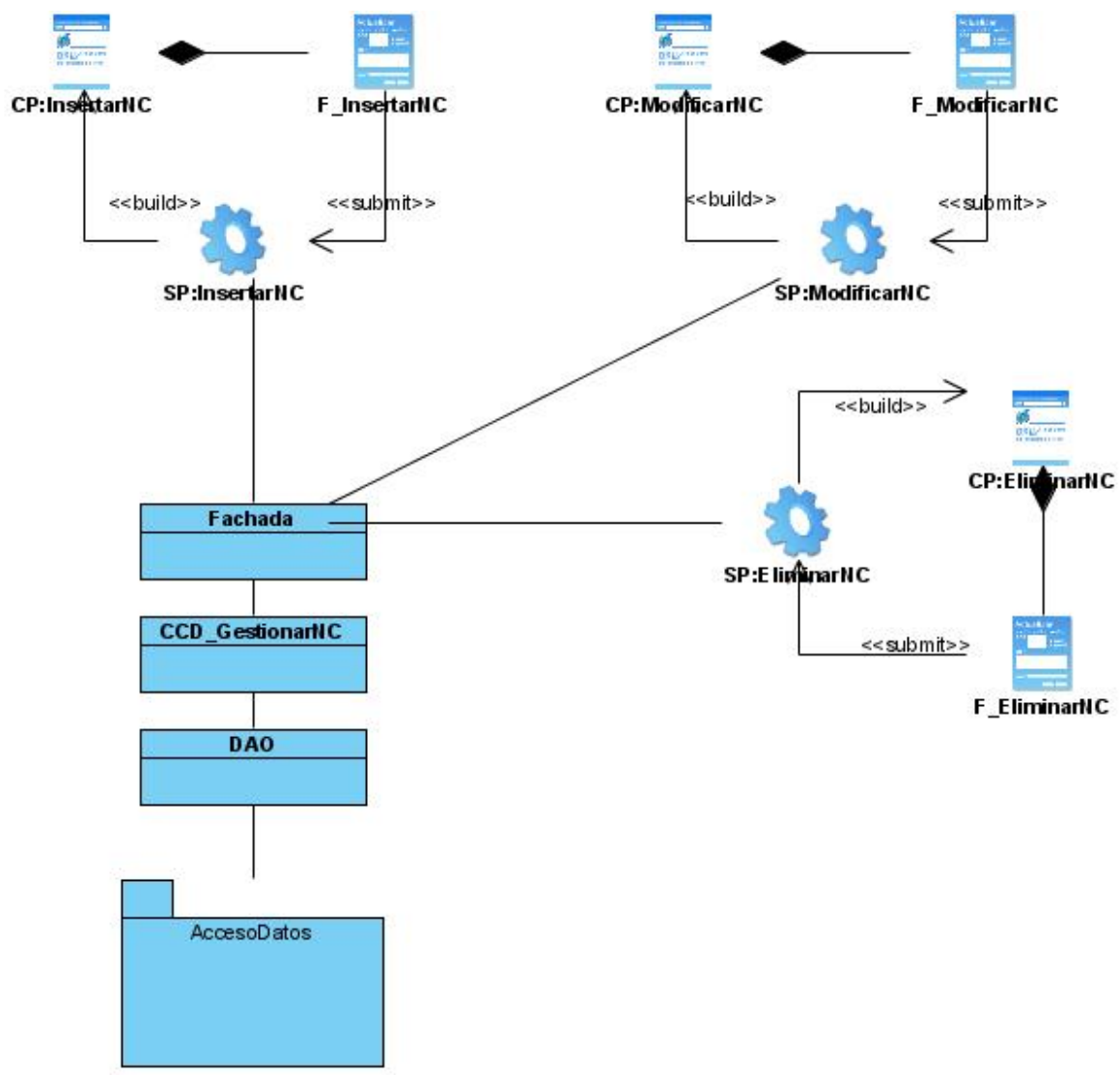


Fig. 54 - Diagrama de Clases del Diseño "Gestionar No Conformidad"

