

Universidad de las Ciencias Informáticas



Facultad 8

Trabajo de Diploma

**Título: “Sistema para la Gestión de la
Tecnología en la Universidad de las Ciencias
Informáticas.”**

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

Autores: Aldo Milanés López.
Renier Montano García.

Tutor: Ing. Frank David Avalos Palomo.

DEDICATORIA

De Aldo:

A mi madre por ser mi guía y faro durante toda mi vida, por darme apoyo incondicional, por luchar junto a mí hasta cumplir este sueño que también es suyo; por estar siempre.

A mis amigos: Daniel, Evelio, Leandro, Borges y Ernesto por ser mis hermanos de toda la vida, por estar junto a mí en todo momento; por ayudarme a ser quien soy.

A Yulía por el apoyo incondicional que me ha brindado, por alentarme a buscar siempre lo mejor; por caminar junto a mí sin miedos hacia el futuro.

A mi tío Heriberto por ser más que un padre para mí, por exhortarme a seguir siempre adelante; por brindarme su apoyo incondicional.

A mi hermana y a Alejandro.

De Renier:

Dedico este trabajo de manera general a todas las personas que de una forma u otra han hecho posible que este sueño mío se haga realidad.

A Diamela por ser mi compañera en las buenas y en las malas luchando junto a mí por nuestro futuro.

A mi tía Lidia que me brindó apoyo de todo tipo.

A mi madre que no descansa para que siempre tenga un porvenir brillante

Especialmente a mi abuela, la cual ha sido prácticamente mi compañera de tesis, la cual ha sufrido conmigo en las malas y disfrutado en las buenas, pero siempre dándome un empujón largo hacia el triunfo final.

AGRADECIMIENTOS

Agradecer a Frank David, por guiarnos y apoyarnos en todo momento; muchas gracias por ayudarnos a realizar este trabajo.

A Yasser, Michel y Ernesto por mantenerse siempre pendientes de cada detalle de nuestra tesis. Gracias por la ayuda y el apoyo incondicionales que nos han brindado.

A Alberto, Arlan, Lucía, Elizabeth, Danielito, Dioleisis, Oreste, Yamir, Norge y a todos aquellos amigos de de una forma u otra nos han ayudado a llegar hasta aquí; gracias de todo corazón.

Aldo y Renier.

DECLARACION DE AUTORIA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Soporte de Desarrollo de la Infraestructura Productiva de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de ____ del año 2009.

Aldo Levian Milanés López

.Renier Montano García

Ing. Frank David Avalos Palomo

DATOS DE CONTACTO

TUTOR: Ingeniero Frank David Avalos Palomo (email: favalos@uci.cu).

Profesor graduado de Ingeniería en Ciencias Informáticas. Ha impartido la asignatura Introducción a la Programación y Programación I. Actualmente se desempeña como Jefe del Grupo de Soporte de Desarrollo de la Infraestructura Productiva (IP) de la Universidad de las Ciencias Informáticas (UCI).

OPINION DEL TUTOR

RESUMEN

Actualmente en la Universidad de las Ciencias Informáticas, se realiza un proceso fundamental en lo referente al manejo y distribución de los recursos tecnológicos. En este proceso denominado Gestión Tecnológica participan diferentes direcciones de la Universidad, y se manipula en torno al mismo una considerable cantidad de información muy variada y de carácter sensible. La gestión de dicha información se torna cada vez más engorrosa debido al creciente desarrollo de proyectos productivos que demandan un alto número de recursos tecnológicos. La Universidad de las Ciencias Informáticas es un centro productor de software con responsabilidad de crear productos para satisfacer el mercado nacional así como las demandas de países y empresas extranjeras que solicitan los servicios de esta entidad, de ahí la importancia de realizar el proceso mencionado de manera más eficiente y óptima.

Por ello surge la necesidad de implementar un sistema informático que de soporte al desarrollo del proceso en estudio, y que proporcione rapidez en el manejo de la información, consistencia y seguridad en los datos, y sobre todo la optimización del proceso. Con este objetivo, se utilizan herramientas y tecnología adecuadas, que permiten que el desarrollo de la aplicación sea con calidad, y originando un producto compatible con plataformas libres, de fácil manejo, y que responda a aquellos requisitos definidos por el cliente de una manera eficiente. Algunas de esas tecnologías y herramientas son: Gestor de Base de Datos PostgreSQL, Entorno de Desarrollo Integrado Eclipse, Framework Symfony, Lenguajes de Programación PHP5 y Java Script, Herramienta CASE Visual Paradigm y como Metodología de Desarrollo, RUP.

El sistema que se propone en la presente investigación se desarrolla en un entorno Web, estableciendo diferentes niveles de acceso a la información para asegurar su protección y adecuado tratamiento.

INDICE

INDICE DE TABLAS	13
INDICE DE FIGURAS	13
INTRODUCCIÓN	14
Métodos Científicos usados en la investigación.	17
Métodos teóricos.	17
Métodos Empíricos.	17
Aportes.	17
CAPITULO 1. Fundamentación Teórica.	17
1.1 Introducción	17
1.2 Conceptos esenciales.	18
1.2.1 ¿Qué es tecnología?	18
1.2.2 ¿Qué es Tecnología Informática?	18
1.2.3 ¿Qué son las Tecnologías de la Información y las Comunicaciones (TIC)?	18
1.2.4 ¿Qué es gestión de la tecnología?	18
1.2.5 ¿Qué es un Cliente?	18
1.3 Estado del Arte.	19
1.3.1 Elementos de la Gestión Tecnológica en el mundo.	19
1.3.2 Elementos de la Gestión Tecnológica en Cuba.	20
1.3.3 Elementos de la Gestión Tecnológica en la Universidad de las Ciencias informáticas.	22
1.5 Soluciones existentes.	25
1.6 Tecnología y Herramientas.	29
1.6.1 Tecnologías de la Información y las Comunicaciones (TIC).	29
1.6.1.1 Sistemas de Gestión.	29
1.6.1.1.1 Sistemas de Gestión con plataforma Web.	30
1.6.2 Servidores Web.	31
1.6.2.1 Servidor HTTP Apache.	31
1.6.2.2 Internet Information Services (IIS).	31
1.6.3 Lenguajes de Programación Web.	32
1.6.3.1 Lenguajes del lado del servidor.	32
1.6.3.2 Lenguajes de Programación del lado del cliente.	33
1.6.4 Metodologías de Desarrollo de Software.	34
1.6.4.1 RUP.	35
1.6.4.2 Programación extrema (XP).	36
1.6.5 Lenguaje Unificado de Modelado (UML).	37
1.6.6 Herramientas.	37
1.6.6.1 Sistema Gestores de Bases de Datos.	37
1.6.6.1.1 PostgreSQL	38
1.6.6.1.2 MySQL.	38
1.6.6.2 Herramienta CASE.	38
1.6.6.2.1 Visual Paradigm.	39
1.6.6.2.2 Rational Rose.	39
1.6.6.3 Framework.	40
1.6.6.3.1 Framework Symfony.	40
1.6.6.3.2 Prado Framework.	42
1.6.6.4 Entorno Integrado de Desarrollo (IDE).	42
1.6.6.4.1 Zend Studio.	43
1.6.6.4.2 Eclipse	44
1.6.7 Justificación de la selección.	44
1.6.8 Conclusiones.	47
CAPÍTULO 2. Presentación de la solución propuesta.	48

2.1	Introducción.	48
2.2	Modelo de Negocio.	48
2.2.1	Actores y trabajadores del negocio.	48
2.2.2	Descripción General del Proceso de Negocio: Solicitar Hardware.	49
2.2.3	Diagrama de Casos de Uso del Negocio (DCUN).	49
2.3	Realización de Casos de Uso.	49
2.3.1	Descripción del Caso de Uso de Negocio: Solicitar Hardware.	50
2.6	Diagrama de Actividades.	52
2.7	Modelo del Sistema.	52
2.7.1	Requerimientos Funcionales.	52
2.7.2	Requerimientos No Funcionales.	54
2.7.4	Diagrama de Casos de Uso del Sistema (DCUS).	56
2.7.5	Descripción de los Casos de Uso.	56
2.7.5.1	Descripción del Caso de Uso: Autenticar Usuario.	56
2.7.5.2	Descripción del Caso de Uso: Gestionar Usuario.	57
2.7.5.3	Descripción del Caso de Uso: Gestionar Componentes.	59
2.7.5.4	Descripción del Caso de Uso: Gestionar Características.	60
2.7.5.5	Descripción del Caso de Uso: Gestionar Laboratorios.	62
2.7.5.6	Descripción del Caso de Uso: Gestionar Facultad.	63
2.7.5.7	Descripción del Caso de Uso: Gestionar Docente.	65
2.7.5.8	Descripción del Caso de Uso: Gestionar Proyecto Productivo.	66
2.7.5.9	Descripción del Caso de Uso: Gestionar Solicitud.	68
2.7.5.10	Descripción del Caso de Uso: Gestionar Medidas.	69
I	Descripción del Caso de Uso: Eliminar Proyecto.	71
II	Descripción del Caso de Uso: Eliminar Docente.	72
III	Descripción del Caso de Uso: Eliminar Facultad.	72
IV	Descripción del Caso de Uso: Eliminar Laboratorio.	73
V	Descripción del Caso de Uso: Eliminar Solicitud.	74
VI	Descripción del Caso de Uso: Eliminar Componente.	75
VII	Descripción del Caso de Uso: Eliminar Características.	76
VIII	Descripción del Caso de Uso: Eliminar Medidas.	77
2.8	Conclusiones.	78
CAPITULO 3. Construcción de la solución propuesta.		79
3.1	Introducción.	79
3.2	Análisis.	79
3.3	Diagramas de Clases del Análisis.	79
3.3.1	Diagramas de Interacción del Análisis.	79
3.4	Diseño.	79
3.4.1	Diagrama de Clases del Diseño.	81
3.5	Diseño de la Base de Datos.	81
3.5.1	Diagrama de Clases Persistentes.	82
3.5.2	Diagrama Entidad Relación.	82
3.6	Modelo de Despliegue.	83
3.7	Modelo de Implementación.	84
3.7.1	Diagrama de Componentes.	84
I.	Paquete Model.	85
II.	Paquete Módulo.	85
3.8	Conclusiones.	85
CONCLUSIONES.		86

RECOMENDACIONES	87
REFERENCIAS BIBLIOGRAFICAS.....	88
BILIOGRAFIA CONSULTADA	90
ANEXOS	91
Anexos 1. Diagrama de Clases del Análisis.	91
Diagramas de Clases del Análisis del Caso de Uso: Autenticar Usuario.	91
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Usuario.	91
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Docente.	92
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Facultad.	92
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Proyecto Productivo.	93
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Laboratorios.	93
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Características.	94
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Componentes.	94
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Medidas.....	95
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Solicitud.	95
Anexos 2. Diagramas de Colaboración	96
Diagramas de Colaboración: Autenticar Usuario.	96
Diagramas de Colaboración: Gestionar Usuario (Sección Adicionar).	96
Diagramas de Colaboración: Gestionar Usuario (Sección Listar).	96
Diagramas de Colaboración: Gestionar Usuario (Sección Modificar).	97
Diagramas de Colaboración: Gestionar Usuario (Sección Eliminar).	97
Diagramas de Colaboración: Gestionar Docente (Sección Adicionar).	97
Diagramas de Colaboración: Gestionar Docente (Sección Listar).	98
Diagramas de Colaboración: Gestionar Docente (Sección Eliminar).	98
Diagramas de Colaboración: Gestionar Facultad (Sección Adicionar).	99
Diagramas de Colaboración: Gestionar Facultad (Sección Listar).	99
Diagramas de Colaboración: Gestionar Facultad (Sección Modificar).	99
Diagramas de Colaboración: Gestionar Facultad (Sección Eliminar).	100
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Adicionar).	100
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Listar).	100
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Modificar).	101
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Eliminar).	101
Diagramas de Colaboración: Gestionar Laboratorios (Sección Adicionar).	101
Diagramas de Colaboración: Gestionar Laboratorios (Sección Listar).	102
Diagramas de Colaboración: Gestionar Laboratorios (Sección Modificar).	102
Diagramas de Colaboración: Gestionar Laboratorios (Sección Eliminar).	102
Diagramas de Colaboración: Gestionar Características (Sección Adicionar).	103
Diagramas de Colaboración: Gestionar Características (Sección Listar).	103
Diagramas de Colaboración: Gestionar Características (Sección Modificar).	103
Diagramas de Colaboración: Gestionar Características (Sección Eliminar).	104
Diagramas de Colaboración: Gestionar Componentes (Sección Adicionar).	104
Diagramas de Colaboración: Gestionar Componentes (Sección Listar).	104
Diagramas de Colaboración: Gestionar Componentes (Sección Modificar).	105

Diagramas de Colaboración: Gestionar Componentes (Sección Eliminar).	105
Diagramas de Colaboración: Gestionar Medidas (Sección Adicionar).	105
Diagramas de Colaboración: Gestionar Medidas (Sección Listar).	106
Diagramas de Colaboración: Gestionar Medidas (Sección Modificar).	106
Diagramas de Colaboración: Gestionar Medidas (Sección Eliminar).	106
Diagramas de Colaboración: Gestionar Solicitud (Sección Adicionar).	107
Diagramas de Colaboración: Gestionar Solicitud (Sección Listar).	107
Diagramas de Colaboración: Gestionar Solicitud (Sección Modificar).	107
Diagramas de Colaboración: Gestionar Solicitud (Sección Eliminar).	108
Anexos 3. Diagramas de Clases de Diseño.	109
Diagramas Clases de Diseño del Caso de Uso: Autenticar Usuario.	109
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Adicionar.	110
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Modificar.	111
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Listar.	112
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Eliminar.	113
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Adicionar.	114
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Modificar.	115
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Listar.	116
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Eliminar.	117
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Adicionar.	118
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Modificar.	119
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Listar.	120
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Eliminar.	121
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Adicionar.	122
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Modificar.	123
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Listar.	124
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Eliminar.	125
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Adicionar.	126
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Modificar.	127
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Listar.	128
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Eliminar.	129
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Adicionar.	130
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Modificar.	131
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Listar.	132
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Eliminar.	133
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Adicionar.	134
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Modificar.	135
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Listar.	136
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Eliminar.	137
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Adicionar.	138
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Modificar.	139
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Listar.	140

Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Eliminar.	141
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Adicionar.	142
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Modificar.	143
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Listar.	144
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Eliminar.	145
GLOSARIO DE TÉRMINOS	146

INDICE DE TABLAS

Tabla 1. Actores del Negocio.....	48
Tabla 2 Trabajadores del Negocio.	49
Tabla 3 Descripción Textual del caso de Uso: Solicitar Hardware	50
Tabla 4 Actores del Sistema.....	56
Tabla 5 Descripción Textual del Caso de Uso: Autenticar Usuario.....	57
Tabla 6 Descripción Textual del Caso de Uso: Gestionar Usuario.	59
Tabla 7 Descripción Textual del Caso de Uso: Gestionar Componente.	60
Tabla 8 Descripción Textual del Caso de Uso: Gestionar Características.	62
Tabla 9 Descripción Textual del Caso de Uso: Gestionar Solicitudes Laboratorio.....	63
Tabla 10 Descripción Textual del Caso de Uso: Gestionar Facultad.	65
Tabla 11 Descripción Textual del Caso de Uso: Gestionar Docente.....	66
Tabla 12 Descripción Textual del Caso de Uso: Gestionar Proyecto Productivo.	68
Tabla 13 Descripción Textual del Caso de Uso: Gestionar Solicitud.	69
Tabla 14 Descripción Textual del Caso de Uso: Gestionar Medidas.	71
Tabla #15: Descripción textual del Caso de Uso: Eliminar Proyecto.....	72
Tabla #16: Descripción textual del Caso de Uso: Eliminar Docente.....	72
Tabla #17: Descripción textual del Caso de Uso: Eliminar Facultad.....	73
Tabla #18: Descripción textual del Caso de Uso: Eliminar Facultad.....	74
Tabla #19: Descripción textual del Caso de Uso: Eliminar Solicitud.....	75
Tabla #20: Descripción textual del Caso de Uso: Eliminar Componente.....	76
Tabla #21: Descripción textual del Caso de Uso: Eliminar Características.....	77
Tabla #22: Descripción textual del Caso de Uso: Eliminar Medidas.....	78

INDICE DE FIGURAS

Fig. 1 Red de Centros de gestión tecnológica en Cuba.....	22
Figura 2. Imagen de la página de Inicio de la aplicación GATSRVER.	26
Figura 3. Imagen de la página de Solicitud de Mantenimiento de la aplicación GATSRVER.	26
Figura 4. Diagrama de Casos de Uso del Negocio.	49
Figura 5. Modelo de Objetos.....	51
Figura 6. Diagrama de Casos de Uso del Sistema.....	56
Figura 7. Diagrama de Clases Persistentes.	82
Figura 8. Diagrama Entidad Relación.	82
Figura 9. Diagrama de Despliegue.	83
Figura 10. Diagrama de Componentes.	84
Figura 11. Paquete Model.....	85
Figura 12. Paquete Módulo.....	85

INTRODUCCIÓN

Históricamente las tecnologías han sido usadas para satisfacer necesidades esenciales del hombre: alimentación, vestuario, vivienda, protección personal, relación social, comprensión del mundo natural y social. La historia de la tecnología es la historia de la invención de herramientas y técnicas con un propósito práctico. La historia moderna está relacionada íntimamente con la historia de la ciencia, pues el descubrimiento de nuevos conocimientos ha permitido crear nuevas cosas y, recíprocamente, se han podido realizar nuevos descubrimientos científicos gracias al desarrollo de nuevas tecnologías, que han extendido las posibilidades de experimentación y adquisición del conocimiento. Los artefactos tecnológicos son productos de una economía, una fuerza del crecimiento económico y una buena parte de la vida. Las innovaciones tecnológicas afectan y están afectadas por las tradiciones culturales de la sociedad.

Tecnología “es el conjunto de saberes, destrezas y medios necesarios para llegar a un fin predeterminado mediante el uso de objetos artificiales o artefactos.”^[1]

Las primeras tecnologías de importancia aparecieron durante la Edad de Piedra y estaban asociadas a la supervivencia, la obtención de alimentos y su preparación. El fuego, las herramientas de piedra, las armas y el atuendo fueron desarrollos tecnológicos de gran importancia de este periodo. En la Edad de Piedra posterior (Neolítico), se desarrollaron los rudimentos de la tecnología agraria, éstas fueron las bases de la tecnología industrial moderna.

La Revolución industrial es un periodo histórico en el que el Reino Unido en primer lugar, y el resto de la Europa continental después, sufren el mayor conjunto de transformaciones socioeconómicas, tecnológicas y culturales de la Historia de la humanidad desde el Neolítico. La economía basada en el trabajo manual fue reemplazada por otra dominada por la industria y la manufactura. La expansión del comercio fue favorecida por la mejora de las rutas de transporte y posteriormente por el nacimiento del ferrocarril. El siglo XIX produjo grandes avances en las tecnologías de transporte, construcción y comunicaciones. Charles Babbage concibió la computación mecánica logrando que diera frutos. La tecnología del siglo XX se desarrolló rápidamente. Las tecnologías de comunicaciones, transporte, la difusión de la educación, el empleo del método científico y las inversiones en investigación contribuyeron al avance de la ciencia y la tecnología modernas. Algunas tecnologías como la computación se desarrollaron rápidamente debido a las guerras o a la amenaza de ellas, pues hubo muchos avances científicos asociados a la investigación y el desarrollo militares, como la computación electrónica. En los pocos años que han transcurrido del siglo XXI la tecnología ha avanzado vertiginosamente, progresando en casi todos los campos de la ciencia. La tasa de desarrollo de los computadores es un ejemplo de la aceleración del progreso tecnológico, lo que lleva a algunos a pronosticar el advenimiento de una singularidad tecnológica en este siglo. Los avances en la tecnología de comunicaciones se traducen en avances en el sistema económico, el sistema político, la distribución de bienes, la igualdad social y otros aspectos de la vida del hombre.

A partir del aumento de velocidad, memoria y disminución de tamaño de las/los computadoras/ordenadores la

tecnologías de las comunicaciones y en particular la tecnología informática han experimentado enormes progresos.

Surge aquí un concepto importante que es el de las Tecnologías de la Información y las Comunicaciones (TIC).

Las Tecnologías de la Información y las Comunicaciones: “son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario, y está destinada a optimizar la comunicación humana.” [2]

Actualmente, todos de una forma u otra reconocen la importancia que reviste el empleo de estas Tecnologías de la Información y las Comunicaciones; siendo imposible por tanto perfeccionar nuestra labor sin la aplicación de éstas. En Cuba, las Tecnologías de la Información y las Comunicaciones se aplican en todas las esferas de la vida, en el sector de la salud, los servicios, la educación, las investigaciones y la gestión económica. La Universidad de las Ciencias Informáticas (UCI), se yergue como principal impulsor de la aplicación de las Tecnologías de la Información y las Comunicaciones en el país, ya que es un centro destinado al estudio y desarrollo de la Tecnología Informática.

Tecnología Informática: “es el estudio, diseño, desarrollo, puesta en práctica, ayuda o gerencia de los sistemas informáticos computarizados, particularmente usos del software y hardware.” [3]

La Universidad de las Ciencias Informáticas, como centro productor de software, tiene la responsabilidad para con sus clientes de crear productos con calidad, para esto necesita de una buena gestión de la tecnología (equipos y hardware) disponible por el centro, de manera que pueda distribuirse orientada a una mejor producción. Actualmente el proceso de Gestión de las Tecnologías en la Universidad de las Ciencias Informáticas se desarrolla de manera manual, lo que provoca deficiencia en la calidad del mismo, debido a lo engorroso que puede tornarse este ante la cantidad de pedidos que realizan los diferentes proyectos, la complejidad de la distribución ya que debe llevarse un control estricto de todo el equipamiento informático que existe en la universidad, que el proceso se realiza a través del llenado de planillas y generación de más planillas para facturar cada uno de los pedidos que se hacen a la Dirección de Tecnología y además se necesitan determinada cantidad de firmas de directivos para que el pedido sea finalmente aceptado.

A partir de esta situación surge la necesidad de implementar un sistema automatizado para la Universidad de las Ciencias Informáticas que ofrezca seguridad, calidad y la funcionalidad que agilice el proceso al que se hace referencia.

Por tanto el **problema científico** queda formulado en la siguiente interrogante:

¿Cómo lograr que la Gestión de las Tecnologías en la Universidad de las Ciencias Informáticas se torne menos engorrosa y más rápida?

En la actualidad en la Universidad de las Ciencias Informáticas no se esta usando ningún sistema para la Gestión de las Tecnologías.

Arazón de esto, el **objeto de investigación** es el proceso de Gestión de las Tecnologías en la Universidad de las Ciencias Informáticas y el **campo de acción** es la automatización del proceso de Gestión de las Tecnologías en la Universidad de las Ciencias Informáticas.

Se plantea entonces como **objetivo general** desarrollar un sistema que permita la Gestión de las Tecnologías en la Universidad de las Ciencias Informáticas de manera más sencilla y más rápida, y concretamente los **objetivos específicos** planteados son los siguientes:

1. Realizar un estudio del entorno de trabajo.
2. Aplicar la metodología, lenguaje de programación y herramientas adecuadas para realizar el diseño óptimo con vista a la implementación de la aplicación.
3. Desarrollar un sistema informático capaz de automatizar el proceso objeto de estudio.
4. Documentar la investigación con vista a la socialización de la información de la misma.

A partir de todo lo antes expuesto surge la siguiente **idea a defender**: La aplicación de un sistema informático permitirá que la Gestión de las Tecnologías, que se realiza en la Universidad de las Ciencias Informáticas, se torne menos engorrosa y más rápida.

Finalmente se propone la realización de las siguientes **tareas para cumplir los objetivos** planteados:

1. Identificar las necesidades de la institución a través de la realización de entrevistas.
2. Estudiar el estado del arte.
3. Profundizar en el estudio de la metodología, lenguaje y herramientas a utilizar.
4. Profundizar en el estudio y desarrollo del modelo de negocio y modelo del sistema de la aplicación.
5. Profundizar en el estudio y desarrollo del análisis y diseño de la aplicación.
6. Implementar las funcionalidades críticas del sistema.
7. Elaborar el documento de tesis.

Cronograma de tareas.

Cronograma de Tareas			
No:	Tarea	Responsable	Fecha
1	Identificar las necesidades de la institución a través de la realización de entrevistas.	Aldo Milanés López	20 – 30 Noviembre
2	Estudiar el estado del arte.	Renier Montano García	1 - 12 Diciembre
3	Profundizar en el estudio de la metodología, lenguaje y herramientas a utilizar.	Aldo Milanés López	1/Diciembre 28/Febrero
4	Profundizar en el estudio y desarrollo del modelo de negocio y modelo del sistema de la aplicación.	Renier Montano García	1 – 20 Febrero
5	Profundizar en el estudio y desarrollo del análisis y diseño de la aplicación	Aldo Milanés López	21/ Febrero 1/Marzo
6	Implementar las funcionalidades críticas del sistema.	Aldo Milanés	1/Marzo

		López	31/Marzo
7	Desarrollar el documento de tesis.	Aldo Milanés López Renier Montano García	20/Enero 30/Abril

Métodos Científicos usados en la investigación.

Métodos teóricos.

1. Analítico–sintético: Este método ha servido para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así, extraer los elementos más relacionados e importantes con el objeto de estudio.

Análisis histórico–lógico: Este método ha ayudado a entender el surgimiento y la evolución del tema de la investigación, así como otras temáticas estrechamente relacionadas con la misma a lo largo de la historia de la Informática.

Modelación: Mediante este método se ha podido modelar la realidad futura que será la investigación, los principales elementos que lo componen y su funcionamiento. Debido a que se escoge Proceso Unificado de Desarrollo Software (RUP) como metodología de desarrollo de software, se utilizará Lenguaje Unificado de Modelado (UML) para hacer la modelación teórica del resultado, generándose así varios modelos. Toda esta modelación ayuda a descubrir y estudiar nuevas cualidades y relaciones del objeto de estudio.

Métodos Empíricos.

Observación: Este método es de vital importancia ya que ha permitido percibir a partir de la situación real que se está investigando cómo se desarrolla a groso modo el proceso que constituye el objeto de estudio.

Entrevista: Para el desarrollo de este método se ha entrevistado al Jefe del Grupo de Soporte de Desarrollo de la Infraestructura Productiva, quien ha aportado elementos significativos a la investigación.

Análisis de Documentos: Este método ha sido de gran ayuda porque a partir de un grupo de resoluciones y documentos que exponen con mucho detalle el flujo que sigue el proceso que representa el objeto de la investigación, ha sido más fácil la comprensión del mismo.

Aportes.

Con la realización de este trabajo diploma se espera solventar los problemas existentes en el proceso de Gestión de Tecnología en la Universidad de las Ciencias informáticas, buscado una vía de solución para desarrollar una aplicación informática capaz de automatizar este proceso, brindando seguridad para la información, fácil manejo del sistema, rapidez en la generación de los datos y por tanto, que el proceso que constituye objeto de estudio se torne menos engorroso y más rápido, con vista de lograr una optimización y eficiencia en la distribución de los recursos tecnológicos por parte de la Vicerrectoría de Tecnología de la Universidad de las Ciencias Informáticas.

CAPITULO 1. Fundamentación Teórica.

1.1 Introducción

En este capítulo se abordará en detalle todo lo relacionado con la fundamentación teórica, se hará un estudio

del estado del arte tanto en el mundo, en nuestro país y específicamente en nuestra universidad. Se especificarán algunos conceptos básicos asociados a la investigación. Se profundizará además en el Objeto de Estudio y la Situación Problemática haciéndose una descripción detallada. Se abordará detalladamente todo lo relacionado con las Tecnologías de la Información y las Comunicaciones (TIC), profundizando en el uso de cada herramienta a utilizar. También se analizarán las metodologías de desarrollo y las Herramientas Case. A partir de todos estos elementos se desarrollará la investigación, haciéndose más sencilla la comprensión de temas que serán tratados en lo adelante.

1.2 Conceptos esenciales.

Para una mayor comprensión de los temas a tratar es importante conocer un grupo de conceptos asociados al dominio del problema.

1.2.1 ¿Qué es tecnología?

Es el conjunto de saberes, destrezas y medios necesarios para llegar a un fin predeterminado mediante el uso de objetos artificiales o artefactos; en el dominio del problema el término tecnología será utilizado específicamente para referirse a los objetos o artefactos.

1.2.2 ¿Qué es Tecnología Informática?

Es el estudio, diseño, desarrollo, puesta en práctica, ayuda o gerencia de los sistemas informáticos computarizados, particularmente usos del software y hardware.

1.2.3 ¿Qué son las Tecnologías de la Información y las Comunicaciones (TIC)?

Son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario, y está destinada a optimizar la comunicación humana.

1.2.4 ¿Qué es gestión de la tecnología?

Es el mecanismo que permite administrar el proceso de distribución de la tecnología entre los diferentes clientes.

1.2.5 ¿Qué es un Cliente?

Es la persona jurídica que solicita un lote determinado de tecnología, en el dominio del problema es una de las estructuras de dirección o productiva que realiza pedidos a la Dirección de tecnología.

1.3 Estado del Arte.

1.3.1 Elementos de la Gestión Tecnológica en el mundo.

El avance tecnológico es hoy uno de los principales medidores del poderío y desarrollo a nivel mundial. Las tecnologías de la información particularmente, aseguran a quienes las posean y tengan la capacidad intelectual de manejarlas, el dominio de los medios de comunicación, de la industria de los servicios informáticos y como consecuencia del equilibrio de todas las demás ramas del desarrollo humano que hoy día depende de estas tecnologías. Muchos países se encuentran en la vanguardia en este sentido debido a que cuentan con los recursos financieros suficientes que les han permitido el desarrollo tecnológico a alto nivel así como la obtención de personal calificado para su manejo.

La Asociación Latino-Iberoamericana de Gestión Tecnológica (ALTEC) es una institución privada internacional no lucrativa, creada en 1984 con el propósito de vincular a personas naturales y jurídicas activas en la reflexión y ejercicio de la Gestión Tecnológica, para realizar actividades de cooperación en esta área. ALTEC está integrada por un grupo significativo de profesionales de la región, interesados en la Gestión Tecnológica, entre los que se incluyen investigadores, profesores universitarios, empresarios y profesionales de las empresas, especialistas gubernamentales, expertos de agencias financieras y de cooperación y profesionales que se desempeñan en consultoras privadas. La misión principal de ALTEC es promover el desarrollo socioeconómico de la región Iberoamericana a través del perfeccionamiento continuo de la Gestión Tecnológica.

ALTEC contribuye además a generar un pensamiento Latino-Iberoamericano sobre problemas relacionados con la innovación, que influya en el desarrollo de políticas de ciencia y tecnología y que fomente la innovación e instrumentación de proyectos de desarrollo tecnológico en los países de la región. ^[4]

Las tendencias actuales de los software de gestión están encaminadas a la implementación de sistemas sobre CMSs (Content Management System) ya existentes, de manera que sea mas rápida el desarrollo de los mismos teniendo la posibilidad de reutilizar este tipo de sistemas gestores de contenido. Son numerosas también las aplicaciones de gestión realizadas con la utilización de frameworks, lo que permite la implementación de estos sistemas sobre la base, estructura y organización del framework que se esté utilizando. Esencialmente, el desarrollo actual de aplicaciones de gestión se basa en la implementación de sistemas sobre plataformas libres, así como de la utilización de tecnologías de desarrollo web –como las antes mencionadas- dirigidas a la construcción rápida y eficiente de tales aplicaciones.

La mayoría de los software de gestión que existen en el mundo están dirigidos a la gestión documental y de información, no a la gestión de recursos tecnológicos como tal, lo que nos acorta la posibilidad de encontrar una aplicación que sea reutilizable para poder dar solución al problema que nos acoge, de ahí la novedad de esta investigación y necesidad de desarrollar un sistema óptimo que sea capaz de realizar el proceso que constituye objeto de la investigación.

1.3.2 Elementos de la Gestión Tecnológica en Cuba.

En nuestro país contamos con un gobierno con políticas de avance en esferas de ciencia, tecnología y educación, un amplio mercado nacional de tecnología y además las empresas que tienen clientes y competidores en todo el contexto nacional y mundial que deben tomar una serie de decisiones sobre gestión tecnológica.

Se trata entonces de elevar la posición competitiva por medio de diversos mecanismos que, en su mayoría, deben encontrarse por las propias empresas, donde la gestión de recursos tecnológicos puede desempeñar una función decisiva.

La primera computadora que se introdujo en Cuba fue una IBM RAMAC, en los meses finales de 1958. Con el Triunfo de la Revolución y la eliminación del sistema de contabilidad financiera en los primeros momentos, el uso de las tecnologías de la informática se volvió poco necesario, aunque dentro del proceso de centralización de la economía se creó Empresa Nacional de Procesamiento de Datos en el año 1963, la que concentró muchos equipos de tabulación de tarjetas perforadas y de saldo directo.

Gracias a la adquisición de nuevos modelos de computadoras, como la Iris 50, se logró comenzar la enseñanza en los altos centros de estudio como La Universidad de La Habana y el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE). Debido a este desarrollo de la enseñanza se logró desarrollar una computadora que fuera completamente cubana. A partir del año 1985 se comenzó a buscar tecnologías e información en otros lugares con el objetivo de actualizar los conocimientos que se tenían por esa época. Pero esta rápida transformación ocasionó que se despreciasen todos los avances de las maquinas cubanas y no se continuase con su desarrollo, acción que retrasó considerablemente el avance de la informática dentro de la sociedad cubana. Con la llegada de los años 90 se volvió presente el dominio de la firma Microsoft en cuanto a las importaciones de software, y con el uso de la Internet se comenzó el desarrollo de redes informáticas como la de INFOMED dedicada a los especialistas de la salud en el país. Gracias a los fuertes software de Microsoft (como la hoja electrónica EXCEL, el procesador de textos WORD y el sistema de gestión de bases de datos relacionales ACCESS.) se pudo difundir aún más la informática dentro de la sociedad cubana. Además de comenzar las carreras universitarias dedicadas especialmente a la informática (Cibernética e Ingenieros en Computación) se abrieron carreras de Ingenieros en Sistemas.

Con el lanzamiento en 1995 de la versión Windows 95 por parte de la Microsoft, comenzó una nueva etapa dentro de la informatización de la sociedad cubana. Gracias a las ventajas que este software proporcionaba comenzó un rápido proceso de informatización en el proceso de producción dentro de la economía y se comenzó a trabajar en los entornos Web, gracias a la extensión de la Fibra Óptica Nacional. Además se comenzó a impartir clases en todas las instituciones estudiantiles desde los más bajos niveles. Se creó el Ministerio de la Informática y las Comunicaciones, institución que se encargaría de enfocar de manera positiva el desarrollo de esta ciencia en nuestro país. Dentro de las ramas de la economía que asimilaron de manera más eficiente la aplicación de estas nuevas tecnologías cabe destacar las empresas del Ministerio de la Industria Básica, las empresas hoteleras y ETECSA. En la producción de software lo más destacado fue la creación de la empresa Segurmática, que desarrollo el antivirus Sav de gran eficacia contra los virus de origen cubanos.

En el año 1987 como una iniciativa de llevar la informática a los lugares más apartados del país se crean los Joven Club de Computación tanto en las ciudades como en las zonas rurales, por lo cual este servicio es asequible para casi la totalidad de la población cubana y se garantiza de esta manera la obtención de los conocimientos más básicos relacionados con la informática y la electrónica. Existen en estos momentos más de 600 instalaciones de este tipo en el país. Su potencial tecnológico incluye más de 6000 computadoras, unido a medios de impresión, digitalización de imágenes, almacenamiento y reproducción de grandes volúmenes de información.

En 2001 se crea la Universidad de las Ciencias informáticas como centro de estudios y además como organismo productor de software para el desarrollo de la informática en Cuba. Ya con todos estos avances en las tecnologías de la información, el país se vio en la obligación de invertir mayor cantidad de capital para la obtención de equipamiento informático de última generación, y por tanto nace la necesidad de tener un control de los recursos tecnológicos que se le iban asignando a cada una de las empresas y organismos necesitados de esta tecnología. El Ministerio de la Informática y las Comunicaciones es el organismo rector de las Tecnologías de la Información en el país, es quien se encarga de regular y promover el desarrollo productivo de las distintas empresas, dentro de ellas, Copextel, es la gerencia que se encarga de la distribución e instalación del equipamiento tecnológico necesario para el desarrollo y producción de las mismas.

Acciones ejecutadas en materia de dirección de la ciencia y la tecnología

La introducción del término gestión tecnológica en Cuba es muy reciente. Su uso creciente se produce a partir del año 1993, luego de celebrarse, en La Habana, el primer encuentro nacional de esta temática. Pero tan importante o más que el aspecto terminológico es lo relacionado con el contenido del concepto. Por una parte, podría argumentarse que aún está en pleno desarrollo; por otra, se puede sostener la idea y ejemplificar el hecho de que en Cuba se hace gestión tecnológica desde hace muchos años. En nuestra opinión, los dos puntos de vista no son tan excluyentes como pudiera parecer a primera vista. El desarrollo científico y tecnológico logrado en las últimas tres décadas y la diversificación de la producción en prácticamente todos los sectores de la economía, no dejan dudas acerca de todo el conjunto de acciones ejecutadas en materia de dirección de la ciencia y la tecnología en Cuba, con impactos totalmente visibles en importantes indicadores económicos y sociales.

¿Cómo se están entendiendo y asimilando estos conceptos?

En primer lugar, los estudiosos del asunto han puesto menos su atención en la discusión de los problemas estrictamente conceptuales de alcance más académico que en los de orden práctico, y no porque los primeros se consideren de menor importancia, sino por lo imperativo de los tiempos y circunstancias en que vive el país. La contribución cubana al perfeccionamiento de estas herramientas y buenas prácticas en el manejo de las variables tecnológicas puede estar, sobre todo, en la validación y asimilación creativa de la riqueza enorme de ideas y principios teóricos que se están desarrollando en Iberoamérica y en el mundo. Podrá ser esto un producto valioso, dirigido especialmente a los países en desarrollo. Cuba trata de llevar a

cabo, dentro de su singular modelo de desarrollo, la asimilación y validación de lo que se está produciendo en este campo, con la participación consciente y mancomunada de productores, investigadores, interfases y otros actores. Podría considerarse casi un caso curioso, pero muy positivo, el hecho de que los representantes más caracterizados de la comunidad de productores de bienes y servicios dominan el tema. Es decir, el que los productores están aprendiendo sobre esta cuestión tan rápidamente como lo hace el sector de I+D(Investigación y Desarrollo). Este hecho singular, no muy repetido en los países en vías de desarrollo del área latinoamericana, es significativo respecto a las potencialidades de los ejecutivos y organizaciones innovadoras cubanas y explica porqué especialistas de otros países se motivan a participar en lo que es ya prácticamente el proyecto de la gestión tecnológica en Cuba.

La incipiente red de centros de gestión tecnológica incluye al Centro de Gerencia de Ciencia y Tecnología, (GECYT), de la Agencia de Ciencia y Tecnología en La Habana; a Gestiones Mercurio (GESMER), en Santa Clara; al Centro de Información, Gestión y Consultoría (CIGEC), de Ciego de Avila; al Centro de Gestión Tecnológica (CEGETEC), de Camagüey; el Centro de Innovación Tecnológica e Intensificación de Procesos (CITIP), de Las Tunas; el Centro de Información y Gestión de Ciencia y Tecnología (MEGACEN), de Santiago de Cuba y a núcleos de gestión tecnológica en las provincias de Guantánamo (GECIEN) y Bayamo (GEDES), y a los de las provincias de Holguín, Ciego de Avila, Sancti Spíritus, Villa Clara y Pinar del Río, además de otros en formación. En la Fig. 1 se muestra la distribución geográfica de estas entidades.

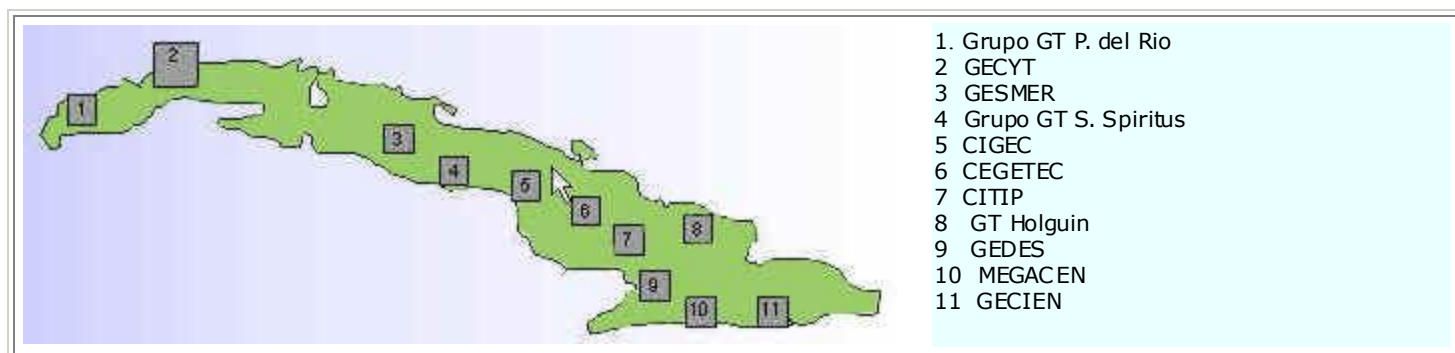


Fig. 1 Red de Centros de gestión tecnológica en Cuba.

1.3.3 Elementos de la Gestión Tecnológica en la Universidad de las Ciencias informáticas.

La Universidad de las Ciencias Informáticas es la principal consumidora de esta tecnología en el país, ya que como centro de altos estudios en Ciencias Informáticas requiere de equipamiento suficiente para el buen desarrollo del programa docente y del aprendizaje de los estudiantes, pero a la vez como organismo productor de software, necesita una basta infraestructura productiva que permita la creación de productos informáticos de calidad. Acompañado a eso, están las diferentes estructuras de dirección del centro que requieren también de recursos para el manejo de información así como para el mejor desarrollo de los flujos de trabajo de la entidad. Dentro de estas estructuras se encuentra la Dirección de Tecnología de la universidad, que es la Oficina encargada de la asignación de todo el equipamiento informático antes mencionado y por tanto, la máxima responsable del proceso de Gestión de la Tecnología en la Universidad de

las Ciencias Informáticas. Desde los inicios de la Universidad en 2001, esta oficina, se encarga de la regulación y distribución de tecnología para los diferentes sectores de la universidad.

Desde los primeros momentos que comenzó a desplegarse la tecnología por las diferentes áreas, fueron surgiendo una serie de documentos, planillas y procedimiento que conformaron lo que se conoce hoy como proceso de Gestión Tecnológica. Dicho proceso genera una considerable cantidad de información de alta importancia y que actualmente se almacena en planillas (documentos Word) y bases de datos en Access, además de que se maneja de forma manual, lo que hace que el proceso se torne engorroso, tardío y exista poca seguridad en la manipulación de la información resultante. Hasta ahora no existe en la Universidad de las Ciencias Informáticas, ni en nuestro país, ninguna aplicación o proceso automatizado que realice la Gestión Tecnológica, y por tanto, capaz de optimizar y distribuir los recursos tecnológicos. A continuación se hará una descripción más detallada de la situación problemática en cuestión, a fin de que se comprenda el proceso mencionado y el objetivo fundamental de la investigación.

1.4 Situación Problemática.

Las entidades dedicadas fundamentalmente a la producción de software no sólo deben disponer de las tecnologías adecuadas que permitan su desarrollo, sino además disponer de los procesos de gestión adecuados para su uso, distribución, optimización y mejora continua.

La gestión de la tecnología es una poderosa herramienta que se debe enmarcar dentro de los procesos generales de perfeccionamiento al que están sometidas todas las empresas. Cada vez en mayor medida, el control del recurso tecnológico proporciona una ventaja competitiva a las organizaciones, sobre todo en aquellas en las que se integra en la estrategia general de la propia organización. Y esto es mucho más importante para el caso de organizaciones dedicadas a la generación de productos o servicios en *sectores de alta tecnología* en las que el periodo de validez de una tecnología concreta (en términos de adecuación y rendimiento comparativo con otras competidoras) es cada vez más reducido (ciclos de producto más cortos). Es por esta razón que las empresas deben contar con sistemas informáticos que les permitan la gestión de las tecnologías que poseen de manera más rápida y eficiente.

Actualmente en la Universidad de las Ciencias Informáticas, como entidad productora de software, se lleva a cabo el proceso de Gestión de la Tecnología para administrar y distribuir el equipamiento adquirido entre las diferentes estructuras de dirección y producción. El Director de Tecnología de la universidad es quien recibe los pedidos de tecnología de las diferentes estructuras mediante una planilla existente para el procedimiento. Si la universidad cuenta con los recursos para atender la solicitud, la tecnología es distribuida por parte de los compañeros de Copextel, encargados de la instalación de la misma. Si por el contrario, la universidad no cuenta con los recursos, el pedido pasa al Departamento de Economía para evaluar el costo de la compra de la tecnología, y es autorizada por el Vicerrector de Tecnología y posteriormente por el Vicerrector de Producción para luego ser asignada al cliente. El procesamiento de la información que se genera durante este proceso se realiza de forma manual, lo que supone una alta complejidad y la aparición de tareas engorrosas que deben realizarse para mantener actualizada, consultar y/o modificar dicha información. Además de que está expuesta a desastres naturales lo que representa un alto riesgo e incrementa la necesidad de poseer una aplicación informática que permita gestionar todos los datos que se manejan y generan durante la

realización del proceso, y que optimice el desempeño de la Dirección de Tecnología. A continuación se muestra un ejemplo de la planilla para solicitar tecnología.

SOLICITUD DE TECNOLOGÍA

FORMULARIO DE SOLICITUD PARA LOS PROYECTOS PRODUCTIVOS
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Día	Mes	Año

DATOS GENERALES DEL PROYECTO PRODUCTIVO	
No. Facultad	
No. Solicitud	
Fecha de Solicitud	
Nombre Asesor de Arquitectura y Tecnologías:	
Nombre Líder Proyecto Productivo	

DATOS DE LA SOLICITUD DE TECNOLOGÍA		
Hardware y Periféricos	Código Proyecto	Cant. aprobada
Modem		
Memoria RAM		
256 MB		
512 MB		
1 GB		
HDD (Disco Duro)		
Motherboard		
Tarjeta de Video		
Tarjeta de Captura		
Teclado		
Mouse		
Quemador		
CD		
DVD		
CD/DVD		
Punto de acceso (red)		
Cable de red		
Switch		
Adaptador de red		
Memoria extendida		
Tarjeta de red		
Tarjeta de Sonido		
Otros (especificar)		

Observaciones (Justificación)

NOMENCLATURA PROYECTO PRODUCTIVO	
Código	Nombre

Solicitado por:	Aprobado por Director de Producción o ViceDecano de Producción.	Aprobado por Consejo Técnico de Producción.
-----------------	---	---

Nombre y Apellidos	Nombre y Apellidos	Nombre y Apellidos
Cargo	Firma y Cuño	Firma y Cuño
Firma		

la posibilidad de utilizar soluciones existentes como apoyo o complemento es casi nula, no obstante a continuación se hará un breve análisis sobre este tema.

1.5 Soluciones existentes.

Muchos países, empresas y compañías utilizan sistemas informáticos que faciliten las actividades que deben realizar y que contribuyan a la optimización de los resultados y mejor aprovechamiento de los recursos, buscando mayor eficiencia en los procesos que realizan y mejores resultados en la producción.

Hasta el momento no se tiene conocimiento de sistemas que tributen al mejoramiento de los procesos que constituyen el objeto de estudio en ningún país y en Cuba tampoco. Actualmente en la Universidad de las Ciencias Informáticas existe una aplicación, en la infraestructura de Soporte Técnico, específicamente en el Nodo Nivel 1, que facilita la gestión de los reportes de solicitud de mantenimiento en la universidad. Además la aplicación está enfocada sólo a los servicios técnicos que se prestan a las distintas estructuras y básicamente a la gestión de los reportes que se derivan de esa actividad, no al control de la distribución de los recursos, o sea que no se tiene una base de datos o inventario de la tecnología existente en las áreas ni tampoco controla la asignación de tecnología, sino la reparación y mantenimiento de la misma. A continuación se muestran imágenes de esta aplicación y específicamente del módulo de solicitud de mantenimiento:



Figura 2. Imagen de la página de Inicio de la aplicación GATSRVER.



Figura 3. Imagen de la página de Solicitud de Mantenimiento de la aplicación GATSRVER.

Como se puede apreciar esta aplicación no es factible puesto que no cumple con los requisitos necesarios que exige el proceso a automatizar. No obstante es válido analizar algunos sistemas de gestión, que pudiesen servir como base o complemento a la solución que se propone en esta investigación.

Tal es el caso de EUROWIN, una solución de gestión de empresa que permite centralizar y compartir información con un sistema parecido a las páginas web de internet, pero dentro de la empresa. ^[5]

El sistema está pensado para que la información de la empresa esté a disposición de todos los trabajadores (siempre que tengan los permisos adecuados para acceder a ella) a través de una serie de módulos con los que cuenta el programa:

Gestión de Noticias

A través de este módulo, los trabajadores de la empresa pueden consultar noticias, comunicados y avisos que vayan surgiendo dentro de la propia empresa

Centralización de documentos

Mediante este módulo, los usuarios podrán tener acceso y descargarse a su ordenador los documentos y ficheros que se publiquen en la intranet, documentos de word, excel, pdf, fotografías, etc...

Gestión de datos de los trabajadores

El programa también permite el acceso a las fichas de datos de los trabajadores para poder consultar su teléfono, e-mail, dirección o cualquier otro tipo de dato.

Gestión de vacaciones, enfermedad y permisos del personal

A través de este módulo, se pueden gestionar y consultar las vacaciones, bajas y permisos de los empleados, así como los días de vacaciones consumidos, el periodo de vacaciones, días de baja, días de asuntos propios, etc..

Mensajería Interna

El programa tiene un sistema muy potente de mensajería vía correo electrónico que se puede consultar desde cualquier lugar, así, el trabajador puede consultar su correo electrónico a cualquier hora, desde cualquier lugar, además de poder enviar nuevos mensajes o responder a los mensajes recibidos.

