

Universidad de las Ciencias Informáticas

Facultad 3



Título: Sistema para la evaluación y Reporte del Estado de proyectos.

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

Autores:

Maylen Fiffe Hernández
Ernesto Miguel Suárez Peña

Tutor:

MSc. Rolando Bonal Cáceres

Ciudad de la Habana

Junio, 2009

“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.”

Aristóteles.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Maylen Fiffe Hernández

Firma del Autor

Ernesto Miguel Suárez Peña

Firma del Autor

Rolando Bonal Cáceres

Firma del Tutor

Son muchas las personas que de una forma u otra han colaborado con el desarrollo de este Trabajo de Diploma, en mi formación como profesional y en mi vida personal. Gracias a todos aquellos que siempre me han apoyado, y a aquellos que han hecho la diferencia.

A mi tutor: por toda su ayuda, paciencia y apoyo, sin él este trabajo no hubiera salido adelante. Muchas gracias.

A Pascual: quien ha sido mi padre durante estos 5 años.

A Chung: realmente no tengo palabras para agradecerte todo lo que has hecho por mí, sin tu ayuda no estuviera escribiendo estos agradecimientos ahora.

A Yanet: mas que mi prima eres mi hermana, gracias por todo lo que has hecho por mí.

A la embajada: mis amigas, mi familia, gracias por permitirme compartir tantas experiencias todos estos años.

A Verónica, Lisbeth, Dailín, Celia, Maylen: gracias por brindarme su amistad y saber que puedo contar con ustedes. A todas las quiero mucho.

A mi familia: por todo su cariño y ayuda.

Maylen.

A mis padres quienes me guiaron por el sendero del saber e inculcarme cada día el deseo de formarme profesionalmente.

A toda mi familia por el apoyo en momento que los necesité.

A mi tutor por su amistad, su dedicación, su profesionalidad y su apoyo incondicional.

A mis amigos, compañeros de aula, a todos los que de una forma u otra contribuyeron al desarrollo del trabajo de diploma, expresarles mis sinceros agradecimientos y gratitud.

A todos: Muchas Gracias.

Ernesto.

A mis padres: quienes me han brindado tanto amor y han guiado mis pasos.

A mis abuelos: que tanto me quieren y siempre me han apoyado.

A mis hermanitos: por darme tantas alegrías y brindarme un poco de su inocencia.

A Zaida y Toni: por quererme como a una hija y brindarme todo su cariño.

Maylen

*A mis padres Lisset Peña y Ernesto Suárez que han sido todo para mí en esta vida, por ustedes soy
alguien en esta vida, gracias por confiar en mí.*

A alguien especial, mi abuela Ana Escobar López, gracias por tu apoyo incondicional.

A mis primos, Kenia y Marcos.

A todos lo que me ayudaron de una forma u otra en mi formación como profesional.

Ernesto

En la actualidad la Universidad de las Ciencias Informáticas cuenta con una gran variedad de proyectos productivos. La situación actual de estos proyectos debe ser reportada continuamente con el objetivo de conocer el estado real de los mismos. Actualmente, la Universidad no posee un sistema generalizado a nivel de todos los proyectos que facilite esta información.

En el presente trabajo de diploma se desarrolla una aplicación que da respuesta a la problemática existente. La misma permitirá al usuario formular el Reporte del Estado de un proyecto con el propósito de comunicar a los interesados el estado actual del mismo. Proporcionará además el Reporte Postmortem con el objetivo de evaluar el producto producido y mejorar un nuevo proyecto a partir de las experiencias anteriores.

La aplicación fue desarrollada con el lenguaje de programación PHP 5 y el sistema gestor de base de datos PostgreSQL. Como metodología de desarrollo se utilizó RUP, como lenguaje de modelado UML y Visual Paradigm como herramienta de modelado.

Palabras Clave:

Reporte de Estado, Reporte Postmortem.

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
Gestión de proyectos.....	5
Gestión del alcance	5
Gestión del tiempo.....	7
Gestión del costo	8
Monitoreo y Control.....	8
Gestión de la información	9
Toma de decisiones.....	10
Reporte	11
Reporte de Estado.....	12
Reporte Postmortem.....	12
Estado del arte de los Reportes de Estado.....	13
En el mundo.....	13
En la UCI	16
Proceso de desarrollo de software	18
Tecnologías y herramientas utilizadas.....	18
Aplicaciones Web	18
Lenguajes de Programación Web	19
Lenguajes del lado cliente	19
Lenguajes del lado servidor.....	21
Servidor Web	22

AJAX.....	23
Sistema Gestor de Bases de Datos	24
Framework.....	26
Metodología de desarrollo.....	27
Lenguaje de modelado UML	28
Ext JS	29
Herramientas utilizadas en la solución del sistema propuesto	29
Patrones de diseño y Arquitectura.....	30
Patrones de diseño GRASP	30
Patrones de diseño GoF	31
Patrón de arquitectura MVC.....	31
Conclusiones del Capítulo 1.....	33
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	34
Modelo del dominio	34
Objeto de automatización.....	36
Propuesta del sistema	36
Especificación de los requerimientos de software	38
Requerimientos Funcionales.....	38
Requerimientos No Funcionales	40
Definición de los casos de uso	42
Justificación de los actores del sistema	42
Descripción de los casos de uso.....	44
Conclusiones del Capítulo 2.....	74

CAPÍTULO 3: ANÁLISIS Y DISEÑO	75
Análisis.....	75
Diseño.....	75
Aplicación de patrones en Symfony	76
Patrones GRASP	76
Patrones GoF.....	77
Patrón MVC.....	78
Descripción de las clases.....	79
Diseño de la BD.....	80
Conclusiones del Capítulo 3.....	80
CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA	81
Implementación.....	81
Estándar de codificación.....	82
Modelo de Prueba	83
Métricas asociadas al diseño	84
Métrica Tamaño de clase (TC).....	84
Métrica Árbol de profundidad de herencia (APH).....	85
Pruebas de rendimiento (stress)	86
Análisis de los resultados de las Pruebas	87
Pruebas de Caja Negra	90
Conclusiones del Capítulo 4.....	90
CONCLUSIONES GENERALES.....	91
RECOMENDACIONES	92

BIBLIOGRAFÍA.....	93
ANEXOS.....	96
Anexo 1: Plantilla de Reporte de Estado.....	96
Anexo 2: Plantilla de Reporte Postmortem.....	103
Anexo 3: Encuesta sobre los Reporte de Estado y Reporte Postmortem realizada a las facultades .	107
Anexo 4: Plantilla propuesta por ALBET	110
Anexo 5: Diagramas de clases del análisis	111
Diagrama de clases del análisis del CU: Autenticar Usuario	111
Diagrama de clases del análisis del CU: Gestionar Área.....	111
Diagrama de clases del análisis del CU: Gestionar Polo	112
Diagrama de clases del análisis del CU: Gestionar Proyecto	112
Diagrama de clases del análisis del CU: Gestionar Rol	113
Diagrama de clases del análisis del CU: Gestionar Usuario	113
Diagrama de clases del análisis del CU: Gestionar Criterio de Valoración	114
Diagrama de clases del análisis del CU: Registrar Reporte de Estado.....	114
Diagrama de clases del análisis del CU: Registrar Reporte Postmortem.....	115
Diagrama de clases del análisis del CU: Emitir Criterio de Valoración.....	115
Diagrama de clases del análisis del CU: Visualizar Reporte.....	115
Diagrama de clases del análisis del CU: Generar Informe.....	116
Anexo 6: Diagramas de clases del diseño	117
Diagrama de clases del diseño del CU: Autenticar Usuario.....	117
Diagrama de clases del diseño del CU: Gestionar Área	118
Diagrama de clases del diseño del CU: Gestionar Polo.....	119

Diagrama de clases del diseño del CU: Gestionar Proyecto.....	120
Diagrama de clases del diseño del CU: Gestionar Rol	121
Diagrama de clases del diseño del CU: Gestionar Usuario	122
Diagrama de clases del diseño del CU: Gestionar Criterio de Valoración.....	123
Diagrama de clases del diseño del CU: Registrar Reporte de Estado	124
Diagrama de clases del diseño del CU: Registrar Reporte Postmortem	125
Diagrama de clases del diseño del CU: Emitir Criterio de Valoración	126
Diagrama de clases del diseño del CU: Generar Informe	127
Anexo 7: Diagramas de secuencia.....	128
Diagrama de secuencia del diseño del CU: Autenticar Usuario	128
Diagrama de secuencia del diseño del escenario Adicionar Área del CU: Gestionar Área.	129
Diagrama de secuencia del diseño del escenario Editar Área del CU: Gestionar Área.....	130
Diagrama de secuencia del diseño del escenario Eliminar Área del CU: Gestionar Área.	131
Diagrama de secuencia del diseño del escenario Adicionar Polo del CU: Gestionar Polo.	132
Diagrama de secuencia del diseño del escenario Editar Polo del CU: Gestionar Polo.	133
Diagrama de secuencia del diseño del escenario Eliminar Polo del CU: Gestionar Polo.	134
Diagrama de secuencia del diseño del escenario Adicionar Proyecto del CU: Gestionar Proyecto.	135
Diagrama de secuencia del diseño del escenario Editar Proyecto del CU: Gestionar Proyecto.....	136
Diagrama de secuencia del diseño del escenario Eliminar Proyecto del CU: Gestionar Proyecto.	137
Diagrama de secuencia del diseño del escenario Adicionar Rol del CU: Gestionar Rol.....	138
Diagrama de secuencia del diseño del escenario Editar Rol del CU: Gestionar Rol.	139
Diagrama de secuencia del diseño del escenario Eliminar Rol del CU: Gestionar Rol.	140
Diagrama de secuencia del diseño del escenario Adicionar Usuario del CU: Gestionar Usuario... ..	141

Diagrama de secuencia del diseño del escenario Editar Usuario del CU: Gestionar Usuario. 142

Diagrama de secuencia del diseño del escenario Eliminar Usuario del CU: Gestionar Usuario..... 143

Diagrama de secuencia del diseño del escenario Adicionar Criterio de Valoración del CU: Gestionar Criterio de Valoración. 144

Diagrama de secuencia del diseño del escenario Editar Criterio de Valoración del CU: Gestionar Criterio de Valoración. 145

Diagrama de secuencia del diseño del escenario Eliminar Criterio de Valoración del CU: Gestionar Criterio de Valoración. 146

Diagrama de secuencia del diseño del CU: Registrar Reporte Postmortem 148

Diagrama de secuencia del diseño del CU: Emitir Criterio de Valoración 149

Diagrama de secuencia del diseño del CU: Generar Informe 150

Anexo 8: Descripción de las clases..... 151

Anexo 9: Diseño de la BD 171

Anexo 10: Diagrama de componentes 172

Anexo 11: Resultados de Pruebas de rendimiento..... 173

Anexo 12: Diseño de Casos de Prueba 175

Anexo 13: Fotos del sistema..... 178

INTRODUCCIÓN

Con el creciente desarrollo de la ciencia y las tecnologías en los últimos tiempos, la informática ha ido evolucionando notablemente, de tal forma que ha logrado convertirse en un ente fundamental en casi todos los sectores de la sociedad. Este desarrollo tecnológico combinado con el gran avance de las ciencias informáticas, ha propiciado que la complejidad de los proyectos de desarrollo de software haya incrementado considerablemente.