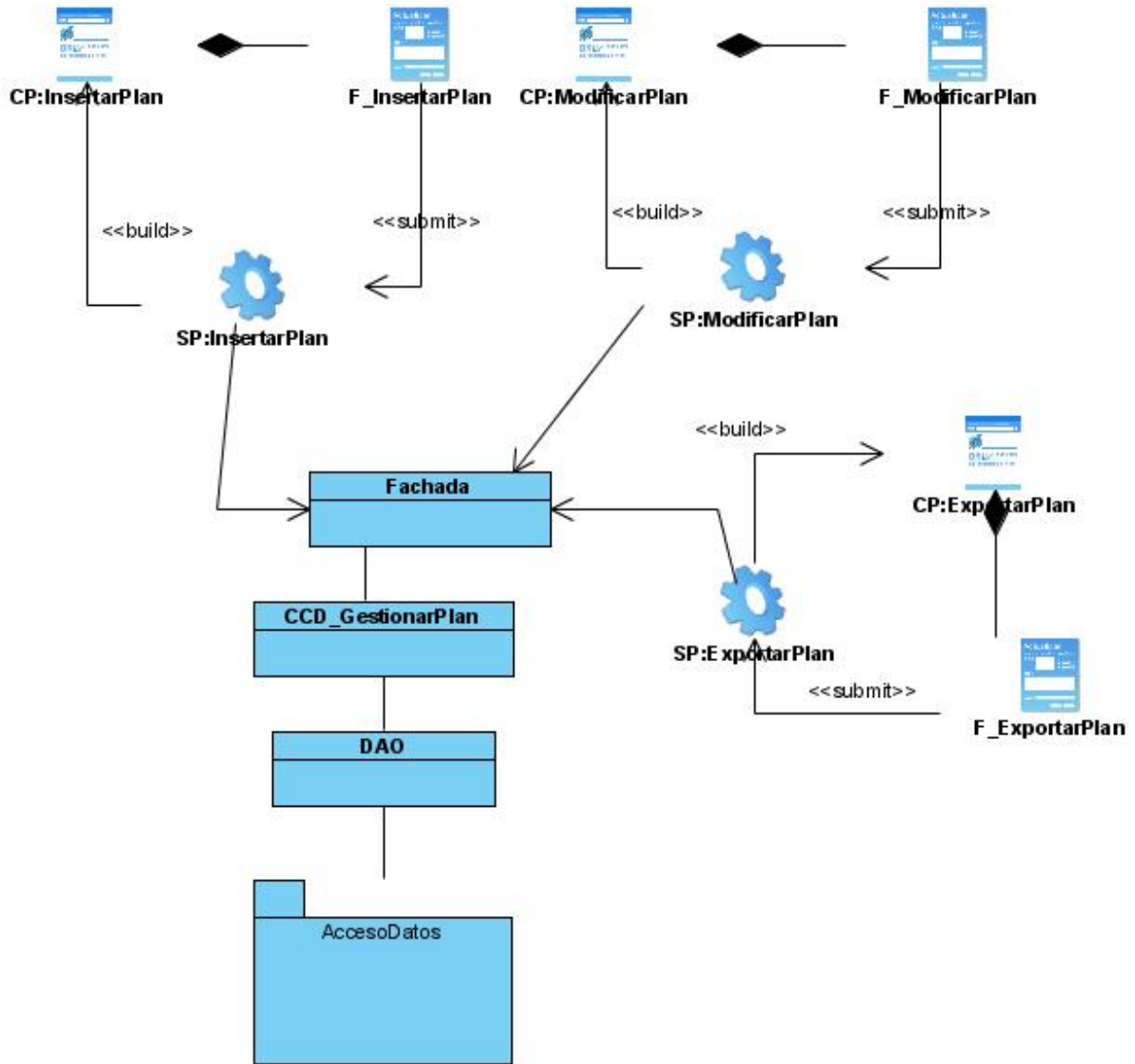


Fig. 55 - Diagrama de Clases del Diseño "Gestionar Plan"

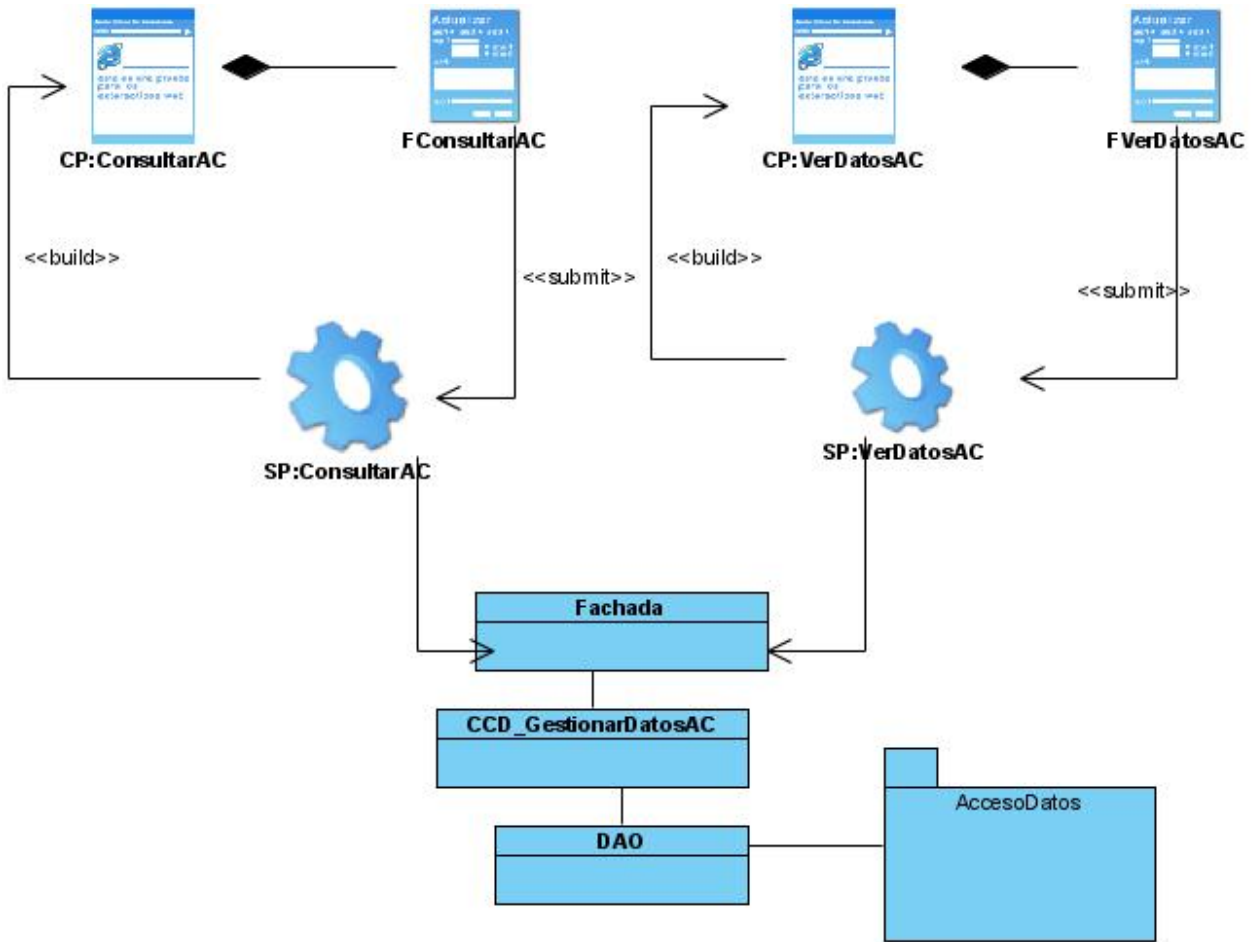


Fig. 56 - Diagrama de Clases del Diseño “Gestionar Plantillas de Actividades de Control”