Los beneficios del programa están claros:

- Optimiza los recursos humanos y económicos que se destinan a la gestión del personal.
- Favorece el flujo de información dentro de la empresa.
- La administración de seguridad, permite implantar una política de seguridad para que cada trabajador sólo tenga acceso al recurso de la intranet que necesita y a ningún otro.

- Permite un acceso rápido y ágil a cualquier información corporativa.

Otro software que permite gran número de funcionalidades enfocadas a la gestión de la información es la Plataforma de Acceso a Información de ZyIMAGE (IAP), que ofrece a las organizaciones la mejor gama de herramientas para archivar, buscar, gestionar y compartir información electrónica y en papel, así como mensajes de correo electrónico y multimedia. ZyLAB lleva desarrollando desde hace dos décadas tecnologías galardonadas que mejoran la eficacia global de los gobiernos locales y federales, organismos de investigación y compañías de investigación exhaustiva, y ayudan a las organizaciones comerciales, como las del sector financiero e hipotecario, a alcanzar la conformidad normativa. 700.000 clientes en todo el mundo confían en la plataforma IAP basada en XML, para archivar, buscar, gestionar y compartir millones de páginas de archivos electrónicos y documentos en digitalizados, multimedia y mensajes de correo electrónico y archivos adjuntos.

[6]

La construcción abierta y modular de ZyIMAGE IAP permite que las soluciones se puedan configurar para adaptarse a los presupuestos y necesidades reales de nuestros clientes, desde las instalaciones más básicas, a las soluciones empresariales exhaustivas. Las herramientas innovadoras están disponibles para el flujo de trabajo, e-Discovery, gestión de contratos, gestión de documentos, gestión de registros, y mucho más.

A pesar de que estos sistemas pueden resultar altamente eficientes en empresas que lo utilicen, no son factibles como parte de la solución que se desea obtener, ya que son sistemas hechos a la medida, y su reusabilidad está limitada por las especificidades propias del software. Se puede decir que es poco factible adquirir un sistema ya desarrollado para otro esquema que habría que personalizar totalmente de acuerdo a las características de la institución, además de que los existentes están concebidos como propietarios y el costo de las licencias es elevado además que difíciles de adquirir por nuestro país debido a las imposiciones del embargo económico que se ha extendido ya al campo de la informática. Es por ello que a pesar de la existencia de disímiles sistemas con funcionalidades altamente eficientes dentro del marco que fueron desarrollados y en muchos de los casos con gran capacidad de reutilización como es el caso de los sistemas antes mencionados (ZyIMAGE IAP, EUROWIN, etc.), su uso no es extensible al campo de acción de esta investigación.

1.6 Tecnología y Herramientas.

1.6.1 Tecnologías de la Información y las Comunicaciones (TIC).

Puede definirse a las Tecnologías de la Información y las Comunicaciones como al conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario, y está destinado a optimizar la comunicación humana.

La introducción masiva de las nuevas Tecnologías de la Información y las Comunicaciones conocidas por las siglas TIC, está influenciando fuertemente la estructura y dinámica de los procesos económicos y sociales, redefiniendo aceleradamente las formas de producir, vender y competir, en prácticamente todos los sectores productivos de bienes y servicios; y también, en las nuevas formas de educar, de interacción y comunicación entre la gente y organismos de la sociedad, y de comunicar y controlar las acciones de gobierno.^[17] Las Tecnologías de la Información y las Comunicaciones están transformando profundamente las economías doméstica y mundial, pues permiten generar riqueza a distancia mediante la red, superando las fronteras geográficas y políticas. La nueva economía no está centrada principalmente en los recursos naturales, que por cierto cada día son más importantes en el contexto de la creciente demanda mundial, sino en los flujos electrónicos de información y conocimiento.^[18] Pero el valor diferencial, en términos de productividad y competitividad, no está ceñido a lo que las computadoras pueden producir, sino a la creatividad del hombre que motiva la innovación sobre los usos de estas herramientas. Nadie podrá permanecer al margen de este proceso revolucionario que supone la incorporación masiva de estas tecnologías en la sociedad, puesto que los beneficios que aportan son por demás relevantes: (i) eliminan las barreras de tiempo y espacio y facilitan las comunicaciones; (ii) Internet, como herramienta estándar de comunicación, permite un acceso igualitario a la información y conocimiento; favorece la cooperación y colaboración entre distintos actores, aumenta la producción de bienes y servicios de valor agregado; y potencialmente, eleva la calidad de vida de los individuos.

1.6.1.1 Sistemas de Gestión.

Las empresas, independientemente de su tamaño, enfrentan demandas respecto a rentabilidad, calidad, tecnología y desarrollo sostenible. Un sistema de gestión eficiente, diseñado a la medida de sus procesos comerciales, puede ayudar a enfrentar los desafíos del cambiante mercado global de hoy. Para convertir las presiones de la competencia en ventajas comparativas, las empresas deben aumentar el rendimiento operativo en forma sistemática. Un sistema de gestión puede ayudar a centrar, organizar y sistematizar los procesos para la gestión y mejora.^[19]

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización.

La implementación de un sistema de gestión eficaz puede ayudar a:

- Gestionar la información, los riesgos sociales, medioambientales y financieros.
- Mejorar la efectividad operativa.

- Reducir costos.
- Aumentar la satisfacción de clientes y partes interesadas.
- Proteger la marca y la reputación.
- Lograr mejoras continuas.
- Potenciar la innovación.
- Eliminar las barreras de comercio.
- Aportar claridad al mercado.

1.6.1.1.1 Sistemas de Gestión con plataforma Web.

Los sistemas de gestión con plataforma Web ofrecen diversas ventajas que los hacen factibles para el manejo de la información. Las aplicaciones Web utilizan lo que se conoce como clientes livianos los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma. Desde el punto de vista de la arquitectura, esta basada en la tecnología cliente-servidor. Del lado del cliente se encuentra el usuario final utilizando la aplicación por intermedio de un navegador (Explorer, Moxilla, Firefox, etc.), y en el lado del servidor es donde residen realmente los datos, reglas y lógica de la aplicación. ^[10]

Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente menos costosas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

Con la división en dos partes, se logra centralizar la administración general a un solo lado: el servidor, resolviendo una gran cantidad de problemas anteriormente encontrados en las aplicaciones de escritorio mono usuario, como son:

- Diseminación de la información y lógica en muchas partes (cada computador que la use).
- Duplicidad de datos por la falta de unificación de los mismos.
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada no es óptima.
- Dificultad para configurar cada una de las instalaciones dependiendo de las necesidades de cada usuario.

Posterior al análisis de las características principales de las aplicaciones Web y de escritorio, se decide implementar la solución de la presente investigación en un ambiente de desarrollo Web, ya que es la plataforma óptima que sirve de infraestructura para todos los usuarios desde el tipo empresarial, institucional

o personal.

1.6.2 Servidores Web.

Un servidor web es un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (Hypertext Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. ^[11]

Un servidor web se mantiene a la espera de peticiones HTTP por parte de un cliente HTTP. El cliente realiza una petición al servidor y éste le responde con un código HTML en el que se encuentra el contenido (página web) que el cliente solicita. El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

A continuación se expondrán algunas de las características principales de los Servidores Web HTTP Apache y del Internet Information Services (IIS), en vista a seleccionar el que se utilizará para el desarrollo de la aplicación.

1.6.2.1 Servidor HTTP Apache.

El servidor HTTP Apache es un software (libre) de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo. ^[12]

Apache posee una estructura de módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo mismo.

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

1.6.2.2 Internet Information Services (IIS).

Internet Information Services, IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP

Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. ^[13]

Este servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

1.6.3 Lenguajes de Programación Web.

Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas "comunicarse" con las computadoras. Existen por lo menos varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. ^[14]

A continuación se expondrán algunas de las características de lenguajes como PHP (del lado del servidor) y JavaScript (del lado del cliente), con la intención de aportar elementos que permitan comprender mejor la selección.

1.6.3.1 Lenguajes del lado del servidor.

PHP

PHP es un lenguaje de programación utilizado para la creación de sitios web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor", (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

Es un lenguaje de script interpretado en el lado del servidor, utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. ^[15]

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

ASP.NET

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzado al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba su antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o Java. Para su funcionamiento se necesita tener instalado IIS con el Framework .Net. Microsoft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores. ^[16]

Ventajas:

- Completamente orientado a objetos.
- Controles de usuario y personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor seguridad.

1.6.3.2 Lenguajes de Programación del lado del cliente.

JavaScript.

Este es un lenguaje interpretado que no requiere compilación. Utilizado principalmente en páginas web permitiendo desarrollar las funcionalidades de las aplicaciones mediante eventos. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento). ^[17]

Las principales ventajas de este lenguaje es que es de scripting seguro y fiable, los script tienen capacidades limitadas, por razones de seguridad, y que el código Javascript se ejecuta directamente en el cliente.

XHTML

XHTML (Extensible HyperText Markup Language) es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0 (HyperText Markup Language). Los tipos de documentos de la

familia XHTML están basados en XML (Extensible Markup Language), y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML. ^[18]

XHTML 1.0 (esta especificación) es el primer tipo de documento de la familia XHTML. Es una reformulación de las tres definiciones de tipo de documento HTML 4.0 como aplicaciones de XML 1.0. Su finalidad es ser usado como lenguaje de contenidos que es a la vez conforme a XML y, si se siguen algunas sencillas directrices, funciona en aplicaciones de usuario conformes con HTML 4.0. Los desarrolladores que migren aplicaciones hacia XHTML 1.0 apreciarán las siguientes mejoras:

- Los documentos XHTML son conformes a XML. Como tales, son fácilmente visualizados, editados y validados con herramientas XML estándar.
- Los documentos XHTML pueden escribirse para que funcionen igual o mejor que lo hacían antes tanto en las aplicaciones de usuario conformes a HTML 4.0 como en las nuevas aplicaciones conformes a XHTML 1.0.
- Los documentos XHTML pueden usar aplicaciones (scripts y applets) que se basen ya sea en el Modelo del Objeto Documento de HTML o XML[DOM].
- A medida que la familia XHTML evolucione, los documentos conformes a XHTML 1.0 estarán más preparados para interactuar dentro y entre distintos entornos XHTML.

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. ^[19]

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML.

1.6.4 Metodologías de Desarrollo de Software.

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen sin éxito. El objetivo de un proceso de desarrollo es elevar la calidad del software (en todas las fases por las que pasa) a través de

una mayor transparencia y control sobre el proceso.

La implantación de un proceso de desarrollo es una labor más a medio-largo plazo que una labor de resultados inmediatos hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo, incluso a cada proyecto.

Para tener un proceso de producción de software con el menor número de fallos, adecuado a las necesidades del cliente y entregar a tiempo el producto, la producción de software debe convertirse en un proceso disciplinado.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, se han desarrollado dos corrientes en lo referente a los procesos de desarrollo: los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los primeros intentan conseguir el objetivo común por medio de orden y documentación, los segundos (también llamados métodos ágiles) tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen el proceso. ^[20]

A continuación se detallarán algunas características de las Metodologías RUP y XP, a razón de que hoy día son de las más utilizadas, en vista de una mayor comprensión de la selección realizada.

1.6.4.1 RUP

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

RUP está basado en 5 principios clave que son: adaptar el proceso, balancear prioridades, demostrar valor iterativamente, elevar el nivel de abstracción y enfocarse en la calidad.

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases: Inicio, Elaboración, Construcción y Transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor énfasis en las distintas actividades. Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base-line (Línea Base) de la arquitectura. En la fase de Construcción, se construye el producto por medio de una serie de iteraciones. En la fase de Transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios. ^[21]

Principales características de RUP:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.

- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

1.6.4.2 Programación extrema (XP).

La programación extrema o eXtreme Programming (XP) es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. ^[22]

Los principios básicos de la programación extrema son: Simplicidad, Comunicación, Retroalimentación (feedback).

Las características fundamentales del método son:

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir una nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento.
- Propiedad del código compartida.
- Simplicidad en el código.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

1.6.5 Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. ^[23]

Se puede aplicar en el desarrollo de software integrando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

En UML 2.0 hay 13 tipos diferentes de diagramas contenidos en grupo dentro de los: Diagramas de Estructura, Diagramas de Comportamiento y Diagramas de Interacción, los que permiten realizar un detallado y organizado diseño de cada uno de los procesos, entidades y artefactos componentes del software.

1.6.6 Herramientas.

Las herramientas son esenciales para el análisis de sistemas. Ellas mejoran la forma en que ocurre el desarrollo y tiene influencia sobre la calidad del resultado final. ^[24] Tanto el proceso de desarrollo de sistemas como el producto que se obtiene con él pueden mejorarse con el uso de herramientas apropiadas.

Las herramientas sugieren procedimientos que conducen al empleo de procesos más eficientes. Si la productividad significa realizar la tarea correcta, la eficiencia significa hacer esta tarea en forma correcta. Las herramientas pueden sugerir la mejor forma para abordar una tarea. Por esto la importancia de seleccionar las mejores de acuerdo con el producto que se vaya a desarrollar y al equipo de trabajo, por lo que a continuación se expondrán las características fundamentales de las posibles herramientas a utilizar.

1.6.6.1 Sistema Gestores de Bases de Datos.

Los sistemas de gestión de base de datos (SGBD); (en inglés: DataBase Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. ^[25]

El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante.

Existen distintos objetivos que deben cumplir los SGBD como son: abstracción de la información, independencia, consistencia, seguridad, integridad, respaldo, un buen manejo de transacciones y un adecuado tiempo de respuesta.

1.6.6.1.1 PostgreSQL

PostgreSQL, también llamado Postgres, es un Sistema de Administración de Bases de Datos de Objetos Relacionales. Pertenece al movimiento de software libre y fue lanzado bajo la licencia BSD (Berkeley Software Distribution).

Algunas de sus principales características son, entre otras: alta concurrencia y una amplia variedad de tipos nativos. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indizables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

1.6.6.1.2 MySQL.

Es un sistema de administración de bases de datos, multi - hilo, multiusuario, con más de seis millones de instalaciones por todo el planeta. MySQL es propietario y patrocinado a partir de la versión 5 por la compañía Sun Microsystems que recientemente fue comprada por Oracle.

MySQL es muy utilizado en aplicaciones web como, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. ^[26] MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

1.6.6.2 Herramienta CASE.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos

pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.^[27]

Las Herramientas CASE tienen como objetivo:

1. Mejorar la productividad en el desarrollo y mantenimiento del software.
2. Aumentar la calidad del software.
3. Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
4. Mejorar la planificación de un proyecto.
5. Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
6. Automatizar el desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
7. Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
8. Gestión global en todas las fases de desarrollo de software con una misma herramienta.
9. Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

1.6.6.2.1 Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue.

El Modelado Visual es el proceso que permite representar gráficamente el sistema, permitiendo resaltar los detalles más importantes.

Visual Paradigm es una Herramienta CASE que da soporte al modelado visual con UML 2.0 como mínimo, entre sus principales características se encuentran:

- Entorno de creación de diagramas para UML 2.0
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo lo 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.

1.6.6.2.2 Rational Rose.

Rational Rose permite que los diseñadores aprovechen del desarrollo iterativo (a veces llamado desarrollo evolutivo) ya que la salida de una iteración se convierte en la entrada al siguiente. De esta manera el desarrollador puede entender cómo los componentes obran recíprocamente y hacen modificaciones en el

diseño. Rational se enmarca en el desarrollo de la “ingeniería ida-vuelta”, yendo detrás y delante de forma que se actualice el resto del modelo para asegurar que el código sigue siendo constante.

Rational ofrece diferentes funcionalidades:

- Diseño dirigido por modelos que redundan en una mayor productividad de los desarrolladores, admitiendo UML, COM, OMT y Booch.
- 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 inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad en múltiples plataformas.

1.6.6.3 Framework.

Un framework, en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. ^[28]

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

No es más que una base de programación que atiende a sus descendientes (manejado de una forma estructural y/o en cascada) posibilitando cualquier respuesta ante las necesidades de sus miembros, o secciones de una aplicación (web, de este modo).

Podemos resaltar que la arquitectura de los frameworks está basada en el modelo MVC (Modelo => Vista => Controlador). Con este apartado podemos controlar el acceso (incluso todo) a nuestra aplicación, esto pueden ser: archivos, scripts o programas; cualquier tipo de información que permita la interfaz. Así, podremos diversificar nuestro contenido de forma dinámica, y estática (a la vez); pues, sólo debemos controlar ciertos aspectos.

El modelo, al igual que el controlador y la vista maneja todos los datos que se relacionen consigo (solo es el proceso medio de la separación por capas que ofrece la arquitectura MVC), y sólo la vista puede demostrar dicha información.

1.6.6.3.1 Framework Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes,

permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

Patrón Arquitectónico.

Symfony está basado en un patrón clásico del diseño Web conocido como arquitectura Modelo – Vista – Controlador (MVC), que está formado por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El

controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

1.6.6.3.2 Prado Framework.

Prado es un framework para PHP basado en componentes y en eventos. La primera versión se realizó para PHP4, pero se reescribió completamente para PHP5. ^[29]

Entre las características que nos ofrece encontramos la separación entre la presentación y la lógica, su arquitectura modular configurable, componentes web, internacionalización y localización, manejo de errores, logs, caché, y mucho más.

Entre sus principales características están:

- Componentes comunes en el diseño web con lógica implementada como TButton, TCheckBox, TTable, TTabPanel, entre otros.
- Desarrollo orientado a componentes con la posibilidad de que los mismos programadores/usuarios los perfeccionen.
- Programación dirigida a eventos. Soporte AJAX.
- Servicio SOAP propio (TSoapService) para aplicaciones que se comunican vía Web Services.
- Manual online en la misma Web con la explicación de cada componente, ejemplos, y sintaxis de los métodos.

1.6.6.4 Entorno Integrado de Desarrollo (IDE).

Un entorno de desarrollo integrado o, en inglés, Integrated Development Environment ('IDE'), es un programa compuesto por un conjunto de herramientas para un programador. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. ^[30]

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, al que mediante pluggins se le puede añadir soporte de lenguajes adicionales.

A continuación se expondrán, las características fundamentales de los entornos de desarrollo integrado: Zend Studio y Eclipse, quienes presentan mayor potencialidad para el trabajo con la Web que otros entornos o

IDEs.

1.6.6.4.1 Zend Studio.

Zend Estudio es un programa de la compañía Zend, orientada a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda, permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. ^[31]

Las características principales del Zend son:

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- phpDocumentor integrado.
- Plegado de código (comentarios, bloques de phpDocumentor, cuerpo de funciones y métodos e implementación de clases).
- Inserción automática de paréntesis y corchetes de cierre.
- Sangrado automático y otras ayudas de formato de código.
- Emparejamiento (matching) de paréntesis y corchetes (si se sitúa el cursor sobre un paréntesis (corchete) de apertura (cierre), Zend Studio localiza el correspondiente paréntesis (corchete) de cierre (apertura)).
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- Cliente FTP integrado.

- Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.

1.6.6.4.2 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). ^[32]

El Entorno de Desarrollo Integrado (IDE) Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente al permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, le permite trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

En cuanto a las aplicaciones clientes, Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, y aplicaciones web.

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Integración con Ant.
- Asistentes (wizards): para creación de proyectos, clases, tests, etc.
- Refactorización.

1.6.7 Justificación de la selección.

Luego de haber fundamentado en las características principales de la tecnología, lenguajes de programación y sistemas gestores de bases de datos candidatos para la implementación de la propuesta de este trabajo se

tuvo como principal elemento a considerar en cada selección la compatibilidad con el sistema operativo Linux, además de que sean herramientas y tecnologías libres, o de código abierto. Se toma esta decisión debido a la inminente migración que se está llevando a cabo hoy en el desarrollo de la informática en Cuba, motivada por las características que presentan, que los hacen factibles para países subdesarrollados como el nuestro, ya que brindan libertad a los usuarios sobre el producto adquirido, dando la posibilidad de ser usado, copiado, estudiado, modificado y redistribuido libremente.

A partir de estas consideraciones se usará PHP como lenguaje de programación del lado del servidor, al ser un lenguaje desarrollado en el ámbito de los sistemas libres, bajo la licencia GNU. PHP es un lenguaje orientado a objetos, simple, elegante y con seguridad en el tratamiento de tipos, y cuenta con las principales funciones para lograr una solución factible al problema en cuestión.

JavaScript, XHTML y CSS fueron los lenguajes de programación seleccionados del lado del cliente, ya que JavaScript nos brinda la posibilidad de una mejor interacción con los usuarios a través del manejo de eventos, XHTML permite estructurar y etiquetar los documentos para su mejor organización y visualización, permitiendo además utilizar objetos tales como imágenes dentro del texto, y CSS facilita crear variados estilos de páginas web, definiendo desde una plantilla CSS la visualización general de los sitios, haciendo las aplicaciones más atractivas y amigables para los usuarios.

Se escogió además la herramienta Eclipse como editor para PHP. La elección se sustenta fundamentalmente en su soporte, posibilidades de depuración y pruebas de PHP con el set más completo de herramientas para la creación de aplicaciones altamente fiables como lo requiere el mercado; en su alto rendimiento y escalabilidad; la seguridad mejorada sobre otras tecnologías web existentes y la posibilidad de hacer despliegue de sistemas desarrollados con esta tecnología en ambiente tanto Windows como Linux, además de ser de código abierto, objetivos específicos de la presente investigación.

Symfony fue el framework seleccionado porque, entre otras razones, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web; proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja; así como automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación y posee como ventaja fundamental, una alta abstracción en el acceso a datos.

Habiendo hecho la selección anterior es conveniente utilizar como Servidor Web, el Apache. Primeramente por su gran potencialidad, y además de que, como se analizó en el epígrafe correspondiente, IIS es propietario, lo que no constituye un aspecto a considerar según los objetivos propuestos. Como sistema de gestión de base de datos, se seleccionó PostgreSQL en su versión 8.1. Es software libre, multiplataforma y contiene todas las características de los SGBD modernos.

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

Luego de haber analizado las características fundamentales de las metodologías de desarrollo RUP y XP se determinó que para un mejor desarrollo del proyecto, era necesario utilizar RUP, ya que, a pesar de que pudiese parecer un proyecto de corto plazo, es decir pequeño, representa un sub-sistema de un proyecto

mayor que necesita de una metodología robusta para su desarrollo, y además de que debe existir uniformidad entre todos los sub-sistemas, por lo que no puede utilizarse una metodología ágil en algunos módulos y una pesada en otros. Además RUP se apoya en el Lenguaje Unificado de Modelado (UML), lo que brinda fortaleza y una mejor perspectiva de lo que se desea lograr.

Con la comparación de las herramientas de modelado CASE se llegó a la conclusión de que se utilizaría Visual Paradigm. Esta herramienta soporta hasta la fecha UML 2.1 completo como mínimo y BPMN, entre otros. La principal razón de esta selección es que es una herramienta libre, multiplataforma.

Luego de haber seleccionado las herramientas y tecnologías necesarias para el desarrollo de la presente investigación, se está en condiciones de hacer una presentación del sistema propuesto.

1.6.8 Conclusiones.

A partir de lo expuesto en este capítulo puede comprenderse con mayor facilidad los subsiguientes temas que se desarrollarán, puesto que representa la fundamentación teórica de la investigación en sí. Se ha hecho un breve análisis sobre el acontecer mundial y específicamente de Cuba y de la Universidad de las Ciencias Informáticas del manejo de la tecnología. En este capítulo se ha descrito el objeto de estudio, así como la situación problemática que da origen al problema científico de esta investigación. Se ha fundamentado además la selección de la tecnología y herramientas a utilizar en el desarrollo del futuro sistema, profundizando en las características principales de cada una de ellas y estableciendo comparaciones entre posibles candidatos que han permitido realizar la elección con mayor efectividad y precisión. Se espera de esta manera, desarrollar un producto con calidad de acuerdo a las posibilidades que brindan las herramientas, metodología y tecnología seleccionadas.

CAPÍTULO 2. Presentación de la solución propuesta.

2.1 Introducción.

En este capítulo se hace una descripción del proceso de negocio, especificando los actores que intervienen en el mismo y el caso de uso que lo representa, especificando además las reglas que definen el proceso. Se hace referencia también a los requisitos funcionales y no funcionales que debe cumplir el sistema que se propone, brindando una concepción general del mismo.

2.2 Modelo de Negocio.

Con el propósito de comprender mejor la dinámica y estructura de la organización en la cual se va a implantar el sistema que se propone en el presente capítulo, a continuación se expondrán los artefactos fundamentales del flujo de trabajo Negocio, perteneciente a la Fase de Inicio, lo que permitirá obtener una visión de la organización, posibilitando definir roles y responsabilidades para el desarrollo del software.

2.2.1 Actores y trabajadores del negocio.

Primeramente se definirán los actores que intervienen en el proceso de negocio que constituye el objeto de la investigación: Solicitar Hardware, conjunto con una descripción detallada del mismo.

➤ Actores:

Actores	Descripción
Usuario.	Persona que interactúa con el sistema y genera solicitudes. (es una generalización de los actores que intervienen en el proceso)
Líder Proyecto.	Persona que interactúa con el sistema y genera solicitudes. Es el individuo que representa el proyecto para el cual se va a realizar la solicitud.
Especialista de IP	Persona que interactúa con el sistema y genera solicitudes.
Asesor Arquitectura Facultad.	Persona que interactúa con el sistema y genera solicitudes. Es el encargado de administrar los recursos tecnológicos en una Facultad.

Tabla 1. Actores del Negocio

➤ Trabajadores:

Trabajadores	Descripción
Grupo de Soporte y Desarrollo	Grupo que se encarga de llevar a cabo la solicitud de hardware realizada.
Director Dirección Técnica	Persona que interactúa con el vale correspondiente a la solicitud de hardware.
Director Dirección IP	Persona que interactúa con el vale correspondiente a la solicitud de hardware.
Vicerrector Tecnología	Persona que interactúa con el vale correspondiente a la solicitud de hardware.
Vicerrector Producción	Persona que interactúa con el vale correspondiente a la solicitud de hardware.

2.2.2 Descripción General del Proceso de Negocio: Solicitar Hardware.

El proceso se inicia cuando un usuario realiza una solicitud de hardware al Grupo de Soporte Y Desarrollo de la Infraestructura Productiva, donde se recepciona esta solicitud creándose un vale con los datos de la misma para realizar el proceso de gestión de la tecnología. Posterior a la creación del vale, el mismo es llevado a la Dirección de la Infraestructura Productiva, luego a la Dirección de Tecnología, posteriormente a la Vicerrectoría de Tecnología y finalmente a la Vicerrectoría de Producción, en cada uno de estos departamentos el vale es aprobado y firmado, y luego es llevado por el Grupo de Soporte y Desarrollo al almacén para recoger los recursos solicitados y ser entregado al cliente.

2.2.3 Diagrama de Casos de Uso del Negocio (DCUN).

El Diagrama de Casos de Uso del Negocio describe los procesos de negocio de una empresa en términos de casos de uso y actores, que se corresponden con los procesos del negocio y los clientes respectivamente.

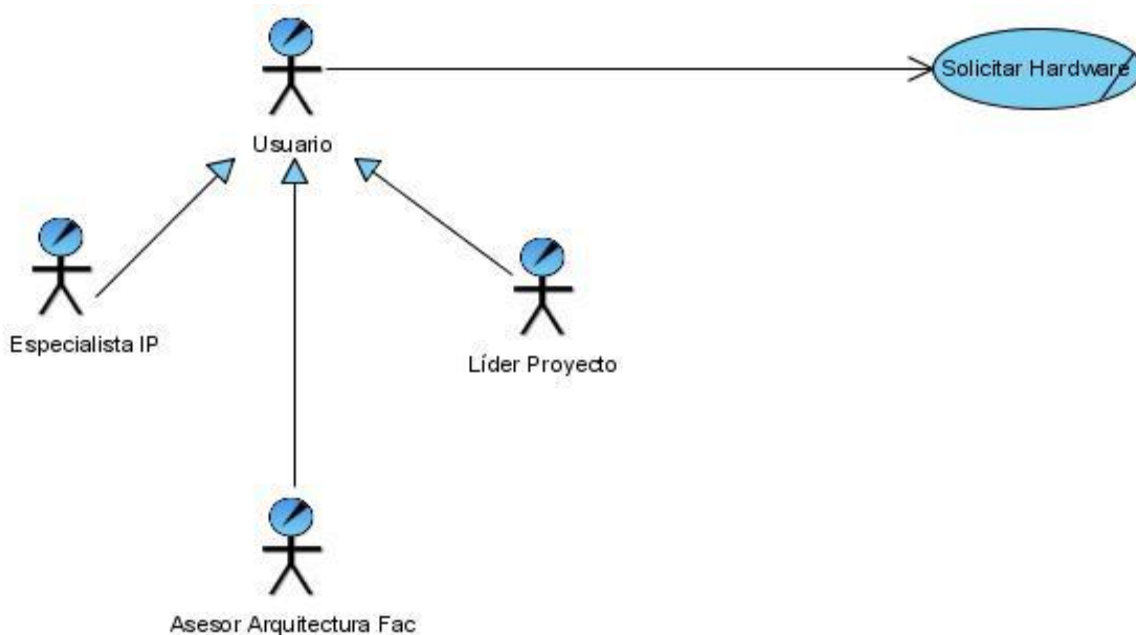


Figura 4. Diagrama de Casos de Uso del Negocio.

2.3 Realización de Casos de Uso.

A continuación se hace una descripción del caso de uso, en vista a una mayor comprensión del desarrollo del proceso.

2.3.1 Descripción del Caso de Uso de Negocio: Solicitar Hardware.

Caso de Uso:	Solicitar Hardware.
Actores:	Usuario.
Resumen:	El caso de uso se inicia cuando un cliente realiza una solicitud de hardware a través de la aplicación. Esta es recogida por el Grupo de Soporte y Desarrollo, quien se encarga de procesar la misma mediante un vale. Este vale es conducido a diferentes locales donde debe de ser firmado por dirigentes de la esfera de tecnología. Al concluir las actividades correspondientes, si no ha ocurrido ningún problema, se le asigna la tecnología al usuario.
Precondiciones:	La solicitud tiene no puede tener errores de confección.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Usuario Selecciona una de las opción de Solicitud de Hardware.	2. El Grupo de Soporte y Desarrollo (GSD), se encarga de revisar la solicitud (debe de haberse concebido correctamente).
	3. El GSD, se encarga de crear el vale correspondiente a la solicitud y llevarlo al Director de la Dirección de IP.
	4. El Director de la Dirección de IP recibe el vale y procede a firmarlo.
	5. El GSD recibe el vale firmado y lo lleva al Vicerrector Producción para que este lo firme.
	6. El Vice Rector Producción recibe el vale y procede a firmarlo.
	7. El GSD recibe el vale firmado y determina si el hardware es un activo o no.
	8. El GSD busca el identificador del activo en el docente número 4. Luego lleva el vale al Vicerrector Tecnología.
	9. El Vicerrector Tecnología recibe el vale y procede a firmarlo.
	10. El GSD recibe el vale y lo lleva al almacén de ATM para que lo firme el Director Dirección Técnica.
	11. El Director Dirección Técnica recibe el vale y procede a firmarlo.
	12. El GSD recibe el vale firmado y se encarga de entregar la solicitud de tecnología al usuario.
13. El usuario recibe la tecnología solicitada.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	2.1 Si la solicitud no ha sido concebida correctamente el GSD se encarga de devolver al usuario la solicitud informándole lo sucedido.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	7.1 Si el GSD determina que el hardware no es un activo, prosigue a la acción numero 9.
Pos condiciones	-

Tabla 3 Descripción Textual del caso de Uso: Solicitar Hardware

Reglas del Negocio.

A continuación se definirán las políticas que deben cumplirse o condiciones que deben satisfacerse para que se ejecute el caso de uso, de manera que regule el proceso de negocio de acuerdo a las especificidades del mismo.

- 1- Cuando adiciona una Facultad, debe seguir el prototipo: *Facultad #*. Además de que no se puede adicionar una facultad que posea por identificador el numero 0.
- 2- Cuando adiciona un Docente, debe seguir el prototipo: *Docente Identificador*.
- 3- Cuando adiciona un Laboratorio, debe seguir el prototipo: *Laboratorio Identificador*.
- 4- Cuando se adiciona una medida, esta debe ser mayor que cero y las unidades de medida deben ser del tipo: b, Kb, Mb, GB, TB.

2.5 Modelo de Objetos.

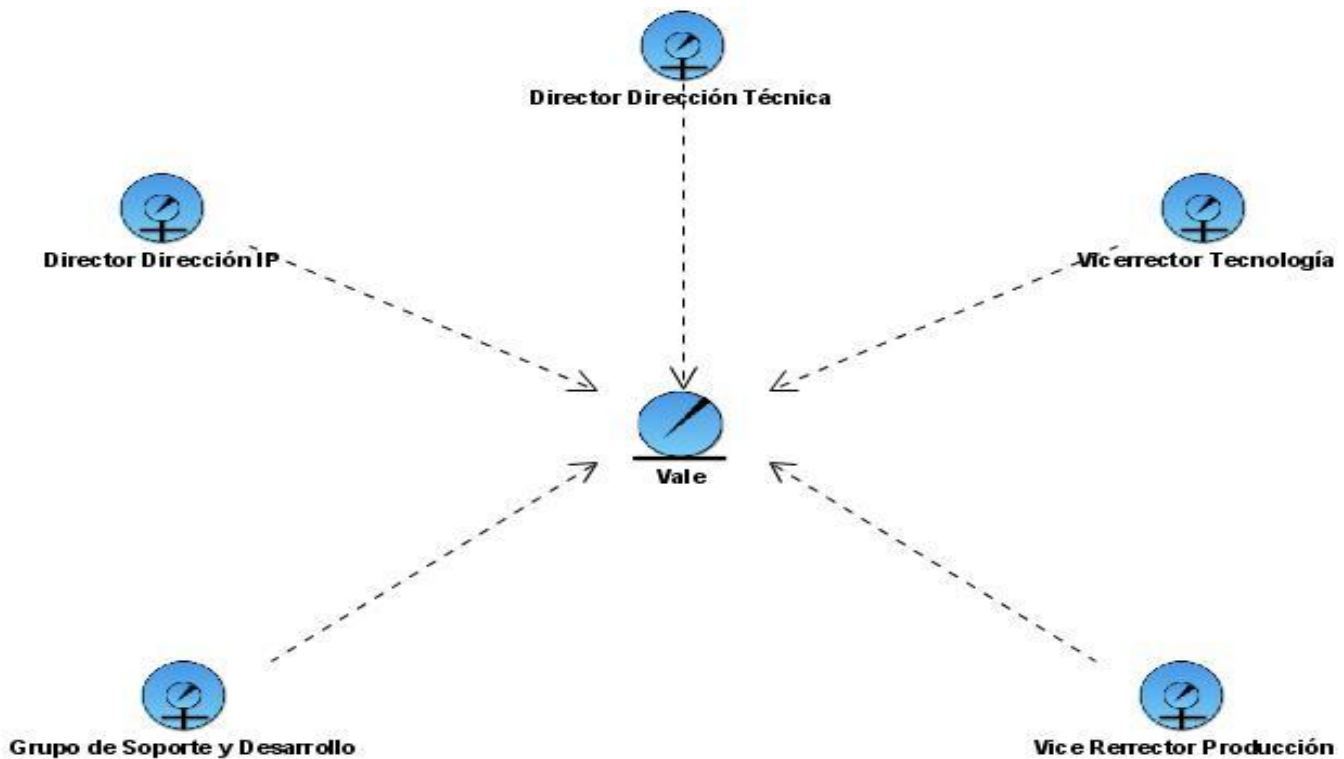
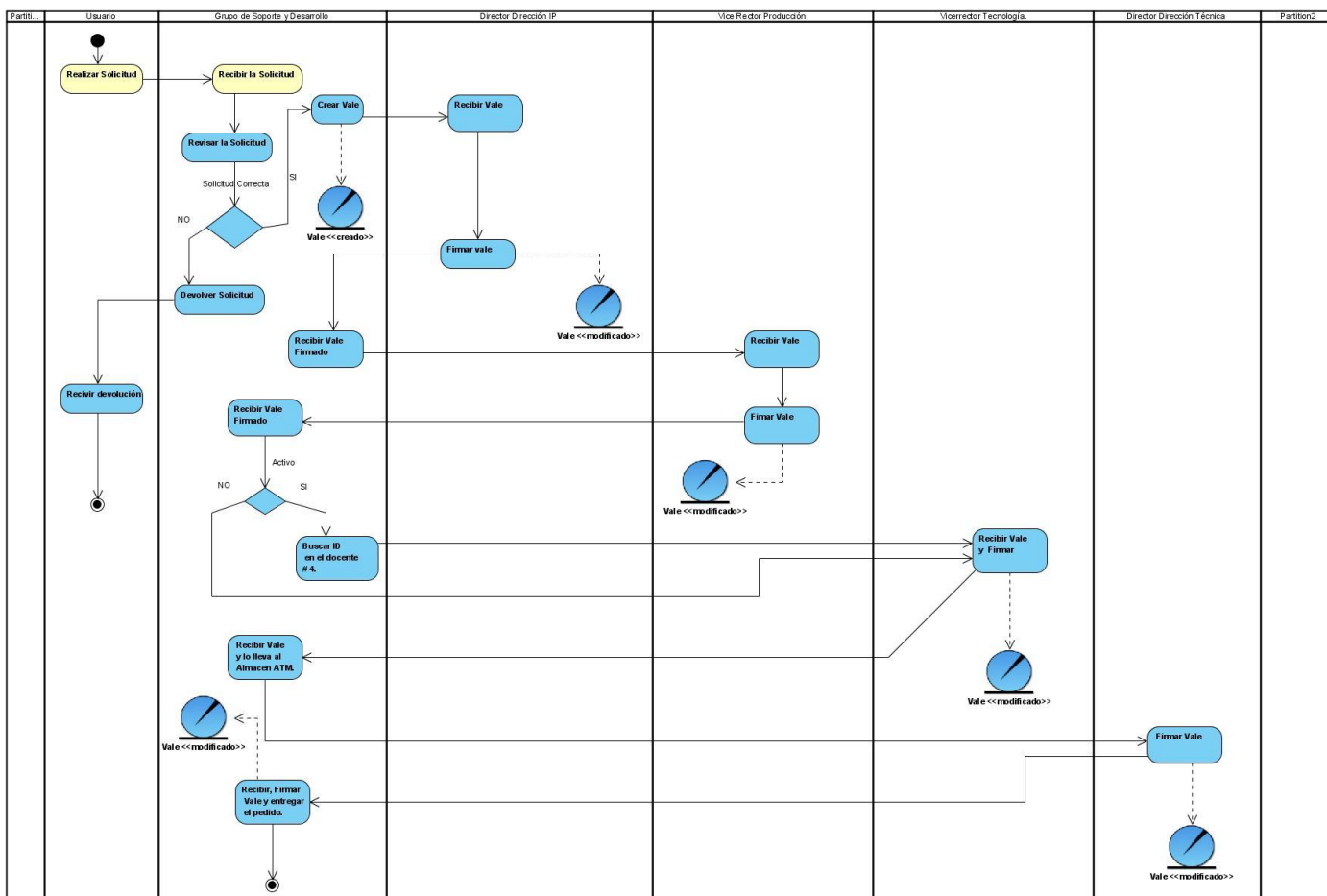


Figura 5. Modelo de Objetos.

2.6 Diagrama de Actividades



2.7 Modelo del Sistema.

A continuación se definirán los artefactos fundamentales del flujo de trabajo: Requerimientos, perteneciente a la Fase de Inicio, identificando y clasificando los requisitos y representado los actores y casos de uso del sistema.

2.7.1 Requerimientos Funcionales.

Para desarrollar un sistema con calidad es necesario definir qué requisitos debe cumplir el mismo en vista a elevar su efectividad, requisitos que al ser representados como casos de uso del sistema constituyen el hilo conductor del proceso de desarrollo. Un requisito puede quedar definido de la siguiente forma: una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.

Un requisito funcional especifica cómo debe reaccionar un sistema a una entrada particular y cómo se debe comportar ante situaciones particulares.

A continuación se expondrán los requisitos funcionales que debe cumplir el sistema que se propone.

RF1. El sistema debe ser capaz de: Autenticar Usuario.

RF2. El sistema debe ser capaz de: Gestionar Usuario.

- ✓ Insertar usuario.
- ✓ Eliminar usuario.

- ✓ Listar usuario.
- ✓ Modificar usuario.

RF3. El sistema debe ser capaz de: Gestionar Docente.

- ✓ Adicionar docente.
- ✓ Eliminar docente.
- ✓ Editar docente.
- ✓ Listar docente.

RF4. El sistema debe ser capaz de: Gestionar Facultad.

- ✓ Adicionar facultad.
- ✓ Eliminar facultad.
- ✓ Editar facultad.
- ✓ Listar facultad.

RF5. El sistema debe ser capaz de: Gestionar Proyecto Productivo.

- ✓ Adicionar proyecto productivo.
- ✓ Eliminar proyecto productivo.
- ✓ Editar proyecto productivo.
- ✓ Listar proyecto productivo.

RF6. El sistema debe ser capaz de Gestionar Laboratorio.

- ✓ Adicionar laboratorio.
- ✓ Eliminar laboratorio.
- ✓ Editar laboratorio.
- ✓ Listar laboratorio.

RF7. El sistema debe ser capaz de: Gestionar Solicitud.

- ✓ Adicionar solicitud.
- ✓ Eliminar solicitud.
- ✓ Editar solicitud.
- ✓ Listar solicitud.

RF8. El sistema debe ser capaz de: Gestionar Componente.

- ✓ Adicionar componente.
- ✓ Eliminar componente.
- ✓ Editar componente.
- ✓ Listar componente.

RF9. El sistema debe ser capaz de: Gestionar Característica.

- ✓ Adicionar característica.
- ✓ Eliminar característica.
- ✓ Editar característica.
- ✓ Listar característica.

RF10. El sistema debe ser capaz de: Gestionar Medidas.

- ✓ Adicionar medidas.
- ✓ Eliminar medidas.

- ✓ Editar medidas.
- ✓ Listar medidas.

2.7.2 Requerimientos No Funcionales.

Los requisitos no funcionales definen propiedades cualitativas del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad. Pueden especificar además la utilización de herramientas en particular, un lenguaje de programación o un método del desarrollo, así como la plataforma sobre la cual debe desarrollarse, la compatibilidad con sistemas operativos entre otras especificaciones que de no cumplirse podría quedar inutilizable.

En este sub-epígrafe se expondrán los requisitos no funcionales que el sistema propuesto debe cumplir.

REQUISITOS DE USABILIDAD.

RNF1. El sistema debe estar disponible las 24 horas del día sin ninguna interrupción.

RNF2. El sistema debe ser lo más sencillo posible y que facilite el trabajo a los usuarios, sin necesidad de que estos posean grandes conocimientos de informática.

RNF3. El acceso al sistema debe ser fácil y rápido.

RNF4. El sistema debe ser accesible desde todos los puntos donde exista una maquina conectada a la red.

RNF5. El sistema debe tener una interfaz que le sea familiar al usuario para aprovechar sus conocimientos en el manejo de herramientas de software.

RNF6. El sistema debe diferenciar las interfaces gráficas y opciones para los usuarios que accedan al sistema con diferentes roles.

RNF7. Capacidades de búsqueda con velocidad apropiada.

RNF8. El sistema debe ser Multiplataforma.

REQUISITOS DE CONFIABILIDAD Y SEGURIDAD.

RNF9. El sistema debe tener la capacidad de identificar con certeza a los diversos usuarios o entidades que interactúan con el.

RNF10. El sistema debe tener la capacidad de darle seguridad al usuario, de que las informaciones sólo serán vistas por quien esté capacitado para esto.

RNF11. El servicio del sistema debe tener una disponibilidad aceptable (99%).

RNF12. El tiempo entre fallos debe ser breve o cero, haciendo lo posible para que esto no ocurra.

RNF13. La base de datos debe estar fraccionada sobre varios esquemas, dividiendo así de una forma lógica las funcionalidades, evitando la pérdida total de la información en caso de algún accidente o ataque.

RNF14. Permitir autenticarse a través de un Servidor de Dominio.

REQUISITOS DE RENDIMIENTO.

RNF15. El sistema debe ser capaz de mantener el mismo rendimiento y estabilidad a medida que aumenta la cantidad de datos a gestionar.

RNF16. El Sistema debe ser capaz de funcionar al ser instalados en una misma PC todos sus componentes tales como Gestor de bases de Datos, Aplicación Web, etc.

REQUISITOS DE MANTENIMIENTO Y ACTUALIZACION.

RNF17. El ambiente de desarrollo donde se implementará dicho software será: Eclipse.

RNF18. Gestor de Base de Datos: PostGreSQL.

RNF19. Lenguaje de Programación: PHP

RNF20. El sistema debe ser construido en un código estándar, cada procedimiento debe estar comentado.

REQUISITOS DE DESEMPEÑO Y ESCALABILIDAD.

RNF21. El tiempo de respuestas debe ser corto con respuestas rápidas y eficientes.

RNF22. El sistema podrá soportar un gran número de clientes online.

RNF23. El sistema debe tener un rendimiento óptimo debido a que presta servicios a un gran número de usuarios.

REQUISITOS DE HARDWARE.

RNF24. Las computadoras clientes deben estar conectadas en red.

RNF25. Las computadoras clientes deben ser Pentium o superior.

RNF26. El servidor debe ser Pentium 2 o superior.

RNF27. El servidor debe tener al menos 1 tarjeta de red para establecer la conexión.

REQUERIMIENTOS DE SOFTWARE.

RNF28. Las computadoras clientes deben contar con Internet Explorer versión 6.

RNF29. La resolución de pantalla debe ser de 1024x768.

RNF30. El sistema operativo debe ser Ubuntu.

RNF31. La computadora servidor y el servidor de base de datos deben tener 2 GB de memoria RAM.

RNF32. El software del sistema debe ser operado por personas debidamente preparadas. Además debe tener revisiones y mantenimientos en el tiempo realizadas por personal capacitado.

2.7.3 Descripción de los actores del sistema.

Luego de haberse definido los requisitos tanto funcionales como no funcionales que el sistema debe cumplir, se definen los actores que intervienen en el mismo.

Actores Sistema	Descripción
Usuario	Persona que interactúa con el sistema y genera solicitudes.

	(es una generalización de los actores que intervienen en el proceso)
Líder Proyecto	Persona que interactúa con el sistema y genera solicitudes. Es el individuo que representa el proyecto para el cual se va a realizar la solicitud.
Especialista IP	Persona que interactúa con el sistema y genera solicitudes.
Asesor Facultad Arquitectura	Persona que interactúa con el sistema y genera solicitudes. Es el encargado de administrar los recursos tecnológicos en una Facultad.
Grupo Soporte y Desarrollo	Grupo que se encarga de gestionar la solicitud de hardware realizada.

Tabla 4 Actores del Sistema.

2.7.4 Diagrama de Casos de Uso del Sistema (DCUS).

Un Diagrama de Casos de Uso del Sistema representa gráficamente los procesos y su interacción con los actores, permitiendo así comprender qué actor interviene en qué proceso.

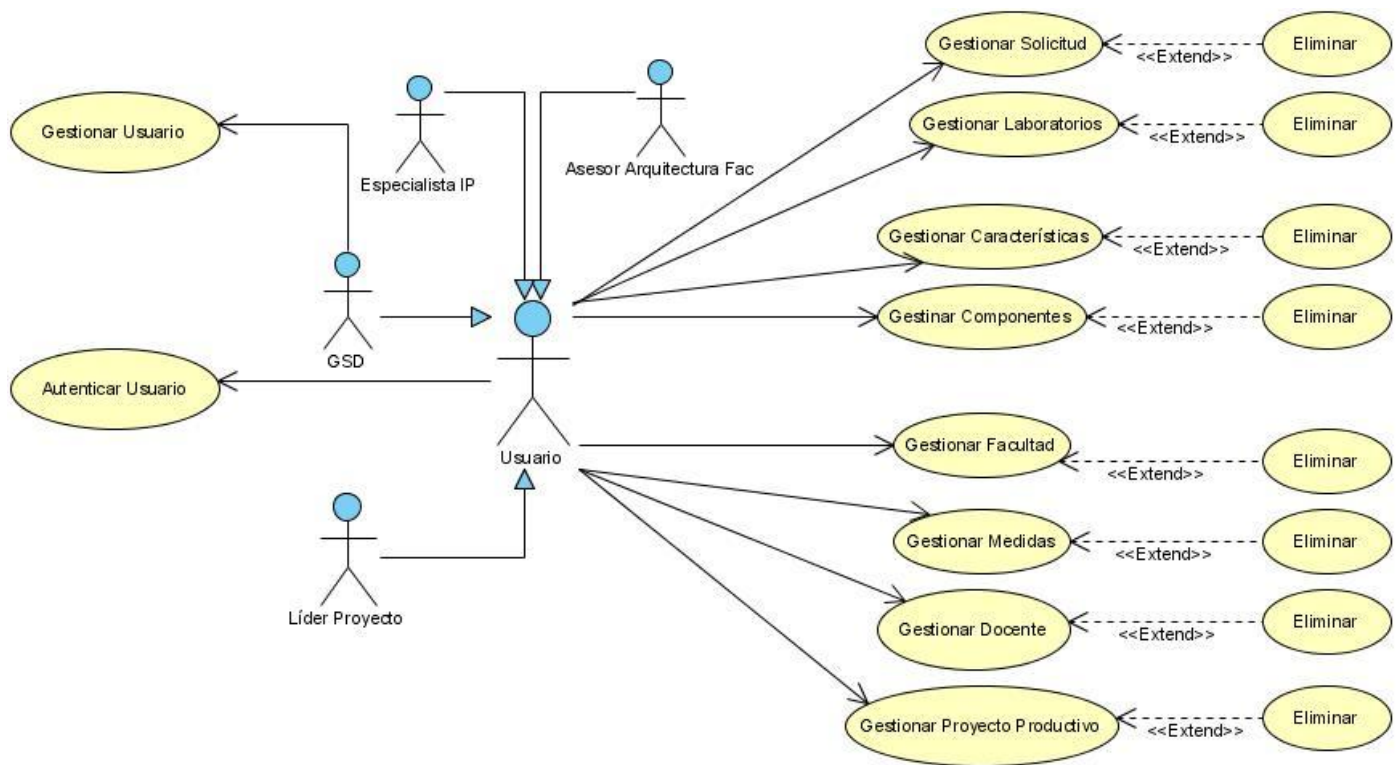


Figura 6. Diagrama de Casos de Uso del Sistema.

2.7.5 Descripción de los Casos de Uso.

Para una mayor comprensión de los casos de uso se explicara en detalle el flujo que describen.