La Universidad de las Ciencias Informáticas no se encuentra exenta de este desarrollo, la misma cuenta con una fuerte base tecnológica y un amplio perfil productivo. Según la Infraestructura Productiva (IP) de la universidad, actualmente se desarrollan más de 100 proyectos productivos. Para lograr el éxito de estos proyectos, con la calidad requerida por los clientes, es necesario que se gestionen correctamente los mismos.

La gestión de proyectos es una disciplina que persigue como finalidad principal la planificación, organización y dirección de un proyecto, así como el seguimiento y control de las actividades y de los recursos humanos y materiales que se disponen en el desarrollo del mismo.

Dentro del conjunto de disciplinas que forman la gestión de proyectos se pueden mencionar: la gestión del alcance, la gestión del tiempo y la gestión del costo. Estas tres disciplinas son factores fundamentales que influyen en el éxito de un proyecto, pues se hace necesario determinar qué está y qué no está en el proyecto, que se cumpla con el plazo previsto y con los costos programados.

Para vigilar el correcto desarrollo de las actividades y tareas establecidas en el proyecto es preciso que se efectúe un adecuado monitoreo y control.

Como consecuencia de este control será posible conocer en todo momento qué problemas se producen y resolverlos o aminorarlos de manera inmediata.

El monitoreo y control deberá considerar como tema central, el control de las variaciones en el alcance, el control de las variaciones de la duración de las actividades y deberá contemplar el control de los costos.

La supervisión de las actividades incluye la recopilación, medición y difusión de información sobre el rendimiento, así como la evaluación de las mediciones y tendencias para llevar a cabo las mejoras del

proceso. Esta supervisión continua le proporciona al equipo de dirección del proyecto una idea acerca del estado del proyecto e identifica cualquier área que necesite más atención.

La gestión de toda la información generada durante el seguimiento de las actividades es decisiva para la toma de decisiones, pues de ella depende que se tomen decisiones efectivas sobre las dificultades que se presenten durante el desarrollo del proyecto.

Toda esta información se recoge en un Reporte de Estado del proyecto, con el objetivo de comunicarle al receptor cómo se va desarrollando el proyecto según lo planeado y por qué.

Actualmente, la UCI no cuenta con un sistema generalizado para la gestión de la información del estado actual de los proyectos, lo que dificulta a las direcciones de la producción o a las facultades, tener un control exhaustivo de toda la documentación referida al estado de los proyectos. Es por ello que resultaría conveniente para la Universidad contar con un sistema que propicie toda esta información y facilite el seguimiento del desarrollo de los mismos.

Por todo lo antes descrito se plantea el siguiente **Problema Científico de la Investigación**: No existe, establecido y generalizado a nivel de todos los proyectos de software de la Universidad de las Ciencias Informáticas, un sistema de procesos o informático para la gestión de la información referente al estado y evaluación del rendimiento durante el proceso de desarrollo de software.

Dicho problema se enmarca dentro del **Objeto de estudio**: Proceso de desarrollo del software, y **Campo de Acción**: Gestión de la información de estado en proyectos de software.

Para darle solución al problema científico planteado anteriormente, el presente trabajo tiene como **Objetivo general**: Desarrollar un sistema que permita de forma eficiente la gestión de la información del estado y el rendimiento durante el proceso de desarrollo de software en los proyectos de la UCI.

Esta investigación tiene como **Idea a defender**: El desarrollo de un sistema de la gestión de la información de estado de los proyectos y su rendimiento, facilitará una explotación más eficiente de los recursos asignados a la producción y contribuirá a la toma de decisiones efectiva sobre las dificultades que se presenten durante el desarrollo.

Para lograr lo planteado en el objetivo general y para desarrollar adecuadamente la idea a defender, se han definido un conjunto de **tareas** que se llevarán a cabo durante el desarrollo de este trabajo:

- Investigar en las diversas fuentes bibliográficas sobre la gestión de la información referente a la evaluación del estado y el rendimiento en los proyectos.
- Definir las necesidades y madurez de la UCI, en cuanto a la gestión informatizada de estado y el rendimiento en los proyectos de desarrollo de software.
- Realizar un levantamiento sobre las variantes con que actualmente se realizan los Reportes de Estado y Reportes Postmortem en los proyectos de software de la UCI.
- Realizar la selección de las herramientas más apropiadas para el desarrollo del sistema, así como del lenguaje y metodología a utilizar.
- Realizar el análisis, diseño e implementación del sistema.
- Evaluar el sistema propuesto desde la perspectiva legal, de su rendimiento ante las condiciones de explotación que se prevén y de su extensibilidad.
- Validar la propuesta del sistema para la gestión informatizada de proyectos en la Universidad de las Ciencias Informáticas.

Para lograr el cumplimiento de las tareas antes planteadas se hizo necesario utilizar determinados **Métodos Científicos**. En el presente trabajo se utilizaron los métodos teóricos y los métodos empíricos.

Los **Métodos Teóricos** posibilitan el estudio de las características del objeto de investigación que no se pueden visualizar de forma directa y proporcionan la construcción de modelos e hipótesis de investigación. De ellos se utilizaron dos en el desarrollo de la investigación:

- **Analítico – sintético:** Se utilizó para el procesamiento de toda la información relacionada con el tema de investigación, analizando los documentos que permitieran extraer los elementos más significativos relacionados con el objeto de estudio.
- **Histórico – lógico:** Este método brinda la posibilidad de analizar la trayectoria histórica real del tema de investigación y constatar teóricamente cómo ha evolucionado el tema tratado durante un determinado período de tiempo.

Los **Métodos Empíricos** describen y explican las características de los objetos, representan un nivel de la investigación cuyo contenido proviene de la experiencia y es sometido a cierta elaboración racional. Se utiliza de estos métodos:

- **Observación:** La aplicación de este método permitió a los actores de esta investigación observar el cumplimiento de las características que debe poseer el sistema y garantizar el cumplimiento de los objetivos trazados.
- **Encuesta:** Este método posibilitó la realización de encuestas a los líderes de proyecto, con el fin de conocer la situación real en que se encuentran los proyectos en cuanto al tema de investigación, y recopilar la información necesaria para la implementación del sistema.

Estructuración del contenido:

El presente trabajo se encuentra estructurado en 4 capítulos.

Capítulo 1: aborda todo lo relacionado con la fundamentación teórica del tema de investigación y una pequeña descripción de las herramientas utilizadas en la implementación del sistema propuesto.

Capítulo 2: en este capítulo se explican las características del sistema propuesto, los procesos que serán objeto de automatización, los requerimientos del sistema y los diagramas pertinentes.

Capítulo 3: abarca el desarrollo de la fase de análisis y diseño de software usando la metodología RUP, se mostrarán y describirán los diagramas de clases del análisis y del diseño de los casos de uso definidos en el capítulo anterior, así como los diagramas de secuencia correspondientes a dichos casos de uso. Se describen además los patrones aplicados para la construcción del sistema.

Capítulo 4: comprende detalles de la fase de elaboración para la implementación del sistema. Se expondrán los diagramas de despliegue y componente pertenecientes a los casos de uso definidos en las iteraciones anteriores, así como las pruebas necesarias para la validación del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan conceptos que constituyen la fundamentación teórica de los elementos investigados. Se hace un estudio del estado del arte de los Reportes de Estado de Proyectos en el mundo y en nuestra Universidad específicamente; y se realiza una breve descripción de las tecnologías y herramientas utilizadas en el proceso de desarrollo de la aplicación.

Gestión de proyectos

La gestión de proyectos es una disciplina que permite la planificación y dirección de proyectos. La misma debe garantizar que los proyectos se desarrollen convenientemente y se obtengan óptimos resultados. Debe permitir el control de la evolución del proyecto y poder explicarlo de forma satisfactoria al equipo de trabajo y al cliente (Rodríguez, y otros, 2007).

Para lograr el resultado deseado en un proyecto y la satisfacción del cliente se debe efectuar un seguimiento durante todo el desarrollo del proyecto. Es preciso llevar un control de las actividades que se realizan en el proyecto, de los recursos necesarios para llevar a cabo su ejecución, así como todos los componentes necesarios para que el proyecto se desarrolle adecuadamente y no se desvíe del cumplimiento de los objetivos planteados inicialmente. Se requiere además de una adecuada planificación para que las tareas, hitos o metas a cumplir se desarrollen en tiempo. Es por ello que todo líder de proyecto debe conocer el triángulo de la gestión de proyectos, o sea, debe tener bien claro cuál es el alcance del proyecto, el tiempo que se requiere para su desarrollo y el presupuesto necesario para completarlo. Estos tres factores deben quedar plasmados en el Reporte de Estado del proyecto, para que éste, como parte del monitoreo y control del proyecto, gestione toda la información necesaria para una toma de decisiones efectiva en el proyecto.

Gestión del alcance

Todo Reporte de Estado debe incluir el alcance del proyecto, esto permite a los interesados conocer si el proyecto se va desarrollando según lo planeado o si han surgido afectaciones sobre el alcance del proyecto.

La Gestión del alcance incluye las actividades necesarias que garanticen que el proyecto desarrollará sólo el trabajo requerido para completarlo satisfactoriamente (Turner, 1993). Su principal objetivo es proteger la viabilidad del proyecto.

El alcance de un proyecto define las fronteras del mismo, limita hasta dónde debe llegar el proyecto en cuanto a sus objetivos, costo, tiempo, funcionalidad, etc (Serer Figueroa, 2001).

Existen un grupo de procesos dentro de la Gestión del alcance que generalmente se definen como parte del ciclo de vida del proyecto y se documentan en el plan de gestión del alcance del proyecto (PMI, 2004):

1. Planificación del alcance: se crea un plan de gestión del alcance del proyecto que proporcione la información sobre cómo el equipo de dirección del proyecto va a definir, documentar, verificar, gestionar y controlar el alcance del proyecto.
2. Definición del alcance: se desarrolla un enunciado del alcance del proyecto bien detallado, que especifique claramente que trabajo se realizará y que trabajo quedará excluido, que proporcione además un buen entendimiento del alcance del proyecto entre los usuarios y precise los principales objetivos del proyecto.
3. Crear documento: desglosa el trabajo del proyecto en fracciones más pequeñas y cómodas de manejar. Organiza y define el alcance total del proyecto.
4. Verificación del alcance: se formaliza la aceptación del alcance del proyecto completado y los productos entregables relacionados.
5. Control del alcance: controla los cambios en el alcance del proyecto y su impacto.