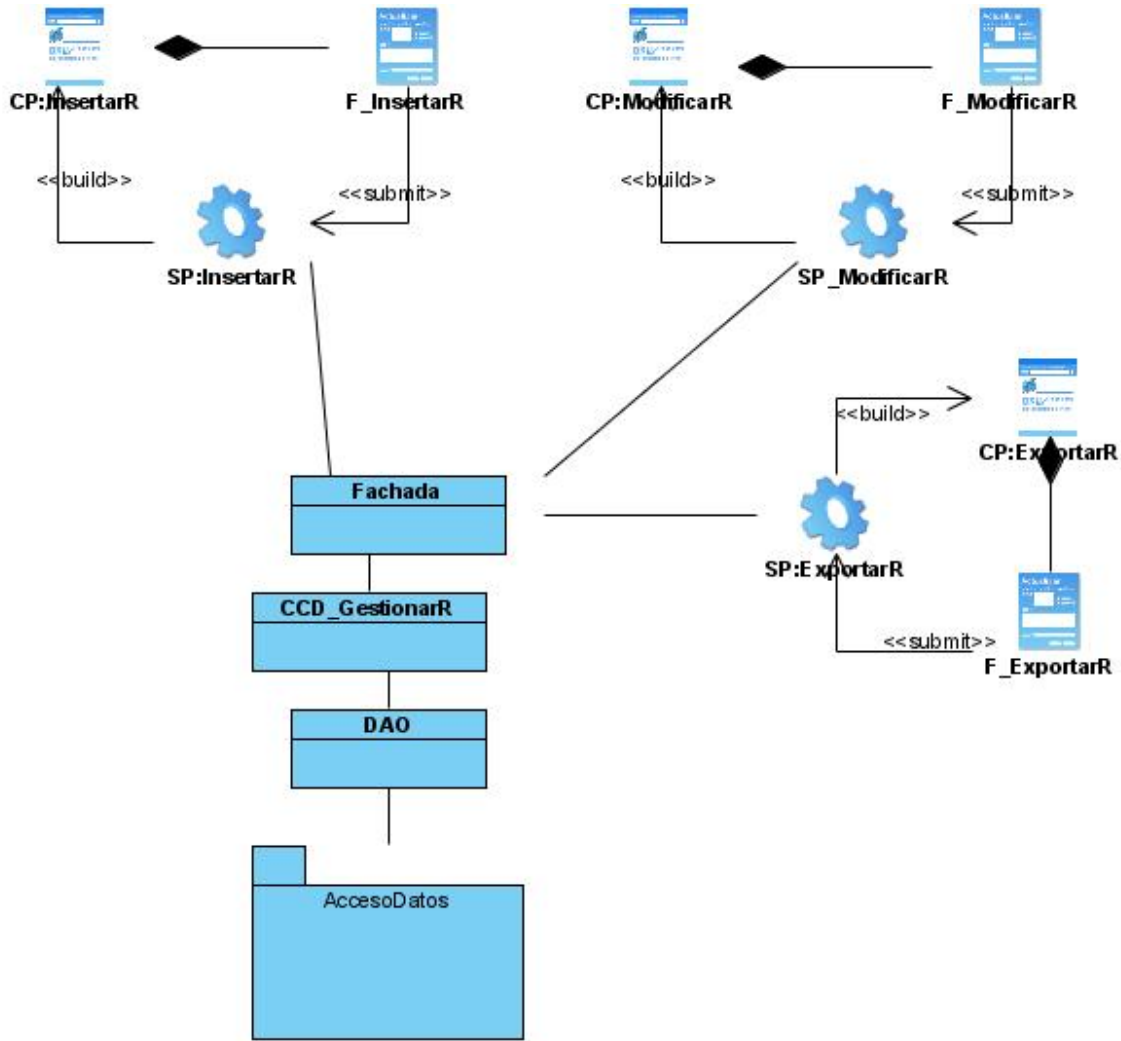


Fig. 57 - Diagrama de Clases del Diseño “Gestionar Recomendaciones”

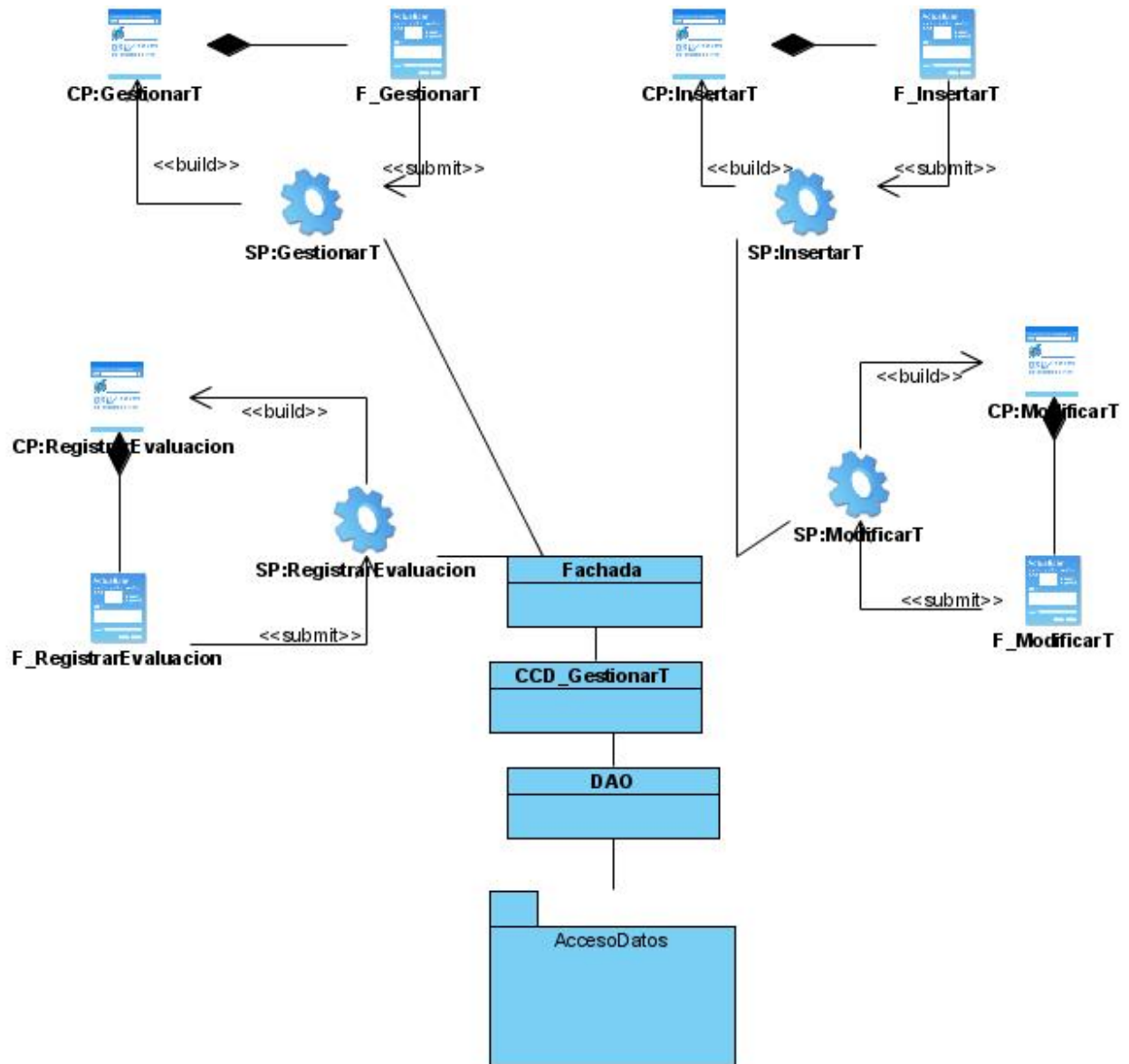


Fig. 58 - Diagrama de Clases del Diseño "Gestionar Tareas"