2.7.5.1 Descripción del Caso de Uso: Autenticar Usuario.

Caso de Uso:	Autenticar Usuario.
---------------------	---------------------


Actores:	Usuario(inicia)
Resumen:	El caso de uso permite que los usuarios con determinadas responsabilidades en el sistema introduzcan sus credenciales (usuario y contraseña) locales para que el sistema las verifique y ejecuten las funcionalidades que puedan según su rol.
Referencia:	RF1
Precondiciones:	-
Poscondiciones:	El sistema permite que el usuario se autentifique y acceda a la zona de trabajo que tiene acceso.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario inicia la aplicación.	2. El sistema le muestra una interfaz con un formulario para que se autentique.
3. El usuario ingresa los datos en el formulario (usuario y contraseña).	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. Verifica la existencia del usuario en el sistema.
	6. El sistema le da acceso al usuario para interactuar con la aplicación y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos").
	5.1. Muestra un mensaje de error indicando que la cuenta no existe. ("Usuario no autenticado o contraseña incorrecta").
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla 5 Descripción Textual del Caso de Uso: Autenticar Usuario.

2.7.5.2 Descripción del Caso de Uso: Gestionar Usuario.

Caso de Uso:	Gestionar Usuario
Caso de Uso Asociado	-
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Insertar, Listar o Modificar Usuario del menú de operaciones. 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.
Referencia:	RF2
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.

Poscondiciones:	<ol style="list-style-type: none"> 1. Información de usuario adicionada a la Base de Datos. 2. Información de usuario modificada en la base de Datos. 3. Mostrar la información que contiene un usuario determinado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
14. El Usuario Selecciona la opción de <u>Adicionar Usuario.</u>	15. El sistema le muestra una interfaz con los datos a completar. (Usuario y contraseña).
16. El Usuario ingresa los datos solicitados por el sistema.	17. El sistema verifica que los campos del formulario no estén vacíos.
	18. El sistema verifica que el usuario no esté registrado en la base de datos.
	19. El sistema procede al registro del usuario en la base de datos correspondiente y termina el CUS.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. Muestra un mensaje de error indicando que la cuenta ya existe. ("El nombre de usuario especificado ya existe").
Sección Modificar Usuario	
1. El Usuario selecciona la opción <u>Modificar Usuario</u>	2. El sistema busca todos los usuarios existentes y los muestra en una pantalla para que el Usuario elija el usuario a modificar.
3. El Usuario elige el usuario y pincha en el botón Modificar Usuario.	4. El sistema muestra una pantalla con todos los datos del usuario elegido de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos del usuario seleccionado.
Flujos Alternos de los Eventos	
	19.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Usuarios	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los usuarios existentes y los muestra en una nueva pantalla y termina el caso de uso.
Sección Eliminar Usuario	
1. El Especialista GSD selecciona la opción <u>Eliminar Usuario.</u>	2. El sistema busca todos los usuarios existentes y los muestra en una pantalla para que el Especialista GSD elija el usuario a eliminar.
3. El Especialista GSD elige el usuario y pincha en el botón	4. El sistema muestra una confirmación para estar seguro de que lo desea eliminar.

Eliminar.	
5. El Especialista GSD confirma que desea eliminar el usuario.	6. El sistema procede a la eliminación de los datos del usuario seleccionado y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 6 Descripción Textual del Caso de Uso: Gestionar Usuario.

2.7.5.3 Descripción del Caso de Uso: Gestionar Componentes.

Caso de Uso:	Gestionar Componente
Caso de Uso Asociado	Eliminar Componente
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Insertar, Buscar o Modificar Componente del menú de operaciones. 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.
Referencia:	RF8
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información del componente adicionado a la Base de Datos. 2. Información del componente modificado en la base de Datos. 3. Mostrar la información que contiene un componente determinado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario Selecciona la opción de <u>Adicionar Componente.</u>	2. El sistema le muestra una interfaz con los datos a completar.
3. El Usuario ingresa los datos solicitados por el sistema.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que el componente no esté registrado en la base de datos.


	6. El sistema procede al registro del componente en la base de datos correspondiente y termina el CUS.
Flujos Alternos	
	1.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. Muestra un mensaje de error indicando que el componente ya existe. ("El componente especificado ya existe").
Sección Modificar Componente	
1. El Usuario selecciona la opción <u>Modificar Componente</u> .	2. El sistema busca todos los componentes existentes y los muestra en una pantalla para que el Usuario elija el componente a modificar.
3. El Usuario elige el componente y pincha en el botón Modificar Componente.	4. El sistema muestra una pantalla con todos los datos del componente elegido de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos del componente seleccionado.
Flujos Alternos de los Eventos	
	6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Componentes	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los componentes existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 7 Descripción Textual del Caso de Uso: Gestionar Componente.

2.7.5.4 Descripción del Caso de Uso: Gestionar Características.

Caso de Uso:	Gestionar Característica
Caso de Uso Asociado	Eliminar Característica
Actores:	Usuario(inicia)

Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Insertar, Buscar o Modificar Característica del menú de operaciones. 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.	
Referencia:	RF9	
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.	
Poscondiciones:	<ol style="list-style-type: none"> 1. Información de la característica adicionada a la Base de Datos. 2. Información de la característica modificada en la base de Datos. 3. Mostrar la información que contiene una característica determinada. 	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El Usuario Selecciona la opción de <u>Adicionar Característica.</u>	2. El sistema le muestra una interfaz con los datos a completar.
	3. El Usuario ingresa los datos solicitados por el sistema.	4. El sistema verifica que los campos del formulario no estén vacíos.
		5. El sistema verifica que la característica no esté registrada en la base de datos.
		6. El sistema procede al registro de la característica en la base de datos correspondiente y termina el CUS.
Flujos Alternos		
		1.2. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
		5.1. Muestra un mensaje de error indicando que la característica ya existe. ("La característica especificada ya existe").
Sección Modificar Característica		
	1. El Usuario selecciona la opción <u>Modificar Característica</u>	2. El sistema busca todas las características existentes y las muestra en una pantalla para que el Usuario elija la característica a modificar.
	3. El Usuario elige la característica y pincha en el botón Modificar Características.	4. El sistema muestra una pantalla con todos los datos de la características elegida de forma editable.
	5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
		7. El sistema modifica todos lo datos de la características seleccionada.
Flujos Alternos de los Eventos		
		6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Características		


1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca las características existentes y las muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 8 Descripción Textual del Caso de Uso: Gestionar Características.

2.7.5.5 Descripción del Caso de Uso: Gestionar Laboratorios.

Caso de Uso:	Gestionar Laboratorio
Caso de Uso Asociado	Eliminar Laboratorio
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar Laboratorios del menú de operaciones. 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.
Referencia:	RF6
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información del laboratorio adicionada a la Base de Datos. 2. Información del laboratorio modificada en la base de Datos. 3. Mostrar los laboratorios existentes.
Flujo Normal de Eventos	
Sección Adicionar Laboratorio.	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Laboratorio</u> .	2. El sistema le muestra una interfaz con los datos a completar.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que ese laboratorio no esté registrado en la base de datos.
	6. El sistema registra los datos del nuevo laboratorio y muestra el nuevo laboratorio ya insertado.
Flujos Alternos	


	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. El sistema muestra un mensaje de error indicando que ese laboratorio ya existe en la base de datos. ("El laboratorio especificado ya existe.")
Sección Modificar Laboratorio.	
1. El Usuario selecciona la opción <u>Modificar Laboratorio</u> .	2. El sistema busca todos los Laboratorios existentes UCI y las muestra en una pantalla para que el Usuario elija el Laboratorio a Modificar.
3. El Usuario elije el Laboratorio a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos del laboratorio de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos lo datos del laboratorio seleccionado.
Flujos Alternos de los Eventos	
	6.1 El sistema muestra un mensaje de error indicando el problema.
Sección Listar Laboratorios.	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los laboratorios existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 9 Descripción Textual del Caso de Uso: Gestionar Solicitudes Laboratorio.

2.7.5.6 Descripción del Caso de Uso: Gestionar Facultad.

Caso de Uso:	Gestionar Facultades
Caso de Uso Asociado	Eliminar Facultad
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a las opciones de Adicionar, Modificar, o Mostrar las Facultades de la UCI. El caso de uso termina cuando el sistema realiza la operación elegida, y el Usuario accede a otra opción o simplemente sale de la aplicación.

Referencia:	RF4
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información de la facultad adicionada a la Base de Datos. 2. Información de la facultad modificada en la base de Datos. 3. Mostrar las facultades existentes.
Flujo Normal de Eventos	
Sección Adicionar Facultad.	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Facultad</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos de la nueva Facultad.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que esa facultad no esté registrada en la base de datos.
	6. El sistema registra los datos de la nueva facultad y muestra la nueva facultad ya insertado.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. El sistema muestra un mensaje de error indicando que ese proyecto ya existe en la base de datos. ("La facultad especificada ya existe en la base de datos").
Sección Modificar Facultad.	
1. El Usuario selecciona la opción <u>Modificar Facultad</u> .	2. El sistema busca todas las Facultades de la UCI y las muestra en una pantalla para que el Usuario elija la Facultad a Modificar.
3. El Usuario elige el Docentes a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos de la facultad de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos de la facultad seleccionada.
Flujos Alternos de los Eventos	
	6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Facultades.	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca las facultades existentes y las muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico



Tabla 10 Descripción Textual del Caso de Uso: Gestionar Facultad.

2.7.5.7 Descripción del Caso de Uso: Gestionar Docente.

Caso de Uso:	Gestionar Docentes
Caso de Uso Asociado	Eliminar Docente
Actores:	Usuario (inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a las opciones de Adicionar, Modificar, o Mostrar los Docentes de la UCI. El caso de uso termina cuando el sistema realiza la operación elegida, y el Usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF3
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información del docente adicionado a la Base de Datos. 2. Información del docente modificado en la base de Datos. 3. Mostrar los docentes existentes.
Flujo Normal de Eventos	
Sección Adicionar Docente.	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Docente</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo Docente.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que ese docente no esté registrado en la base de datos.
	6. El sistema registra los datos del nuevo docente y muestra el nuevo docente ya insertado.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. El sistema muestra un mensaje de error indicando que ya existe ese docente en la


	base de datos. ("El docente especificado ya existe en la base de datos").
Sección Modificar Docente.	
1. El Usuario selecciona la opción <u>Modificar Docente</u> .	2. El sistema busca todos los Docentes de la UCI y las muestra en una pantalla para que el Usuario elija el Docente a Modificar.
3. El Usuario elige el Docentes a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos del docente de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos del docente seleccionado.
Flujos Alternos de los Eventos	
	1.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Docentes.	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los docentes existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 11 Descripción Textual del Caso de Uso: Gestionar Docente.

2.7.5.8 Descripción del Caso de Uso: Gestionar Proyecto Productivo.

Caso de Uso:	Gestionar Proyecto Productivo
Caso de Uso Asociado	Eliminar Proyecto Productivo
Actores:	Usuario (inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar proyectos productivos del menú de operaciones. 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.
Referencia:	RF5
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información del proyecto adicionado a la Base de Datos. 2. Información del proyecto modificado en la base de

	Datos. 3. Mostrar los proyectos productivos existentes.
Flujo Normal de Eventos	
Sección Adicionar Proyecto Productivos.	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Proyecto.</u>	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo proyecto productivo.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón Aceptar.	4. El sistema verifica que los campos del proyecto no estén vacíos.
	5. El sistema verifica que ese proyecto no esté registrado en la base de datos.
	6. El sistema registra los datos del nuevo proyecto productivo y muestra el nuevo proyecto productivo ya insertado.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. El sistema muestra un mensaje de error indicando que ese proyecto ya existe en la base de datos. ("El proyecto especificado ya existe en la base de datos")
Sección Modificar Proyecto Productivos.	
1. El Usuario selecciona la opción <u>Modificar Proyecto Productivo.</u>	2. El sistema busca todos los Proyectos Productivos de la UCI y las muestra en una pantalla para que el Usuario elija el Proyecto Productivo a Modificar.
3. El Usuario elije el Proyecto Productivo a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos del proyecto productivo de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que todos lo datos modificados estén correctos.
	7. El sistema modifica todos lo datos del proyecto productivo seleccionado.
Flujos Alternos de los Eventos	
	1.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Proyecto Productivos.	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los proyectos Productivos existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico



Tabla 42 Descripción Textual del Caso de Uso: Gestionar Proyecto Productivo.

2.7.5.9 Descripción del Caso de Uso: Gestionar Solicitud.

Caso de Uso:	Gestionar Solicitud
Caso de Uso Asociado	Eliminar Solicitud
Actores:	Usuario (inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar solicitud del menú de operaciones. 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.
Referencia:	RF7
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información de la solicitud adicionada a la Base de Datos. 2. Información de la solicitud modificada en la base de Datos. 3. Mostrar las solicitudes existentes.
Flujo Normal de Eventos	
Sección Adicionar Solicitud.	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Solicitud.</u>	2. El sistema muestra la interfaz para que el Usuario introduzca los datos de la nueva solicitud.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos de la solicitud no estén vacíos.
	5. El sistema registra los datos de la nueva solicitud y muestra la nueva solicitud ya insertada.
Flujos Alternos	
	4.1 Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
Sección Modificar Solicitud.	

1. El Usuario selecciona la opción <u>Modificar Solicitud</u> .	2. El sistema busca todas las solicitudes de la UCI y las muestra en una pantalla para que el Usuario elija la solicitud a Modificar.
3. El Usuario elige la Solicitud a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos de la solicitud de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que todos los datos modificados estén correctos.
	7. El sistema modifica todos los datos de la solicitud seleccionada.
Flujos Alternos de los Eventos	
	6.1 El sistema muestra un mensaje de error indicando el problema.
Sección Listar Solicitud.	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca las solicitudes existentes y las muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla 13 Descripción Textual del Caso de Uso: Gestionar Solicitud.

2.7.5.10 Descripción del Caso de Uso: Gestionar Medidas.

Caso de Uso:	Gestionar Medidas
Caso de Uso Asociado	Eliminar Medidas
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Insertar, Buscar o Modificar Medidas del menú de operaciones. 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.
Referencia:	RF10
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Información de la medida adicionada a la Base de Datos. 2. Información de la medida modificada en la base de Datos. 3. Mostrar la información que contiene una medida determinada.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario Selecciona la opción de <u>Adicionar Medida.</u>	2. El sistema le muestra una interfaz con los datos a completar.
3. El Usuario ingresa los datos solicitados por el sistema.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que la medida no esté registrada en la base de datos.
	6. El sistema procede al registro de la medida en la base de datos correspondiente y termina el CUS.
Flujos Alternos	
	4.1 Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1 Muestra un mensaje de error indicando que la medida ya existe. ("El nombre de la medida especificada ya existe").
Sección Modificar Medida	
7. El Usuario selecciona la opción <u>Modificar Medida.</u>	8. El sistema busca todas las medidas existentes y las muestra en una pantalla para que el Usuario elija la medida a modificar.
9. El Usuario elige la medida y pincha en el botón Modificar Medida.	10. El sistema muestra una pantalla con todos los datos de la medida elegida de forma editable.
11. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	12. El sistema verifica que los campos modificados estén correctos.
	13. El sistema modifica todos los datos de la medida seleccionada.
Flujos Alternos de los Eventos	
	6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Listar Medida	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca las medidas existentes y las muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico



Tabla 14 Descripción Textual del Caso de Uso: Gestionar Medidas.

I Descripción del Caso de Uso: Eliminar Proyecto

Caso de Uso:	Eliminar Proyecto <<extend>>	
Actores:	Especialista GSD (inicia)	
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un proyecto del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF5	
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.	
Poscondiciones:	Información del proyecto eliminada de la base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista GSD selecciona la opción Eliminar Proyecto Productivo.	2. El sistema busca todos los proyectos productivos existentes y los muestra en una pantalla para que el Especialista GSD elija el proyecto productivo a eliminar.	
3. El Especialista GSD elige el proyecto productivo y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.	
5. El Especialista GSD confirma que desea eliminar el proyecto.	6. El sistema procede a la eliminación de los datos del proyecto productivo seleccionado y termina el CUS.	
Flujos Alternos		
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.		
Prioridad:	Secundario	



Tabla #15: Descripción textual del Caso de Uso: Eliminar Proyecto

II Descripción del Caso de Uso: Eliminar Docente

Caso de Uso:	Eliminar Docente <<extend>>	
Actores:	Especialista GSD (inicia)	
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar docente del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF3	
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.	
Poscondiciones:	1. Información del docente eliminado de la base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista GSD selecciona la opción <u>Eliminar Docente</u> .	2. El sistema busca todos los docentes existentes y los muestra en una pantalla para que el Especialista GSD elija el docente a eliminar.	
3. El Especialista GSD elige el docente y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.	
5. El Especialista GSD confirma que desea eliminar el docente.	6. El sistema procede a la eliminación de los datos del docente seleccionado y termina el CUS.	
Flujos Alternos		
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.		
Prioridad:	Secundario	
Prototipo de Interfaz:		

Tabla #16: Descripción textual del Caso de Uso: Eliminar Docente.

III Descripción del Caso de Uso: Eliminar Facultad

Caso de Uso:	Eliminar Facultad <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar una facultad del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la

	aplicación.
Referencia:	RF4
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	1. Información de la facultad eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Facultad</u> .	2. El sistema busca todas las facultades existentes y las muestra en una pantalla para que el Especialista GSD elija la facultad a eliminar.
3. El Especialista GSD elige la facultad y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que la desea eliminar.
5. El Especialista GSD confirma que desea eliminar la facultad.	6. El sistema procede a la eliminación de los datos de la facultad seleccionada y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #17: Descripción textual del Caso de Uso: Eliminar Facultad

IV Descripción del Caso de Uso: Eliminar Laboratorio

Caso de Uso:	Eliminar Laboratorio <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un Laboratorio del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF6
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	1. Información del laboratorio eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Laboratorio</u> .	2. El sistema busca todos los laboratorios existentes y los muestra en una pantalla

	para que el Especialista GSD elija el laboratorio a eliminar.
3. El Especialista GSD elige el laboratorio y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.
5. El Especialista GSD confirma que desea eliminar el laboratorio.	6. El sistema procede a la eliminación de los datos del laboratorio seleccionado y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #18: Descripción textual del Caso de Uso: Eliminar Facultad

V Descripción del Caso de Uso: Eliminar Solicitud

Caso de Uso:	Eliminar Solicitud. <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar una Solicitud del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF7
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	1. Información de la solicitud eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Solicitud</u> .	2. El sistema busca todas las solicitudes existentes y las muestra en una pantalla para que el Especialista GSD elija la solicitud a eliminar.
3. El Especialista GSD elige el solicitud y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que la desea eliminar.
5. El Especialista GSD confirma que desea eliminar la solicitud.	6. El sistema procede a la eliminación de los datos de la solicitud seleccionada y termina el CUS.
Flujos Alternos	

5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #19: Descripción textual del Caso de Uso: Eliminar Solicitud

VI Descripción del Caso de Uso: Eliminar Componente

Caso de Uso:	Eliminar Componente. <<extend>>	
Actores:	Especialista GSD (inicia)	
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un Componente del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF8	
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.	
Poscondiciones:	14. Información del componente eliminado de la base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista GSD selecciona la opción <u>Eliminar Componente</u> .	2. El sistema busca todos los componentes existentes y los muestra en una pantalla para que el Especialista GSD elija el componente a eliminar.	
3. El Especialista GSD elige el componente y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.	
5. El Especialista GSD confirma que desea eliminar el componente.	6. El sistema procede a la eliminación de los datos del componente seleccionado y termina el CUS.	
Flujos Alternos		
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.		
Prioridad:	Secundario	



Tabla #20: Descripción textual del Caso de Uso: Eliminar Componente

VII Descripción del Caso de Uso: Eliminar Características

Caso de Uso:	Eliminar Características. <<extend>>	
Actores:	Especialista GSD (inicia)	
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar Característica del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF9	
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.	
Poscondiciones:	7. Información de la característica eliminada de la base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista GSD selecciona la opción <u>Eliminar Característica</u> .	2. El sistema busca todas las características existentes y las muestra en una pantalla para que el Especialista GSD elija la característica a eliminar.	
3. El Especialista GSD elige la característica y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que la desea eliminar.	
5. El Especialista GSD confirma que desea eliminar la característica.	6. El sistema procede a la eliminación de los datos de la característica seleccionada y termina el CUS.	
Flujos Alternos		
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.		
Prioridad:	Secundario	



Tabla #21: Descripción textual del Caso de Uso: Eliminar Características

VIII Descripción del Caso de Uso: Eliminar Medidas

Caso de Uso:	Eliminar Medidas. <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar Medida del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF10
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	7. Información de la medida eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Medida</u> .	2. El sistema busca todas las medidas existentes y las muestra en una pantalla para que el Especialista GSD elija la medida a eliminar.
3. El Especialista GSD elige la medida y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que la desea eliminar.
5. El Especialista GSD confirma que desea eliminar la medida.	6. El sistema procede a la eliminación de los datos de la medida seleccionada y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario

Prototipo de Interfaz:



Tabla #22: Descripción textual del Caso de Uso: Eliminar Medidas

2.8 Conclusiones.

En este capítulo se han definido y descrito textualmente todos los casos de uso del negocio así como del sistema, actores y trabajadores, se ha desarrollado el modelo de objetos y las reglas del negocio. Se han detallado los requisitos funcionales y no funcionales que el sistema debe cumplir, constituyendo en su conjunto la base esencial para poder implementar la solución.

CAPITULO 3. Construcción de la solución propuesta.

3.1 Introducción.

En el presente capítulo, se expondrán los artefactos fundamentales correspondientes a los flujos de trabajo Análisis y Diseño, así como el de Implementación. La construcción de la solución propuesta tiene como base el análisis de ésta a nivel de casos de uso. Aquí se define el modelo de datos, el modelo de despliegue y el diagrama de componentes, elementos todos que contribuyen a la construcción del sistema.

3.2 Análisis.

En este flujo de trabajo se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. El objetivo principal de este flujo son los diagramas de clases de análisis, los cuales muestran qué clases participan en las realizaciones de los distintos casos de usos, constituyendo la base sobre la cual debe realizarse el Modelo de Diseño.

3.3 Diagramas de Clases del Análisis.

Estos diagramas se encuentran en los Anexos1.

3.3.1 Diagramas de Interacción del Análisis.

Los diagramas de interacción o colaboración son utilizados en el modelado de los aspectos dinámicos de un sistema, constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Los Diagramas de Interacción se encuentran en los Anexos2.

3.4 Diseño.

El Modelo de Diseño describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a considerar. Está ligado a un lenguaje de programación específico, por lo que para un mayor entendimiento del diseño de las clases realizado, es conveniente hacer una breve descripción del funcionamiento del framework que se utilizó, pues sin conocer el funcionamiento general de Symfony es muy difícil entender el flujo que describen los diagramas.

A continuación se hace una descripción del funcionamiento de Symfony.

Symfony organiza el código fuente en una estructura de tipo proyecto y almacena los archivos del proyecto en una estructura estandarizada de tipo árbol. Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones. Cada aplicación a su vez está constituida por uno o más módulos, y un módulo normalmente representa a una página web o a un grupo de páginas con un propósito relacionado. Los módulos además almacenan las acciones, que representan cada una de las operaciones que se pueden realizar dentro de él.

Symfony toma lo mejor de la arquitectura Modelo Vista Controlador y la implementa de forma que el desarrollo

de aplicaciones sea rápido y sencillo. En primer lugar, el controlador frontal y el layout son comunes para todas las acciones de la aplicación. El controlador frontal es un componente que sólo tiene código relativo al Modelo Vista Controlador, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita (o pinchada) por el usuario.

Las Actions son el motor de la aplicación, puesto que contienen toda la lógica de la misma. Utilizan el modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición. Son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreModuloActions` que hereda de la clase `sfActions` y se encuentran agrupadas por módulos. La clase que representa las acciones de un módulo se encuentra en el archivo `actions.class.php`, en el directorio `actions` del módulo.

Symfony maneja automáticamente las sesiones del usuario y es capaz de almacenar datos de forma persistente entre peticiones. Utiliza el mecanismo de manejo de sesiones incluido en PHP y lo mejora para hacerlo más configurable y más fácil de usar. Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida.

Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. La librería Propel se encarga de esta generación automática, ya que crea el *esqueleto* o estructura básica de las clases y genera automáticamente el código necesario.

La abstracción de la base de datos es completamente invisible al programador, ya que la realiza otro componente específico llamado Creole. Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración.

La vista se encarga de producir las páginas que se muestran como resultado de las acciones. Normalmente se trabaja con las plantillas (que son la presentación de los datos de la acción que se está ejecutando) y con el layout (que contiene el código HTML común a todas las páginas). Estas partes están formadas por código HTML que contiene pequeños trozos de código PHP, que normalmente son llamadas a los diversos *helpers* disponibles. Los *helpers* son funciones de PHP que devuelven código HTML y que se utilizan en las plantillas. La plantilla global o Layout, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en él, o si se mira desde el otro punto de vista, *decora* la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “decorator”. Normalmente se utiliza el layout para mostrar la navegación, el logotipo del sitio, etc. Incluso es posible definir más de un layout y decidir en cada acción cuál se va a utilizar.

3.4.1 Diagrama de Clases del Diseño.

Estos diagramas se encuentran en los Anexos3.

3.5 Diseño de la Base de Datos.

El diseño de la Base de Datos es fundamental dentro del proceso de desarrollo de un software. Luego de haber definido las clases del diseño que intervienen en los casos de uso, a continuación se representan cuales de estas clases poseen un carácter permanente, para luego desarrollar el modelo de datos del sistema.

3.5.1 Diagrama de Clases Persistentes.

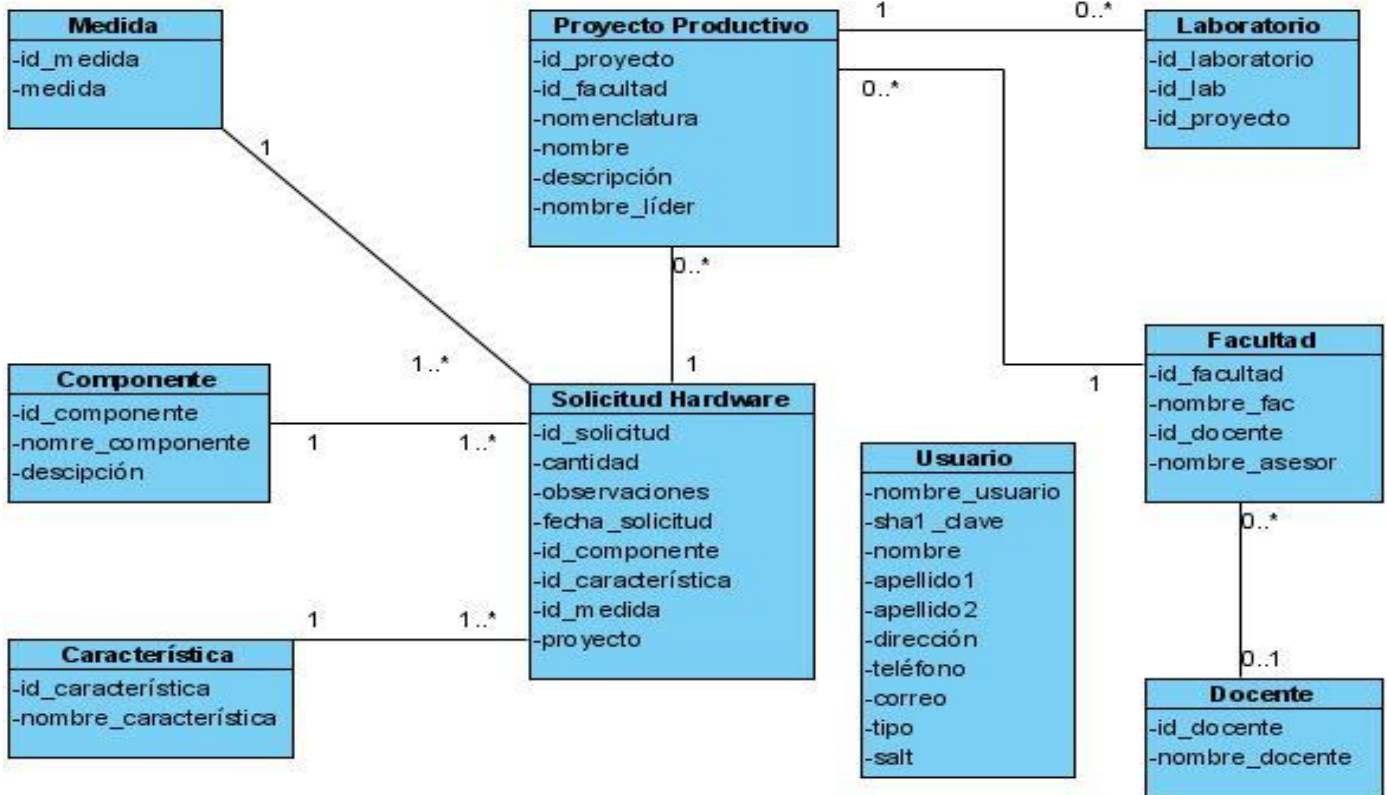


Figura 7. Diagrama de Clases Persistentes.

3.5.2 Diagrama Entidad Relación.

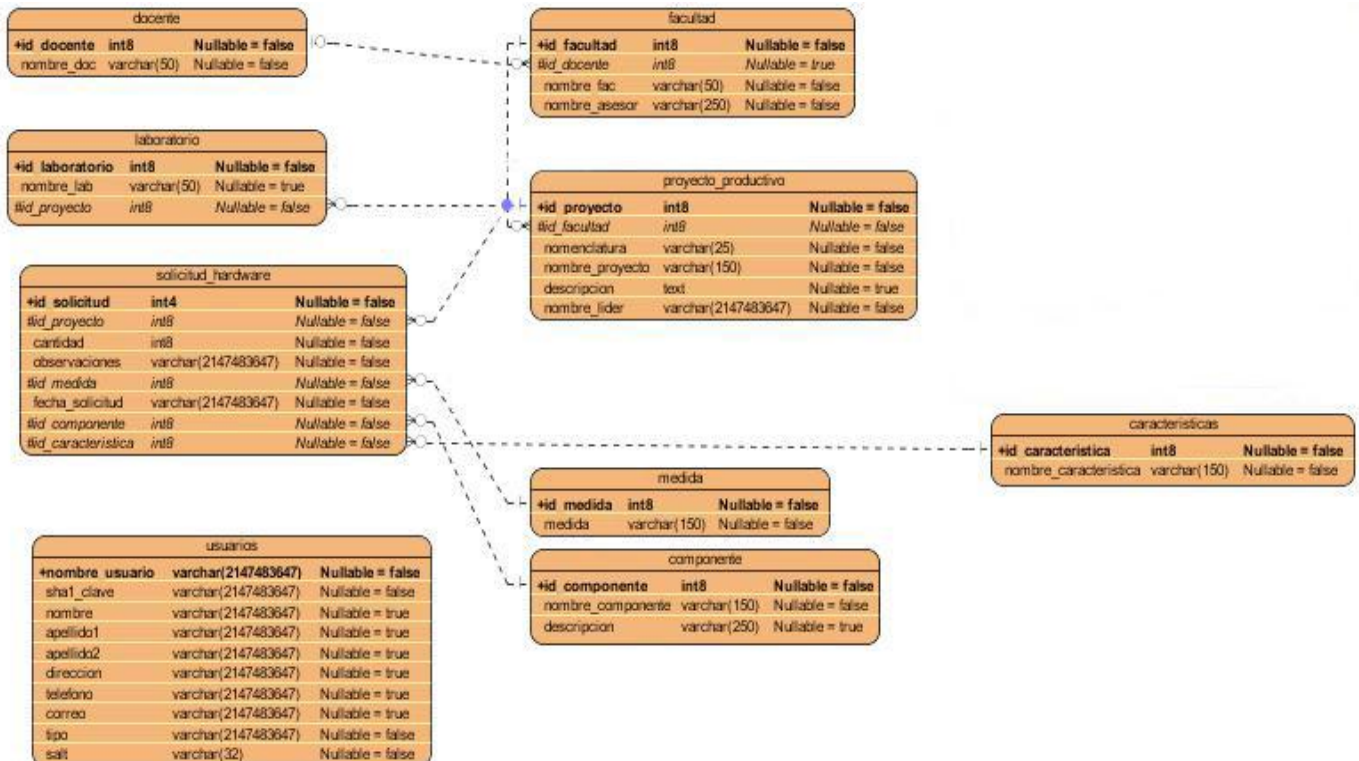


Figura 8. Diagrama Entidad Relación.

3.6 Modelo de Despliegue.

El Modelo de Despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo a través de una infraestructura.

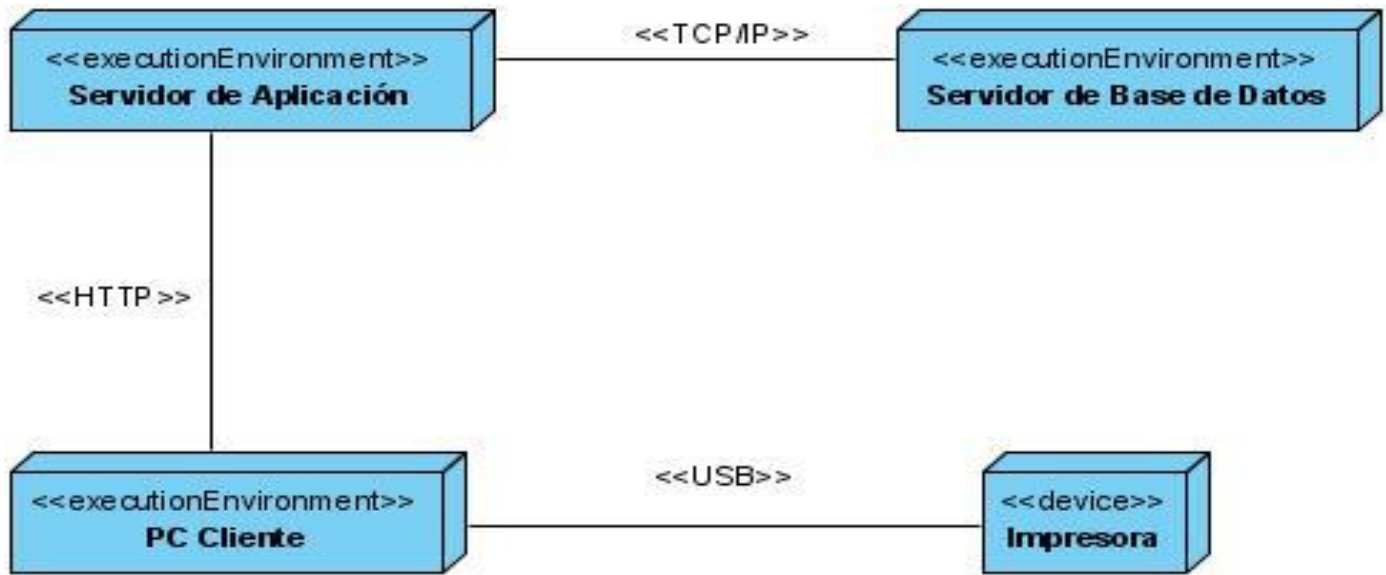


Figura 9. Diagrama de Despliegue.

3.7 Modelo de Implementación.

3.7.1 Diagrama de Componentes.

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado incluidos en un diagrama de componentes serán componentes y paquetes.

El Diagrama de Componentes, en resumen: define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos.

A continuación se muestra el diagrama de componentes correspondiente al sistema que se propone en la presente investigación.

Vista General:

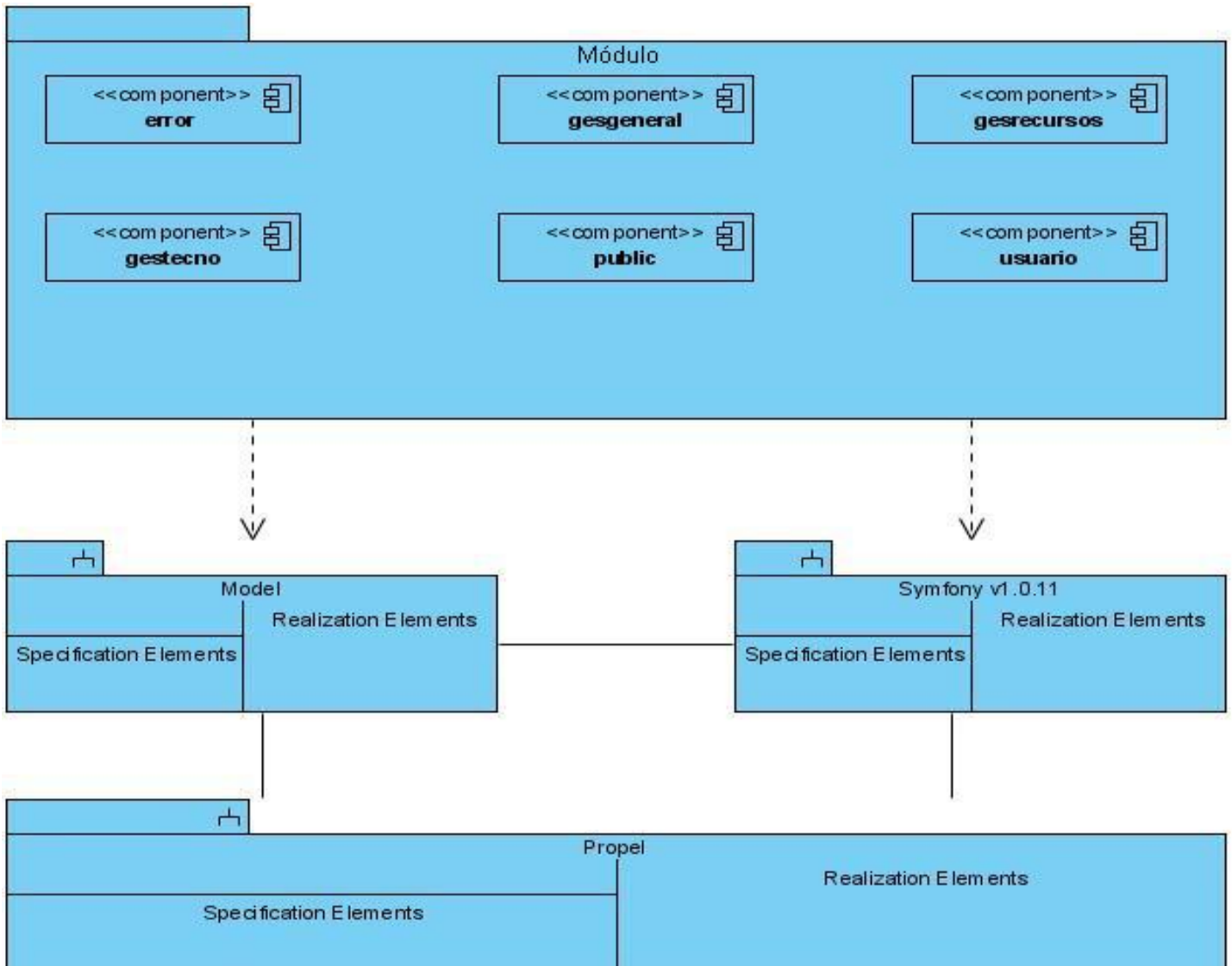


Figura 10. Diagrama de Componentes.

Para mejor comprensión del Diagrama de Componentes, se expondrá una vista detallada de los paquetes Model y Módulo.

I. Paquete Model.

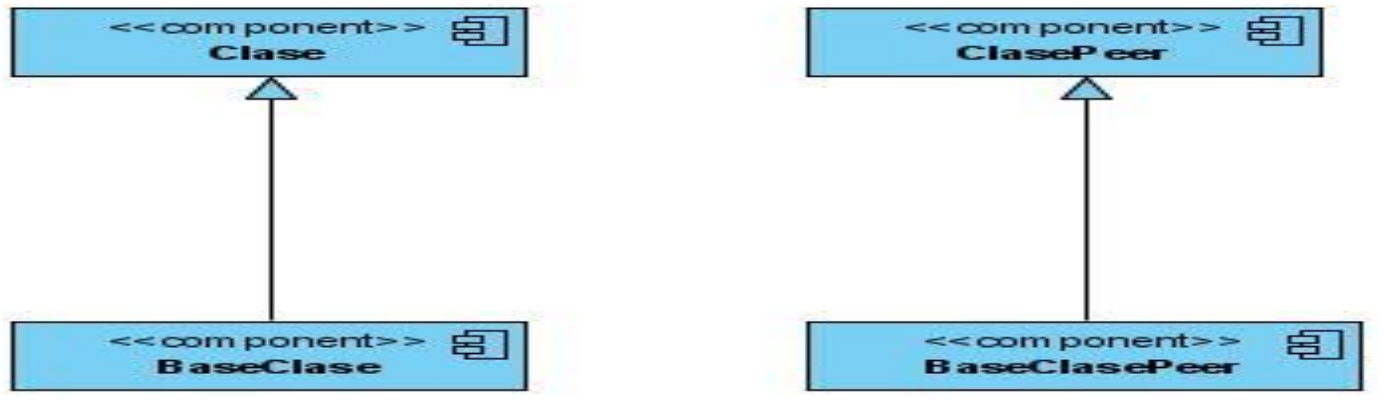


Figura 11. Paquete Model.

II. Paquete Módulo.

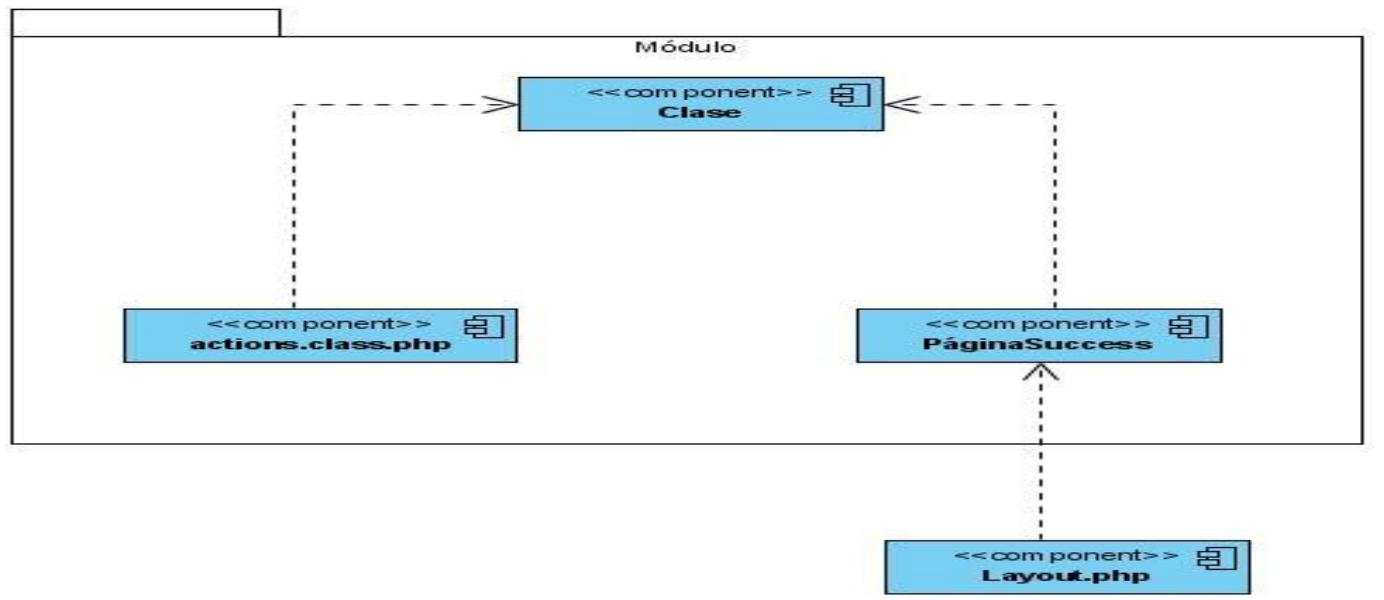


Figura 12. Paquete Módulo

3.8 Conclusiones.

En este capítulo se han representado los principales artefactos de los flujos de trabajo Análisis, Diseño e Implementación, partiendo del modelado del negocio y la especificación y realización detalladas de los casos de uso del sistema, quedando así creada la base sobre la cual ha de desarrollarse el sistema propuesto.

CONCLUSIONES

Con la implantación del sistema propuesto se logra optimizar el proceso que desarrolla la dirección de Tecnología en lo referente a la Gestión Tecnológica en la Universidad de las Ciencias Informáticas, puesto que cada una de las acciones que describen el proceso que requieren de tiempo, y de datos de los cuales se necesita tener una información precisa, con la utilización del sistema propuesto se logran realizar con mayor rapidez y mejor exactitud. Además de que la información se encuentra protegida, ya que sólo pueden acceder a ella aquellos que tienen los privilegios requeridos y la gestión de la misma se torna menos engorrosa; todo ello contribuye a que los datos sean mas consistentes y por ende mas confiables.

Mediante el uso de las herramientas adecuadas, se logró desarrollar un sistema de fácil manejo, seguro en la gestión de los datos, que brinda la posibilidad de un despliegue en la plataforma libre, garantizando un mayor tiempo de vida para el producto, que sea más reutilizable y por supuesto más extensible. El software que se propone en este trabajo como se ha fundamentado en capítulos anteriores con mayor profundidad, cumple con las exigencias del cliente, logrando de esta manera el objetivo planteado en la investigación y proporcionando calidad, rapidez y optimización al proceso de Gestión Tecnológica, brindando así la posibilidad, a partir de su plena utilización, de un despliegue más adecuado y organizado de los recursos tecnológicos en la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Sobre el presente trabajo los autores recomiendan:

- Someter el sistema a pruebas de calidad de software para su posterior instalación.
- Realizar un estudio de las posibilidades de integración otros subsistemas desarrollados en la Universidad de las Ciencias Informáticas, involucrados en el proceso de la informatización de servicios tecnológicos.
- Ampliar las funcionalidades del sistema de manera que pueda utilizarse en otras direcciones dentro de la Universidad de las Ciencias Informáticas que se involucran en los procesos a los que responde.
- Generalizar las funcionalidades de forma tal que su uso pueda extenderse a entidades análogas a la Universidad de las ciencias Informáticas en Cuba.

REFERENCIAS BIBLIOGRAFICAS

- [1] Concepto de Tecnología Obtenido en noviembre, 2008, de http://www.punchador.com/index.php?option=com_content&view=article&id=58&Itemid=76.
- [2] Blog de Mildred Aponte **creado para la materia Gestión de tecnología**. Obtenido en noviembre, 2008, de http://mildredaponte.blogspot.com/2009_03_01_archive.html.
- [3] Tecnología Informática de América (ITAA). Obtenido en noviembre, 2008, de <http://www.newspcs.com/5646/definiendo-la-blogesfera.html> .
- [4] Ciencia, Tecnología, Sociedad e Innovación en Iberoamérica (ALTEC). Obtenido en diciembre, 2008, de <http://www.oei.es/altecoei.htm>.
- [5] EUROWIN. Obtenido en noviembre, 2008, de <http://www.datacysoftware.com/software-informacion-empresa-corporativa.php>.
- [6] ZyLab. Obtenido en noviembre, 2008, de <http://www.zylab.es/>.
- [7] Tecnología de la Información y las Comunicaciones. Obtenido en diciembre, 2008, de http://www.conicet.gov.ar/becas/archivos_gral/2005/listado_areas_vacancia/tecn_info_y_comunic.do
c.
- [8] Tareas TICs. Obtenido en diciembre, 2008, de <http://ucenavalenzuela.blogspot.com/2009/04/informe-tics.html>.
- [9] Sistemas de Gestión. Obtenido en diciembre, 2008, de <http://www.dnv.es/certificacion/sistemasdegestion/index.asp>.
- [10] Desarrollo de Aplicaciones. Obtenido en enero, 2009, de <http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>.
- [11] Servidor Web. Obtenido en enero, 2009, de <http://www.brendingdesigners.com.ar/articulos/servidorweb.html>.
- [12] Apache. Obtenido en enero, 2009, de <http://www.arcos.inf.uc3m.es/~folcina/pfc-html/node34.html>.
- [13] ISS (Internet Information Services). Obtenido en enero, 2009, de <http://edgarrairez.wordpress.com/2008/08/19/iis-internet-information-server/> .
- [14] Lenguaje de Programación. Obtenido en enero, 2009, de <http://entren.dgsca.unam.mx/introduccion/lenguajes.html>.
- [15] Los diferentes lenguajes de programación para la web. Obtenido en enero, 2009, de <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
- [16] Los diferentes lenguajes de programación para la web. Obtenido en enero, 2009, de <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

- [17] Los diferentes lenguajes de programación para la web. Obtenido en enero, 2009, de <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
- [18] XHTML. Obtenido en enero, 2009, de <http://www.hipertexto.info/documentos/xhtmll.htm>.
- [19] Guía breve de CSS. Obtenido en enero, 2009, de <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
- [20] Procesos de Desarrollo-Asociación javaHispano. Obtenido en enero, 2009, de http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/.
- [21] Rup. Obtenido en enero, 2009, de <http://www.scribd.com/doc/13071186/Rup>.
- [22] Extreme Programming -Dos Ideas. Obtenido en enero, 2009, de http://www.dosideas.com/wiki/Extreme_Programming.
- [23] Herramientas para modelado UML. Obtenido en enero, 2009, de <http://paraisolinux.com.ar/herramientas-para-modelado-uml/>.
- [24] Guía Herramientas CASE. Obtenido en enero, 2009, from <http://members.fortunecity.com/software1/herramie.htm>.
- [25] Sistemas de Gestión de Bases de Datos. Obtenido en enero, 2009, de <http://rucas23.wordpress.com/2008/04/11/sistema-de-gestion-de-base-de-datos/>.
- [26] Instituto de Informática y Sistemas-Programación Web. Obtenido en enero, 2009, de http://www.isi.edu.ni/index.php?option=com_content&task=view&id=29&Itemid=55.
- [27] Herramientas Case. Obtenido en enero, 2009, de <http://www.mitecnologico.com/Main/HerramientasCase>.
- [28] Documento de Arquitectura de Software. Obtenido en febrero, 2009, de http://74.125.93.132/search?q=cache:hzbG2Q8_cAJ:janalizercode-aqp-epis-destroyer.googlecode.com/files/informe%25201.1.doc.
- [29] Prado: Framework para PHP5. Obtenido en febrero, 2009, de <http://sentidoweb.luissacristan.com/2007/08/28/prado-framework-para-php5/>.
- [30] ingsoft: Entorno Integrado de Desarrollo. Obtenido en febrero, 2009, de <http://ingpau.blogspot.com/2007/09/entorno-de-desarrollo-integrado.html>.
- [31] Zend Studio. Obtenido en febrero, 2009, de <http://www.desarrolloweb.com/articulos/1178.php>.
- [32] Eclipse. Obtenido en febrero, 2009, de <http://www.webmaster-mexico.com/eclipse>.