El cumplimiento del alcance del proyecto se evalúa en comparación con el plan de gestión del proyecto, el enunciado del alcance del proyecto y toda su documentación.

Una buena gestión del alcance es indispensable para evitar retrasos en el proyecto y garantizar el cumplimiento de los requisitos funcionales.

En el libro *Controlling Software Projects*, Tom DeMarco expresa: “aquellos gerentes que hayan tenido éxito en congelar el alcance de sus proyectos, tendrán oportunidad de conversar sobre ello, cuando se encuentren en las filas de desempleados”.

Es importante destacar que el “principio” de la gestión del alcance no es precisamente “rechazar cualquier cambio”, sino que “si se tenía un compromiso y la situación cambió, el compromiso que se tenía debe ser revisado y actualizado” (Llorens Fabregas, 2005).

Gestión del tiempo

La Gestión del tiempo es una disciplina que juega un papel importante a la hora de elaborar el Reporte de Estado de un proyecto. En este documento debe quedar bien definido el tiempo de cada una de las tareas durante todo el período que abarca el reporte, pues mediante este documento se podrá conocer si el proyecto va marchando correctamente según el tiempo programado.

La Gestión del tiempo organiza el tiempo de forma tal que se pueda gestionar todo lo que se maneja, surge o interfiere en el trabajo a realizar (Esteban Alberdi, y otros, 2007).

La Gestión del Tiempo del Proyecto encierra un grupo de procesos necesarios para que el proyecto termine en el tiempo requerido (PMI, 2004):

1. Definición de las Actividades: se identifica y documenta todo el trabajo que se planifica realizar, o sea, todas aquellas actividades que se deben realizar para producir los diferentes productos entregables del proyecto.
2. Establecimiento de la Secuencia de las Actividades: implica la identificación y documentación de todas las dependencias entre las actividades del cronograma.
3. Estimación de Recursos de las Actividades: determina qué tipo de recurso se utilizará, la cantidad disponible y cuándo se puede utilizar para realizar cada actividad del proyecto.
4. Estimación de la Duración de las Actividades: estima la cantidad de esfuerzo de trabajo necesario, la cantidad prevista de recursos a ser aplicados y determina la cantidad de períodos laborables necesarios para completar las actividades del cronograma.

5. Desarrollo del Cronograma: determina las fechas de inicio y fin para cada actividad planificada del proyecto.
6. Control del Cronograma: controla los cambios en las actividades del proyecto.

Gestión del costo

Es importante que en el Reporte de Estado se incluya el estado financiero del proyecto, de tal forma que se pueda conocer la varianza entre el presupuesto planeado y el costo actual del proyecto. Es necesario que el costo del proyecto se gestione correctamente, pues no es factible que el presupuesto gastado sea mayor que el presupuesto asignado.

La Gestión del costo de un proyecto se encarga de gestionar el costo de los recursos que se utilizarán en las actividades desarrolladas en el proyecto.

La Gestión del costo incluye todos aquellos procesos implicados en la planificación, estimación, preparación del presupuesto y control de costos de manera tal que el proyecto se pueda realizar con el presupuesto admitido. Estos procesos incluyen las siguientes tareas (PMI, 2004):

1. Estimación de Costos: se desarrolla una aproximación de los costos de los recursos necesarios para completar las actividades del cronograma.
2. Preparación del Presupuesto de Costos: se ocupa de sumar los costos estimados de las actividades del proyecto o paquetes de trabajo, con el fin de establecer una línea base de costo y poder medir el rendimiento del proyecto.
3. Control de Costos: investiga sobre los factores que influyen en las variaciones del costo y controla los cambios en el presupuesto del proyecto.

Monitoreo y Control

El monitoreo y control es un conjunto de actividades de gestión que permiten verificar si el proyecto va marchando según lo planificado. Controla el avance del proyecto en su ejecución, compara el desempeño y mide los resultados reales contra lo planeado, y revisa el comportamiento de los indicadores de desempeño (Gómez Sánchez S, 2007).

El monitoreo y control es una necesidad de cada proyecto para lograr resultados exitosos. Debe hacerse de forma regular y consistente, y debe monitorear la diferencia entre lo planificado y lo real. Es una actividad que muestra cuándo y dónde existieron desviaciones al plan y pone en marcha las acciones correctivas para que el proyecto retorne a su camino normal. Mediante el monitoreo y control se puede comprobar la gestión del alcance, la gestión del tiempo y la gestión del costo, o sea, que el alcance del proyecto se haya establecido correctamente, se examina la programación del proyecto, se revisa la línea base y se controla que se hayan estimado los recursos, en calidad, cantidad y oportunidad. El monitoreo y control es la acción de verificar que se realicen adecuadamente los reportes previstos para el control del cumplimiento del proyecto, y se valoren los resultados operativos que va teniendo el proyecto durante todo su desarrollo.

Para realizar un control efectivo se deben tener en cuenta dos aspectos fundamentales: la función del control debe centrarse más en prevenir los problemas que en arreglarlos y debe relacionarse tanto con la calidad del producto, como con su cantidad y oportunidad.

El seguimiento de las actividades del proyecto es una de las razones del 95% de su progreso, pues permite medir la verdadera situación del proyecto, y por consiguiente, el avance real del mismo (Llorens Fabregas, 2005).

Gestión de la información

La gestión de la información es de vital importancia en el desarrollo de un proyecto, pues toda la información que se genera durante el monitoreo y control debe incluirse en el Reporte de Estado del proyecto. Toda esa información permitirá a los responsables conocer la situación actual del proyecto y realizar decisiones efectivas sobre el mismo.

La gestión de la información se puede definir como un grupo de actividades con el propósito de recopilar, supervisar y recuperar la información producida por cualquier organización durante el desarrollo de sus actividades (Boletín del Instituto Andaluz de Patrimonio Histórico, 2001).

La gestión de información debe comprender toda la información requerida, desde la identificación de las necesidades hasta el uso de la misma.

En una organización es fundamental que la gestión de la información se base en el comportamiento de sus integrantes y en el nivel en que reconozcan que la información, como recurso, demanda niveles de atención equivalentes o superiores a otros recursos como las finanzas o las propias personas.

Una verdadera gestión de información evalúa constantemente el desempeño de la organización; continuamente analiza el ciclo de vida de cada información y si ésta sufre variaciones en la medida que la dinámica de la organización se modifique. Es importante tener presente que la gestión de información exige evaluar permanentemente sus principios, y conocer si sus bases han sufrido transformaciones para tener certeza que se trabaja sobre elementos reales (Ponjuán Dante, 2004).

Para lograr una efectiva gestión de la información es necesario conocer las fuentes, servicios y sistemas, su ciclo de vida y sus criterios de calidad, así como definir bien las políticas entorno al papel de cada cual en el manejo de información.

Toma de decisiones

La toma de decisiones es un proceso mediante el cual una persona debe elegir una opción entre dos o más alternativas para la solución de un problema. Es la esencia de las acciones administrativas en cualquier organización.

La toma de decisiones en una organización se divide en tres niveles (Davis, 2000):

1. **Estratégico:** son las decisiones que influyen en la dirección general de la organización.
2. **Táctico:** son decisiones orientadas a las operaciones, ya que resuelven los problemas en un menor rango tiempo. Cuentan con una capacidad de programación limitada debido a que no poseen una estructura sólida en la situación del problema.
3. **Operativo:** son aquellas decisiones de naturaleza rutinaria, relacionadas con el control de las tareas específicas.

El proceso de tomar decisiones se divide en una serie de pasos que conducen a la elección final (Davis, 2000):

- Identificación del problema: se debe reconocer la existencia del problema y buscar alternativas al mismo.
- Búsqueda de información: se debe buscar toda la información relacionada con la situación problemática identificada.
- Análisis del problema: se determinan las causas del problema y sus consecuencias.
- Evaluación de opciones: se identifican todas las posibilidades para resolver el problema.
- Decisión: se escoge la opción más conveniente y adecuada para la resolución del problema.

Para tomar decisiones correctamente es necesario reunir toda la información relevante entorno al ambiente en el que se vaya a decidir. Esta información debe tener la calidad suficiente para resolver los problemas de la organización.

Reporte

El propósito de este trabajo es desarrollar una herramienta que permita llevar un seguimiento y control de todos los proyectos productivos de la Universidad. Para ello se implementará un sistema que generará y almacenará todos los Reportes de Estado de los proyectos y los Reportes Postmortem, con el objetivo de que los responsables puedan evaluar el desarrollo de los proyectos.

Es importante que se defina qué es un Reporte, qué es un Reporte de Estado y qué es un Reporte Postmortem para un mejor entendimiento del tema.

Un Reporte es un informe que se formula o se presenta por una persona, entidad o sistema. Crear reportes es un proceso crítico para tareas y proyectos de investigación.

Siempre que se piense en elaborar reportes, se deben tener en cuenta algunos aspectos importantes:

1) Definición del reporte: el autor del reporte precisa los datos y la manera de presentación de los mismos. Se definen las conexiones a los diferentes orígenes de datos para obtener los resultados que debe reflejar el reporte.

2) Administración del reporte: puesto que en las organizaciones actuales existen varias categorías de usuarios, como los gerentes, los usuarios de servicio al cliente, etc.; es importante que se definan los usuarios del reporte, para ello es necesario publicar los reportes.

3) Entrega del reporte: normalmente en las organizaciones los reportes se entregan de manera periódica. Para que los reportes lleguen a los usuarios requeridos se pueden aprovechar varios servicios como el de mensajería.

Estas tres acciones conforman lo que se denomina "El Ciclo de Vida de un reporte" (HIDALGO, 2001).

Reporte de Estado

Un Reporte de Estado es un documento que informa el estado actual de un proyecto. Su principal propósito es comunicar al receptor si el proyecto se va desarrollando según lo planeado y por qué, o si no se va desarrollando según lo planeado, también por qué. Este reporte no es producido para registrar qué trabajo hizo o hará el equipo del proyecto, sino que su intención es describir los desvíos del plan y cómo serán corregidos. En dicho informe deben constar, al menos, un resumen que describa si el proyecto se está desarrollando según lo planeado, si está cumpliendo con las fechas estimadas de los hitos y fechas de entrega de los entregables, si surgieron riesgos nuevos, o aumentó la probabilidad o el impacto de riesgos conocidos. También debe tener una breve descripción de aquellas partes del proyecto que no se desarrollan según lo planeado y qué se está haciendo para corregir este problema, detallar los hitos principales alcanzados y por alcanzar en el corto plazo, el porcentaje de avance en los entregables más importantes y el costo actual del proyecto (Esterkin, 2008).