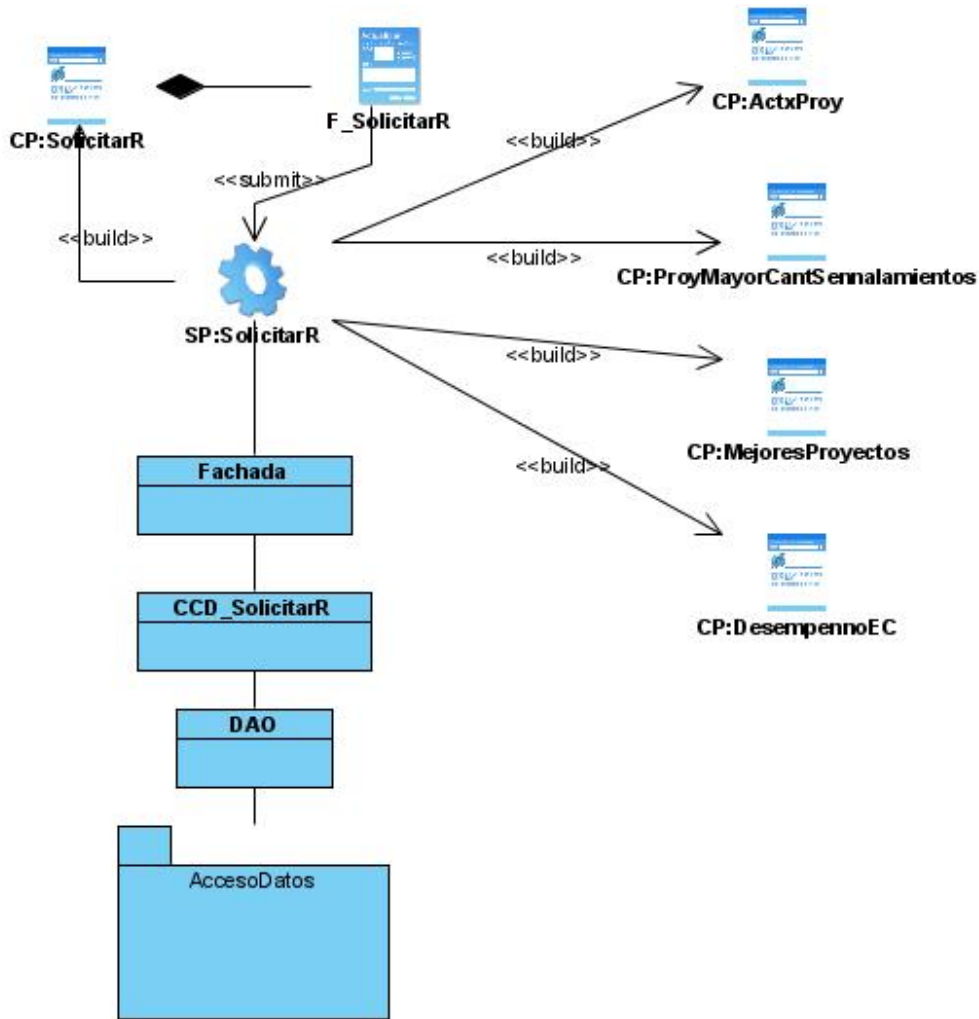


Fig. 59 - Diagrama de Clases del Diseño “Solicitar Reportes de Actividades de Control”

## 2.5.1 Descripciones de las clases del diseño

### Clases Controladoras

Nombre: CCD_GestionarNC	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	

Nombre:	Lista_Proyectos()
	Lista_AC(Id_Proj,TipoAC)
	Insertar_NC(lugar,lineamiento,descripción,impacto)
	Lista_NC(Id_Proj,Id_AC)
	Modificar_NC(Id_Proj, Id_AC ,lugar,lineamiento,descripción,impacto)
	Eliminar_NC(Id_NC,Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Accion</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Insertar_Accion(elemento,accion,impacto,estado)
	Lista_Acciones(Id_Proj,Id_AC)
	Modificar_Accion(Id_Proj, Id_AC ,elemento,accion,impacto,estado)
	Eliminar_Accion(Id_A,Id_Proj,Id_AC)
	Exportar_Accion(Id_Proj,Id_AC)

<b>Nombre: CCD_GenerarDNC</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>

<b>Para cada responsabilidad:</b>	
Nombre:	GenerarDNC(Id_Proj,Id_AC)
	Modificar_DNC(Id_Proj, Id_AC ,elemento,accion,impacto,estado)
	Eliminar_DNC(Id_Proj,Id_AC)
	Exportar_DNC(Id_Proj,Id_AC)

<b>Nombre: CCD_GenerarIF</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	GenerarIF(Id_Proj,Id_AC)
	Modificar_IF(Id_Proj, Id_AC ,resumen,acta_evaluacion)
	Eliminar_IF(Id_Proj,Id_AC)
	Exportar_IF(Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Actividad_Control</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>



<b>Para cada responsabilidad:</b>	
Nombre:	Lista_Proyectos()
	Lista_AC(Id_Proj,TipoAC)
	Insertar_AC(liderAC,equipoAC)
	Modificar_AC(Id_Proj, Id_AC ,liderAC,equipoAC)
	Eliminar_AC(Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_MR</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Insertar_MR(hora_inicio,hora_fin,asunto,asistentes,ausentes,orden_dia,acuerdos)
	Modificar_MR(Id_Proj, <span style="float: right;">Id_AC</span> ,hora_inicio,hora_fin,asunto,asistentes,ausentes,orden_dia,acuerdos)
	Eliminar_MR(Id_MR,Id_Proj,Id_AC)
	Exportar_MR(Id_MR ,Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Plan</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>

<b>Para cada responsabilidad:</b>	
Nombre:	Insertar_Plan(objetivo,alcance,criteriosAC,Riesgos,Tareas)
	Modificar_Plan(Id_Proj, Id_AC ,objetivo,alcance,criteriosAC,Riesgos,Tareas)
	Exportar_Plan(Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Datos_AC</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Lista_Proyectos()
	Lista_AC(Id_Proj,TipoAC)
	Mostrar_Datos_AC(Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Recomendacion</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Insertar_R(criterio_eval,elemento,recomendacion,impacto,etapa,estado)
	Modificar_R(Id_Proj, <span style="float: right;">Id_AC</span> ,criterio_eval,elemento,recomendacion,impacto,etapa,estado)

	Eliminar_R(Id_Proj,Id_AC,Id_R)
	Exportar_R(Id_Proj,Id_AC)

<b>Nombre: CCD_Gestionar_Tareas</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	Lista_Tareas(Id_Proj, Id_AC)
	Insertar_Tarea(nombre,responsable,fecha_cumplimiento)
	Modificar_Tarea(Id_Proj, Id_AC ,nombre,responsable,fecha_cumplimiento)
	Eliminar_Tarea(Id_T,Id_Proj,Id_AC)
	Evaluar_Equipo(eval)

<b>Nombre: CCD_Solicitar_Reportes</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	ACxProy(Id_proy)
	Ver_Sennalamientos(Id_proy, Id_Act)

	Proy_Mas_Sennalamientos()
	Mejores_Proj()
	Desempeño_Equipo()

### Clases Entidad

<b>Nombre: CActividad_Control</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$Id_AC	
\$nombre	
\$tipo_AC	
\$proyecto	
\$lider	
\$equipo	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_Id_AC()
	Get_nombre()
	Get_tipo_AC()
	Get_proyecto()
	Get_lider()
	Get_equipo()
	Set_Id_AC(pid)