BILIOGRAFIA CONSULTADA

Martino, P. C. P. Diseño Teórico de la Investigación Científica. Cuba.

LARMAN, C. UML y PATRONES. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, 1999. 536 p.

Midel A. Remedios Prado. La computación en Cuba. Disponible en <http://www.cmg.jovenclub.cu/munic/cesp/index.php> .

Rodolfo Faloh Bejerano. Bases de la Gestión Tecnológica en Cuba. Disponible en http://www.innred.net/iber/Eventos/1996/Confer_004.htm.

HOJTSY, G.; Á. PARDO, et al. Manual de PHP, 2006. [2009]. Disponible en: <http://www.php.net/manual/es/>

DESARROLLOWEB. Programación en PHP, 2007. [2009]. Disponible en: <http://www.desarrolloweb.com/manuales/12/>

NAVARRA, U. D. Tutorial de JavaScript 2007. [2009]. Disponible en: <http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>

GROUP, P. G. D. *Sitio oficial de PostgreSQL*, 2008 [Disponible en: <http://www.postgresql.org/>]

POSTGRESQL, E. E. D. D. D. *Tutorial de PostgreSQL*, 2002.

Potencier, F., & Zaninotto, F. (2008). *Symfony la guia definitiva*.

Programación extrema. [cited 2009 enero]; Available de: www.programacionextrema.org.

Guerrero, L. A. "Arquitectura Física. Diagramas de Despliegue." Obtenido marzo, 2009, de www.dcc.uchile.cl/~luguerre/cc61j/recursos/clase12.ppt.

Vila, A. F. "Diagrama de Despliegue." Obtenido marzo, 2009, de <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.

Figueroa, P. "Conceptos en un Diagrama de Implementación." Obtenido marzo, 2009, de <http://www.cs.ualberta.ca/~pfiguero/>

(2005). "Prado." Obtenido febrero, 2009, de <http://www.xisc.com/>.

"Eclipse." Obtenido febrero, 2009, de <http://beta.selfplatform.eu/SELF/viewL?ssid=3672&cssid=4548>.

"Zend Studio." Obtenido febrero, 2009, de <http://www.desarrolloweb.com/articulos/1178.php> .

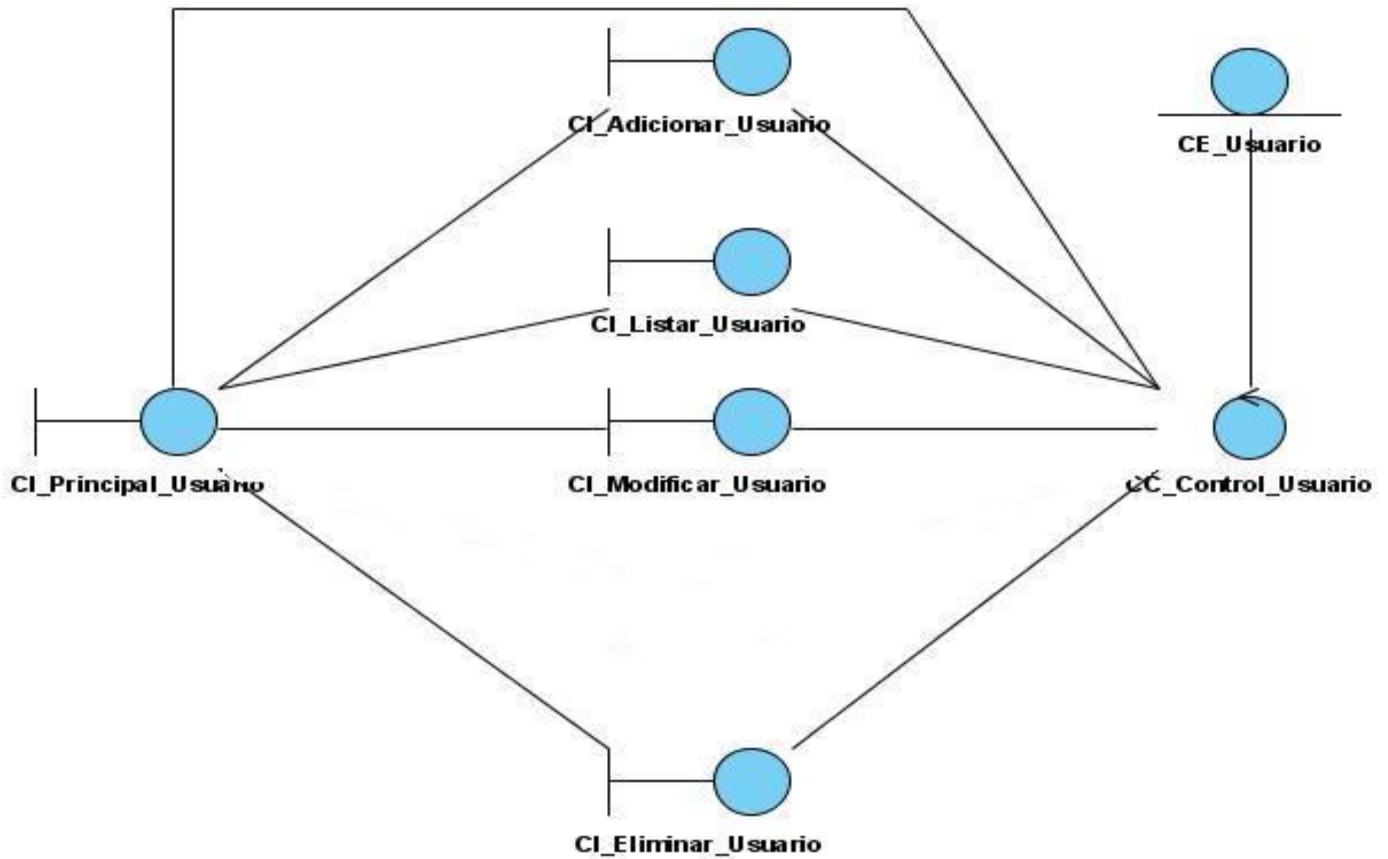
ANEXOS

Anexos 1. Diagrama de Clases del Análisis.

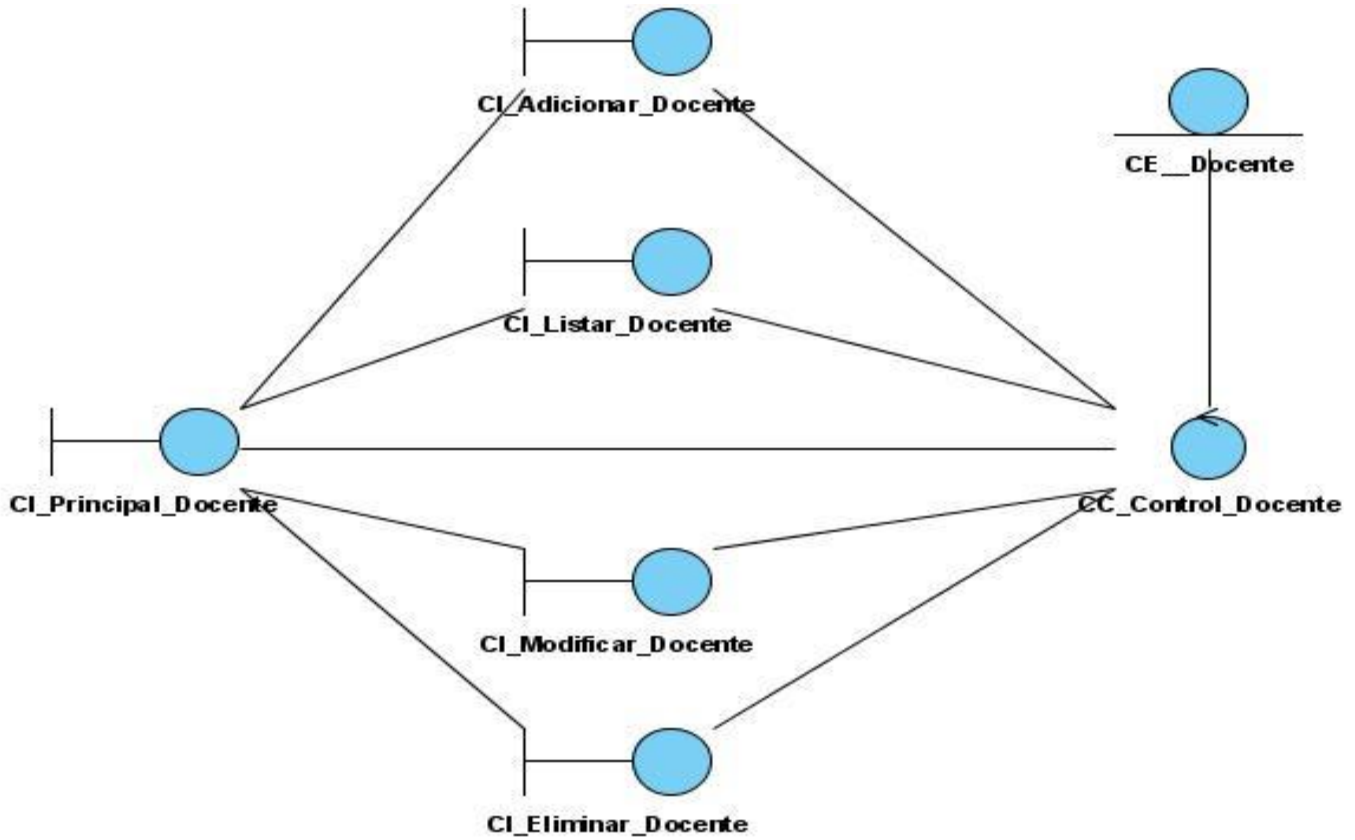
Diagramas de Clases del Análisis del Caso de Uso: Autenticar Usuario.



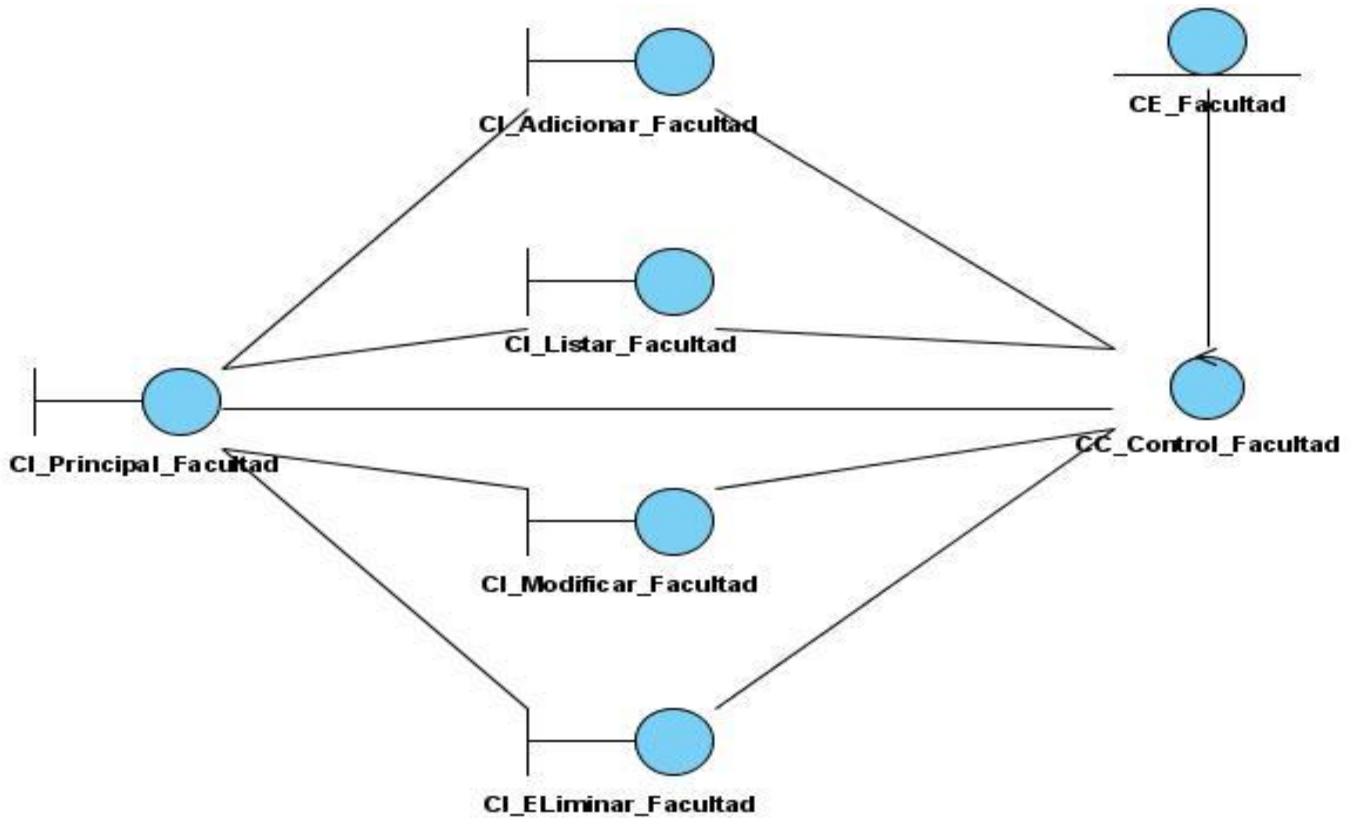
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Usuario.



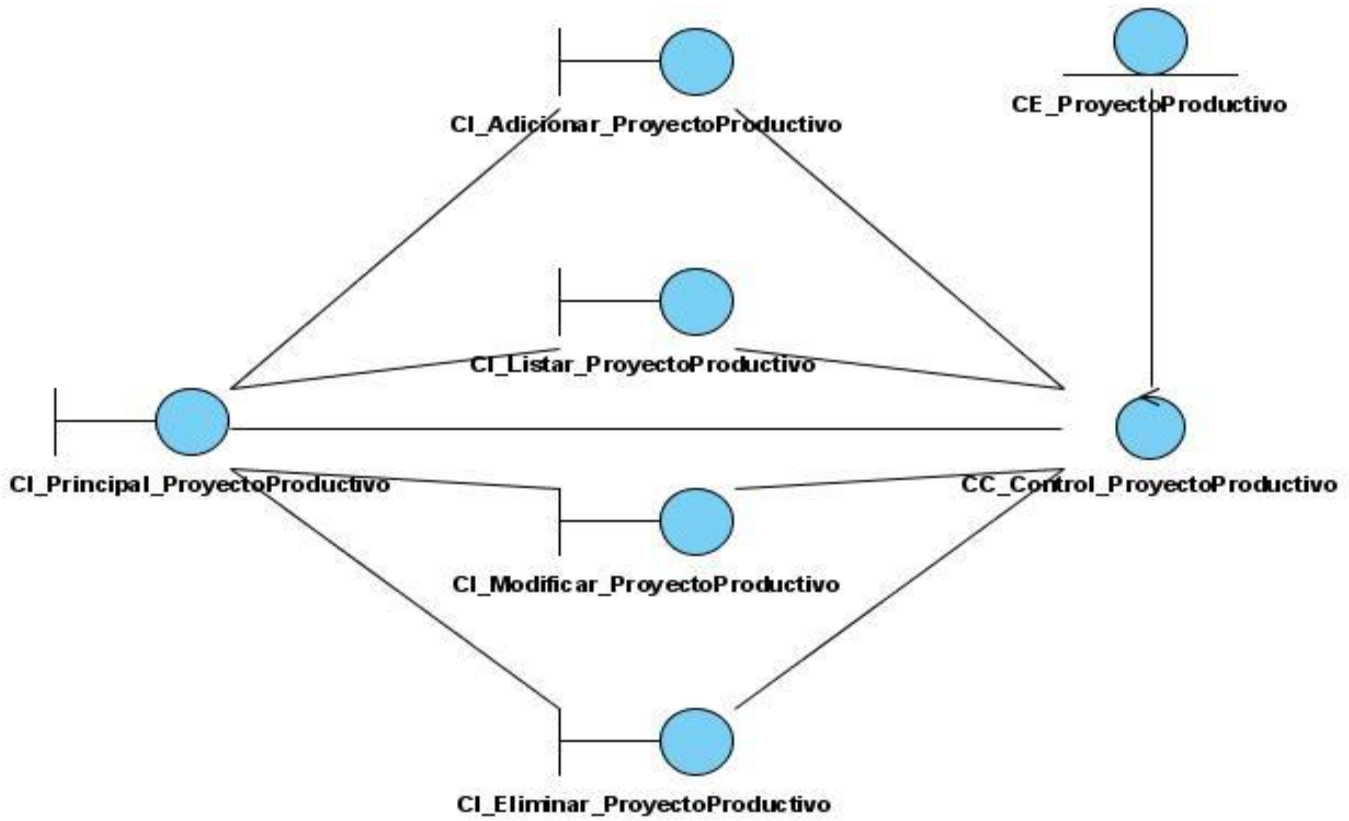
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Docente.



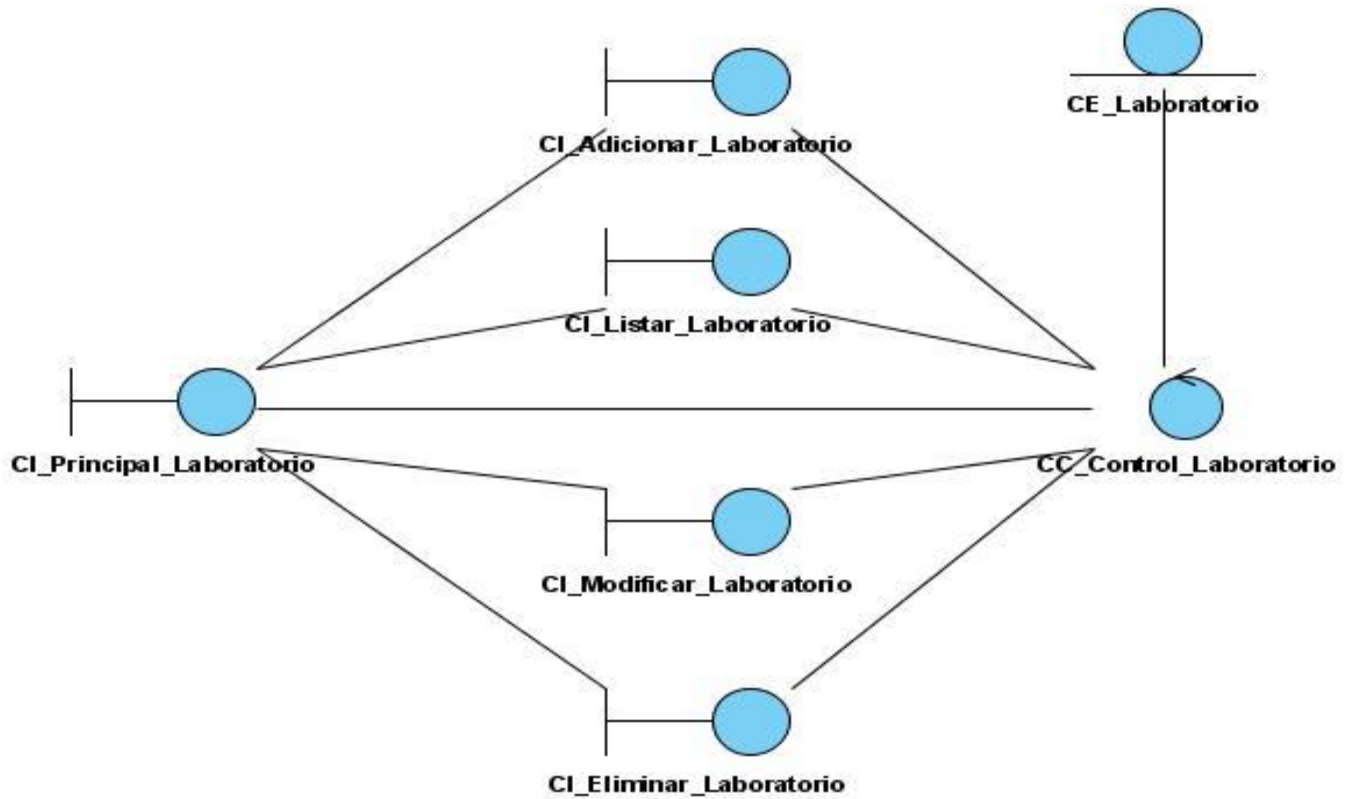
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Facultad.



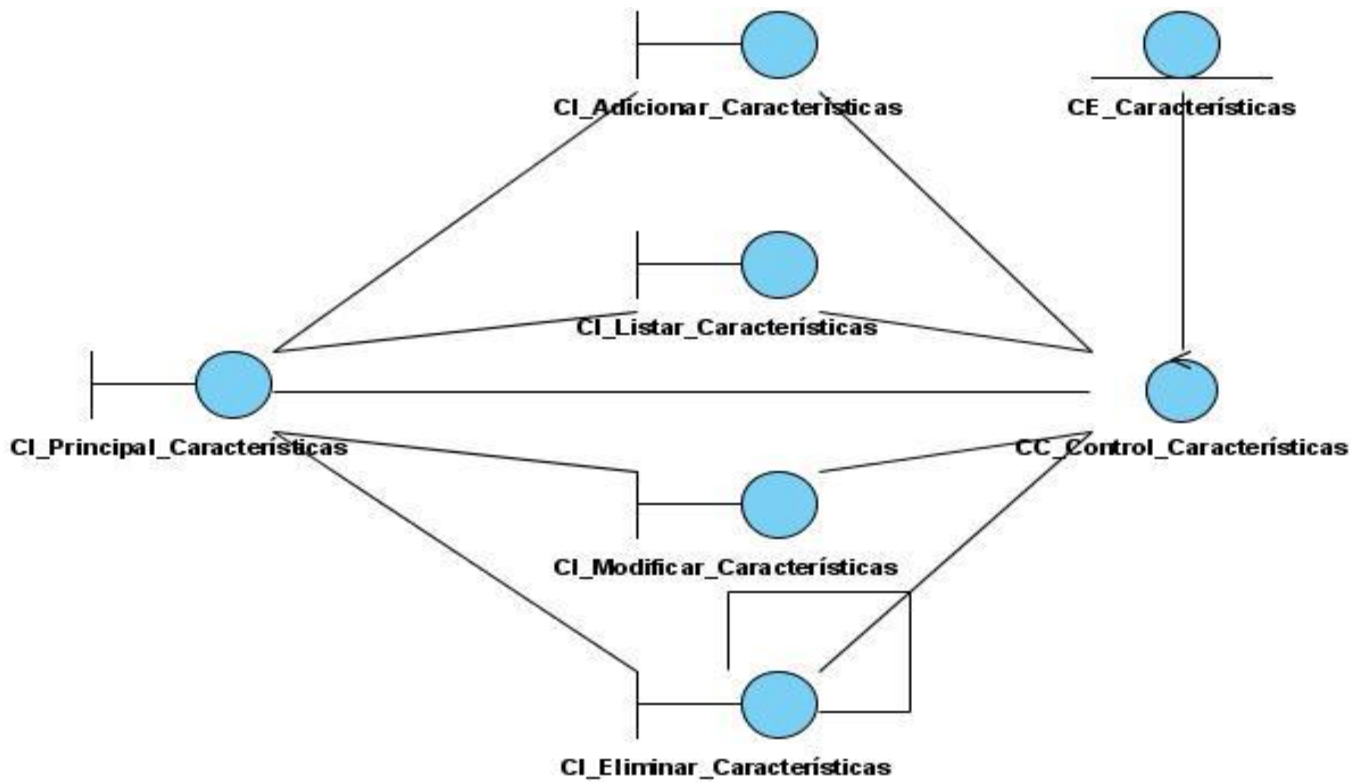
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Proyecto Productivo.



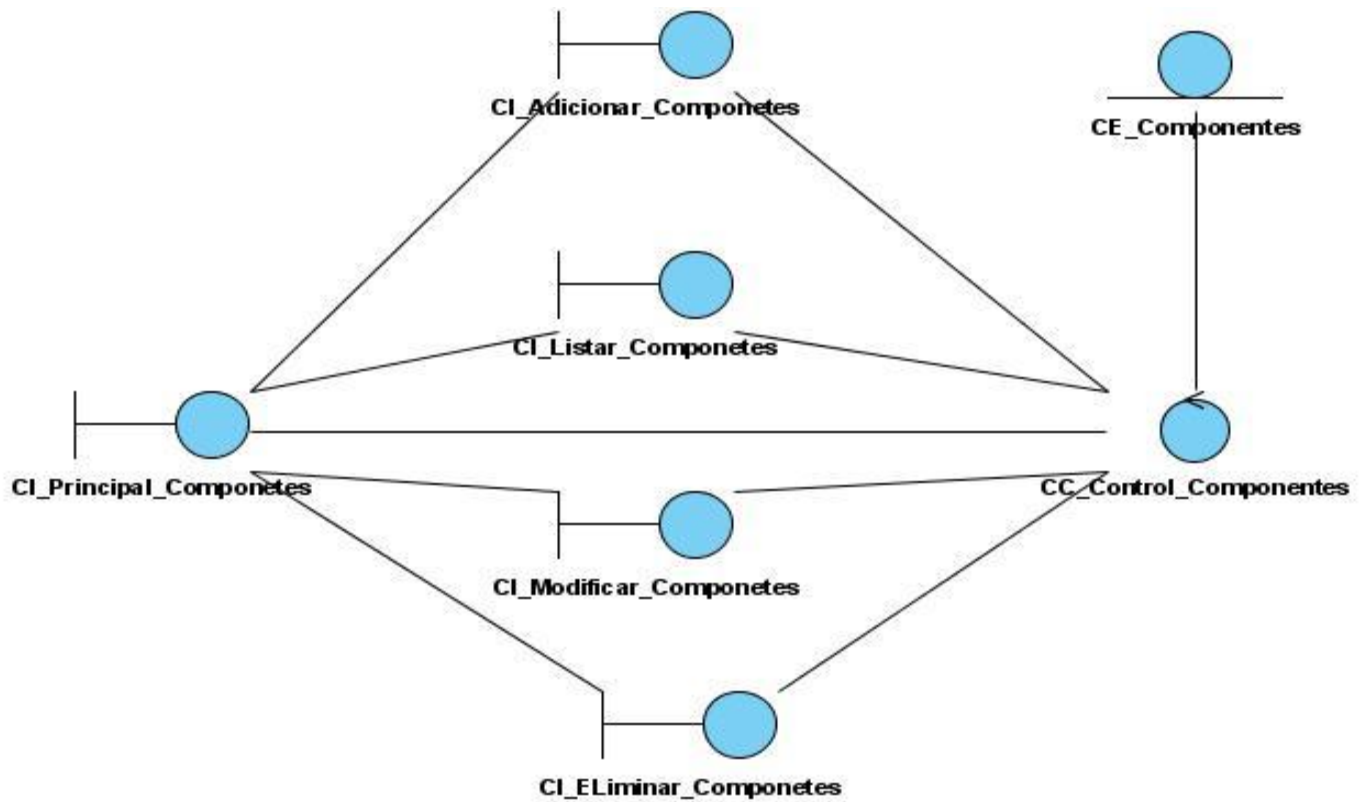
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Laboratorios.



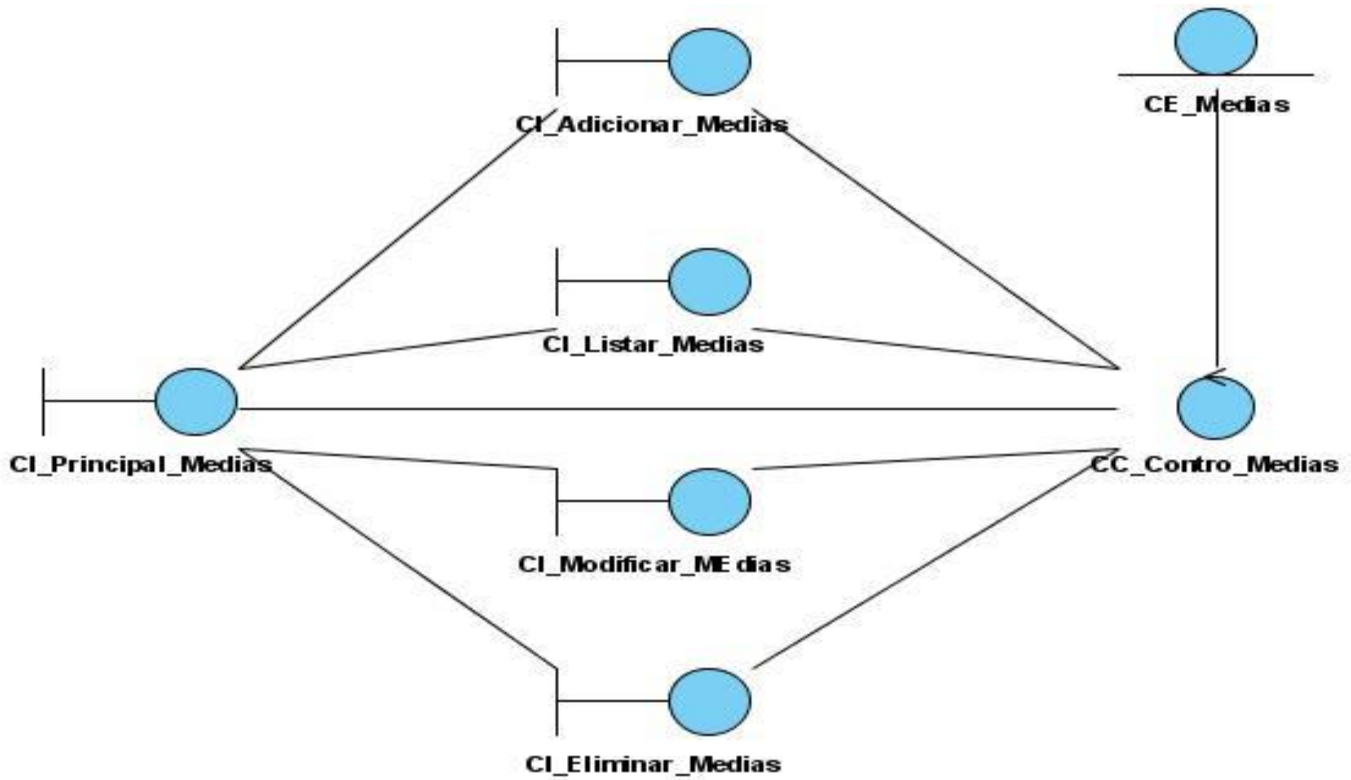
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Características.



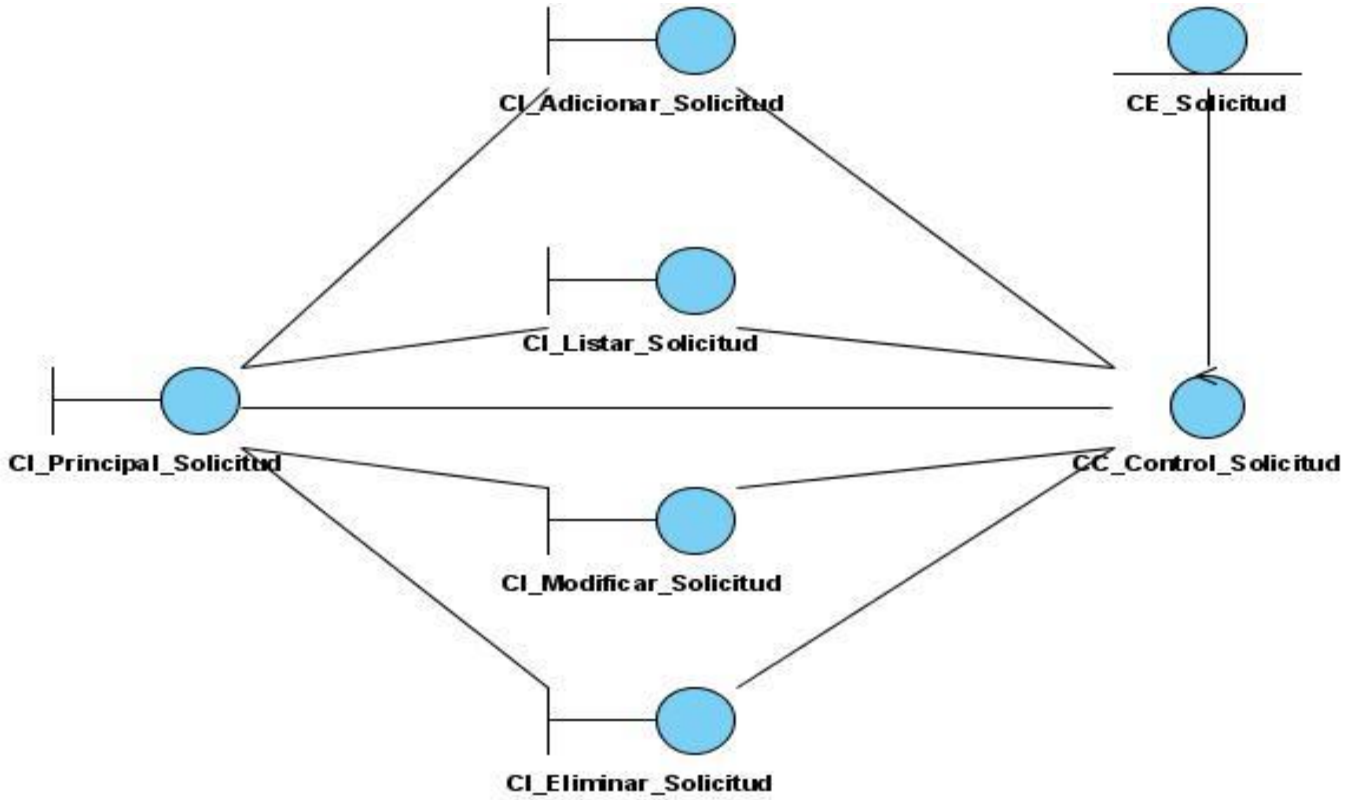
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Componentes.



Diagramas de Clases del Análisis del Caso de Uso: Gestionar Medidas.

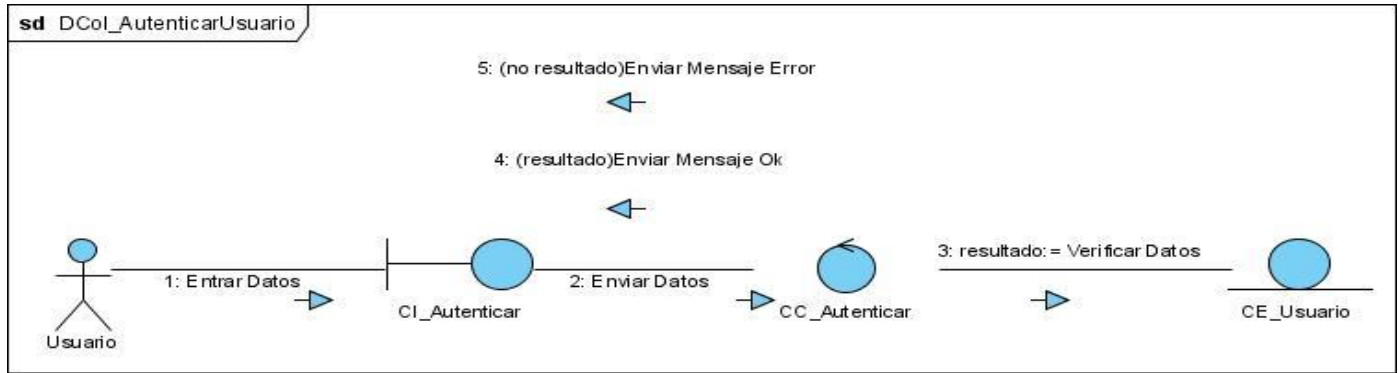


Diagramas de Clases del Análisis del Caso de Uso: Gestionar Solicitud.

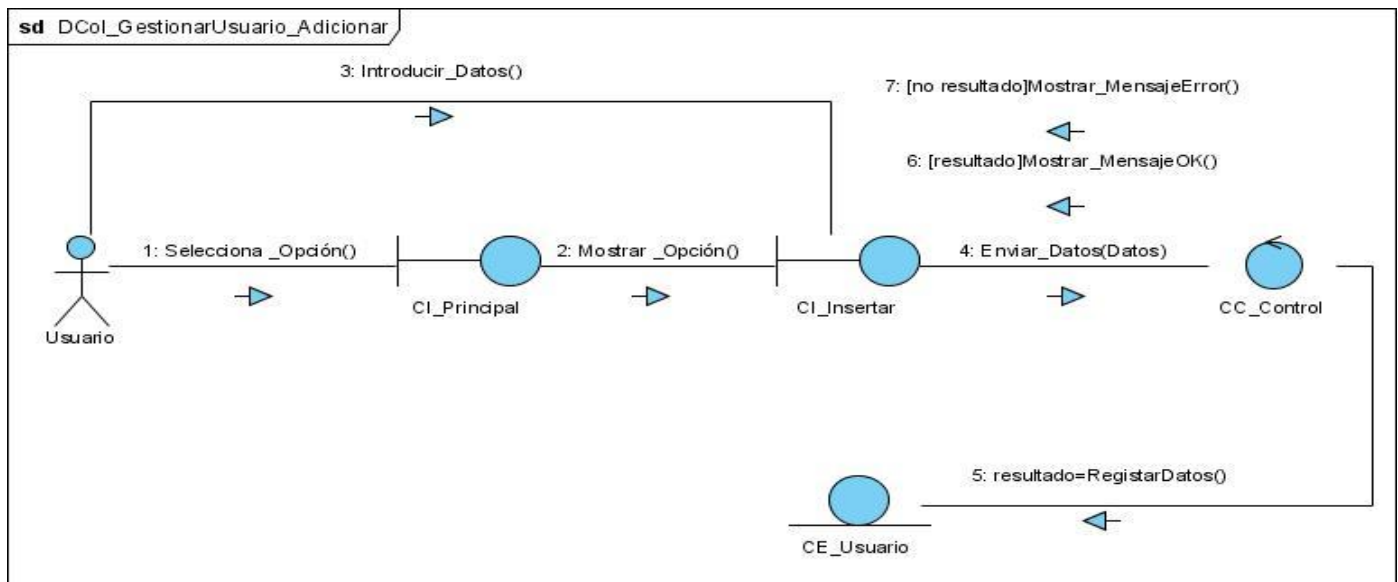


Anexos 2. Diagramas de Colaboración

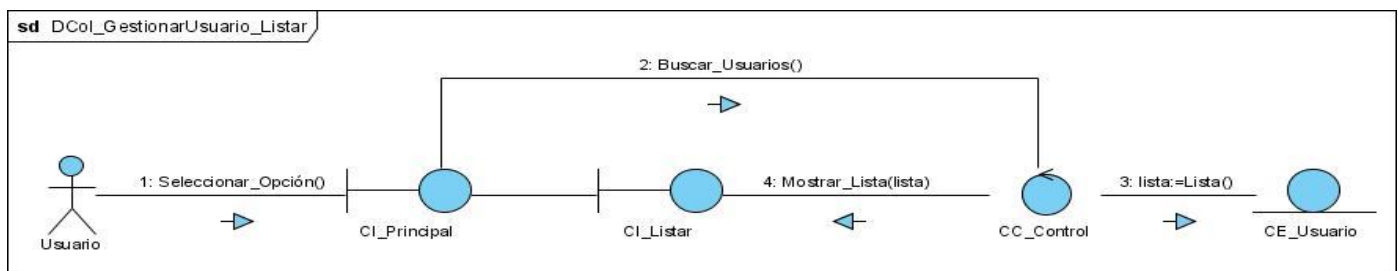
Diagramas de Colaboración: Autenticar Usuario.



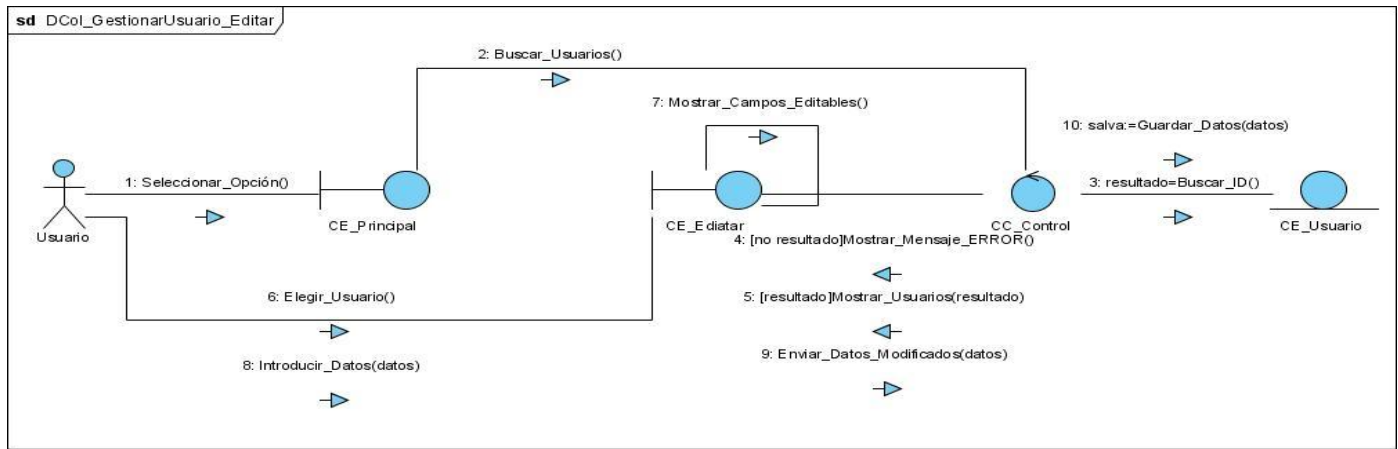
Diagramas de Colaboración: Gestionar Usuario (Sección Adicionar).



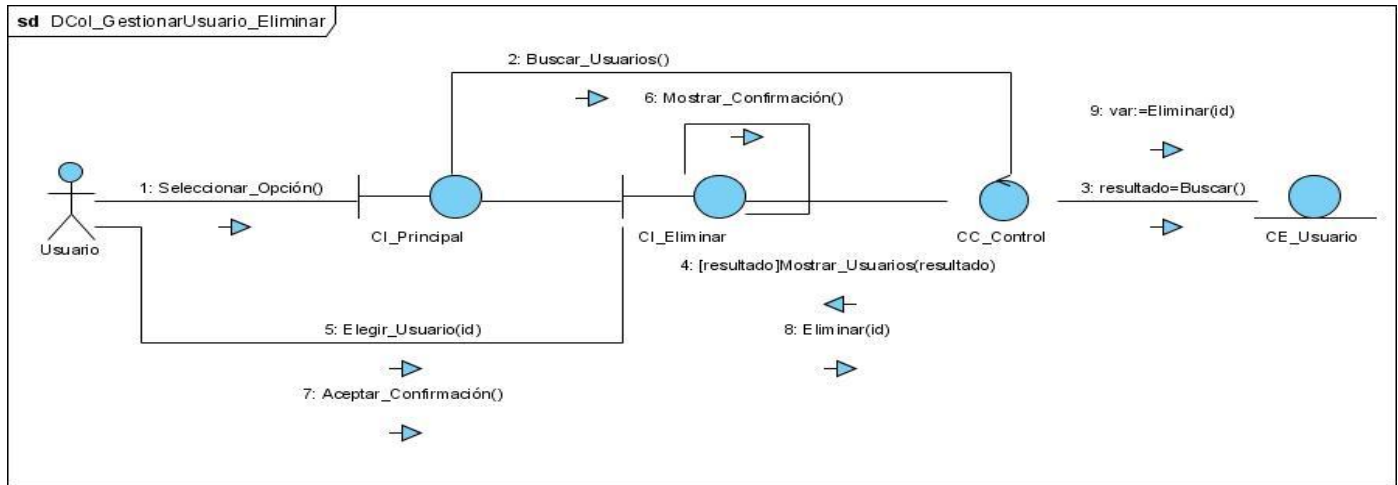
Diagramas de Colaboración: Gestionar Usuario (Sección Listar).



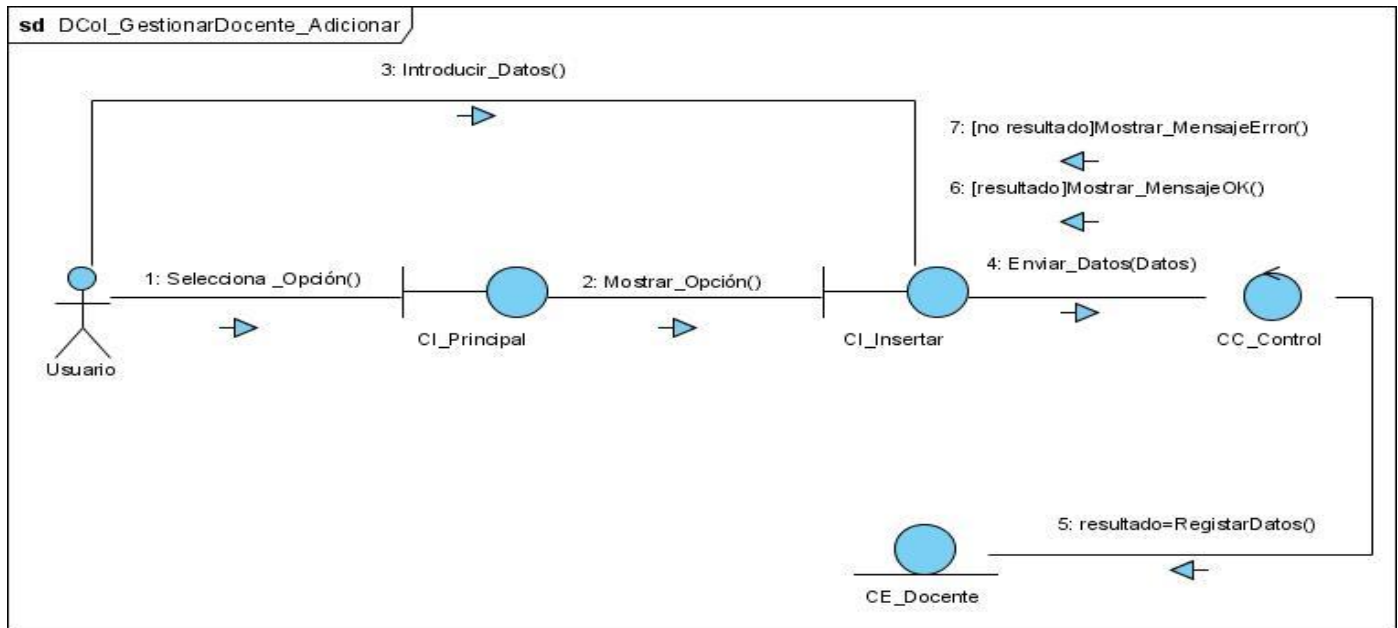
Diagramas de Colaboración: Gestionar Usuario (Sección Modificar).



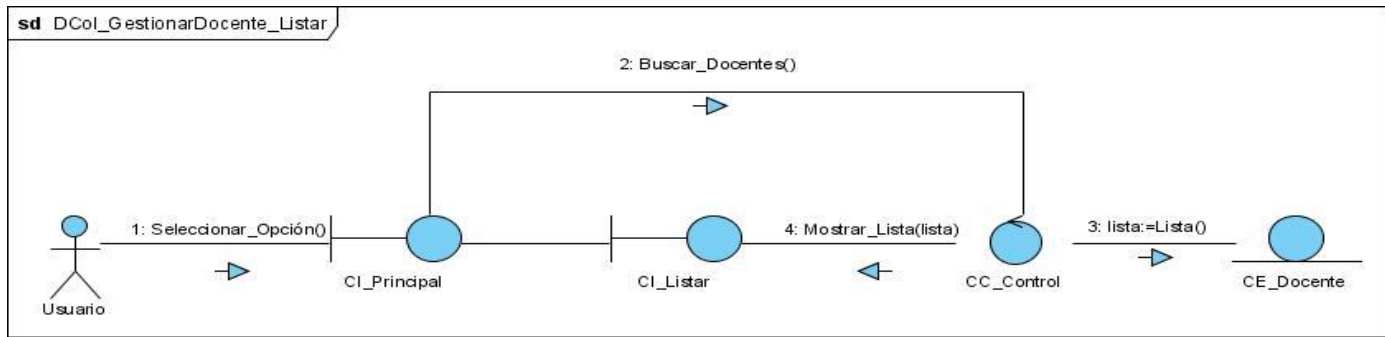
Diagramas de Colaboración: Gestionar Usuario (Sección Eliminar).



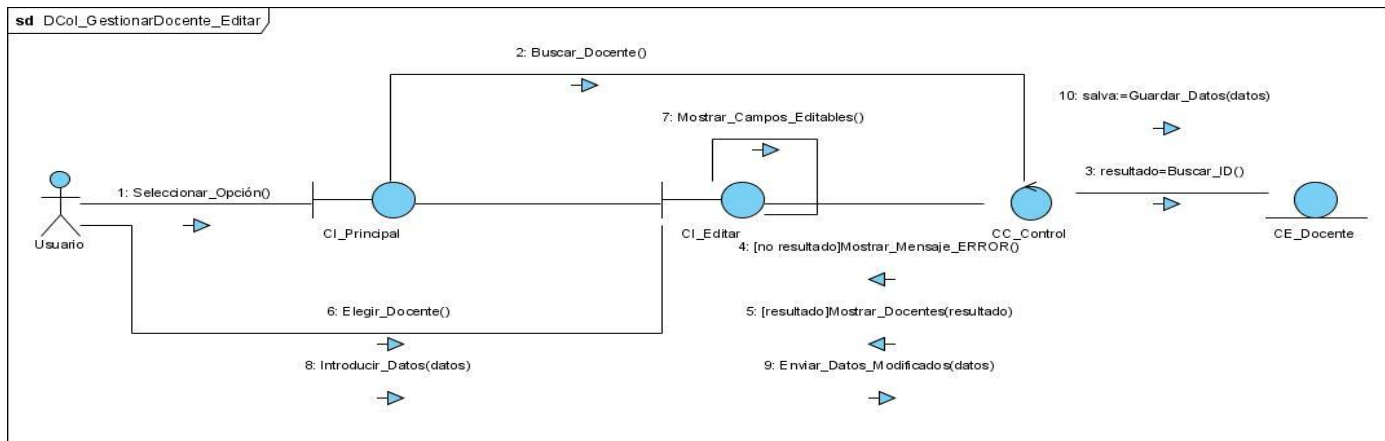
Diagramas de Colaboración: Gestionar Docente (Sección Adicionar).