Reporte Postmortem

Un Reporte Postmortem se realiza cuando concluye un proyecto o cuando el líder de un proyecto culmina su labor en el mismo.

¿Por qué se hace necesario un reporte Postmortem? (HUMPHREY, 1999)

- Mejoramiento continuo de un proyecto: un nuevo proyecto tiene la posibilidad de mejorar aprendiendo de las experiencias anteriores.
- A partir de las oportunidades de mejoramiento se puede definir cómo variar las prácticas en función de un nuevo ciclo o un nuevo proyecto.
- Permite evaluar el producto producido, el esfuerzo empleado y el proceso que se siguió para hacerlo.

- Facilita el conocimiento de los problemas y las causas acontecidas hasta el momento y a partir de ahí puedes identificar las medidas de prevención.
- La clave para el mejoramiento exitoso son los pequeños cambios, los cuales son fáciles de olvidar.
- Es necesario anotar las ideas de mejoramiento y poderlas ordenar en el postmortem.

En el Reporte Postmortem deben describirse todos los datos de un proyecto durante su desarrollo especificados por área, o sea, todo lo referente a los objetivos, la calidad, el cronograma, la gestión del proyecto, etc. De esta forma, se puede comparar lo planeado con lo real, mirar los tiempos gastados en cada una de las fases indicando el porcentaje respecto al total, evaluar las metas de lanzamiento y mirar si se cumplieron o no; así como las metas de cada uno de los roles y si fueron cumplidas o no, describir todo el proceso de mejora, qué problemas hubo y qué solución se le puede dar. En un Reporte Postmortem deben quedar plasmados todos los detalles de un proyecto desde su inicio hasta su culminación.

Estado del arte de los Reportes de Estado

En el mundo

El Instituto Americano de Administración de Proyectos (IAAP) es una organización dedicada a la Administración de Proyectos en Latinoamérica, cuya misión es ayudar a las organizaciones y a los profesionales a implementar mejores proyectos. Está formado por: Argentina, Estados Unidos, México y Chile. Esta organización establece que un Reporte de Estado debe ser un documento corto, de no más de 2 páginas, aunque el proyecto sea grande (Esterkin, 2008). El propósito principal de este reporte es comunicar al receptor si el proyecto se va desarrollando según lo planeado o no y por qué. Este reporte no es producido para registrar qué trabajo hizo o hará el equipo del proyecto, sino que su foco es describir los desvíos del plan y cómo serán corregidos. En este reporte se deben reflejar, por lo menos, los siguientes capítulos:

- **Resumen ejecutivo:** un resumen en el que se detalle si el proyecto está marchando según lo planeado, si cumple con las fechas estimadas de los hitos y fechas de entrega de los entregables, que indique además si surgieron riesgos nuevos, o si aumentó la probabilidad o el impacto de

riesgos conocidos. Debe indicar también qué acciones a corto plazo se le piden al cliente externo y/o interno, o patrocinadores, para que el proyecto sea exitoso.

- **Detalle de avances y desvíos:** describir brevemente las partes del proyecto que no se desarrollan según lo planeado y qué se está haciendo para corregir este problema. Detallar además los principales hitos alcanzados y por alcanzar en el corto plazo. En esta sección se debe comentar qué se hará en el próximo mes para regresar el proyecto a su cauce normal.
- **Registro de riesgos:** se indica si surgieron riesgos nuevos o aumentó la probabilidad o el impacto de riesgos conocidos.
- **Métricas:** se reporta el progreso de las métricas elegidas para el proyecto, tales como: uso de recursos en horas, por día, días hábiles trabajados, días hábiles para finalizar, porcentaje de avance en los entregables más importantes, porcentaje del presupuesto total gastado, costo hasta hoy, etc (Esterkin, 2008).

Por otro lado, Microsoft propone una plantilla de Reporte de Estado de Proyecto en la que deben constar algunos puntos como:

- **Control del Reporte de Estado:** versión, fecha, autor y descripción de cambio.
- **Propósito del Reporte de Estado:** resumen del objetivo del reporte referente a las necesidades específicas del proyecto.
- **Resumen Ejecutivo del Reporte de Estado:** debe incluir detalles del estado del proyecto para sus principales jefes, centrandose en la entrega de los hitos.
- **Estado del proyecto:** breve declaración de la situación del proyecto.
- **Resumen del proyecto:** breve descripción de los resultados del proyecto no incluidos en el resto del informe.
- **Hitos a entregar para el último período reportado:** entregables, fecha prevista, % completado y estado de la entrega.
- **Hitos a entregar en el período actual del proyecto:** entregables, fecha prevista, % completado y estado de la entrega.
- **Afectaciones sobre el proyecto:** Cambios sobre hitos o alcance del proyecto.
- **Informe de presupuesto:** presupuesto planeado, costo actual y varianza.

- **Informe de Administración de riesgos:** especifica cualquier cambio en los riesgos más importantes identificados desde el informe anterior y cualquier modificación en las estrategias para abordarlos, si lo considera necesario.
- **Estado de las tareas del proyecto:** tarea y descripción, impacto en el proyecto, fecha prevista, estado de la tarea y solución de la tarea.

El Project Management Institute (PMI) Capítulo México a través de su sitio LiderDeProyecto.com propone cómo se debería hacer un Reporte de Estado de Proyecto. Este sitio está dedicado a intercambiar, difundir y promover experiencias, técnicas y prácticas en materia de manejo de proyectos, y es uno de los principales sitios de difusión de contenidos de buenas prácticas en administración de proyectos en los países de habla hispana. En el mismo se especifica que en el Reporte de Estado se deben tener en cuenta los parámetros que hacen de un proyecto algo exitoso (si se cumplen) o que lo convierten en un fracaso (si no se cumplen) y usarlos como los elementos o secciones a reportar. Este reporte puede ser un documento formal y entre sus secciones debe incluir al menos (OROZCO, 2009):

- **Progreso.** ¿Se terminó en tiempo lo que se tenía que terminar? ¿cuánto falta? ¿cuánto se retrasó? ¿en qué se retrasó? ¿por qué?
- **Alcance.** ¿Se identificó lo que se requiere hacer? ¿ha pedido cambios el cliente? ¿se negociaron esos cambios? ¿el cliente va a pagar los cambios? ¿cuánto ha cambiado el alcance original?
- **Tiempos.** ¿Cuál es el retraso del proyecto en tiempo y porcentaje? ¿se han cumplido los hitos del proyecto? ¿cuál es el retraso del proyecto?
- **Costos.** ¿Cuánto se ha gastado hasta el momento? ¿cuánto se debería de haber gastado? ¿cuánto se va a gastar? ¿va a alcanzar?
- **Rentabilidad.** ¿se está ganando lo que se quería? ¿estamos perdiendo? ¿cuánto se está ganando o perdiendo? ¿cuánto se va a ganar o perder si seguimos así?
- **Riesgos.** ¿Cuáles son los principales riesgos? ¿qué se va a hacer para eliminarlos?
- **Problemas.** ¿Cuáles son las situaciones problemáticas actuales? ¿Qué se está haciendo para resolverlas?
- **Calidad.** ¿Se está haciendo el producto como el cliente quiere y necesita? ¿cuántos defectos tiene? ¿cuáles son los principales defectos? ¿se están cumpliendo los estándares? ¿el cliente está quedando satisfecho con los resultados?

- **Recursos Humanos.** ¿se cuenta en el proyecto con el personal adecuado? ¿se tienen suficientes recursos? ¿hay recursos problemáticos?
- **Recursos Materiales.** ¿Tenemos lo necesario para trabajar?

En la UCI

La Universidad de las Ciencias Informáticas (UCI) cuenta con una amplia gama de proyectos dedicados a la producción de software, tanto en el ámbito nacional como internacional. Existen varios procedimientos a la hora de reportar el estado actual de estos proyectos.

Para conocer la situación actual en la que se encuentra la Universidad referente a este tema, los autores de esta investigación realizaron encuestas a las facultades, donde se demuestra que no existe una plantilla o un sistema generalizado para reportar el estado de los proyectos para todas las facultades.

El modelo de encuesta se muestra en el [Anexo 3](#). Entre los indicadores que se midieron en la encuesta se tuvieron en cuenta los siguientes:

- ¿Se reporta el estado del proyecto?
- ¿Cada qué tiempo realiza este reporte?
- Medio en que reporta el estado del proyecto.
- ¿Ha elaborado un Reporte Postmortem?
- Medio en que ha realizado el Reporte Postmortem.

Según los resultados de estas encuestas, en un 70% de las facultades los responsables se reúnen periódicamente en el Consejo de Producción donde analizan la situación de cada proyecto según el cronograma de trabajo, de forma tal que puedan saber si el proyecto está atrasado o si ha sufrido alguna afectación.

El resto de las facultades utiliza la plantilla definida por ALBET. Esta plantilla fue creada para los proyectos de ámbito internacional. Se considera que es una plantilla muy general para efectuar un

adecuado monitoreo y control en los proyectos, pues no especifica todos los detalles que deben quedar plasmados en un Reporte de Estado de un proyecto. La misma se muestra en el [Anexo 4](#).

De los proyectos encuestados, ningún Jefe de proyecto ha realizado un Reporte Postmortem, lo que trae consigo la necesidad de crear un plantilla que responda a esta dificultad.

El siguiente gráfico representa el resultado de las encuestas.