	Set_nombre(pnom)
	Set_tipo_AC(ptipo)
	Set_proyecto(pproy)
	Set_lider(plider)
	Set_equipo(pequipo)

<b>Nombre: CAuditoria</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$evaluacion	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_evaluacion()
	Set_evaluacion(peval)

<b>Nombre: CRevision</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$recomendaciones	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_Recomendaciones()
	Set_Recomendaciones(precomendacion)

<b>Nombre: CMinuta_Reunion</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$Id_MR	
\$autor	
\$fecha	
\$lugar	
\$hora_inicio	
\$hora_fin	
\$asunto	
\$asistentes	
\$ausentes	
\$orden_dia	
\$acuerdos	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_Id_MR()
	Get_autor()
	Get_fecha()
	Get_lugar()
	Get_hora_inicio()
	Get_hora_fin()

	Get_asunto()
	Get_asistentes()
	Get_ausentes()
	Get_orden_dia()
	Get_acuerdos()
	Set_Id_MR(p)
	Set_autor(p)
	Set_fecha(p)
	Set_lugar(p)
	Set_hora_inicio(p)
	Set_hora_fin(p)
	Set_asunto(p)
	Set_asistentes(p)
	Set_ausentes(p)
	Set_orden_dia(p)
	Set_acuerdos(p)

<b>Nombre: CAccion</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$id	

\$accion	
\$impacto	
\$estado	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()
	Get_accion()
	Get_impacto()
	Get_estado()
	Set_id(p)
	Set_accion(p)
	Set_impacto(p)
	Set_estado(p)

<b>Nombre: CNo_Conformidad</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$id	
\$lugar	
\$aspecto_incumplido	
\$descripcion	
\$impacto	



<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()
	Get_lugar()
	Get_impacto()
	Get_aspecto_incumplido()
	Get_descripcion()
	Set_id(p)
	Set_lugar(p)
	Set_impacto(p)
	Set_aspecto_incumplido(p)
	Set_descripcion(p)

<b>Nombre: CInforme_Final</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$fecha	
\$autor	
\$resumen	
\$evaluacion	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_fecha()

	Get_autor()
	Get_resumen()
	Get_evaluacion()
	Set_fecha(p)
	Set_autor(p)
	Set_resumen(p)
	Set_evaluacion(p)

<b>Nombre: CPlan</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$objetivo	
\$alcance	
\$responsables	
\$criterios	
\$riesgos	
\$areas	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_objetivo()
	Get_alcance()
	Get_responsables()

	Get_criterios()
	Get_riesgos()
	Get_Tareas()
	Set_objetivo(p)
	Set_alcance(p)
	Set_responsables(p)
	Set_criterios(p)

<b>Nombre: CRiesgo</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$id	
\$identificacion	
\$objetivos_control	
\$analisis_cualitativo	
\$seguimiento	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()
	Get_identificacion()
	Get_objetivos_control()
	Get_analisis_cualitativo()

	Get_seguimiento()
	Set_id(p)
	Set_identificacion(p)
	Set_objetivos_control(p)
	Set_analisis_cualitativo(p)
	Set_seguimiento(p)

<b>Nombre: CTarea</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$id	
\$nombre	
\$objetivo	
\$fecha_inicio	
\$fecha_fin	
\$hora	
\$lugar	
\$participantes	
\$responsable	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()

	Get_nombre()
	Get_objetivo()
	Get_fecha_inicio()
	Get_fecha_fin()
	Get_hora()
	Get_lugar()
	Get_participantes()
	Get_responsable()
	Set_id(p)
	Set_nombre(p)
	Set_objetivo(p)
	Set_fecha_inicio(p)
	Set_fecha_fin(p)
	Set_hora(p)
	Set_lugar(p)
	Set_participantes(p)
	Set_responsable(p)

<b>Nombre: CPersona</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>

\$id	
\$nombre	
\$apellidos	
\$cargo	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()
	Get_nombre()
	Get_apellidos()
	Get_cargo()
	Set_id(p)
	Set_nombre(p)
	Set_apellidos(p)
	Set_cargo(p)

<b>Nombre: CRecomendacion</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
\$id	
\$criterio	
\$elemento	
\$recomendaciones	

\$impacto	
\$etapa_deteccion	
\$estado	
<b>Para cada responsabilidad:</b>	
Nombre:	Get_id()
	Get_criterio()
	Get_elemento()
	Get_recomendaciones()
	Get_impacto()
	Get_etapa_deteccion()
	Get_estado()
	Set_id(p)
	Set_criterio(p)
	Set_elemento(p)
	Set_recomendaciones(p)
	Set_impacto(p)
	Set_etapa_deteccion(p)
	Set_estado(p)

<b>Nombre: CProyecto</b>
<b>Tipo de clase: Entidad</b>

Atributo		Tipo
\$id		
\$nombre		
\$lider		
\$asesor_calidad		
<b>Para cada responsabilidad:</b>		
Nombre:	Get_id()	
	Get_nombre()	
	Get_lider()	
	Get_asesor_calidad()	
	Set_id(p)	
	Set_nombre(p)	
	Set_lider(p)	
	Set_asesor_calidad(p)	

<b>Nombre: CLista_Chequeo</b>		
<b>Tipo de clase: Entidad</b>		
Atributo		Tipo
\$pregunta		
\$peso		
\$lineamiento_asoiado		



<b>Para cada responsabilidad:</b>	
Nombre:	Get_pregunta()
	Get_peso()
	Get_lineamiento()
	Set_pregunta(p)
	Set_peso(p)
	Set_lineamiento(p)

