

# Universidad de las Ciencias Informáticas

Facultad 4



**Título:** Sistema para la Gestión de Mantenimiento para los Proyectos Productivos en la UCI.

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

**Autor(es):**

Ariagna Guerrero Hernández.

Suridelmis Arias Lescay.

**Tutor:**

Yulio Seriocha García Gallardo.

**Ciudad de La Habana, Junio 2009.**

## *Declaración de autoría*

---

Declaramos ser las únicas autoras del trabajo titulado:

---

---

y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los días \_\_\_\_ del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Suridelmis Arias Lescay

Firma del Autor

---

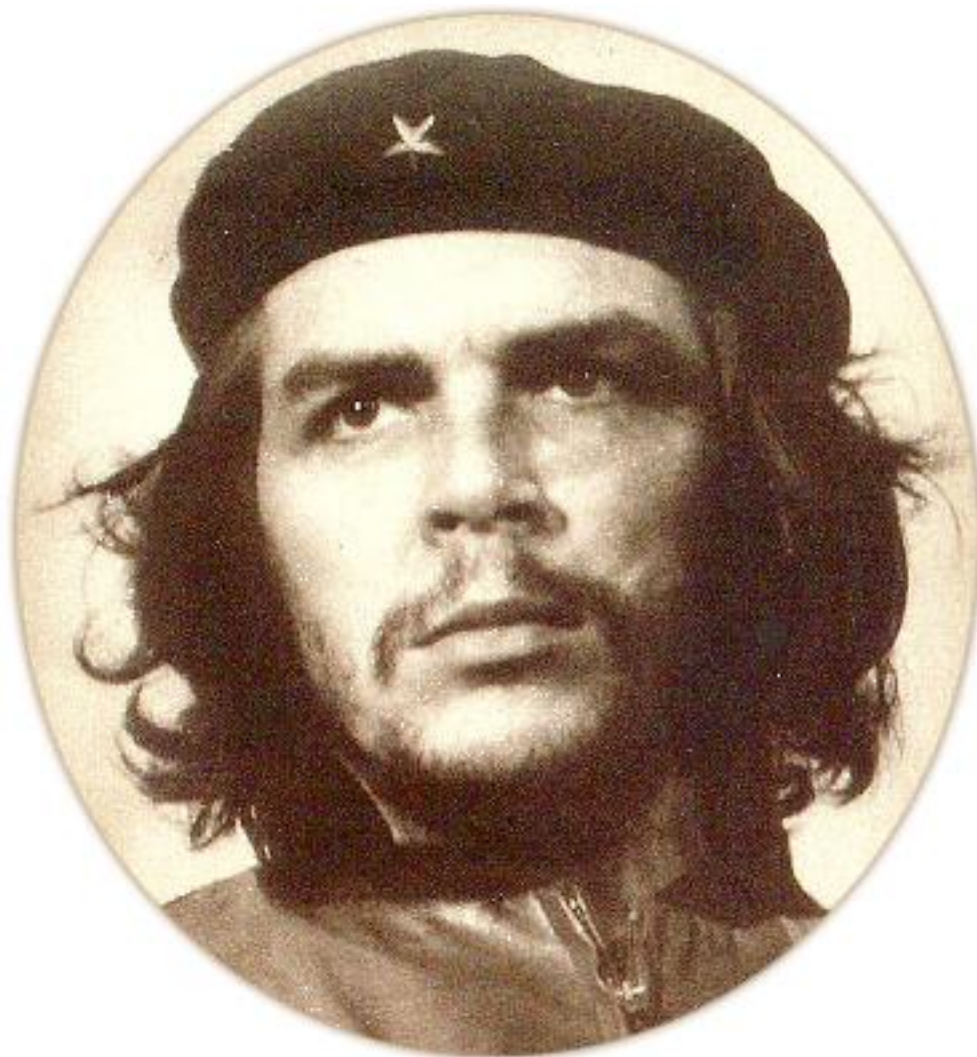
Ariagna Guerrero Hernández

Firma del Autor

---

Yulio Seriocha García Gallardo

Firma del Tutor



**"No se vive celebrando victorias, sino superando derrotas."**

**Ernesto Guevara**

Agradecerles a mis dos madres: Mi mamita y a La Virgen de la Caridad del Cobre, pues gracias a ellas he llegado hasta aquí. A mi papito por apoyarme en todo y soportar todos mis ataques. A Kaddiel que ha sido mi guía en todos mis años de estudio. A mi abuela Teresa y mi abuelo Bello, que en donde quiera que estén, se que están orgullosos de mi. A mi familia por el apoyo que me han dado, especialmente a mi tía Arelis y a mis primos Ali y Ale. A mi primo el Yeco, por aguantarme que se que no ha sido fácil. A mis mejores amigas y amigos de toda la vida: mi hermana Yicky y Chanel, Aliuska, Kenia, Carmen, Anita, Maylen y Alejandro, gracias por no olvidarme a pesar de la distancia. Al profesor Carlos Santana por creer en mí y a la Vicerrectora Rosa por darme una 2da oportunidad. A Estercita por cuidarme como si fuera mi propia madre. A todos mis grandes amigos que me apoyaron a lo largo de estos 6 años, especialmente a los que se quedaron en el camino, que siempre me dolerá el hecho de que hayan dejado de formar parte de mi vida antes de tiempo: Enma, Roly, Diana, Lisandra, Michel, mi título también es de ustedes. A mi primera familia de la UCI: Greylan (mi mamá), Ili (mi hija), Asley (mi abuela), mi niño Roly, mi nenenene Leo, Enma (mi hermana), Willy, Yuyo, Cesar y Herson, Michel y Kaddiel. A mi segunda familia: Diana, Mi cuerpo, La Mimi, La Muñi, Lisi y Lioni. A mi tercera familia: Anay (mi otra mamá), el Ne (Mi papá), a Renmy(Mi Ale), mis hermanas: Ari, Susi, Yuri, Paula, a Rosa (mi abuela), mi cuñado Manuel, a Marlito, a todos muchas gracias por acompañarme, escucharme, ayudarme en todo lo que me hizo falta y sobre todo por hacer que mis años universitarios sean hasta ahora los mejores de mi vida. Agradecer también a mis compañeros de aula, de todos los grupos por las que he pasado desde primer año hasta ahora. A todas, todas las personas con he logrado tener un gran lazo de amistad a lo largo de estos años. A todos mis profesores por sus grandes enseñanzas, especialmente: al profesor Santiago, Livan Kavir, Cueto, Pepe, Gorge Ernesto, Yaniselis, Yoinet, Yosvanis, Yiset, Erick, Martoris, Zenel y Yordanis. A nuestro tutor, gracias por pensar en nosotras para desarrollar este trabajo. A nuestra oponente y tribunal por brindarnos ayuda sin reparo. A Lianet por ayudarme cuando lo necesite. Un agradecimiento especial a mi compañera de tesis Arita, por enseñarme lo que es responsabilidad, por su cuenta de internet y por haber sido más que una hermana para mí. Y por último mi mayor agradecimiento a la Revolución y a Fidel, que gracias a que creó una universidad como esta, hoy yo soy ingeniera.

**Suridelmis Arias Lescay.**

## *Agradecimientos*

---

Agradecer a mi familia y en especial a mis padres por el apoyo, comprensión y amor brindados en los 5 años de universidad. A mis tíos Niurka y Leandro que con su atención y cuidados me hicieron sentir como en mi hogar. Agradecer a mi abuela Melba, donde te encuentres espero que te sientas orgullosa de mí.

Agradecer a Dignora y Capiro por permitirme entrar en su familia, a mi novio Leonel por su cariño, comprensión y apoyo incondicional. Gracias por estar siempre conmigo.

Gracias a Weeden, Ernesto Daniel, Julio César, Leydis, Lianet y Lázaro por brindarnos su ayuda y consejo para desarrollar este trabajo.

Agradecer a mis amigos que el destino convirtió en familia: Suri, Sucel, Yurisel, Paula, Anay, Weeden, Rosa y Marlon. Gracias por estar siempre y por su ayuda incondicional, sin ustedes no lo hubiésemos logrado. Agradecer a mi hermana Suri por siempre ser tan optimista ante todo, por enseñarme a tener paciencia y a trabajar en equipo.

Gracias a nuestro tutor por darnos la oportunidad de trabajar con él.

A las muchachas del apartamento y en especial a Yelenis, Maydalis y Mariesly por sus tiempos de máquina.

A todos mis amigos, nunca los voy a olvidar y pueden estar seguros que son el recuerdo más lindo que guardaré de la universidad. Agradecer a Odalys, quien me incitó a optar por esta escuela y me acompañó en estos 5 años apoyándome en los momentos de lágrimas y sonrisas. Gracias por ser más que una buena amiga.

A todos mis compañeros de grupo, los actuales y los de grupos anteriores por su preocupación.

A todos aquellos que alguna vez preguntaron por la tesis.

**Ariagna Guerrero Hernández.**

Dedico este trabajo que representa todos mis años de estudio a mis padres: Surella Lescay y Edilme Arias, que han sido mi impulso e inspiración en la vida.

**Suridelmis Arias Lescay.**

Dedico este sueño hecho realidad a mis padres María Luisa Hernández Salazar y Jorge Guerrero González por enseñarme a nunca rendirme.

A mi hermano Jorge Emilio por ayudarme a ser mejor persona cada día.

A la memoria de mi abuela Melba, gracias por educarme y cuidar de mí, este logro es tuyo.

**Ariagna Guerrero Hernández.**

Actualmente la Infraestructura Productiva de la Universidad de las Ciencias Informáticas no cuenta con un sistema que les permita llevar un control actualizado de los procesos de mantenimiento en los proyectos productivos y agilizar los procesos estadísticos vinculados al tema. Por lo antes expuesto este trabajo tiene como objetivo principal realizar el análisis, diseño e implementación de un sistema para la gestión de mantenimiento en la Universidad de las Ciencias Informáticas que facilite el control de los reportes de mantenimiento a los Especialistas del Grupo de Soporte al Desarrollo.

Para dar cumplimiento a este objetivo se realizó un estudio al compendio de tecnologías, metodologías y herramientas que se utilizan en el desarrollo web en la actualidad, además de analizar el negocio que tributa al levantamiento de requisitos funcionales y no funcionales que deberá tener el sistema, así como los artefactos propios del análisis y diseño del mismo. También se realiza la implementación según los artefactos resultantes de los flujos anteriores.

**Palabras claves:** Estadísticas, Mantenimiento, Reportes.

**Índice**

Introducción .....	1
Capítulo 1 Fundamentación teórica.....	5
Introducción.....	5
1.1 Conceptos asociados al tema.....	5
1.2 Soluciones de Software existentes que facilitan la gestión de información de mantenimiento.....	6
1.2.1    Ámbito internacional .....	6
1.2.2    Ámbito de la Universidad de las Ciencias Informáticas .....	11
1.2.3    Sistema de Gestión de Mantenimiento en la UCI. Valoraciones.....	11
1.3 Tendencias y tecnologías actuales a desarrollar .....	12
1.3.1 Estilos y patrones arquitectónicos .....	12
1.3.2    Lenguajes de programación.....	18
1.3.4    Sistemas Gestores de Bases de Datos.....	25
1.3.5    Servidor Web.....	28
1.3.6    ¿PDT + Subclipse + Mylyn + Aptana? .....	28
1.3.7    Herramientas CASE.....	31
1.3.8    Lenguaje de modelado .....	32
1.3.9    Metodologías de desarrollo de software.....	34
Conclusiones.....	38
Capítulo 2 Presentación de la solución propuesta.....	41
Introducción.....	41
2.1 Modelo de negocio .....	41
2.1.1 Estado actual del negocio .....	41
2.1.2 Actores y trabajadores del negocio .....	42
2.1.3 Diagrama de Casos de Uso del Negocio .....	43
2.2 Descripción de los procesos del negocio. ....	43
2.2.1 Descripción textual del Caso de Uso de Negocio.....	44
2.2.2 Reglas del negocio .....	45
2.2.3 Modelo de Objeto .....	46
2.3 Requisitos del sistema .....	46



2.3.1	Requerimientos funcionales.....	46
2.3.2	Requerimientos no funcionales.....	48
2.4	Descripción del Sistema Propuesto.....	51
2.4.1	Descripción de los actores.....	51
2.4.2	Diagrama de Casos de Uso del Sistema.....	52
2.5	Descripción de los Casos de Uso del Sistema.....	53
2.5.1	Caso de uso Autenticar Usuario.....	53
2.5.2	Caso de Uso Gestionar Usuario.....	54
2.5.3	Caso de Uso Gestionar Docente.....	56
2.5.4	Caso de Uso Eliminar Docente.....	58
2.5.5	Caso de uso Gestionar Facultad.....	60
2.5.6	Caso de uso Eliminar Facultad.....	63
2.5.7	Caso de uso Gestionar Laboratorio.....	64
2.5.8	Caso de uso Eliminar Laboratorio.....	67
2.5.9	Caso de uso Gestionar proyecto.....	69
2.5.10	Caso de uso Eliminar Proyecto.....	71
2.5.11	Caso de uso Gestionar Rotura.....	72
2.5.12	Caso de uso Eliminar Rotura.....	75
2.5.13	Caso de uso Gestionar Reporte.....	77
2.5.15	Caso de uso Buscar Datos Generales.....	81
	Conclusiones.....	82
	Capítulo 3 “Construcción de la solución propuesta”.....	83
	Introducción.....	83
3.1	Modelo de Clases del Análisis.....	83
3.1.1	Diagrama de Clases del Análisis.....	83
3.1.2	Diagrama de Interacción del Análisis.....	83
3.2	Modelo de Diseño.....	84
3.2.1	Patrones.....	84
3.3	Diagramas Clases del Diseño.....	90
3.4.1	Diagrama de Clases Persistentes.....	91
3.4.2	Diagrama Entidad Relación.....	92
3.5	Modelo de Despliegue.....	92

3.6 Modelo de Implementación.....	93
3.6.1 Diagrama de Componentes .....	93
Conclusiones.....	95
Conclusiones .....	96
Recomendaciones .....	97
Referencias bibliográficas .....	98
Bibliografía.....	100
Anexos.....	101
Anexo 1 Diagramas de Actividades .....	101
Anexo 2 Diagramas de clases del análisis.....	102
Caso de uso Autenticar Usuario.....	102
Caso de uso Gestionar Usuario .....	102
Caso de uso Gestionar Docente .....	103
Caso de uso Gestionar Facultad.....	103
Caso de Uso Gestionar Proyecto Productivo .....	104
Caso de Uso Gestionar Laboratorio .....	104
Caso de Uso Gestionar Rotura .....	106
Caso de Uso Gestionar Reporte .....	106
Anexo 3 Diagramas de Colaboración .....	108
Caso de uso Autenticar Usuario.....	108
Caso de uso Gestionar Usuario: Escenario Adicionar Usuario.....	109
Caso de uso Gestionar Usuario: Escenario Eliminar Usuario.....	110
Caso de uso Gestionar Docente: Escenario Adicionar Docente.....	111
Caso de uso Gestionar Docente: Escenario Modificar Docente .....	111
Caso de uso Gestionar Docente: Escenario Eliminar Docente.....	112
Caso de uso Gestionar Docente: Escenario Mostar Docente.....	112
Caso de uso Gestionar Facultad: Escenario Adicionar Facultad .....	113
Caso de uso Gestionar Facultad: Escenario Modificar Facultad .....	113
Caso de uso Gestionar Facultad: Escenario Eliminar Facultad .....	114
Caso de uso Gestionar Facultad: Escenario Mostrar Facultad.....	114
Caso de uso Gestionar Proyecto: Escenario Adicionar Proyecto .....	115
Caso de uso Gestionar Proyecto: Escenario Modificar Proyecto.....	115

Caso de uso Gestionar Proyecto: Escenario Eliminar Proyecto .....	116
Caso de uso Gestionar Proyecto: Escenario Mostrar Proyecto .....	116
Caso de uso Gestionar Laboratorio: Escenario Adicionar Laboratorio .....	117
Caso de uso Gestionar Laboratorio: Escenario Modificar Laboratorio .....	117
Caso de uso Gestionar Laboratorio: Escenario Eliminar Laboratorio .....	118
Caso de uso Gestionar Laboratorio: Escenario Mostrar Laboratorio .....	118
Caso de uso Gestionar Rotura: Escenario Adicionar Rotura .....	119
Caso de uso Gestionar Rotura: Escenario Modificar Rotura .....	119
Caso de uso Gestionar Rotura: Escenario Eliminar Rotura .....	120
Caso de uso Gestionar Rotura: Escenario Mostrar Rotura .....	120
Caso de uso Gestionar Reporte: Escenario Adicionar Reporte .....	121
Caso de uso Gestionar Reporte: Escenario Modificar Reporte .....	121
Caso de uso Gestionar Reporte: Escenario Eliminar Reporte .....	122
Caso de uso Gestionar Reporte: Escenario Mostrar Reporte .....	122
Caso de uso Gestionar Reporte: Escenario Exportar Reporte .....	123
Anexo 4 Diagramas de Clases del Diseño.....	124
Diagrama de Clases del Diseño del Caso de Uso: Autenticar Usuario.....	124
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Adicionar .....	125
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Modificar. ....	126
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Eliminar. ....	127
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Adicionar. ....	128
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Modificar. ....	129
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Mostrar. ....	130
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Eliminar. ....	131
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Adicionar. ....	132
Figura 3.45 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Adicionar.....	132

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Modificar. ....	133
Figura 3.46 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Modificar. ....	133
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Mostrar. ....	134
Figura 3.47 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Mostrar. ....	134
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Eliminar. ....	135
Figura 3.48 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Eliminar. ....	135
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Adicionar. ....	136
Figura 3.49 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Adicionar. ....	136
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Modificar. ....	137
Figura 3.49 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Modificar. ....	137
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Mostar. ....	138
Figura 3.50 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Mostrar. ....	138
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Eliminar. ....	139
Figura 3.51 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Eliminar. ....	139
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Adicionar. ....	140
Figura 3.52 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Adicionar. ....	140
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Modificar. ....	141
Figura 3.53 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Modificar. ....	141

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Mostrar.	142
Figura 3.54 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Mostrar.	142
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Eliminar.	143
Figura 3.55 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Eliminar.	143
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Adicionar.	144
Figura 3.56 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Adicionar.	144
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Modificar.	145
Figura 3.57 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Modificar.	145
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Mostrar....	146
Figura 3.58 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Mostrar.	146
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Eliminar. ...	147
Figura 3.59 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Eliminar.	147
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Adicionar.	148
Figura 3.60 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Adicionar.	148
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Modificar.	149
Figura 3.61 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Modificar.	149
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Mostar. ...	150
Figura 3.62 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Mostrar.	150
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Eliminar.	151
Figura 3.63 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Eliminar.	151
Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Exportar	152

Figura 3.64 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Exportar. ....	152
Glosario de términos.....	153

## **Introducción**

La Universidad de las Ciencias Informáticas (UCI) tuvo su génesis al calor de la Batalla de Ideas. Es el lugar que hoy acoge a los estudiantes y profesores que en su conjunto suman una cifra cercana a las 12000 personas. Su objetivo es convertirse en un centro de excelencia, pues así lo afirma quien sería su autor intelectual: Fidel Castro Ruz.

"...Universidad de excelencia es la idea de hacer lo mejor que pueda hacerse, de desarrollar la mejor universidad que se haya desarrollado jamás, y podemos, no es un chovinismo ridículo, buscamos lo óptimo, lo más perfecto posible dentro de las cosas humanas, lo más nuevo, lo más creativo, algo que no sólo sirva a los intereses de nuestro país, sino sirva de ejemplo para el resto del mundo."

Para lograr este objetivo la institución se sirve de la integración del estudio, la investigación y la producción para la formación integral de un ingeniero informático con los conocimientos y capacidades al servicio de nuestra sociedad. Su organización es muy peculiar, se divide en 10 facultades. Las mismas cuentan con los Polos Productivos donde se agrupan sus proyectos. Estos polos ya suman 30 y están bajo la tutela de la Infraestructura Productiva. La Dirección General de la Infraestructura Productiva es la máxima rectora en la toma de decisiones en cuanto a producción en la UCI. Supeditada a esta se encuentran otras 11 direcciones especializadas en diversas ramas. Entre ellas destaca la Dirección Técnica, formada por grupos especializados en Arquitectura y Tecnologías, Arquitectura de Información, Gestión de Comunidades Virtuales y Soporte al Desarrollo. Entre las importantes labores que desempeña este último grupo se encuentra la gestión de la infraestructura tecnológica de los proyectos productivos, donde los procesos de mantenimiento son un punto clave.

Debido a que el número de proyectos productivos se encuentra en crecimiento, la información con respecto a los procesos de mantenimiento representan un flujo de información cuantiosa. Los procesos de tratamiento de esta son realizados manualmente por lo que muchas veces se hace complejo este trabajo. Como mantenimiento se entiende: a los procesos referentes a roturas, estado de los medios así como su control y cambios necesarios.

Como resultado de esto existe la siguiente **situación problemática**:

Hay dificultad para satisfacer las necesidades de mantenimiento de los proyectos productivos en la UCI, ya que los reportes de mantenimiento se realizan manualmente provocando la ocurrencia de errores humanos y retrasos para brindar los servicios requeridos a cada una de las solicitudes realizadas por los mencionados proyectos. No existe un control actualizado de los procesos de mantenimiento y se dificulta el control de los medios y los cambios que se realizan. Todo esto provoca que los procesos estadísticos sean muy engorrosos.

Por lo expuesto anteriormente se ha definido como **problema** a resolver:

¿Cómo proporcionar eficiencia a la gestión de mantenimiento de los proyectos productivos en la UCI?

Se plantea como **objeto de estudio** el proceso de gestión de mantenimiento de los proyectos productivos.

Se considera como **campo de acción** la automatización de los procesos de gestión de mantenimiento de los proyectos productivos en la UCI.

El **objetivo general** que persigue esta investigación es desarrollar un sistema que controle la gestión de mantenimiento en los proyectos productivos en la UCI.

Para desarrollar la presente investigación se proponen las siguientes **tareas de investigación**:

- ✓ Realizar entrevistas a los Especialistas del Grupo de Soporte al Desarrollo para comprender el negocio.
- ✓ Realizar un estudio de las herramientas necesarias para desarrollar el sistema, desde la fase inicial.
- ✓ Obtener los requisitos funcionales y no funcionales del sistema a partir del análisis del negocio.
- ✓ Elaboración del modelo de datos.
- ✓ Realización de los diagramas correspondientes al análisis del sistema.



- ✓ Realización de los diagramas correspondientes al diseño del sistema.
- ✓ Realizar implementación del sistema.

## **Métodos Científicos utilizados en la investigación**

### **Métodos teóricos**

**Analítico–sintético:** consiste en la reconstrucción de un todo, a partir del estudio de los elementos previamente descompuestos y revisados ordenadamente cada uno por separado.

**Análisis histórico–lógico:** el método histórico estudia la trayectoria de los fenómenos y acontecimientos en el devenir de su historia. El método lógico investiga la existencia o no de leyes generales del funcionamiento y desarrollo de los fenómenos. El método lógico requiere apoyo del método histórico para descubrir la existencia de leyes fundamentales de los fenómenos basados en los datos que va proporcionando este último, para despojarse de toda posibilidad de generar razonamientos especulativos.

### **Métodos Empíricos**

**Observación:** es el primer procedimiento de carácter empírico, en el cual pueden distinguirse el objeto, el sujeto, los medios y las condiciones de la observación. Además del sistema de conocimientos a partir del cual se formula la finalidad de la observación y se interpretan los resultados de ésta.

**Entrevista:** se define como un método empírico complementario de investigación del cual se obtiene información amplia, abierta y directa de forma oral durante una conversación planificada entre el entrevistador y el o los entrevistados. En el desarrollo de esta investigación se entrevistan a especialistas del Grupo de Soporte de Desarrollo de la Infraestructura Productiva.

A continuación se brinda una argumentación de los capítulos por los que está constituida la presente investigación.

**Capítulo 1:** se realizará un análisis al compendio de tecnologías, metodologías y herramientas a utilizar en el desarrollo de la investigación así como también de conceptos relacionados. Además se efectuará un estudio del estado del arte con relación a la investigación en cuestión.

**Capítulo 2:** se realizará un estudio del Negocio especificando los actores, trabajadores, casos de uso, descripciones y diagramas correspondientes. Esto tributara a la especificación de requisitos funcionales y no funcionales propios del sistema, igualmente se efectuará una descripción del sistema propuesto.

**Capítulo 3:** se efectuarán los artefactos concernientes a los flujos de trabajo Análisis, Diseño e Implementación.

## **Capítulo 1 Fundamentación teórica**

### **Introducción**

En el presente capítulo se realiza un estudio del estado del arte correspondiente a la Gestión de Mantenimiento a nivel global. El objetivo de este estudio será revelar las ventajas que reportaría la Gestión de Mantenimiento en el proceso de desarrollo de software en los proyectos productivos de la UCI. Conjuntamente se efectuará un estudio sobre las metodologías y herramientas vigentes que puedan tributar a la propuesta más eficiente.

### **1.1 Conceptos asociados al tema**

#### **Sistema de gestión**

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. El mismo ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

#### **Proyecto productivo**

Se entiende por proyecto productivo todo proceso encaminado a conseguir un objetivo establecido de forma previa, con unos recursos económicos y temporales limitados en los que la finalidad es desarrollar una actividad de tipo económico fundamentalmente caracterizada por la creación de productos tales como servicios o bienes.

#### **Mantenimiento**

Se entiende por **mantenimiento** el conjunto de operaciones y cuidados necesarios para mejorar aspectos operativos relevantes de un establecimiento tales como: funcionalidad, seguridad, productividad, confort, imagen corporativa, salubridad e higiene. Otorga la posibilidad de racionalizar costos de operación.

Planteado esto, las autoras definen como gestión de mantenimiento los procesos encaminados al cuidado, seguridad y máximo rendimiento de los medios tecnológicos y tangibles que residen hoy en la Infraestructura Tecnológica de los proyectos productivos de cada una de las facultades de la Universidad de las Ciencias Informáticas.

## **Reporte**

Documento que refleja el comportamiento de un conjunto de indicadores. Se entiende por reporte de mantenimiento a la acción de notificar la solicitud de mantenimiento por determinada entidad, en el caso en cuestión por un proyecto productivo de la UCI.

## **Estadísticas**

Recolección, análisis e interpretación de datos numéricos resultantes de la gestión de mantenimiento.

## **1.2 Soluciones de Software existentes que facilitan la gestión de información de mantenimiento**

### **1.2.1 Ámbito internacional**

A nivel internacional existen herramientas muy utilizadas para la generación de reportes de gran aceptación. Esta aceptación se debe a las grandes ventajas resultantes del manejo de los mencionados softwares por parte de los usuarios. Entre las ventajas destacan:

Manipulación de los datos a través de consultas parametrizadas.

Generación de nuevas consultas sin modificar los datos ni la aplicación.

Diversidad de Presentación:

- ✓ Generación de gráficos.
- ✓ Generación de Tablas.

- ✓ Diversos formatos de salida: TXT, PDF, XML, HTML

A continuación una breve argumentación de las herramientas tomadas para la investigación:

## Herramienta de Reportes: iReport

Es una herramienta para construir y diseñar informes visual, poderoso, intuitivo y fácil de usar para JasperReport<sup>1</sup> escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos. Está integrado además con JFreeChart<sup>2</sup>, una de las bibliotecas gráficas Open Source más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC<sup>3</sup>, JavaBeans<sup>4</sup>, XML, etc. Es Open Source y gratuito. (Ver figura 1.1) [1]

Principales características a destacar:

- ✓ Configuración vía cliente pesado.
- ✓ Exportación a diversos formatos.
- ✓ Integración simple con aplicaciones Java.
- ✓ Integración con aplicaciones vía WebServices<sup>5</sup>.
- ✓ Incluye asistentes para crear informes automáticamente, así como también para la generación de subreportes y plantillas.

---

<sup>1</sup> Es una herramienta de creación de informes Java open source y multiplataforma.

<sup>2</sup> Es una librería gráfica de Java 100% libre que facilita a los implementadores mostrar gráficos de calidad profesional en sus aplicaciones.

<sup>3</sup> Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

<sup>4</sup> Es un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java. Se utilizan para encapsular varios objetos en un único objeto.

<sup>5</sup> Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, a través de la Web.

## Desventajas iReport

Cuando se realizan informes más complejos en los que intervienen gran cantidad de tablas y relaciones de la BD, el trabajo con esta herramienta se vuelve tedioso ya que se crean una multitud de subinformes o envía más parámetros de los deseados.

## Valoraciones iReport

Se clasifica a esta herramienta como no apta para utilizar en la generación de reportes sobre mantenimiento en la UCI apoyándose en la desventaja planteada anteriormente. Esto ralentizaría la emisión de reportes, podría generar errores, y disgustos a los usuarios.

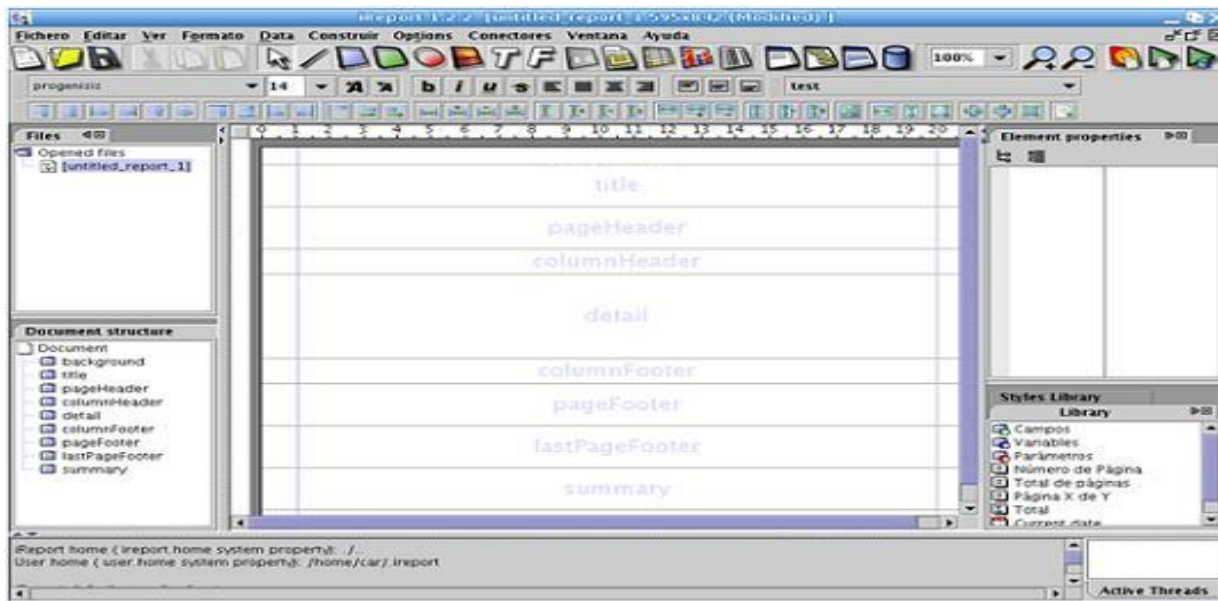


Figura 1.1 Herramienta de Reportes: iReport

## Herramienta de Reportes: AGATA

Agata Report es un generador de informes multiplataforma, una herramienta para la investigación y la generación de gráficos. Esta es una aplicación publicada bajo la licencia GNU

GPL y está escrita en PHP GTK<sup>6</sup>. Puede interactuar con diversos sistemas gestores de bases de datos como PostgreSQL, MySQL y Oracle. Proporciona un módulo para crear gráficas en curvas o en bastones y permite cruzar informes con otras bases de datos. Además, el programa soporta parámetros temporales de ejecución. [2]

Principales características:

- ✓ Capacidad de utilizarse vía web o vía cliente pesado.
- ✓ Diversos formatos de datos para consultar.
- ✓ Capacidad de generar ONLINE u OFFLINE.

## **Desventajas AGATA**

La principal desventaja encontrada a esta herramienta fue que entre los idiomas que solo soporta parcialmente se encuentra el español.

## **Valoraciones AGATA**

Se clasifica a esta herramienta como no apta para utilizar en la generación de reportes sobre mantenimiento en la UCI ya que al solo soportar parcialmente el idioma español podría desencadenar algunos conflictos usuario-herramienta. Esto retrasaría los procesos de mantenimiento en la institución repercutiendo en el correcto desempeño de la tecnología en los proyectos productivos.

---

<sup>6</sup> Es una extensión para el lenguaje de programación PHP que permite la utilización de Gimp Tool Kit , que es una librería para el diseño de interfaz gráfica.

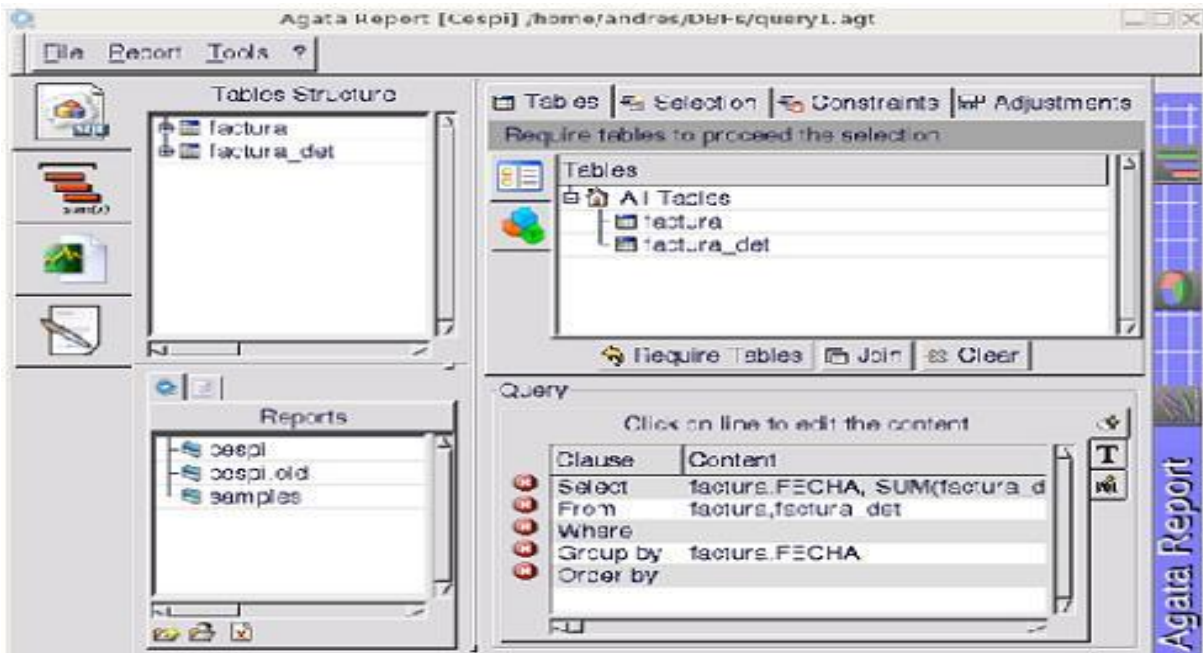


Figura 1.2.1 Herramienta de Reportes: AGATA

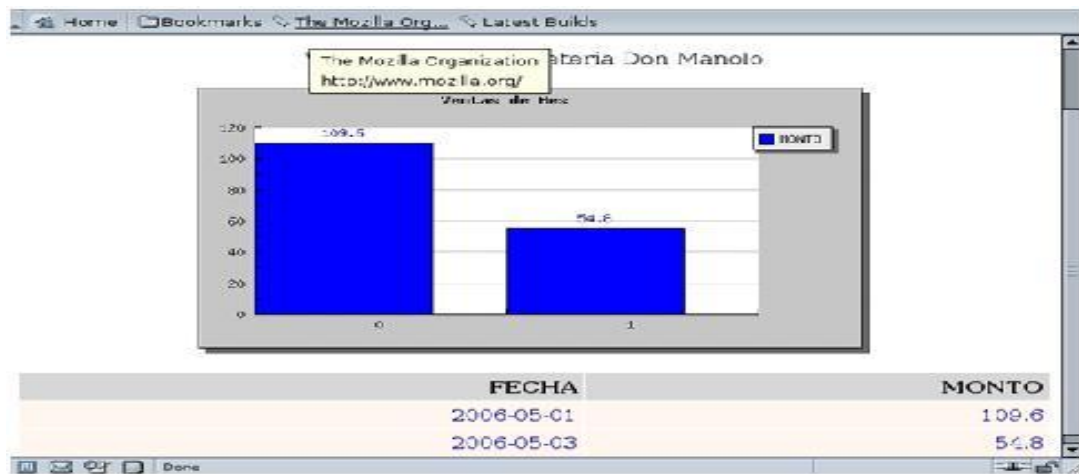


Figura 1.2.2 Herramienta de Reportes: AGATA



## 1.2.2 Ámbito de la Universidad de las Ciencias Informáticas

Dentro del ámbito de la UCI se encuentra la aplicación GatServer, acrónimo de Grupo de Asistencia Técnica. Esta aplicación se encarga de la gestión tecnológica de la universidad. A pesar de que esta aplicación facilita la gestión de los reportes de solicitud de mantenimiento, no permite que los especialistas del Grupo de Soporte al Desarrollo de la Dirección Técnica de la Infraestructura Tecnológica tengan un control sobre estos servicios. Para el desarrollo de esta aplicación se utilizó Active Server Pages como lenguaje y SQL Server como gestor de base de datos.



Figura 1.3 Herramienta de Reportes: GatServer.

## 1.2.3 Sistema de Gestión de Mantenimiento en la UCI. Valoraciones

La utilización de un sistema para la gestión de mantenimiento de los proyectos productivos en la UCI implantará una forma de crear valor para la institución. Con ella se persiguen objetivos

como lograr la optimización de la disponibilidad de los equipos productivos y la maximización de la vida útil de las máquinas. Esto resultaría una atenuación de las roturas de tecnología y de los recursos que se emplean en su compra para restituirlos. A fin de que se mitigaran los procesos tardíos para la emisión de reportes encaminados a la gestión de información referente al mantenimiento, se realizarán de forma inmediata las consultas, se atenuarán los errores humanos, se podrá disminuir los costos de mantenimiento así como también optimizar los recursos humanos que se dedican a esta labor.

## **1.3 Tendencias y tecnologías actuales a desarrollar**

### **1.3.1 Estilos y patrones arquitectónicos**

#### **1 Estilos arquitectónicos**

Según lo expuesto por Mark Klein y Rick Kazman en el año 1999, “un estilo arquitectónico es una descripción del patrón de los datos y la interacción de control entre los componentes, ligada a una descripción informal de los beneficios e inconvenientes aparejados por el uso del estilo”. Los estilos arquitectónicos, afirman, “son artefactos de ingeniería importantes porque definen clases de diseño junto con las propiedades conocidas asociadas a ellos”. [3]

#### **2 Patrones arquitectónicos**

Se entiende por patrón de arquitectura de software la descripción de un problema particular y recurrente del diseño, este surge en un contexto específico representando un esquema genérico y probado de su solución.

Existen numerosas clasificaciones y definiciones de diversos autores. A continuación se argumentan algunos de los patrones de arquitectura más utilizados en la actualidad.

#### **Arquitectura Cliente-Servidor**

Pertenece a la clasificación de estilos jerárquicos definida por Roy Fielding [Fie00] donde se consideran los estilos que incluyen alguna forma de distribución o topología de red. [4] Es un

modelo para desarrollar sistemas de información donde las transacciones se dividan en procesos independientes cooperando entre sí para intercambiar recursos, información y servicios. El cliente es el proceso que inicia el dialogo y el servidor el que responde a las solicitudes. (Ver Figura 1.4)

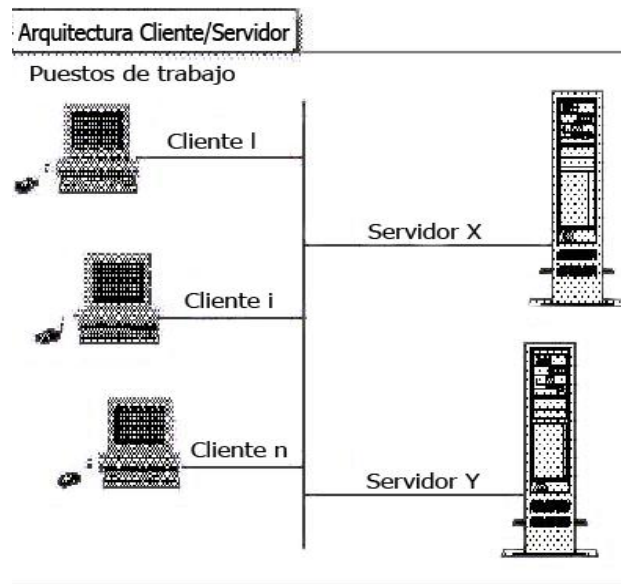


Figura 1.4 Arquitectura Cliente/Servidor

Existen numerosos aspectos que caracterizan a esta arquitectura:

- ✓ El Cliente y el Servidor pueden desempeñarse como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- ✓ Las funciones de Cliente y Servidor pueden existir en plataformas separadas, o en la misma plataforma.
- ✓ Un Servidor puede suministrar servicio a numerosos Clientes en forma concurrente.
- ✓ Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes.
- ✓ Los cambios realizados en las plataformas de los Clientes o de los Servidores se realizan de una manera transparente para el usuario final.

## Arquitectura en Capas

Pertenece a una clasificación elaborada por Buschmann, Meunier, Rohnert, Sommerlad y Stal en POSA [BMR+96]. Muchos sistemas de información en la actualidad se basan en este patrón de arquitectura en la modalidad de dos capas. Estas capas pertenecen al nivel de aplicación y nivel de base de datos. La utilización de este estilo acarrea inconvenientes como la recarga del nivel de aplicaciones lo que ha repercutido en la tendencia a utilizar arquitectura de tres capas. (Ver Figura 1.5)

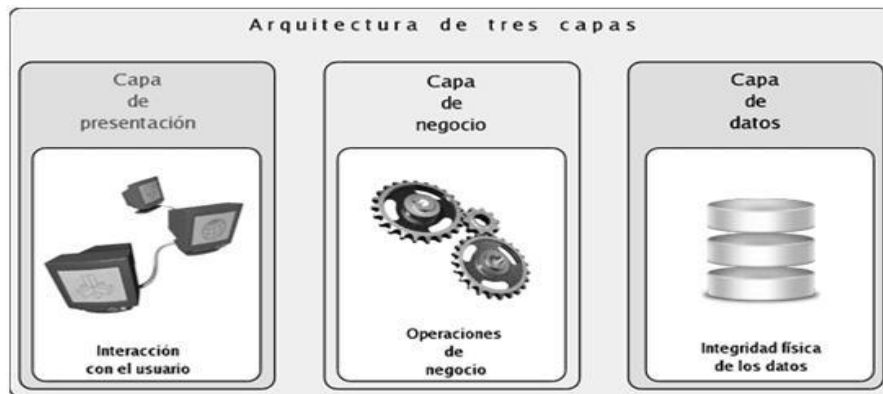


Figura 1.5 Arquitectura de tres capas.