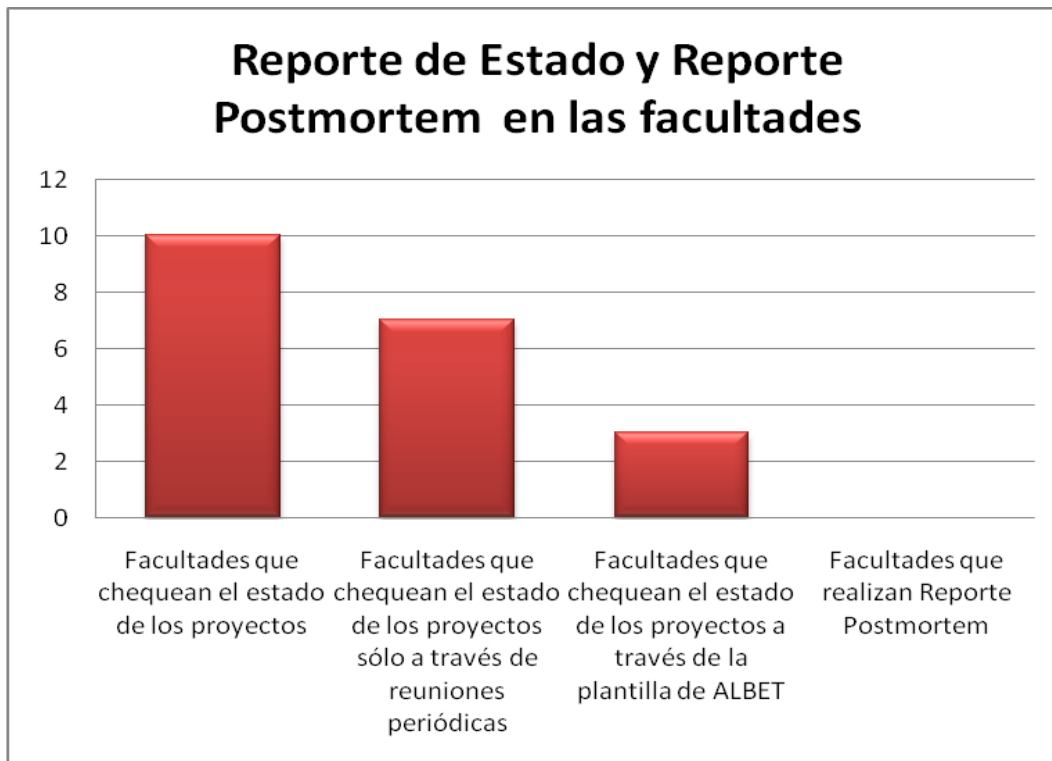


Figura 1. Resultados de las encuestas realizadas en las facultades.

Estas encuestas demuestran la necesidad de crear un medio para reportar el estado de los proyectos, el sistema propuesto por el presente trabajo constituye un componente de gestión necesario para centralizar toda la información referida al estado actual de los proyectos productivos de la Universidad.

Proceso de desarrollo de software

Un proceso define quién está haciendo qué, cuándo y cómo lograr un determinado objetivo. Un proceso de desarrollo de software es el procedimiento de establecer las actividades relacionadas con la creación, presentación y mantenimiento de los sistemas de software (LARMAN, 1999). El objetivo del proceso de desarrollo de software es convertir los requisitos de un usuario en un sistema de software.

En la ingeniería del software el propósito es construir un producto de software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad, y captura y presenta las mejores prácticas que permite el estado actual de la tecnología. Asimismo, reduce el riesgo y hace que el proyecto sea más predecible.

Un proceso de desarrollo de software debe ser flexible y configurable para cumplir con las necesidades reales de un proyecto y/o organización concreta (JACOBSON, y otros, 2000).

Tecnologías y herramientas utilizadas

Aplicaciones Web

Con la evolución de Internet, han surgido infinitas posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto constituye un gran reto para los desarrolladores de aplicaciones, pues con los avances de la tecnología se requieren aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web.

Una aplicación Web se crea con el objetivo de dar solución a diversas necesidades o problemas de los usuarios, y permite a cualquier persona consultarla e interactuar con ella a través de la red. El desarrollo de una aplicación Web permite al usuario publicar cualquier tipo de información que se pueda incorporar a una base de datos. Una vez que el contenido de la página cambia, evita al diseñador Web tener que actualizar continuamente el código HTML del sitio. Permitirá además a los usuarios localizar información de forma rápida y sencilla en un sitio Web en el que se almacena gran cantidad de contenido, así como guardar datos de formularios directamente en una base de datos, extraer datos y crear informes basados en la Web para su análisis (ADOBE, 2008).

Las aplicaciones web no requieren de grandes conocimientos en informática para su uso. Por todo lo anteriormente planteado, los autores de este trabajo concluyeron que la solución propuesta consistiera en una aplicación web.

Lenguajes de Programación Web

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, éstos han ido surgiendo debido a las tendencias y necesidades de las plataformas. A continuación se presentan los lenguajes de programación que se utilizan para la construcción del sistema.

Lenguajes del lado cliente

Son aquellos lenguajes que son asimilados directamente por el navegador y no necesitan pre tratamiento.

HTML:

La información publicada en la Web debe ser mediante un lenguaje que sea entendido universalmente. El lenguaje utilizado por la World Wide Web es el HTML, actualmente el lenguaje hipertexto más aceptado a nivel mundial. Por tanto, es uno de los lenguajes de programación web más importante y uno de los más usados para la creación de documentos.

El HyperText Markup Language (HTML) es un lenguaje de marcado que se diseñó con el objetivo de estructurar documentos y mostrarlos en forma de hipertexto. El mismo brinda la información correspondiente relacionada con el contenido a mostrar en cada fragmento de información, y permite establecer relaciones unidireccionales entre documentos. Los archivos pueden tener las extensiones (htm, html). Este lenguaje cumple con dos objetivos fundamentales para el diseño y visualización de un documento digital (FRESNO, 2006):

- Organiza un documento en elementos lógicos, tales como: encabezado, párrafo, etc.
- Define las operaciones tipográficas y las funciones que debe ejecutar un programa visualizador sobre dichos elementos.

Permite además a los autores (W3C, 1997):

- La publicación de documentos en línea con títulos, textos, tablas, listas, fotos, etc.
- La recuperación de la información en línea mediante enlaces de hipertexto.

- Diseñar formularios para la realización de transacciones con servicios remotos, para su uso en la búsqueda de información, hacer reservas, pedir productos, etc.
- Insertar hojas de cálculo, videoclips, clips de sonido, y otras aplicaciones directamente en sus documentos.

Este lenguaje es interpretado por los navegadores de acuerdo a su criterio, esto permite que una misma página web se pueda mostrar de diferentes formas según el navegador.

JAVASCRIPT:

JavaScript es un lenguaje de programación orientado a objetos. Es un lenguaje dinámico, las variables no necesitan ser introducidas antes de su uso y los tipos de variables se resuelven dinámicamente durante su ejecución. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

Fue creado por Brendan Eich en la empresa Netscape Communications. El código JavaScript que se encuentra dentro de las páginas web puede ser interpretado por todos los navegadores. Permite que las definiciones de funciones y otro tipo de código sean modificados mientras el programa se esté ejecutando. El modelo de ejecución de JavaScript se basa en la interpretación del código fuente. Es un lenguaje de alto nivel, multiplataforma y no necesita compilación. Está basado en objetos, admite la programación estructurada y maneja la mayoría de los eventos que se pueden producir sobre la página web (MIKKONEN, y otros, 2007).

La mayoría de los navegadores en sus últimas versiones interpretan el código Javascript integrado dentro de las páginas web.

Hojas de Estilo en Cascada:

Las Hojas de estilo en Cascada, en inglés Cascading Style Sheets (CSS), fueron diseñadas y desarrolladas por la World Wide Web Consortium (W3C).

Una hoja de estilos CSS es el tipo de documento que utiliza un navegador Web para redefinir las propiedades de los distintos elementos y las etiquetas en el código HTML. Permite dar formato a los documentos de forma global. Proporciona al diseñador de páginas web definir un conjunto de ampliaciones HTML especiales y aplicarlas al documento. Provee la especificación e intercambio de los

fondos para textos y documentos, así como sus tipos y tamaños de fuente. Las definiciones del formato de un documento se pueden colocar en archivos separados y aplicarlas a un grupo de documentos. Posibilitan además aplicar un formato modificado a documentos HTML ya existentes. Con los CSS se puede aplicar a un documento diferentes estilos de orígenes. Los CSS constituyen una herramienta poderosa para el diseño de documentos HTML, pues permiten modificar la representación del documento mediante la asignación de un nuevo estilo (MEYER, 2001).

Lenguajes del lado servidor

Son aquellos lenguajes que se ejecutan por el propio servidor y son enviados al cliente en un formato claro para él.

PHP 5

En la implementación de la aplicación propuesta se utiliza el lenguaje de programación PHP, por ser un lenguaje que está implementado especialmente para el desarrollo web.

PHP: Hypertext Preprocessor (PHP) es un lenguaje de programación web de alto nivel que se ejecuta en el servidor.

PHP 5 se lanzó oficialmente en septiembre del 2004, utilizando el motor Zend Engine 2.0 (o Zend Engine 2). El objetivo principal de PHP 5 ha sido perfeccionar los mecanismos de la Programación Orientada a Objetos para dar solución a las carencias de las versiones anteriores (ÁLVAREZ, 2004).

En el desarrollo de este trabajo se escogió esta versión, pues entre sus mejoras, incluye modificadores de control de acceso para implementar el encapsulamiento y el manejo de excepciones, lo cual no existía en versiones anteriores. En PHP 5, no es necesario pasar objetos por referencia. Incluye numerosas funcionalidades explícitas como constructores y destructores, objeto de clonación, la clase de abstracción, herencia, no requiere definición de tipos de variables ni manejo detallado del bajo nivel. Se pueden aplicar tantas interfaces como quiera y brinda un mejor soporte para la programación orientada a objetos y la lectura de archivos XML. Es un lenguaje que se caracteriza por ser Open-Source y de obtención gratuita, lo que lo convierte en una alternativa de fácil acceso para todos. Su flexibilidad lo convierte en un lenguaje muy sencillo de aprender; principalmente para los desarrolladores que han trabajado con lenguajes como Perl, C o Java, ya que posee similitudes de sintaxis con los mismos. Es portable y multiplataforma (Linux,

Windows, entre otros). Trabaja sobre la mayoría de servidores web y está preparado para interactuar con la mayoría de Sistemas de Gestión de Bases de Datos (MySQL, PostgreSQL, Oracle, SQL Server, entre otros) (Gutmans, y otros, 2004).

Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol) y que se ejecuta continuamente en un ordenador. El servidor web se mantiene a la espera de las peticiones HTTP que se llevan a cabo por un cliente HTTP, conocido usualmente como navegador web. Una vez realizada la petición del navegador al servidor, éste le responde con el contenido solicitado por el cliente (USERO, 2007).

Entre los servidores web más importantes se pueden citar el **Internet Information Server (IIS)**, el **Servidor HTTP Cherokee** y el **Servidor HTTP Apache**.

Internet Information Server (IIS)

Internet Information Services/Server (IIS): son servicios para los ordenadores que funcionan con Windows. Inicialmente formó parte del Option Pack para Windows NT. Luego se integra en otros sistemas operativos de Microsoft destinados a brindar servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de Internet Information Server. Brinda servicios como FTP, SMTP, NNTP y HTTP/HTTPS. Además procesa páginas de ASP y ASP.NET; y puede incluir también PHP o Perl (MANSO, 2000). Una de las desventajas que presenta este servidor web es que solo se puede utilizar en sistemas de Windows.