## 2.5.2 Subsistema de acceso a datos

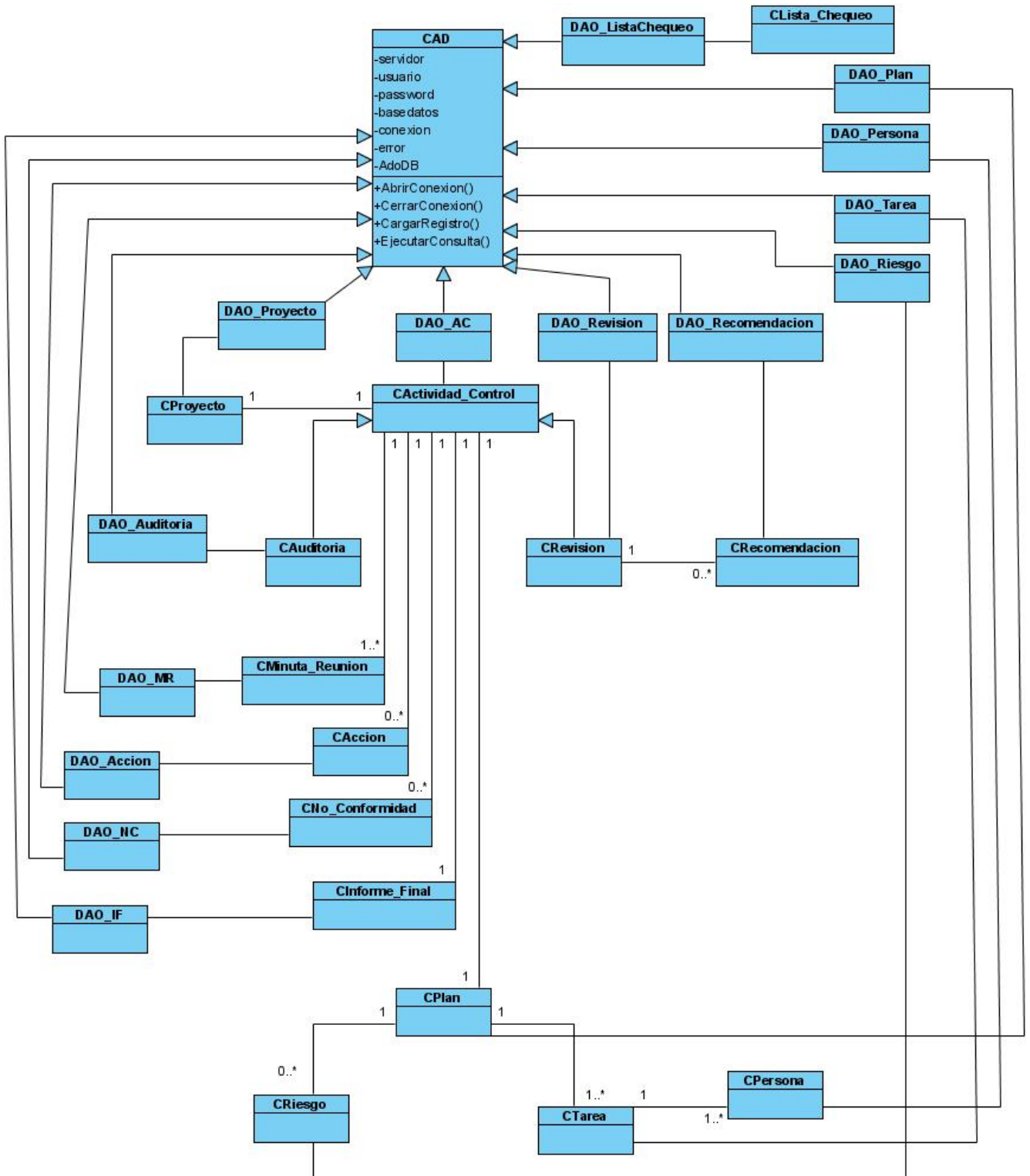


Fig. 60 - Diagrama de Clases del Subsistema de acceso a datos

## 2.6 Diagrama de clases persistentes

Un diagrama de clases persistentes es justo un tipo de diagrama que muestra las clases que manejan datos persistentes, es decir, que deben ser almacenados en algún repositorio como una base de datos relacional. (Ingeniería de Software II 2008).

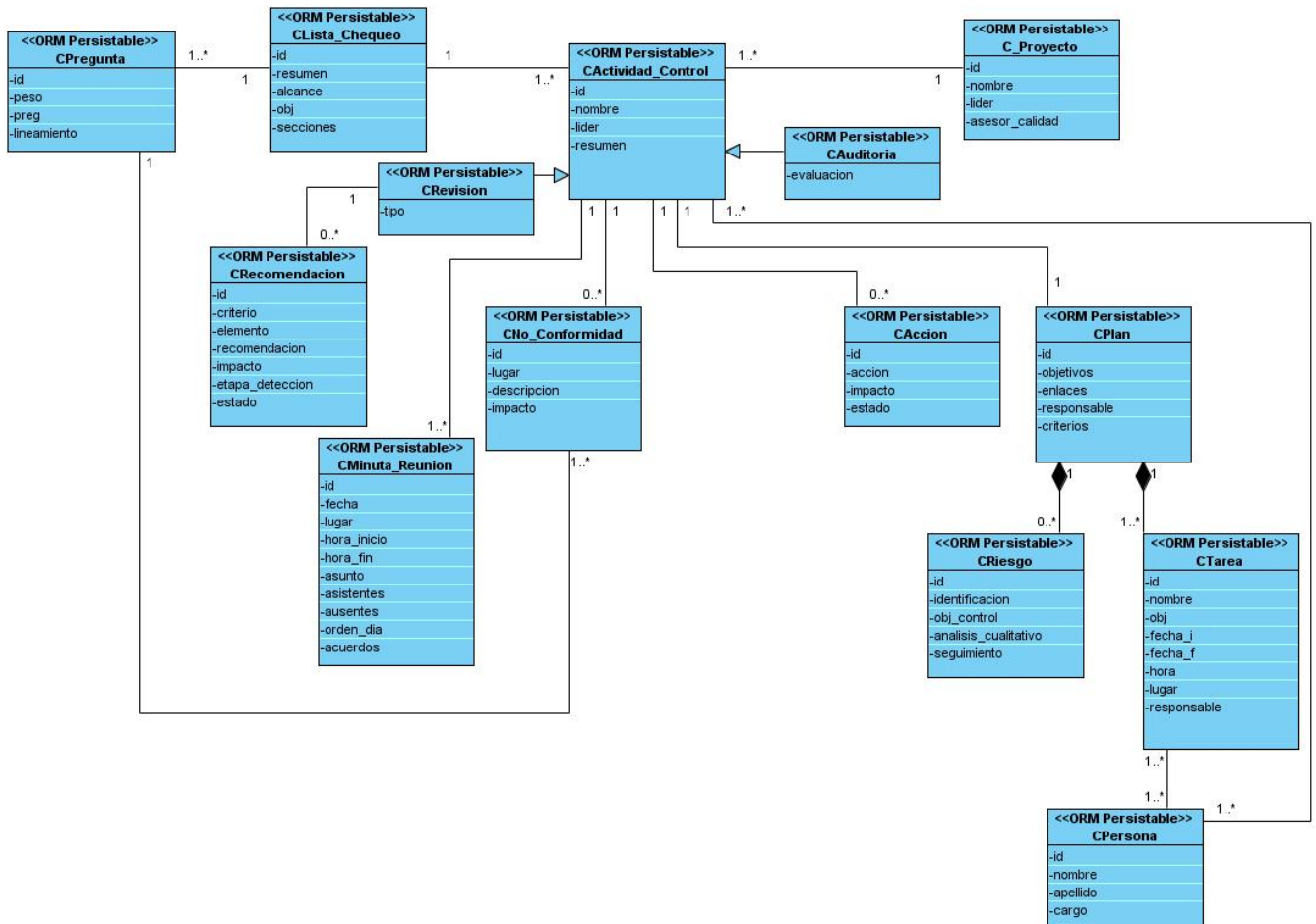


Fig. 61 - Diagrama de clases persistentes

## 2.7 Modelo físico de datos

El modelo entidad relación describe la representación lógica y física de datos persistentes en el sistema. A continuación se muestra el modelo entidad relación. (Ingeniería de Software II 2008).