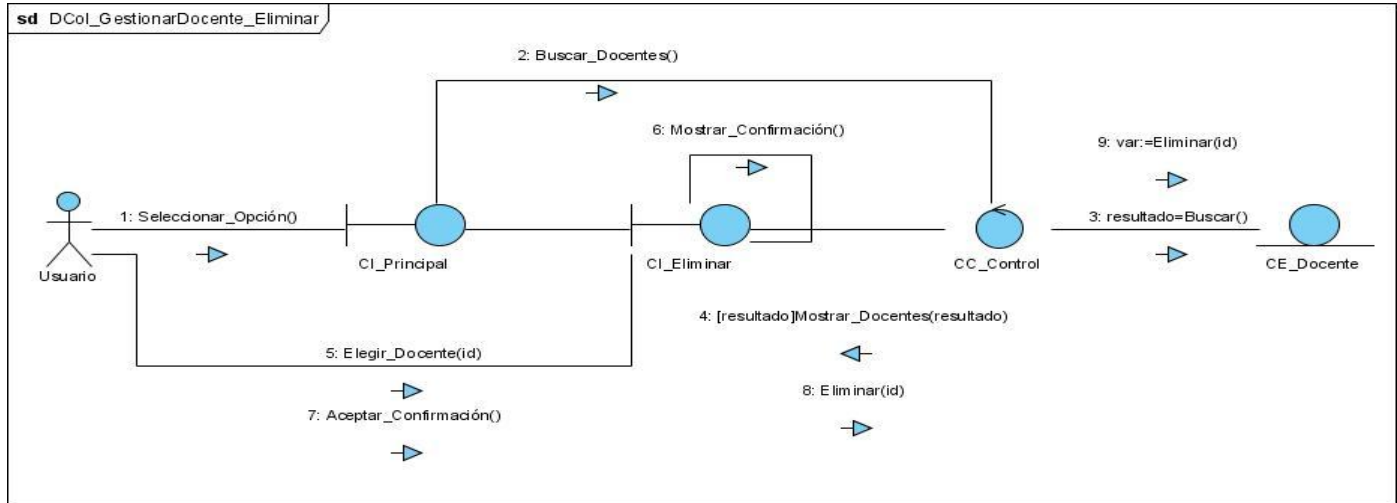
Diagramas de Colaboración: Gestionar Docente (Sección Listar).



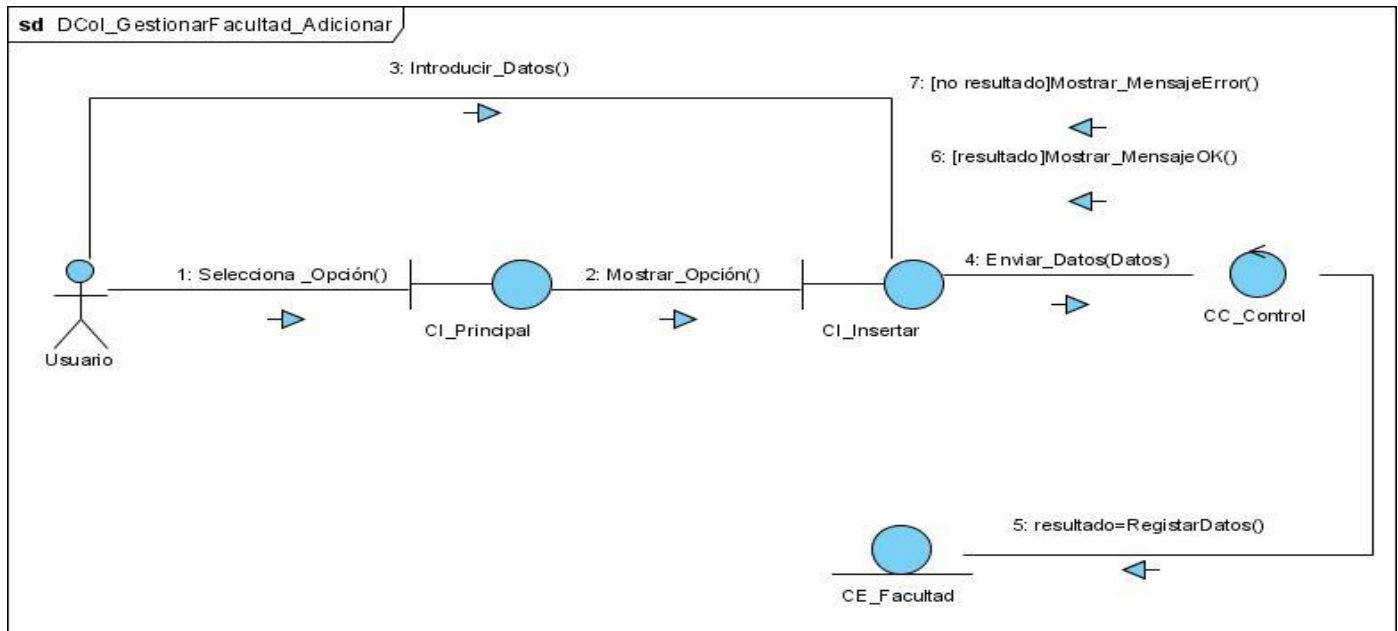
Diagramas de Colaboración: Gestionar Docente (Sección Modificar).



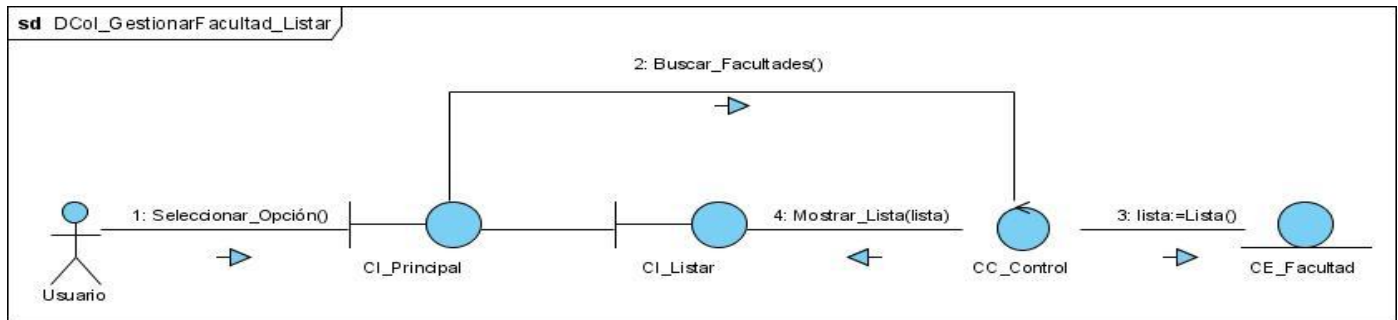
Diagramas de Colaboración: Gestionar Docente (Sección Eliminar).



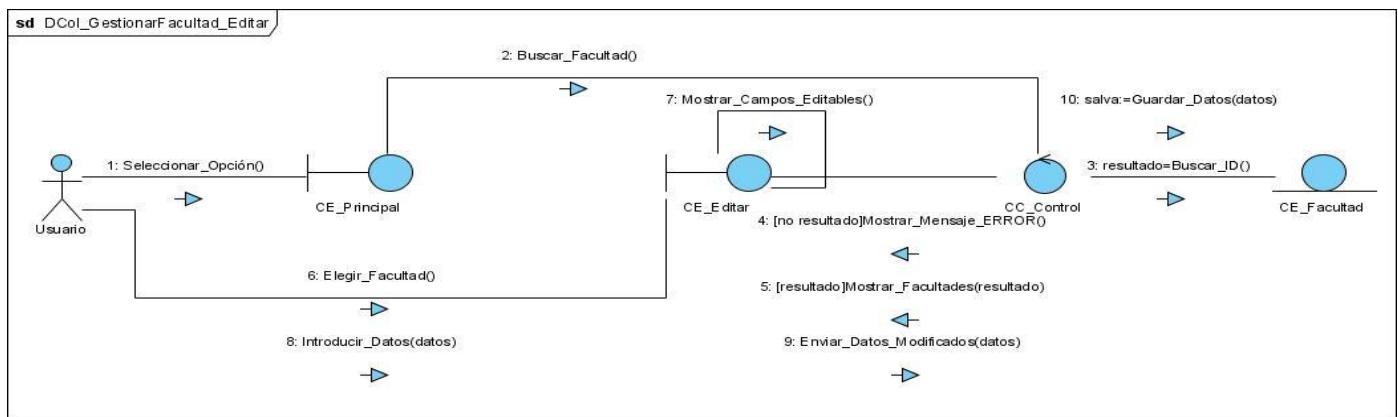
Diagramas de Colaboración: Gestionar Facultad (Sección Adicionar).



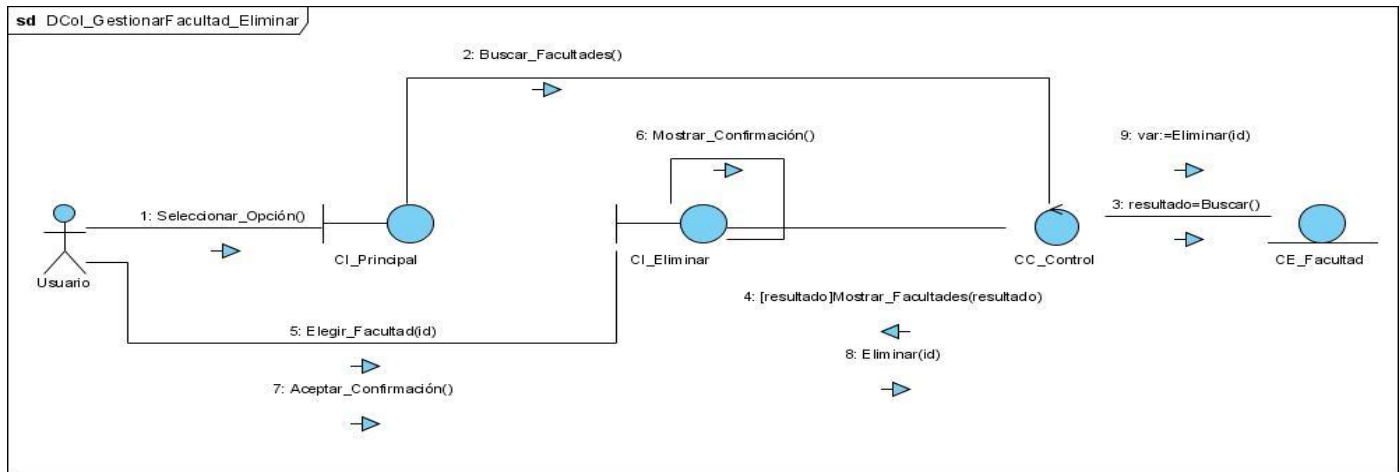
Diagramas de Colaboración: Gestionar Facultad (Sección Listar).



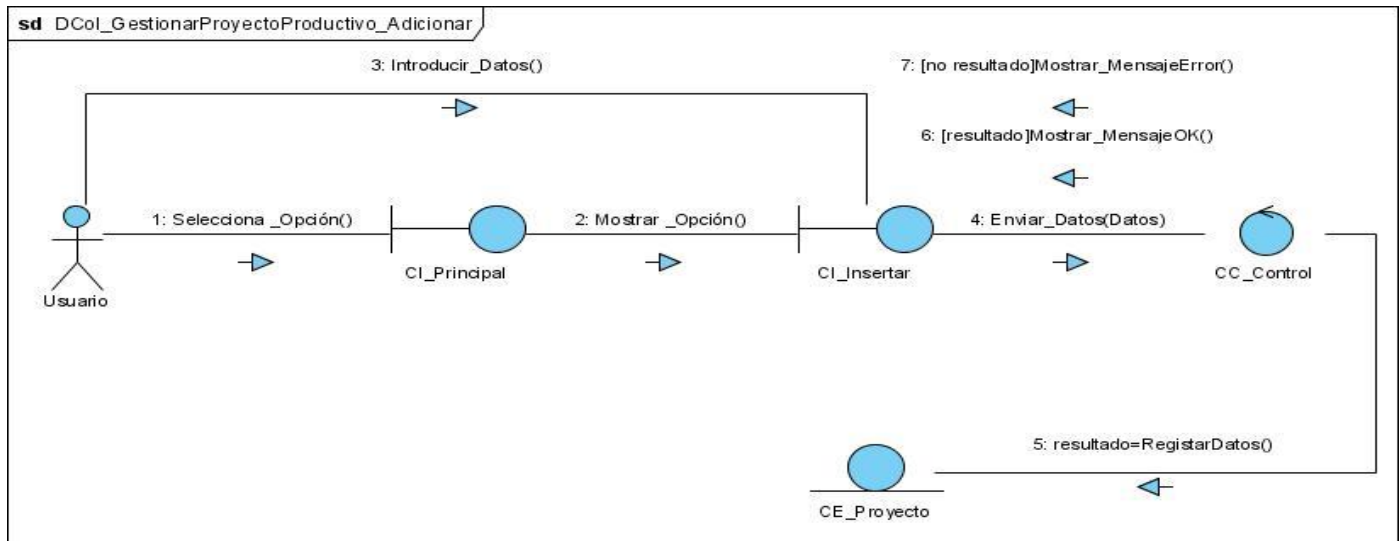
Diagramas de Colaboración: Gestionar Facultad (Sección Modificar).



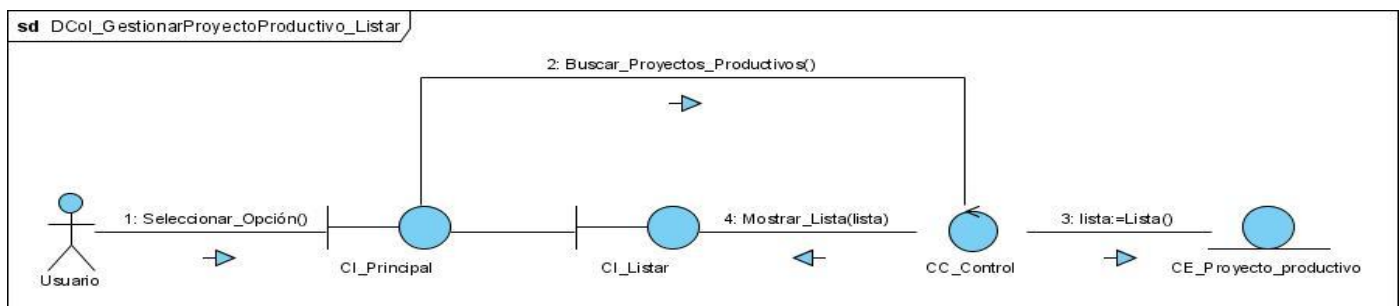
Diagramas de Colaboración: Gestionar Facultad (Sección Eliminar).



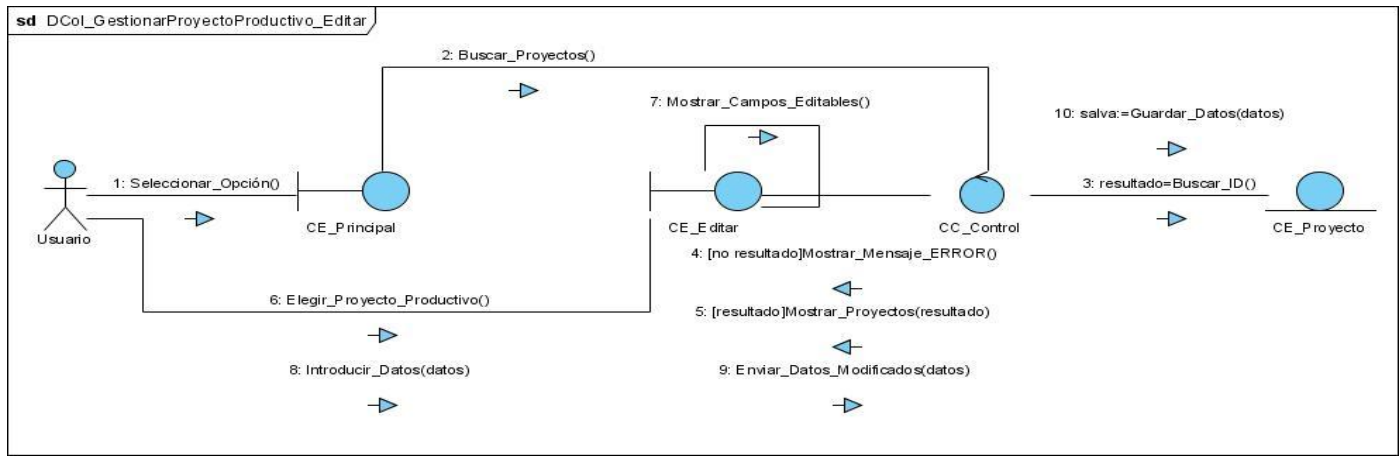
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Adicionar).



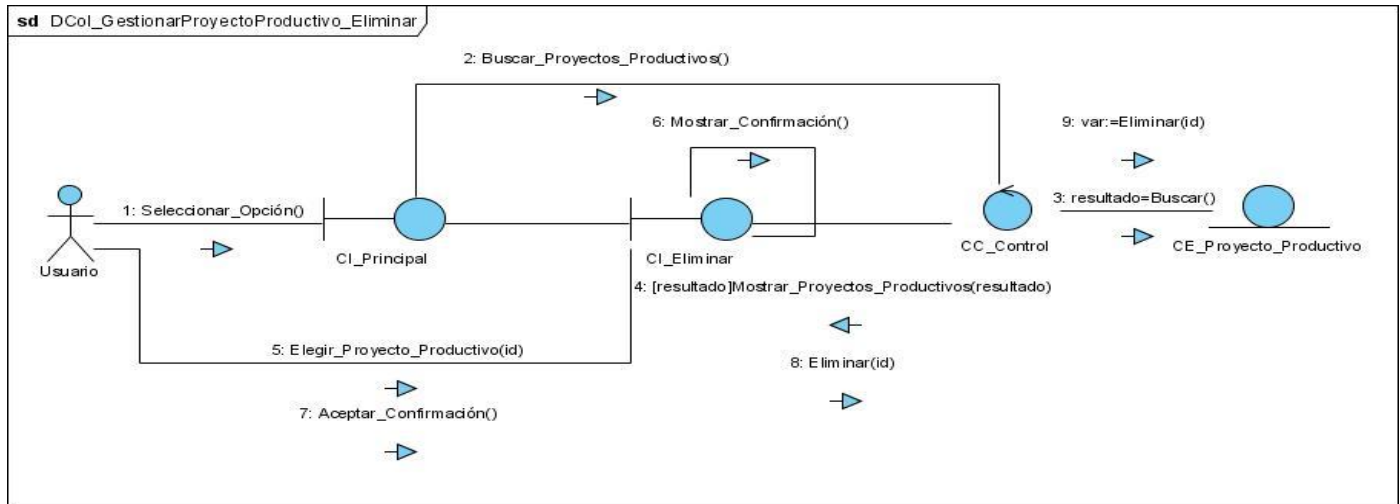
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Listar).



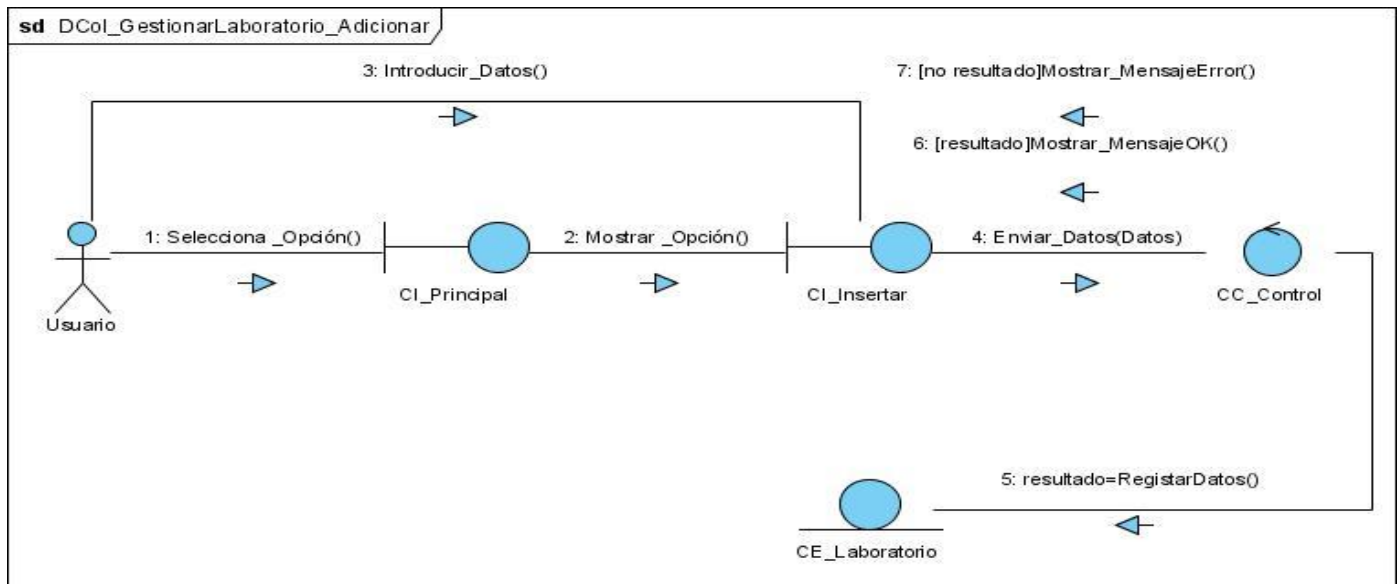
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Modificar).



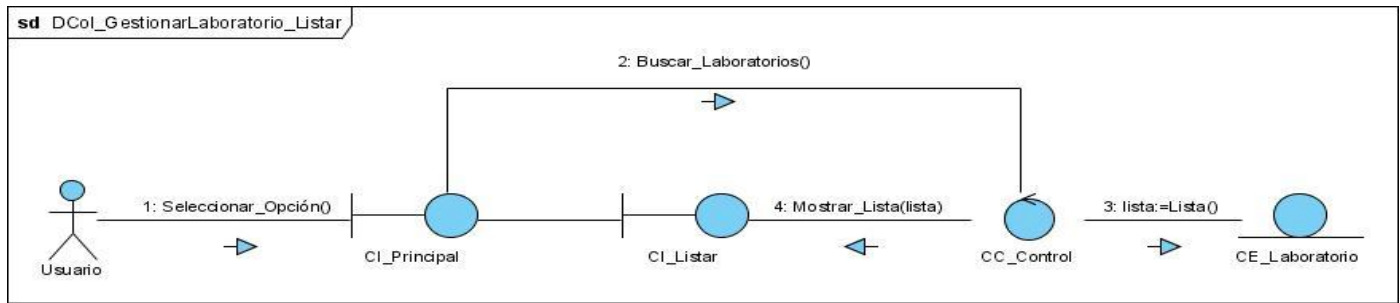
Diagramas de Colaboración: Gestionar Proyecto Productivo (Sección Eliminar).



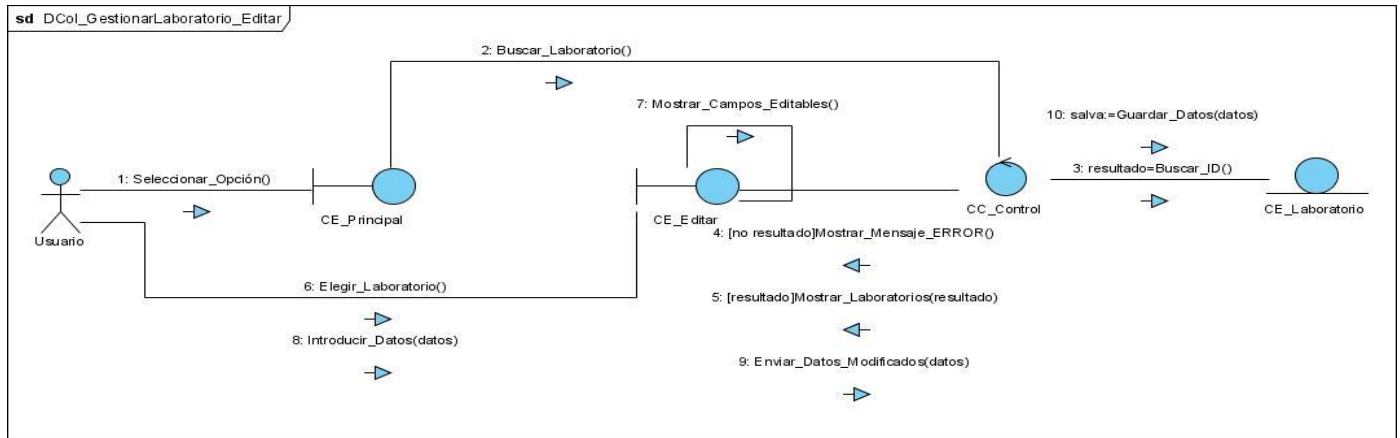
Diagramas de Colaboración: Gestionar Laboratorios (Sección Adicionar).



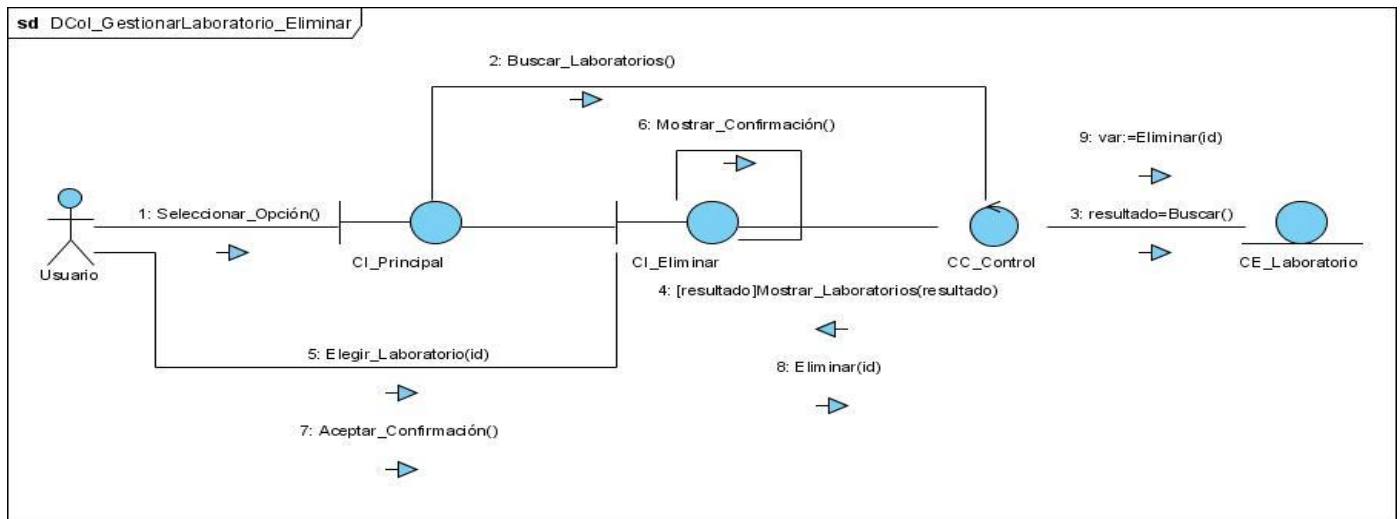
Diagramas de Colaboración: Gestionar Laboratorios (Sección Listar).



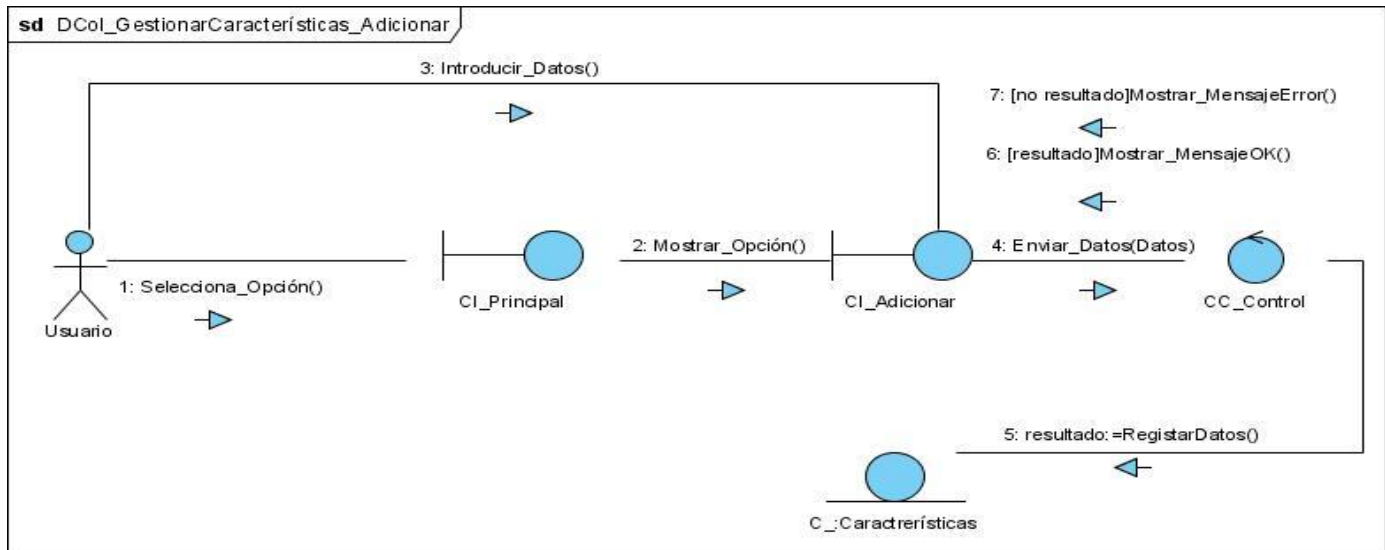
Diagramas de Colaboración: Gestionar Laboratorios (Sección Modificar).



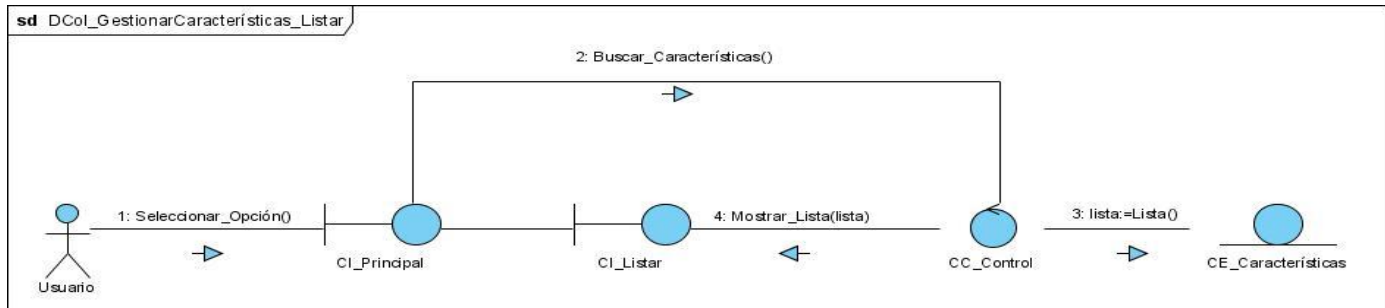
Diagramas de Colaboración: Gestionar Laboratorios (Sección Eliminar).



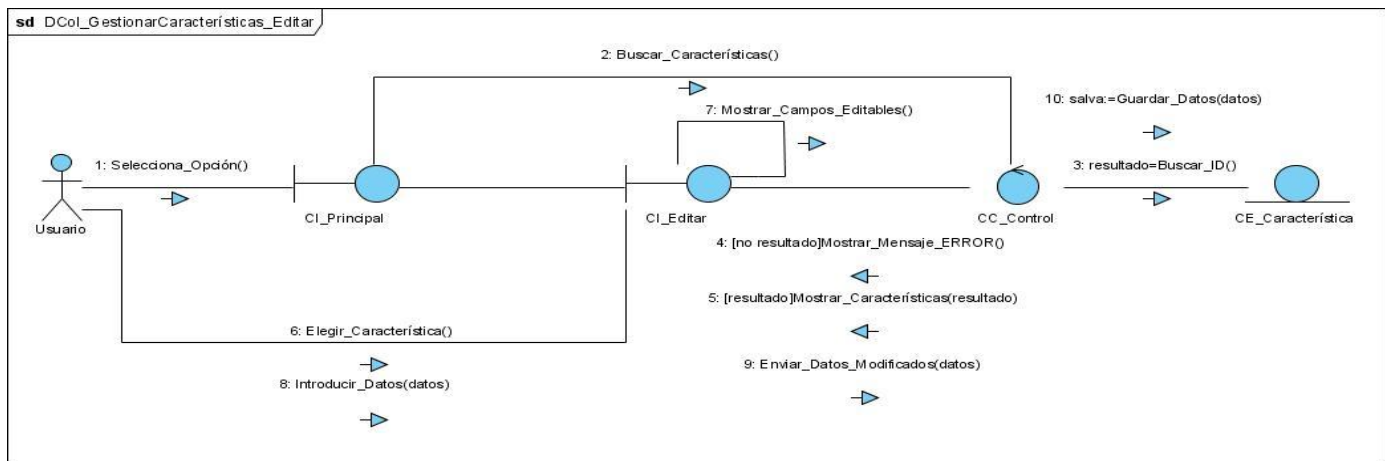
Diagramas de Colaboración: Gestionar Características (Sección Adicionar).



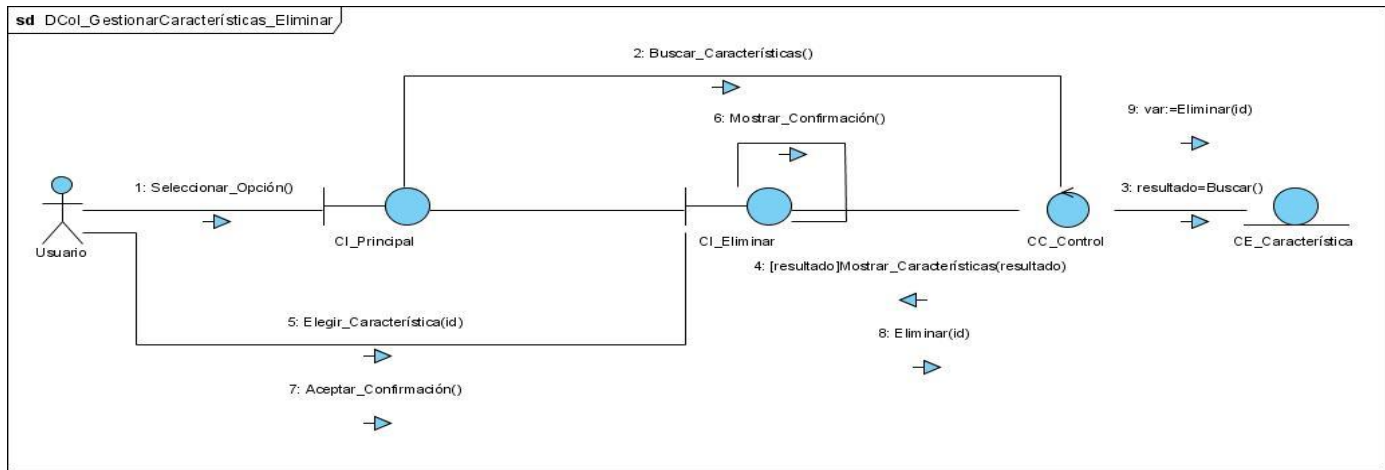
Diagramas de Colaboración: Gestionar Características (Sección Listar).



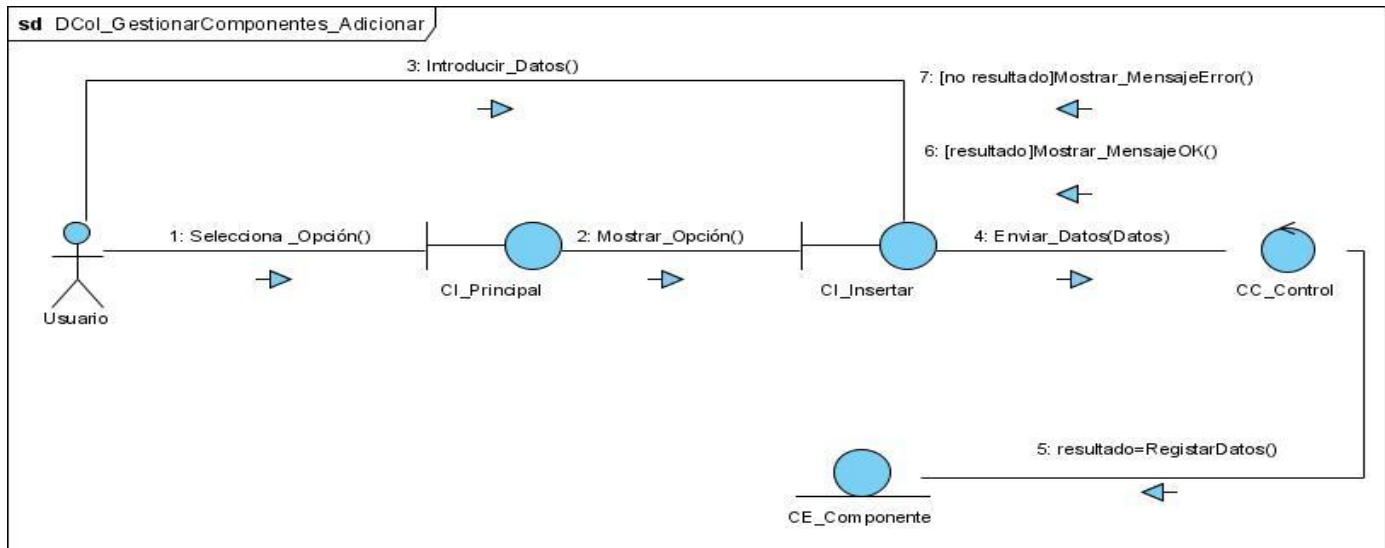
Diagramas de Colaboración: Gestionar Características (Sección Modificar).



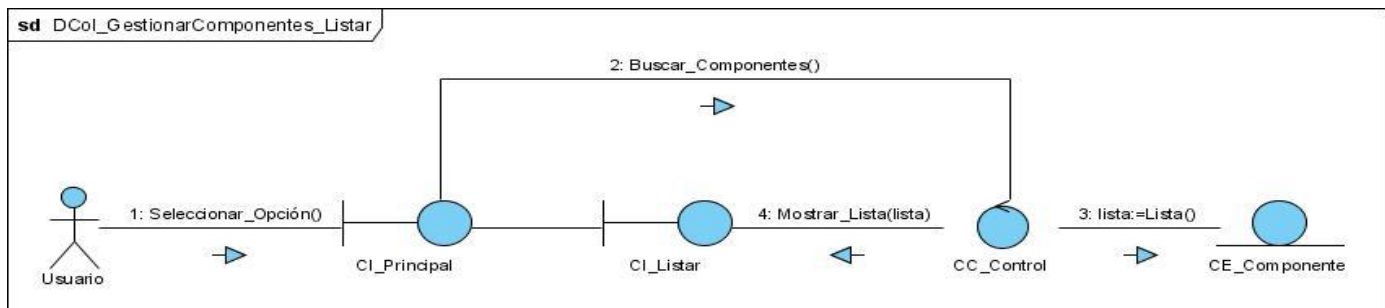
Diagramas de Colaboración: Gestionar Características (Sección Eliminar).



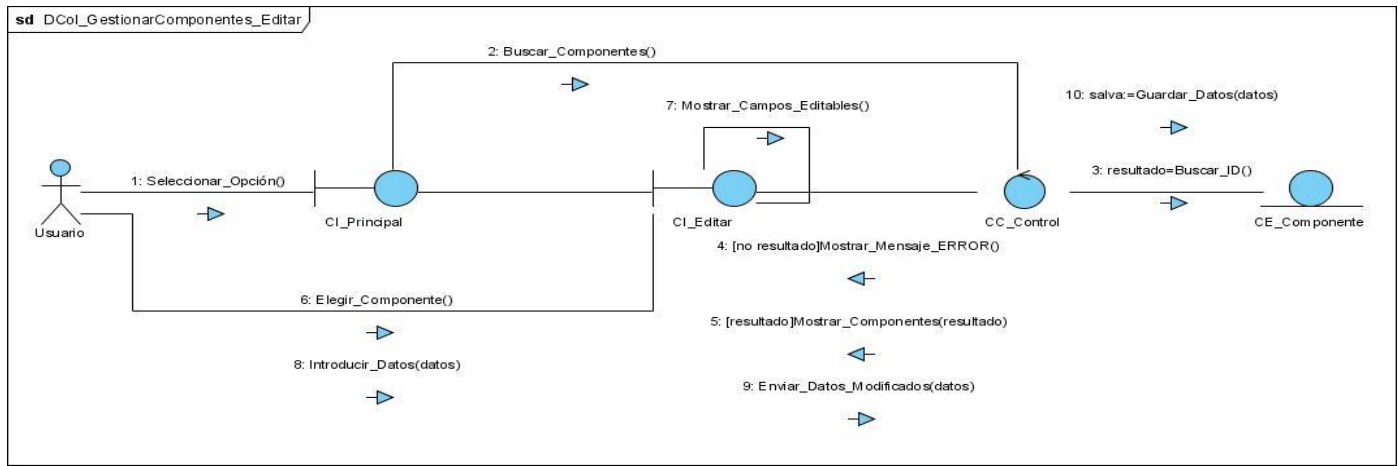
Diagramas de Colaboración: Gestionar Componentes (Sección Adicionar).



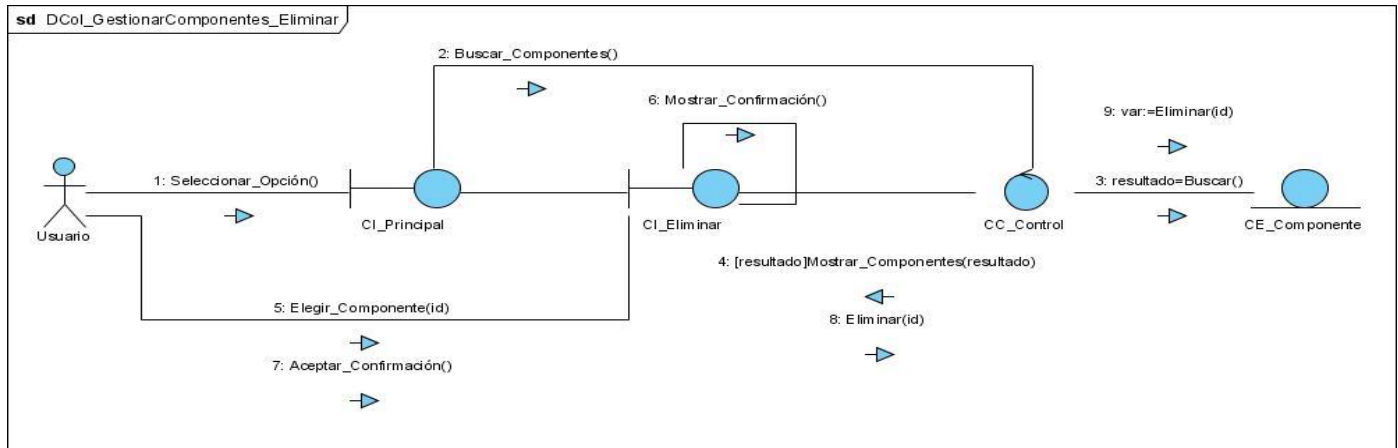
Diagramas de Colaboración: Gestionar Componentes (Sección Listar).



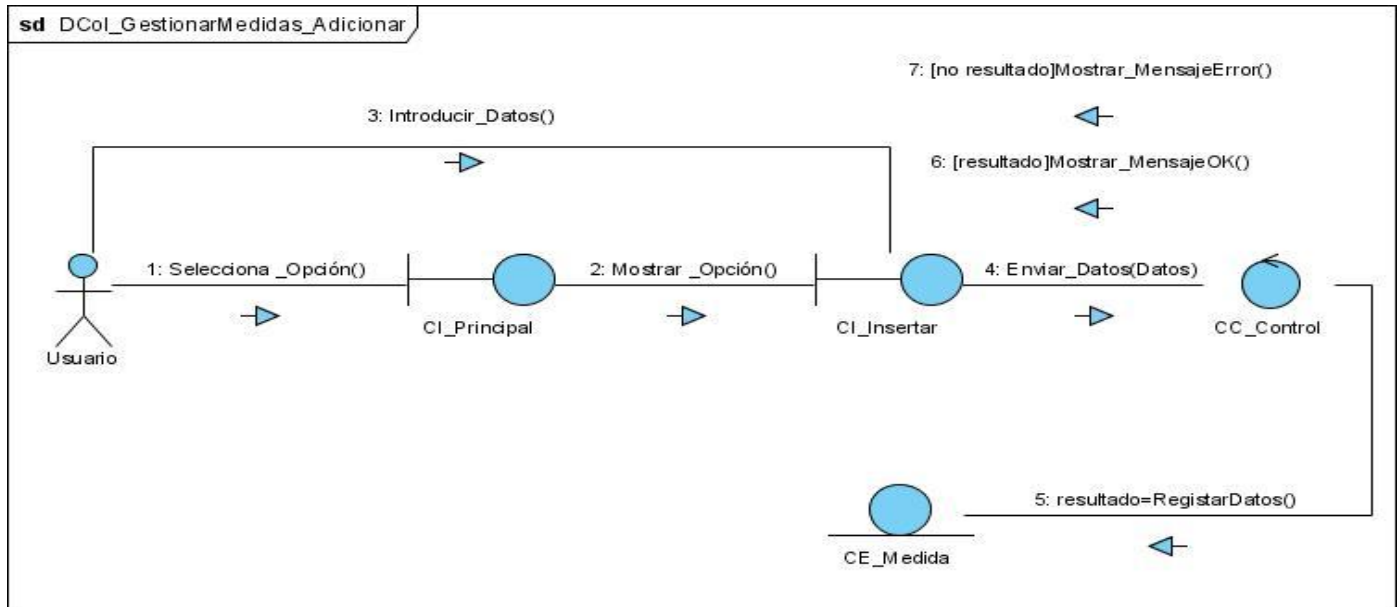
Diagramas de Colaboración: Gestionar Componentes (Sección Modificar).



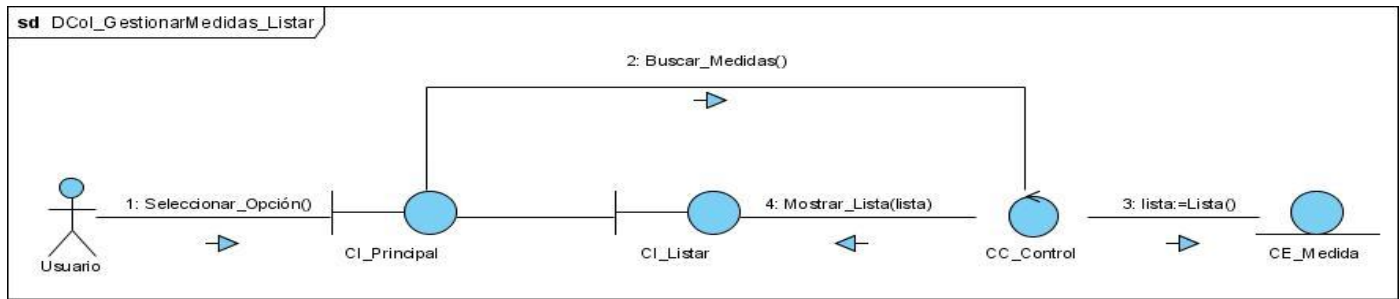
Diagramas de Colaboración: Gestionar Componentes (Sección Eliminar).



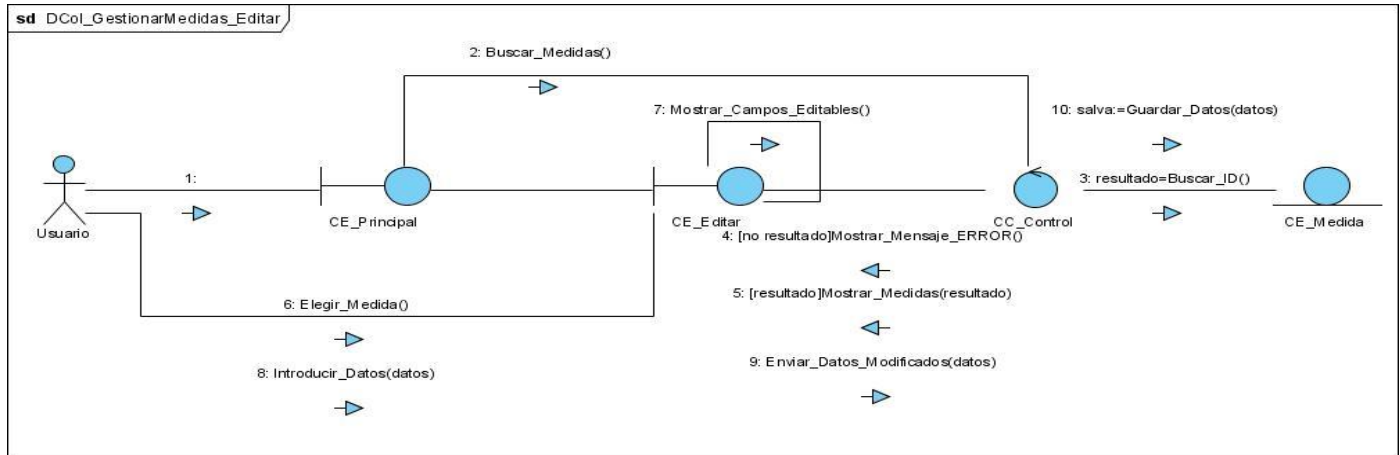
Diagramas de Colaboración: Gestionar Medidas (Sección Adicionar).