Las tres capas propuestas por esta arquitectura son:

**Lógica de Presentación:** esta es la parte del código que interactúa con la PC o Terminal de autoservicio utilizada por el usuario. Se encarga de presentar el sistema, comunicar la información a los usuarios y capturarla de la interacción usuario-sistema. Esta capa se encarga de tareas como la disposición de los elementos gráficos en la pantalla, escribir los datos en pantalla, manejo de ventanas y solamente se comunica con la capa de negocio.

**Lógica de Negocio:** en esta capa es donde se establecen y codifican las reglas del negocio. Recibe las peticiones de los usuarios a través de la capa de presentación y mediante esta envía las respuestas solicitadas.

**Lógica del Procesamiento de los Datos:** esta capa se encarga del acceso y almacenamiento de los datos persistentes en la base datos.

## **Ventajas del estilo en capas**

- ✓ Soporte de un diseño establecido en niveles de abstracción crecientes permitiendo a los implementadores del equipo de desarrollo el fraccionamiento de un problema complejo en una secuencia de pasos incrementales.
- ✓ Permite desarrollar acciones de optimización y refinamiento.
- ✓ Suministra amplia reutilización ya que permite utilizar implementaciones o versiones de una capa mientras soporten las mismas interfaces de cara a las capas adyacentes posibilitando definir interfaces de capa estándar.

## **Desventajas del estilo en capas**

Muchos autores han definido un grupo de desventajas para este estilo:

- ✓ En muchos casos existen problemas que no admiten un buen mapeo en una estructura jerárquica.
- ✓ En numerosas ocasiones existe extrema dificultad para encontrar el nivel de abstracción correcto.
- ✓ Los cambios a realizar en las capas de bajo nivel tienden a filtrarse hacia las de alto nivel, especialmente si se utiliza una modalidad relajada.
- ✓ La arquitectura en capas ayuda a controlar y encapsular aplicaciones complejas, pero complica no siempre razonablemente las aplicaciones simples [MS04c].
- ✓ Pérdida de eficiencia.
- ✓ Trabajo innecesario por parte de capas más internas o redundante entre varias capas.

## Modelo-Vista-Controlador (MVC).

Según Taylor y Medvidovic [TMA+95], es un estilo que pertenece a la familia de estilos de Llamada y Retorno que enfatiza la modificabilidad y la escalabilidad, pero otros autores como por ejemplo Buschmann, lo clasifican como patrón correspondiente al estilo de los sistemas interactivos [BMR+96].

El patrón MVC aísla el modelado del dominio, la presentación y las acciones establecidas por los usuarios en el ingreso de información en tres clases diferentes [Bur92]:

- ✓ Modelo: el modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- ✓ Vista: maneja la visualización de la información.
- ✓ Controlador: interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. [5] (Ver figura 1.6).

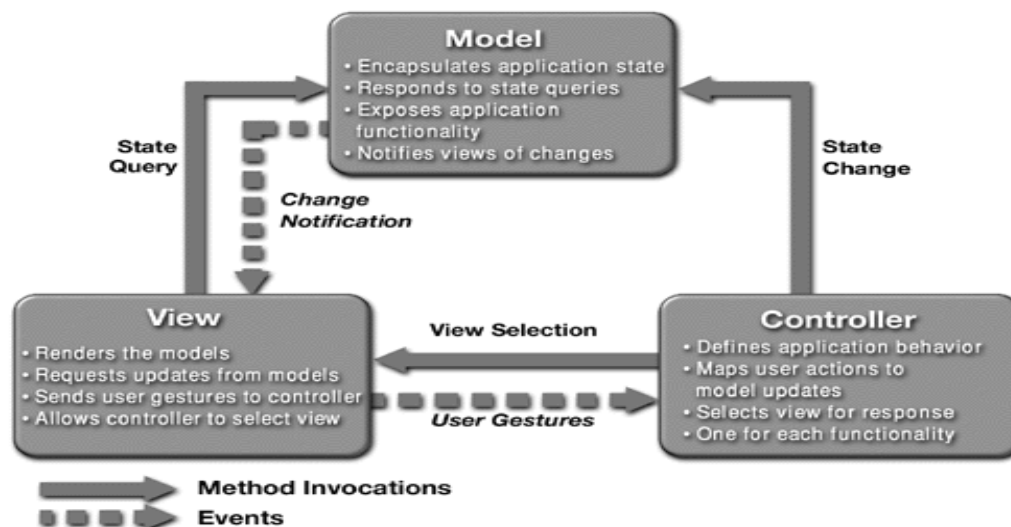


Figura 1.6 Las capas modelo, vista y controlador de la arquitectura MVC.

## Ventajas de MVC

Esta arquitectura provee grandes ventajas, como la organización del código, la reutilización y la flexibilidad.

- ✓ Separa los datos de la representación visual de los mismos.
- ✓ Crea independencia de funcionamiento.
- ✓ Ofrece maneras sencillas de probar el correcto funcionamiento del sistema.
- ✓ Permite el escalamiento de la aplicación en caso requerido.
- ✓ Soporte de vistas múltiples, debido a que no existe una dependencia directa entre la vista y el modelo, la interfaz de usuario posibilita mostrar varias vistas de la misma información de forma simultánea.
- ✓ Adaptación al cambio, ya que al existir nuevas necesidades por parte de los usuarios la interfaz de usuario se ve sometida a cambios con frecuencia. Como el modelo no depende de las vistas, facilita la adición de nuevas opciones de presentación generalmente sin afectar al modelo. Esto sentó las bases para especializaciones posteriores de este patrón, tales como Page Controller y Front Controller.

## Desventajas de MVC

- ✓ Su estructura agrega complejidad al sistema.
- ✓ Costos de actualizaciones frecuentes.

### 3 Patrón de arquitectura a utilizar: MVC.

Tras el análisis de algunos patrones de arquitectura se escoge el Modelo-Vista-Controlador como el patrón a utilizar. Esta elección está basada en las ventajas resultantes de su uso que en todo momento inclinan la balanza a favor de este a pesar de que las desventajas no son de ignorar.

- ✓ Este patrón es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios.
- ✓ Esta implementado por la mayoría de los frameworks web de la actualidad.

- ✓ A pesar de que el número de archivos a mantener y desarrollar crece de forma considerable, la separación entre modelo y vista consiguen un mantenimiento más sencillo de las aplicaciones.

## 1.3.2 Lenguajes de programación

Se define como lenguaje de programación a una construcción mental del hombre para desarrollar programas. En la actualidad existen un numeroso conjunto de lenguajes que se utilizan en el desarrollo de aplicaciones web. Estos se clasifican en lenguajes del lado del Servidor y del lado del Cliente según su ubicación guarde correspondencia con la arquitectura Cliente/Servidor.

**Lenguajes de lado cliente:** son aquellos que pueden ser directamente procesados por el navegador y no necesitan tratamiento previo.

**Lenguajes de lado servidor:** son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible.

### 1 Lenguajes del lado del cliente

Entre los lenguajes del lado del cliente encontramos a JAVA, JavaScript, Applets de Java, Flash, HTML y CSS, los dos últimos organizan a los demás dentro de la página web. A continuación la fundamentación de algunos de ellos.

#### JavaScript

Es un lenguaje de programación que permite crear un grupo de acciones en el ámbito de la página web. Sus primeras versiones vieron la luz bajo el nombre de Livescript en el año 1990 y fue desarrollada por Netscape. Pero no es hasta diciembre de 1995 que aparece con el nombre de JavaScript con autoría del binomio Netscape y Sun Microsystems. Permite realizar validaciones en el lado del cliente así como también la programación de eventos.



## HyperText Markup Language (HTML)

Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Fue desarrollado en 1990 por Tim Berners-Lee con el objetivo de facilitar el acceso a investigaciones científicas.

## Cascading Style Sheets (CSS)

El origen de Cascading Style Sheets o Hojas de Estilo en Cascada data alrededor del año 1970 con el objetivo de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos. Permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web de forma simultánea. Funciona a base de reglas, estas reglas se dividen en dos partes: un selector y la declaración que a su vez, está compuesta por una propiedad y el valor que se le asigne.

### Las principales ventajas de las hojas de estilo son

**Control del diseño:** al definirse el formato de un documento electrónico mediante CSS se puede cambiar dicho formato modificando solamente la hoja de estilo. Esto permite reducción de tiempo y uniformidad en los procesos de diseño.

**Redefinición de etiquetas:** se declara mediante una sola línea de código el formato deseado especificando color, tamaño a la etiqueta/as deseada/as. En caso de no utilizar CSS este proceso sería muy engorroso.

**Personalización:** permite definir aspectos concretos de un documento facilitando su diseño.

**Maquetación y accesibilidad:** mayor precisión al colocar los elementos en el diseño al exceptuar el uso de tablas para la maquetación. Esto permite un documento con un formato más exquisito a pesar de que el navegador no soporte CSS.

**Tamaño:** se reduce el tamaño de los ficheros ya que todos los requerimientos que guardan relación con el formato de todas las páginas estarán en un solo archivo que pertenece a CSS.

**Usabilidad:** CSS permite adecuar el formato del documento a las necesidades del cliente confiriéndole mayor usabilidad y accesibilidad. Como por ejemplo las personas con deficiencia visual pueden aplicar una hoja de estilo para aumentar el tamaño de las letras o remarcar los enlaces.

## 2 Lenguajes del lado del servidor

### Active Server Pages (ASP)

ASP es una tecnología creada por Microsoft para el desarrollo web. La misma estaría incluida dentro de los lenguajes **ISS (Include Server Side)** de 2ª generación, por tanto se cataloga como una evolución de los **CGI's (Common Gateway Interface)**. [6]

Se pueden definir como archivos que se ejecutan en el servidor cuyo contenido puede ser tanto HTML como código de script. Soporta VBScript, Java y JavaScript.

ASP no define un lenguaje de programación sino que define una serie de objetos de servidor, constituidos por una serie de métodos que se pueden utilizar en actividades como acceso a base de datos, lectura de ficheros entre otras.

### Practical Extraction and Report Language (PERL)

Lenguaje de script de tipo BCPL como PHP, desarrollado por Larry Wall inspirado en otras herramientas de UNIX para la administración de tareas propias de estos sistemas. No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas.

Se utiliza en el desarrollo de aplicaciones y herramientas de forma rápida especialmente orientadas al tratamiento de textos y archivos, aunque también se utiliza para entornos gráficos, en combinación con sistemas como Perl/TK o GTK. [7]

## Hypertext Preprocessor (PHP)

Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Fue desarrollado por Rasmus Lerdorf en 1994 pero la versión a utilizar por otros usuarios sale a principios de 1995 bajo el nombre de *Personal Home Page Tools*. Es ampliamente utilizado para el desarrollo web y puede ser alojado dentro de código HTML. El código de PHP se escribe dentro de estos tags "<?,?>", y la sintaxis que se utiliza para un mensaje es la siguiente:

```
<? php echo "Tesis";?>
```

## Principales características

- ✓ Es multiplataforma.
- ✓ Publicación bajo licencia de software libre.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables.
- ✓ A partir del lanzamiento de PHP5 contiene manejo de excepciones.

## 3 Lenguaje a utilizar: PHP

Se elige a PHP como lenguaje de programación porque a diferencia de ASP es multiplataforma. En relación con PERL se puede decir que:

- ✓ PHP es mucho más sencillo en la definición de interfaces.
- ✓ Es más fácil incluir código PHP en HTML.
- ✓ Desde el punto de vista de script para webs, PHP tiene prácticamente las mismas posibilidades y potencia que el Perl.

✓

## 1.3.3 Frameworks en PHP

### 1 ¿Qué es un framework?

Un framework es una estructura de software que está formado por diversos componentes personalizables que se pueden configurar y adicionar elementos para el desarrollo de una aplicación. Su utilización fomenta la reutilización de código, promueve buenas prácticas de desarrollo y proporciona una reducción de tiempo en los procesos de desarrollo.

Entre los diversos frameworks que existen para PHP se pueden citar a:

#### **PHP Prado**

Es un framework de programación basado en componentes y conducido por eventos, para desarrollar aplicaciones Web en PHP 5. PRADO significa Desarrollo Rápido De aplicaciones orientado a objetos para PHP.

#### **¿Qué requiere PRADO?**

El único requisito para ejecutar PRADO basado en aplicaciones es un servidor web sostenido con PHP 5.1.0 o versiones superiores. PRADO es libre y se puede utilizar para desarrollar código abierto o aplicaciones comerciales.

#### **Características principales de PRADO**

- ✓ Orientado a objetos y alta reusabilidad del código.
- ✓ Programación conducida por eventos.
- ✓ Separación de la presentación y la lógica.
- ✓ Arquitectura configurable y modular actualizable.
- ✓ Completa diversidad de soporte a bases de datos.
- ✓ Componentes habilitados de AJAX.
- ✓ Configurable y localizable manejo de excepciones/errores.
- ✓ Framework de autenticación extensa y autorización.

- ✓ Medidas de seguridad: prevención XSS (a través de los scripts de los sitios), protección de cookies.
- ✓ Ejecución XHTML. [8]

## **Zend Framework**

Es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Implementa código totalmente orientado a objetos. Presenta una arquitectura débilmente acoplada permitiendo a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). Ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de utilizar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el framework. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework. [9]

## **Symfony**

Symfony es un framework que optimiza el desarrollo de aplicaciones web. Su objetivo es acelerar la creación y mantenimiento de aplicaciones web sustituyendo las repetitivas tareas de codificación para lo cual proporciona varias herramientas y clases.

Es fácil de instalar en cualquier configuración, sólo es necesario Unix o Windows con un servidor web y PHP instalado. Es compatible con la mayoría de los sistemas de bases de datos. Está desarrollado en PHP 5 y publicado bajo una licencia de software libre. [10]

## Principales características

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Es independiente del sistema gestor de bases de datos.
- ✓ Fomenta las mejores prácticas y patrones de diseño para la web.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor<sup>7</sup> y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, permitiendo su integración con librerías desarrolladas por terceros.

## Entorno de desarrollo y herramientas

Este framework contiene diferentes entornos de desarrollo y varias herramientas que posibilitan la automatización de las tareas que son comunes a la ingeniería de software para desarrollar aplicaciones:

- ✓ Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- ✓ El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas ("test-driven development").
- ✓ La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- ✓ La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- ✓ Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).
- ✓ El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

---

<sup>7</sup> Es un generador de documentación de código abierto escrito en PHP.

## 2 ¿Por qué utilizar Symfony?

- ✓ Es adecuado para proyectos de gran envergadura y que contengan algún grado de complejidad.
- ✓ Posee calidad en el código fuente y cuenta con gran documentación disponible.
- ✓ Sus herramientas y clases proporcionan una reducción del tiempo en el desarrollo de una aplicación web compleja.
- ✓ Implementa el MVC.
- ✓ Implementa las tareas que son comunes en el desarrollo web fomentando la reutilización.

### 1.3.4 Sistemas Gestores de Bases de Datos.

“El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.” (MÁRQUES, 2001). [11]

Según esta afirmación podemos decir que un SGBD es un software que gestiona los datos de una base de datos, se encarga de su integridad, confidencialidad y seguridad.

En la actualidad existe un potente conjunto de SGBD, entre ellos destacan:

#### **MySQL**

Es un sistema de gestión de bases de datos relacional con licencia GPL. Fue desarrollado por la empresa sueca MySQL AB, que mantiene el derecho de autor del código fuente del servidor SQL y de la marca. Contiene un gran número de herramientas así como de librerías. Es de fácil instalación y configuración.

#### **Principales características**

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

## **Ventajas de MySQL**

- ✓ Es uno de los SGBD más famosos debido a su velocidad a la hora de realizar las operaciones.
- ✓ Bajo consumo de recursos.
- ✓ Facilidad de configuración e instalación.
- ✓ Posee una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

## **PostgreSQL**

Es un poderoso sistema de base de datos objeto-relacional de código abierto. Posee una arquitectura probada y una sólida reputación de confiabilidad, integridad de los datos y corrección. Es multiplataforma, posee soporte para llaves foráneas, vista, procedimientos almacenados e incluye la mayoría de los tipos de datos SQL92 y SQL99, incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeo. Ejecuta procedimientos almacenados en más de una docena de lenguajes de programación. Además permite a los desarrolladores definir y crear sus propios tipos de datos junto con las funciones y operadores que definen su comportamiento. Es altamente escalable, tanto en la enorme cantidad de datos que puede administrar y en el número de usuarios concurrentes que puede acomodar. [12]



## Ventajas de PostgreSQL

- ✓ Posee una gran escalabilidad. Se ajusta perfectamente al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, posibilitando el soporte de una mayor cantidad de peticiones simultáneas de manera correcta.
- ✓ Implementa soporte para subconsultas y transacciones.
- ✓ Comprueba la integridad referencial.
- ✓ Capacidad de almacenar procedimientos en la propia base de datos, característica de los gestores de bases de datos de alto nivel como Oracle.

### 1 Sistema Gestor de Base de Datos a utilizar: PostgreSQL

Se escoge PostgreSQL como SGBD debido a que es un gestor GPL que incluye numerosas de las ventajas de los gestores propietarios, es de gran escalabilidad y contiene un gran número de funcionalidades de agrado para los desarrolladores. Es cierto que la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta favorable.

### 2 ¿Por qué utilizar pgAdmin III?

Esta es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que la convierte en multiplataforma, por lo que se puede utilizar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

pgAdmin III está diseñado para escribir desde sencillas consultas SQL hasta el desarrollo de bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y

facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas \*nix), y puede encriptarse mediante SSL para mayor seguridad. [13]

### 1.3.5 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol) y mediante este trasfiere una página web al navegador de un usuario.

#### ¿Por qué Apache?

Es un servidor web que debe su génesis a un proyecto iniciado en febrero de 1995. Inicialmente solo era un servidor para sistemas operativos UNIX aunque más tarde se realiza la versión para Windows. Su licencia es BSD, lo que permite realizar cambios en el código fuente siempre que se les reconozca el trabajo a sus creadores de inicio.

#### Principales características

- ✓ Es multiplataforma.
- ✓ Es una tecnología gratuita de código fuente abierto.
- ✓ Es un servidor altamente configurable de diseño modular. Actualmente existen muchos módulos para Apache que son adaptables a este.
- ✓ Apache trabaja con gran cantidad de lenguajes de script.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- ✓ Tiene un alto grado de configuración en la creación y gestión de logs.

### 1.3.6 ¿PDT + Subclipse + Mylyn + Aptana?

PDT Project es una distribución de Eclipse que incorpora todas las utilidades creadas para simplificar el trabajo a los desarrolladores de PHP. [14]

Entre sus principales características destacan:

- ✓ Es multiplataforma: disponible en Windows, Linux o Mac.
- ✓ Licencia de tipo Open Source.
- ✓ Está integrado por un editor de código PHP con ayuda, un editor de proyectos o archivos, que deben estar guardados en algún servidor web con soporte para PHP, y un sistema para hacer debug en PHP y depurar las aplicaciones web en PHP de forma sencilla.

Pero esta herramienta carece de apoyo SVN y su editor JavaScript por defecto solo soporta resaltado de sintaxis. Para lograr un mejor desempeño con Eclipse PDT se le adiciona apoyo SVN.

**Subclipse** es un plugin<sup>8</sup> para el entorno de desarrollo Eclipse que se encarga de interactuar con sistemas de control de versiones Subversión desde el entorno de desarrollo Eclipse, permitiendo que un determinado proyecto se encuentre en un repositorio común al equipo de desarrollo. [15]

### **Funcionalidades destacadas que ofrece Subclipse**

- ✓ Desarrollar un módulo en el repositorio a partir de un proyecto.
- ✓ Desarrollar un proyecto a partir de un módulo del repositorio.
- ✓ Mantenimiento de ficheros.
- ✓ Sincronización de ficheros con el repositorio.
- ✓ Actualizar ficheros.
- ✓ Guardar nuevos cambios en un fichero.
- ✓ Asignar etiquetas.
- ✓ Crear copias.

---

<sup>8</sup> Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño.

## Adicionando un editor de JavaScript

Aptana es una robusta y avanzada interfaz de desarrollo Web, enfocado a JavaScript para el desarrollo de aplicaciones dinámicas. Está basada en Eclipse, es multiplataforma y Open Source.

### Principales características

- ✓ Reconocimiento de errores de JavaScript y CSS.
- ✓ Ayuda para definir qué funciones y propiedades están soportadas en los distintos navegadores (CSS, HTML, JavaScript).
- ✓ Asistente en código JavaScript, HTML y CSS; incluyendo sus propias funciones en JavaScript.
- ✓ Herramientas para facilitar una mejor documentación: con ScriptDoc<sup>9</sup>: la documentación se visualiza en el asistente de código.
- ✓ Total integración con librerías de JavaScript.
- ✓ Posee un depurador de JavaScript similar al depurador clásico de eclipse.
- ✓ Mediante conexión FTP a un servidor externo permite modificar directamente los archivos guardados en él.
- ✓ Creación de Plantillas de Código.

## Mylyn

Mylyn es una interfaz para Eclipse centrada en tareas, que reduce la sobrecarga de información y que hace fácil la ejecución de tareas múltiples. Lo efectúa convirtiendo las tareas en una parte de primera clase de Eclipse e integrando ricos y autónomos repositorios tales como Bugzilla, Trac y Jira. Una vez que tus tareas están integradas, Mylyn monitorea la actividad de trabajo para identificar la información relevante para las tareas manuales y usa este contexto de tarea para centrar la interfaz de usuario en la información interesante,

---

<sup>9</sup> Establece un modelo uniforme para documentar el código creado con lenguajes dinámicos incluyendo JavaScript.

esconde la insignificante, y automáticamente encuentra que es lo que está relacionado. Este coloca la información que se necesita para completar trabajo y mejora la productividad reduciendo la búsqueda, el desplazamiento en las páginas o scrolling y la navegación. Haciendo explícito el contexto de tareas Mylyn también facilita la multitarea, planificación, la reutilización de esfuerzos pasados y el intercambio de experiencia o maestría. [16]

### 1.3.7 Herramientas CASE

Se puntualiza como herramientas CASE al conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. [17]

A continuación se argumentan algunas de ellas:

#### 1 Rational Rose Enterprise

Rational Rose es una herramienta Case de diseño, unificada, orientada al objeto del software que utiliza para la modelación el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language).

#### Características de Rational Rose Enterprise

- ✓ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ✓ Capacidad de análisis de calidad de código.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✓ Integración con otras herramientas de desarrollo de Rational.
- ✓ Publicación Web y generación de informes para optimizar la comunicación dentro del equipo. [18]

## 2 Visual Paradigm

Es una herramienta CASE multiplataforma que se integra fácilmente con varios Entornos Integrados de Desarrollo (IDEs) y brinda numerosas funcionalidades entre las que destacan:

- ✓ Entorno de creación de diagramas para UML.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Disponibilidad en múltiples plataformas.

## 3 Herramientas CASE a utilizar: Visual Paradigm

Se selecciona a Visual Paradigm como herramienta CASE para la modelación visual del sistema software debido a:

- ✓ Cuba posee las licencias que permiten su uso.
- ✓ Se puede utilizar en el sistema operativo Linux y Cuba se encuentra en la realización de estudios para migrar a este sistema operativo tomando a la UCI como punto de referencia.
- ✓ Las funcionalidades que brinda Visual Paradigm hacen de esta herramienta la candidata ideal para el modelado visual del sistema ya que cumple con las expectativas de las implicadas en el estudio.

### 1.3.8 Lenguaje de modelado

Como lenguaje de modelado se ha elegido por el colectivo de autoras al Lenguaje Unificado de Modelado (UML por su nombre en inglés Unified Modeling Language). Este es un lenguaje

desarrollado por Rumbaugh, Booch y Jacobson a partir de la necesidad de un lenguaje que apoyara los procesos de análisis de un problema y sirviera de base en la comunicación de ideas entre desarrolladores. La figura 1.7 muestra la evolución de UML. [19]

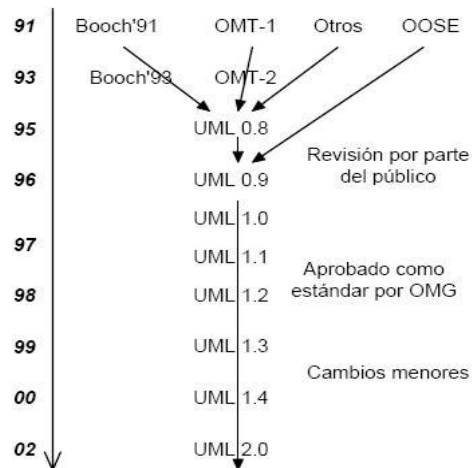


Figura 1.7 Evolución de UML.

## UML permite

**Visualizar:** con UML se puede expresar de una forma gráfica un sistema.

**Especificar:** con UML se puede especificar las características de un sistema antes de su construcción.

**Construir:** con los modelos especificados con UML se pueden construir los sistemas diseñados.

**Documentar:** los elementos gráficos sirven como documentación del sistema desarrollado y sirven de base en futuras revisiones.

## 1.3.9 Metodologías de desarrollo de software

### Extreme Programming (XP)

La Programación Extrema o eXtreme Programming (XP) como también se le conoce mundialmente, es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [20]

### Agile Software Development (ASD)

La Agile Software Development o Desarrollo Ágil de Software (ASD) es una metodología de trabajo que consiste en el desarrollo de soluciones informáticas para clientes, quienes solicitan el software a una empresa desarrolladora, la que en un proceso cíclico, que incluye revisiones periódicas del producto, logra cumplir las expectativas del cliente en períodos más cortos. El objetivo de ella es que la aplicación se implemente exactamente con los requerimientos que el cliente necesita realmente, aún cuando esos requisitos vayan evolucionando durante el proceso de desarrollo, lo que en este caso es algo clave.

Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios.

Según Austin Broyles, Ingeniero de Sistemas de Google, el desarrollo exitoso de sistemas informáticos con ASD conlleva 5 etapas clave, que son:



1. Escribir el código de prueba.
2. Realizar diseños simples.
3. Revisar constantemente el código fuente.
4. Trabajar a un ritmo sostenible y real.
5. Afinar el proceso.

La principal ventaja del trabajo con ASD es que los requerimientos solicitados por el cliente se van implementando periódicamente, y así el tiempo se puede administrar en forma precisa conforme avanza el proyecto. [21]

### **SCRUM**

Metodología ágil desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Esta define un marco para la gestión de proyectos, utilizada exitosamente los últimos 10 años. Se centra en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos.

El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.

La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming. [22]

## **Rational Unified Process (RUP)**

El Proceso Unificado de Rational (Rational Unified Process) es un conjunto de procesos web-enabled de ingeniería de software que dan guía para conducir las actividades de desarrollo del equipo 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 un compendio de prácticas de ingeniería de software que provee a los miembros de un equipo de desarrollo un entorno de proceso configurable basado en estándares.

El ciclo de vida de RUP está constituido por tres características principales:

**Dirigido por casos de uso:** al realizar el modelado del negocio los casos de uso encapsulan los requisitos que serán definidos a partir de las expectativas de los clientes, por lo que se encargan de guiar el ciclo de vida del proyecto.

**Iterativo e incremental:** se divide el proyecto en miniproyectos donde cada uno de ellos es una iteración que contribuye al crecimiento del producto.

**Centrado en la arquitectura:** a partir de los casos de uso más relevantes para la arquitectura se desarrollan las diferentes iteraciones que propone RUP.

RUP está compuesto por nueve flujos de trabajo agrupando a las actividades en conjuntos lógicos, los 6 primeros son los flujos de ingeniería y los restantes son los conocidos como flujos de apoyo. A continuación una breve descripción de dichos flujos:

- ✓ **Modelado del negocio:** describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- ✓ **Requerimientos:** define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ **Análisis y diseño:** describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✓ **Implementación:** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ **Prueba (Testeo):** busca los defectos a lo largo del ciclo de vida.
- ✓ **Instalación:** produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ✓ **Administración del proyecto:** involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ **Administración de configuración y cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ✓ **Ambiente:** contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Estos nueve flujos de trabajo de RUP están concebidos para cuatro fases:

**Inicio:** su objetivo es determinar la viabilidad del sistema.

**Elaboración:** tiene como objetivo definir la línea base de la arquitectura.

**Construcción:** en esta etapa el objetivo es llegar a obtener una capacidad operacional inicial.

**Transición:** el objetivo es llegar a obtener una versión del producto ya lista para su instalación. Puede implicar reparación de errores.

## Ventajas de RUP

- ✓ Mitigación temprana de posibles riesgos altos.
- ✓ Progreso visible en las primeras etapas.
- ✓ Temprana retroalimentación que se ajuste a las necesidades reales.
- ✓ Gestión de la complejidad.
- ✓ Conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.

### 1 Metodología a utilizar: RUP

Al realizar un análisis de las diversas metodologías de desarrollo existentes en la actualidad se escoge a RUP como la candidata ideal ya que:

RUP está constituida por los mejores aspectos de metodologías que le sirven de marco entre las que se puede citar SCRUM.

Es una metodología diseñada para proyectos medianos y grandes, con un enfoque basado en modelos utilizando un lenguaje bien definido para tal fin, el UML. La presente investigación se realiza para un proyecto de corto plazo pero de gran envergadura al cual están vinculados diversos módulos. Por tal motivo el desarrollo debe estar regido por una metodología robusta que provea al equipo de desarrollo las mejores tácticas para la correcta documentación del sistema para lograr uniformidad en el trabajo actual y en futuras versiones.

## Conclusiones

A lo largo del presente capítulo se han manifestado los principales problemas que hoy presenta la gestión de mantenimiento en la UCI. Para ello se emitieron valoraciones sobre las ventajas a proveer por un sistema que se encargue de la gestión de mantenimiento de los proyectos productivos. Además se realizó un análisis de las diversas metodologías, herramientas y lenguajes que se utilizan actualmente en el desarrollo web y se escogió en cada caso la más adecuada según los intereses que se persiguen.

Según las consideraciones anteriores se especifica que para el desarrollo de un sistema que permita la gestión de Mantenimiento en la UCI se utilizará como lenguaje de programación del lado del servidor el PHP, utilizando el framework Symfony, el JavaScript y el CSS del lado del cliente para lograr una mejor interactividad con el usuario en el navegador. El SGBD a utilizar será PostgreSQL y el servidor web será Apache. El proceso de realización estará regido por la metodología RUP.



## **Capítulo 2 Presentación de la solución propuesta**

### **Introducción**

El presente capítulo trata lo referente a las características del sistema. A través del modelo de negocio se brinda una perspectiva del sistema a automatizar, tomando como referencia las necesidades del cliente. Se definen los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar y se realiza una descripción del sistema propuesto.

### **2.1 Modelo de negocio**

El modelo de negocio es una técnica utilizada por los desarrolladores para especificar los requisitos más importantes de un sistema. De esta forma se evita cometer errores que favorezcan el desarrollo de un sistema que no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido.

#### **2.1.1 Estado actual del negocio**

En la actualidad no existe en la UCI un sistema para la gestión de mantenimiento regido por los procedimientos establecidos por la Infraestructura Productiva de la universidad. Debido a esto, el proceso se realiza de forma manual incurriendo en errores humanos, lentificando los procesos estadísticos y generando pérdidas en recursos informáticos, económicos y humanos. El flujo da inicio cuando algún Líder de Proyecto Productivo, Asesor de Arquitectura y Tecnologías (AAT) o Especialista realiza una solicitud de mantenimiento mediante el llenado de una planilla. Esta planilla es enviada a la cuenta de correo [tecnica.ip@uci.cu](mailto:tecnica.ip@uci.cu) para que el Especialista del Grupo de Soporte al Desarrollo (EGSD) las revise y decida cual aceptar o no. Luego el EGSD confecciona la planilla de solicitud a nivel de universidad (PSMUCI), registra la información asociada y notifica al usuario si fue aceptada o no su solicitud.

### 2.1.2 Actores y trabajadores del negocio

Se entiende por **actor del negocio** a cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos que interactúan con el negocio y obtienen resultados perceptibles.

Actor	Descripción
Solicitante:	Es una generalización entre el Líder del Proyecto Productivo, el Especialista, el Asesor de Arquitectura y Tecnología del vicedecano de producción de la facultad. Inicializa el caso de uso Solicitar Mantenimiento mediante una planilla de solicitud.

Tabla 2.1 Descripción de los actores del negocio.

Un **trabajador del negocio** es una abstracción de una o más 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. Esta abstracción recibe el nombre de rol.



Trabajadores	Descripción
Especialista del GSD de la IP:	Se encarga de revisar las planillas procedentes de las facultades y elaborar la planilla a nivel de universidad (PSMUCI).

Tabla 2.2 Descripción de los trabajadores.

### 2.1.3 Diagrama de Casos de Uso del Negocio

El diagrama de casos de uso del negocio es una representación gráfica de los procesos del negocio y su interacción con los actores del negocio.

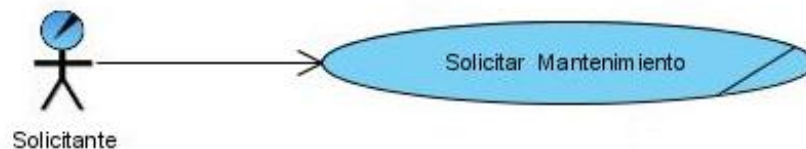


Figura 2.1 Diagrama de Casos de Uso del Negocio

## 2.2 Descripción de los procesos del negocio.

A continuación se expone la descripción del caso de uso del negocio. Para una mayor comprensión en el [Anexo 1](#) se presenta el diagrama de actividades correspondiente.

## Capítulo 2 Presentación de la solución propuesta

---

### 2.2.1 Descripción textual del Caso de Uso de Negocio

<b>Caso de Uso:</b>	Solicitar Mantenimiento.
<b>Actores:</b>	Solicitante.
<b>Trabajadores:</b>	Especialistas del GSD de la IP.
<b>Resumen:</b>	El caso de uso inicia cuando el Solicitante llena la planilla de solicitud de mantenimiento y la envía al Especialista del GSD de la IP para que sea procesada terminando el caso de uso.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1 El Solicitante llena la solicitud de mantenimiento del Proyecto Productivo llenando la planilla PSM y la envía al Especialista del GSD.	2 El Especialista del GSD de la IP recibe la planilla.
	3 Verifica que los datos recibidos sean correctos para la aprobación de la solicitud.
	4 El Especialista del GSD elabora la planilla a nivel UCI (PSMUCI) con las solicitudes aprobadas.
	5 Se le informa al Usuario de la aprobación de la solicitud.
6 Recibe la información y termina el caso de uso.	

Flujo Alternativo de los Eventos	
Acción del Actor	Respuesta del Negocio
	3.1. El especialista del GSD no aprueba la solicitud.
4.1 Recibe la información y termina el caso de uso.	
<b>Prioridad:</b>	Crítico

Tabla 2.3 Descripción de casos de uso del negocio.

### 2.2.2 Reglas del negocio

- ✓ Cada asesor de tecnología de las facultades deben crear una firma digital para firmar las planillas de forma tal que al llegar a su destino la misma no haya sido alterada ni creada por otras personas ajenas al proceso.
- ✓ En cada proyecto productivo, facultad y en el GSD deben quedar almacenadas las planillas generadas para su seguimiento y control.
- ✓ El envío de las planillas a nivel de Facultad se puede realizar en cualquier momento a la cuenta de correo [tecnica.ip@uci.cu](mailto:tecnica.ip@uci.cu).
- ✓ En todos los Consejos Técnicos se debe dar un parte de las planillas generadas a nivel UCI en la semana así como del estado de aprobación y nivel de respuesta o estado dado a las mismas.
- ✓ Las planillas a nivel de Facultad pueden ser aprobadas por el Vicedecano de Producción o el Asesor de Tecnología.
- ✓ Las planillas enviadas por el Proyecto Productivo puede o no estar firmadas digitalmente pero las planillas enviadas por el Asesor de Tecnología hacia el GSD deben contener una firma.
- ✓ Las firmas (llaves públicas) de los Asesores y Vicedecanos de Producción estarán disponibles en la dirección <http://llaves.prod.uci.cu>.

## 2.2.3 Modelo de Objeto



Figura 2.3 Modelo de Objetos del Negocio

## 2.3 Requisitos del sistema

A partir de las necesidades del cliente se desarrollara una captura de requisitos para establecer:

- ✓ Definir el ámbito del sistema.
- ✓ Definir una interfaz de usuarios para el sistema de acuerdo a las metas del cliente.
- ✓ Establecer y mantener un acuerdo entre los involucrados en el desarrollo del sistema.

### 2.3.1 Requerimientos funcionales

Son capacidades o condiciones que el sistema debe cumplir y se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

#### RF1. El sistema debe ser capaz de: Gestionar Usuario.

- ✓ Insertar usuario.
- ✓ Eliminar Usuario.
- ✓ Buscar Usuario.
- ✓ Modificar Usuario.

### **RF2. El sistema debe ser capaz de: Gestionar Docentes.**

- ✓ Insertar Docente.
- ✓ Eliminar Docente.
- ✓ Modificar Docente.
- ✓ Mostrar cantidad de Docentes de la UCI.

### **RF3. El sistema debe ser capaz de: Gestionar Facultad.**

- ✓ Insertar Facultades.
- ✓ Eliminar Facultades.
- ✓ Modificar Facultades.
- ✓ Mostrar cantidad de Facultades de la UCI.

### **RF4. El sistema debe ser capaz de: Gestionar Proyectos.**

- ✓ Insertar Proyectos Productivos.
- ✓ Eliminar Proyectos Productivos.
- ✓ Modificar Proyectos Productivos.
- ✓ Mostrar cantidad de Proyectos Productivos UCI.

### **RF5. El sistema debe ser capaz de: Gestionar Laboratorios**

- ✓ Insertar Laboratorios.
- ✓ Eliminar Laboratorios.
- ✓ Modificar Laboratorios.
- ✓ Mostrar cantidad de Laboratorios de la UCI.

### **RF6. El sistema debe ser capaz de: Gestionar Roturas.**

- ✓ Insertar Roturas.
- ✓ Eliminar Roturas.
- ✓ Modificar Roturas.

- ✓ Mostrar cantidad de Roturas existentes en la UCI.

### **RF7. El sistema debe ser capaz de: Gestionar Reportes.**

- ✓ Insertar Reportes.
- ✓ Modificar Reportes.
- ✓ Eliminar Reportes.
- ✓ Mostrar datos de los Reportes.
- ✓ Mostrar listado de Reportes en la UCI por criterio.

### **RF8. El sistema debe ser capaz de: Buscar Datos Generales.**

### **RF9. El sistema debe ser capaz de: Exportar Reportes.**

- ✓ Exportar los reportes generados a un formato PDF.

### **RF10. El sistema debe ser capaz de: Generar Reportes.**

- ✓ Generar reportes.

## **2.3.2 Requerimientos no funcionales**

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Por estas cualidades se entiende a aquellas características que hacen al producto atractivo, usable, rápido o confiable.

### **Usabilidad**

**RNF1.** El sistema debe estar disponible las veinticuatro horas del día, sin ninguna interrupción.

**RNF2.** El sistema debe ser accesible desde todos los puntos donde exista una máquina conectada a la red.

**RNF3.** El sistema debe tener una interfaz que le sea familiar al usuario para aprovechar sus conocimientos en el manejo de herramientas de software.

**RNF4.** El sistema debe ser una interfaz de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse lo más pronto posible y le sea cómodo el manejo del software.

**RNF5.** El sistema debe diferenciar las interfaces gráficas y opciones para los usuarios que accedan al sistema con diferentes roles.

### **Confiabilidad y Seguridad**

**RNF6.** El sistema debe tener la capacidad de identificar con certeza a los diversos usuarios o entidades que interactúan con él.

**RNF7.** El sistema debe tener la capacidad que tiene el sistema de darle seguridad al usuario, de que las informaciones solo serán vistas por quien esté capacitado para esto.

**RNF8.** El servicio del sistema debe tener una disponibilidad aceptable (99%).

**RNF9.** El tiempo entre fallos debe ser breve o cero, haciendo lo posible para que esto no ocurra.

**RNF10.** La base de datos debe estar fraccionada sobre varios esquemas, dividiendo así de una forma lógica las funcionalidades, evitando así la pérdida total de la información en caso de algún accidente o ataque.

**RNF11.** Permitir autenticarse a través de un Servidor de Dominio.

### **Rendimiento**

**RNF12.** El sistema debe ser capaz de mantener el mismo rendimiento y estabilidad a medida que aumenta la cantidad de datos a gestionar.

**RNF13.** El sistema debe ser capaz de correr al ser montado en una misma PC todos sus componentes tales como Gestor de bases de Datos, Aplicación Web, etc.

### **Mantenimiento y actualización**

**RNF14.** El ambiente de desarrollo donde se implementara dicho software será: Eclipse. Gestor de Base de Datos: PostgreSQL. Lenguaje de Programación: PHP. El sistema debe ser construido en un código estándar, cada procedimiento debe estar comentado.

### **Soportabilidad y operabilidad**

**RNF15.** El software podrá operar en cualquier sistema operativo debido a que se desarrollará en PHP y herramientas de software libre como es el caso de PostgreSQL. En cuanto al soporte debe tenerse en cuenta en el sistema operativo en que se esté trabajando y buscar la tecnología necesaria para las herramientas que requiere el software en el mismo.

### **Funcionalidad**

**RNF16.** Capacidades de búsqueda con velocidad apropiada. El sistema debe ser Multiplataforma.

### **Desempeño y escalabilidad**

**RNF17.** El sistema podrá soportar un gran número de clientes online, 150 clientes.

**RNF18.** El tiempo de respuestas debe ser corto con respuestas rápidas y eficientes, 3.0 segundos.

**RNF 19.** El sistema debe tener un rendimiento óptimo debido a que presta servicios a un gran número de usuarios.

### **Requerimientos de software del Sistema**

**RNF20.** Las computadoras clientes deben contar con Mozilla Firefox.

**RNF21.** El sistema operativo debe ser Linux Ubuntu.



**RNF22.** La computadora servidora debe tener instalado el SGBD PostgreSQL8.2.

**RNF23.** La computadora servidora debe tener instalado el servidor Apache2.2.

**RNF24.** La computadora servidora debe tener instalado el Framework Symfony y PHP5.

### Requerimientos de hardware del sistema

**RNF25.** Las computadoras clientes deben estar conectadas en red.

**RNF26.** Las computadoras clientes deben ser Pentium III o superior.

**RNF27.** El servidor debe ser Pentium III o superior.

**RNF28.** La computadora servidor debe tener como mínimo 13 GB de HDD y 2GB de RAM.

**RNF29.** El servidor debe tener Intel(R) PRO/1000 PM Network Connection.

### Requerimiento de ayuda y documentación

**RNF31.** El proyecto debe brindar un manual de usuario para el correcto uso de sus funcionalidades y brindarles una mayor experiencia a los usuarios del sistema.