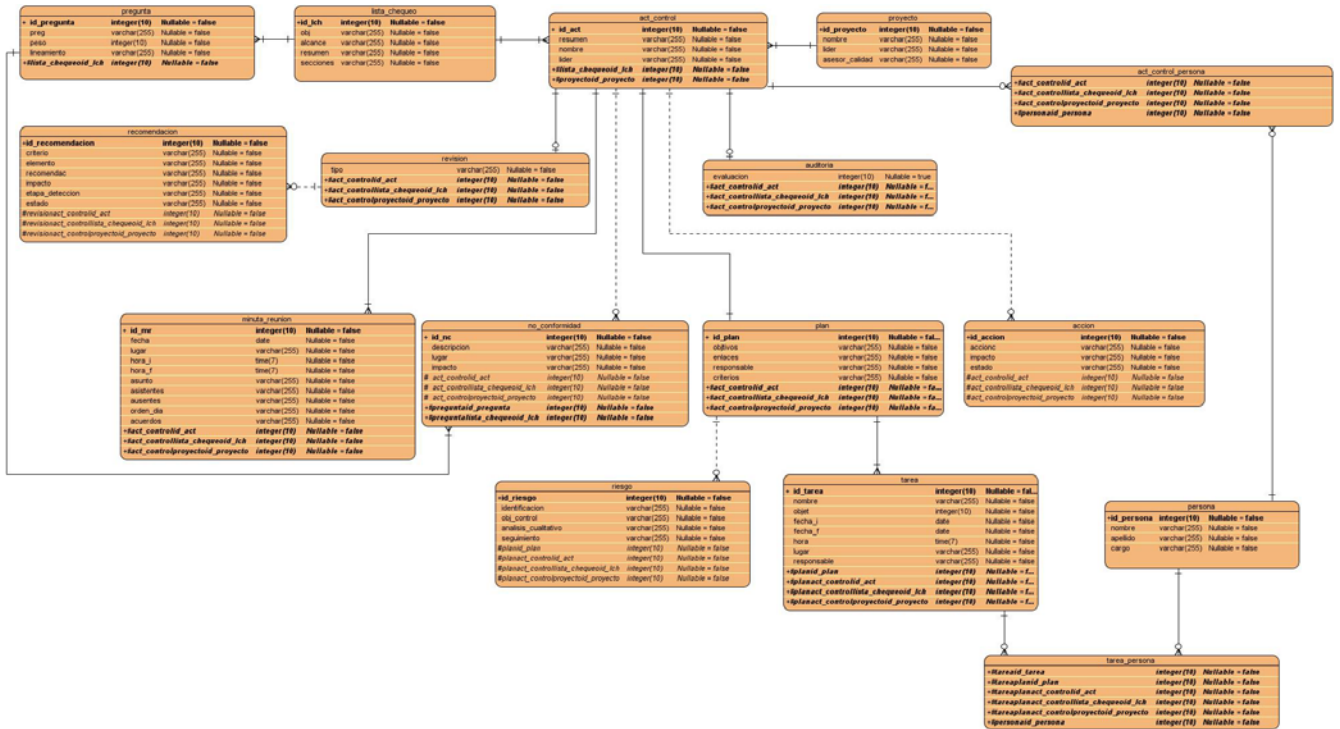


Fig. 62 – Modelo físico de datos

## 2.8 Diagrama de despliegue

Es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en la implementación de sistemas y las relaciones entre sus componentes.

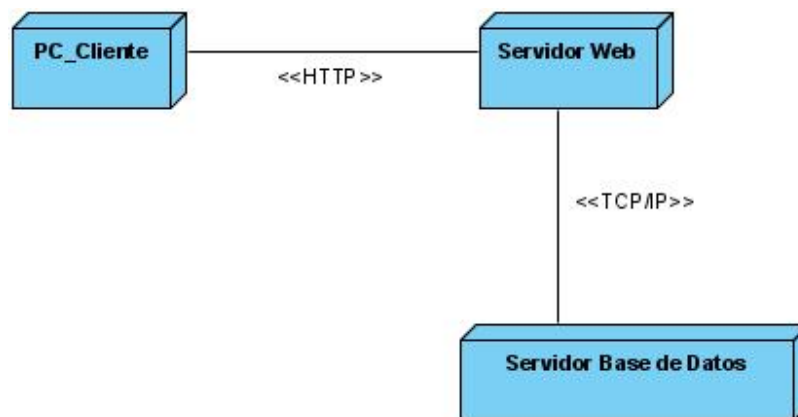


Fig. 63 - Diagrama de despliegue

Nombre del Nodo	Descripción
PC_Cliente	Computadora que no requiere altas prestaciones en cuanto a capacidad y velocidad de procesamiento debido a que no se realizan operaciones

	complejas en ella. El navegador realiza la solicitud de las páginas de la aplicación, interpreta el código HTML que recibe y procesa el código Java Script. Los requisitos mínimos necesarios de este nodo 100 MHz, 16 MB RAM.
Servidor Web	Éste requiere altas prestaciones pues en él se realizan todas las operaciones con los datos que se manipulan en el sistema, además éste se encarga de responder las llamadas realizadas al servicio Web, comunicarse con el servidor de Base de Datos y garantizar que el servicio Web brinde la respuesta en un tiempo aceptable, para lo cual requiere realizar gran cantidad de cálculos y una significativa cantidad de memoria. Los requisitos mínimos necesarios de este nodo 600 MHz, 256 MB RAM.
Servidor de Base de Datos	Necesita altas prestaciones de capacidad y rendimiento, debido al gran volumen de información que debe manipular y por el control de concurrencia e integridad referencial. Se encarga de transferir los datos a través de transacciones. El sistema gestor de base de datos utilizado requiere de los requisitos mínimos mostrados en el diagrama. Los requisitos mínimos necesarios de este nodo 1.7 GHz, 256 MB RAM.

### Descripción de la Vista de Instalación o Despliegue

#### 2.9 Conclusiones del capítulo

En este capítulo fueron expuestos diferentes elementos que ilustran cómo está construido el sistema, en términos de clases del análisis y del diseño. Este último dio la posibilidad de comprender la lógica del sistema en general. Por otro lado, fueron detalladas textualmente cada una de las clases Web, lo cual trae consigo que se conozca la responsabilidad de cada una de estas y qué funcionalidad específica realizan, además de que fue presentado el diagrama de clases persistentes y el modelo de datos respectivo.