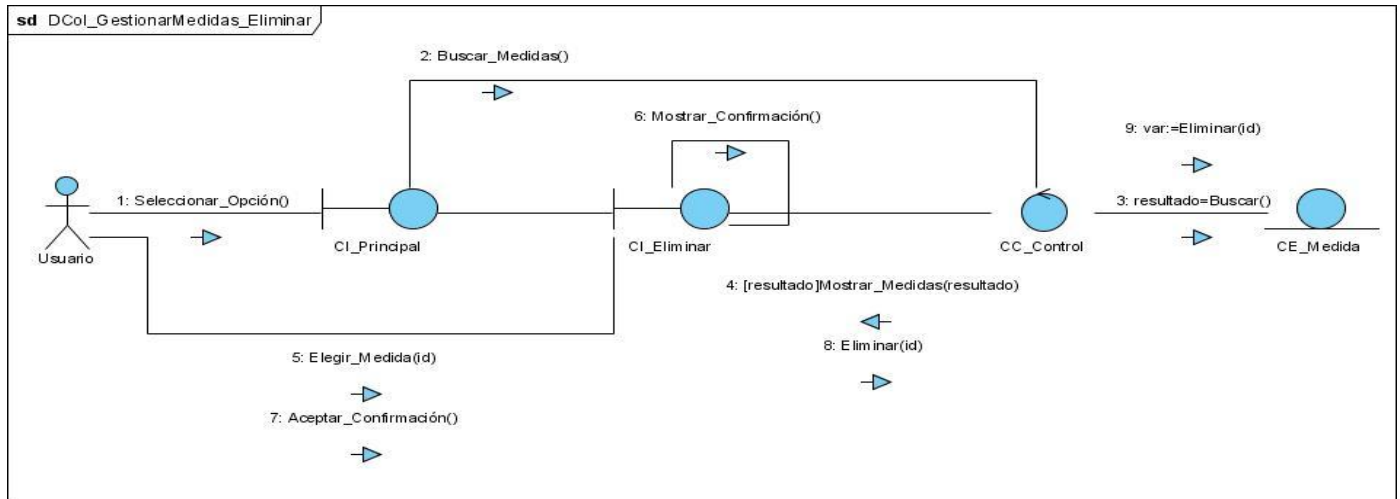
Diagramas de Colaboración: Gestionar Medidas (Sección Listar).



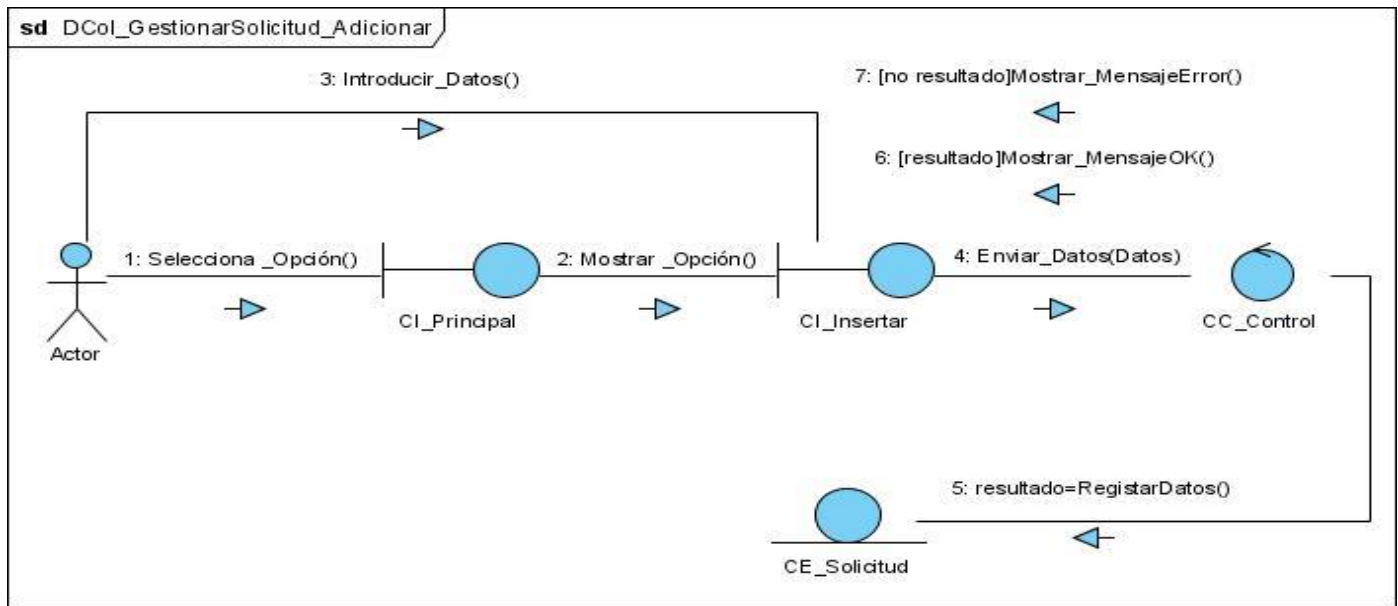
Diagramas de Colaboración: Gestionar Medidas (Sección Modificar).



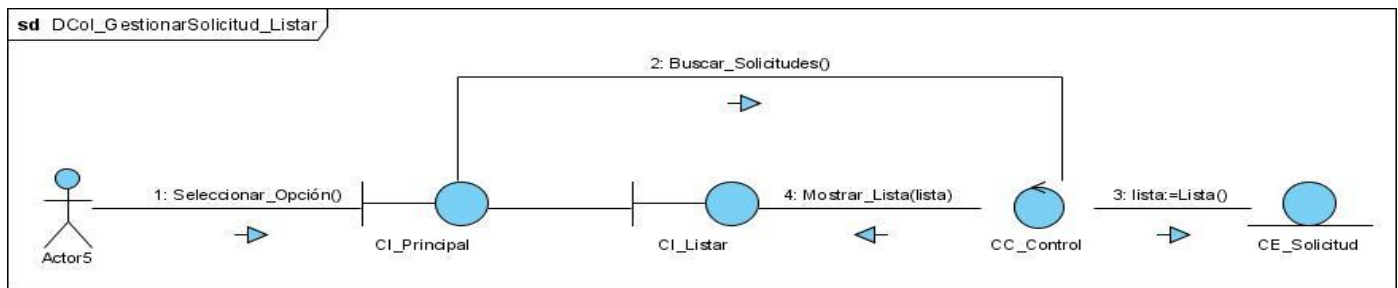
Diagramas de Colaboración: Gestionar Medidas (Sección Eliminar).



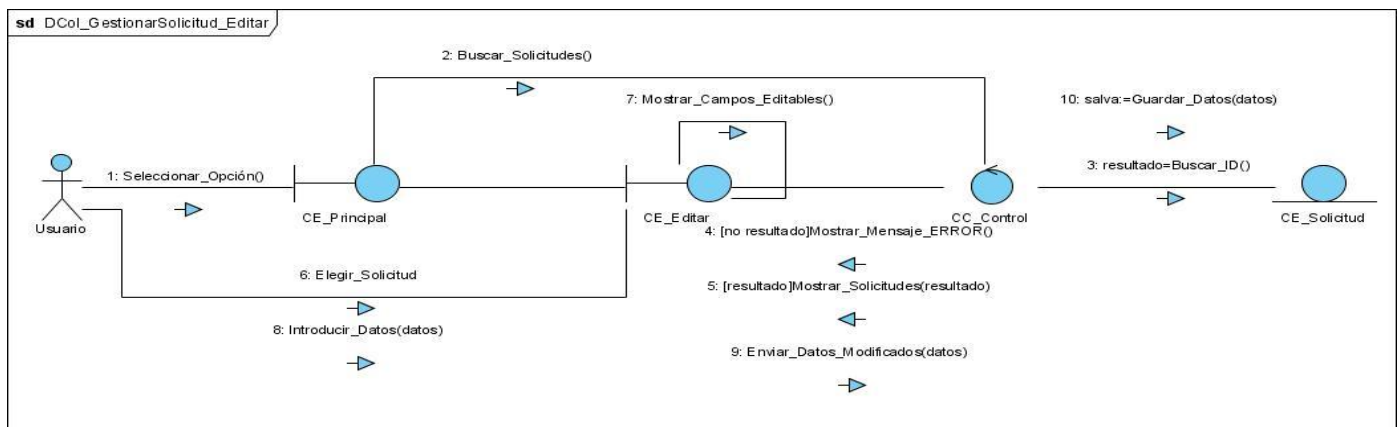
Diagramas de Colaboración: Gestionar Solicitud (Sección Adicionar).



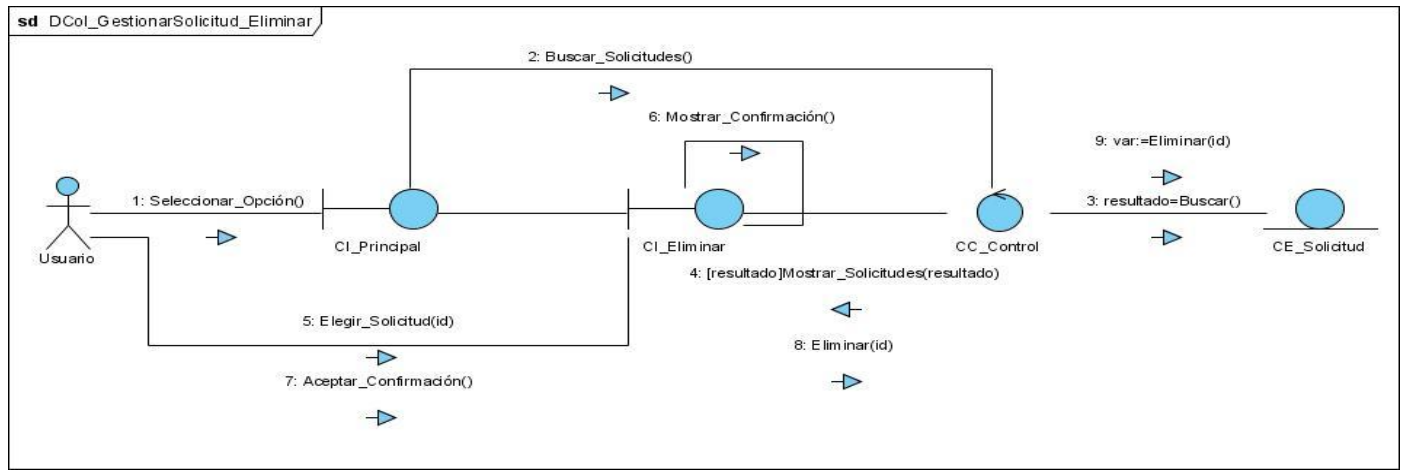
Diagramas de Colaboración: Gestionar Solicitud (Sección Listar).



Diagramas de Colaboración: Gestionar Solicitud (Sección Modificar).

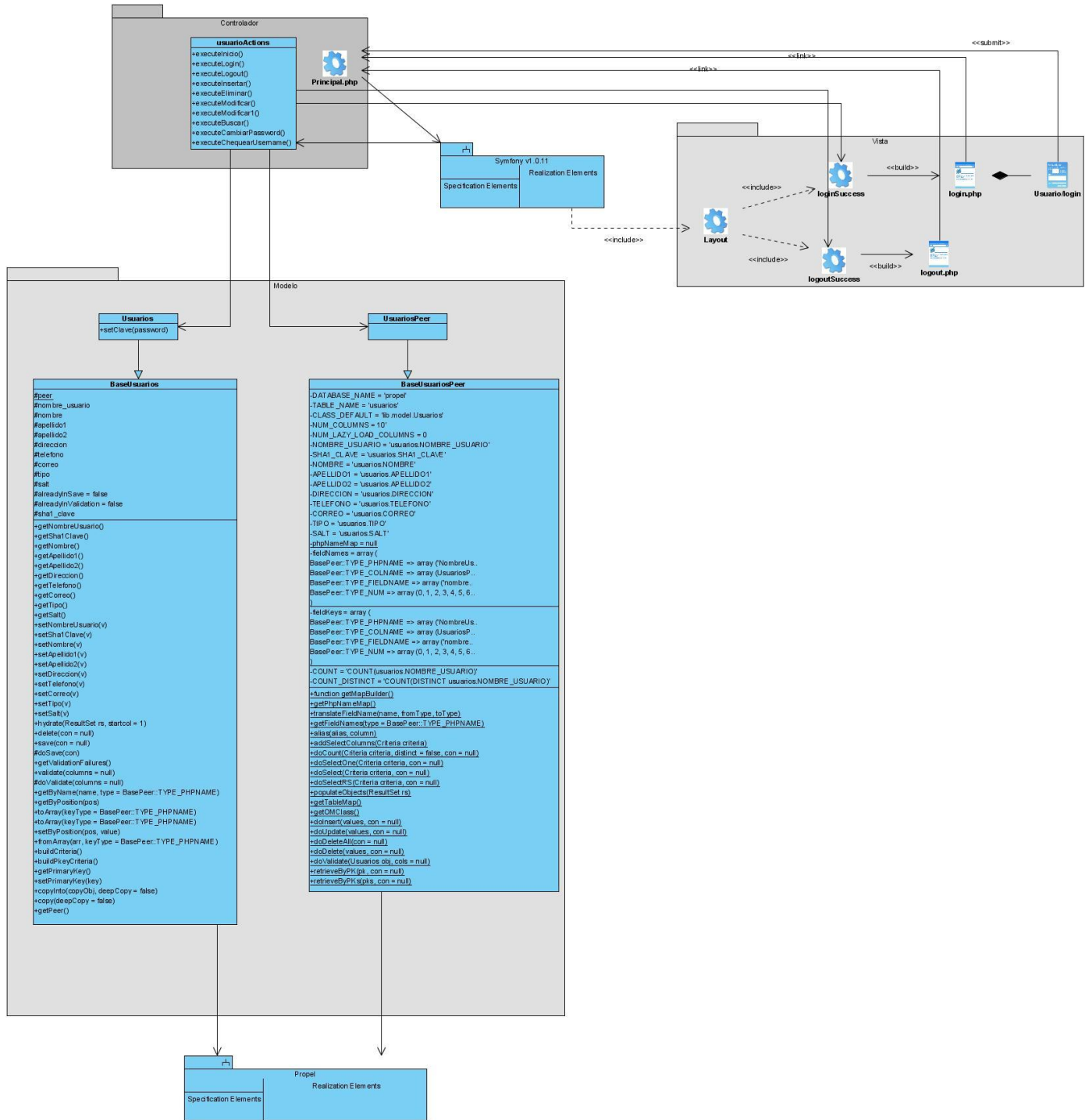


Diagramas de Colaboración: Gestionar Solicitud (Sección Eliminar).

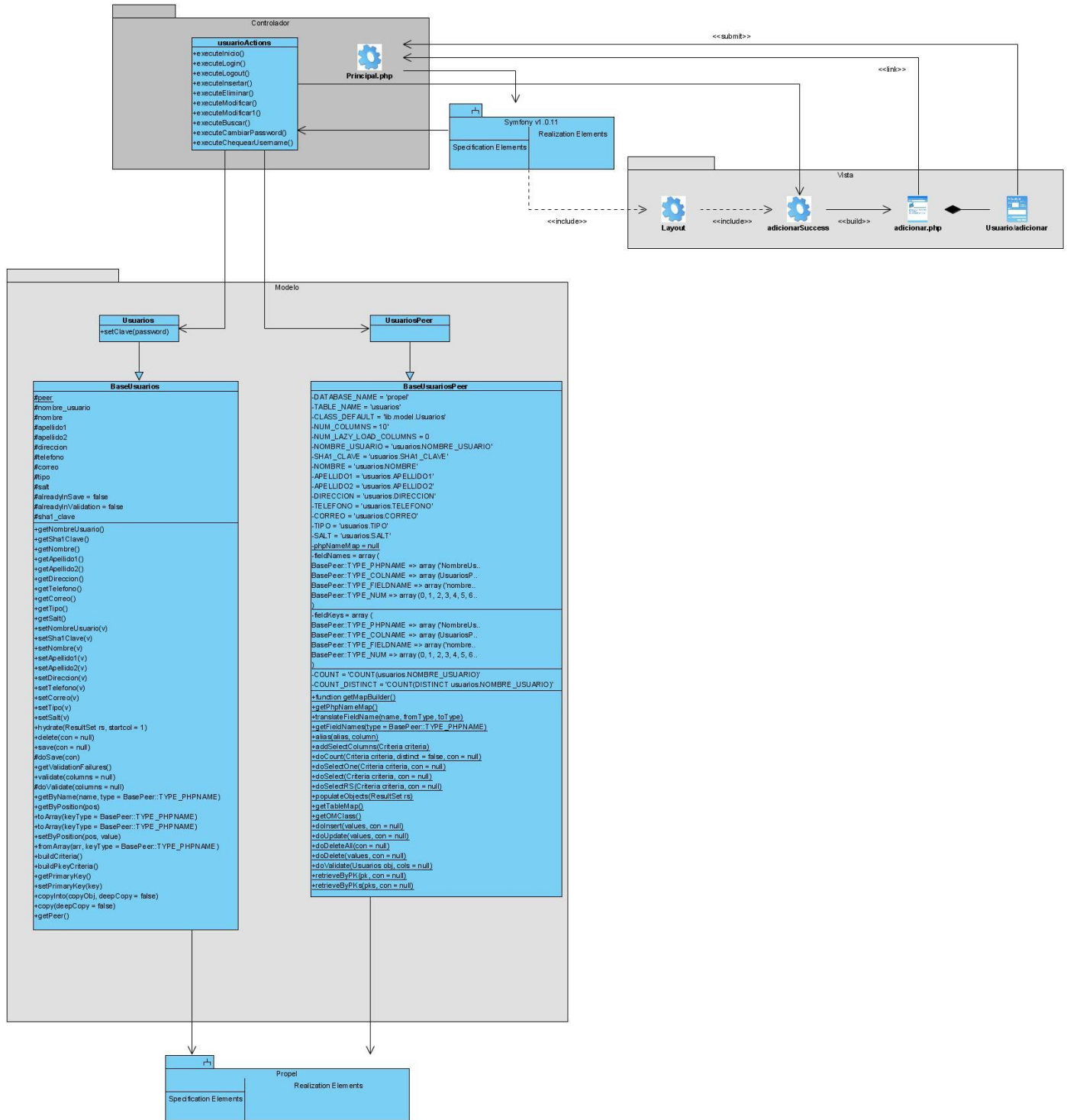


Anexos 3. Diagramas de Clases de Diseño.

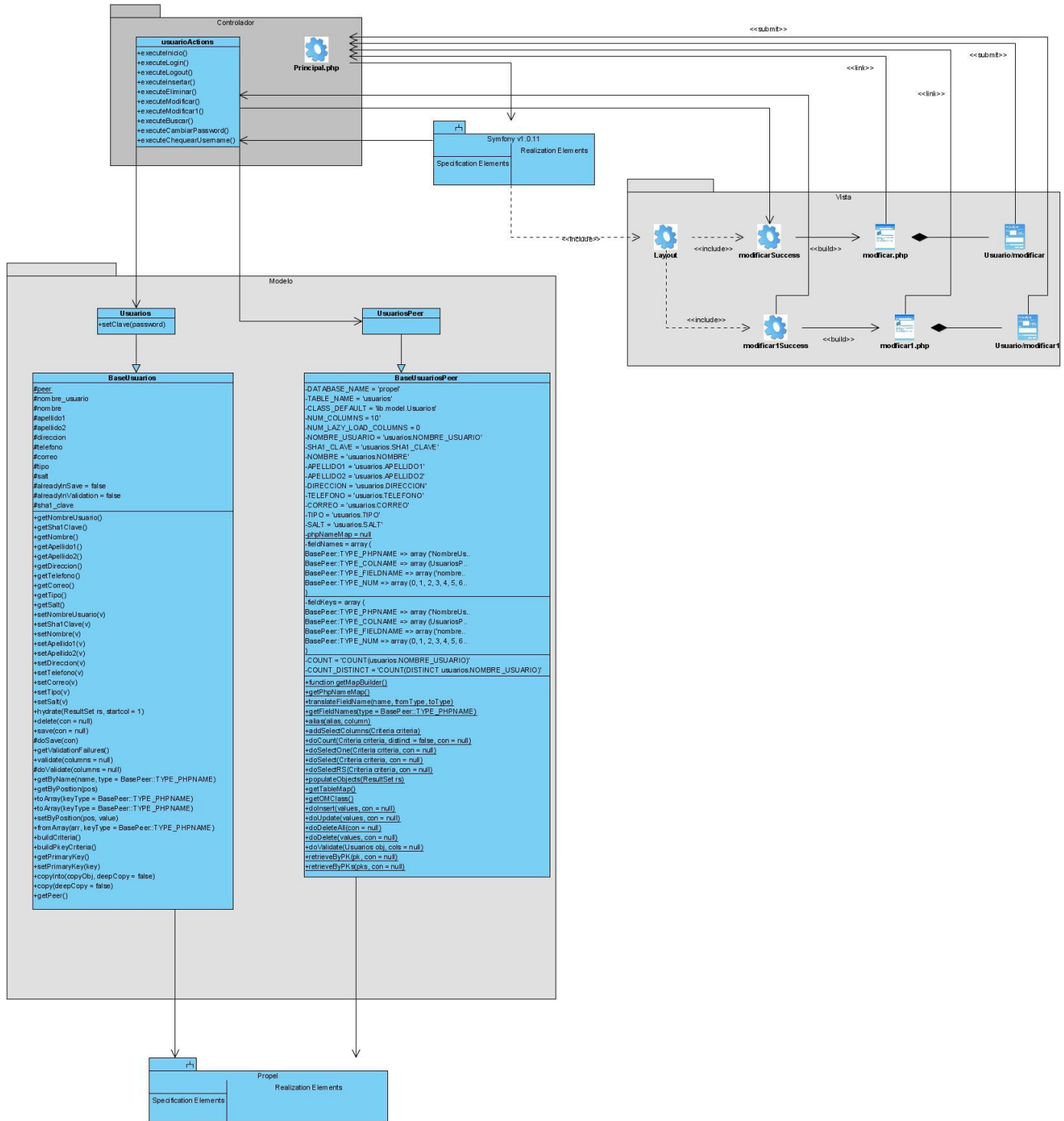
Diagramas Clases de Diseño del Caso de Uso: Autenticar Usuario.



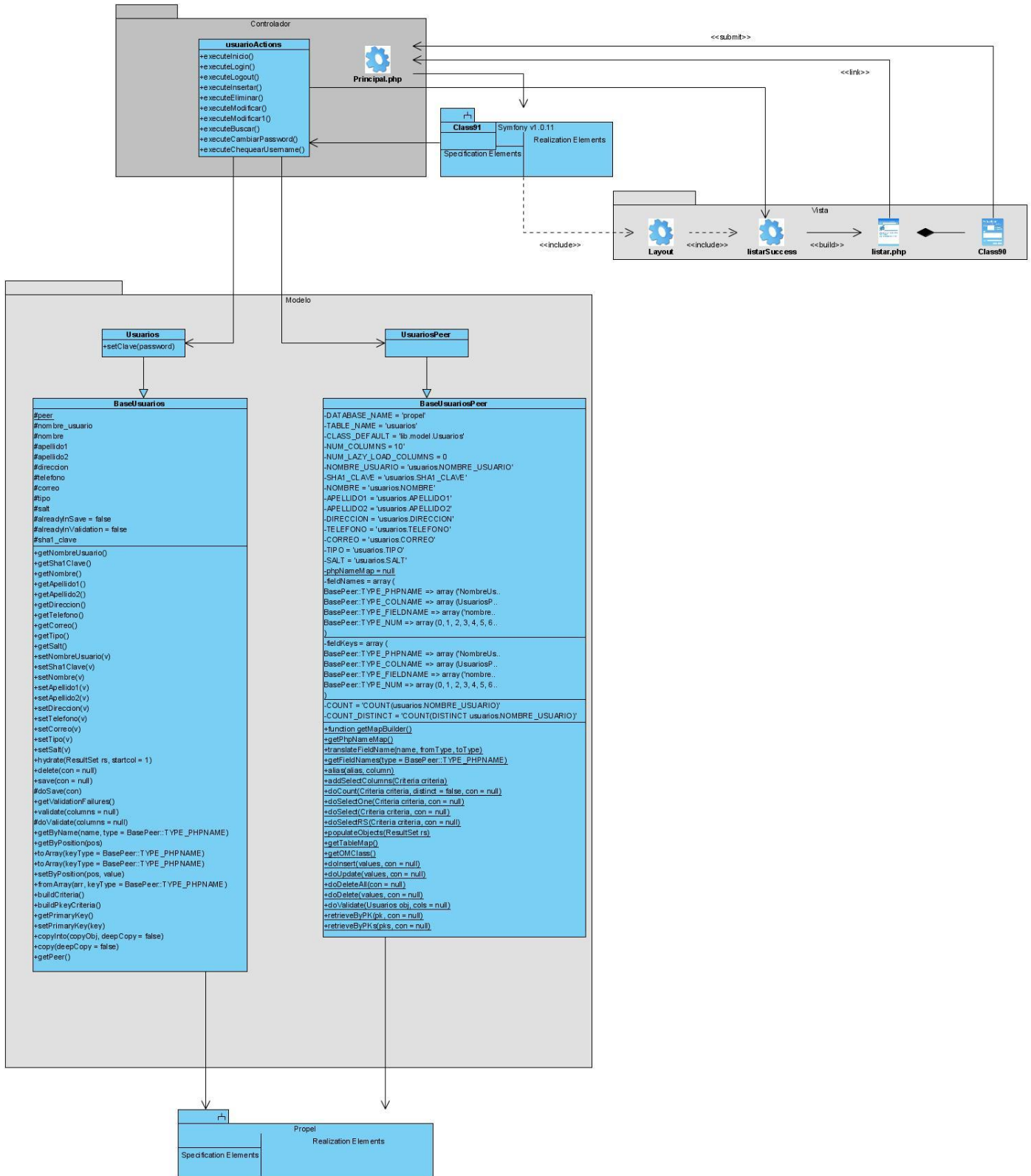
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Adicionar.



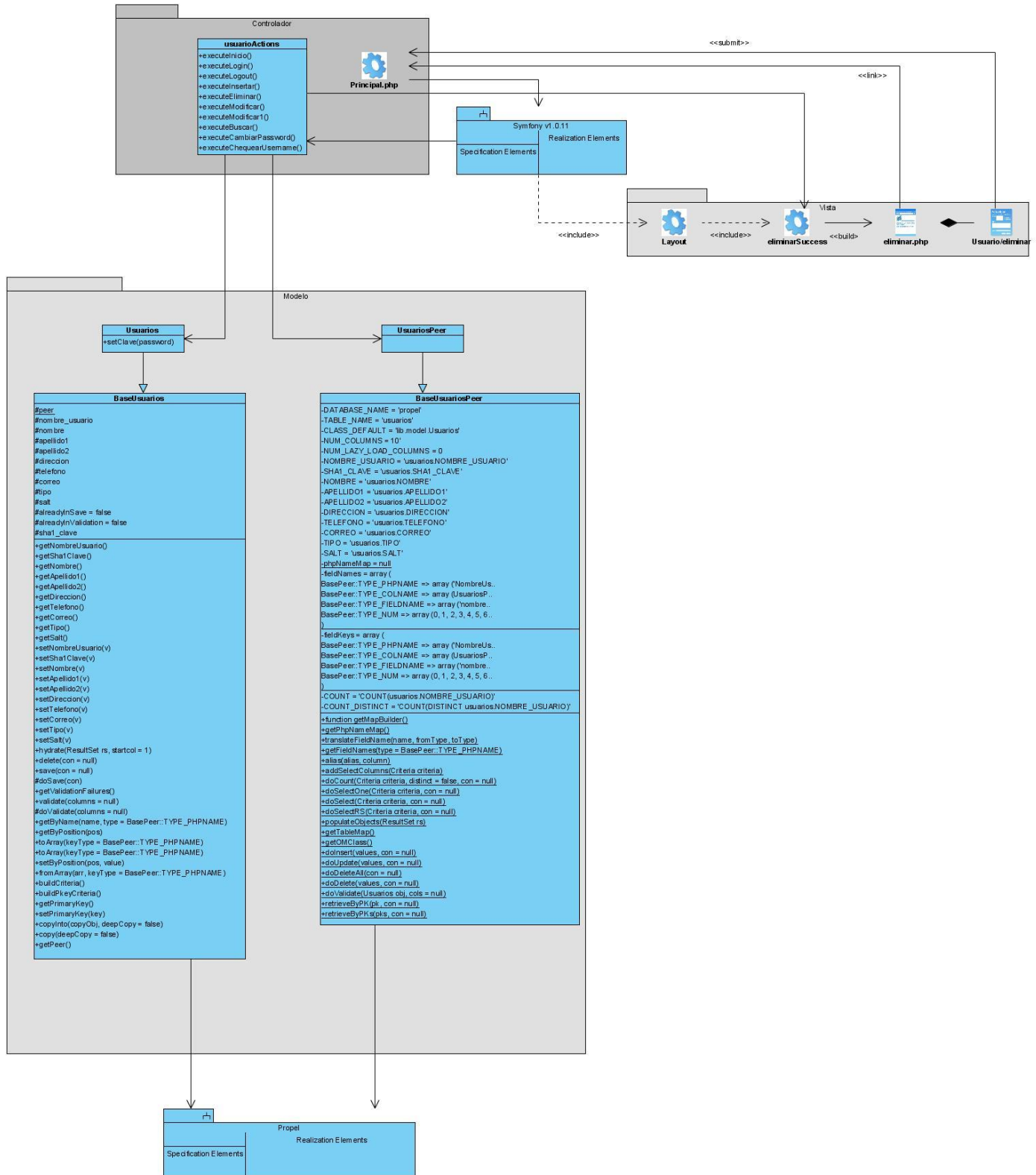
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Modificar.



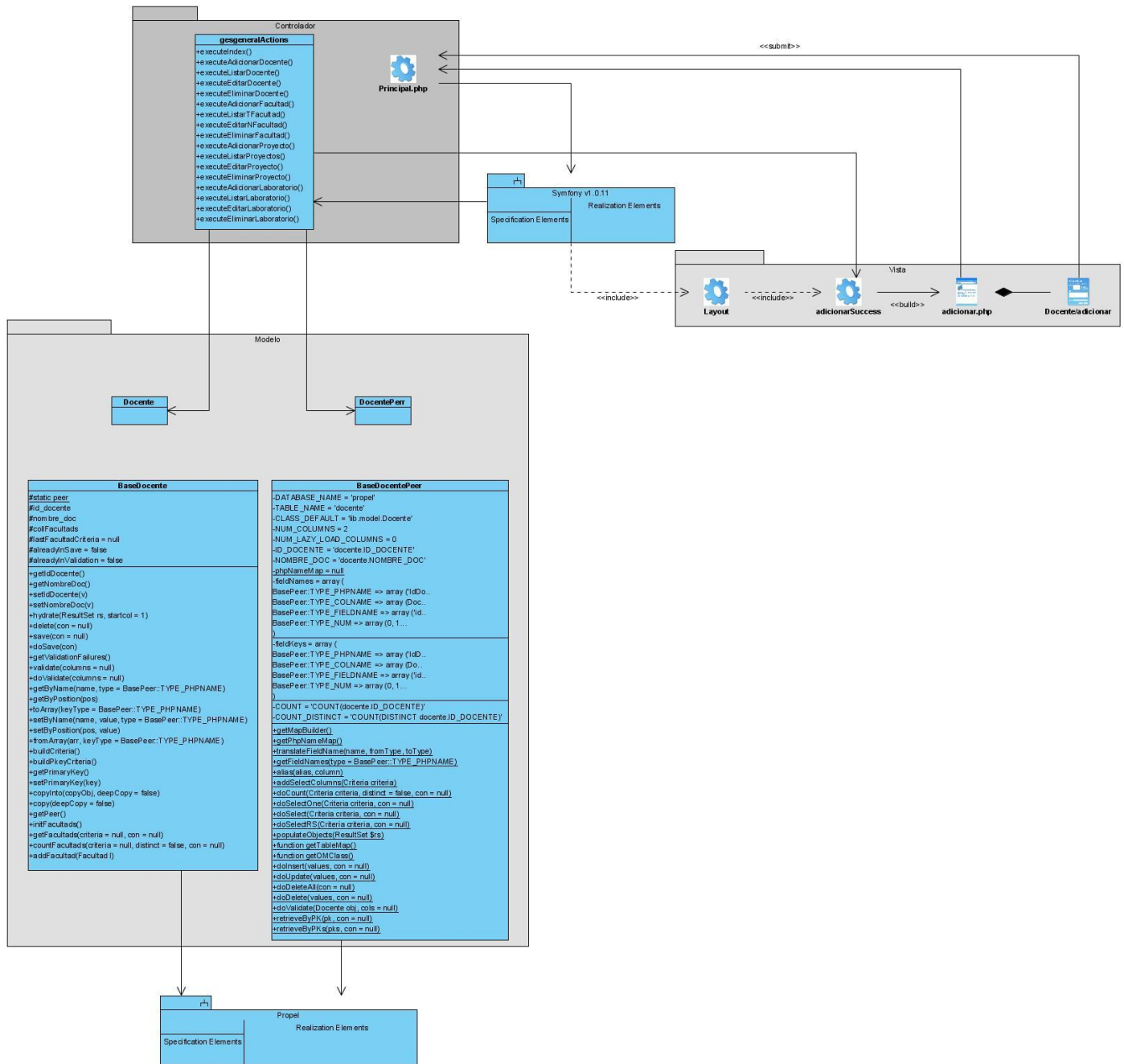
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Listar.



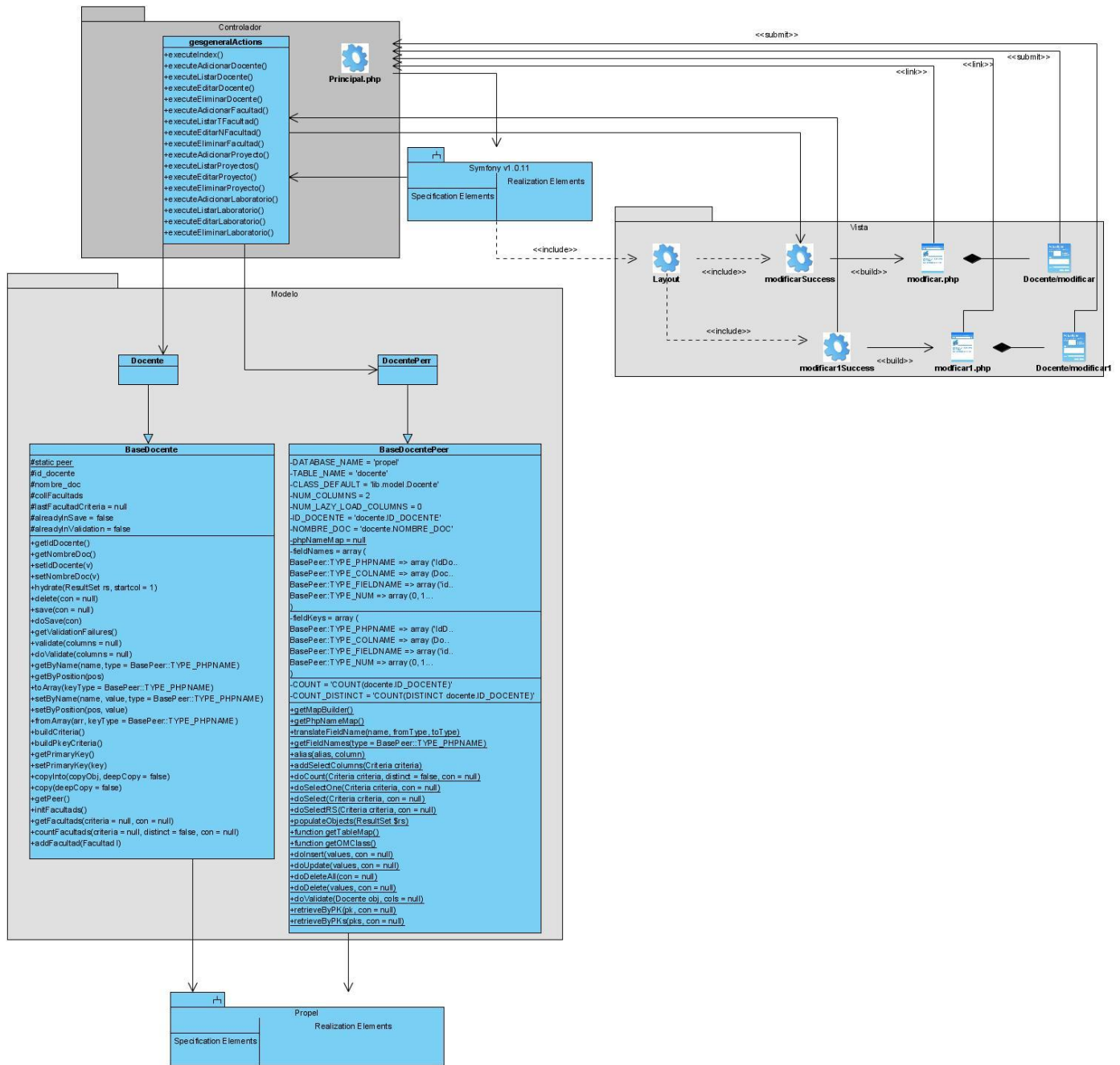
Diagramas Clases de Diseño del Caso de Uso: Gestionar Usuario sección Eliminar.



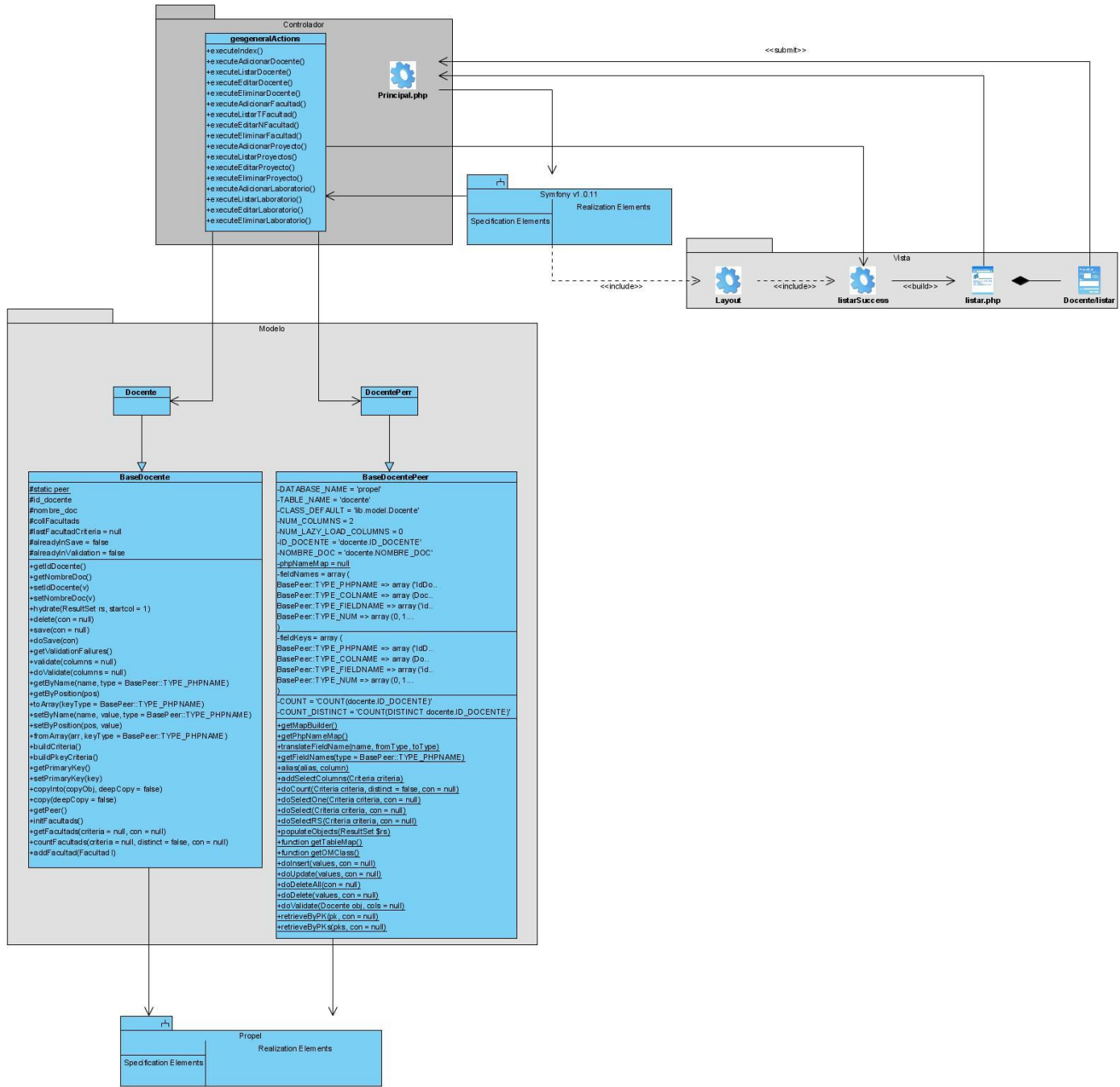
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Adicionar.



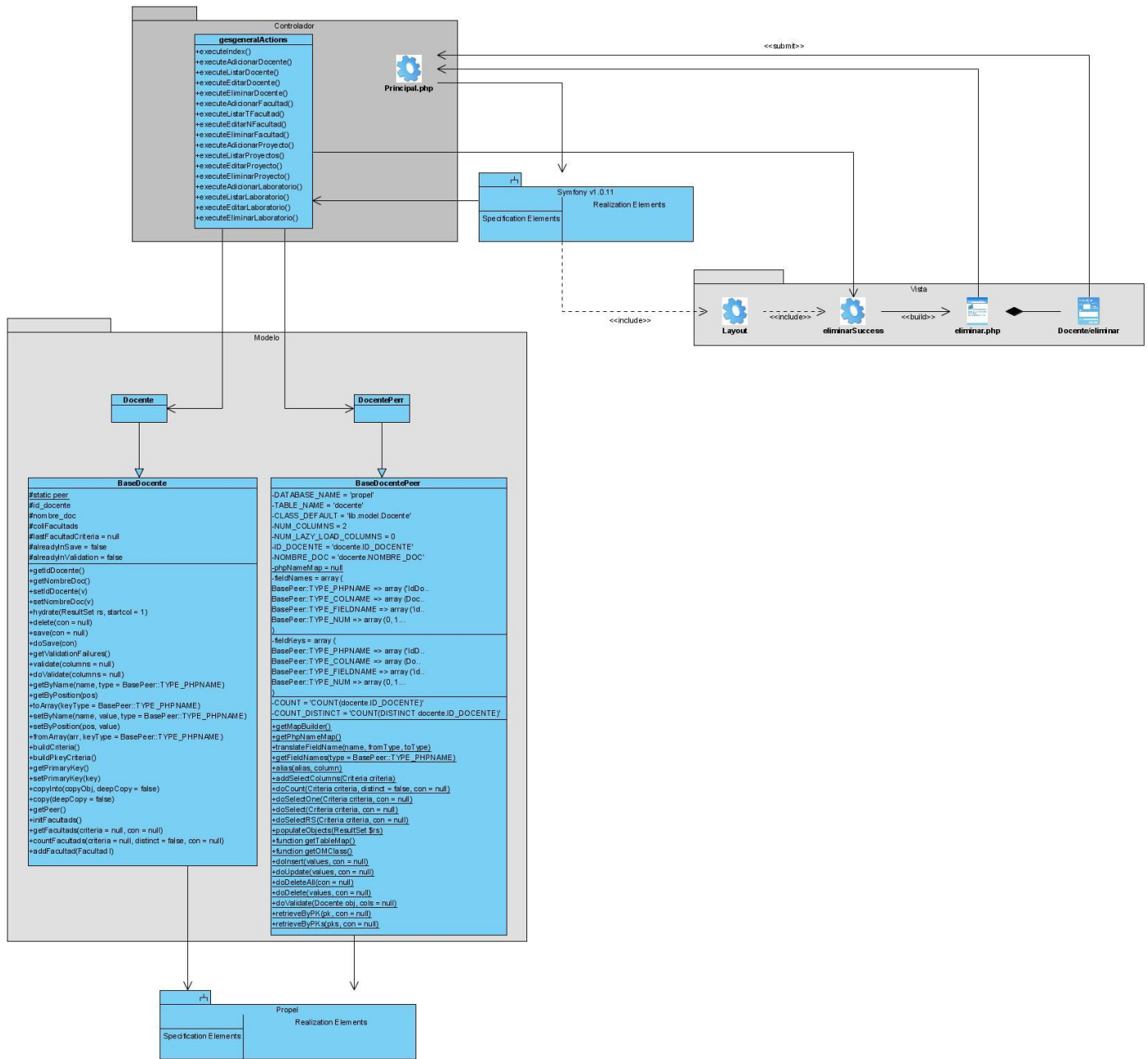
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Modificar.



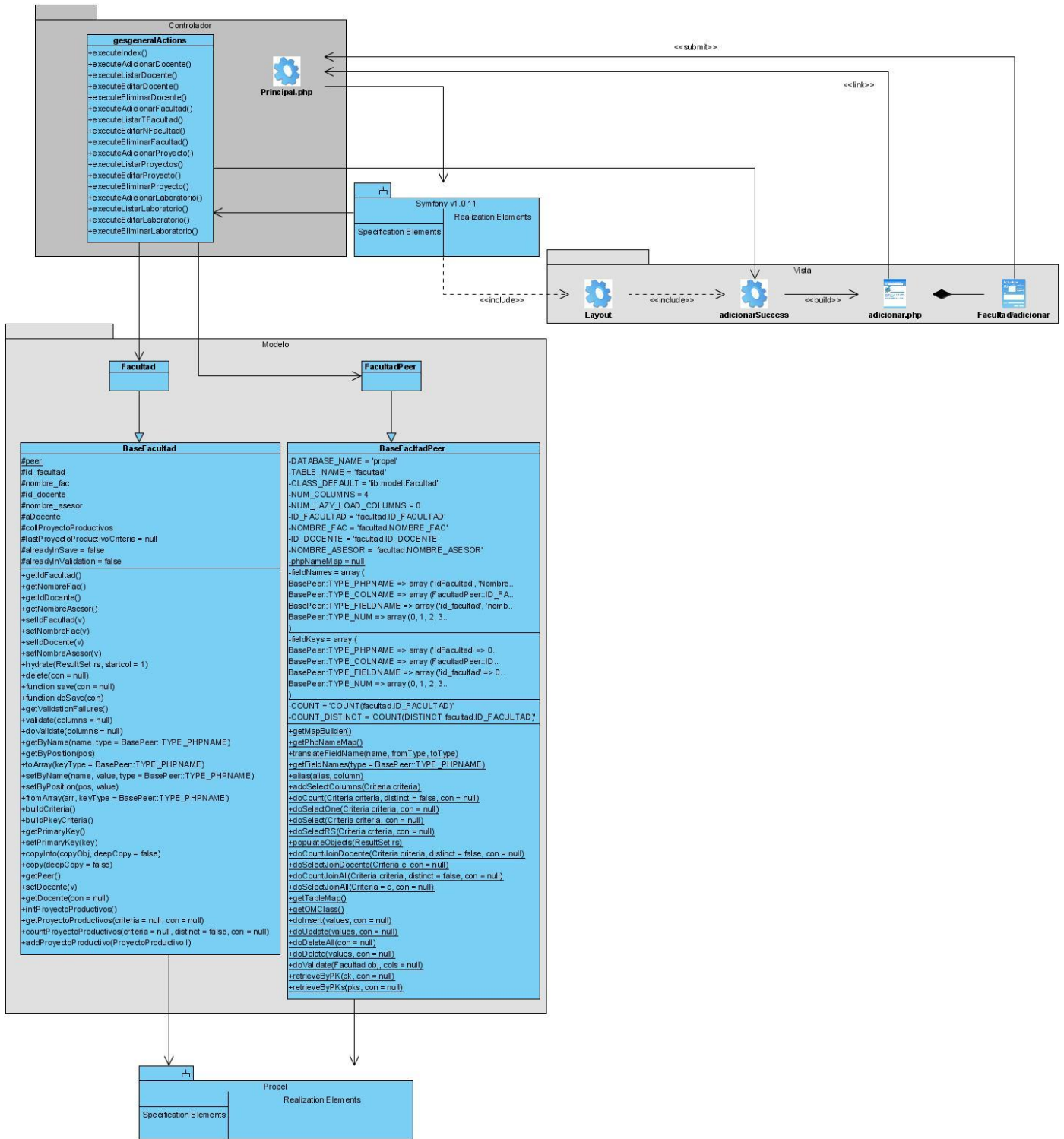
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Listar.



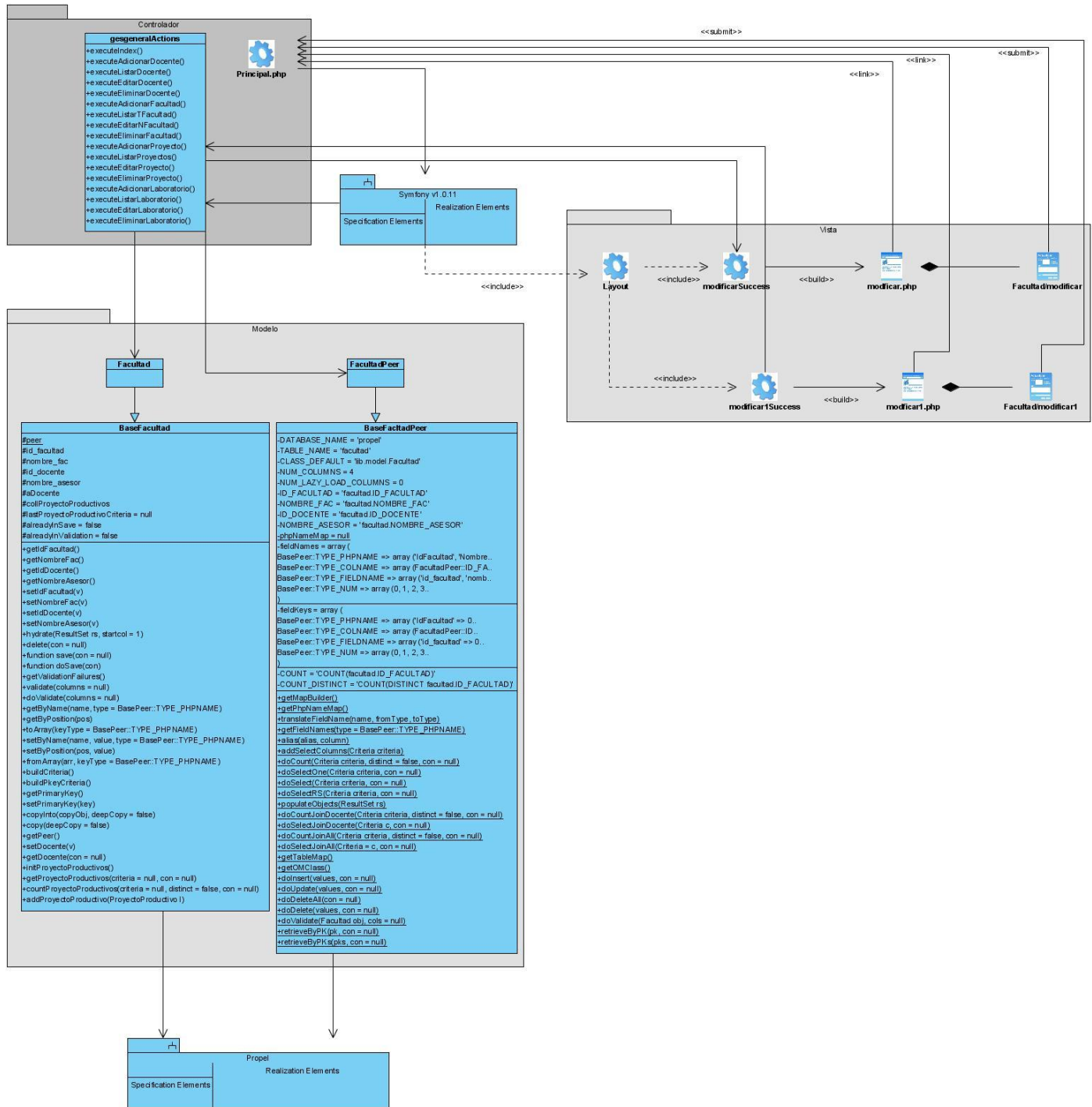
Diagramas Clases de Diseño del Caso de Uso: Gestionar Docente sección Eliminar.