## 2.4 Descripción del Sistema Propuesto

### 2.4.1 Descripción de los actores

Actor	Justificación
<b>Especialista del GSD de la IP:</b>	Es el encargado de configurar los perfiles de los usuarios del sistema. Además es el único que puede eliminar algún campo de la BD del sistema.
<b>Usuario:</b>	Es una generalización entre el Líder del Proyecto Productivo, el Especialista, el

## Capítulo 2 Presentación de la solución propuesta

	Asesor de Arquitectura y Tecnología del vicedecano de producción de la facultad y el especialista del GSD de la IP.
--	---

Tabla 2.4 Descripción de los actores del sistema.

### 2.4.2 Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema es una representación gráfica de la interacción entre los actores y casos de uso del sistema. Representa los requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno.

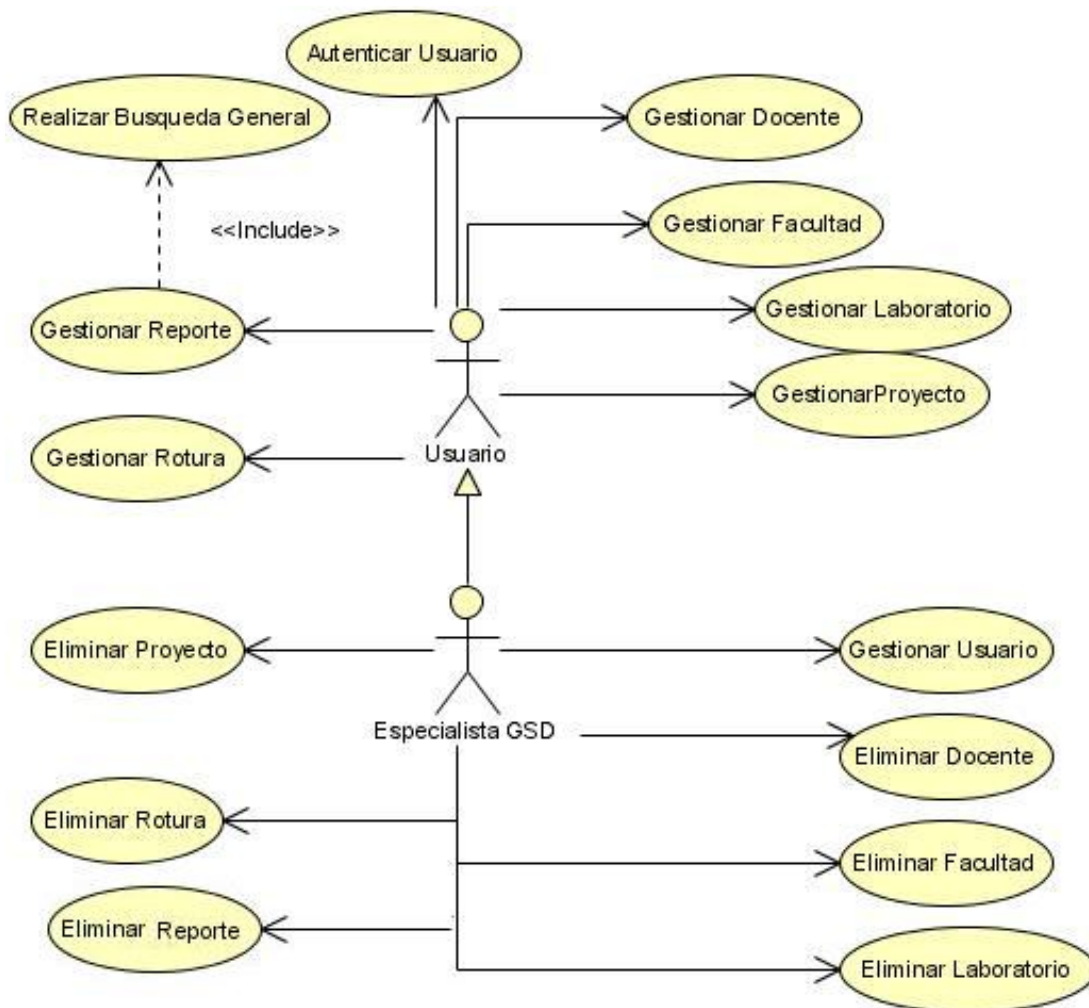


Figura 2.4 Diagrama de Casos de Uso del Sistema.

## 2.5 Descripción de los Casos de Uso del Sistema

### 2.5.1 Caso de uso Autenticar Usuario

<b>Caso de uso:</b>	Autenticar Usuario.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Permitir a los usuarios identificarse en el sistema para acceder a las opciones disponibles según su nivel de acceso.
<b>Resumen:</b>	El caso de uso permite que el usuario introduzca su usuario y contraseña para autenticarse, el sistema lo verifique y le permita ejecutar las funcionalidades según el rol que desempeñe.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	
<b>Referencias:</b>	
<b>Prioridad:</b>	Critico.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario introduce los datos necesarios (usuario, contraseña) para autenticarse y obtener los privilegios conferidos.	2. El sistema verifica la existencia del usuario.
	3. Si el Usuario existe se le da acceso a las funcionalidades definidas para él.
<b>Flujo Alternativo de los Eventos</b>	
	3.1 Si no existe se muestra un mensaje de error y debe retorna a la acción 1.

Tabla 2.5 Descripción del caso de uso Autenticar Usuario.

## Capítulo 2 Presentación de la solución propuesta

---

### 2.5.2 Caso de Uso Gestionar Usuario.

<b>Caso de uso:</b>	Gestionar Usuario.
<b>Actor:</b>	Especialista del GSD
<b>Propósito:</b>	Gestionar los usuarios del sistema, es decir, Adicionar nuevos usuarios a la aplicación, Eliminar y Modificar sus datos.
<b>Resumen:</b>	El caso de uso comienza cuando el Especialista del GSD accede a la opción de Adicionar, Eliminar, Modificar, Mostrar usuario. El caso de uso termina cuando el sistema devuelve la información solicitada y el usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El Especialista del GSD debe estar previamente autenticado. Para acceder a la opción de modificar antes se deben listar los usuarios.
<b>Poscondiciones:</b>	La base de datos debe estar actualizada con los datos insertados, modificados o eliminados del usuario.
<b>Referencias:</b>	RF.1.
<b>Prioridad:</b>	Crítico.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Especialista del GSD accede a la opción de Adicionar.	2. El sistema muestra una interfaz para que introduzca los datos del usuario (usuario y contraseña).
3. El Especialista del GSD introduce los datos y presiona el botón Adicionar.	4. El sistema verifica que los datos estén correctos.

## *Capítulo 2 Presentación de la solución propuesta*

---

	5. El sistema no encuentra errores en los datos y procede a registrar los datos del usuario mostrando un mensaje de notificación al finalizar la operación.
<b>Flujo Alternativo de los Eventos</b>	
	5.1 El sistema encuentra errores en los datos y muestra un mensaje de error.
<b>Sección Modificar Usuario</b>	
1. El Especialista del GSD accede a la opción de Modificar.	2. El sistema busca todos los usuarios existentes y los muestra en una interfaz para que el Especialista del GSD elija el usuario a modificar.
3. El Especialista del GSD elige el usuario y presiona el botón Modificar.	4. El sistema muestra una interfaz con todos los datos del usuario elegido de forma editable.
5. El Especialista del GSD modifica los datos necesarios y presiona el botón Aceptar.	6. El sistema muestra todos los datos del usuario modificados si se logró con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6.1 El sistema muestra un mensaje de error informando que no se pudo modificar los datos seleccionados.
<b>Sección Eliminar Usuario</b>	
1. El Especialista del GSD accede a la opción de Eliminar.	2. El sistema busca todos los usuarios existentes y los muestra en una interfaz para que el Especialista del GSD elija el usuario a eliminar.
3. El Especialista del GSD elige el usuario a eliminar y presiona el botón Eliminar.	4. El sistema elimina al usuario y muestra un mensaje notificando que se eliminó con éxito.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra eliminar al usuario y muestra un mensaje de error.

Tabla 2.6 Descripción del caso de uso Gestionar Usuario.

## *Capítulo 2 Presentación de la solución propuesta*

---

### 2.5.3 Caso de Uso Gestionar Docente

<b>Caso de uso:</b>	Gestionar Docente.
<b>Caso de uso relacionado:</b>	Eliminar Docente.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Gestionar los datos de los docentes de la UCI, es decir, Adicionar, Modificar y Mostrar la cantidad de docentes de la universidad.
<b>Resumen:</b>	El caso de uso comienza cuando el usuario accede a las opciones de Adicionar, Modificar y Mostrar la cantidad de docentes de la UCI. El caso de uso termina cuando el sistema realiza la operación seleccionada y el usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Se debe acceder a la opción de Listar los docentes antes de modificar.
<b>Poscondiciones:</b>	Base de datos actualizada con los datos adicionados o modificados del docente.
<b>Referencia:</b>	RF 2.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción de Adicionar del menú Docente.	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo docente.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos del nuevo docente y lo muestra ya adicionado.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar el docente y

## *Capítulo 2 Presentación de la solución propuesta*

---

muestra un mensaje de error.

## *Capítulo 2 Presentación de la solución propuesta*


<b>Sección Modificar Docente</b>	
1. El Usuario accede a la opción de Listar del menú Docente.	2. El sistema busca los docentes y los muestra en una interfaz para que el Usuario elija el docente a editar.
3. El Usuario elije el docente a modificar y presiona la opción Editar.	4. El sistema muestra una interfaz con los datos del docente de manera editable.
5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con los datos del docente modificados si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6.1 El sistema muestra un mensaje de error.
<b>Sección Mostrar Cantidad de Docentes UCI</b>	
1. El Usuario elige la opción Listar del menú Docente.	2. El sistema busca los docentes existentes y los muestra en una nueva interfaz.
<b>Prototipo de Interfaz</b>	
	

Tabla 2.7 Descripción del caso de uso Gestionar Docente.

### 2.5.4 Caso de Uso Eliminar Docente

<b>Caso de uso</b>	Eliminar Docente.
<b>Actor:</b>	Especialista del GSD.
<b>Propósito:</b>	Eliminar un determinado docente.



## *Capítulo 2 Presentación de la solución propuesta*

---

<b>Resumen:</b>	El caso de uso comienza cuando el Especialista del GSD accede a la opción de Eliminar un docente. Termina cuando el sistema realiza la operación seleccionada y el Especialista del GSD accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de Eliminar antes se deben listar los docentes.
<b>Poscondiciones:</b>	Datos del docente eliminados de la base de datos.
<b>Referencia:</b>	RF 2.
<b>Prioridad:</b>	Media.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Especialista del GSD accede a la opción Listar del menú Docente.	2. El sistema busca los docentes y los muestra en una interfaz para que el Especialistas del GSD elija el docente a eliminar.
3. El Especialista del GSD elije el docente a eliminar y presiona la opción Eliminar.	4. El sistema muestra un mensaje preguntando si en verdad desea eliminar el docente.
5. El Especialista del GSD presiona el botón Eliminar.	6. El sistema elimina el docente satisfactoriamente.
<b>Flujo Alternativo de los Eventos</b>	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	

## Capítulo 2 Presentación de la solución propuesta



Tabla 2.8 Descripción del caso de uso Eliminar Docente.

### 2.5.5 Caso de uso Gestionar Facultad

<b>Caso de uso:</b>	Gestionar Facultad.
<b>Caso de uso asociado:</b>	Eliminar Facultad.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Gestionar los datos de las facultades de la UCI, es decir, Adicionar, Modificar y Mostrar la cantidad de facultades existentes.
<b>Resumen:</b>	El caso de uso comienza cuando el Usuario accede a las opciones de Adicionar, Modificar y Mostrar la cantidad de facultades existentes. El caso de uso termina cuando el sistema realiza la operación elegida, y el usuario accede a otra opción o implemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de modificar antes se deben listar las facultades.
<b>Poscondiciones:</b>	La base de datos actualizada con la información

## *Capítulo 2 Presentación de la solución propuesta*

---

insertada, modificada o eliminada.

## *Capítulo 2 Presentación de la solución propuesta*

---

<b>Referencia:</b>	RF 3.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción Adicionar del menú Facultad.	2. El sistema muestra la interfaz para que Usuario introduzca los datos de la nueva Facultad.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos de la nueva Facultad y la muestra ya adicionada.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar la Facultad deseada y muestra un mensaje de error.
<b>Sección Modificar Facultad</b>	
1. El Usuario accede a la opción Listar del menú Facultad.	2. El sistema busca las facultades y las muestra en una interfaz para que el Usuario elija la facultad a modificar.
3. El Usuario elige la facultad a modificar y elige la opción Editar.	4. El sistema muestra una interfaz con los datos de la facultad de manera editable.
5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con el dato de la facultad modificados si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6.1 El sistema muestra un mensaje de error.
<b>Sección Cantidad de Facultades UCI</b>	
1. El Usuario elige la opción Listar del menú Facultad.	2. El sistema busca las facultades existentes y los muestra una nueva interfaz.

Prototipo de Interfaz	
<b>Adicionar Facultad</b>	
<b>Docentes UCI</b>	<input type="text" value="[-Seleccione-]"/>
<b>Nombre Facultad</b>	<input type="text" value="[-Seleccione-]"/>
<b>Nombre Asesor</b>	<input type="text"/>
<input type="button" value="Adicionar"/>	

Tabla 2.9 Descripción del caso de uso Gestionar Facultad.

### 2.5.6 Caso de uso Eliminar Facultad

<b>Caso de uso:</b>	Eliminar Facultad.
<b>Actor:</b>	Especialista del GSD.
<b>Propósito:</b>	Eliminar la información referente a determinada facultad de la UCI.
<b>Resumen:</b>	El caso de uso comienza cuando el Especialistas del GSD selecciona la opción de eliminar a una determinada facultad. El caso de uso termina cuando el sistema realiza la operación seleccionada y el Especialista del GSD accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de Eliminar antes se deben listar las facultades.
<b>Poscondiciones:</b>	
<b>Referencia:</b>	RF 3.
<b>Prioridad:</b>	Media.

## *Capítulo 2 Presentación de la solución propuesta*

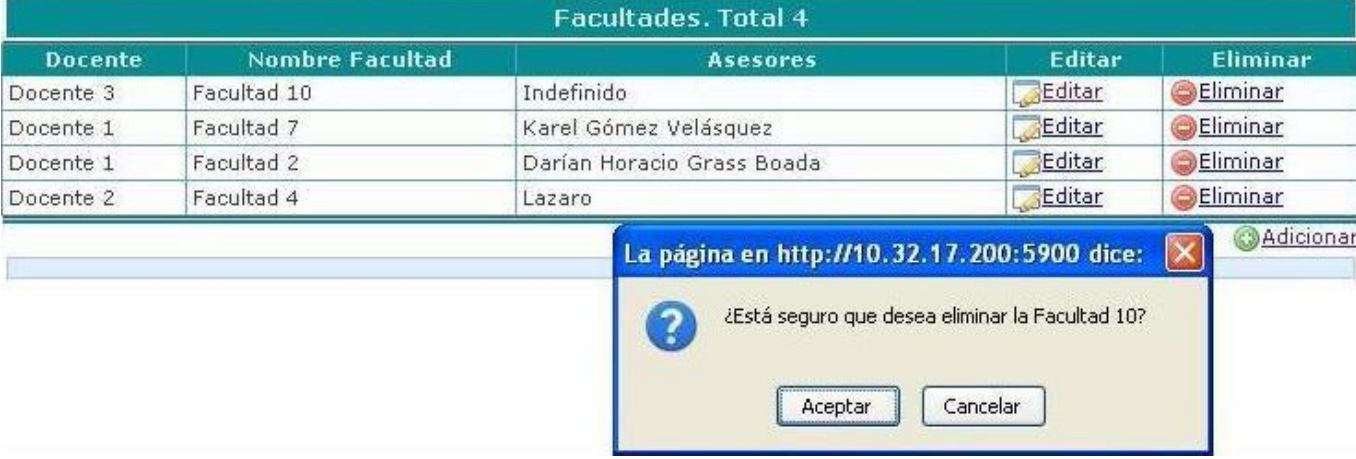
Curso normal de eventos para el caso de uso																										
Acción del actor	Respuesta del sistema																									
1. El Especialista del GSD accede a la opción Listar del menú Facultad.	2. El sistema busca las facultades y las muestra en una interfaz para que el Especialista del GSD elija la Facultad a Eliminar.																									
3. El Especialista del GSD elige la facultad a eliminar y presiona la opción Eliminar.	4. El sistema muestra un mensaje preguntando si en verdad desea eliminar la facultad.																									
5. El Especialista del GSD presiona el botón Eliminar.	6. El sistema elimina la facultad satisfactoriamente.																									
Flujo Alternativo de los Eventos																										
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.																										
Prototipo de Interfaz																										
 <p>The screenshot shows a web interface with a table titled "Facultades. Total 4". The table has five columns: "Docente", "Nombre Facultad", "Asesores", "Editar", and "Eliminar". There are four rows of data. Below the table, there is a confirmation dialog box with a question mark icon and the text "¿Está seguro que desea eliminar la Facultad 10?". The dialog has two buttons: "Aceptar" and "Cancelar".</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th style="width: 15%;">Docente</th> <th style="width: 25%;">Nombre Facultad</th> <th style="width: 35%;">Asesores</th> <th style="width: 15%;">Editar</th> <th style="width: 10%;">Eliminar</th> </tr> </thead> <tbody> <tr> <td>Docente 3</td> <td>Facultad 10</td> <td>Indefinido</td> <td> Editar</td> <td> Eliminar</td> </tr> <tr> <td>Docente 1</td> <td>Facultad 7</td> <td>Karel Gómez Velásquez</td> <td> Editar</td> <td> Eliminar</td> </tr> <tr> <td>Docente 1</td> <td>Facultad 2</td> <td>Darían Horacio Grass Boada</td> <td> Editar</td> <td> Eliminar</td> </tr> <tr> <td>Docente 2</td> <td>Facultad 4</td> <td>Lazaro</td> <td> Editar</td> <td> Eliminar</td> </tr> </tbody> </table>		Docente	Nombre Facultad	Asesores	Editar	Eliminar	Docente 3	Facultad 10	Indefinido	Editar	Eliminar	Docente 1	Facultad 7	Karel Gómez Velásquez	Editar	Eliminar	Docente 1	Facultad 2	Darían Horacio Grass Boada	Editar	Eliminar	Docente 2	Facultad 4	Lazaro	Editar	Eliminar
Docente	Nombre Facultad	Asesores	Editar	Eliminar																						
Docente 3	Facultad 10	Indefinido	Editar	Eliminar																						
Docente 1	Facultad 7	Karel Gómez Velásquez	Editar	Eliminar																						
Docente 1	Facultad 2	Darían Horacio Grass Boada	Editar	Eliminar																						
Docente 2	Facultad 4	Lazaro	Editar	Eliminar																						

Tabla 2.10 Descripción del caso de uso Eliminar Facultad.

### 2.5.7 Caso de uso Gestionar Laboratorio

<b>Caso de uso:</b>	Gestionar Laboratorio.
<b>Caso de uso asociado:</b>	Eliminar Laboratorio.
<b>Actor:</b>	Usuario.

## *Capítulo 2 Presentación de la solución propuesta*

---

<b>Propósito:</b>	Gestionar los datos de los laboratorios, es decir, Adicionar, Modificar y Mostrar la cantidad de laboratorios existentes.
<b>Resumen:</b>	El caso de uso comienza cuando el usuario accede a las opciones de Adicionar, Modificar, o Mostrar la cantidad de laboratorios existentes. El caso de uso termina cuando el sistema realiza la operación seleccionada y el usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de Modificar antes se deben listar los laboratorios.
<b>Poscondiciones:</b>	La base de datos actualizada con la información adicionada o modificada del laboratorio.
<b>Tipo:</b>	
<b>Referencia:</b>	RF 5.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción Adicionar del menú Laboratorio.	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo laboratorio.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos del nuevo laboratorio y lo muestra ya adicionado.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar el laboratorio y muestra un mensaje de error.
<b>Sección Modificar Laboratorios</b>	
1. El Usuario accede a la opción de Listar del menú Laboratorio.	2. El sistema busca los laboratorios y los muestra en una interfaz para que el Usuario elija el

## *Capítulo 2 Presentación de la solución propuesta*

---

	laboratorio a modificar.
--	--------------------------



## *Capítulo 2 Presentación de la solución propuesta*

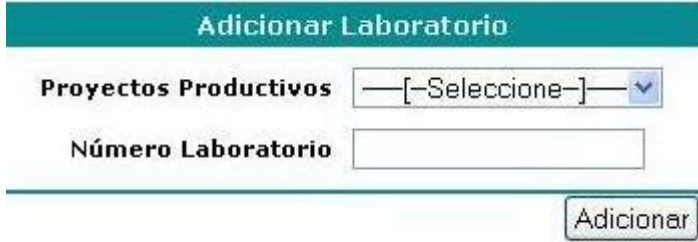
3. El Usuario elige el laboratorio a modificar y presiona la opción Editar.	4. El sistema muestra una interfaz los datos del laboratorio de manera editable.
5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con los datos del laboratorio modificados si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6.1 El sistema muestra un mensaje de error.
<b>Sección Cantidad de Laboratorios UCI</b>	
1. El Usuario elige la opción Listar del menú Laboratorio.	2. El sistema busca los laboratorios existentes y los muestra una nueva interfaz.
<b>Prototipo de Interfaz</b>	
	

Tabla 2.11 Descripción del caso de uso Gestionar Laboratorio.

### 2.5.8 Caso de uso Eliminar Laboratorio

<b>Caso de uso:</b>	Eliminar Laboratorio.
<b>Actor:</b>	Especialista del GSD
<b>Propósito:</b>	Eliminar los datos de un laboratorio.
<b>Resumen:</b>	El caso de uso comienza cuando el Especialistas del GSD accede a la opción de eliminar un laboratorio. Termina cuando el sistema realiza la operación seleccionada y el Especialista del GSD accede a otra opción o simplemente sale de la aplicación.

## *Capítulo 2 Presentación de la solución propuesta*






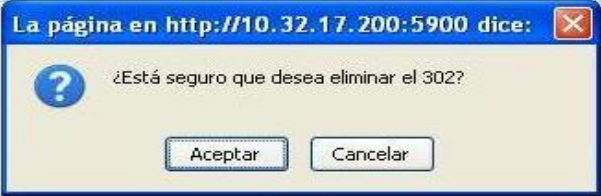
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de Eliminar antes se deben listar los laboratorios.				
<b>Poscondiciones:</b>	Datos del laboratorio eliminados de la base de datos.				
<b>Referencia:</b>	RF 5.				
<b>Prioridad:</b>	Media.				
<b>Curso normal de eventos para el caso de uso</b>					
<b>Acción del actor</b>	<b>Respuesta del sistema</b>				
1. El Especialista del GSD accede a la opción de Listar del menú Laboratorio.	2. El sistema busca los laboratorios existentes y los muestra en una interfaz para que el Especialistas del GSD elija el laboratorio a eliminar.				
3. El Especialista del GSD elije el laboratorio a eliminar y presiona la opción Eliminar.	4. El sistema muestra un mensaje preguntando si en verdad desea eliminar el laboratorio.				
5. El Especialista del GSD presiona el botón Eliminar.	6. El sistema elimina el laboratorio satisfactoriamente.				
<b>Flujo Alternativo de los Eventos</b>					
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.					
<b>Prototipo de Interfaz</b>					
<b>Laboratorios. Total 2</b>					
<b>Docente</b>	<b>Facultad</b>	<b>Proyecto</b>	<b>Laboratorio</b>	<b>Editar</b>	<b>Eliminar</b>
Docente 3	Facultad 10	DCSOFTWARE	302	 Editar	 Eliminar
Docente 2	Facultad 4	aduana	Laboratorio 305	 Editar	 Eliminar
					 Adicionar
					

Tabla 2.11 Descripción del caso de uso Eliminar Laboratorio.

## *Capítulo 2 Presentación de la solución propuesta*

---

### 2.5.9 Caso de uso Gestionar proyecto

<b>Caso de uso:</b>	Gestionar Proyecto.
<b>Caso de uso asociado:</b>	Eliminar Proyecto.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Gestionar los datos del proyecto productivo, es decir, Adicionar, Modificar y Mostrar la cantidad de proyectos productivos de la universidad.
<b>Resumen:</b>	El caso de uso comienza cuando el usuario accede a las opciones de Adicionar, Modificar o Mostrar la cantidad de proyectos productivos existentes. El caso de uso termina cuando el sistema realiza la operación seleccionada y el usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de modificar antes se deben listar los proyectos.
<b>Poscondiciones:</b>	La base de datos actualizada con la información insertada o modificada del Proyecto.
<b>Referencia:</b>	RF 4.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción de Adicionar del menú Proyecto.	1. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo Proyecto productivo.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos del nuevo proyecto productivo y lo muestra ya adicionado.

## *Capítulo 2 Presentación de la solución propuesta*

<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar la solicitud y muestra un mensaje de error.
<b>Sección Modificar Proyectos Productivos</b>	
1. El Usuario accede a la opción Listar del menú Proyectos.	2. El sistema busca los datos de los proyectos productivos existentes y los muestra en una interfaz para que el Usuario elija el proyecto productivo a modificar.
3. El Usuario elige el proyecto productivo a Modificar y presiona la opción Editar.	4. El sistema muestra una pantalla los datos del proyecto de manera editable.
5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con todos los datos del proyecto modificados si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6. El sistema muestra un mensaje de error.
<b>Sección Cantidad de Proyectos Productivos UCI</b>	
1. El Usuario elige la opción Listar del menú Proyectos.	2. El sistema busca los proyectos productivos existentes y los muestra una nueva interfaz.
<b>Prototipo de Interfaz</b>	
<div style="background-color: #008080; color: white; padding: 5px; margin-bottom: 10px; display: inline-block;"><b>Adicionar Proyectos Productivos</b></div> <div style="display: flex; flex-direction: column; align-items: center; gap: 10px;"> <div style="display: flex; align-items: center; gap: 10px;"> <span><b>Facultades</b></span> <input style="border: 1px solid #ccc; padding: 2px 10px;" type="text" value="[-Seleccione-]"/> </div> <div style="display: flex; align-items: center; gap: 10px;"> <span><b>Nomenclatura</b></span> <input style="border: 1px solid #ccc; width: 150px; height: 20px;" type="text"/> </div> <div style="display: flex; align-items: center; gap: 10px;"> <span><b>Nombre</b></span> <input style="border: 1px solid #ccc; width: 150px; height: 20px;" type="text"/> </div> <div style="display: flex; align-items: center; gap: 10px;"> <span><b>Descripción</b></span> <input style="border: 1px solid #ccc; width: 150px; height: 20px;" type="text"/> </div> <div style="display: flex; align-items: center; gap: 10px;"> <span><b>Nombre del Líder</b></span> <input style="border: 1px solid #ccc; width: 150px; height: 20px;" type="text"/> </div> </div> <div style="text-align: right; margin-top: 10px;"> <input style="border: 1px solid #ccc; padding: 5px 15px;" type="button" value="Adicionar"/> </div>	

Tabla 2.12 Descripción del caso de uso Gestionar Proyecto.

## *Capítulo 2 Presentación de la solución propuesta*

---

### 2.5.10 Caso de uso Eliminar Proyecto

<b>Caso de uso:</b>	Eliminar Proyecto.
<b>Actor:</b>	Especialista del GSD.
<b>Propósito:</b>	Eliminar los datos de un proyecto productivo.
<b>Resumen:</b>	El caso de uso comienza cuando el Especialista del GSD accede a la opción de eliminar una computadora de un proyecto productivo de la universidad. Termina cuando el sistema realiza la operación seleccionada y el Especialista accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de eliminar antes se deben listar los proyectos.
<b>Poscondiciones:</b>	Datos del proyecto eliminados de la base de datos.
<b>Referencia:</b>	RF 4.
<b>Prioridad:</b>	Media.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Especialista del GSD accede a la opción Listar del menú Proyecto.	2. El sistema busca los proyectos productivos existentes y los muestra en una interfaz para que el Especialista del GSD elija el proyecto productivo a eliminar.
3. El Especialista del GSD elije el proyecto productivo a eliminar y presiona la opción Eliminar.	4. El sistema muestra un mensaje preguntando si en verdad desea eliminar el proyecto.
5. El Especialista del GSD presiona el botón Eliminar.	6. El sistema elimina el proyecto.

## *Capítulo 2 Presentación de la solución propuesta*

Flujo Alternativo de los Eventos				
5.1. Si se cancela la acción se culmina el CUS sin ejecutar ninguna acción.				
Prototipo de Interfaz				
Proyectos Productivos. Total 2				
Nomenclatura	Descripción	Líder	Editar	Eliminar
aduana	la ad	ernesto	Editar	Eliminar
DCSOFTWARE	DCSoft	Indefinido	Editar	Eliminar
				Adicionar

La página en http://10.32.17.200:5900 dice:

?

¿Está seguro que desea eliminar el Proyecto aduana?

Aceptar
Cancelar

Tabla 2.13 Descripción del caso de uso Eliminar Laboratorio.

### 2.5.11 Caso de uso Gestionar Rotura

<b>Caso de uso:</b>	Gestionar Rotura.
<b>Caso de uso asociado:</b>	Eliminar Rotura.
<b>Actor:</b>	Usuario
<b>Propósito:</b>	Gestionar los datos de las roturas, es decir, Adicionar, Modificar y Mostrar la cantidad de roturas existentes en la UCI.
<b>Resumen:</b>	El caso de uso comienza cuando el usuario accede a las opciones de Adicionar, Modificar o Mostrar la cantidad de roturas. Termina cuando el sistema realiza la operación seleccionada y el usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de modificar antes se deben listar las

## *Capítulo 2 Presentación de la solución propuesta*

---

roturas.

## *Capítulo 2 Presentación de la solución propuesta*

---

<b>Poscondiciones:</b>	Base de datos actualizada con los datos de la rotura adicionados o modificados.
<b>Referencia:</b>	RF 6.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción Adicionar del menú Rotura.	1. El sistema muestra la interfaz para que el Usuario introduzca los datos de la nueva rotura.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos de la rotura y la muestra ya adicionada.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar la rotura y muestra un mensaje de error.
<b>Sección Modificar Rotura</b>	
1. El Usuario accede a la opción Listar del menú Rotura.	2. El sistema busca las roturas existentes y las muestra en una interfaz para que el Usuario elija la Rotura a modificar.
3. El Usuario elige la rotura a modificar y presiona la opción Editar.	4. El sistema muestra en una interfaz los datos de la rotura de manera editable.
5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con los datos de la rotura modificada si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6. El sistema muestra un mensaje de error.
<b>Sección Cantidad de Roturas</b>	
1. El Usuario elige la opción Listar del menú Rotura.	2. El sistema busca las roturas existentes y las muestra una nueva interfaz.



Prototipo de Interfaz	
	<div style="background-color: #008080; color: white; padding: 5px; margin-bottom: 5px; display: inline-block;">Adicionar Elemento Roto</div> <p>Elemento <input style="width: 150px;" type="text"/></p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Adicionar"/> </div>

Tabla 2.14 Descripción del caso de uso Gestionar Rotura.

### 2.5.12 Caso de uso Eliminar Rotura

<b>Caso de uso:</b>	Eliminar Rotura.
<b>Actor:</b>	Especialista del GSD.
<b>Propósito:</b>	Eliminar los datos de las roturas existentes en la UCI.
<b>Resumen:</b>	El caso de uso comienza cuando el Especialistas del GSD accede a la opción de eliminar roturas. Termina cuando el sistema realiza la operación seleccionada y el Especialista del GSD accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de eliminar antes se deben listar las roturas.
<b>Poscondiciones:</b>	Datos de la rotura eliminados de la base de datos.
<b>Referencia:</b>	RF 6.
<b>Prioridad:</b>	Media.
<b>Curso normal de para el caso de uso eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Especialista del GSD accede a la opción Listar del menú Rotura.	2. El sistema busca las roturas y las muestra en una interfaz para que el Especialista del GSD elija la rotura a eliminar.
3. El Especialista del GSD elije la rotura a eliminar	4. El sistema muestra un mensaje preguntando si

## *Capítulo 2 Presentación de la solución propuesta*

---

y presiona la opción Eliminar.

en verdad desea eliminar la rotura.

## *Capítulo 2 Presentación de la solución propuesta*

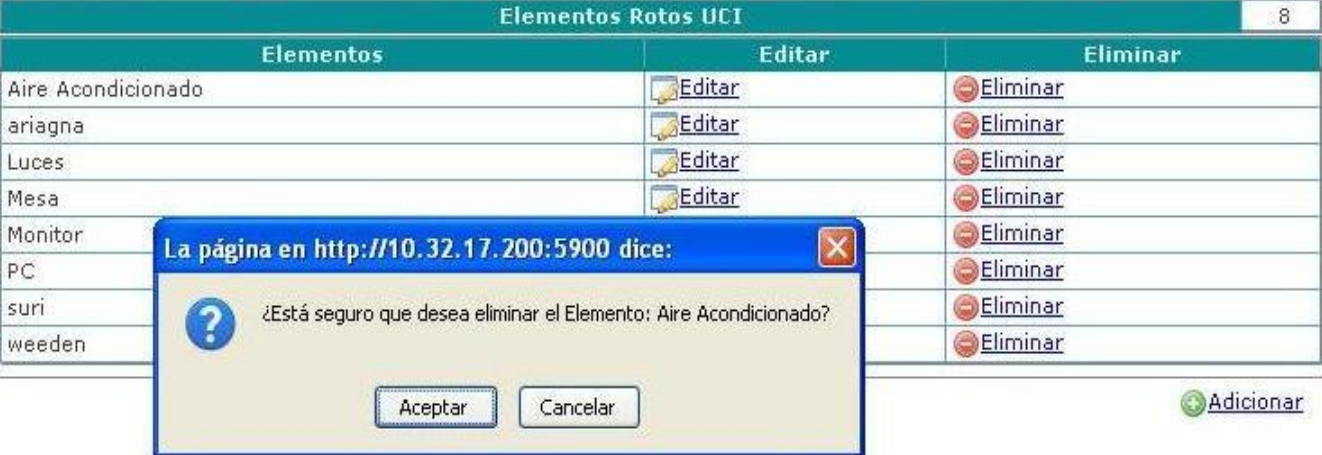
5. El Especialista del GSD presiona el botón Eliminar.	6. El sistema elimina la rotura satisfactoriamente.
<b>Flujo Alternativo de los Eventos</b>	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
<b>Prototipo de Interfaz</b>	
	

Tabla 2.15 Descripción del caso de uso Eliminar Rotura.

### 2.5.13 Caso de uso Gestionar Reporte

<b>Caso de uso:</b>	Gestionar Reporte.
<b>Caso de uso asociado:</b>	Realizar Búsqueda General.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Gestionar los datos de los reportes, es decir, Adicionar, Modificar, Exportar Reportes y Mostrar listado de reportes por criterio.

## *Capítulo 2 Presentación de la solución propuesta*

---

<b>Resumen:</b>	El caso de uso comienza cuando el usuario accede a las opciones de Adicionar, Modificar, Exportar Reportes y Mostrar listado de reportes por criterio. El caso de uso termina cuando el sistema realiza la operación seleccionada y el Usuario accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para adicionar un nuevo reporte se debe realizar una búsqueda general. Para acceder a la opción de modificar antes se deben listar los reportes.
<b>Poscondiciones:</b>	Base de datos actualizada con los datos del reporte adicionados o modificados.
<b>Referencia:</b>	RF.7, RF.8, RF9, RF.10.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción de Adicionar del menú Reporte.	1. El sistema muestra la interfaz para que el Usuario introduzca los datos del reporte.
3. El Usuario introduce los datos y presiona el botón Adicionar.	4. El sistema registra los datos del reporte y lo muestra ya adicionado.
<b>Flujo Alternativo de los Eventos</b>	
	4.1 El sistema no logra adicionar el reporte y muestra un mensaje de error.
<b>Sección Modificar Reporte</b>	
1. El Usuario accede a la opción Listar del menú Reporte.	2. El sistema busca los reportes existentes y los muestra en una interfaz para que el Usuario elija datos a modificar.
3. El Usuario elige el reporte a modificar y presiona la opción Editar.	4. El sistema muestra una interfaz con los datos del reporte de manera editable.

## *Capítulo 2 Presentación de la solución propuesta*

5. El Usuario modifica los datos necesarios y presiona el botón Editar.	6. El sistema le muestra una interfaz con el reporte modificado si se logra con éxito la operación.
<b>Flujo Alternativo de los Eventos</b>	
	6.1 El sistema muestra un mensaje de error.
<b>Sección Generar listado de Reportes por criterio</b>	
1. El Usuario accede a la opción de Listar del menú Reportes.	2. El sistema muestra los criterios existentes para listar los reportes.
3. El Usuario elige el criterio y presiona el botón Listar.	4. El sistema busca los reportes existentes que cumplan con el criterio seleccionado y los muestra en una nueva interfaz.
<b>Sección Exportar Reporte</b>	
1. El Usuario genera una listado por criterio.	2. El sistema muestra la opción de exportar este listado como PDF.
3. El usuario presiona el botón Exportar.	4. El sistema exporta el listado generado directamente a PDF.
<b>Flujo Alternativo de los Eventos</b>	
	2.1 El sistema no puede exportar reportes mostrando un mensaje de error.
<b>Prototipo de Interfaz</b>	
<p>El prototipo de interfaz para 'Adicionar Reporte' muestra un formulario con los siguientes elementos:</p> <ul style="list-style-type: none"> <li><b>Encabezado:</b> 'Adicionar Reporte'.</li> <li><b>Elemento:</b> Un menú desplegable con el texto '[- Selecciona -]'.</li> <li><b>Roto:</b> Un campo de texto vacío.</li> <li><b>Falta:</b> Un campo de texto vacío.</li> <li><b>Fecha Inicio:</b> Un campo de texto con un icono de calendario.</li> <li><b>Responsable:</b> Un campo de texto grande y vacío.</li> <li><b>Observación:</b> Un campo de texto grande y vacío.</li> <li><b>Reporte:</b> Un campo de texto grande y vacío.</li> <li><b>Botón:</b> 'Reportar' ubicado en la esquina inferior derecha.</li> </ul>	

Tabla 2.15 Descripción del caso de uso Gestionar Reporte.

## *Capítulo 2 Presentación de la solución propuesta*

---

### 2.5.14 Caso de uso Eliminar Reporte

<b>Caso de uso:</b>	Eliminar Reporte.
<b>Actor:</b>	Especialista de GSD
<b>Propósito:</b>	El caso de uso comienza cuando el Especialistas del GSD accede a la opción de eliminar reportes. El caso de uso termina cuando el sistema realiza la operación seleccionada y el Especialista del GSD accede a otra opción o simplemente sale de la aplicación.
<b>Precondiciones:</b>	El actor debe estar autenticado. Para acceder a la opción de Eliminar antes se deben listar los reportes.
<b>Poscondiciones:</b>	Datos del reporte eliminados de la base de datos.
<b>Referencia:</b>	RF.7.
<b>Prioridad:</b>	Media.
<b>Curso normal de para el caso de uso eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Especialista del GSD accede a la opción de Listar del menú Reporte y escogerá un criterio de búsqueda.	2. El sistema realizará una búsqueda según el criterio de búsqueda escogido por el actor.
3. El Especialista del GSD escogerá el reporte a eliminar.	4. El sistema mostrará un mensaje pidiendo confirmación.
5. El Especialista del GSD acepta eliminar el reporte.	6. El reporte es eliminado.
<b>Flujo Alternativo de los Eventos</b>	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	

## *Capítulo 2 Presentación de la solución propuesta*

---

### Prototipo de Interfaz

Tabla 2.16 Descripción del caso de uso Eliminar Reporte.

#### 2.5.15 Caso de uso Buscar Datos Generales

<b>Caso de uso:</b>	Buscar Datos Generales <<include>>.
<b>Actor:</b>	Usuario.
<b>Propósito:</b>	Realizar búsqueda de un laboratorio para adicionar un reporte.
<b>Resumen:</b>	El caso de uso inicia cuando el Usuario accede a la opción de adicionar un reporte.
<b>Precondiciones:</b>	El actor debe estar autenticado
<b>Poscondiciones:</b>	
<b>Referencia:</b>	RF 9.
<b>Prioridad:</b>	Critica.
<b>Curso normal de eventos para el caso de uso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Usuario accede a la opción de Adicionar un reporte.	2. El sistema muestra la interfaz para que el usuario seleccione los criterios de la nueva búsqueda.
3. El Usuario selecciona los criterios y presiona el botón Buscar.	4. El sistema muestra un reporte a llenar para la búsqueda realizada.
<b>Flujo Alternativo de los Eventos</b>	
	2.1 El sistema no tiene especificado el laboratorio al cual pertenece el proyecto productivo por lo que no permite adicionar el reporte.

## Capítulo 2 Presentación de la solución propuesta

---

Prototipo de Interfaz		
Buscar Laboratorio...		
Facultades	Proyectos	Laboratorios
<input type="text" value="[-Seleccione-]"/>	<input type="text" value="[-Seleccione-]"/>	<input type="text" value="[-Seleccione-]"/>
		<input type="button" value="Buscar..."/>

Tabla 2.17 Descripción del caso de uso Buscar Datos Generales.

### Conclusiones

En este capítulo se realizó el estudio del negocio definiendo caso de uso, descripción de caso de uso, actores y trabajadores. De igual forma se definen los casos de uso del sistema, actores y descripciones en vías de realizar la mejor propuesta. Para esto se tuvo presente los intereses del cliente.



## **Capítulo 3 “Construcción de la solución propuesta”**

### **Introducción**

En este capítulo se desarrollarán los artefactos correspondientes al análisis, diseño e implementación de la aplicación. Se realizarán los diagramas de clases del análisis, las clases del diseño con estereotipos Web, el diseño de la Base de Datos, el diagrama de despliegue y el diagrama de componentes correspondiente al modelo de implementación. Además se realizará una especificación de los patrones GRASP a utilizar así como también una argumentación de la implementación del patrón arquitectónico MVC implementado por Symfony.

### **3.1 Modelo de Clases del Análisis**

Este modelo se utiliza para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del Modelo de Diseño y se centra en los requerimientos no funcionales. No es un diagrama final que describe todos los posibles conceptos y sus relaciones, es un primer intento por definir los conceptos claves que describen el sistema. Simplifica las actividades de diseño e implementación, delimitando los temas que debe resolverse y las decisiones que deben tomarse en esas actividades.