## Conclusiones

Luego de haber realizado el estudio del estado del arte de los procesos objeto de automatización, una captura de requisitos que colectó las funcionalidades necesarias y siguiendo las mejores prácticas reunidas en la disciplina análisis y diseño con sus respectivos artefactos, se puede concluir lo siguiente:

- Existen muchas alternativas para controlar la calidad de un software durante su desarrollo, por lo que se debe establecer un mecanismo único y efectivo para controlar los proyectos en la facultad.
- Se analizaron un conjunto de herramientas existentes con el objetivo de aplicarlas durante algunas actividades, pero se pudo constatar que ninguna se podía adquirir y adaptar con facilidad. Además, se requerían determinadas facilidades a la medida que no estaban incluidas de manera conjunta en estas herramientas.
- Las orientaciones de la Dirección de Calidad de la UCI para la realización de las revisiones y auditorías son muy generales, la facultad debía establecer su procedimiento propio para lograr un control eficiente del personal y los proyectos.
- Existen muchos artefactos, listas de chequeo y elementos a tener en cuenta para realizar el control de calidad, por lo que deben ser recogidos y monitoreados utilizando una única lista de chequeo que contenga en diferentes secciones los aspectos a evaluar para cada artefacto o subproceso a controlar en la facultad.
- Para lograr un diseño óptimo y consistente es necesario el uso de patrones tales como el patrón Fachada y DAO, además de los conocidos y frecuentemente aplicados patrones GRASP para la asignación de responsabilidades.
- El uso de la metodología RUP propició la generación de gran cantidad de artefactos con sus consecuentes especificaciones que facilitan la comprensión a todos los involucrados así como su implementación futura.
- Con el módulo propuesto el grupo de control de calidad de la facultad podrá tener una mayor organización, logrando realizar controles más frecuentes y efectivos. Además, se podrá poner a disposición del personal autorizado la información que requiera en todo momento para apoyar la toma de decisiones, por ejemplo por parte del vicedecano de producción y el decano.
- No sólo se podrán controlar los proyectos, sino a los miembros del grupo de control permitiendo conocer las tareas asignadas, el estado de las mismas y la calidad con que han sido realizadas.

## Recomendaciones

Con el objetivo de aplicar y mejorar las propuestas que se realizan en este trabajo se recomienda lo siguiente:

- Realizar encuestas a expertos de otras facultades o entidades externas a la UCI para lograr determinar nuevos reportes que pueden ser de utilidad para un control efectivo de la calidad.
- Incorporar el uso del módulo Administración para establecer los niveles de acceso a los usuarios.

## Referencias Bibliográficas

- AB., M. *Web de MySQL*, 2007. [Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is-mysql-ab.html>]
- BUADES, G. *Calidad en Ingeniería del Software*, 2002. [Disponible en: <http://dmi.uib.es/~bbuades/calidad/index.htm>]
- Calidad*. 2009. [Disponible en: <http://www.degerencia.com/tema/calidad>]
- CRAIG LARMAN. *UML y Patrones*. 1999.[Disponible en: <http://biblioteca.uci.cu>].
- CORUÑA, U. D. *Una revisión sistemática de la adaptación del proceso software*, 2009.
- CHAVEZ, R. S. *Verificación y Validación del Software. Módulo 2: Revisiones de SW (VyV Estática)*, 2005.
- FERNÁNDEZ CARRASCO, O. M. G. L. *Un enfoque actual sobre la calidad del software*, 2007. [Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94351995000300005&lng=es&nrm=iso](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351995000300005&lng=es&nrm=iso). ISBN 1024-9435]
- HERNÁNDEZ, R. L. Y. C., S, G. . *El paradigma cuantitativo de la investigación científica*. Ciudad de La Habana, 2002. p.
- Ingeniería de Software I Conferencia 2 Fase de Inicio. Modelo del Negocio 2008[Disponible en: <http://eva.uci.cu>]
- Ingeniería de Software II Conferencia 2 Culminación de la Fase de Elaboración 2006[Disponible en: <http://eva.uci.cu>]
- J.K. *QUÉ SON LAS AUDITORÍAS DE SISTEMAS DE CALIDAD Y TIPOS DE AUDITORÍAS*, 2009.
- JACOBSON, Ivar, Booch, Grady and Rumbaugh, James. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley : s.n., 2000.
- JURAN, J. M. *Introducción a los conceptos de Calidad*, 2009]. Disponible en: <http://mgar.net/soc/isointro.htm>
- KMKEY, K. O. *Software de gestión de calidad (SGC) para ISO 9001 2000 o ISO 9000 ISO 14001*, 2008.
- LANZILLOTTA., A. *Definición de HTML*. 2004. p.
- LUQUE., L. C. *Curso de Javascript*. 2001. 127 p.



Norma internacional ISO 9000. 2000. p.

Mendoza, M, S. Metodologías de desarrollo de software. 2004. Perú. [Disponible en:  
<http://www.informatizate.net-MetodologíasDeDesarrolloDeSoftware.htm>]

Metodologías de desarrollo de software Mayo del 2008 [Disponible en:  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)]

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico.*, 2002. p. 8448132149

ROCHE, D., 2009]. [Disponible en: <http://www.slideshare.net/Tonymx/control-de-calidad-del-software>]

SOFTEXPERT. *ISOSYSTEM Quality Pack*, 2008. [Disponible en:  
<http://www.catalogodesoftware.com/producto.aspx?pid=1607>

TIENDALINUX.COM. *PostgreSQL*, 2007. [Disponible en:  
[http://soporte.tiendalinux.com/porta/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/porta/Portfolio/postgresql_ventajas_html)

Trygve Reenskaug Modelo Vista Controlador – Wikipedia 1979[Disponible en:  
[http://www.es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://www.es.wikipedia.org/wiki/Modelo_Vista_Controlador)]

ULTRASIST, I. E. S. A. *¿Porqué utilizamos ASP?*, 2008. [Disponible en:  
<http://www.ultrasist.com.mx/tecnologias/asp.htm>]

VÁZQUEZ, J. *Desarrollo Web con PHP y MYSQL*, 2007. [Disponible en:  
<http://www.gnuinos.com/drupal/?q=node/84>]

Visual Paradigm for UML. Visual Paradigm for UML. [En línea] Visual Paradigm. Febrero de 2007  
[Disponible en:  
<http://www.visual-paradigm.com/product/vpuml>]

## Glosario de Términos

- **BD:** Base de datos. Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.
- **CUN:** Caso de uso del negocio.
- **Herramientas CASE:** Computer Aided Software Engineering. Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **HTML:** *Hypertext Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.
- **MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- **PHP:** *PHP: Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- **PostgreSQL:** es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- **RUP:** *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- **SGBD:** Sistema de gestión de base de datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez
- **UML:** *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.