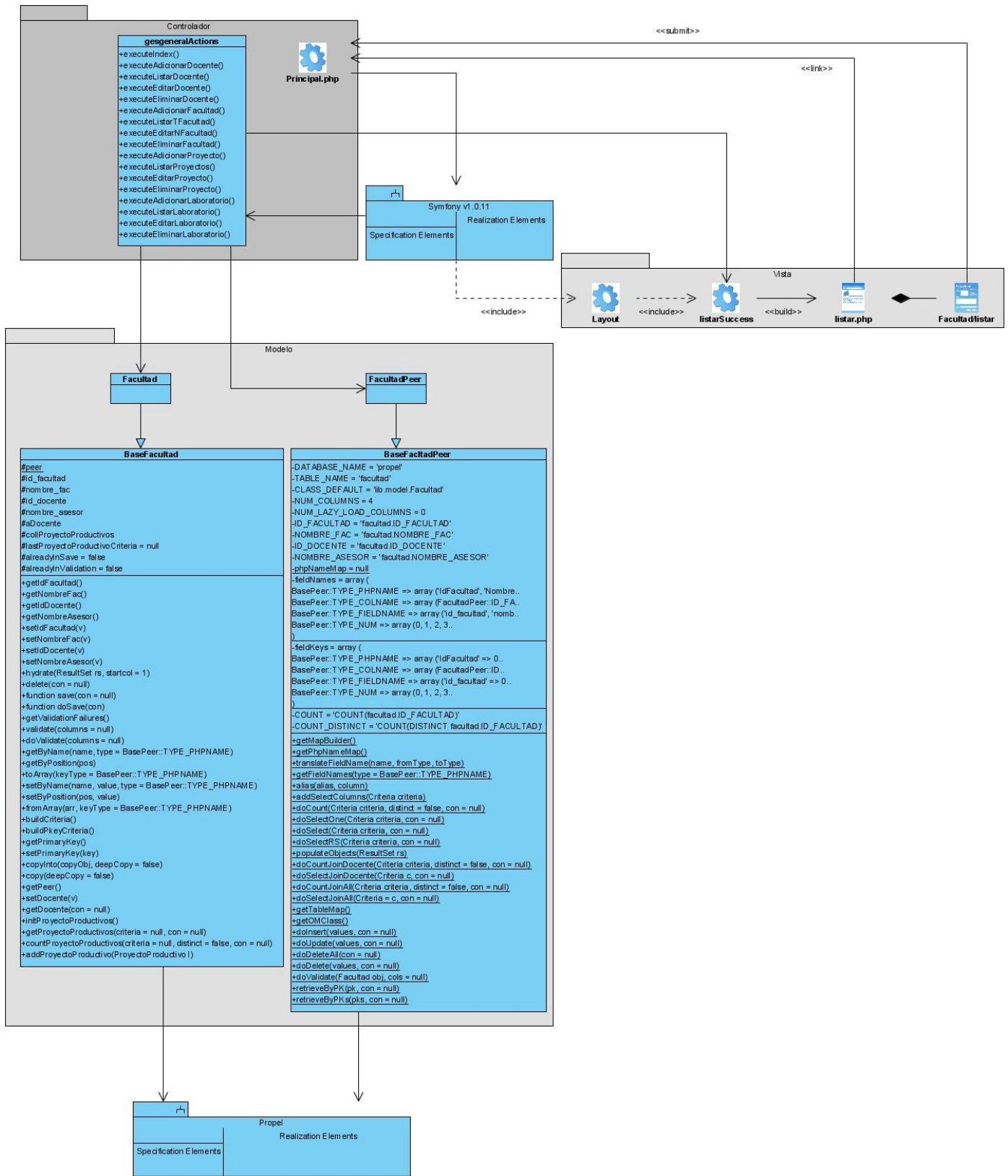
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Adicionar.



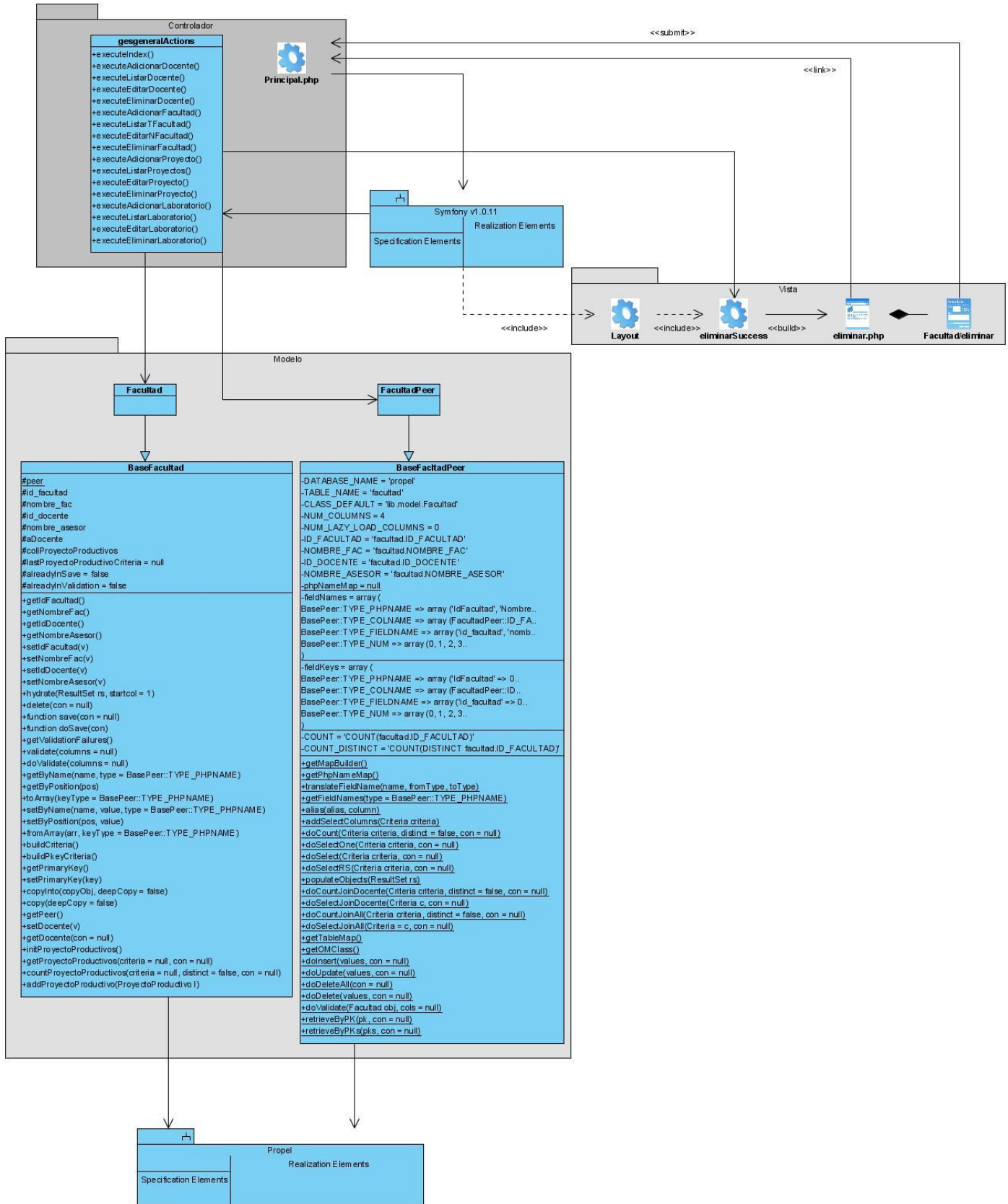
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Modificar.



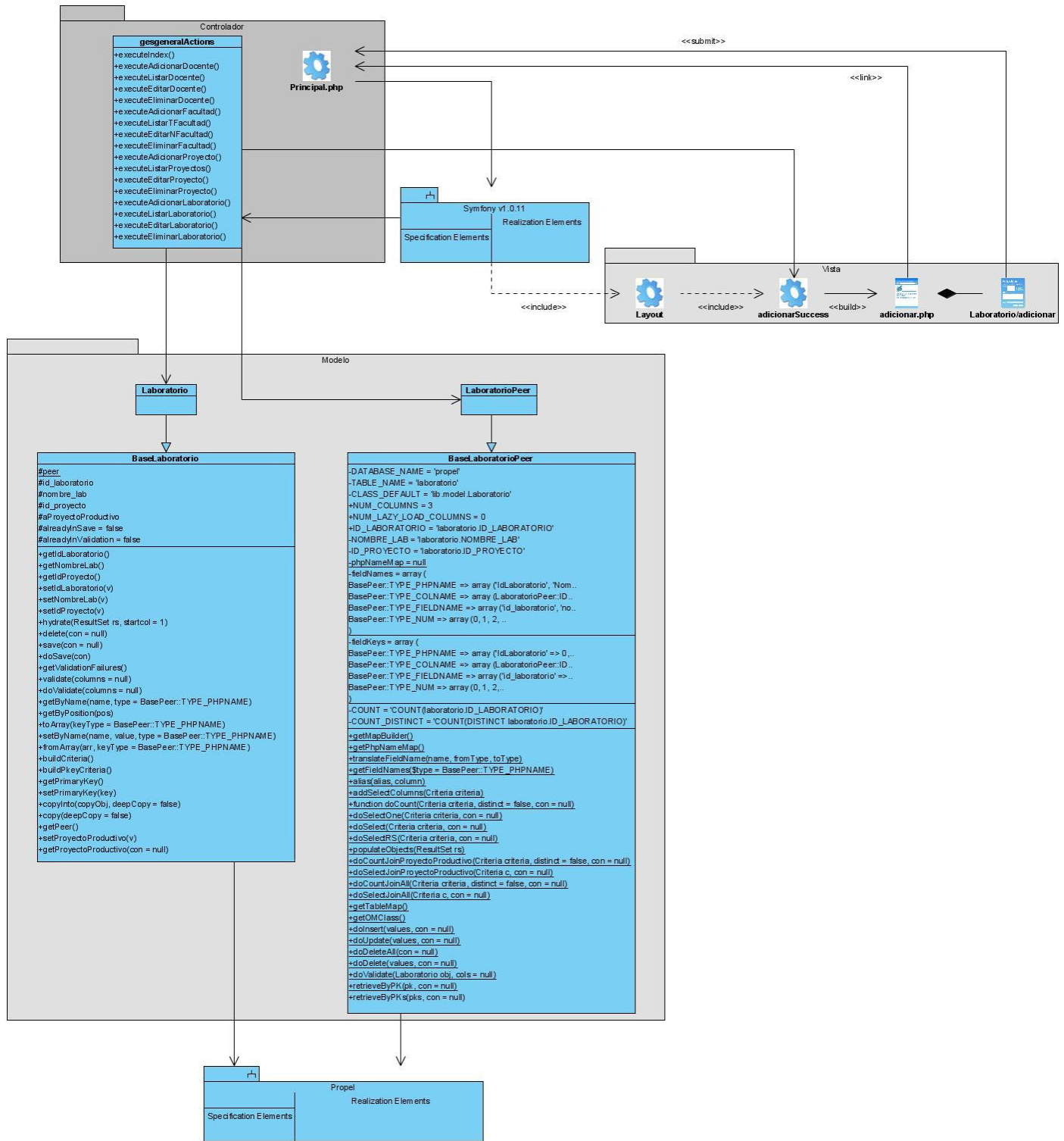
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Listar.



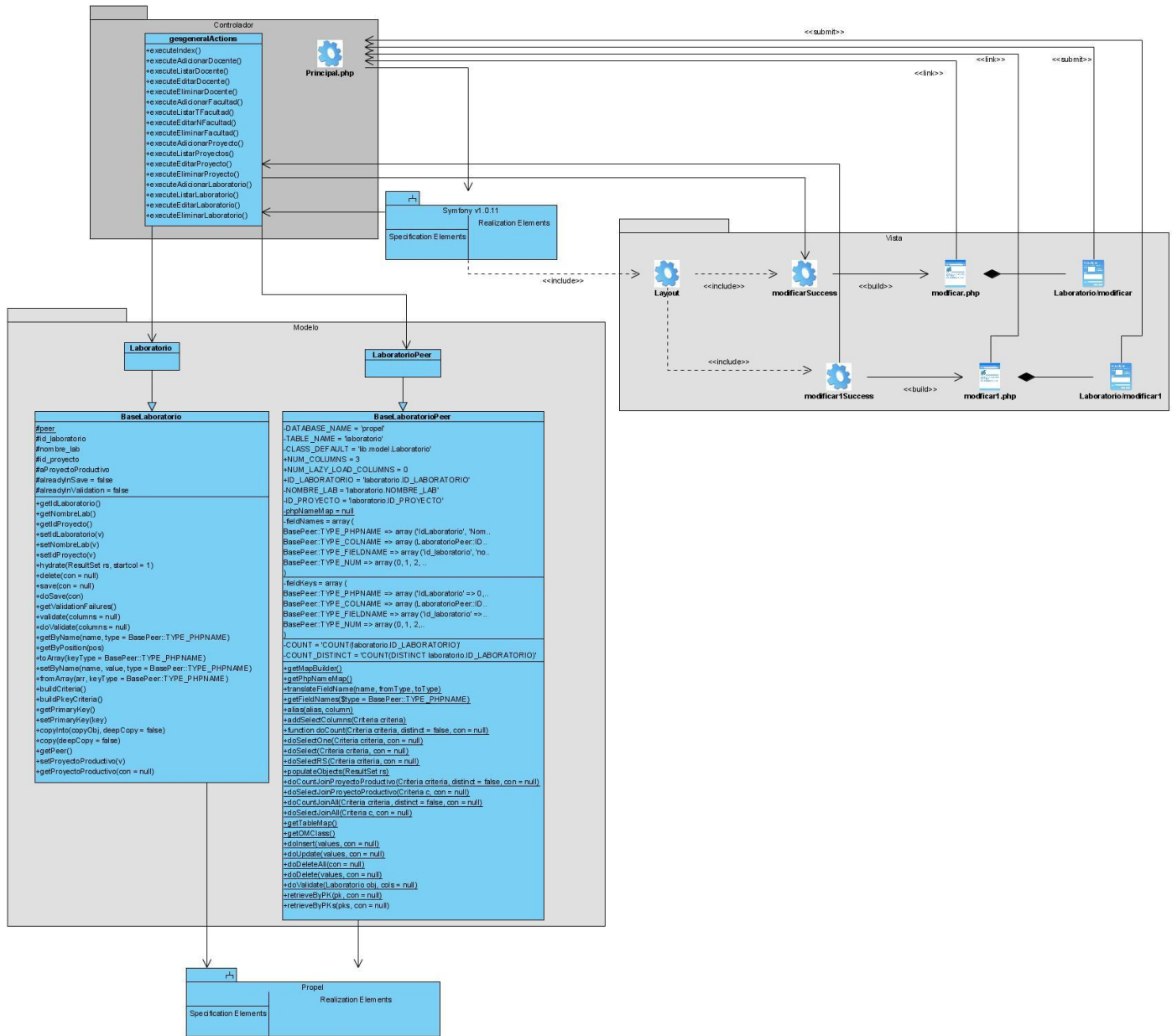
Diagramas Clases de Diseño del Caso de Uso: Gestionar Facultad sección Eliminar.



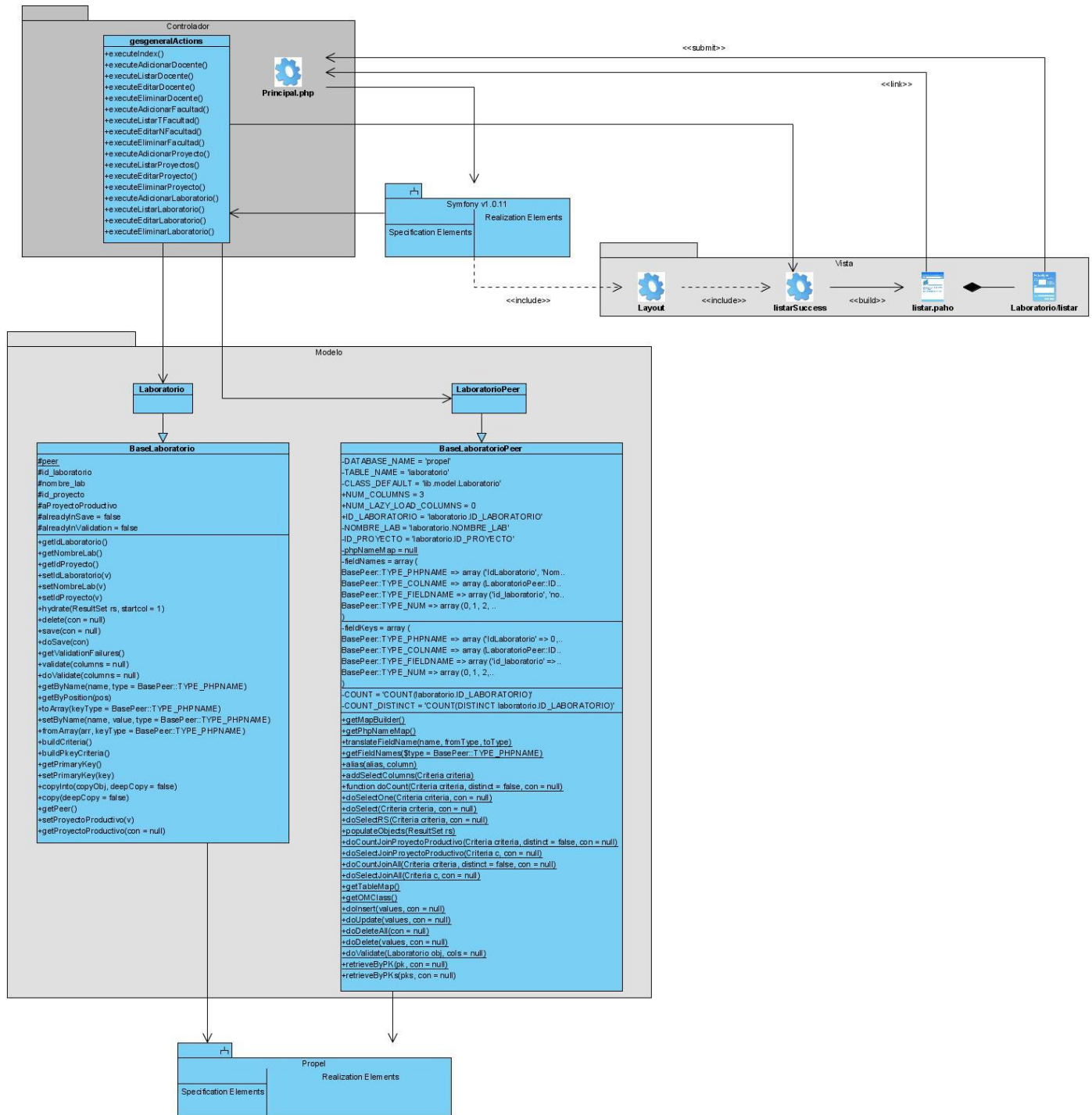
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Adicionar.



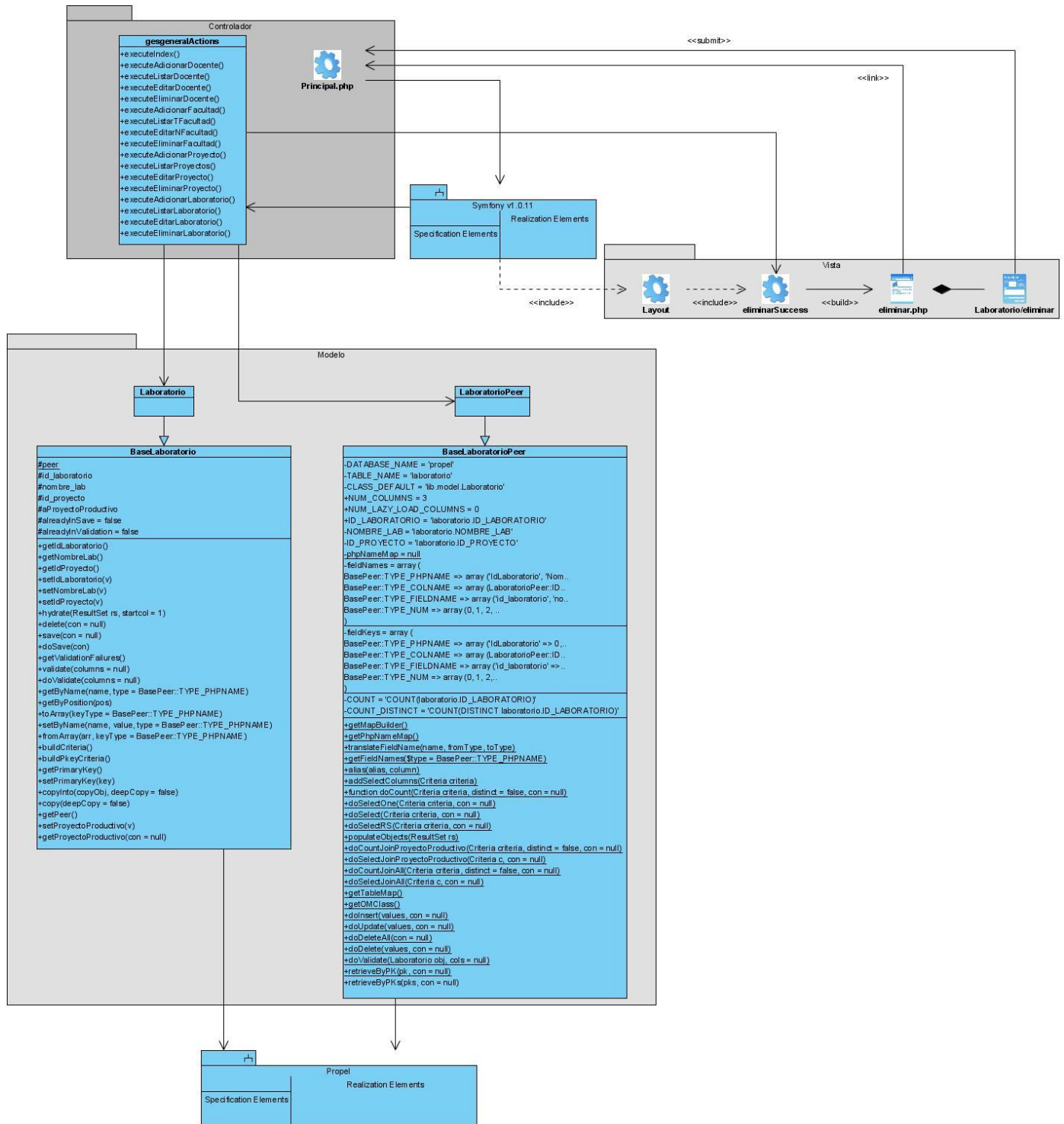
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Modificar.



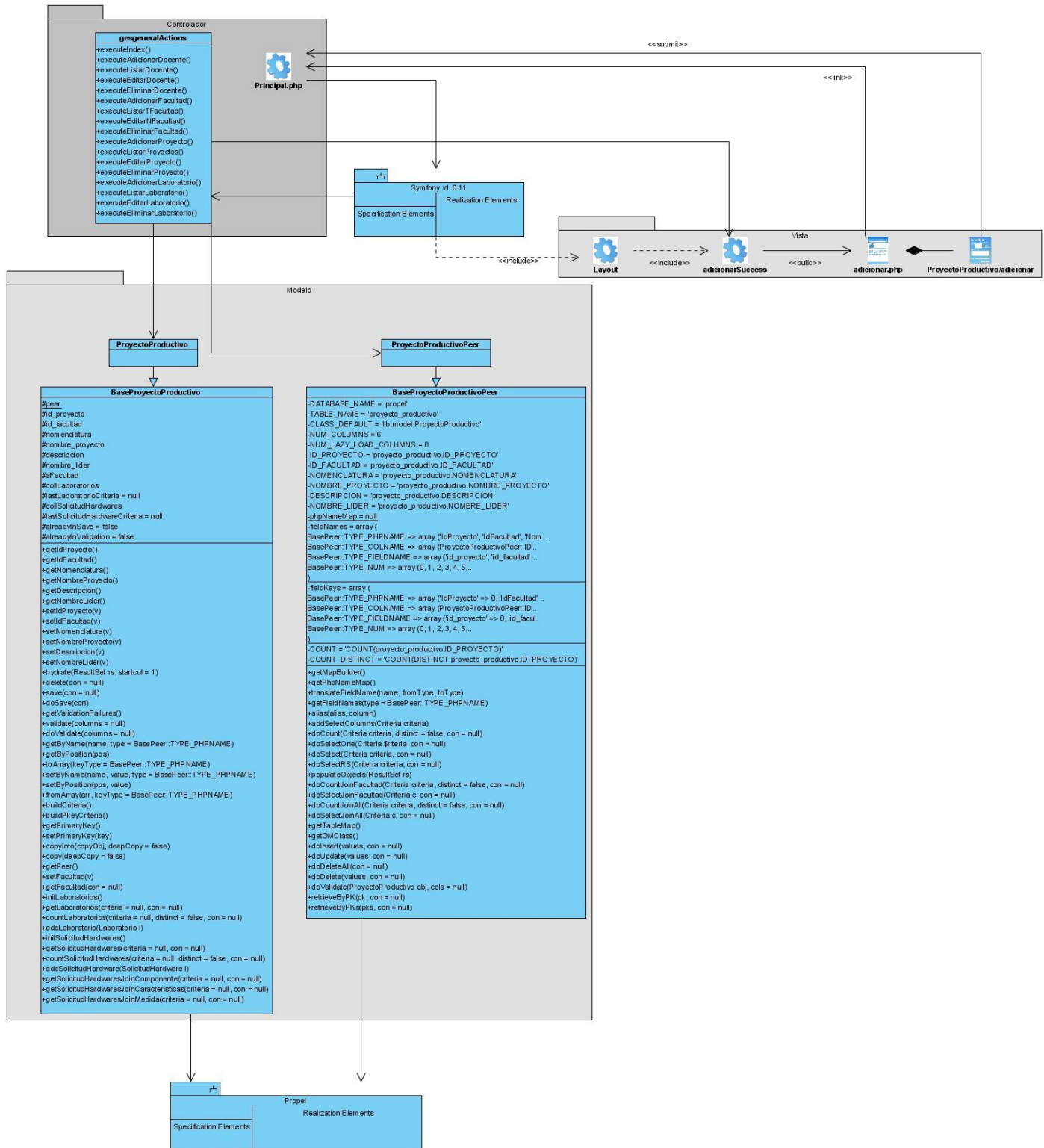
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Listar.



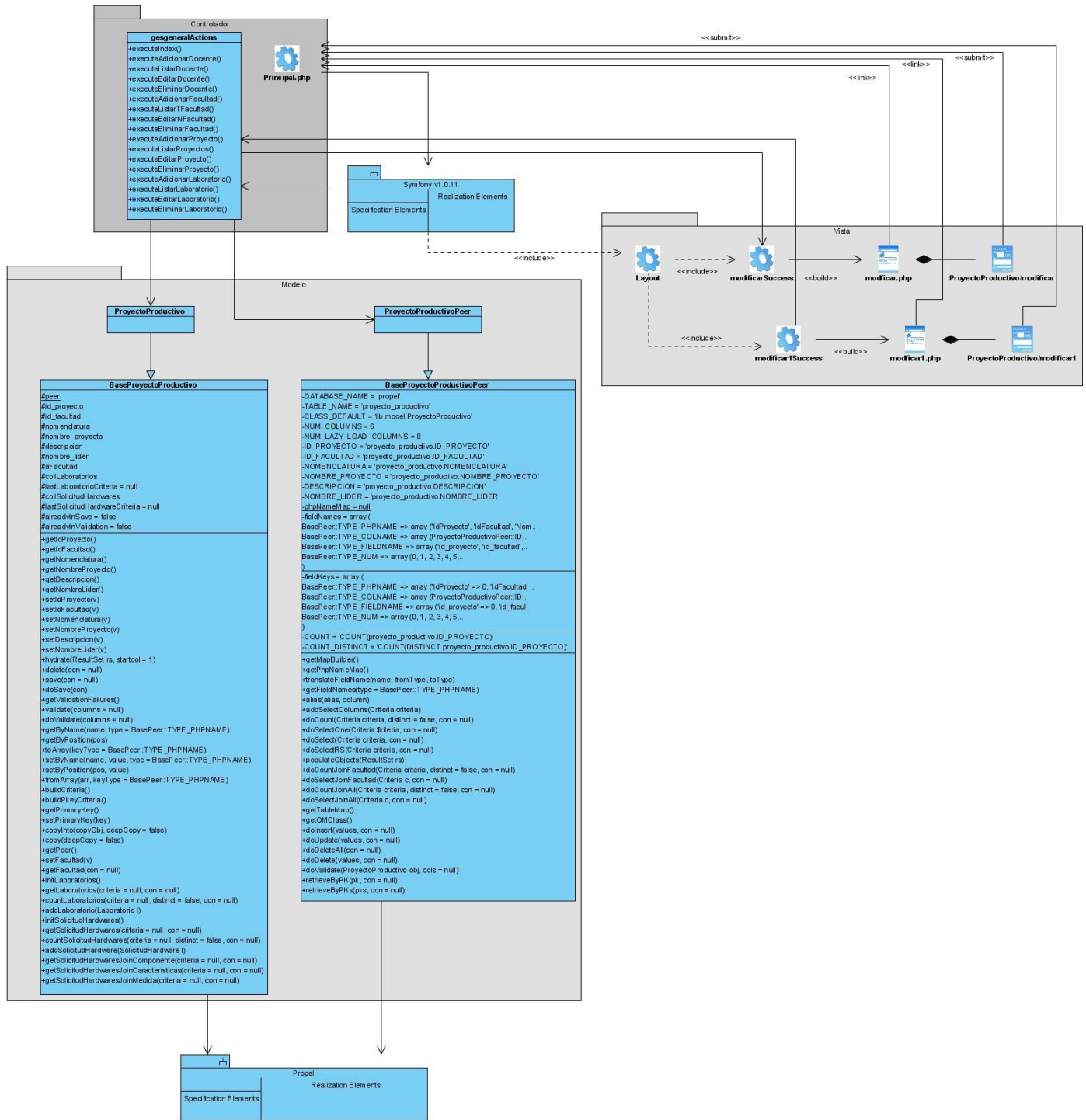
Diagramas Clases de Diseño del Caso de Uso: Gestionar Laboratorios sección Eliminar.



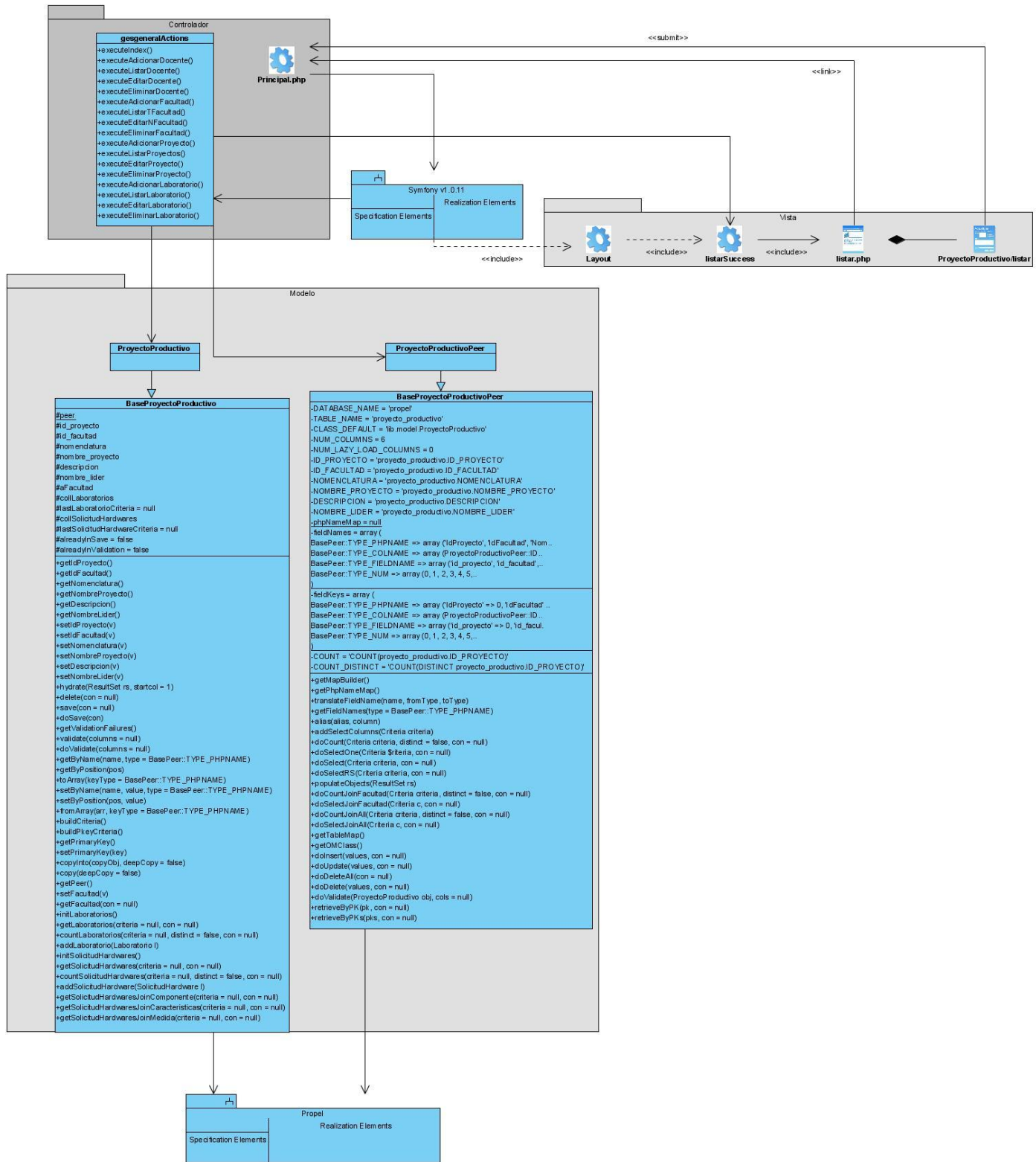
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Adicionar.



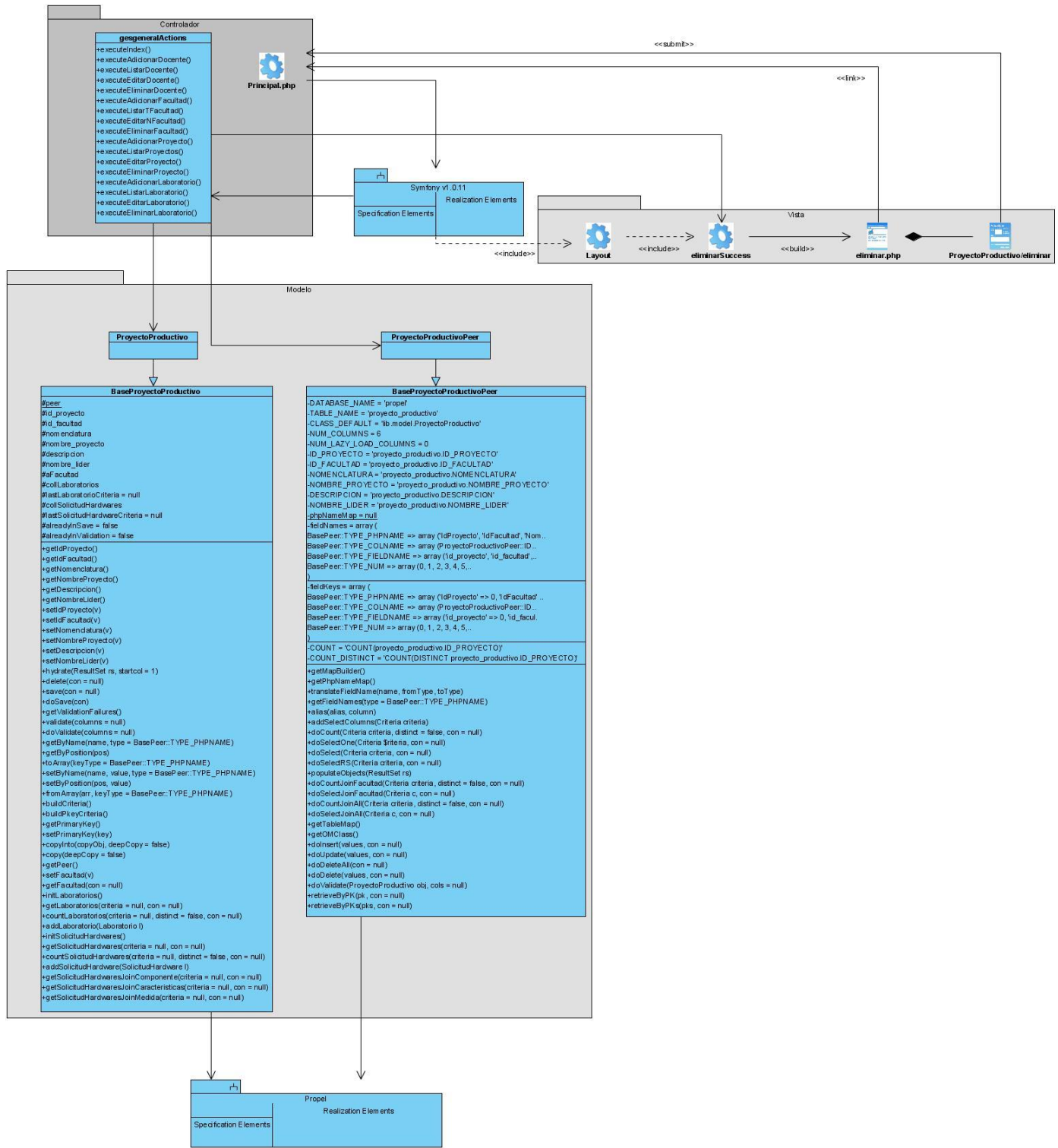
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Modificar.



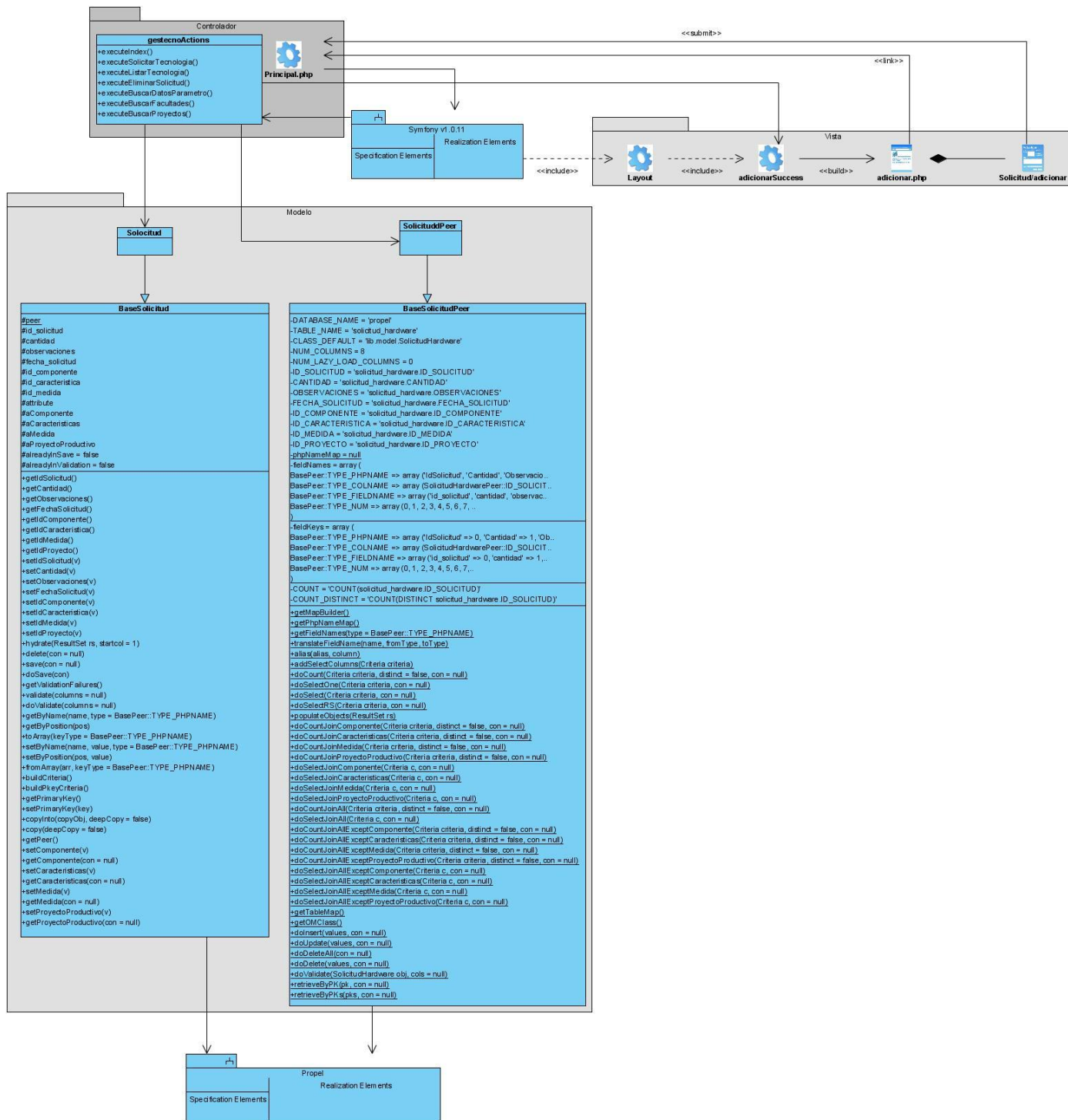
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Listar.



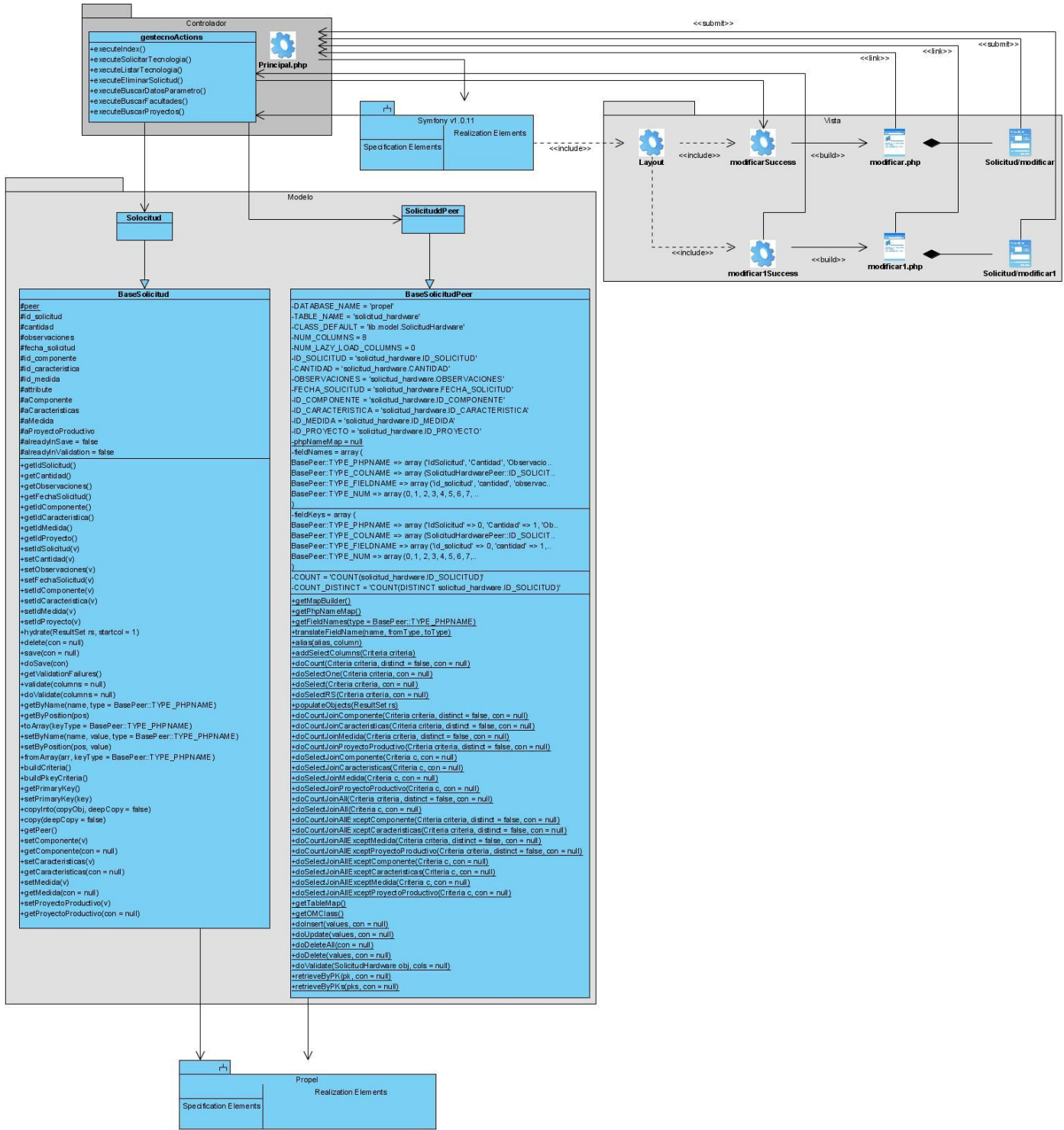
Diagramas Clases de Diseño del Caso de Uso: Gestionar Proyecto Productivo sección Eliminar.



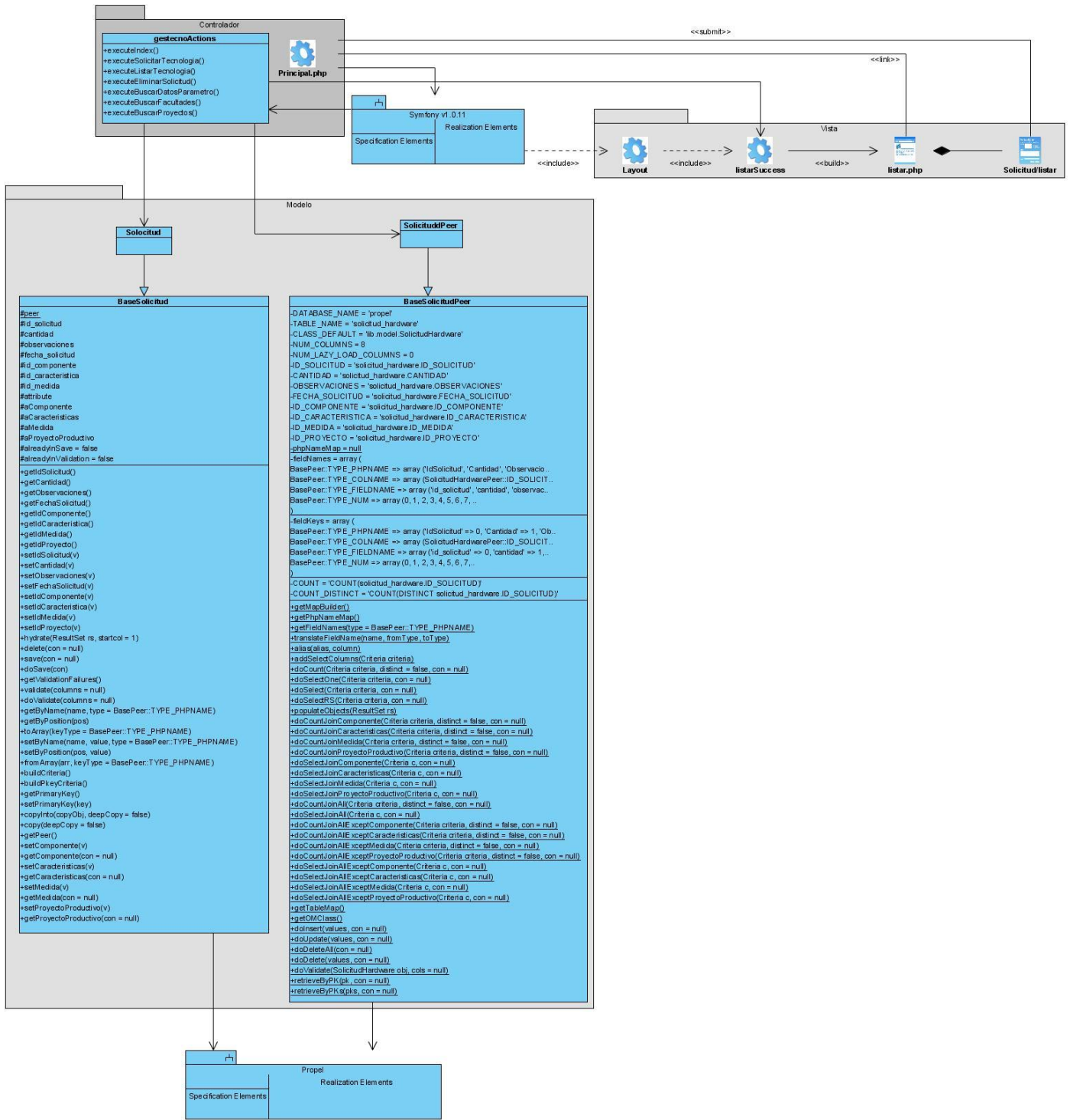
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Adicionar.