#### **3.1.1 Diagrama de Clases del Análisis**

En el [Anexo 2](#) aparecen los Diagramas de Clases del Análisis correspondientes.

#### **3.1.2 Diagrama de Interacción del Análisis**

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un módulo de un sistema, mostrando el conjunto de objetos que participan y sus relaciones, incluyendo los mensajes que intercambian entre ellos. Los diagramas de interacción se clasifican en dos tipos: los diagramas de Colaboración y los diagramas de Secuencia.

# *Capítulo 3 Construcción de la solución propuesta*

---

## **1 Diagrama de Secuencia**

Los diagramas de secuencia muestran los objetos que participan en una interacción a través de sus líneas de vida y los mensajes que intercambian entre ellos a medida que el tiempo transcurre.

## **2 Diagrama de Colaboración**

Muestra cómo las instancias específicas de las clases trabajan juntas para alcanzar un objetivo común. En ellos se detalla de cierta forma las asociaciones que se muestran en un diagrama de clases, describiendo el intercambio de mensajes entre objetos y sus relaciones, sin tomar en cuenta la dimensión temporal de dichas relaciones.

En el [Anexo 3](#) aparecen los Diagramas de Colaboración correspondientes.

## **3.2 Modelo de Diseño**

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. Representa a los casos de uso en el dominio de la solución.

El Modelo de Diseño puede contener: los diagramas, las clases, paquetes, subsistemas, capsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo.

### **3.2.1 Patrones**

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, y entonces describe el núcleo de la solución a ese problema, de una manera que pueda usar esta solución un millón de veces, sin hacerlo nunca, de la misma manera dos veces. [23]

## *Capítulo 3 Construcción de la solución propuesta*

---

Un patrón es un conjunto de problema solución, es un amplio repertorio de principios generales basados en la experiencia que guían la creación de un software.

### **1 Patrones para asignación de responsabilidades**

Patrones básicos de GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) El nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente un software orientado a objetos.

A continuación se argumentan estos patrones:

<b>Patrón</b>	<b>Problema</b>	<b>Solución</b>
<b>Experto</b>	¿Cuál es un principio general para asignar responsabilidades a los objeto?	Asignar una responsabilidad al experto en información

## *Capítulo 3 Construcción de la solución propuesta*

<p><b>Creador</b></p>	<p>¿Quién debería ser el responsable de la creación de una nueva instancia de una clase?</p>	<p>Asignar a la clase B crear las instancias de A si:</p> <ul style="list-style-type: none"> <li>• B agrega los objetos de A</li> <li>• B contiene los objetos de A</li> <li>• B registra las instancias de los objetos de A.</li> <li>• B tiene los datos de inicialización que serán enviados a A cuando este objeto sea creado.</li> </ul>
<p><b>Controlador</b></p>	<p>¿Quién debería encargarse de atender un evento del sistema?</p>	<p>Asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase que represente alguna de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• El sistema global.</li> <li>• La empresa u organización global.</li> <li>• Algo activo en el mundo real que pueda participar en la tarea.</li> <li>• Un manejador artificial de todos los eventos del sistema de un</li> </ul>

## *Capítulo 3 Construcción de la solución propuesta*

		caso de uso (controlador de casos de uso)
<b>Bajo acoplamiento</b>	¿Cómo dar soporte a una mínima dependencia y a un aumento de la reutilización?	Asignar la responsabilidad de forma que el acoplamiento permanezca bajo.
<b>Alta cohesión</b>	¿Cómo mantener la complejidad manejable?	Asignar la responsabilidad de forma que la cohesión permanezca bajo.

Tabla 3.1 Patrones básicos de GRASP

## *Capítulo 3 Construcción de la solución propuesta*

---

### 2 ¿Cómo se evidencian estos patrones?

#### **Experto**

Propel es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5, este presenta un completo grado de integración en Symfony. Esta librería se encarga de la generación automática de las clases de la capa del modelo en función de la estructura de datos de la aplicación. Cada una de estas clases es experta en el manejo de su información.

#### **Controlador**

El controlador frontal ofrece un punto de entrada único para toda la aplicación encargándose del manejo de todas las peticiones del usuario.

#### **Creador**

El directorio actions contiene un único archivo llamado actions.class.php y que corresponde a la clase que almacena todas las acciones del módulo. Estas acciones se encargan de crear los objetos de las clases que representan las entidades.

#### **Patrones GOF utilizados**

**Singleton** (Instancia única) Garantizar que una clase posea una sola instancia y proporcionar un punto de acceso global a la misma. [24]

Este patrón se evidencia al invocar la función `sfContext::getInstance()` en el controlador frontal garantizando acceder siempre a la misma instancia.

**Decorator** (Decorador): Adhiere responsabilidades adicionales a un objeto dinámicamente. Ofrece una alternativa flexible a subclases para extender la funcionalidad. Este patrón se evidencia cuando una plantilla es decorada después de que el contenido es mostrado por una plantilla global, llamada layout en Symfony.

## Capítulo 3 Construcción de la solución propuesta

---

### 3 Descripción de la implementación del patrón MVC por Symfony

En el desarrollo de la aplicación Web se decidió utilizar el framework Symfony por las diversas facilidades que brinda tales como un código legible y reutilizable, optimización del desarrollo y facilidad de mantenimiento a la aplicación.

Symfony contiene una implementación sencilla pero eficiente del MVC. De esta forma se garantiza que el desarrollo Web sea de manera ágil. En un inicio el controlador frontal y el layout son comunes para todas las acciones de la aplicación. A pesar de que solamente es obligatorio tener uno de cada uno, se pueden tener varios controladores y varios layouts. El controlador frontal es un componente que sólo tiene código relativo al MVC y es generado de forma automática por Symfony. Este es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.

El Layout es también denominado plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación. Es utilizado para mostrar elementos tales como la navegación y el logotipo del sitio. Es posible definir más de un layout y decidir que layout utilizar en cada caso. Las plantillas son la presentación de los datos de la acción que se está ejecutando. Symfony tiene una relación curiosa entre layouts y plantillas. Se plantea que el contenido de la plantilla se integra en el layout, o que el layout *decora* la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “decorator”.

Existen numerosas herramientas y utilidades para manipular la capa correspondiente a la presentación. Las plantillas se pueden construir en pocos segundos, gracias al uso de los helpers. Los helpers son funciones de PHP que devuelven código HTML. Los layouts, los elementos parciales, los componentes y los slots de componentes permiten aplicar los conceptos de modularidad y reutilización de componentes. La configuración de la vista aprovecha la velocidad de YAML para manejar la mayoría de cabeceras de las páginas. La

## *Capítulo 3 Construcción de la solución propuesta*

---

configuración en cascada evita tener que definir todas las opciones para cada vista. Si una modificación de la presentación requiere el uso de datos dinámicos, se puede realizar la modificación en la acción mediante el objeto `sfResponse`. Además, la vista puede protegerse ante ataques de tipo XSS gracias al mecanismo de escape de los datos de las variables.

Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. Esta generación automática es responsabilidad de la librería Propel ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario. Cuando Propel encuentra restricciones de claves foráneas o encuentra datos de tipo fecha, crea métodos especiales para acceder y modificar esos datos, por lo que la manipulación de datos se torna sencilla. La abstracción de la base de datos es completamente invisible al programador porque la realiza un componente llamado Creole. Así, en caso de que ocurra un cambio del sistema gestor de bases de datos no se debe reescribir ni una línea de código, sólo es necesario modificar un parámetro en un archivo de configuración.

### **3.3 Diagramas Clases del Diseño**

En el [Anexo 4](#) aparecen los Diagramas de Clases del Diseño correspondientes.

### **Diagramas de secuencia del diseño**

Un diagrama de secuencia del diseño muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Si los casos de uso tienen varios flujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos.

### **3.4 Diseño de la Base de Datos**

El diseño de la Base de Datos es un aspecto primordial dentro del proceso de desarrollo de software. Para desarrollar este diseño se deben tener en cuenta un conjunto de aspectos que



# Capítulo 3 Construcción de la solución propuesta

influyen de forma relevante sobre este. Por ejemplo el número de registros y campos que serán almacenados. También se definen las clases del diseño que integraran el diagrama de clases persistentes para el posterior desarrollo del modelo de datos del sistema.

## 3.4.1 Diagrama de Clases Persistentes

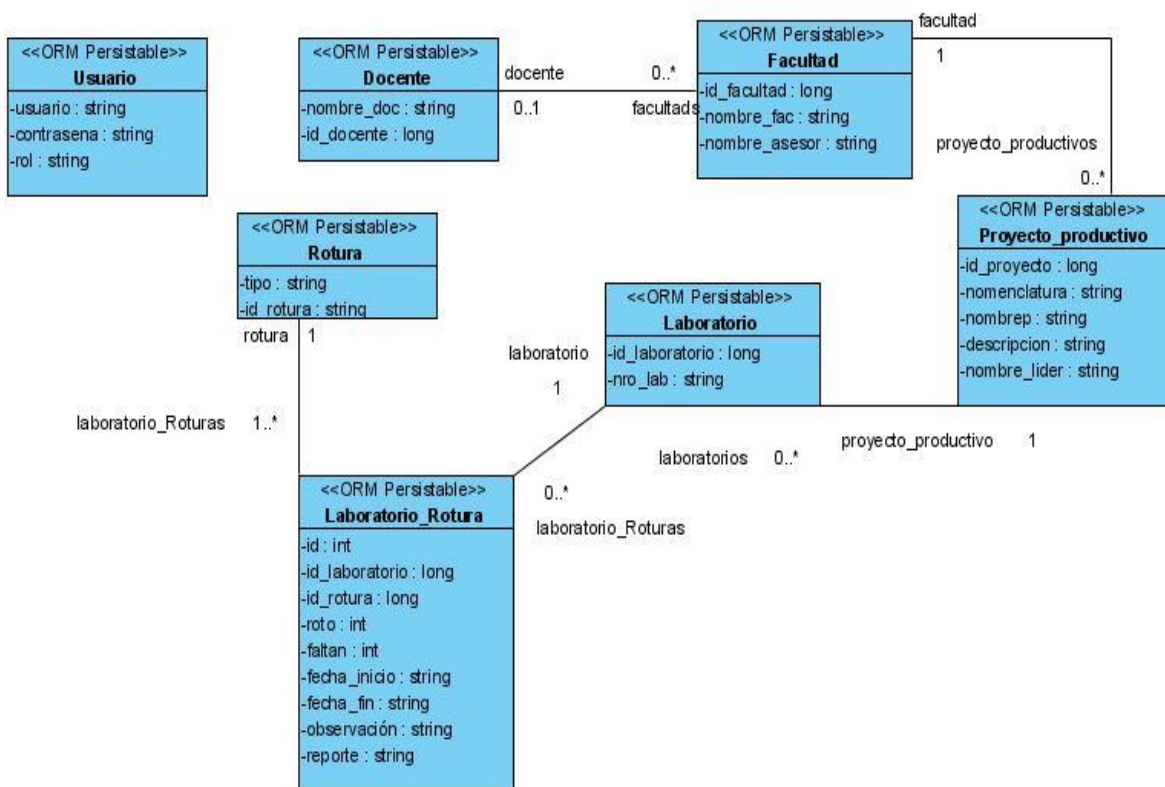


Figura 3.65 Diagrama de Clases Persistentes.

# Capítulo 3 Construcción de la solución propuesta

## 3.4.2 Diagrama Entidad Relación

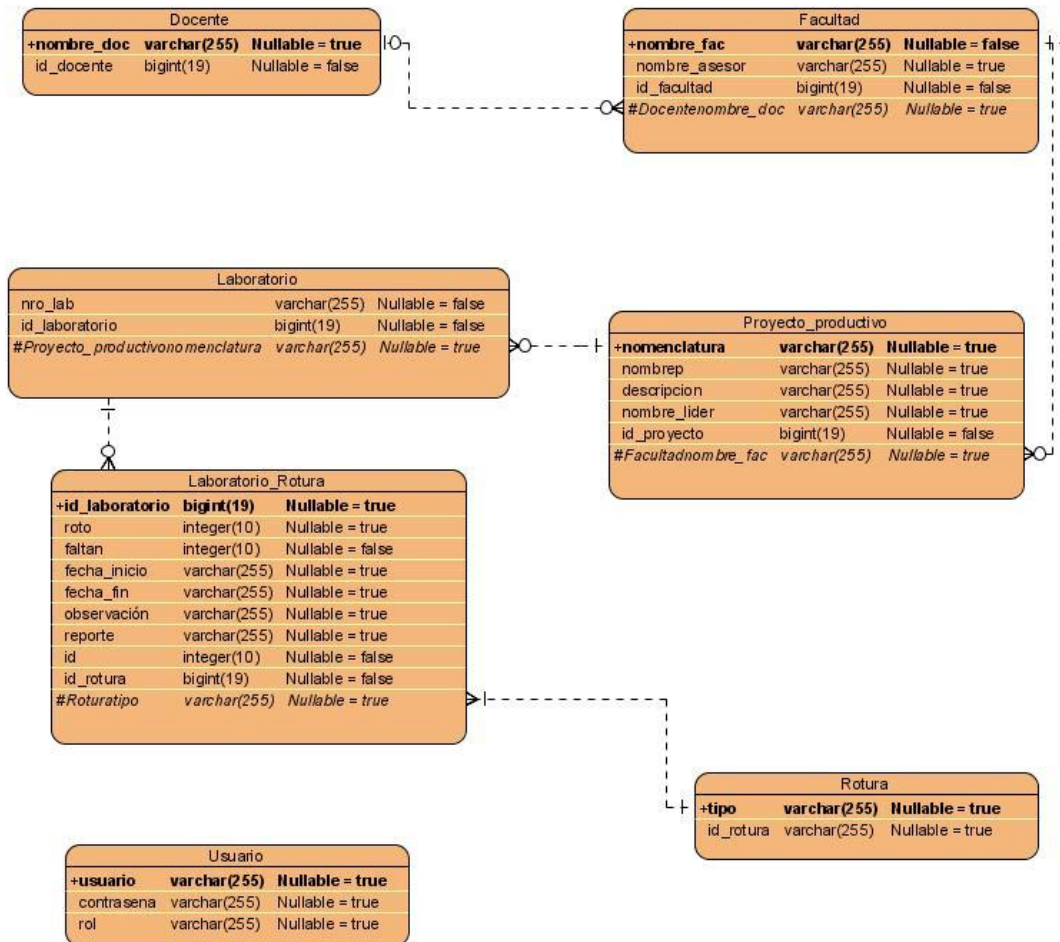


Figura 3.66 Diagrama Entidad Relación.

## 3.5 Modelo de Despliegue

El Modelo de Despliegue define la arquitectura física del sistema. Captura las características físicas del sistema como son hardware, sistemas operativos, capacidades de la red, software de interfaces y soporte conformarán el nuevo sistema, donde se desplegará y que parámetros aplica para recuperarse de los desastres, confiabilidad, copias de seguridad y soporte.

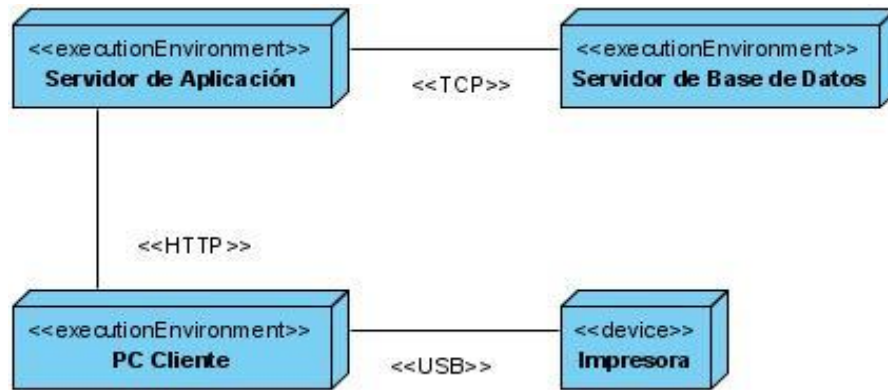


Figura 3.67 Modelo de Despliegue.

### 3.6 Modelo de Implementación

El Modelo de Implementación está integrado por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos. Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes.

#### 3.6.1 Diagrama de Componentes

Un Diagrama de Componentes ilustra los fragmentos de software, controladores embebidos y demás elementos que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase - usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de

# Capítulo 3 Construcción de la solución propuesta

construcción, como así eventualmente un componente puede comprender una gran porción de un sistema. [25]

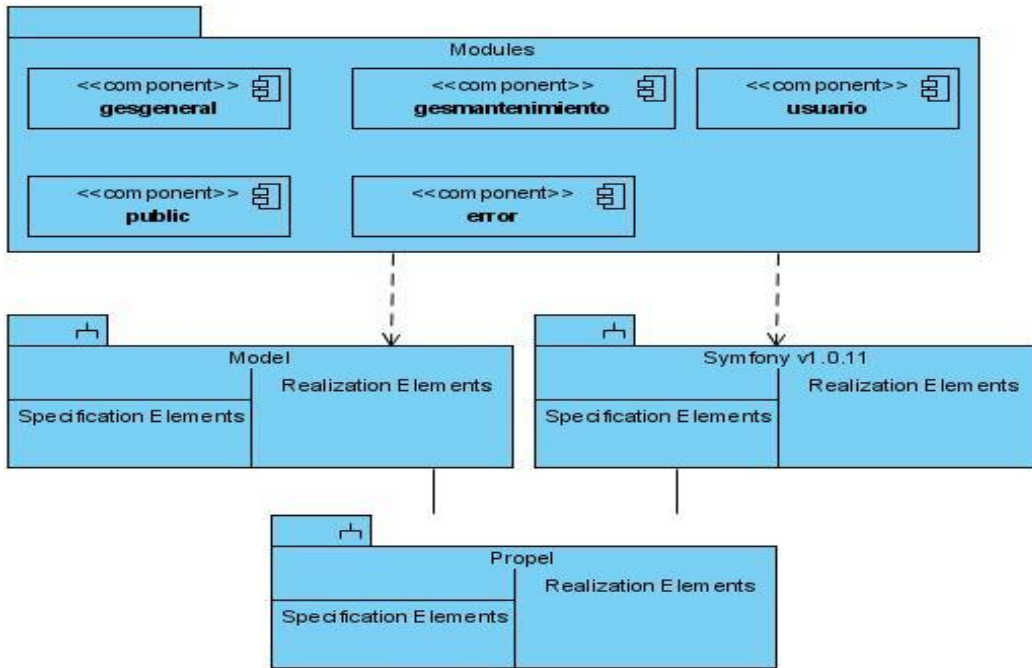


Figura 3.68 Diagrama de Componentes.

## Vista detallada: Paquete Model

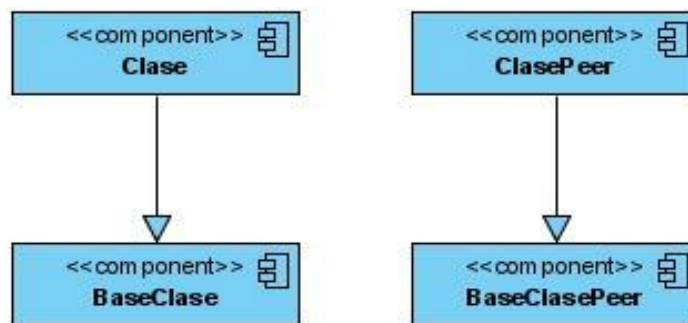


Figura 3.69 Paquete Model.

### Vista detallada: Paquete Módulo.

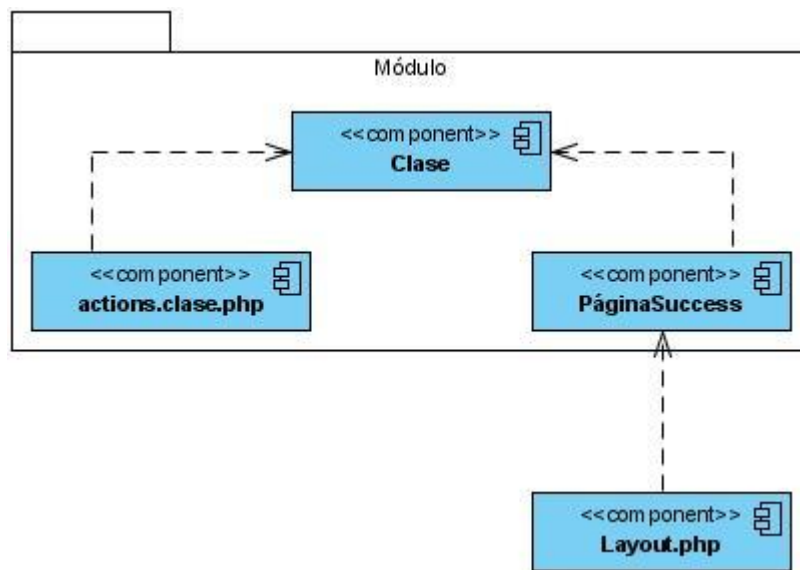


Figura 3.70 Paquete Módulo.

### Conclusiones

En este capítulo se obtuvieron los artefactos del análisis, diseño e implementación de la aplicación. Se realizaron diagramas de clases del análisis, las clases del diseño con estereotipos Web, el diseño de la Base de Datos, el diagrama de despliegue y el diagrama de componentes correspondiente al modelo de implementación. Además se hace una especificación de los patrones GRASP utilizados así como también una argumentación de la implementación del patrón arquitectónico MVC implementado por Symfony.

### **Conclusiones**

- ✓ La Gestión de Mantenimiento juega un papel fundamental en los procesos de gestión tecnológica que despliega el grupo de Soporte al Desarrollo de la Infraestructura Productiva de la UCI.
- ✓ Para el correcto desarrollo del Sistema de Mantenimiento se aplicaron técnicas de Ingeniería de Software.
- ✓ Se evaluaron las metodologías, tecnologías y tendencias actuales permitiendo la mejor selección en cada caso.
- ✓ Mediante el análisis y el diseño se determinó la arquitectura a construir, así como todos los patrones de diseño a utilizar.
- ✓ Se desarrolló la implementación de un Sistema para la Gestión de Mantenimiento de los proyectos productivos en la UCI que cumple con los requisitos planteados por los Especialistas del Grupo de Soporte al Desarrollo.

### **Recomendaciones**

- ✓ Agregar nuevas funcionalidades al Sistema de Mantenimiento.
- ✓ Realizar diversos estudios con el objetivo de obtener métricas y patrones en función de evaluar el impacto de una solicitud de cambio en el sistema.
- ✓ Incorporar al sistema mecanismos necesarios para la visualización gráfica de los procesos estadísticos.

## Referencias bibliográficas

1. Adictos al trabajo. [En línea] [Citado el: 20 de 5 de 2009.]  
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>.
2. aplicaciones empresariales. *Software y Soluciones Tecnológicas Open Source para las PYME*. [En línea] [Citado el: 20 de 05 de 2009.]  
<http://www.aplicacionesempresariales.com/agata-report-tu-base-de-datos-al-descubierto.html>.
3. **Reynoso, Carlos y Kiccillof, Nicolás**. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] [Citado el: 04 de 02 de 2009.]  
<http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/estiloypatron.pdf>.
4. —. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] [Citado el: 06 de 02 de 2009.] <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/estiloypatron.pdf>.
5. —. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] [Citado el: 10 de 02 de 2009.] <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/estiloypatron.pdf>.
6. aulambra.com. [En línea] [Citado el: 18 de 02 de 2009.]  
<http://www.aulambra.com/ver2.asp?id=20&tipo=>.
7. The Perl Directory at Perl.org. [En línea] [Citado el: 30 de 05 de 2009.] <http://www.perl.org/>.
8. PRADO Componet Framework for PHP 5. *PRADO Group*. [En línea] [Citado el: 27 de 05 de 2009.] <http://www.pradosoft.com/>.
9. Zend Framework. *Zend Technologies Ltd*. [En línea] [Citado el: 27 de 05 de 2009.]  
<http://framework.zend.com>.
10. Symfony. *Symfony*. [En línea] [Citado el: 27 de 05 de 2009.] [http://www.symfony-project.org/jobeeet/1\\_2/Doctrine/es/24](http://www.symfony-project.org/jobeeet/1_2/Doctrine/es/24).
11. [En línea]  
[http://www.igac.gov.co:8080/igac\\_web/UserFiles/File/ciaf/TutorialSIG\\_2005\\_26\\_02/paginas/ctr\\_sistemasdegestiondebasededatos.htm](http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistemasdegestiondebasededatos.htm).
12. **Group, PostgreSQL Global Development**. PostgreSQL: The world's most advanced open source database. [En línea] [Citado el: 09 de 05 de 2009.] <http://www.postgresql.org>.
13. PgAdmin PostgreSQL Tools. [En línea] [Citado el: 17 de 03 de 2009.]  
<http://www.pgadmin.org/index.php>.
14. Eclipse Foundation. [En línea] [Citado el: 04 de 03 de 2009.] <http://www.eclipse.org/pdt/>.



15. SISTEMA DE SOPORTE AME. [En línea] [Citado el: 04 de 03 de 2009.] <http://ame.endesa.es/confluence/display/AMEDev/Subclipse>.
16. Eclipse Foundation. [En línea] [Citado el: 04 de 03 de 2009.] <http://www.eclipse.org/mylyn/>.
17. Departamento de Informática IES Gonzalo Nazareno. [En línea] [Citado el: 04 de 03 de 2009.] <http://informatica.gonzalonazareno.org/file.php/8/case.pdf>.
18. Grupo Soluciones Innova. *Grupo Soluciones Innova*. [En línea] [Citado el: 16 de 02 de 2009.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
19. **Orallo, Enrique Hernández**. Departament d'Informàtica de Sistemes i Computadors . [En línea] [Citado el: 18 de 02 de 2009.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
20. [En línea] [Citado el: 04 de 02 de 2009.] <http://www.extremeprogramming.org/what.html>.
21. EKA, La Revista Empresarial. [En línea] [Citado el: 04 de 02 de 2009.] [http://www.ekaenlinea.com/index.php?option=com\\_content&task=view&id=3518&Itemid=66](http://www.ekaenlinea.com/index.php?option=com_content&task=view&id=3518&Itemid=66).
22. **Schwaber K., Beedle M., Martin R.C.** *baufest*. [En línea] [Citado el: 04 de 02 de 2009.] [http://www.baufest.com/spanish/scrum/scrumconference2006/Que\\_es\\_scrum.pdf](http://www.baufest.com/spanish/scrum/scrumconference2006/Que_es_scrum.pdf).
23. **Fowler, Martin, y otros.** *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. 0-321-12742-0.
24. **Gamma, Erich, y otros.** *Design Patterns: Elements of Reusable Object-Oriented Software*. LONDON : ADDISON-WESLEY, 1994. 9780201633610.
25. **Ltd., Sparx Systems Pty.** Guía de Usuario de Enterprise Architect 7.0. [En línea] [Citado el: 30 de 05 de 2009.] <http://www.sparxsystems.com.ar/download/ayuda/index.html?componentdiagram.htm>.

## **Bibliografía**

1. **Powell, Thomas and Schneider, Fritz.** *JavaScript 2.0: The Complete Reference, Second Edition.* s.l. : McGraw-Hill/Osborne , 2004. 0072253576.
2. **Quigley, Ellie.** *JavaScript™ by Example.* s.l. : Prentice Hall , 2003. 0-13-140162-9.
3. **Fowler, Martin, et al.** *Patterns of Enterprise Application Architecture.* s.l. : Addison Wesley, 2002. 0-321-12742-0.
4. **Clements, Paul, et al.** *Documenting Software Architectures: Views and Beyond.* s.l. : Addison Wesley, 2002. 0-201-70372-6.

Anexos

Anexo 1 Diagramas de Actividades

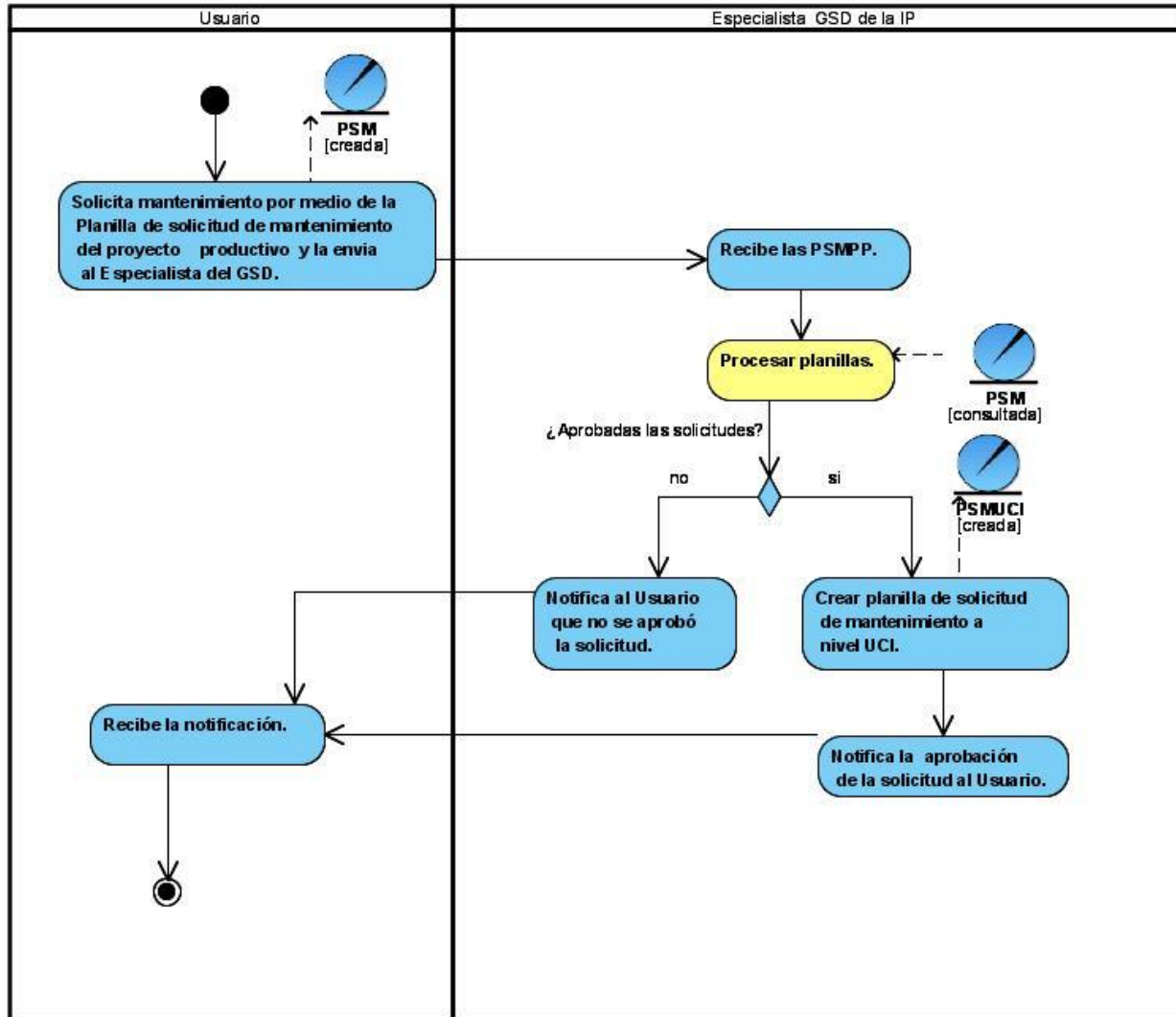


Figura 2.2 Diagrama de Actividades.

**Anexo 2 Diagramas de clases del análisis**

**Caso de uso Autenticar Usuario**

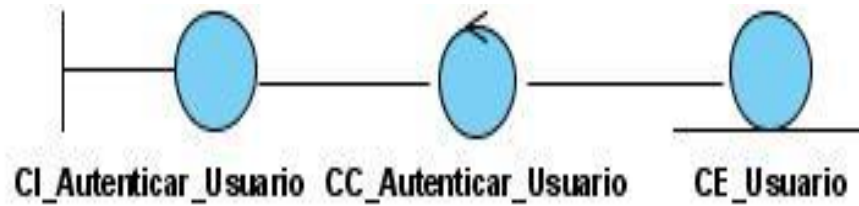


Figura 3.1 Diagrama de Caso de Uso Autenticar Usuario.

**Caso de uso Gestionar Usuario**

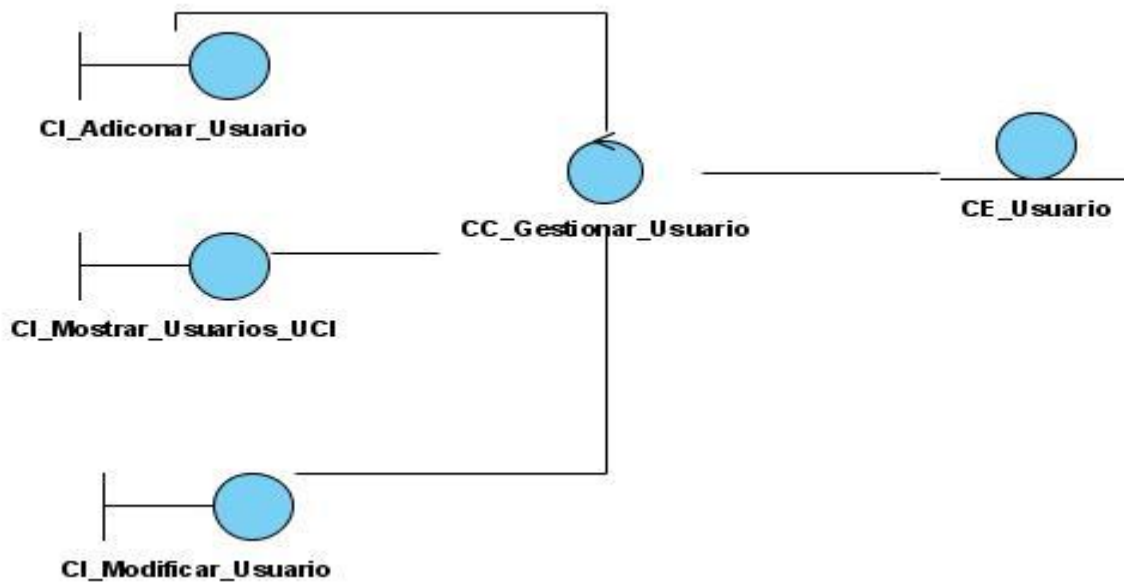


Figura 3.2 Diagrama de Caso de Uso Gestionar Usuario.

**Caso de uso Gestionar Docente**

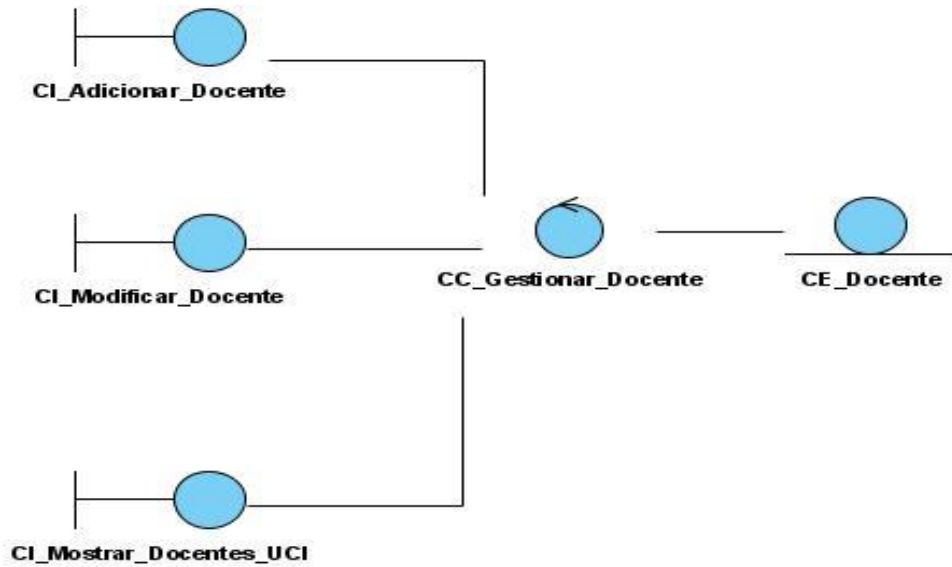


Figura 3.3 Diagrama de Caso de Uso Gestionar Docente.

**Caso de uso Gestionar Facultad**

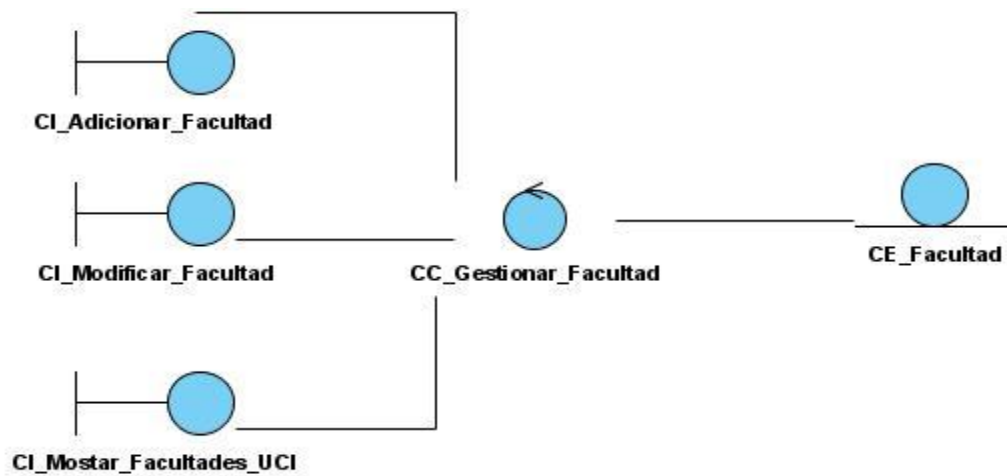


Figura 3.4 Diagrama de Caso de Uso Gestionar Facultad.

### Caso de Uso Gestionar Proyecto Productivo

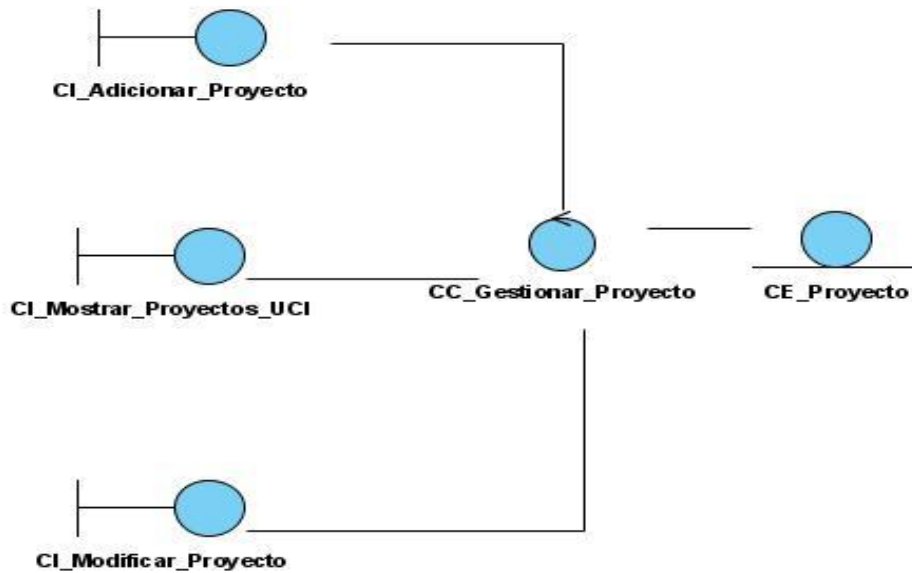


Figura 3.5 Diagrama de Caso de Uso Gestionar Proyecto Productivo.

### Caso de Uso Gestionar Laboratorio

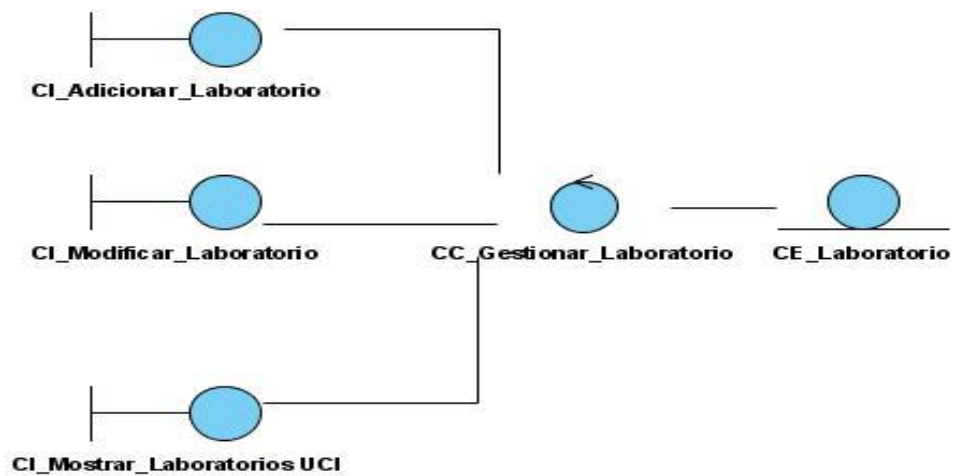


Figura 3.6 Diagrama de Caso de Uso Gestionar Laboratorio.



### Caso de Uso Gestionar Rotura

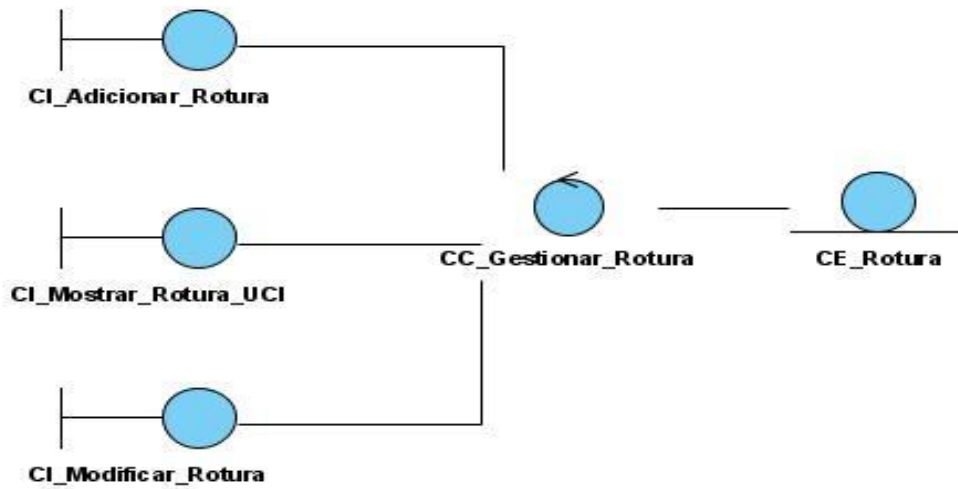


Figura 3.7 Diagrama de Caso de Uso Gestionar Rotura.