Apache

El servidor web Apache fue desarrollado por el Apache Server Project (Proyecto Servidor Apache). Su función principal es analizar todos los archivos solicitados por un navegador y mostrar los resultados correctos de acuerdo con el código dentro de ese archivo. Es un servidor web muy potente y flexible, y puede realizar prácticamente cualquier tarea que el usuario requiera. Cuenta con una arquitectura modular que le permite personalizarse mejor para las necesidades de cada sitio web y facilita a los usuarios adicionar cómodamente funcionalidad a sus ambientes específicos. Es robusto y seguro. Es

multiplataforma, pues trabaja sobre todas las versiones recientes de UNIX y Linux, Windows, BeOs, mainframes. Facilita la visualización de código HTML. Una de las grandes ventajas que posee es su capacidad de autenticación, lo que proporciona el control de acceso de los usuarios y estaciones de trabajo a varios sitios Web; esto permite una regulación en la Internet en cuanto a los usuarios que están capacitados o no de conocer la información de una organización (APACHE, 2007).

Para el desarrollo de esta investigación se escogió el Servidor HTTP Apache, por ser un servidor web perteneciente al software libre, además es un servidor gratuito y Open Source, lo cual ya constituye uno de sus principales beneficios.

AJAX

Para el desarrollo de la aplicación propuesta se utilizará AJAX, ésta no es una tecnología en sí, sino la unión de varias tecnologías existentes con el propósito de crear aplicaciones interactivas sin la necesidad de cargar la página completamente entre una solicitud y otra.

El término AJAX es un acrónimo para “*Asynchronous JavaScript + XML*”, lo que se podría traducir como “*JavaScript asíncrono + XML*”. AJAX posibilita que se mejore totalmente la interacción del usuario con la aplicación. Las peticiones HTTP hechas al servidor son sustituidas por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no necesitan intervención del servidor, facilitando así una respuesta inmediata. En caso de que la interacción requiera de una respuesta del servidor, la petición se realiza asíncronamente mediante AJAX, provocando que la interacción del usuario tampoco se vea interrumpida por recargas de página o largas esperas por la respuesta del servidor. La reducción del tamaño de los envíos de paquetes entre el servidor y el cliente, facilita mayor rapidez e interactividad a medida que se manipula la interfaz del navegador con el lenguaje JavaScript. El desarrollo de AJAX es multiplataforma, pues funciona tanto con el motor Javascript del Internet Explorer, como con el motor de Mozilla Firefox y el motor de Safari. AJAX incorpora presentación basada en estándares usando XHTML y CSS, exhibición e interacción dinámicas usando el Document Object Model, intercambio y manipulación de datos usando XML, XSLT y JSON, intercambio asíncronico de datos usando XMLHttpRequest, y JavaScript para la unión de las demás tecnologías (PÉREZ, 2009).

Sistema Gestor de Bases de Datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación.

Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos (BERTINO, y otros, 1995).

MySQL

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras (COBO, y otros, 2005).

Microsoft SQL Server

SQL Server es un sistema gestor de base de datos relacionales producido por Microsoft. Es un sistema cliente/servidor que funciona como una extensión natural del sistema operativo Windows. Entre otras características proporciona integridad de datos, optimización de consultas, control de concurrencia y backup y recuperación. Es relativamente fácil de administrar a través de la utilización de un entorno gráfico para casi todas las tareas de sistema y administración de bases de datos. Utiliza servicios del

sistema operativo Windows para ofrecer nuevas capacidades o ampliar la base de datos, tales como enviar y recibir mensajes y gestionar la seguridad de la conexión. Es fácil de usar y proporciona funciones de almacenamiento de datos que sólo estaban disponibles en Oracle y otros sistemas gestores de bases de datos más caros (PETKOVIĆ, 2005).

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley.

Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación. Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales. El código fuente se encuentra disponible para todos sin costo alguno. Está disponible para 34 plataformas con la última versión estable (LOCKHART, 1996).

Es totalmente compatible con ACID (acrónimo de Atomicity, Consistency, Isolation and Durability; en español: Atomicidad, Consistencia, Aislamiento y Durabilidad). Posee una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico (POSTGRESQL, 2006).

Como sistema de gestión de base de datos a utilizar se seleccionó el PostgreSQL por ser un potente gestor de Bases de Datos, multiplataforma, software libre y corre en un gran número de sistemas operativos. Además posee todas las características de los SGBD modernos, como son las llaves foráneas,

vistas, procedimientos almacenados, varios tipos modernos de datos, etc.; igualmente posee una característica importante, que es la realización de múltiples backups o salvadas de la información.

Framework

El objetivo principal de un framework es facilitar el proceso de desarrollo de una aplicación, a través de la automatización de algunos de los patrones utilizados para solucionar las tareas comunes. Debe permitirle al desarrollador implementar aplicaciones de forma rápida y sencilla.

Proporciona una estructura al código fuente, obligando al desarrollador a crear código más comprensible y más fácil de mantener. Facilita, además, la programación de aplicaciones, debido a que encapsula operaciones complejas en instrucciones sencillas.

Los frameworks son valiosos en todas las etapas de desarrollo, desde el diseño hasta la implementación y más allá. Por lo general se aplican a casi todas las etapas del ciclo de vida de una aplicación.

Symfony

Para la implementación del sistema se utilizará el framework Symfony, debido a que proporciona un enorme conjunto de herramientas que simplifican el desarrollo y la construcción de las aplicaciones web.

Symfony es uno de los frameworks PHP más utilizados entre los usuarios y las empresas, implementado para perfeccionar el desarrollo y mantenimiento de aplicaciones Web construidas en este lenguaje y suplantar las tareas de codificación reiterativas por poder, control y placer. Ofrece varias ventajas importantes que facilitan el trabajo del programador:

- Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.
- Facilita varias herramientas y clases que reducen el tiempo de desarrollo de una aplicación web compleja.
- Automatiza las tareas más comunes, lo que permite que el desarrollador se dedique por completo a los aspectos específicos de cada aplicación.

Es un framework desarrollado completamente con PHP 5 que se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de los sistemas gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se ejecuta tanto en plataformas Windows

como en plataformas *nix (Unix, Linux, etc.) y es fácil de instalar y configurar en las mismas. En la mayoría de los casos es fácil de usar y lo suficientemente adaptable como para aplicarse a los casos más complejos. Se basa en la premisa de "convenir en vez de configurar", donde el desarrollador sólo debe configurar aquello que no es convencional. Está preparado para aplicaciones empresariales y es configurable a las políticas y arquitecturas de cada empresa. No depende del sistema gestor de bases de datos, es fácil de extender y sigue la mayoría de mejores prácticas y patrones de diseño para la web (ZANINOTTO, y otros, 2009).

Metodología de desarrollo

Extreme Programing (XP)

Extreme Programing (XP) es una metodología de desarrollo de software que consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Esta metodología se basa en (SÁNCHEZ, 2004):

- Pruebas Unitarias: son las pruebas que se realizan a los principales procesos, de forma tal que adelantándose en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran suceder. Es como si se adelantara la obtención de los posibles errores.
- Refabricación: es la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

RUP

La metodología RUP, llamada así por sus siglas en inglés: Rational Unified Process, es una metodología que divide en 4 fases el desarrollo del software: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Presenta tres características fundamentales que

constituyen la esencia de todo el proceso de desarrollo: Dirigido por los casos de uso, Centrado en la arquitectura e Iterativo e Incremental.

RUP divide el trabajo en nueve Flujos de Trabajo (JACOBSON, y otros, 2000), los seis primeros de Ingeniería y los tres últimos de Apoyo, cada uno de ellos empleados con diferente énfasis en cada una de las 4 fases del desarrollo de software:

- Modelado del negocio
- Análisis de requisitos
- Análisis y diseño
- Implementación
- Pruebas
- Despliegue
- Administración de configuración y cambios
- Administración del proyecto
- Administración del entorno

RUP posee un grupo de características que hacen más dinámico el desarrollo del trabajo y constituye una propuesta de proceso para el desarrollo de software orientado a objetos.

Debido a que en el desarrollo de esta tesis se realizará el análisis y el diseño de un sistema para la evaluación y Reporte de Estado de proyectos, se selecciona RUP en lugar de XP, puesto que esta última es un proceso muy orientado a la implementación, lo que convierte a RUP en una metodología más adecuada para un análisis y diseño exitoso. Se seleccionó esta metodología además por las ventajas que presenta en la organización de la información. Igualmente, posee un grupo de características y facilidades que hacen más dinámico el desarrollo del trabajo.

Lenguaje de modelado UML

Unified Modeling Language (UML) es el Lenguaje Unificado de Construcción de Modelos. Es una notación (esquemática en su mayor parte) para la construcción de sistemas de modelado que utilizan conceptos orientados a objetos. Este lenguaje permite especificar, visualizar y construir los artefactos de los sistemas de software. Es un lenguaje flexible de modelado que permite la definición de modelos arbitrarios. Estandariza los artefactos y la notación, pero no define un proceso oficial de desarrollo. No

define un formato rígido; se puede transformar para atender las necesidades y ajustarse al espíritu de la documentación: ante todo, una comunicación clara. Este lenguaje permite explicar de forma concreta las perspectivas de notación y terminología muy afines; de ahí la importancia de tener presente qué perspectiva se va a adoptar (un análisis, un diseño o una vista de la implementación) (LARMAN, 1999).

Ext JS

Ext JS fue desarrollado por un grupo de programadores con el objetivo de proporcionar una interfaz de usuario principal y una librería de interacción. Ext es fácil de utilizar, proporciona una rica interfaz de usuario, al igual que se encontraría en una aplicación de escritorio. Esto permite a los desarrolladores de la web concentrarse en la funcionalidad de aplicaciones web en lugar de las reservas técnicas.

Ext JS permite la interacción con el usuario y el navegador a través de la EventManager, en respuesta a las pulsaciones de teclado los usuarios, clics del ratón y el seguimiento de eventos en un navegador, tales como cambiar el tamaño de una ventana o el tamaño de la letra. Proporciona además la comunicación con el servidor en segundo plano y sin la necesidad de actualizar la página. Esto permite al usuario solicitar o publicar los datos hacia o desde su servidor web utilizando AJAX y procesar la información en tiempo real (FREDERICK, y otros, 2008).

Herramientas utilizadas en la solución del sistema propuesto

Se usará Visual Paradigm como herramienta de modelado. Esta herramienta ayuda a construir aplicaciones de forma rápida, mejor y más barata. Está diseñada para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas, analistas de negocios y los arquitectos del sistema, quienes están interesados en la construcción de sistemas de software fiable para el uso de un enfoque orientado a objetos (PARADIGM, 2007).

Como Entorno de desarrollo integrado (IDE, por sus siglas en inglés) se utilizará Eclipse. Eclipse es una comunidad de código abierto, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles y herramientas para crear, desplegar y gestionar software en todo el ciclo de vida. Permite centrarse en el uso de la tecnología de código abierto en

software comercial de productos y servicios. Todos los proyectos de Eclipse están registrados bajo la licencia Eclipse Public License (EPL) (FOUNDATION, 2004).