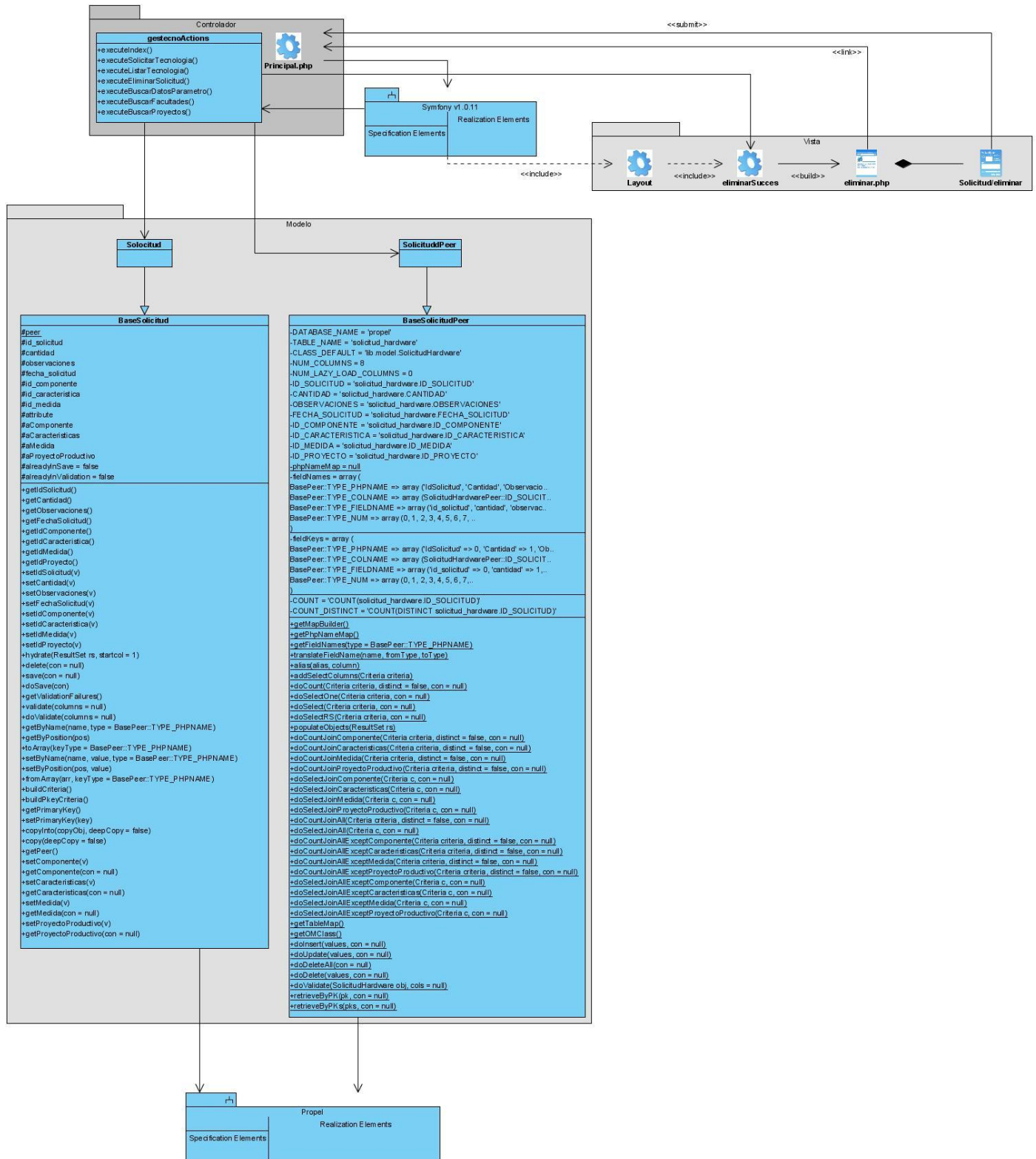
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Modificar.



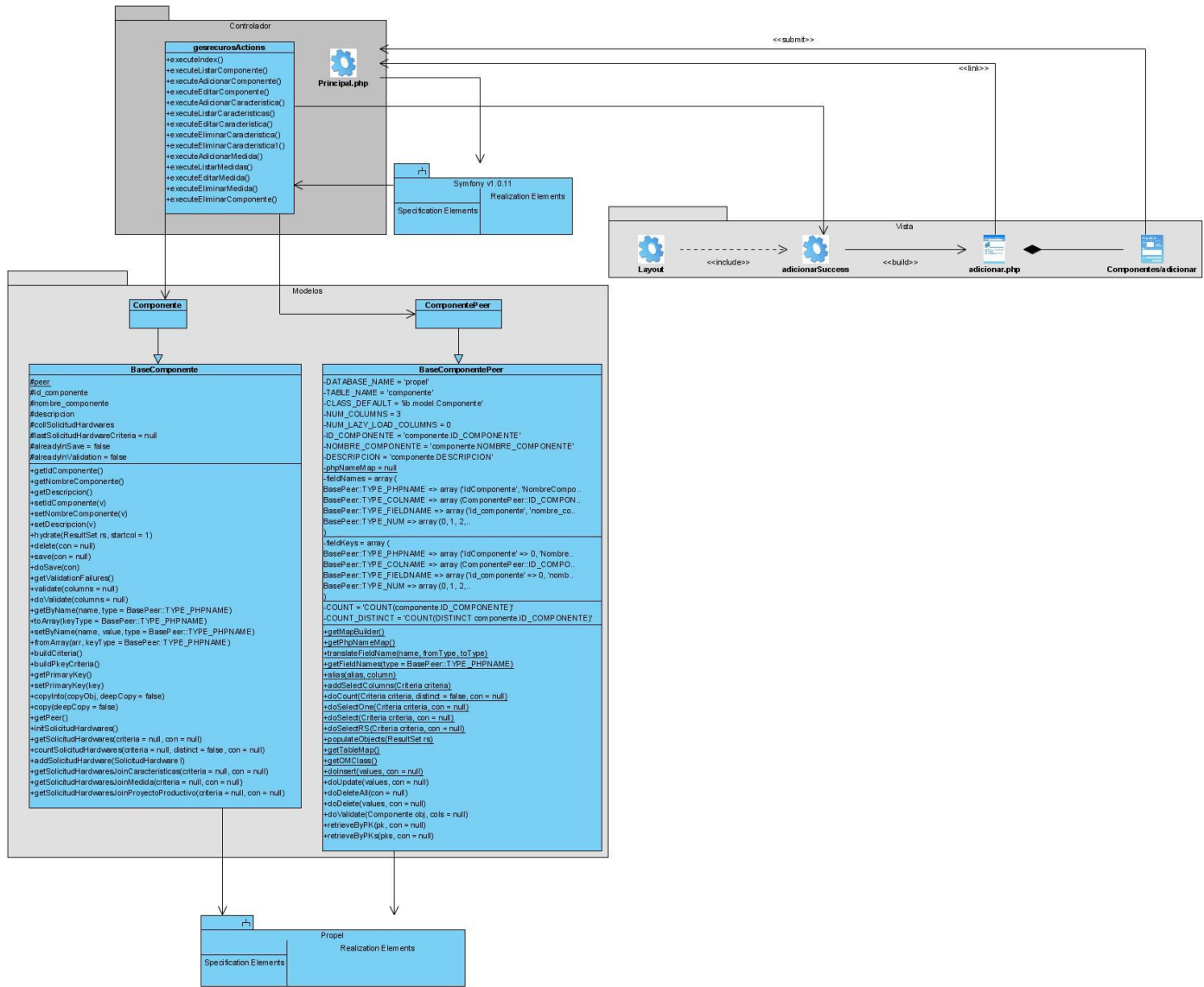
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Listar.



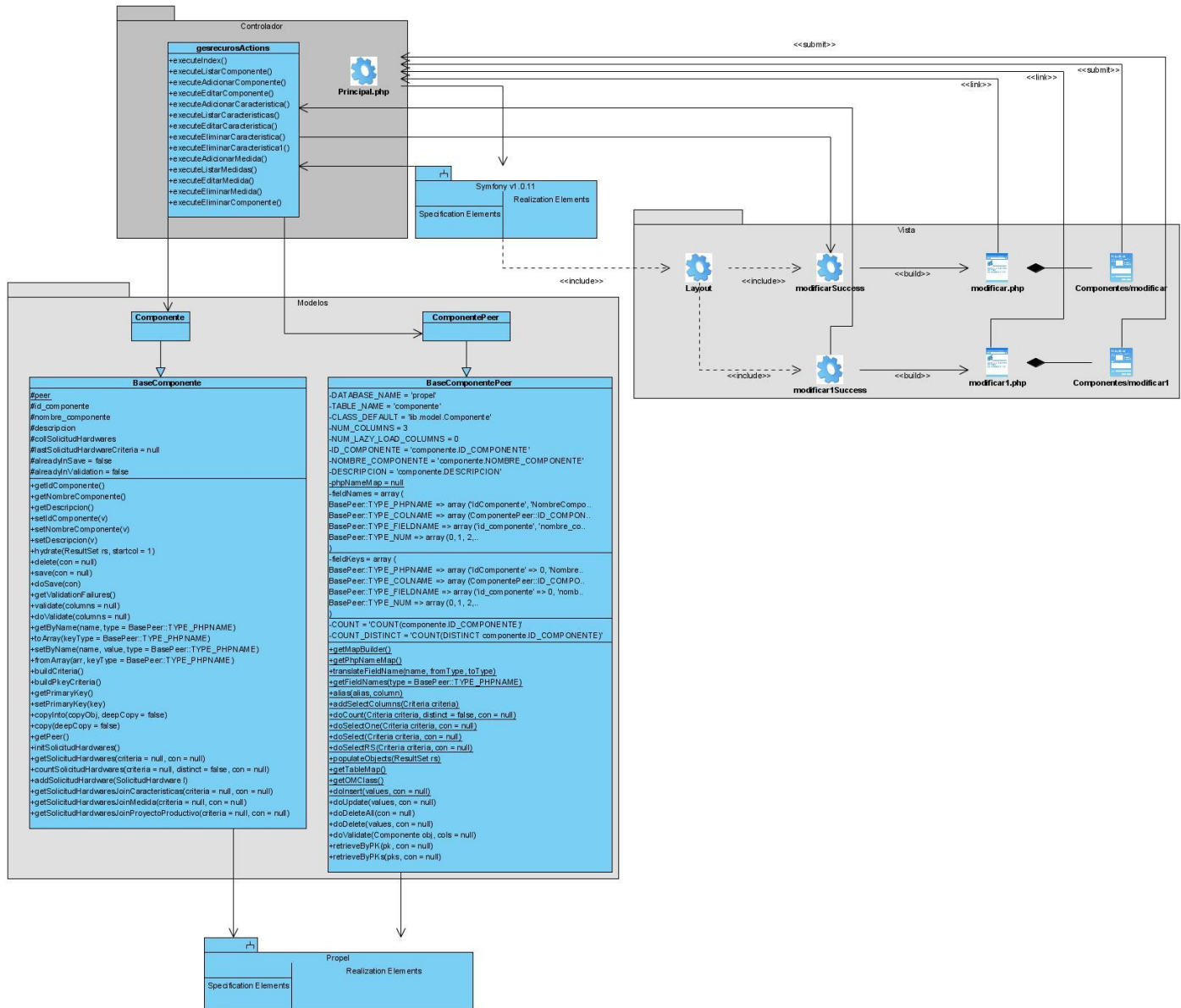
Diagramas Clases de Diseño del Caso de Uso: Gestionar Solicitud sección Eliminar.



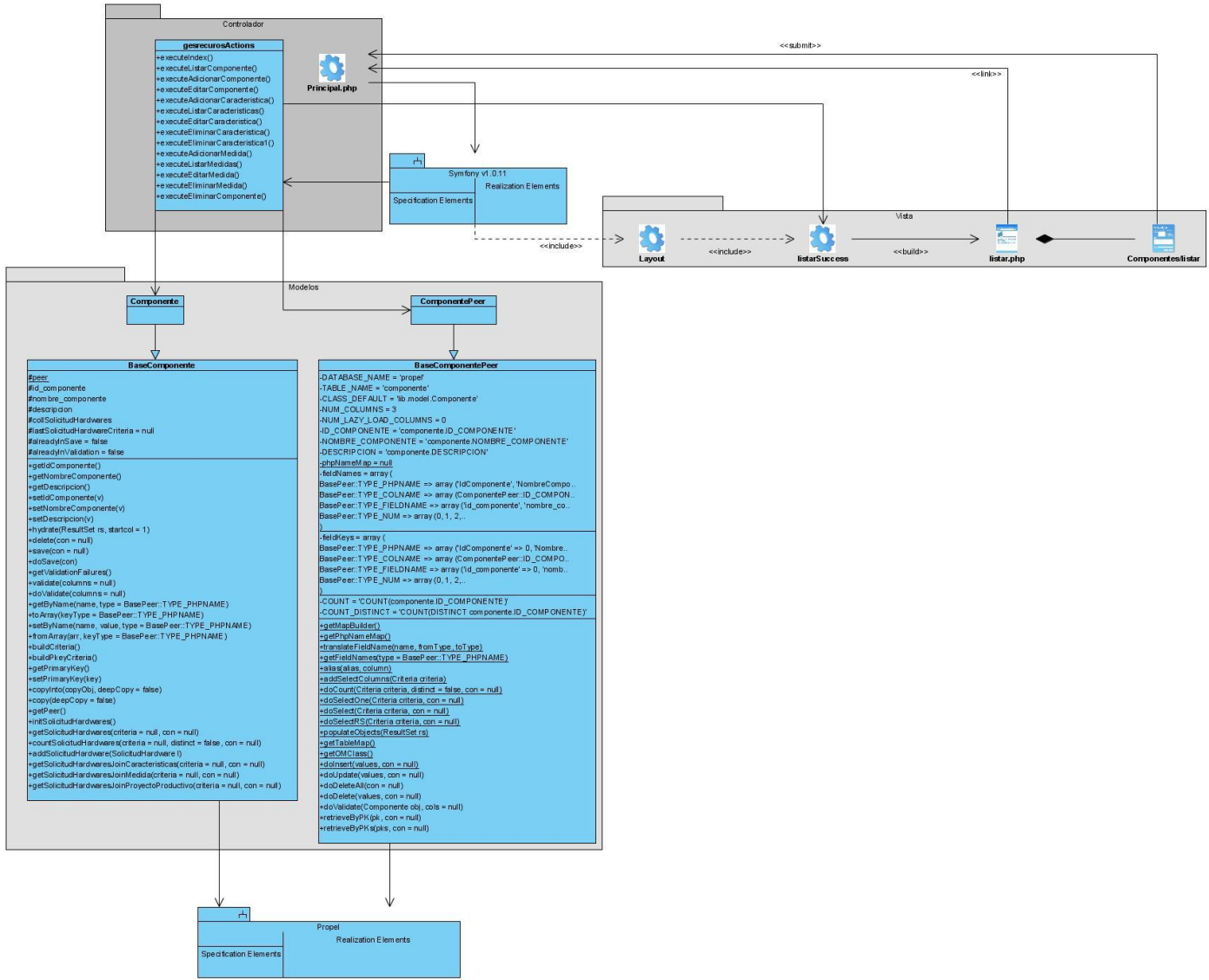
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Adicionar.



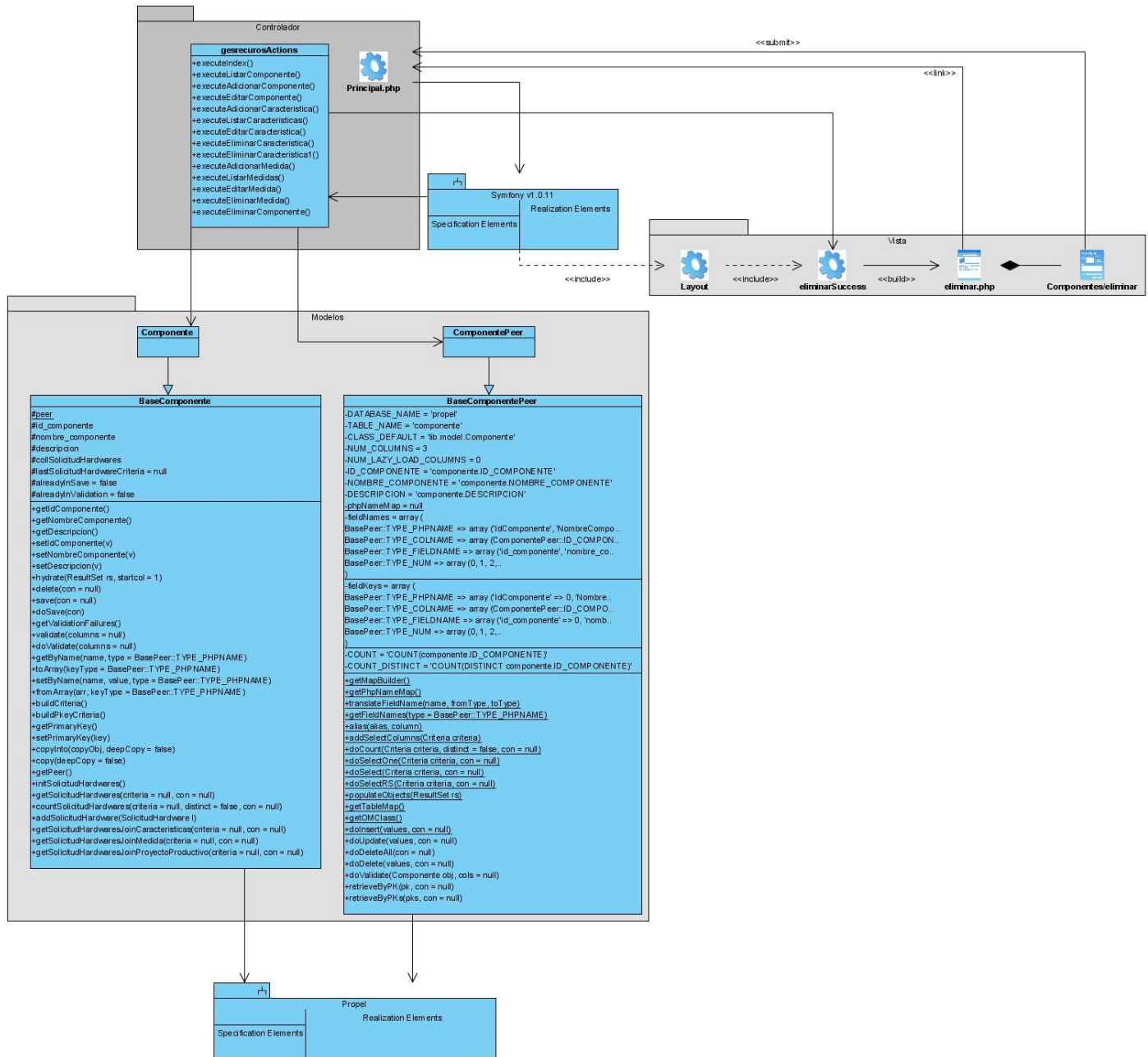
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Modificar.



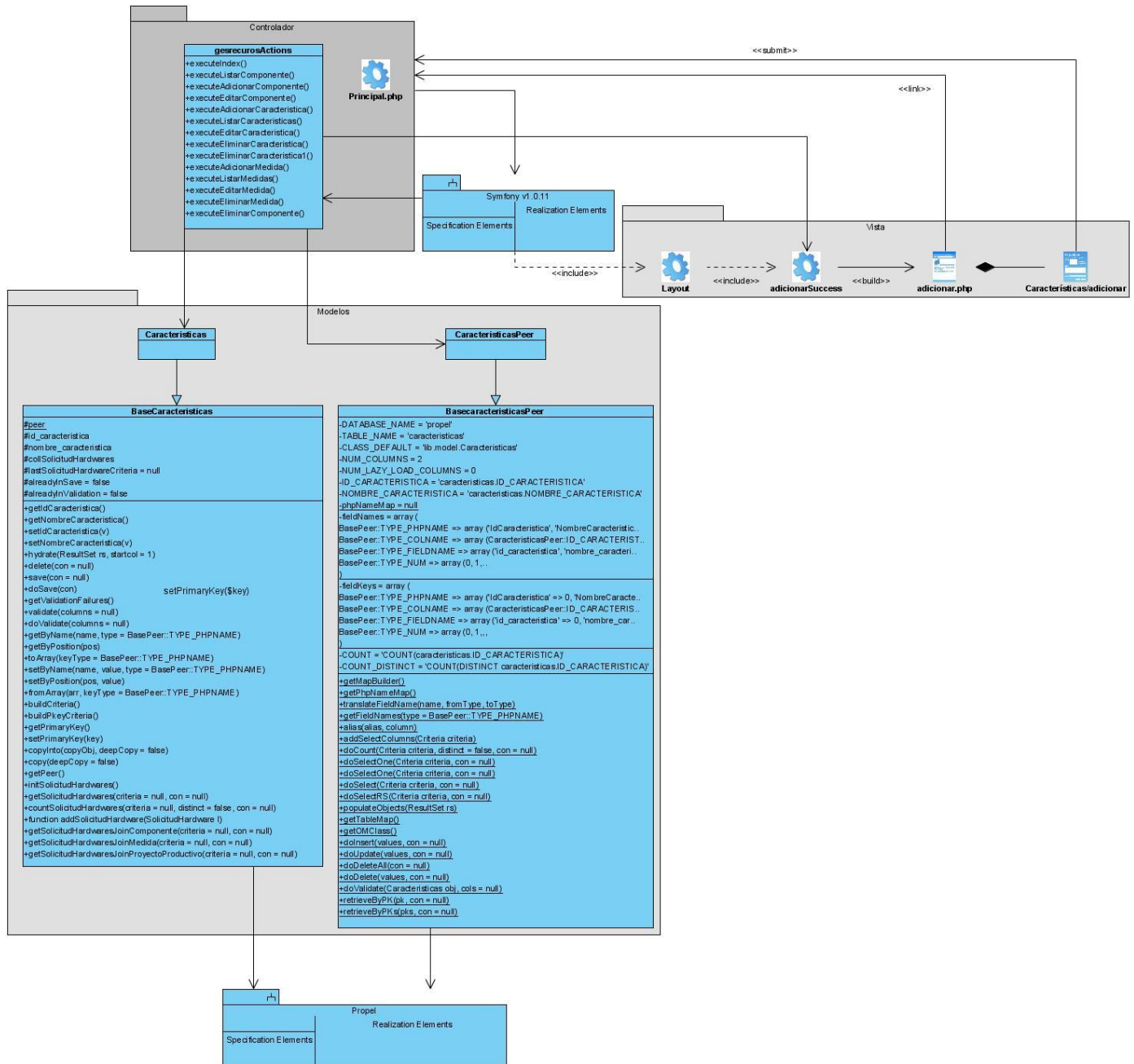
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Listar.



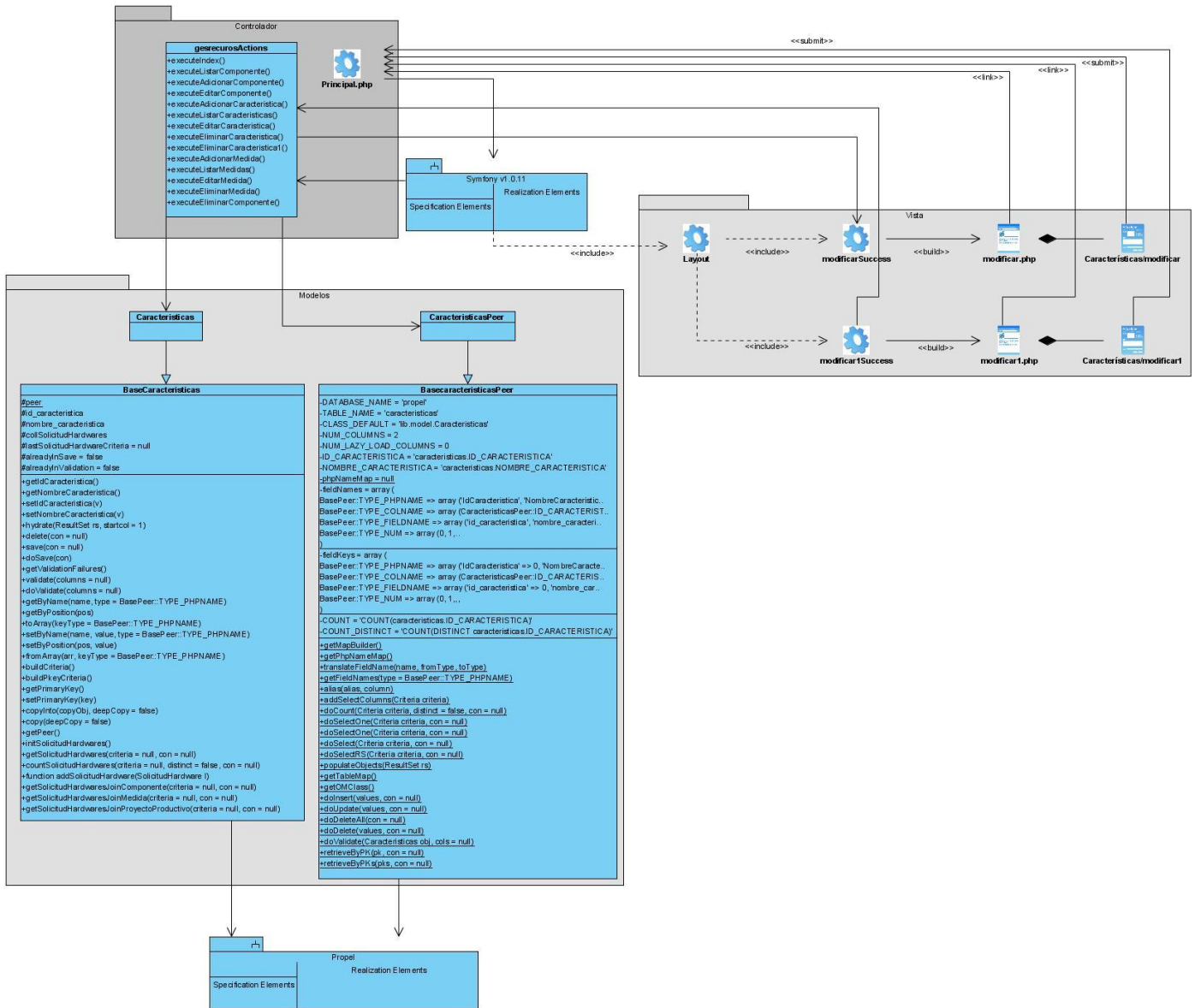
Diagramas Clases de Diseño del Caso de Uso: Gestionar Componentes sección Eliminar.



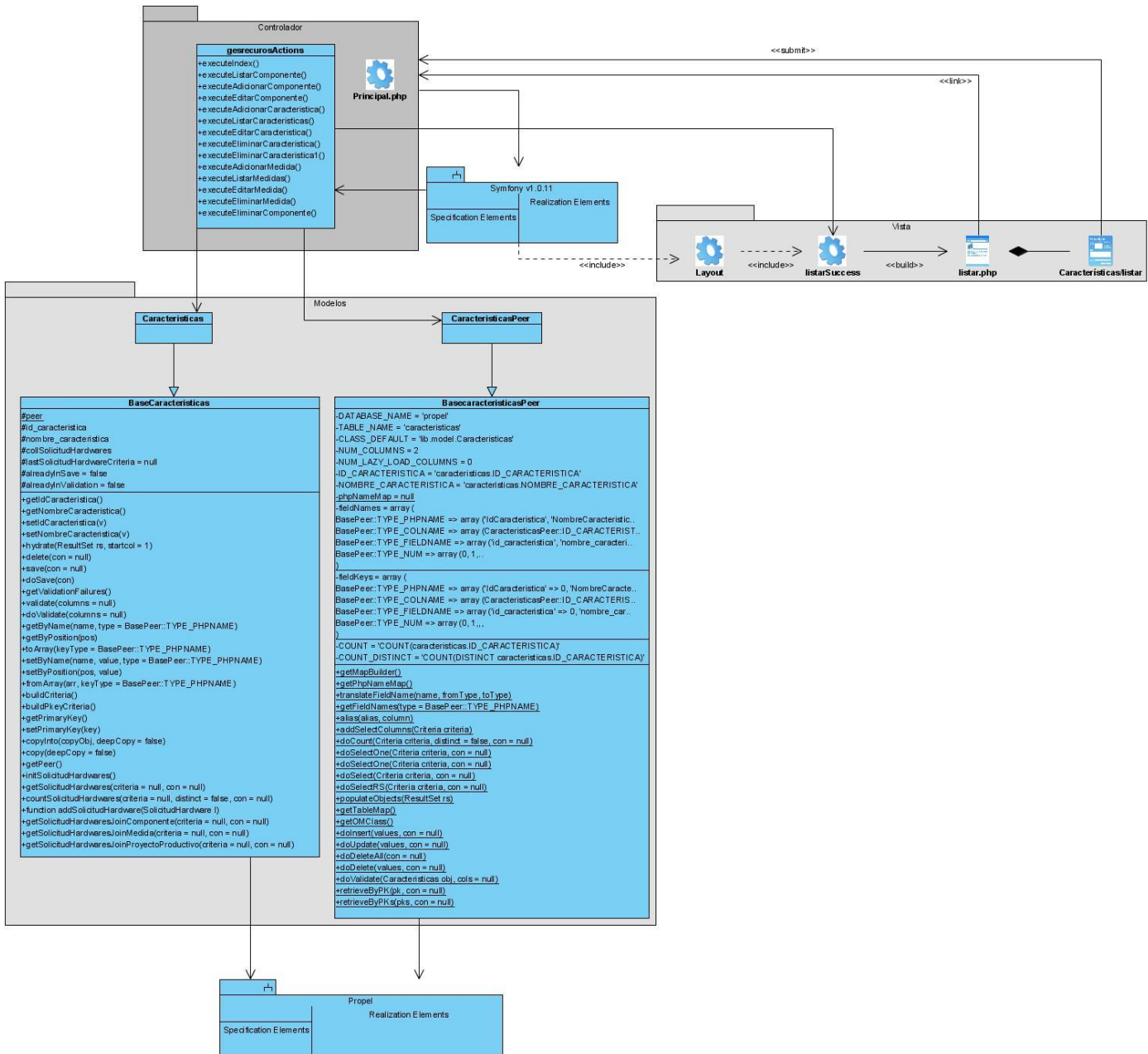
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Adicionar.



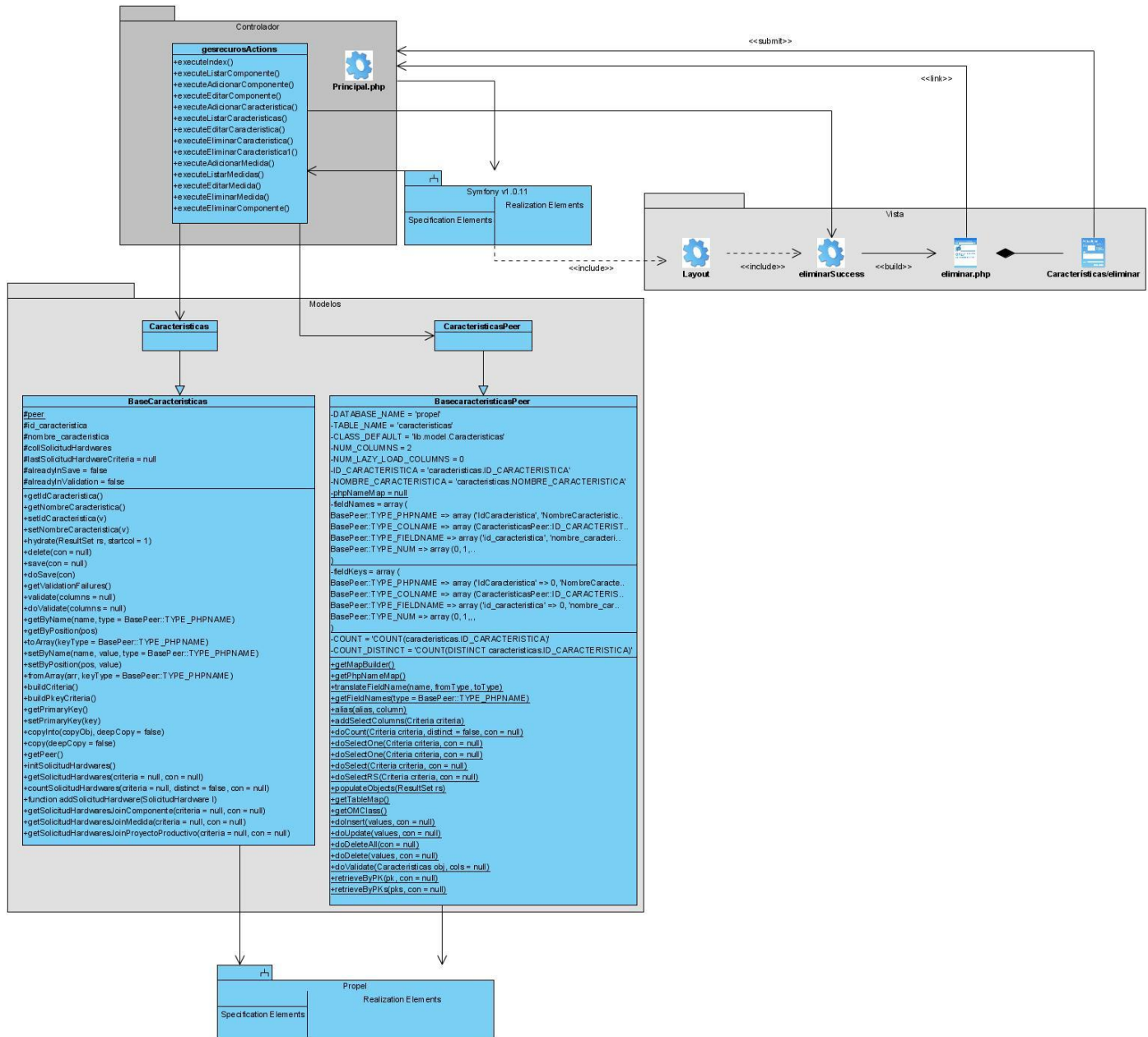
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Modificar.



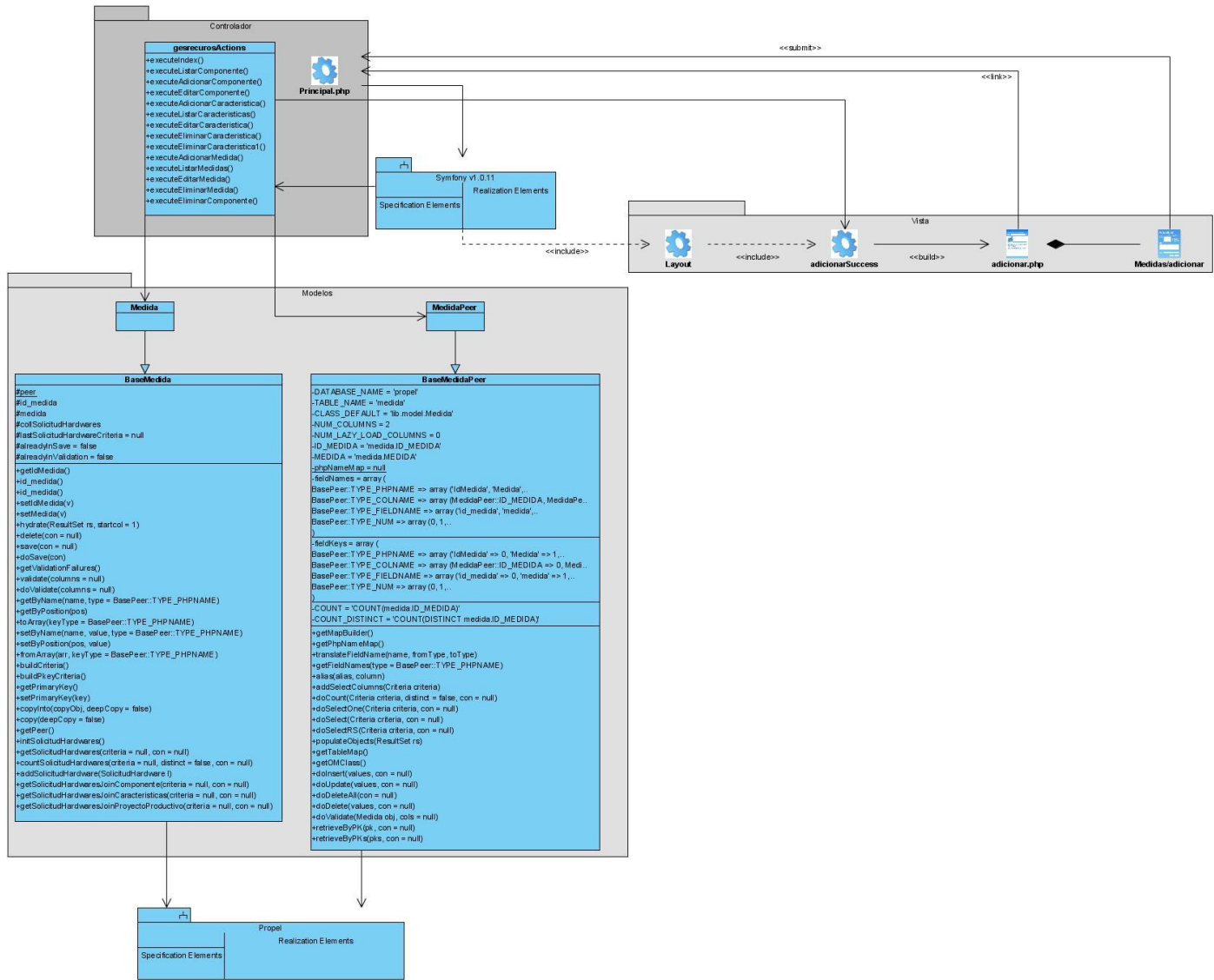
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Listar.



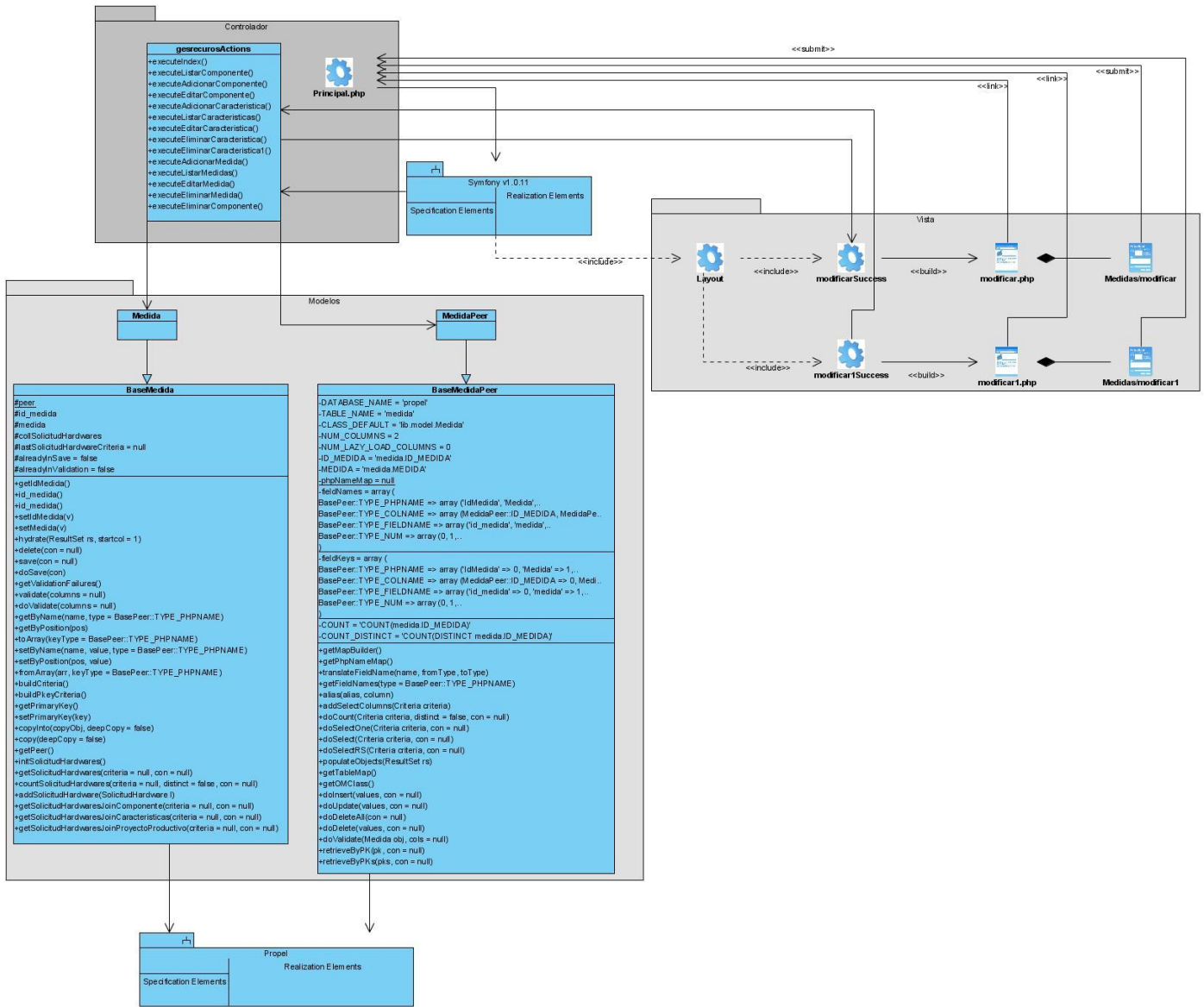
Diagramas Clases de Diseño del Caso de Uso: Gestionar Características sección Eliminar.



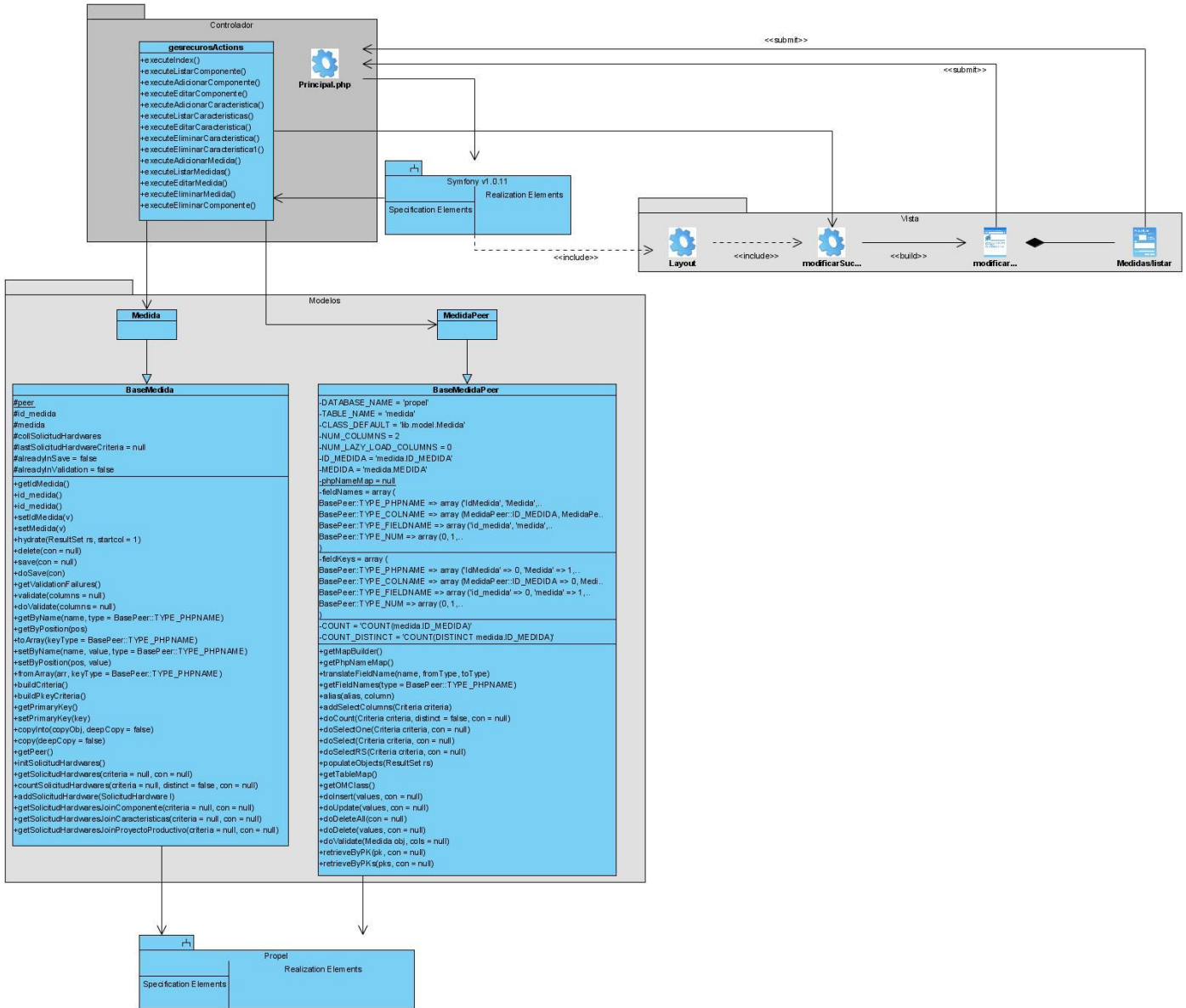
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Adicionar.



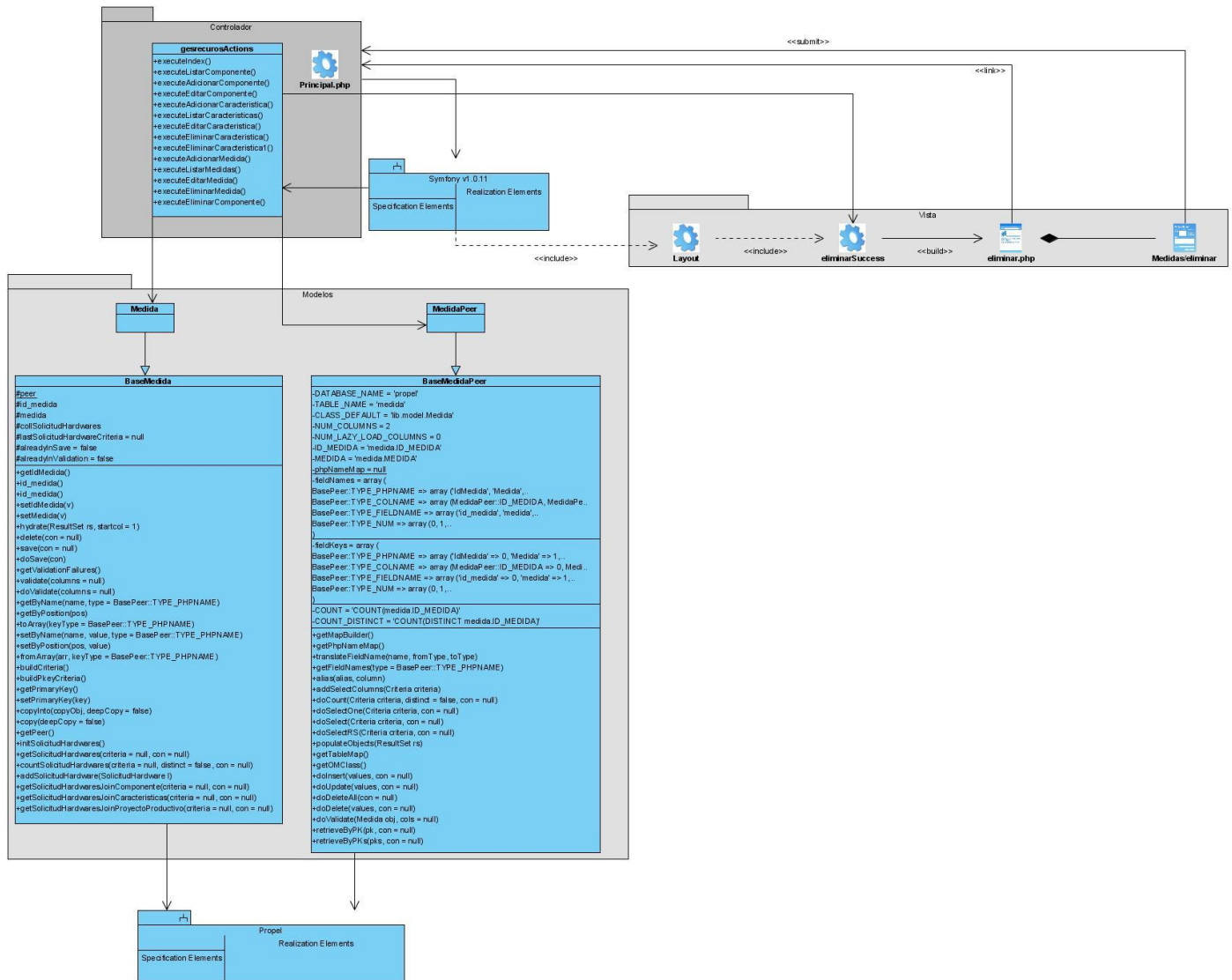
Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Modificar.



Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Listar.



Diagramas Clases de Diseño del Caso de Uso: Gestionar Medidas sección Eliminar.



GLOSARIO DE TÉRMINOS

Término	Definición
TIC	Tecnología de la Información y las Comunicaciones.
GSD	Grupo de Soporte de Desarrollo
PHP	PHP Hypertext Pre-processor
HTML	HyperText Markup Language
XML	Extensible Markup Language
XHTML	Extensible HyperText Markup Language
CSS	Hojas de Estilo en Cascada
UML	Lenguaje Unificado de Modelado
RUP	Proceso Unificado de Rational
XP	Xtreme Programming
IDE	Entorno Integrado de Desarrollo
GNU	GNU Not Unix