### Caso de Uso Gestionar Reporte

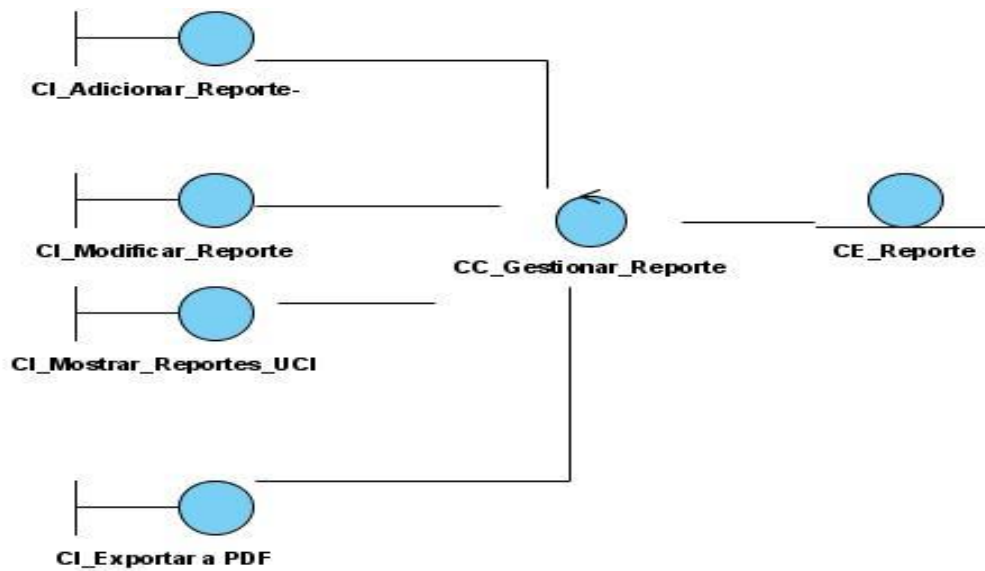


Figura 3.8 Diagrama de Caso de Uso Gestionar Reporte.





**Anexo 3 Diagramas de Colaboración**

**Caso de uso Autenticar Usuario**

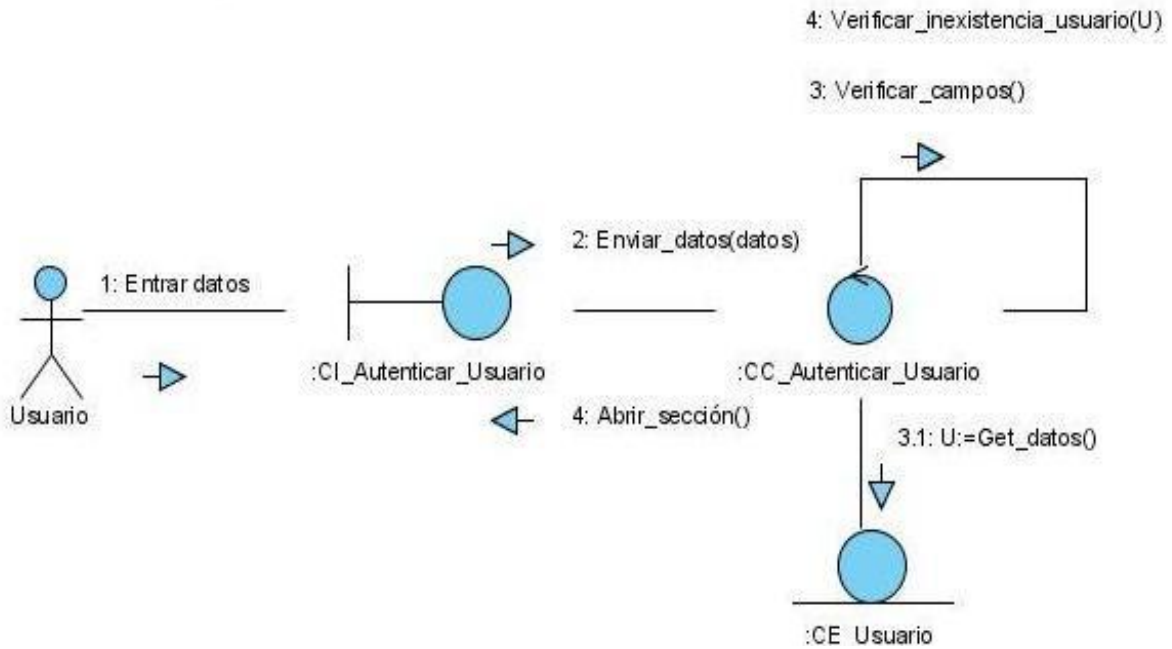


Figura 3.9 Diagrama de Caso de Uso Autenticar Usuario.

**Caso de uso Gestionar Usuario: Escenario Adicionar Usuario**

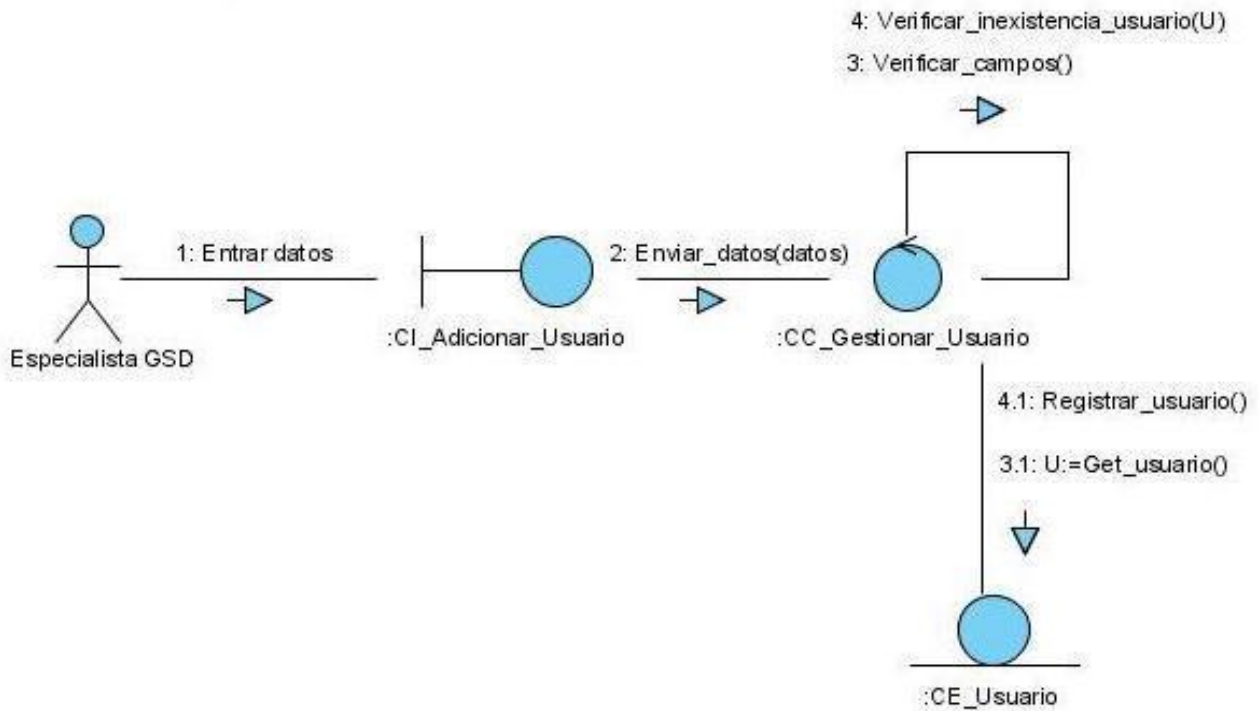


Figura 3.10 Diagrama de Caso de Uso Gestionar Usuario: Escenario Adicionar Usuario.

**Caso de uso Gestionar Usuario: Escenario Modificar Usuario**

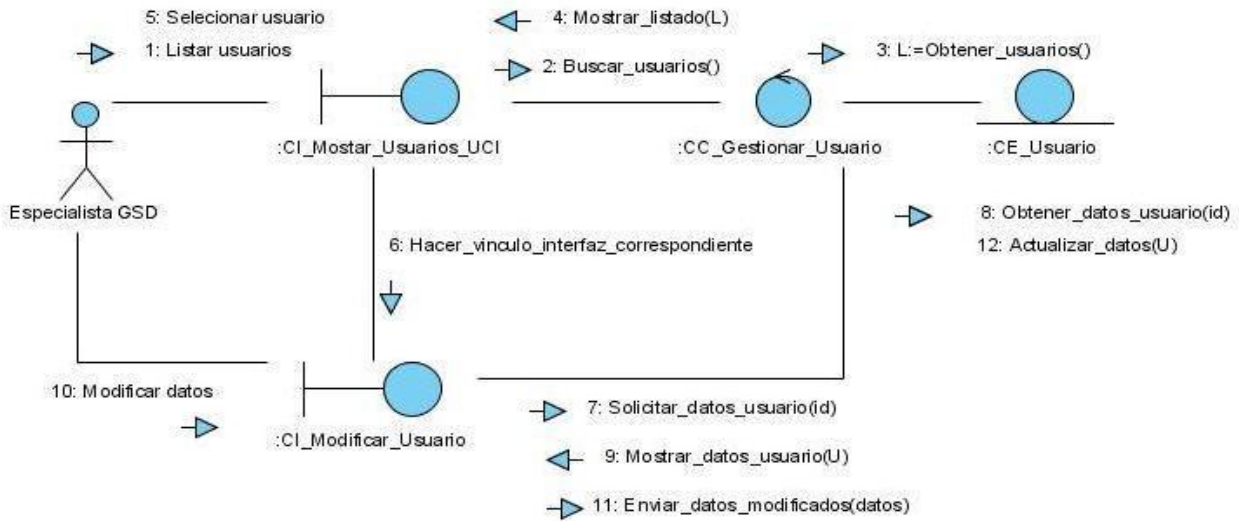


Figura 3.11 Diagrama de Caso de Uso Gestionar Usuario: Escenario Modificar Usuario.

**Caso de uso Gestionar Usuario: Escenario Eliminar Usuario**

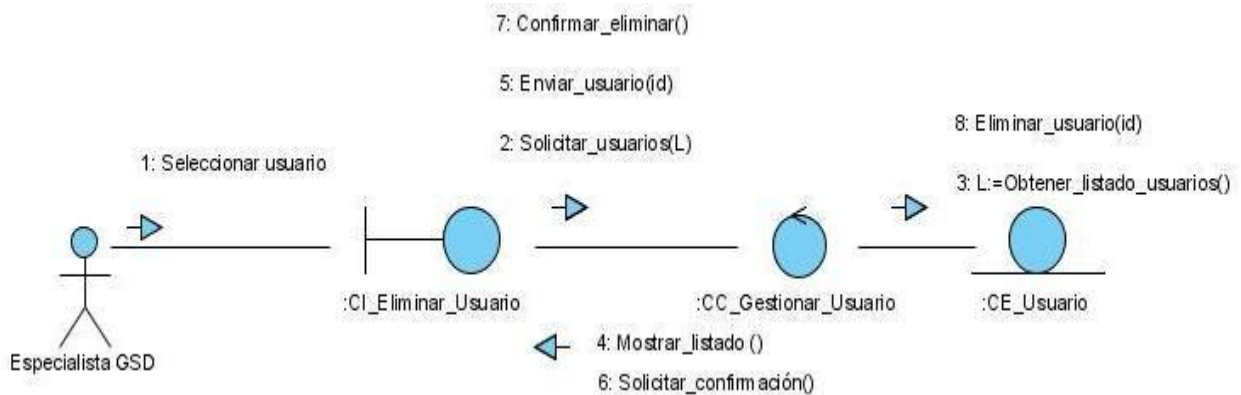


Figura 3.12 Diagrama de Caso de Uso Gestionar Usuario: Escenario Eliminar Usuario.

**Caso de uso Gestionar Docente: Escenario Adicionar Docente**

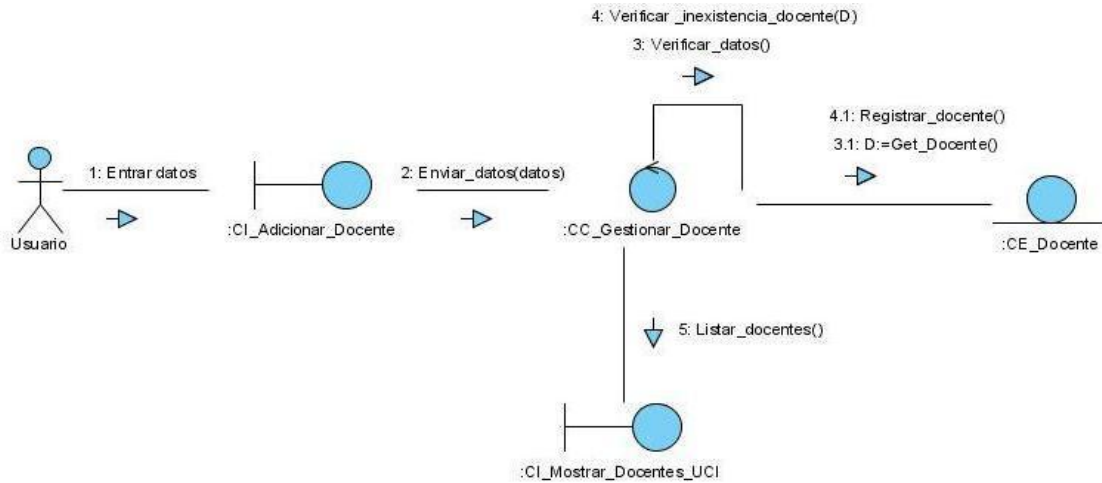


Figura 3.13 Diagrama de Caso de Uso Gestionar Docente: Escenario Adicionar Docente.

**Caso de uso Gestionar Docente: Escenario Modificar Docente**

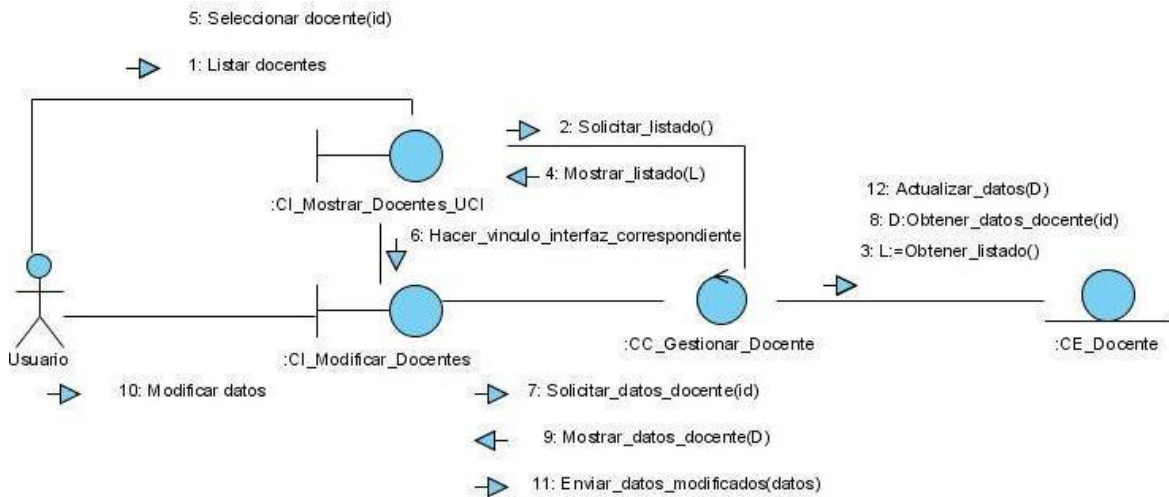


Figura 3.14 Diagrama de Caso de Uso Gestionar Docente: Escenario Modificar Docente.

**Caso de uso Gestionar Docente: Escenario Eliminar Docente**

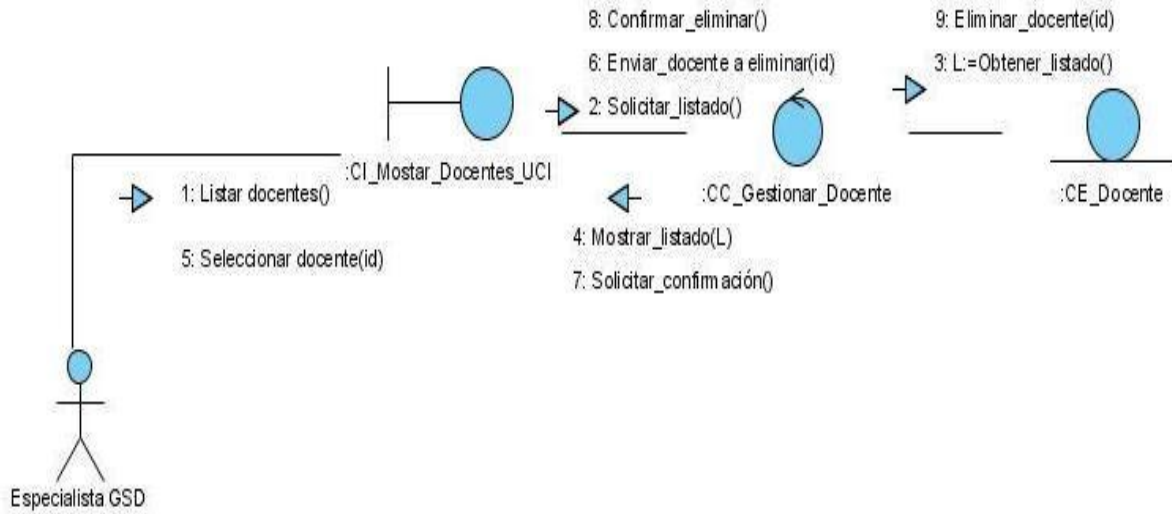


Figura 3.15 Diagrama de Caso de Uso Gestionar Docente: Escenario Eliminar Docente.

**Caso de uso Gestionar Docente: Escenario Mostrar Docente**

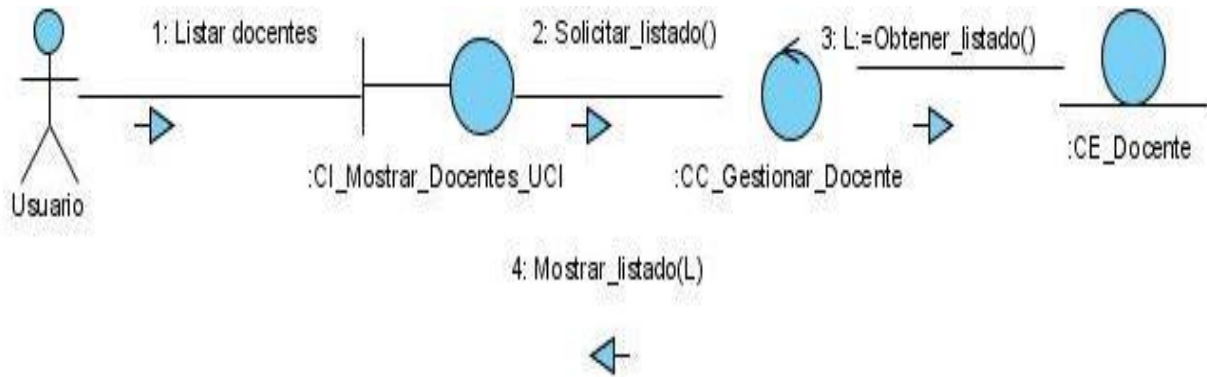


Figura 3.16 Diagrama de Caso de Uso Gestionar Docente: Escenario Mostrar Docente.

**Caso de uso Gestionar Facultad: Escenario Adicionar Facultad**

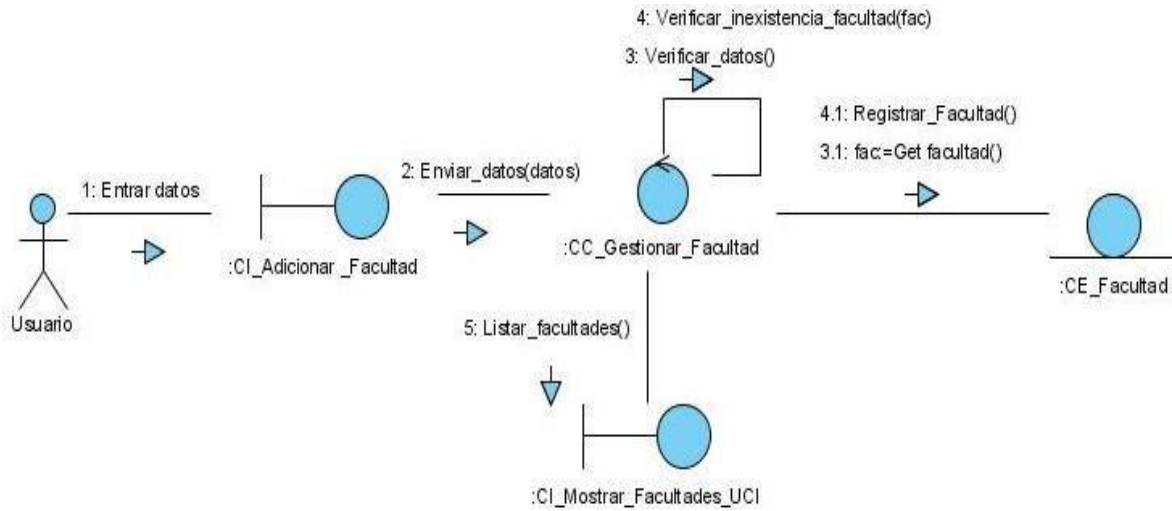


Figura 3.17 Diagrama de Caso de Uso Gestionar Facultad: Escenario Adicionar Facultad.

**Caso de uso Gestionar Facultad: Escenario Modificar Facultad**

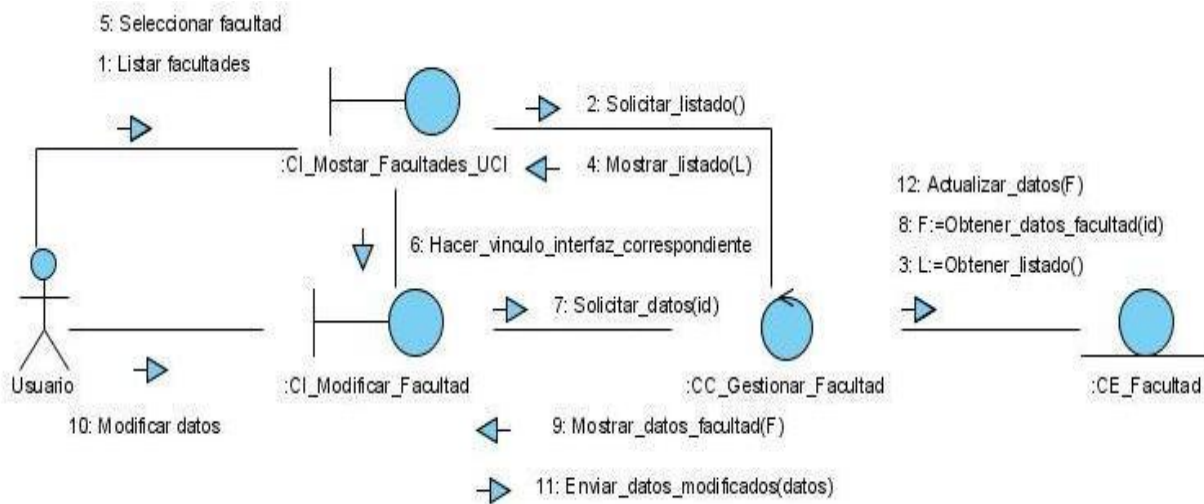


Figura 3.17 Diagrama de Caso de Uso Gestionar Facultad: Escenario Modificar Facultad.

**Caso de uso Gestionar Facultad: Escenario Eliminar Facultad**

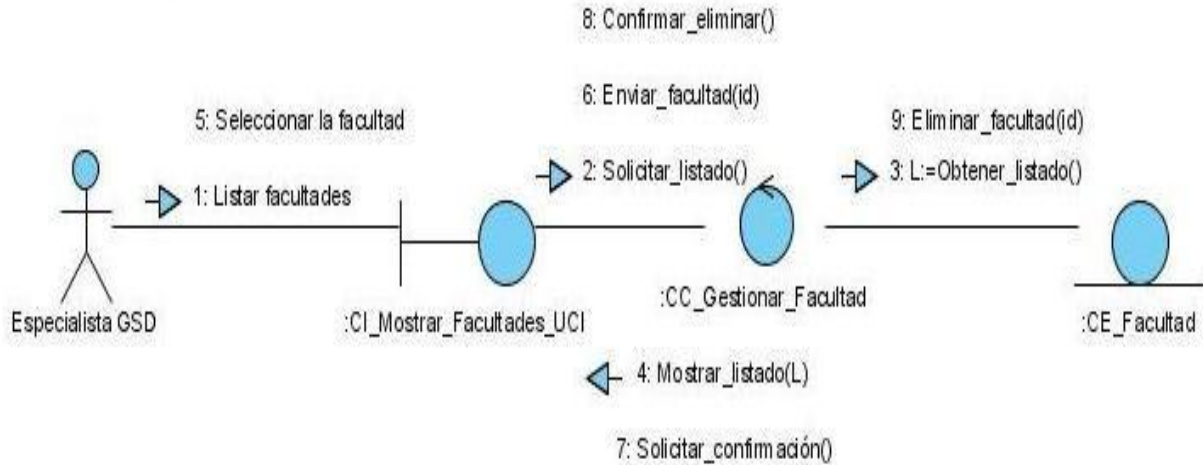


Figura 3.18 Diagrama de Caso de Uso Gestionar Facultad: Escenario Eliminar Facultad.

**Caso de uso Gestionar Facultad: Escenario Mostrar Facultad**

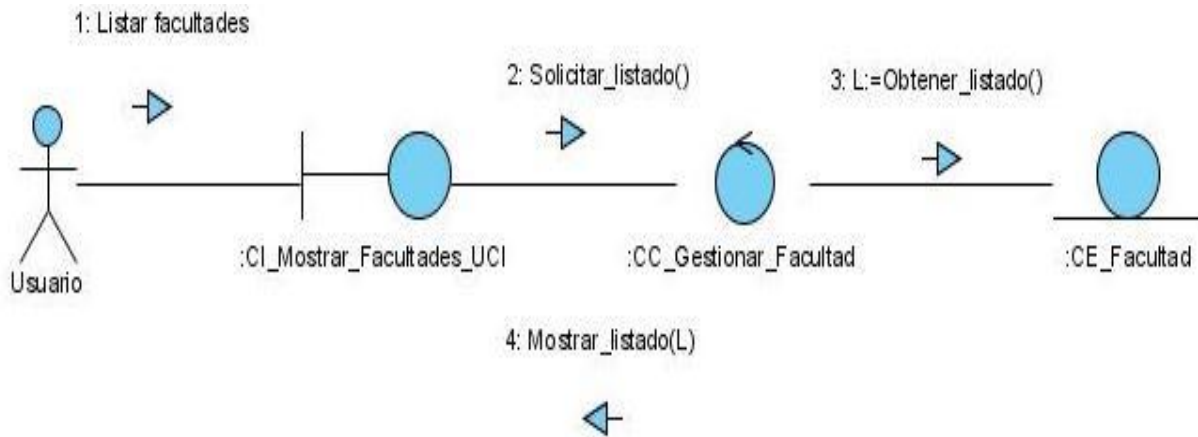


Figura 3.19 Diagrama de Caso de Uso Gestionar Facultad: Escenario Mostrar Facultad.



**Caso de uso Gestionar Proyecto: Escenario Adicionar Proyecto**

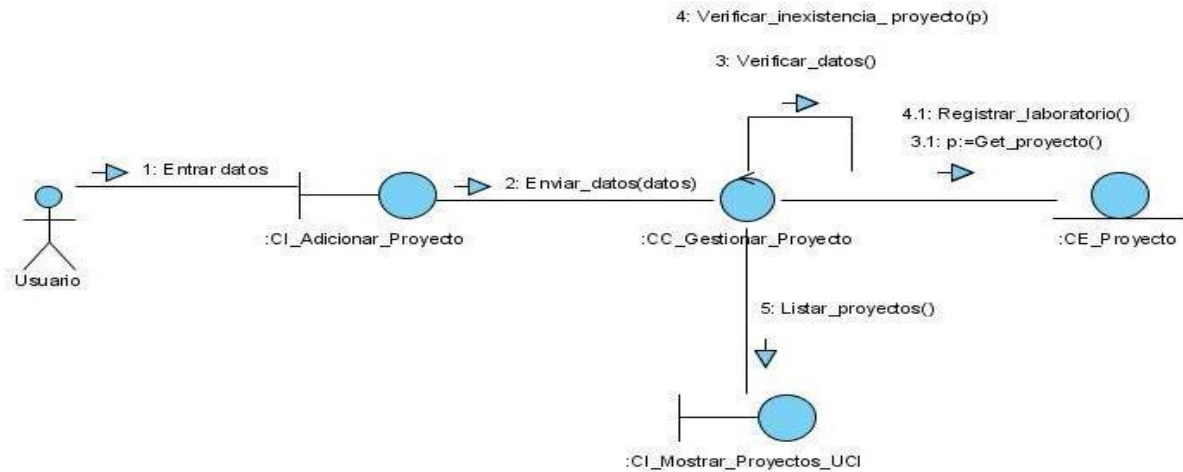


Figura 3.20 Diagrama de Caso de Uso Gestionar Proyecto: Escenario Adicionar Proyecto.

**Caso de uso Gestionar Proyecto: Escenario Modificar Proyecto**

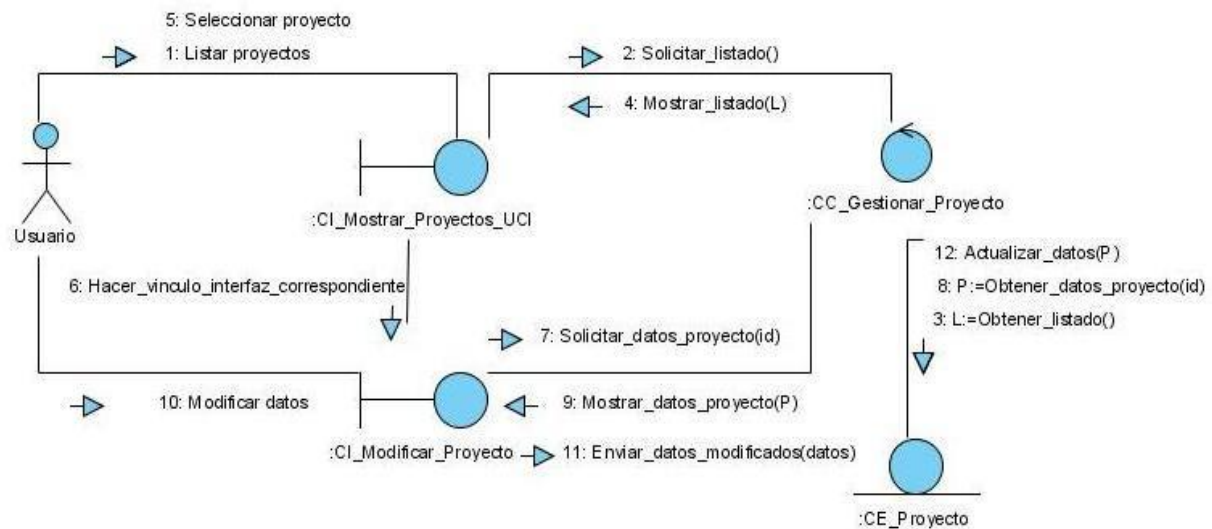


Figura 3.21 Diagrama de Caso de Uso Gestionar Proyecto: Escenario Modificar Proyecto.

**Caso de uso Gestionar Proyecto: Escenario Eliminar Proyecto**

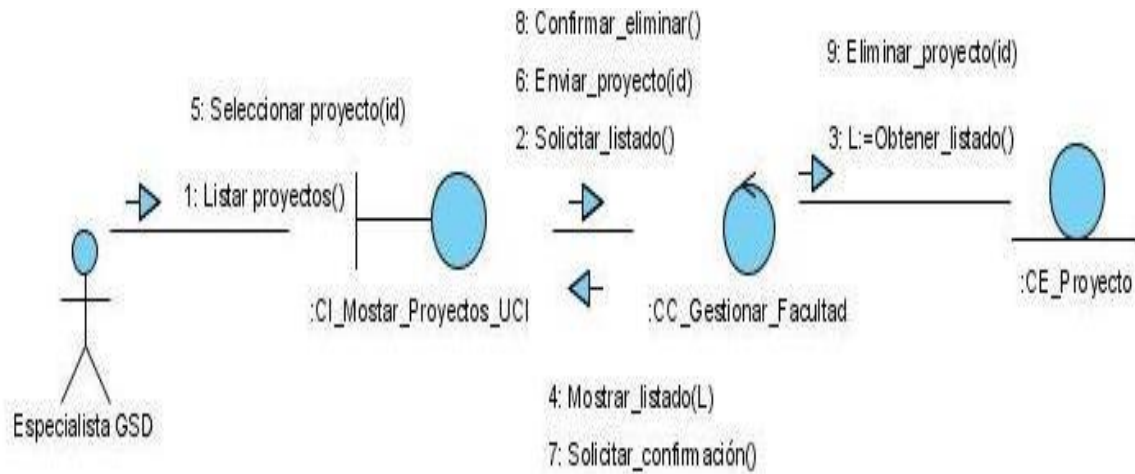


Figura 3.22 Diagrama de Caso de Uso Gestionar Proyecto: Escenario Eliminar Proyecto.

**Caso de uso Gestionar Proyecto: Escenario Mostrar Proyecto**

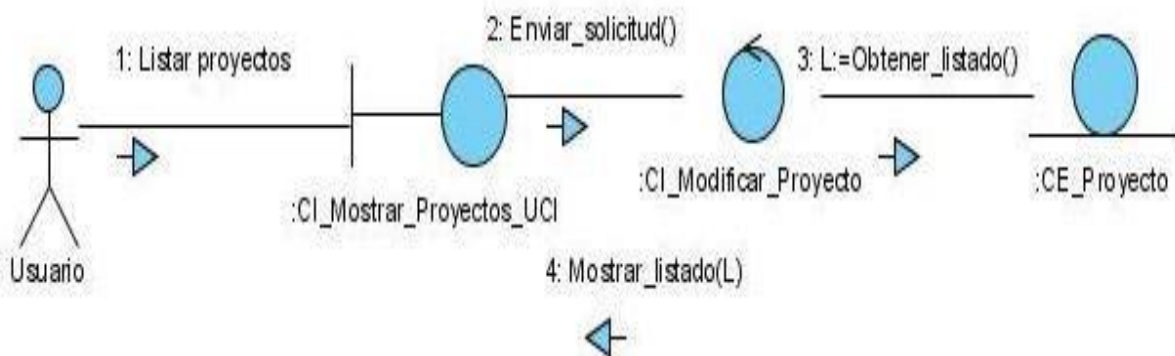


Figura 3.23 Diagrama de Caso de Uso Gestionar Proyecto: Escenario Mostrar Proyecto.

**Caso de uso Gestionar Laboratorio: Escenario Adicionar Laboratorio**

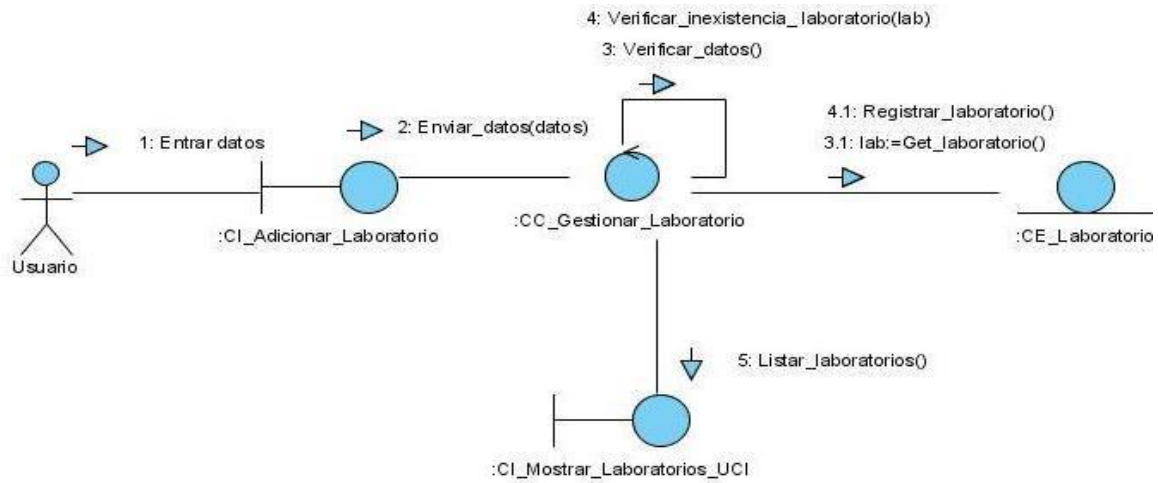


Figura 3.24 Diagrama de Caso de Uso Gestionar Laboratorio: Escenario Adicionar Laboratorio.

**Caso de uso Gestionar Laboratorio: Escenario Modificar Laboratorio**

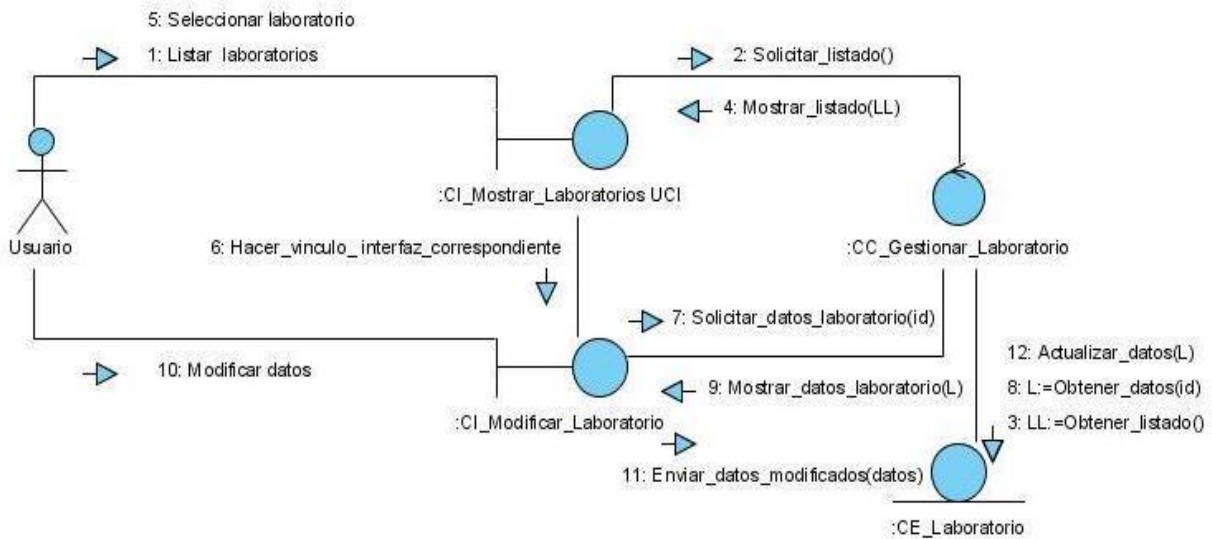


Figura 3.25 Diagrama de Caso de Uso Gestionar Laboratorio: Escenario Modificar Laboratorio.

**Caso de uso Gestionar Laboratorio: Escenario Eliminar Laboratorio**

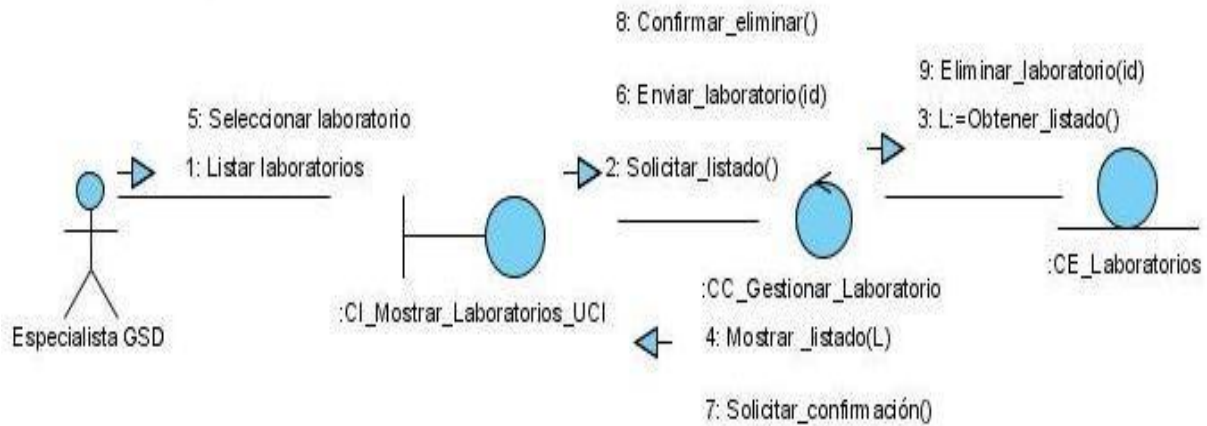


Figura 3.26 Diagrama de Caso de Uso Gestionar Laboratorio: Escenario Eliminar Laboratorio.

**Caso de uso Gestionar Laboratorio: Escenario Mostrar Laboratorio**

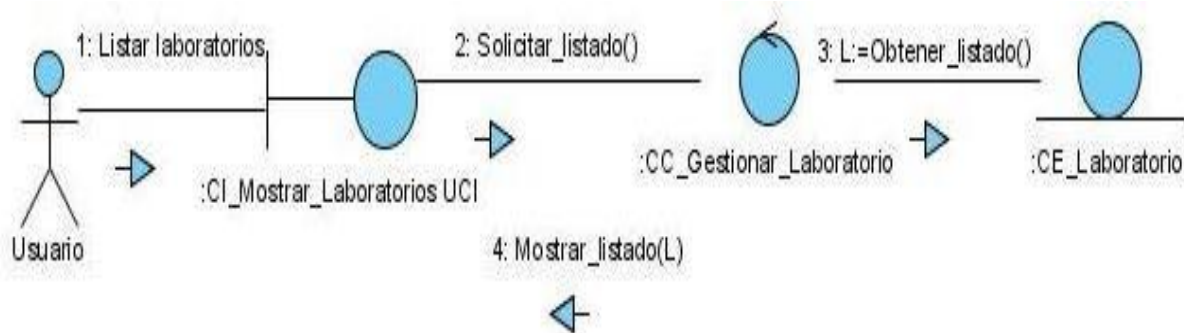


Figura 3.27 Diagrama de Caso de Uso Gestionar Laboratorio: Escenario Mostrar Laboratorio.