Patrones de diseño y Arquitectura

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software. Los patrones de arquitectura expresan un esquema organizativo estructural fundamental para sistemas software.

En el desarrollo del presente trabajo se exponen algunos patrones de diseño y arquitectura utilizados. En el capítulo de análisis y diseño se explica cómo se aplican estos patrones en Symfony.

Patrones de diseño GRASP

Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades).

A continuación se describen los patrones básicos de asignación de responsabilidades utilizados en este trabajo:

- **Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Este patrón es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.
- **Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta.
- **Bajo Acoplamiento:** Este patrón es el encargado de asignar una responsabilidad para conservar bajo acoplamiento.
- **Controlador:** Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (LARMAN, 1999).

Patrones de diseño GoF

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides) se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento (GAMMA, y otros, 1995).

- **Creacionales:** Los patrones creacionales abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** Los patrones estructurales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionar nuevas funcionalidades.
- **Comportamiento:** Los patrones de comportamiento están relacionadas con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos.

Patrones GoF que implementa Symfony

Para la implementación del sistema se utiliza el framework Symfony, el cual utiliza una serie de patrones GoF:

En la categoría de creacionales utiliza (GAMMA, y otros, 1995):

Singleton (Instancia única): Garantiza que una clase tenga una única instancia y proporciona un mecanismo de acceso global a dicha instancia.

En la categoría de estructurales utiliza:

Decorator (Envoltorio): Se encarga de añadir dinámicamente funcionalidad a un objeto.

En la categoría de comportamiento utiliza:

Command: Encapsula una operación como un objeto, permitiendo parametrizar los clientes utilizando distintas solicitudes.

Patrón de arquitectura MVC

Como ya se ha enunciado anteriormente, en el desarrollo de la aplicación se utilizará el framework Symfony, el cual está basado en un patrón clásico del diseño web conocido como arquitectura MVC,

conocido por sus siglas en inglés Model View Controller, que significa Modelo Vista Controlador. El mismo permite realizar la programación multicapa, separando en tres componentes distintos los datos de una aplicación, la interfaz del usuario y la lógica de control. Este patrón se ve usualmente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y el controlador representa la lógica del negocio, que está formado por tres niveles:

- Modelo: representa la información con la que trabaja la aplicación, o sea, su lógica de negocio.
- Vista: convierte el modelo en una página web que facilita al usuario interactuar con ella.
- Controlador: es el encargado de procesar las interacciones del usuario y ejecuta los cambios adecuados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista), lo que permite un mantenimiento más sencillo de las aplicaciones. El controlador es el encargado de aislar al modelo y a la vista de los detalles del protocolo usado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica referida a los datos, lo que permite que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos que la aplicación utiliza (ZANINOTTO, y otros, 2009).

El sistema que se implementará en el presente trabajo tendrá un estilo arquitectónico Modelo Vista Controlador, como se muestra en la siguiente figura:

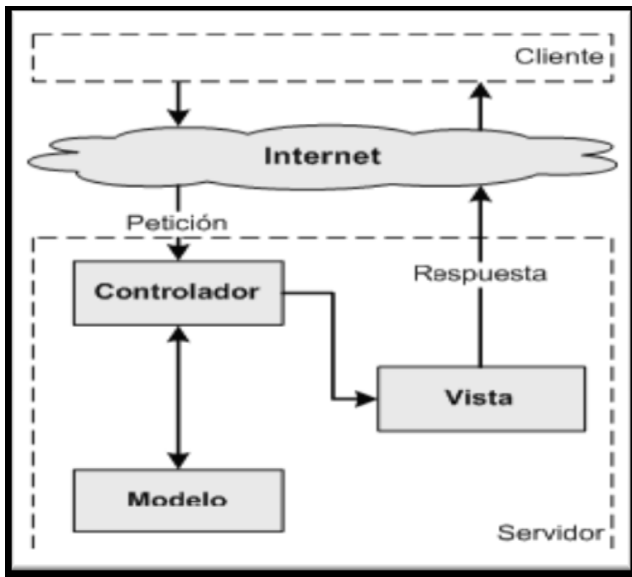


Figura 2 Modelo Vista Controlador

Conclusiones del Capítulo 1.

El seguimiento y control de las actividades de un proyecto es un proceso fundamental para lograr el éxito en el mismo. Como parte de este seguimiento y control se realizó un estudio del estado del arte de los Reportes de Estado y Reporte Postmortem en el mundo y en la UCI, y se llegó a la conclusión que ambos reportes son necesarios para el correcto desarrollo de los proyectos. El análisis que se realiza a partir de este estudio permite concluir que un Reporte de Estado es un proceso que debe efectuarse periódicamente en un proyecto, y en el que se deben incluir aquellos factores que hacen que el proyecto sea un éxito o un fracaso. Por su parte, el Reporte Postmortem permitirá que se trabaje en nuevos proyectos sobre las experiencias anteriores, con el objetivo de mejorar el desarrollo del proyecto y erradicar las dificultades.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se hace una propuesta del sistema que se desea implementar. Se realiza una breve representación del análisis del negocio que se plantea automatizar, obteniendo el modelo del dominio. Se muestran los requerimientos funcionales y no funcionales, se definen los actores que intervienen en el sistema, se hace una descripción de los casos de uso del sistema y se muestran los diagramas correspondientes.

Modelo del dominio

El negocio estudiado tiene muy bajo nivel de estructuración, se puede llegar a esta conclusión después de haber estudiado todos los procesos que se van a efectuar, donde los flujos de información se encuentran difusos, y cuando se desea realizar una actividad, múltiples personas intervienen en la misma, lo que implica un solapamiento de responsabilidades, además es difícil establecer las reglas de funcionamiento, por lo que se propone realizar un modelo de dominio.

Se realizará el modelo de dominio, porque permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio de los procesos actuales del negocio. Esto ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se enmarca el negocio. Es necesario saber cómo debe funcionar el proceso en cuestión para capturar correctamente los requisitos y así poder construir un sistema con las características que el cliente desee. Como primera tarea hay que identificar todos los conceptos que se utilizarán en el diagrama, mediante un glosario de términos sobre los nombres:

Decano: Máximo directivo en una Facultad.

VD. Producción: Representante del decano en las esferas de investigación y producción. Responsable máximo a nivel de Facultad de la gestión de recursos humanos en la producción, asume los compromisos en esas esferas.

Jefe de Polo: Jefe de un área especializada de desarrollo. Un polo está conformado por varios proyectos de los cuales él es el máximo responsable y el supervisor de su desarrollo.

Jefe de Proyecto: El Jefe de Proyecto se destaca como la figura clave en la planificación, ejecución y control del proyecto, es el motor que ha de impulsar el avance del mismo mediante la toma de decisiones tendentes a la consecución de los objetivos.

Reporte de Estado: Un Reporte de Estado es un documento que informa el estado actual de un proyecto.

Reporte Postmortem: Un Reporte Postmortem se realiza cuando finaliza un proyecto o cuando el líder de un proyecto concluye su labor en el mismo.

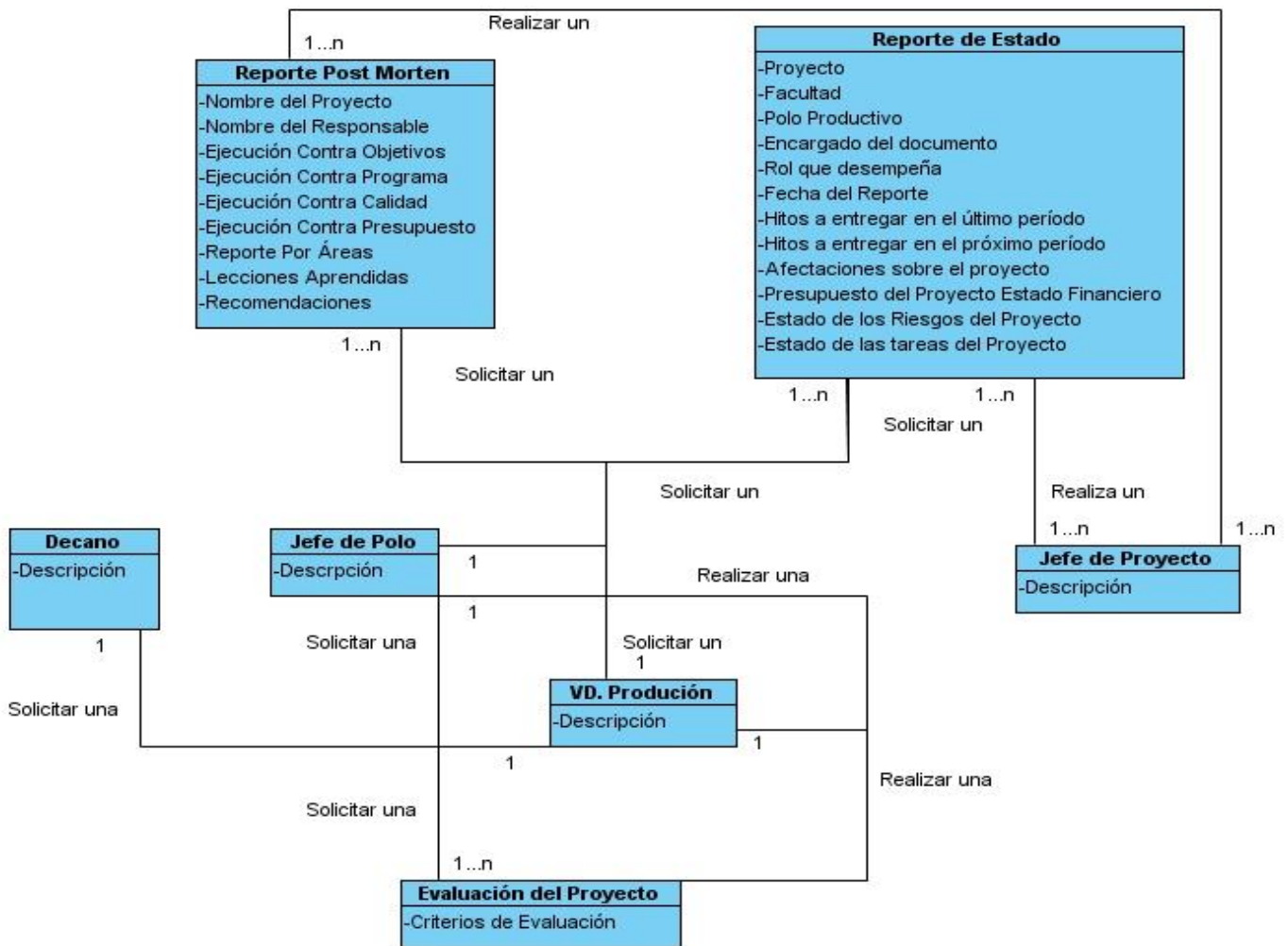


Figura 3 Modelo del dominio

Objeto de automatización

- **Reporte de Estado**

Proceso en el que el líder de proyecto llena los datos de la plantilla del Reporte de Estado de un proyecto.