**Caso de uso Gestionar Rotura: Escenario Adicionar Rotura**

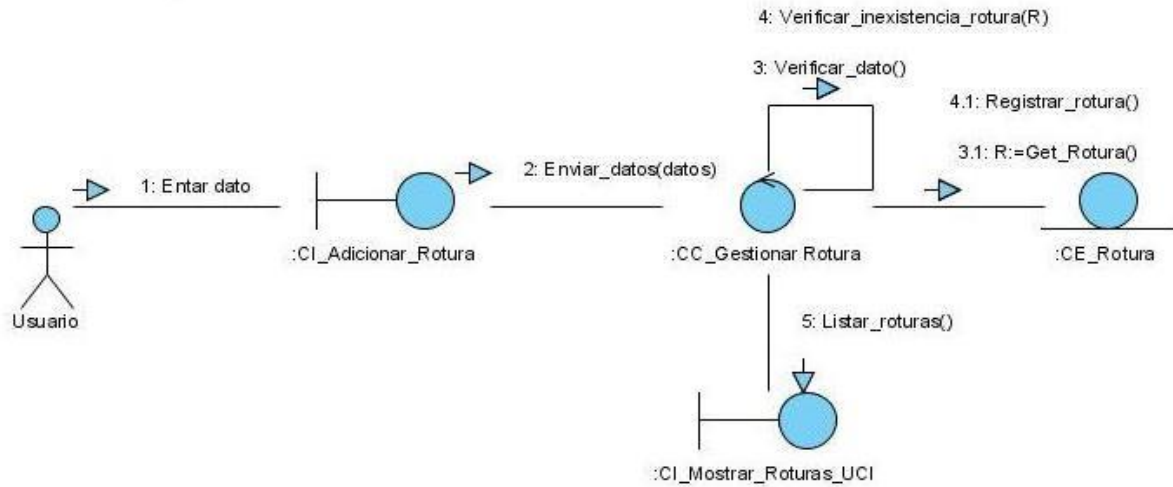


Figura 3.28 Diagrama de Caso de Uso Gestionar Rotura: Escenario Adicionar Rotura.

**Caso de uso Gestionar Rotura: Escenario Modificar Rotura**

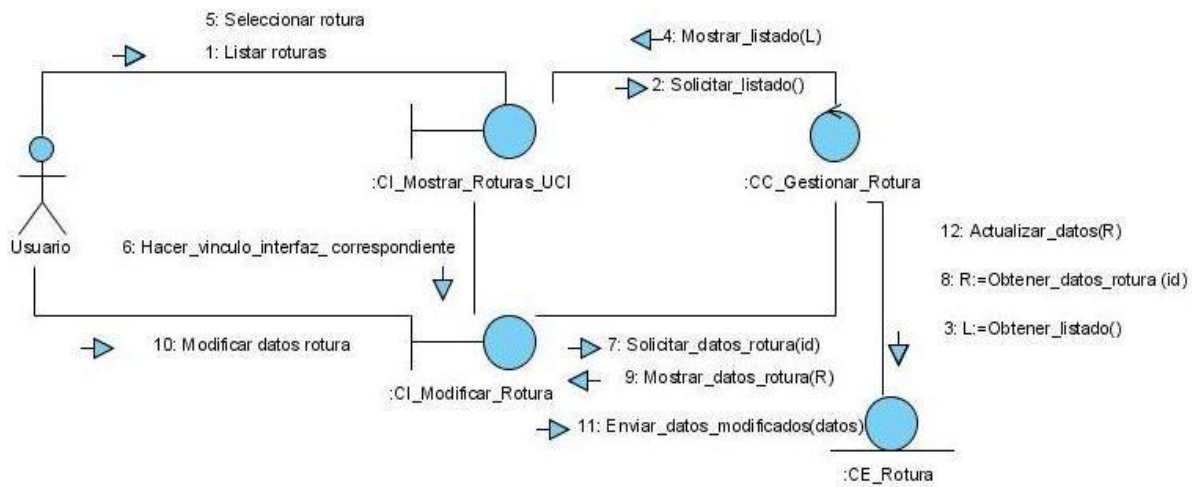


Figura 3.29 Diagrama de Caso de Uso Gestionar Rotura: Escenario Modificar Rotura.

**Caso de uso Gestionar Rotura: Escenario Eliminar Rotura**



Figura 3.30 Diagrama de Caso de Uso Gestionar Rotura: Escenario Eliminar Rotura.

**Caso de uso Gestionar Rotura: Escenario Mostrar Rotura**

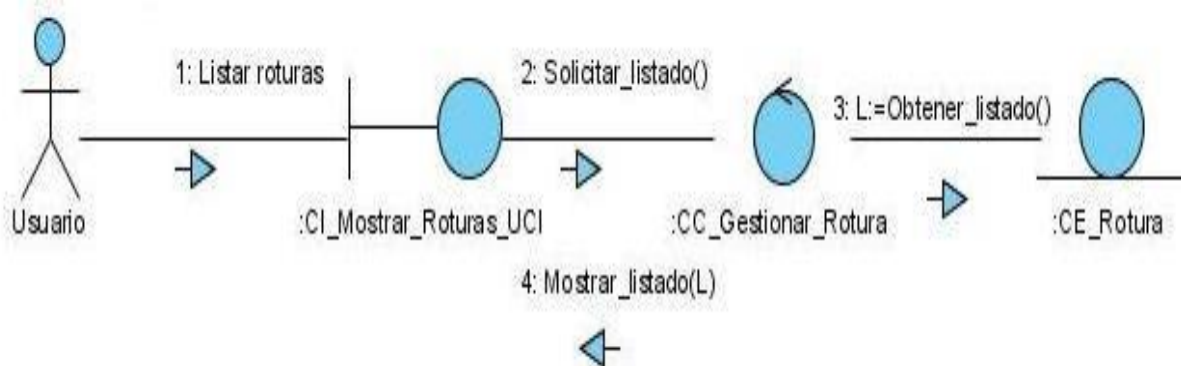


Figura 3.31 Diagrama de Caso de Uso Gestionar Rotura: Escenario Mostrar Rotura.

**Caso de uso Gestionar Reporte: Escenario Adicionar Reporte**

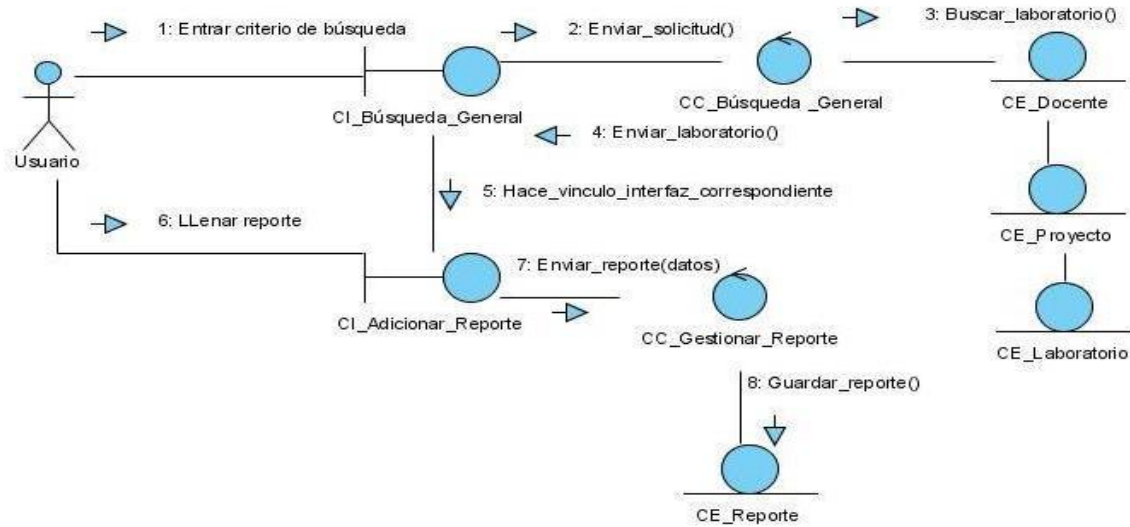


Figura 3.32 Diagrama de Caso de Uso Gestionar Reporte: Escenario Adicionar Reporte.

**Caso de uso Gestionar Reporte: Escenario Modificar Reporte**

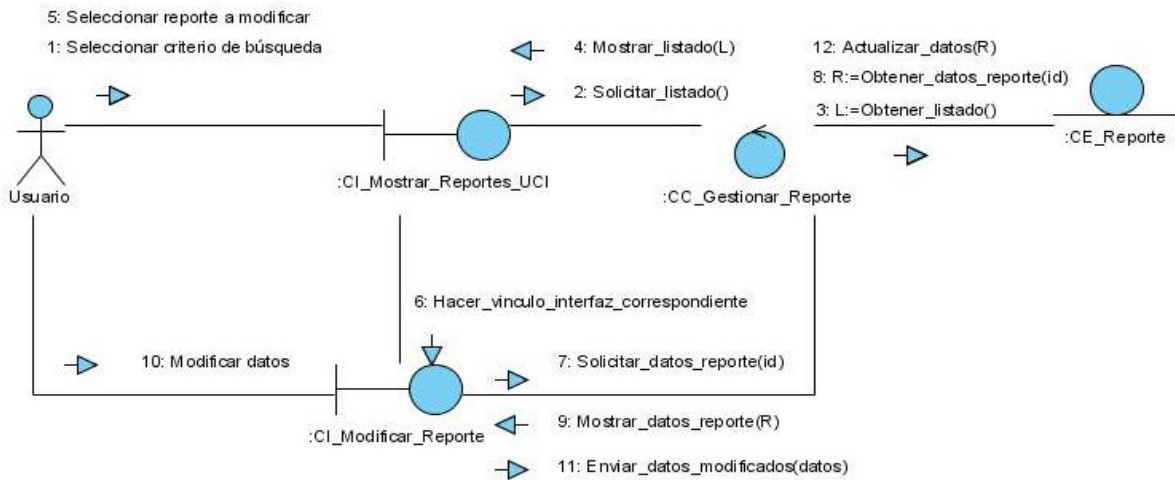


Figura 3.33 Diagrama de Caso de Uso Gestionar Reporte: Escenario Modificar Reporte.

**Caso de uso Gestionar Reporte: Escenario Eliminar Reporte**

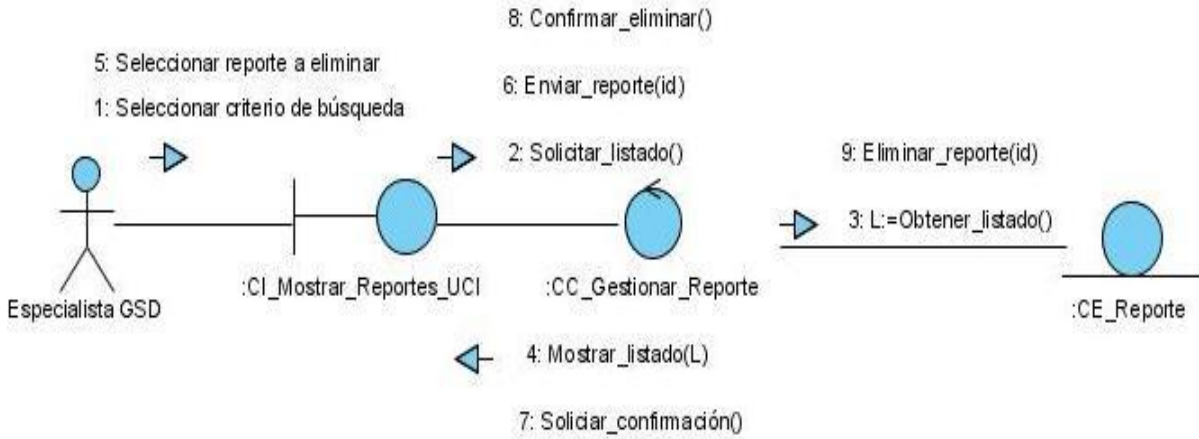


Figura 3.34 Diagrama de Caso de Uso Gestionar Reporte: Escenario Eliminar Reporte.

**Caso de uso Gestionar Reporte: Escenario Mostrar Reporte**

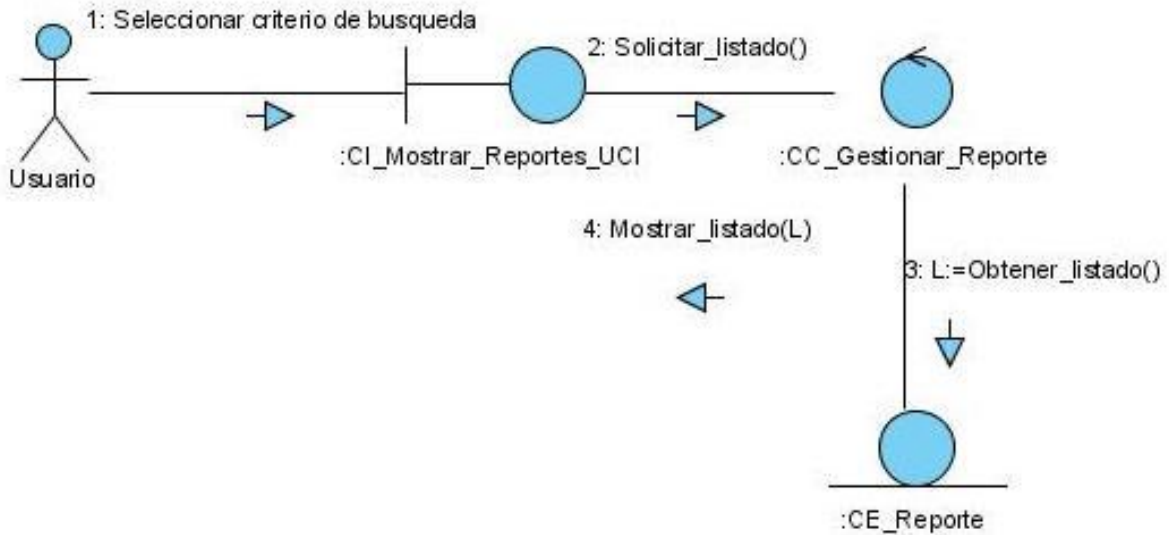


Figura 3.35 Diagrama de Caso de Uso Gestionar Reporte: Escenario Mostrar Reporte.



**Caso de uso Gestionar Reporte: Escenario Exportar Reporte**

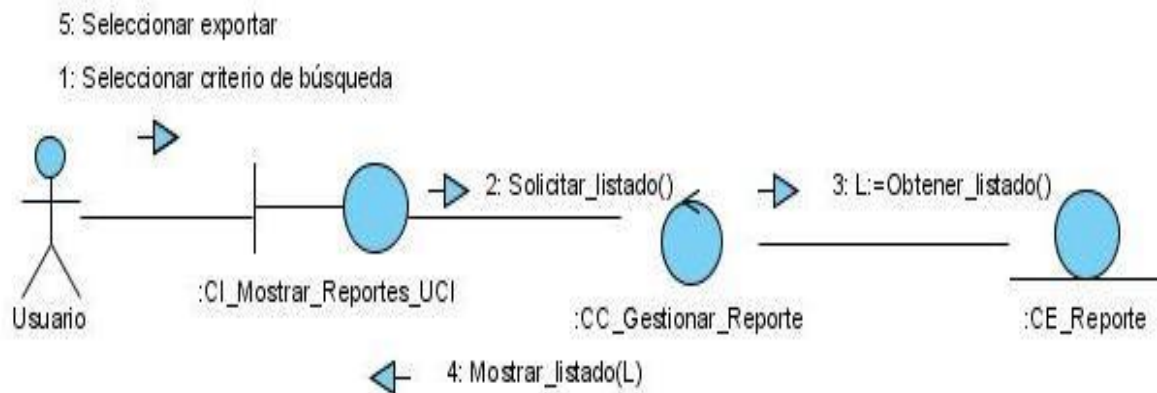


Figura 3.36 Diagrama de Caso de Uso Gestionar Reporte: Escenario Exportar Reporte.

Anexo 4 Diagramas de Clases del Diseño.

Diagrama de Clases del Diseño del Caso de Uso: Autenticar Usuario.

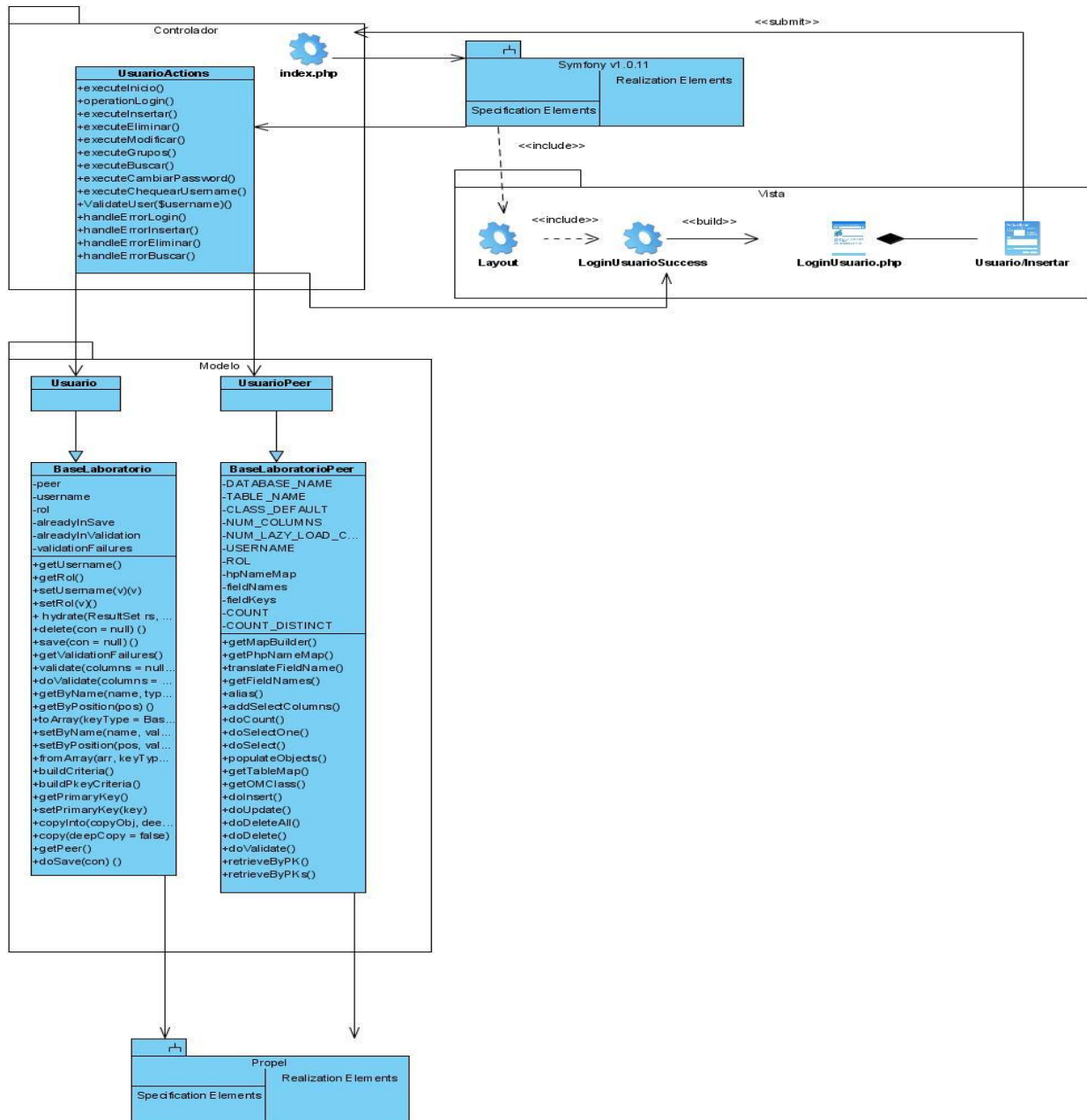


Figura 3.37 Diagrama de Clases del Diseño del Caso de Uso: Autenticar Usuario.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Adicionar

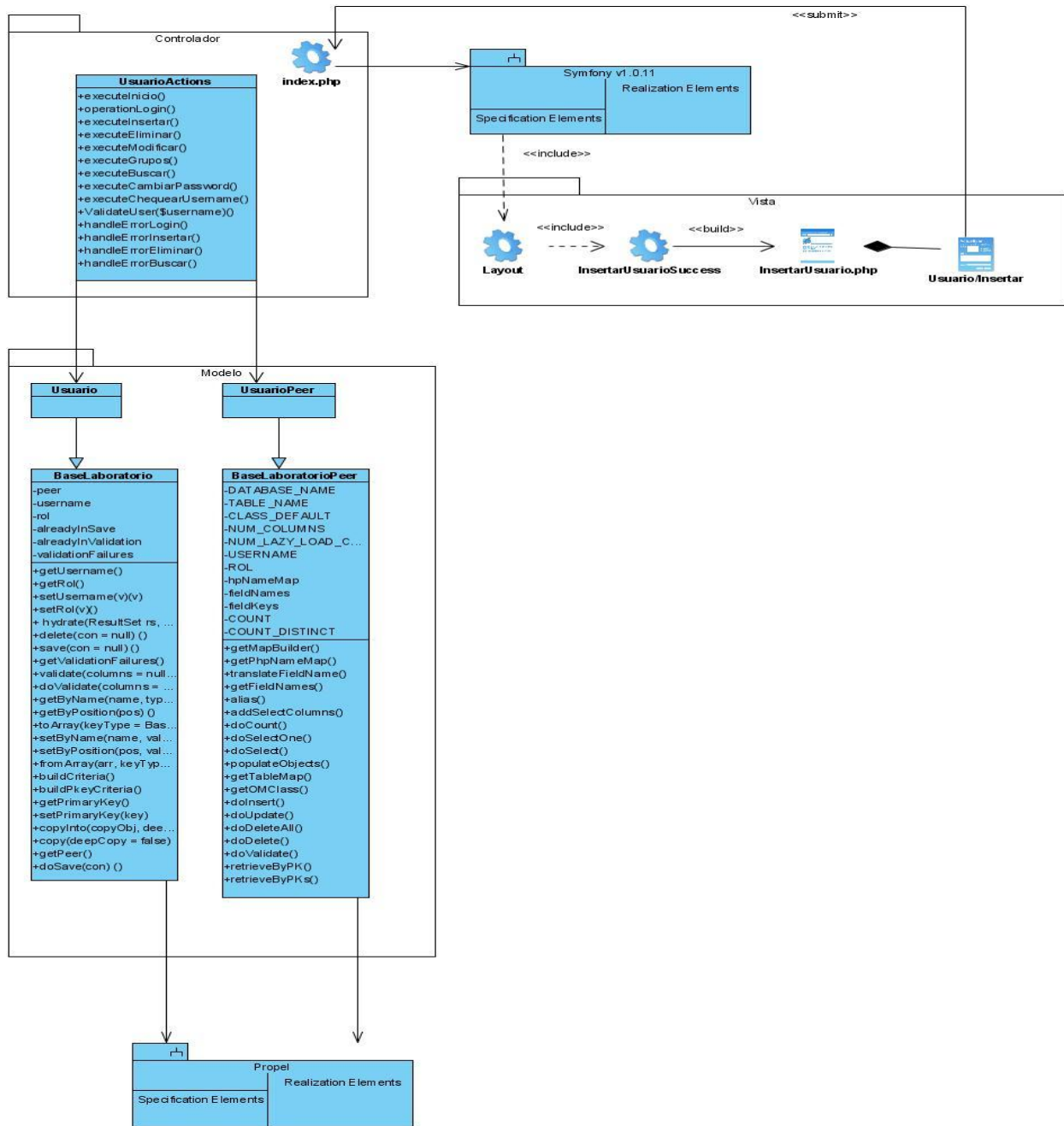


Figura 3.38 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Modificar.

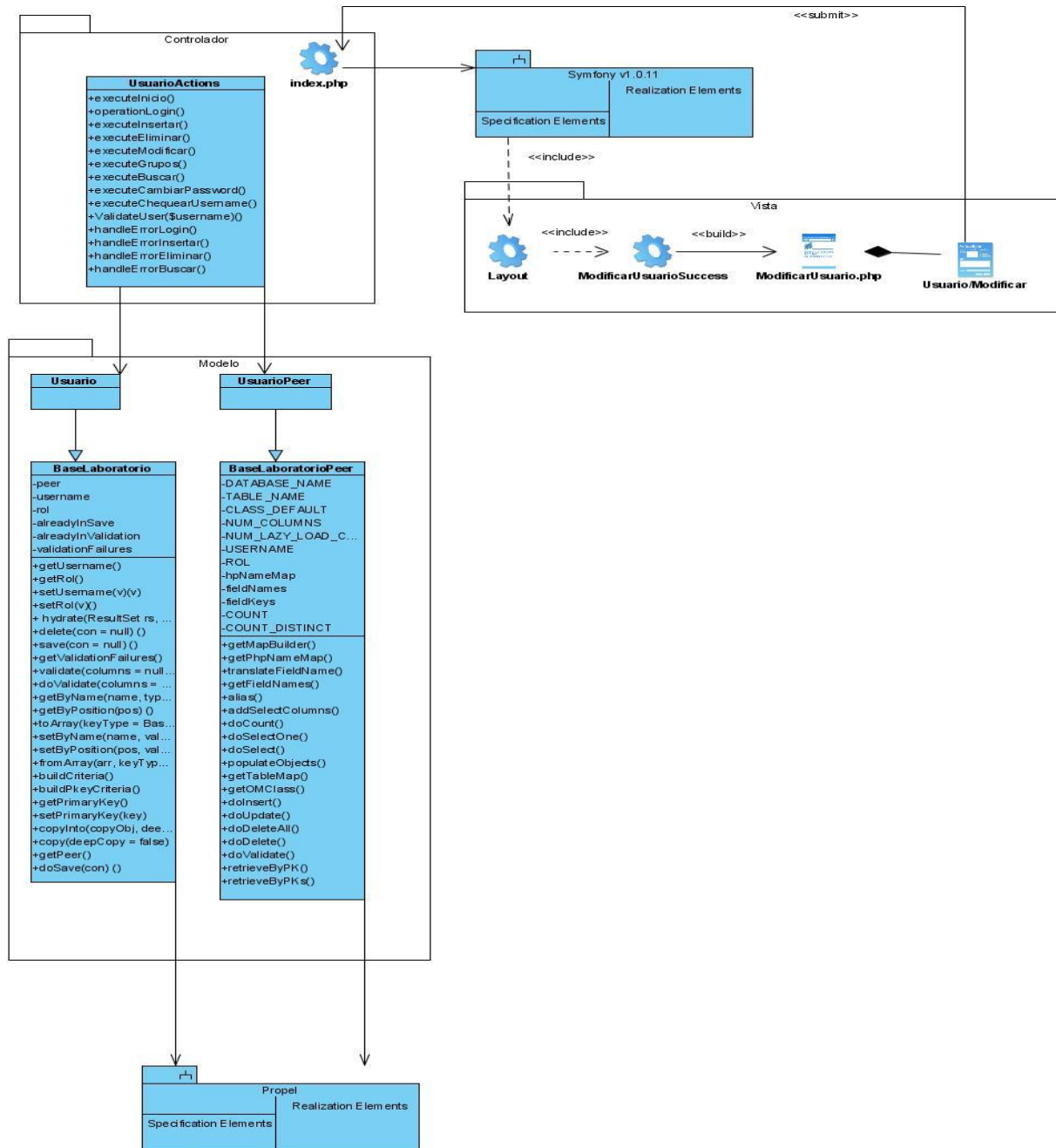


Figura 3.39 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Eliminar.

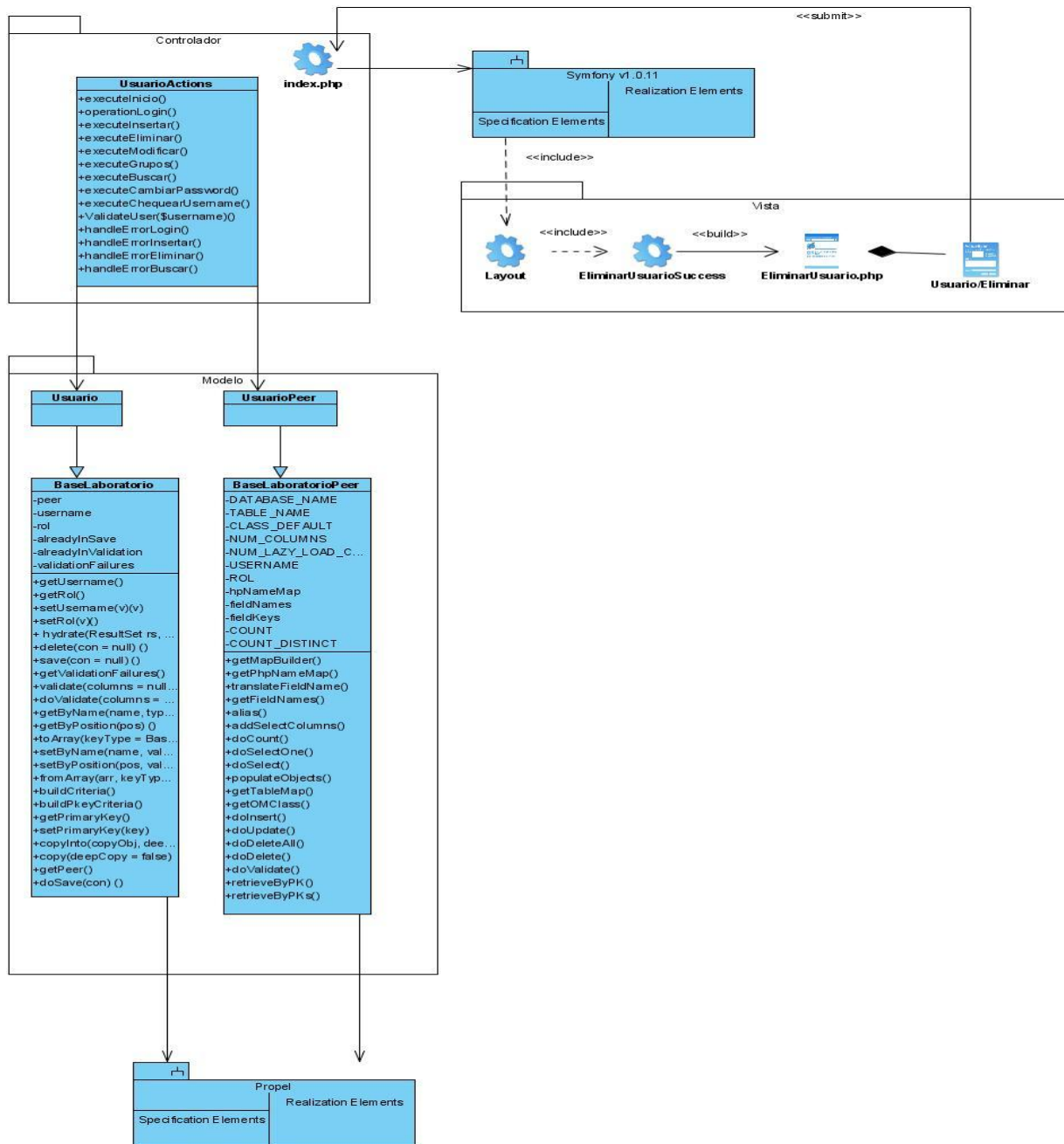


Figura 3.40 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Adicionar.

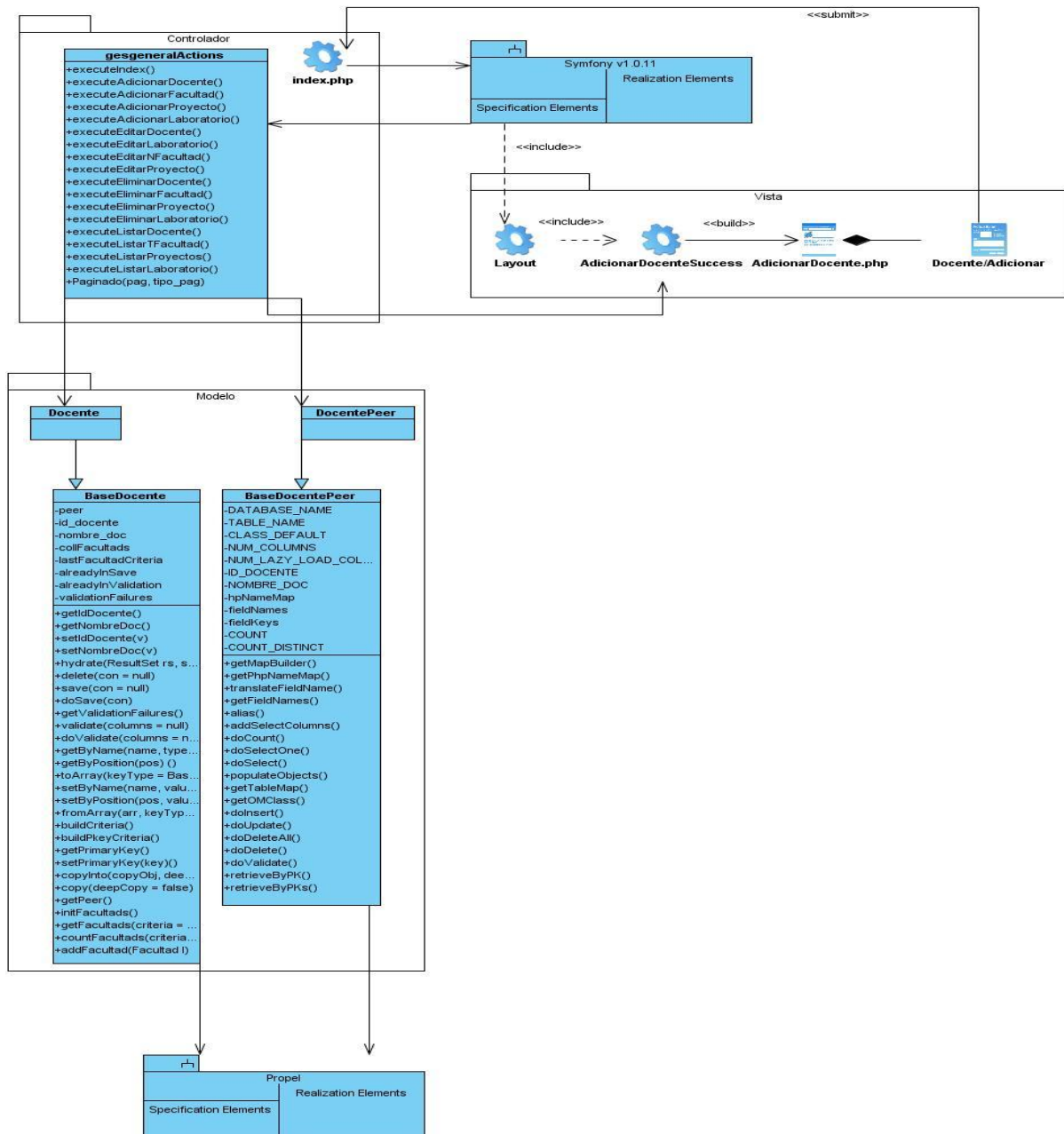


Figura 3.41 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Modificar.

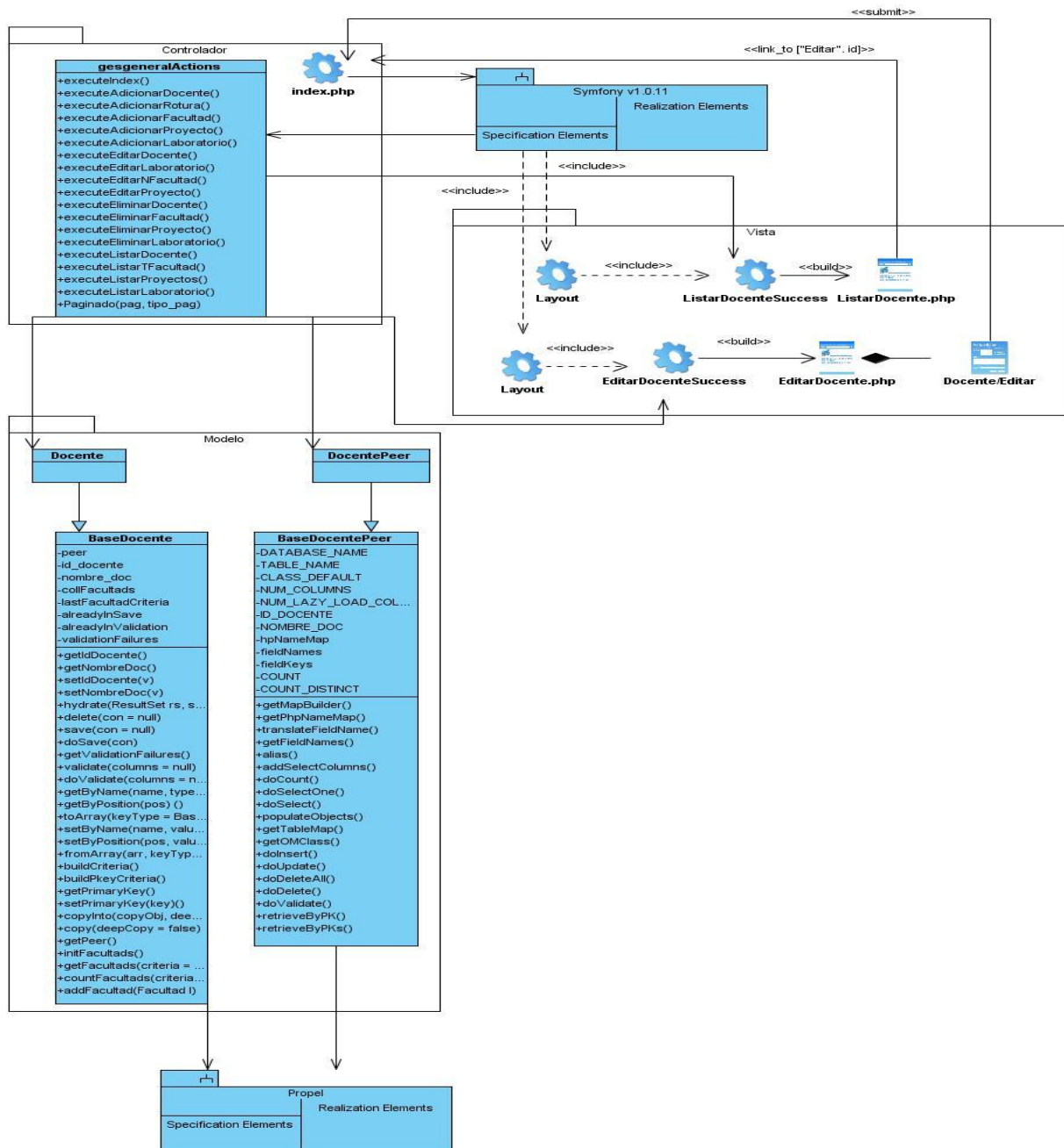


Figura 3.42 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Mostrar.

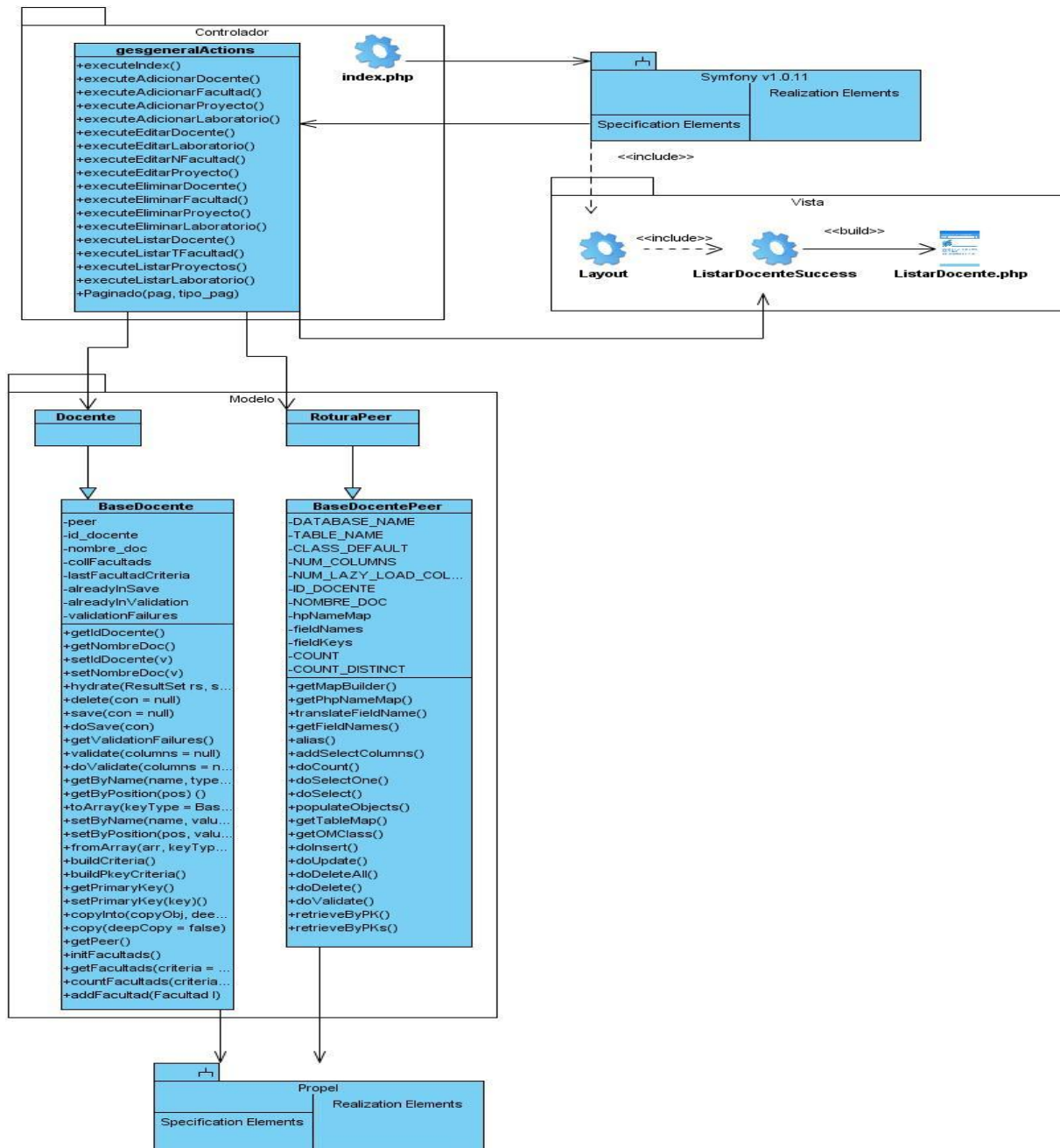


Figura 3.43 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Mostrar.



Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Eliminar.

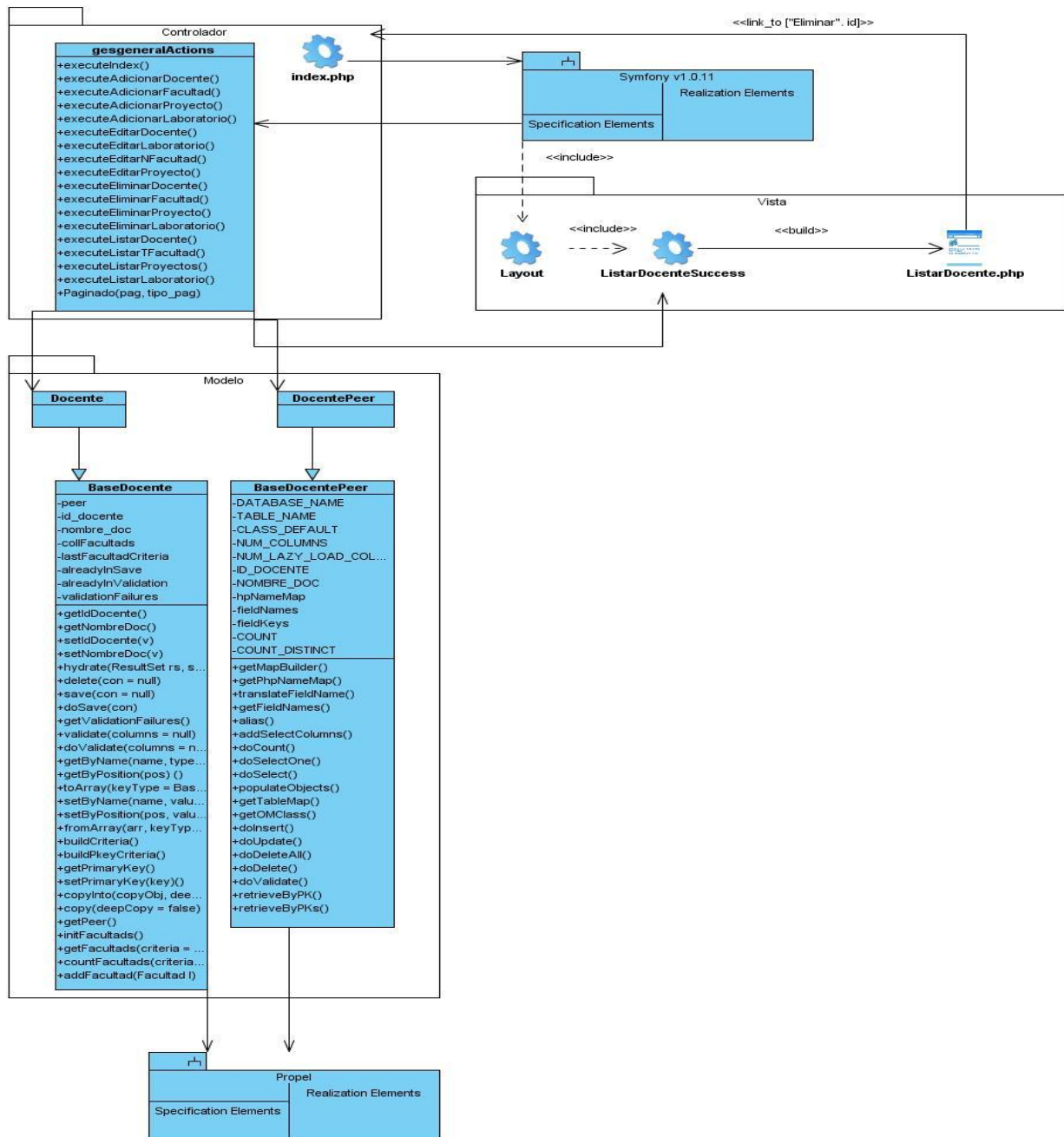


Figura 3.44 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Adicionar.

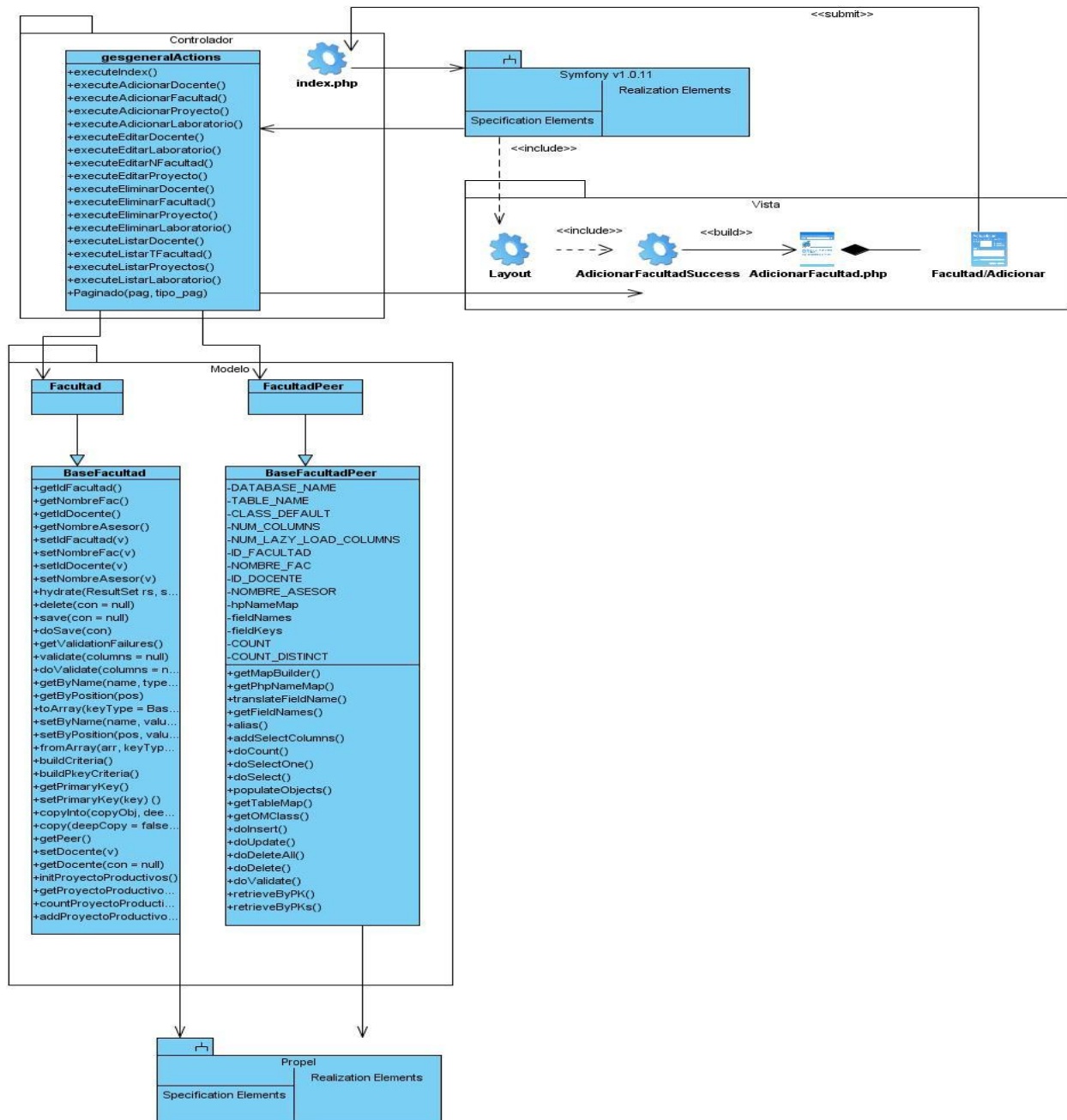


Figura 3.45 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Modificar.

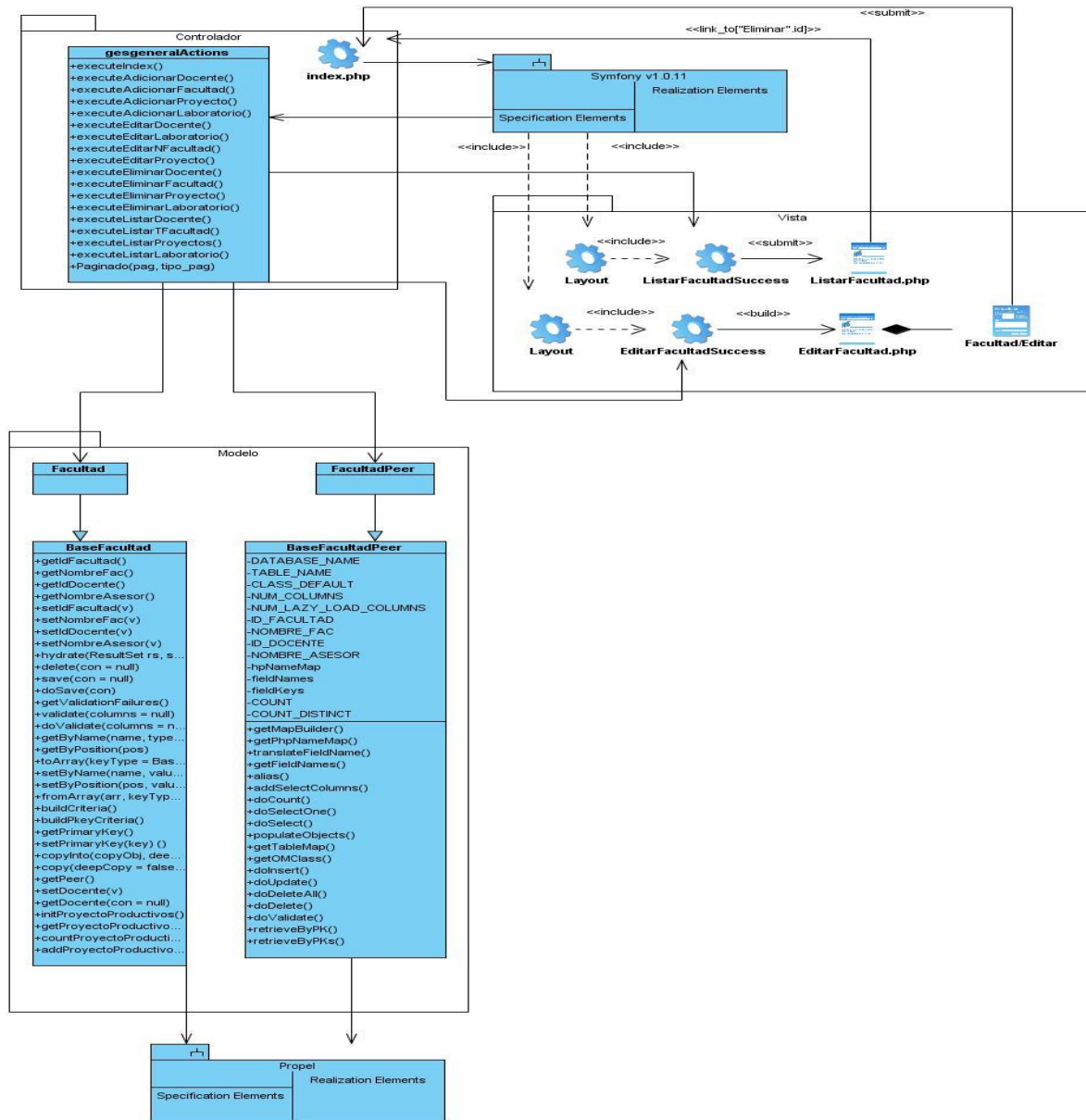


Figura 3.46 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Mostrar.

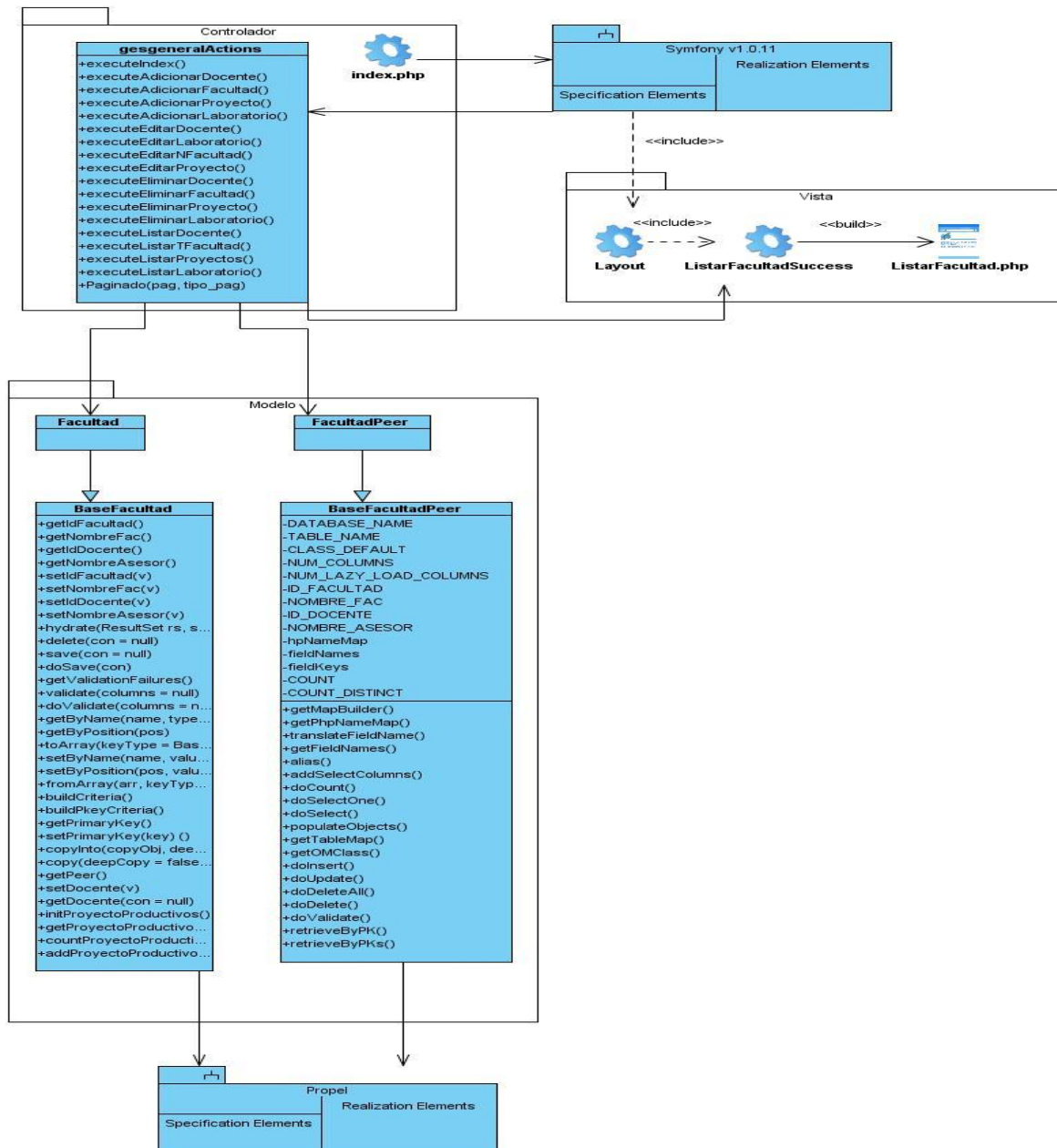


Figura 3.47 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Mostrar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Eliminar.

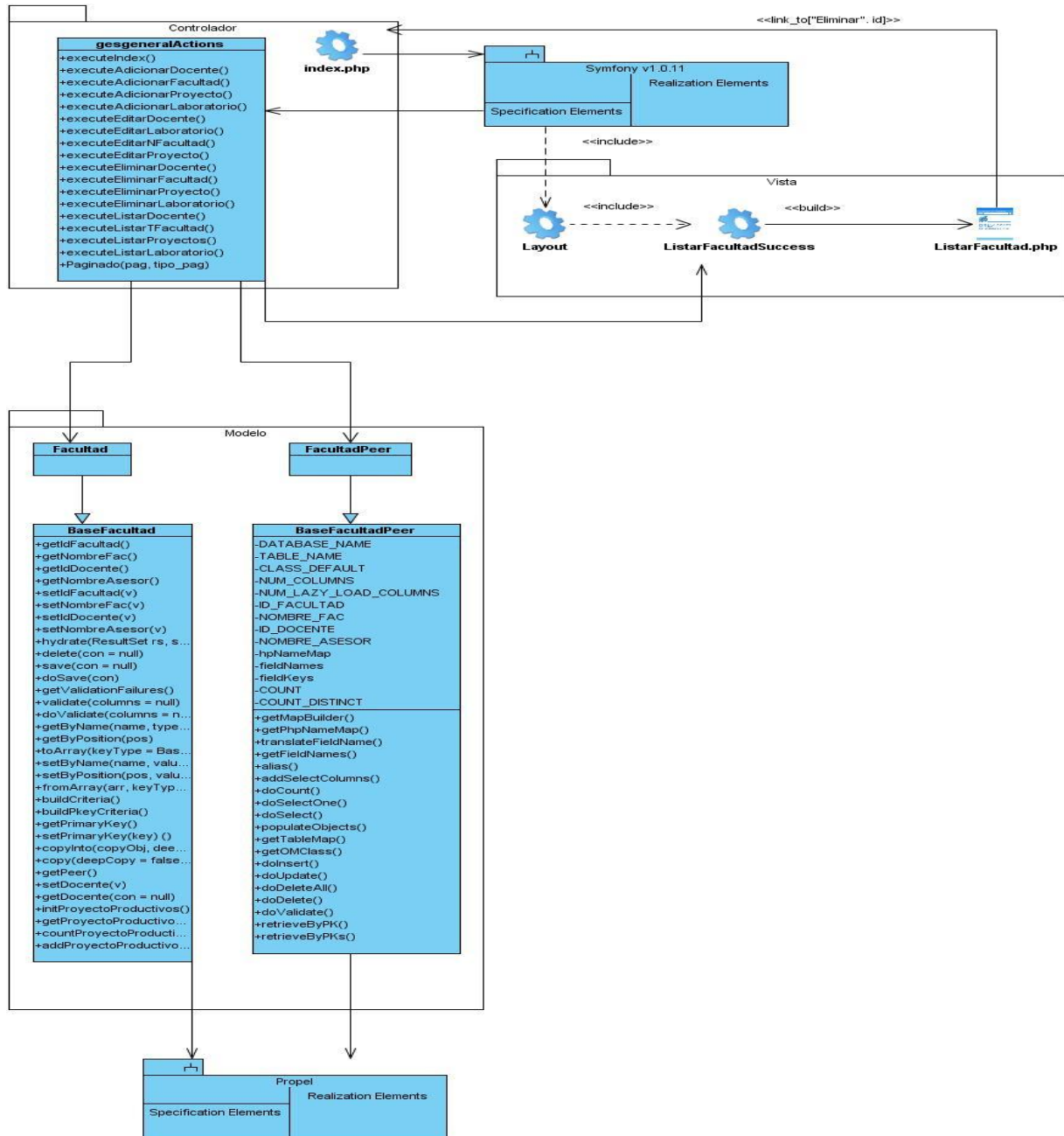


Figura 3.48 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Adicionar.

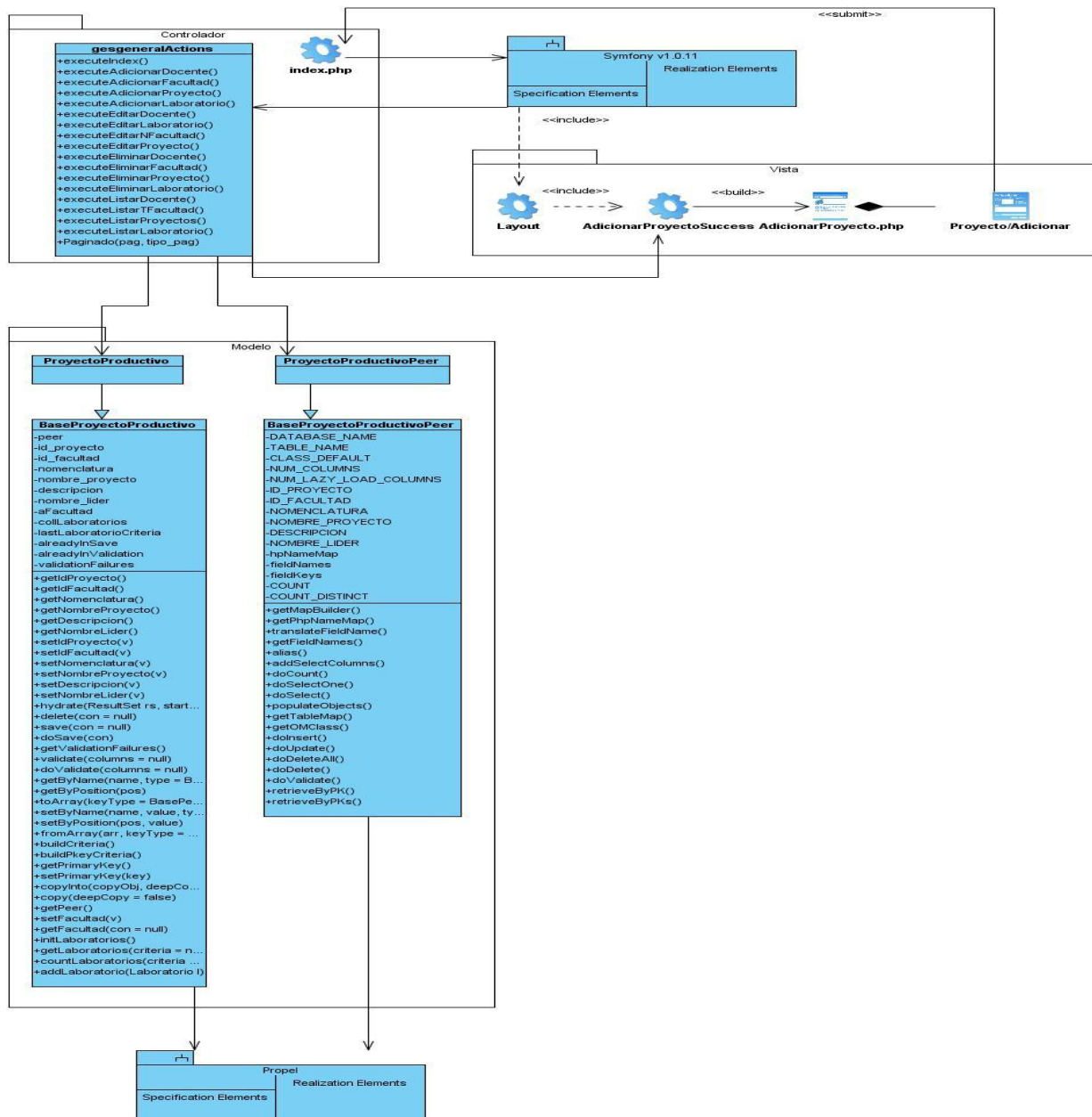


Figura 3.49 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Modificar.

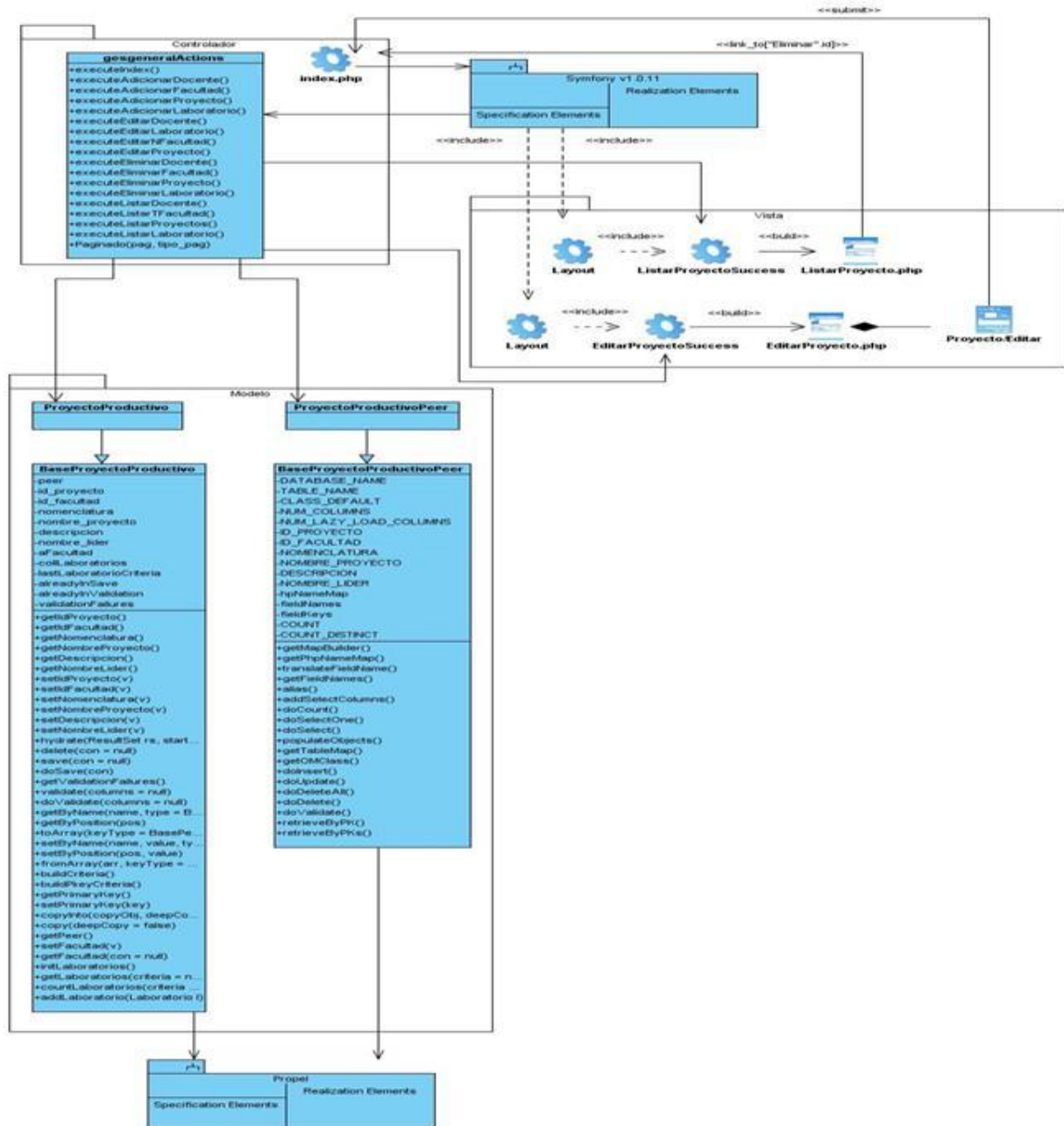


Figura 3.49 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Modificar.

## Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo. Sección Mostrar.

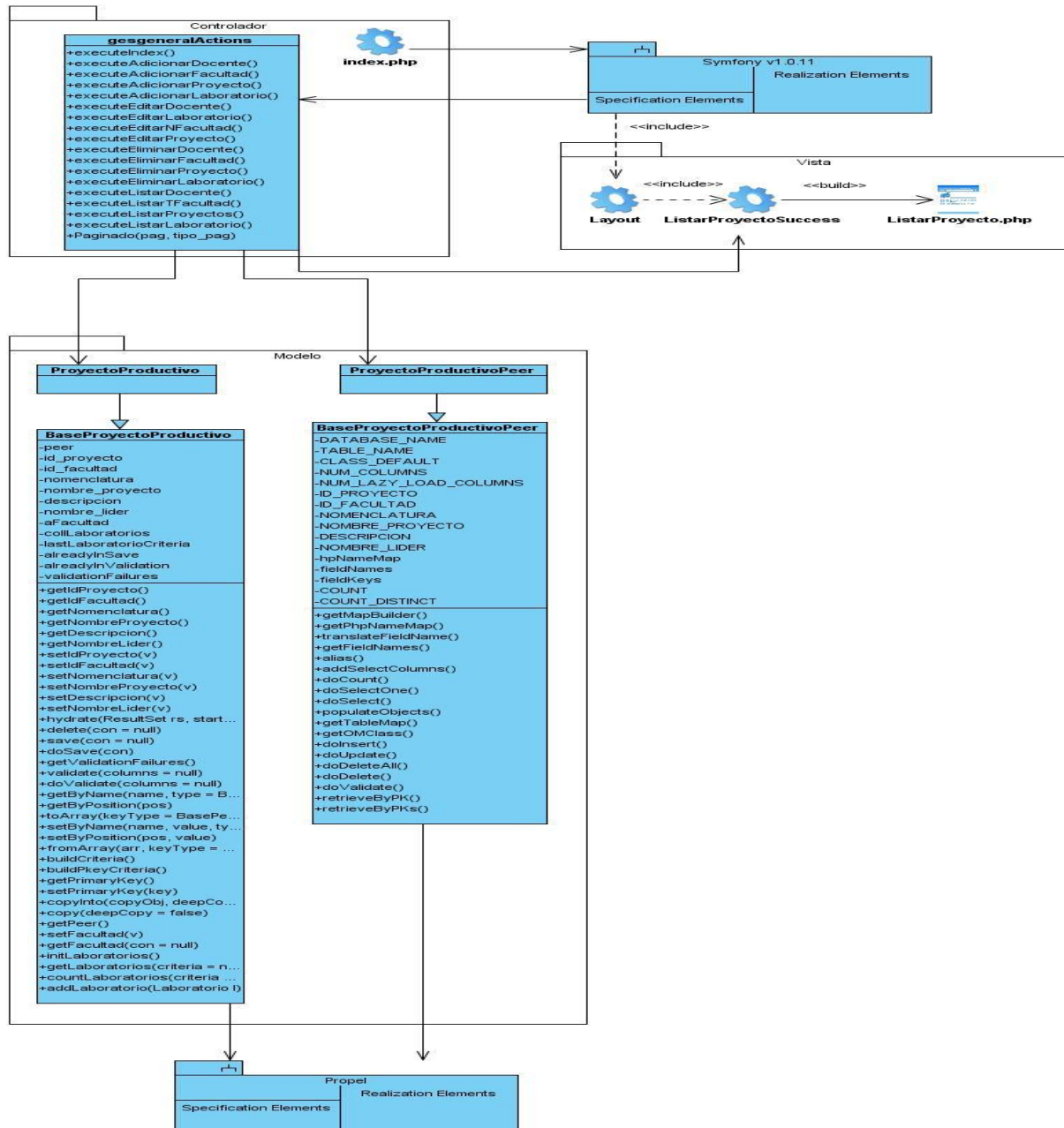


Figura 3.50 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo.

Sección Mostrar.



Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo.  
Sección Eliminar.

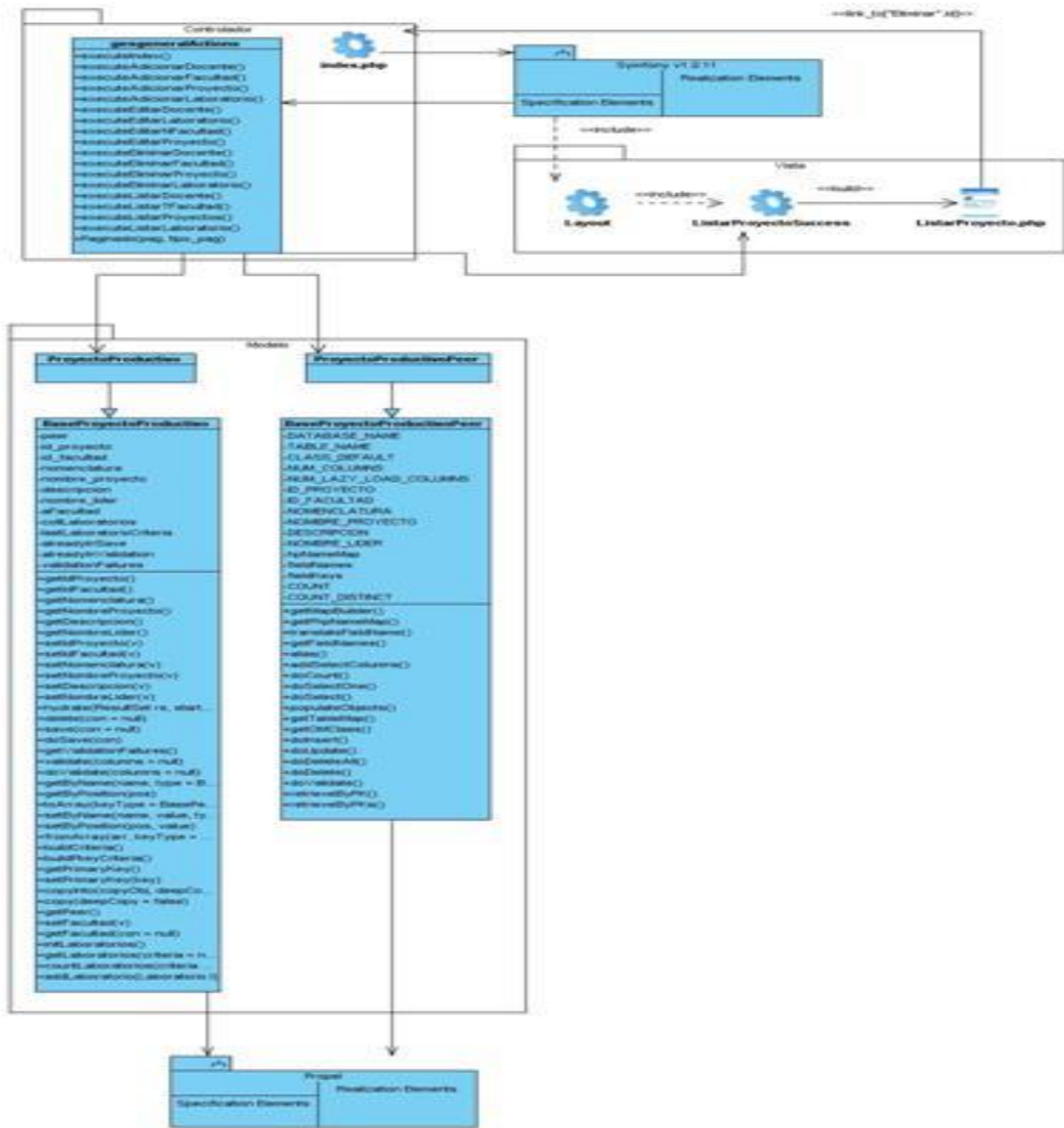


Figura 3.51 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto Productivo.  
Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Adicionar.

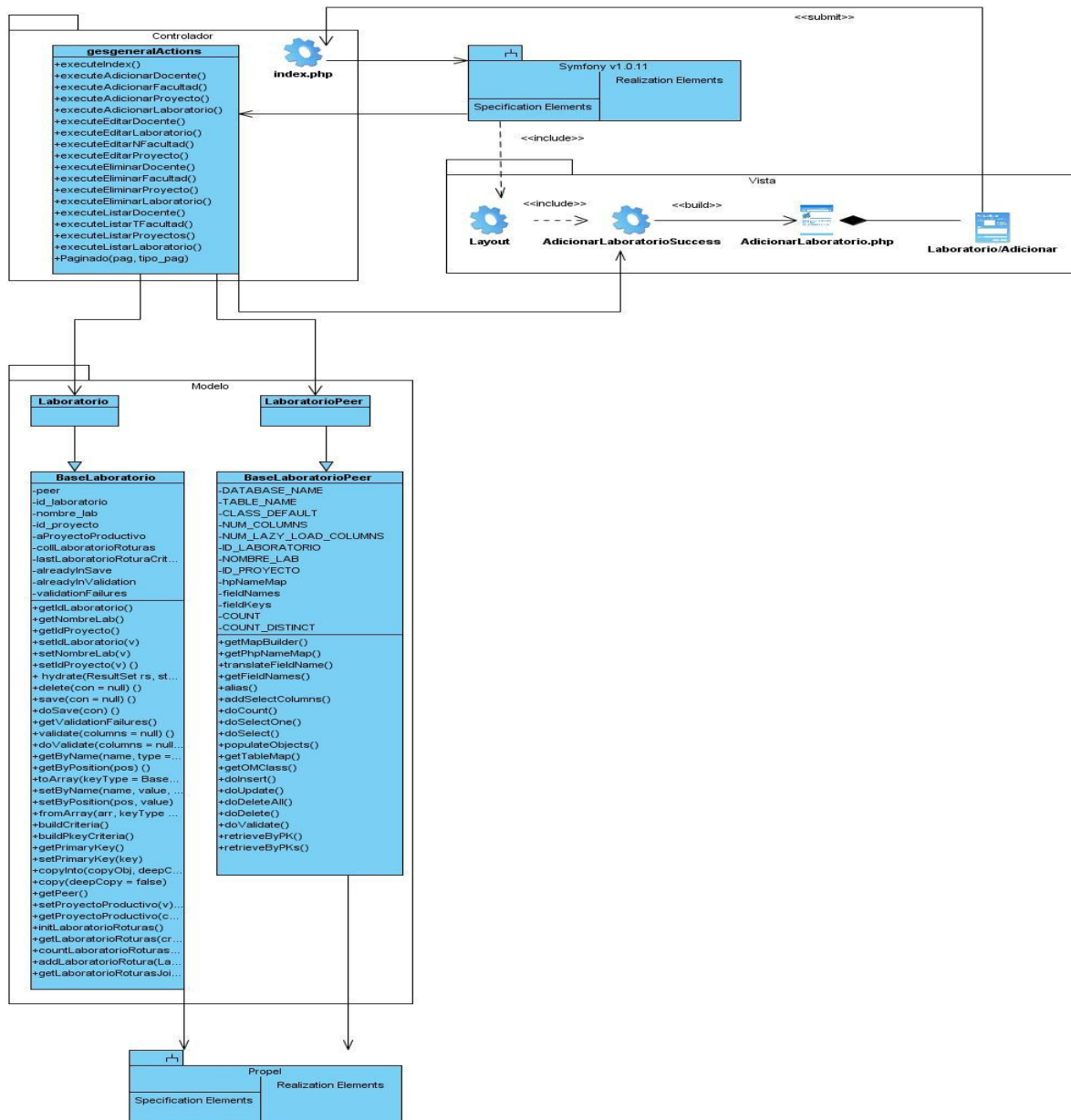


Figura 3.52 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Modificar.

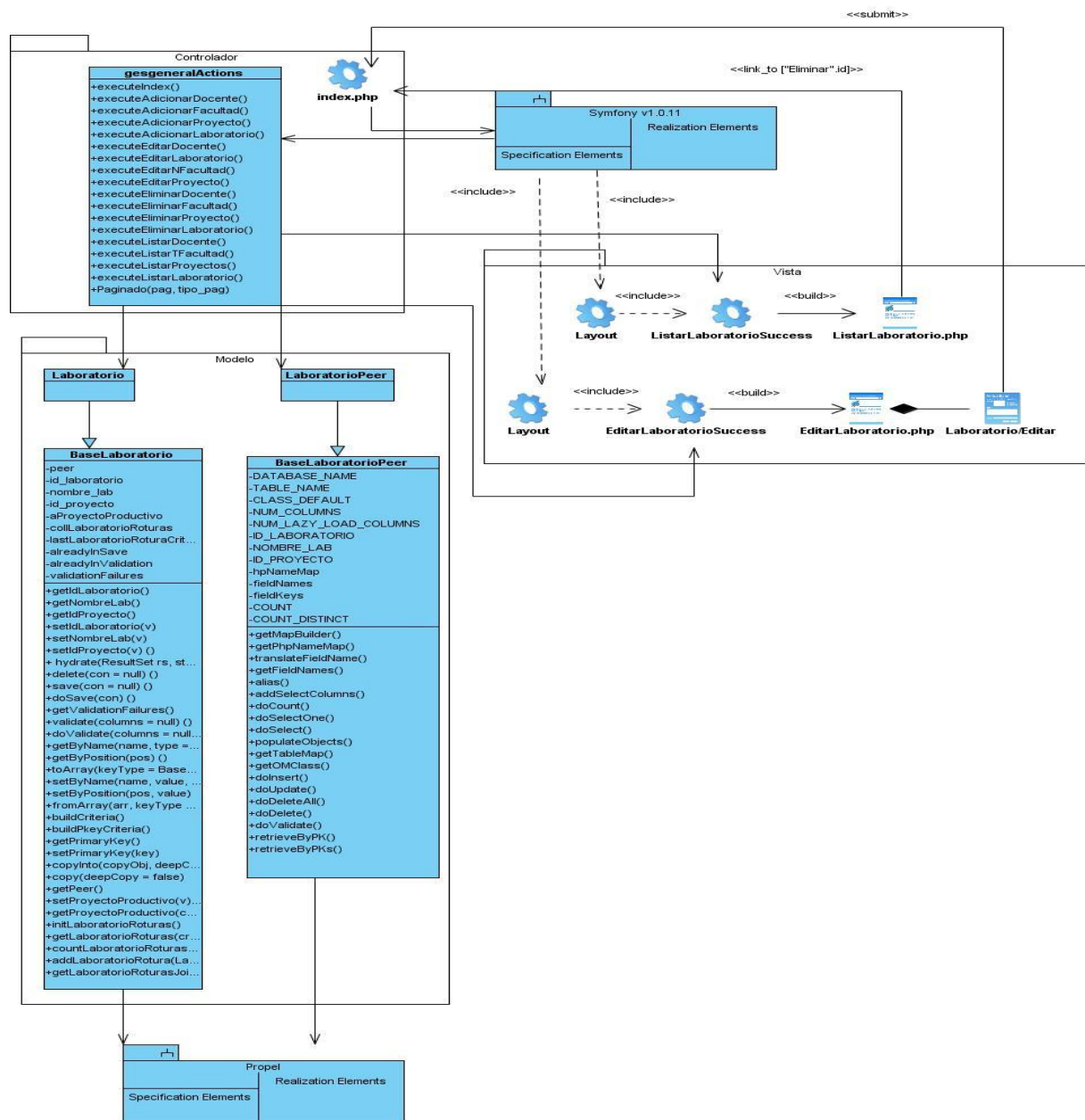


Figura 3.53 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Mostrar.

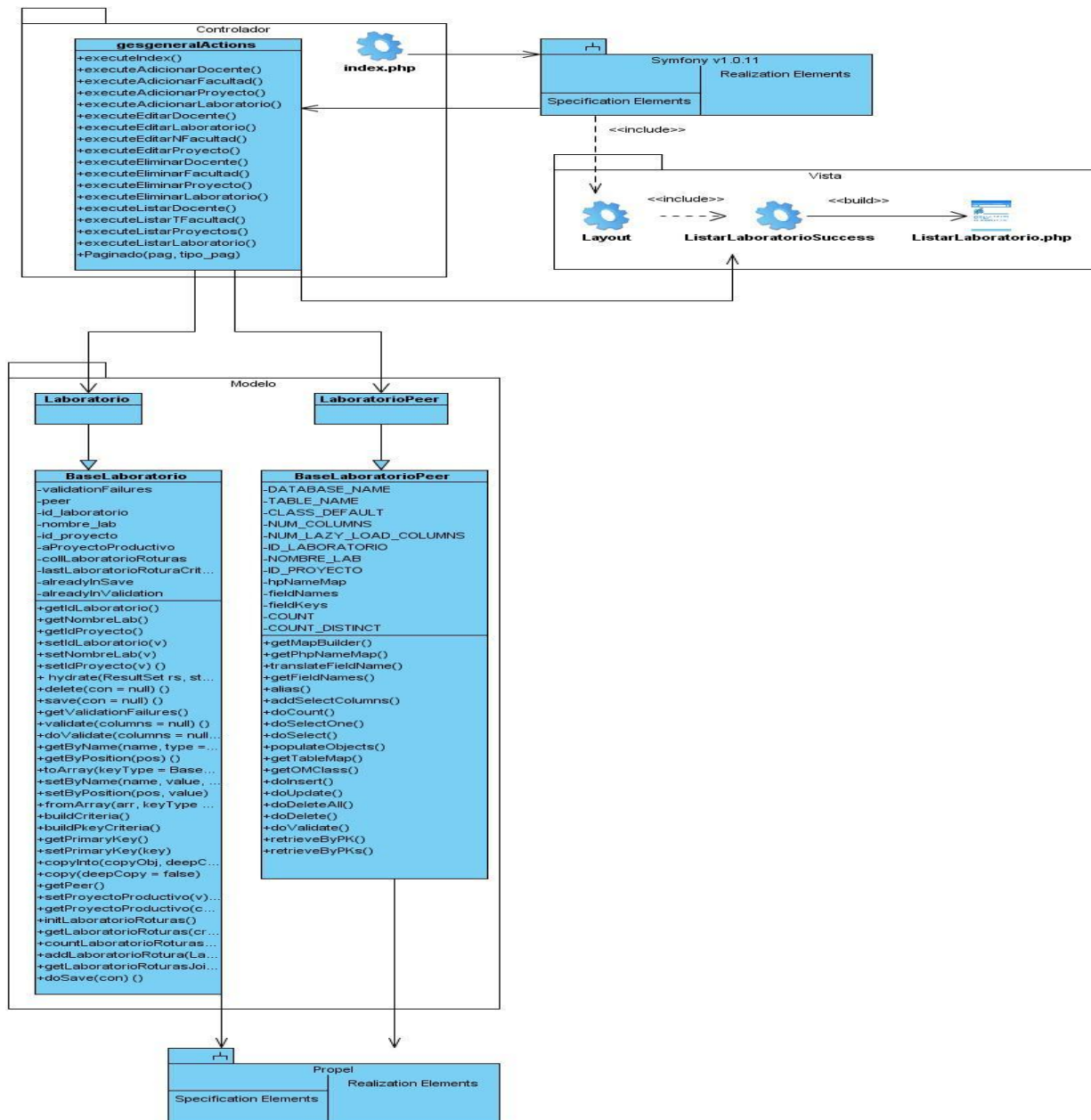


Figura 3.54 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Mostrar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Eliminar.

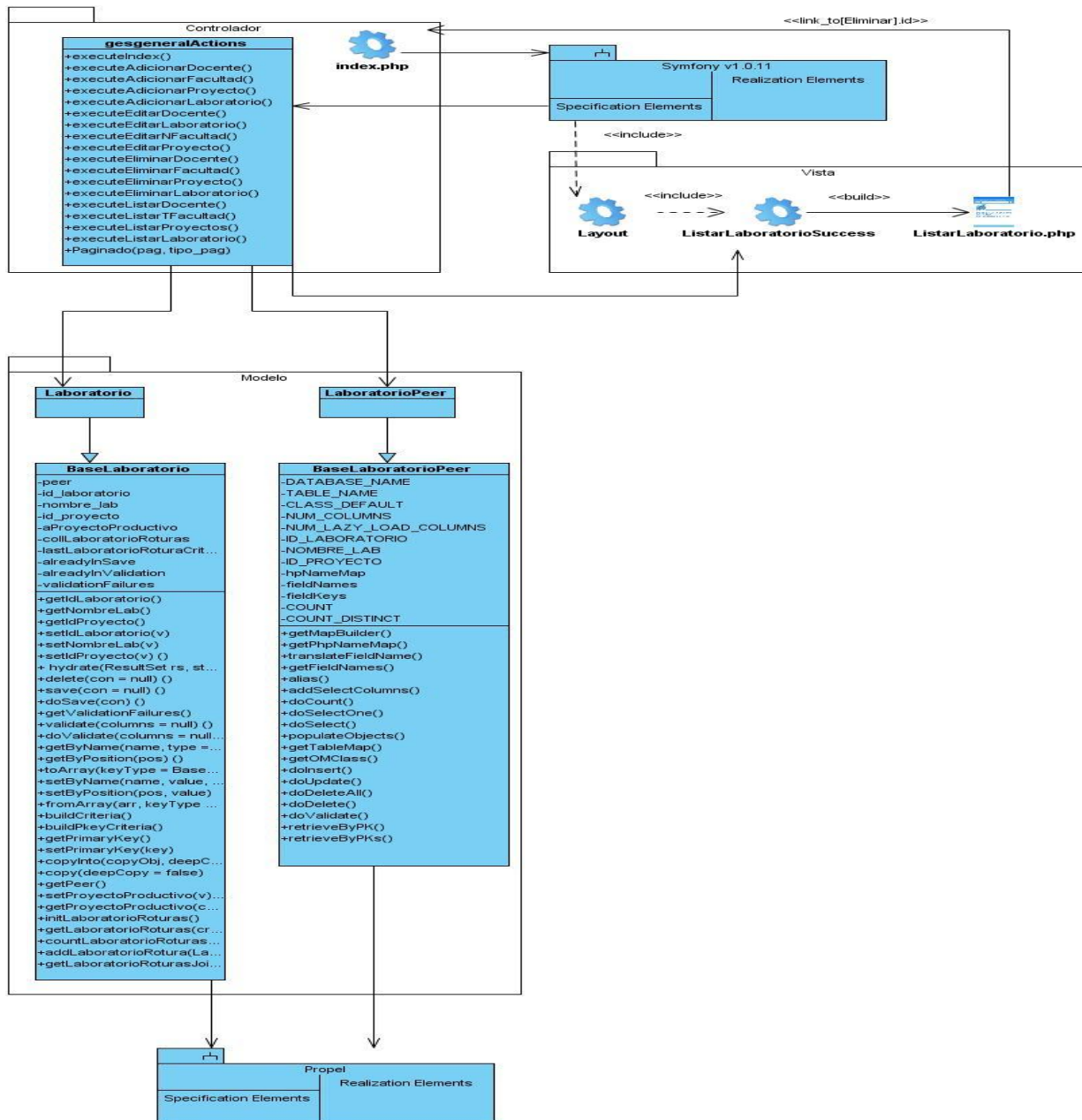


Figura 3.55 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Adicionar.

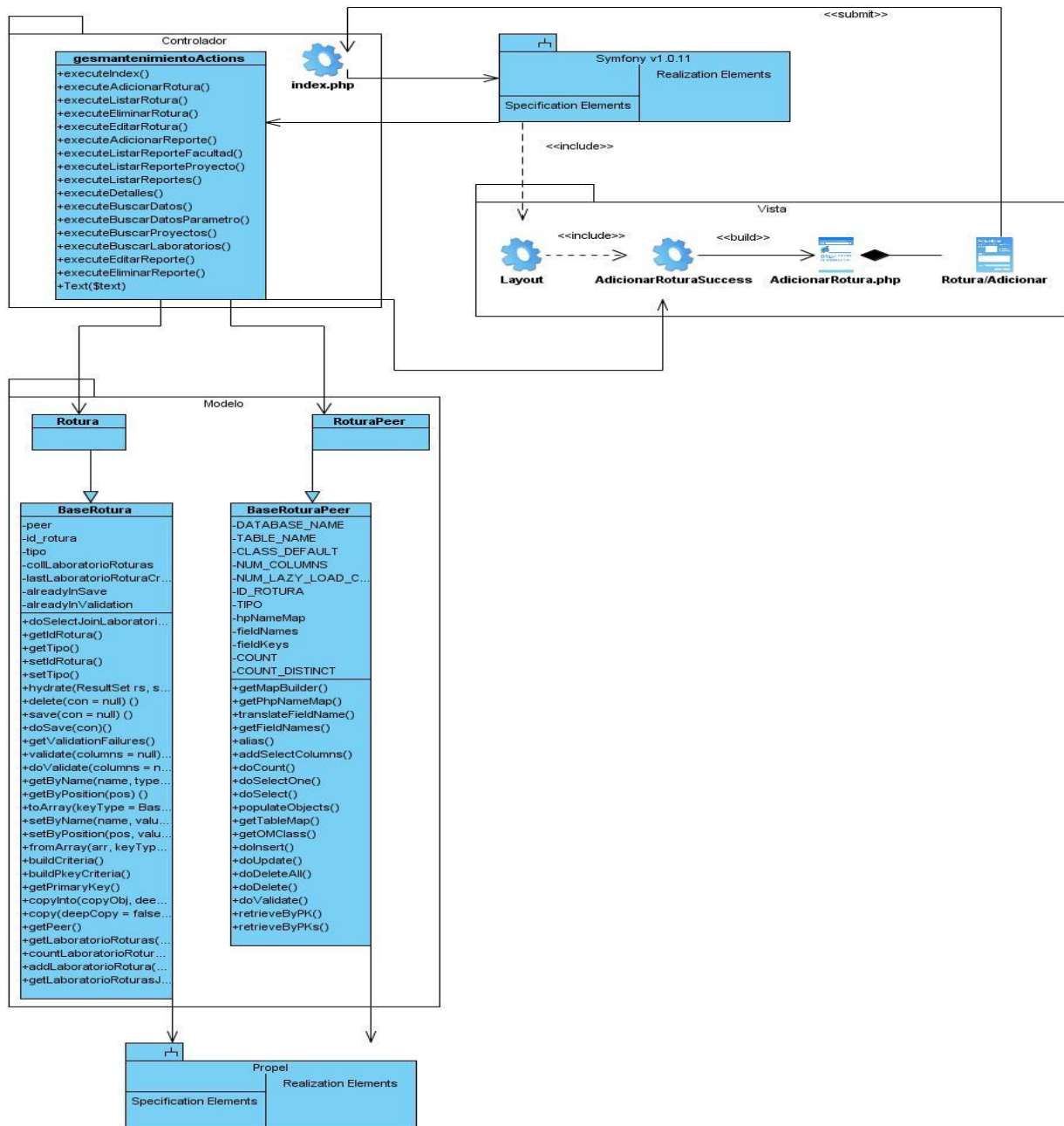


Figura 3.56 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Modificar.

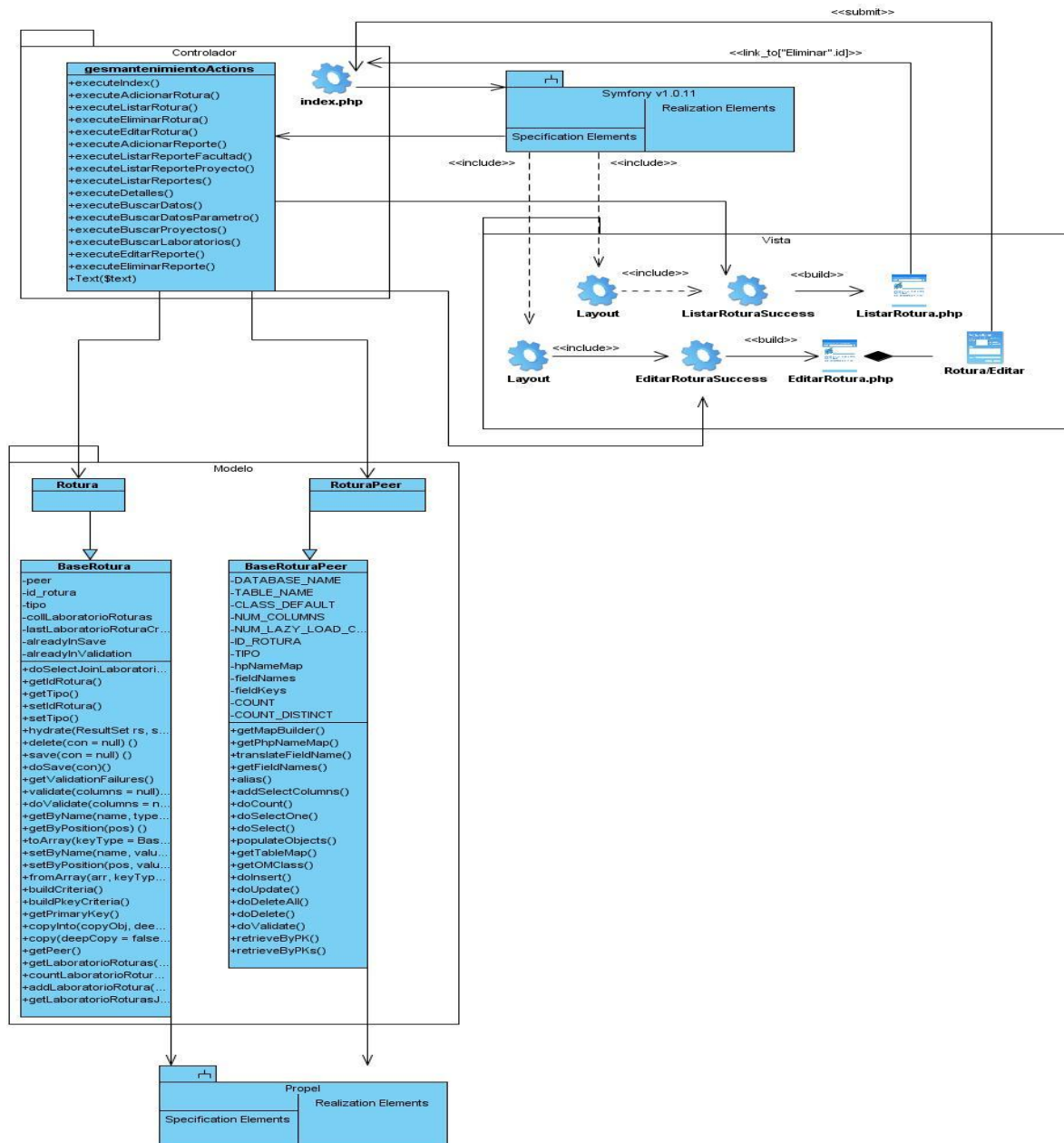


Figura 3.57 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Mostrar.

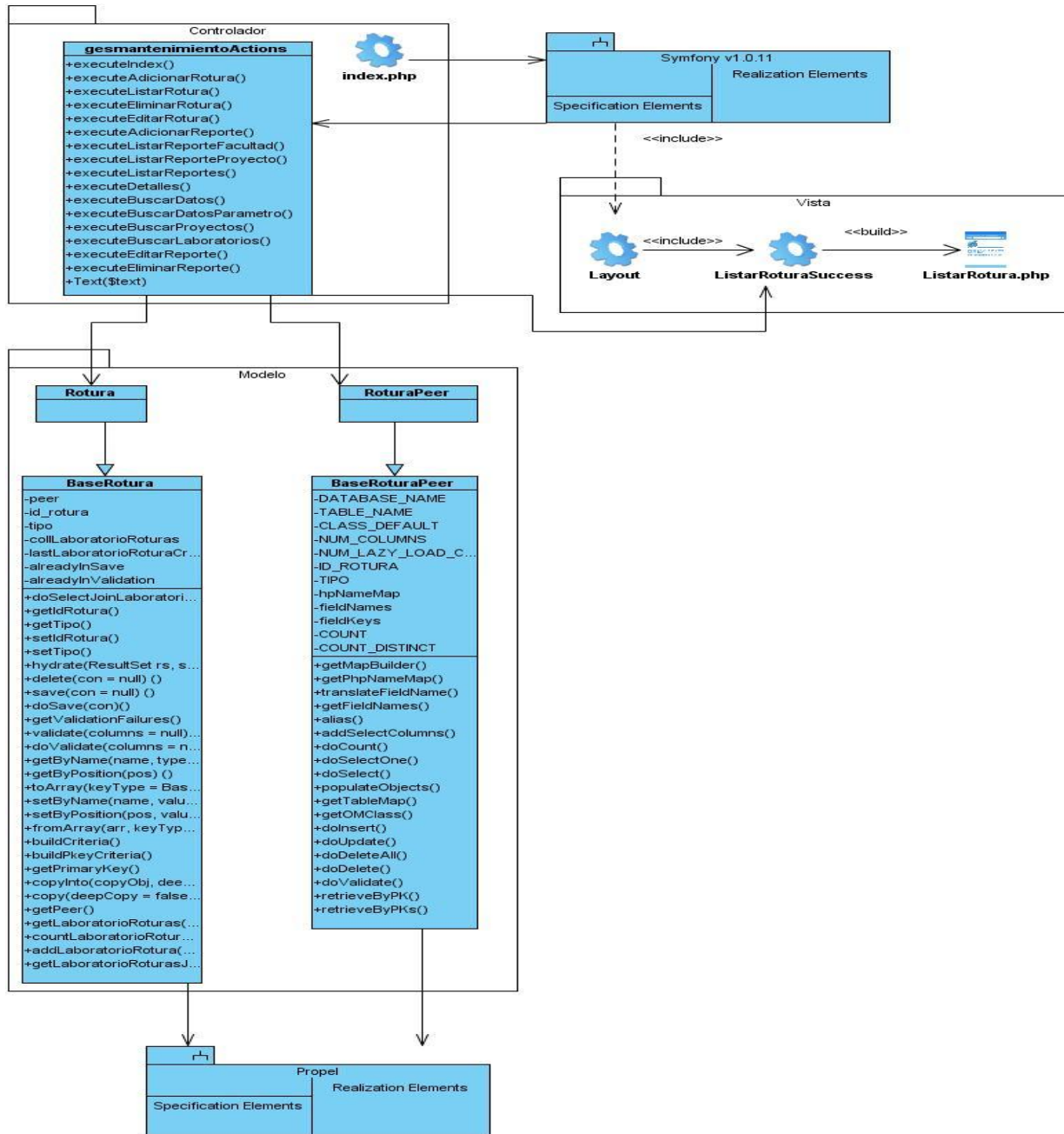


Figura 3.58 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Mostrar.



Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Eliminar.

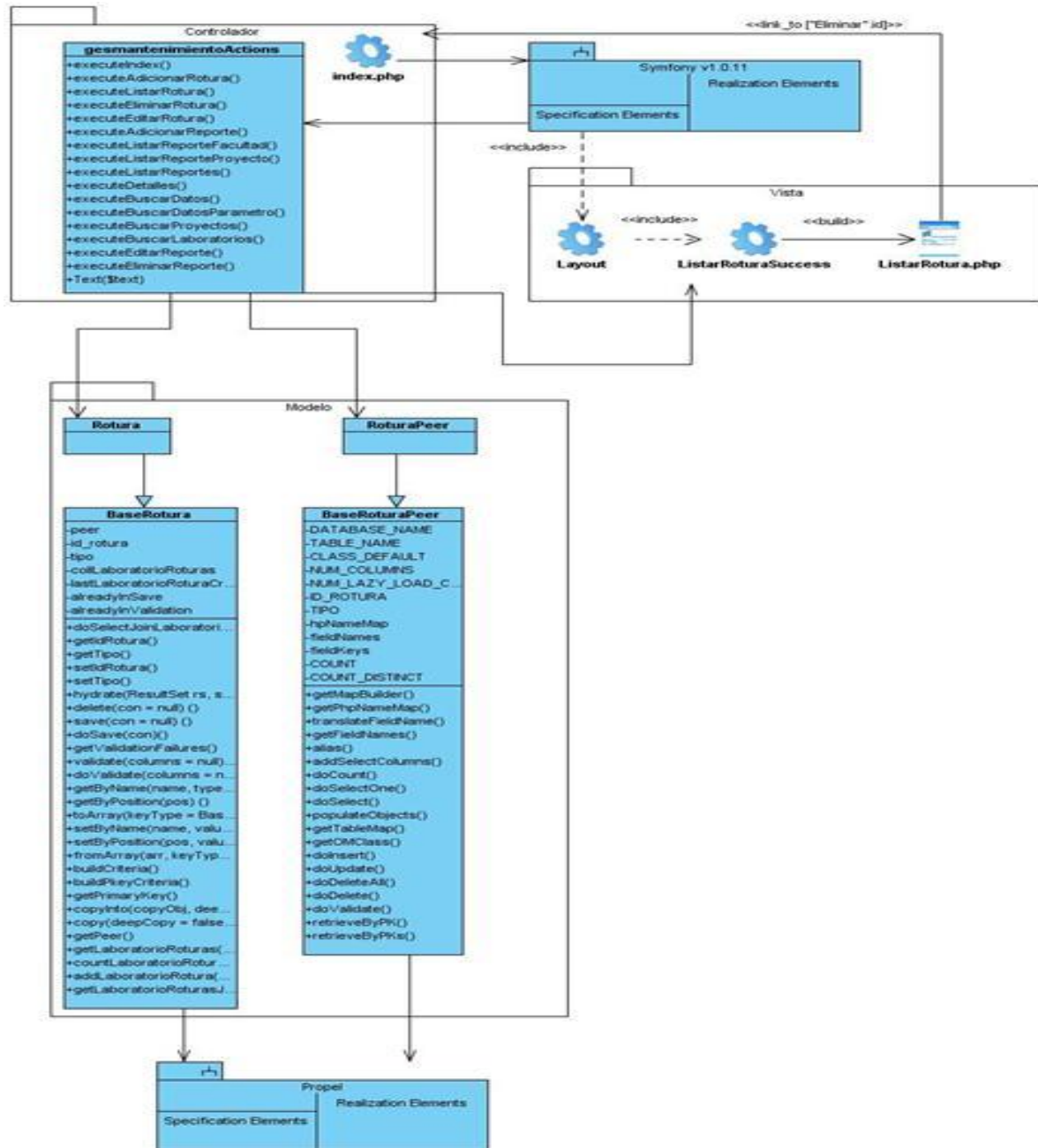


Figura 3.59 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Rotura. Sección Eliminar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Adicionar.

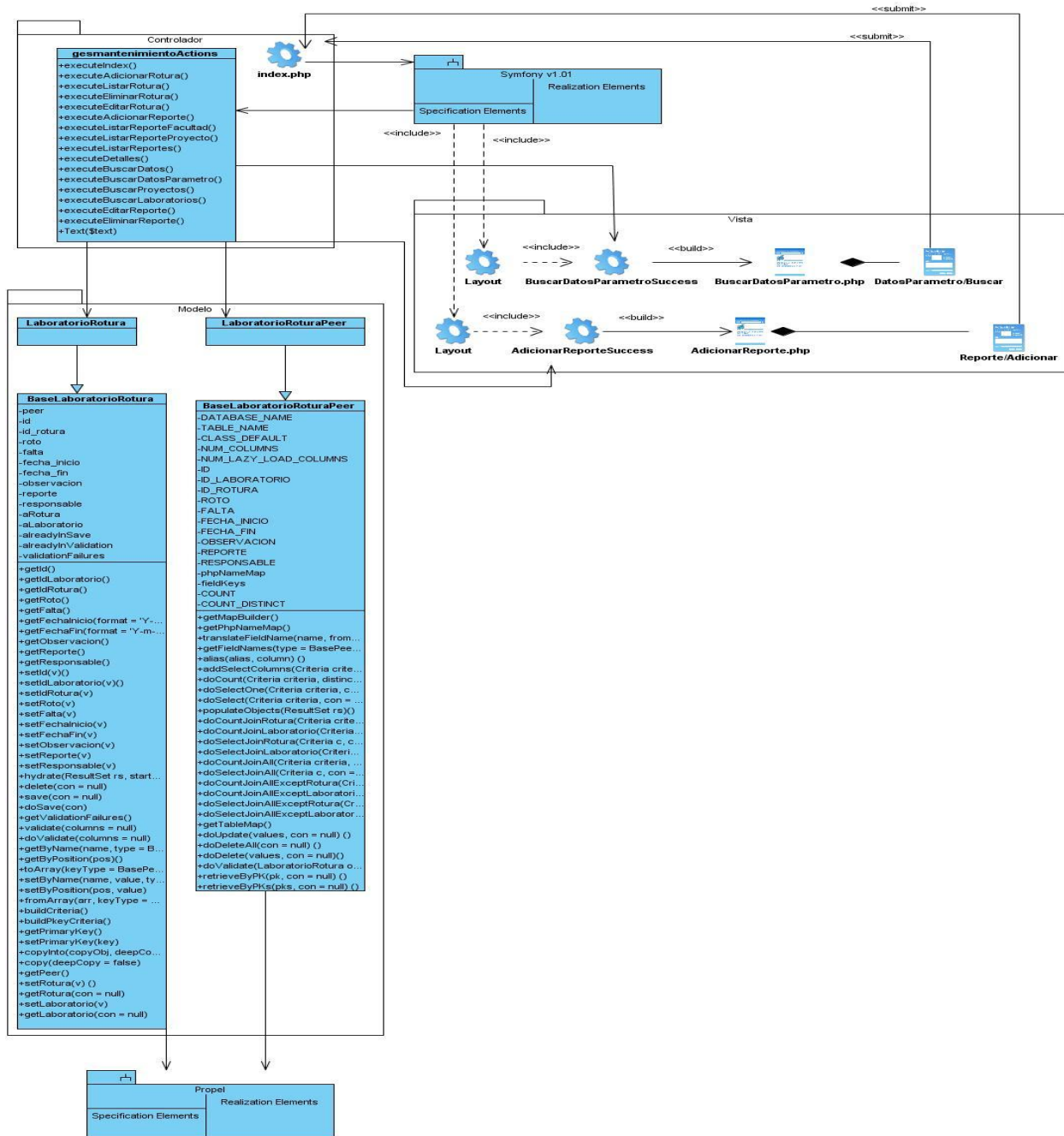


Figura 3.60 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Adicionar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Modificar.

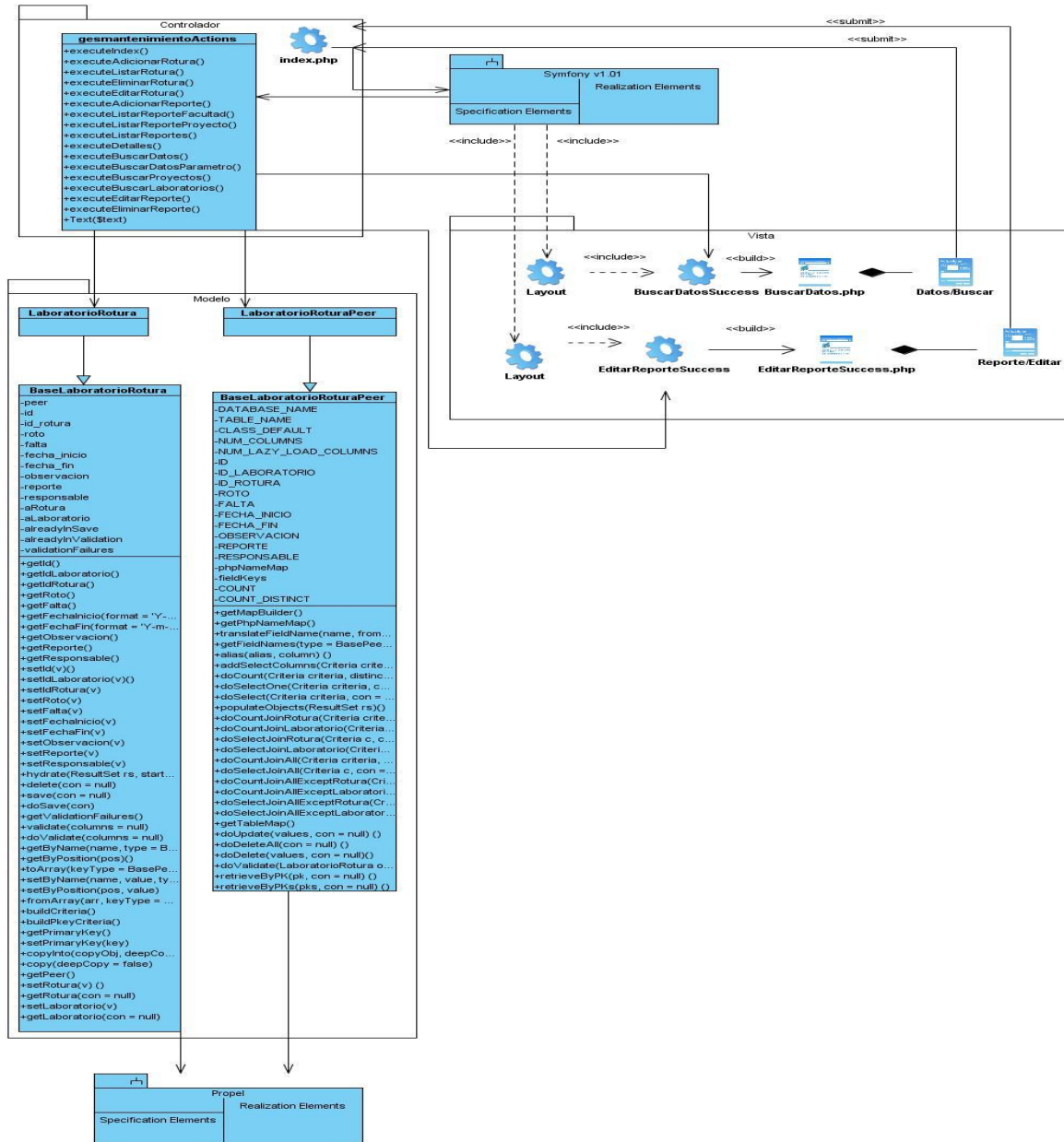


Figura 3.61 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Modificar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Mostar.

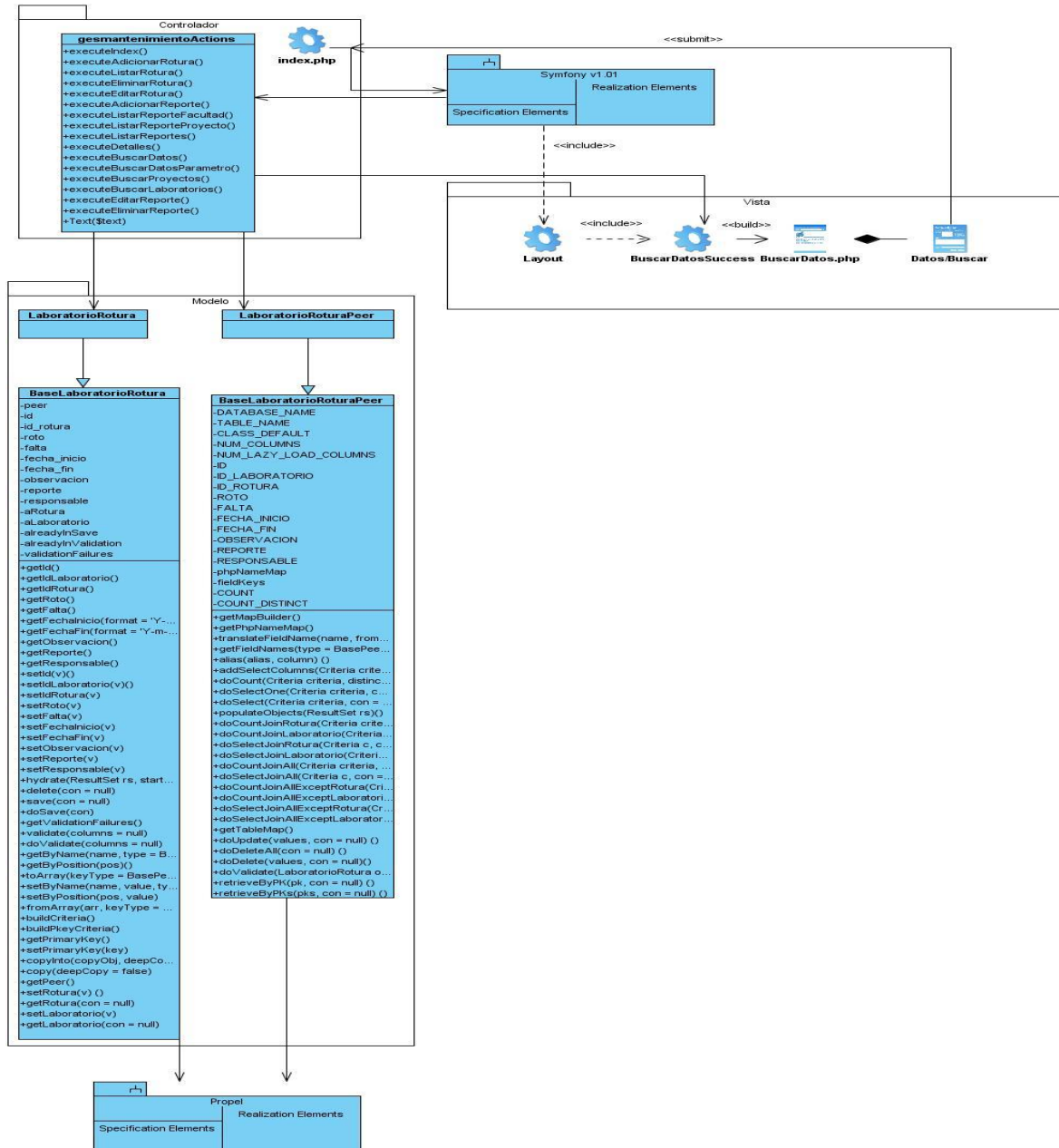


Figura 3.62 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Mostrar.

Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Eliminar

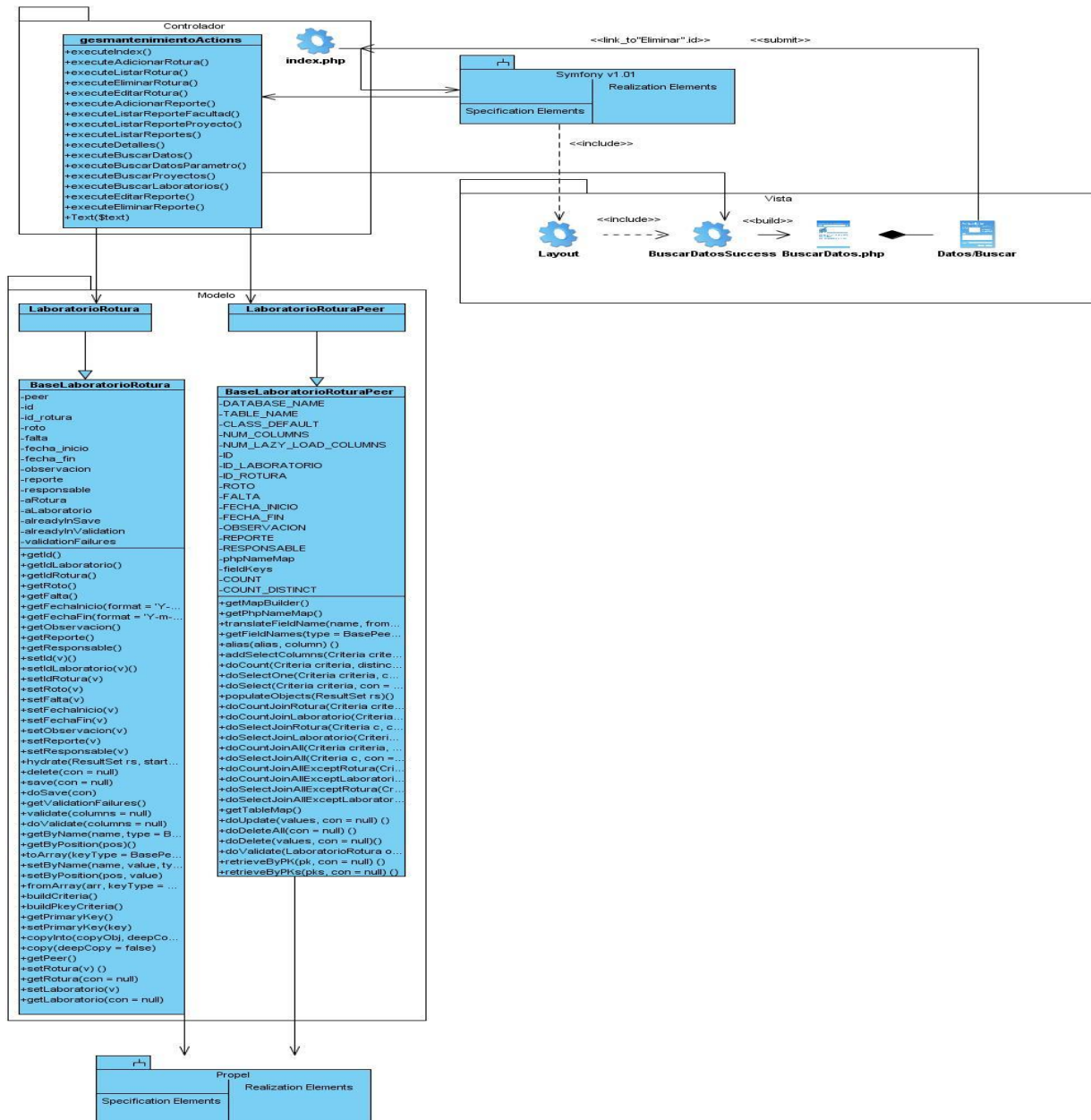


Figura 3.63 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Eliminar.

## Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Exportar

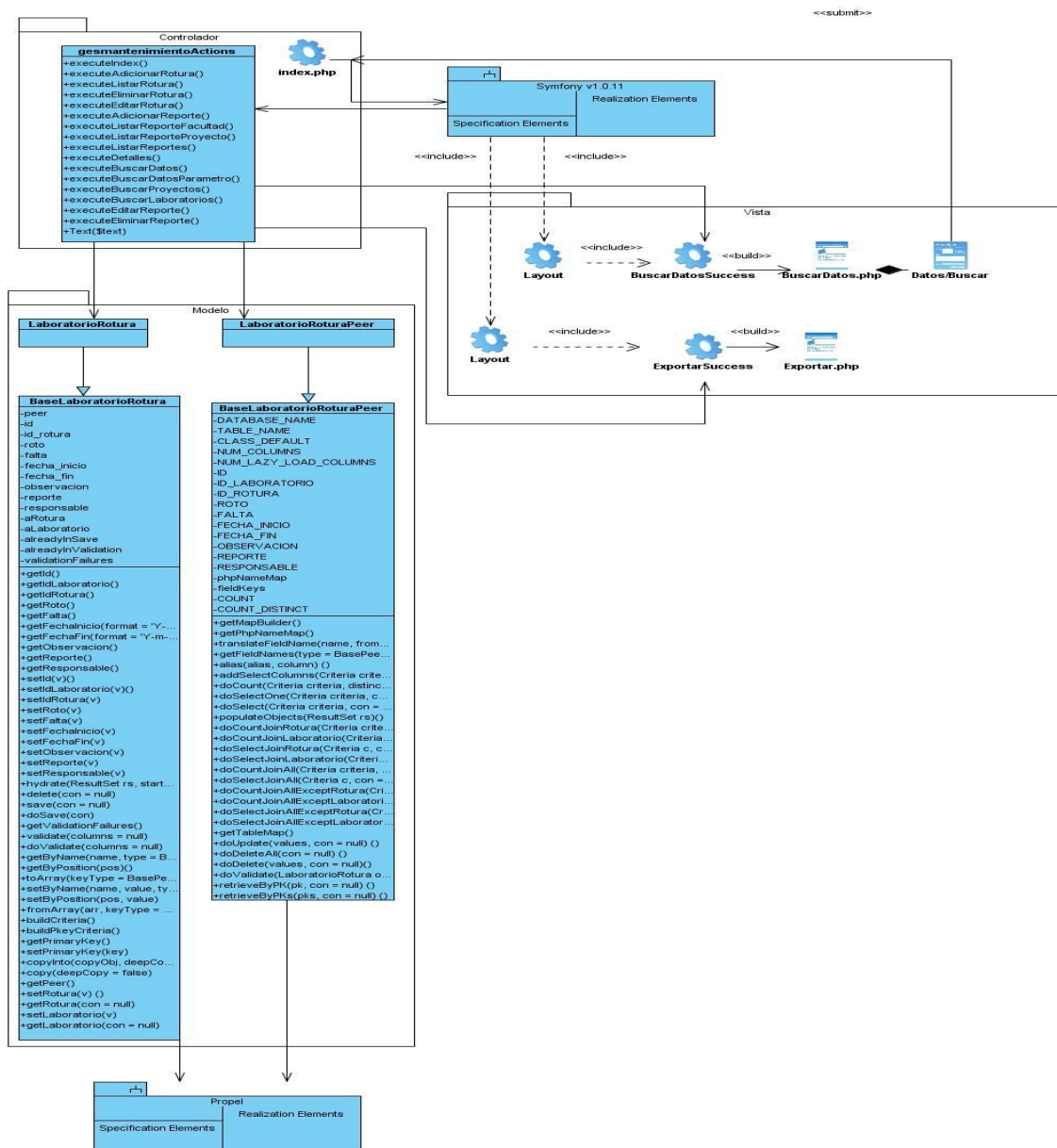


Figura 3.64 Diagrama de Clases del Diseño del Caso de Uso: Gestionar Reporte. Sección Exportar.

## **Glosario de términos**

**ACL (Lista de Control de Acceso):** es un concepto de seguridad informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso que hace el pedido.

**API (Application Programming Interface):** conjunto de funciones residentes en bibliotecas que permiten que una aplicación corra bajo un determinado sistema operativo.

**BCP (Basic Combined Programming Language.):** Es un lenguaje de programación ordenado, potente y muy fácil de adaptar a diferentes arquitecturas.

**CGI (Common Gateway Interface):** Es un acuerdo entre los implementadores del servidor HTTP sobre como integrar los scripts de las puertas de enlace y los programas.

**HTML(Hyper Text Markup Language)** es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web.

**IDE:** Entorno Integrado de Desarrollo.

**ISS (Include Server Side):** es un lenguaje simple del lado del servidor utilizado exclusivamente para la web.

**Licencia BSD:** es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

**Licencia GPL:** Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License. Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

**Metodología:** en un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

**Servicios Web:** Un servicio Web (en inglés Web service) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, a través de la Web.

**SSL (Secure Sockets Layer):** el Protocolo de Capa de Conexión Segura es un protocolo criptográfico que proporciona comunicaciones seguras por una red, comúnmente Internet.

**UML (Unified Modeling Language):** Lenguaje Unificado de Modelado

**XML (Extensible Markup Language):** es un metalenguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos.

**XSS (Cross-site scripting):** es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.