- **Reporte Postmortem**

Es el proceso mediante el cual el líder de un proyecto llena la plantilla del Reporte Postmortem, una vez concluido el proyecto o el líder haya finalizado su labor en el mismo.

- **Evaluación de Proyectos**

El usuario selecciona los criterios de valoración para evaluar el desarrollo de un proyecto. Este proceso puede ser efectuado por cualquier usuario o rol definido por el administrador del sistema.

Propuesta del sistema

Luego de realizar estudios acerca de la situación problemática planteada en este trabajo, y hacer un profundo análisis del problema científico de la investigación y el campo de acción, se propone implementar un sistema que gestione toda la información referente al estado de los proyectos de la Universidad de las Ciencias Informáticas.

Se espera que a través de este sistema se pueda efectuar un seguimiento y control de los proyectos de la Universidad, y de esta forma conocer si los proyectos se están desarrollando según lo planeado o no, con el propósito de solucionar los problemas existentes y lograr mejores resultados en los mismos.

El sistema tendrá restricciones de seguridad que permitirán a los que trabajen en la aplicación guardar los datos de forma segura.

La información en los proyectos debe manejarse de forma sistemática. El sistema principalmente debe permitir la entrada de datos de todos los Jefes de proyectos, Jefes de polos, Vicedecano de producción y el Decano. Estos datos serán recogidos en formularios Web y los campos se validarán mediante Javascript. Los mismos serán guardados en una base de datos desarrollada en PostgreSQL, la cual tendrá la funcionalidad de archivar todos los datos de los reportes y las valoraciones. Se utilizará PHP 5

como lenguaje de programación para intermediar entre los formularios y la base de datos, para una mejor seguridad de los datos.

El sistema debe entrar en función desde que un usuario entre a la aplicación, ya sea para la entrada de datos de algún reporte o para el análisis de la información contenida en la aplicación, hasta que se retire de la misma.

Para entrar al sistema el usuario deberá autenticarse, una vez que se haya autenticado, sólo tendrá los permisos que se le hayan asignado. El encargado de crear los usuarios o roles y designar los permisos a los mismos es el administrador del sistema.

El administrador del sistema también será el encargado de otras funcionalidades para la entrada de datos como: gestionar área, gestionar polo, gestionar proyecto, gestionar rol, gestionar usuario y gestionar criterio de valoración.

Para llevar a cabo el proceso de la gestión de la información de los Reportes de Estado de proyectos, el sistema debe almacenar el estado de los proyectos durante su desarrollo mediante la emisión de reportes de los mismos, para ello se crearon funcionalidades que permitirán a los usuarios autorizados, en este caso los Jefes de proyectos, a formular los Reportes de Estado.

En la aplicación también deberán quedar plasmados los Reportes Postmortem, estos reportes serán formulados por el Jefe de proyecto, una vez que el proyecto haya finalizado o el Jefe de proyecto haya culminado su labor en el mismo.

El sistema permitirá a los Jefes de Polos, al Vicedecano de producción y al Decano valorar la situación de los proyectos según los Reportes de Estado de los mismos.

Debe proporcionar además representaciones de la información de los datos de proyectos, de forma tal que los usuarios puedan hacer comparaciones entre proyectos y períodos.

Con la información almacenada en el sistema, el Jefe de Polo, el Vicedecano de Producción y el Decano podrán evaluar los proyectos según su funcionamiento, para ello la aplicación brinda un conjunto de criterios de valoración con el objetivo de que los proyectos sean valorados por los roles designados.

El sistema debe ser capaz de generar un conjunto de informes según los soliciten los roles autorizados, y debe permitir a los Jefes de proyectos visualizar los reportes elaborados hasta el momento, así como las evaluaciones emitidas sobre el desempeño de sus proyectos.

El sistema debe permitir a los interesados conocer las afectaciones sobre los proyectos y brindará toda la información necesaria para la toma de decisiones en los mismos.

Especificación de los requerimientos de software

Los requerimientos describen las necesidades o deseos de un producto. Éstos deben representar de manera concisa qué es lo que el sistema debe hacer. En la captura de requisitos se debe centrar el esfuerzo en reconocer el problema tal y como lo ve el usuario. El objetivo principal de este flujo de trabajo es guiar el desarrollo para obtener el sistema correcto. Los requerimientos se clasifican en funcionales y no funcionales.

Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Expresan las operaciones que se deben implementar para satisfacer a los clientes. El sistema que aquí se expone, consta de 12 requerimientos.

1. Autenticar usuario.

RF 1.1. Autenticar usuario.

2. Gestionar Área.

RF 2.1. Adicionar área.

RF 2.2. Modificar área.

RF 2.3. Eliminar área.

3. Gestionar Polo.

RF 3.1. Adicionar Polo.

RF 3.2. Modificar Polo.

RF 3.3. Eliminar Polo.

4. Gestionar Proyecto.

RF 4.1. Adicionar proyecto.

RF 4.2. Modificar proyecto.

RF 4.3. Eliminar proyecto.

5. Gestionar Rol.

RF 5.1. Adicionar rol.

RF 5.2. Modificar rol.

RF 5.3. Eliminar rol.

6. Gestionar Usuario.

RF 6.1. Adicionar usuario.

RF 6.2. Modificar usuario.

RF 6.3. Eliminar usuario.

7. Gestionar Criterio de Valoración.

RF 7.1. Crear criterio de valoración.

RF 7.2. Modificar criterio de valoración.

RF 7.3. Eliminar criterio de valoración.

8. Registrar Reporte de Estado.

RF 8.1 Registrar Reporte de Estado.

9. Registrar Reporte Postmortem.

RF 9.1 Registrar Reporte de Postmortem.

10. Emitir Criterio de Valoración.

RF 10.1 Emitir Criterio de Valoración.

11. Visualizar Reporte.

RF 11.1 Visualizar Reporte.

12. Generar Informe.

RF 12.1. Generar Informe por período.

RF 12.2. Generar Informe por proyecto.

RF 12.3. Generar Informe por Polo.

RF 12.4. Generar Informe por proyecto y período.

RF 12.5. Generar Informe por Polo y período.

Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Se debe pensar en estas propiedades como las características que convierten al producto en usable, rápido o seguro.

➤ Apariencia o interfaz externa.

- El sistema debe tener una interfaz agradable y cómoda para los usuarios que trabajarán en el mismo.
- Se debe tratar de no sobrecargar las páginas con textos, imágenes o gráficos que dificulten la visibilidad de los componentes de las páginas.
- Utilizar colores acordes, ofrecer suficiente contraste entre texto y fondo para no dificultar la lectura y garantizar que los iconos estén en correspondencia con lo que representan.
- Mantener una coherencia y estilo común entre todas las páginas.

➤ Usabilidad.

- El sistema debe ser lo más atractivo posible y facilitar el trabajo a los usuarios.
- Se debe acceder a él de forma rápida y fácil, los usuarios no necesitan poseer grandes conocimientos de informática para trabajar en él.

- Brindar indicaciones en la interfaz que aceleren el aprendizaje del usuario cuando trabajen con el software.
- Rendimiento.
 - El sistema debe tener un soporte de hardware que garantice un ágil procesamiento de los datos con el objetivo de dar una respuesta rápida al usuario.
- Portabilidad.
 - El sistema debe ser multiplataforma.
- Seguridad.
 - Todo usuario deberá identificarse antes de poder ejecutar cualquier acción sobre el sistema.
 - Se crearán diferentes cuentas de usuario y se les asignará los permisos según el rol que tengan en el sistema.
 - Sólo se mostrará a cada usuario las funcionalidades del sistema sobre las cuales tiene permiso de acceso.
- Políticos – Culturales
 - El sistema estará disponible en el idioma español.
 - Los logotipos e imágenes usadas se corresponden con el tipo de institución donde será usado.
- Legales.
 - El sistema se implementará con el lenguaje de programación PHP perteneciente a la licencia de software libre GNU/GPL.
- Confiabilidad.
 - El sistema debe estar disponible las 24 horas del día, de forma tal que se pueda acceder en cualquier momento.
- Software.
 - Se utilizará la tecnología Apache para el servidor web.
 - Se usará el Sistema Gestor de Base de Datos PostgreSQL.

- El sistema se implementará con el lenguaje de programación PHP 5.
- Framework Symfony.

➤ Ayuda.

- Cada funcionalidad tendrá una ayuda, en la que se dará una explicación detallada del trabajo con la misma.

Definición de los casos de uso

Justificación de los actores del sistema

Actores del sistema	Justificación
Usuario	Este actor representa los roles de los usuarios que acceden al sistema.
Administrador del sistema	Este actor es el encargado de gestionar usuario, rol, área, proyecto, polo y criterio de valoración. Se encarga además de designar el permiso a cada rol.
Jefe de proyecto	Este actor es el encargado de registrar el Reporte de Estado y el Reporte Postmortem, y tendrá el permiso para visualizar los reportes de su proyecto.
Evaluador	Este actor representa al Jefe de Polo, al Vicedecano de Producción y al Decano. Es el encargado de emitir los criterios de valoración y generar informes.

Una vez que se definen los actores del sistema, el diagrama de actores del sistema quedaría de la siguiente forma:

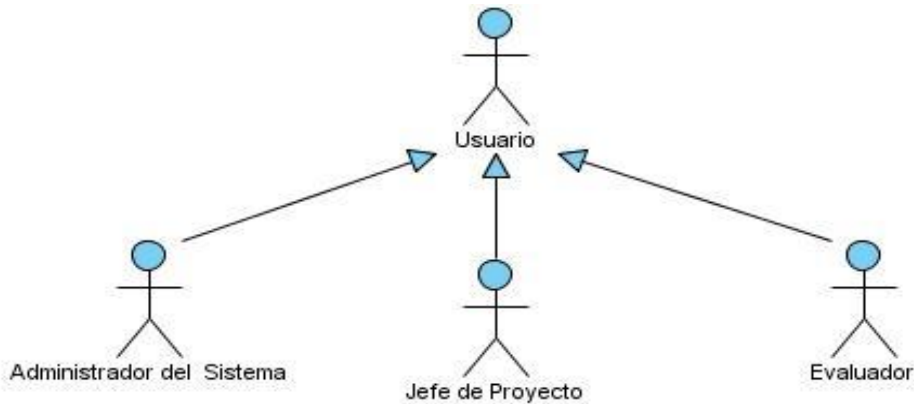


Figura 4 Diagrama de actores del sistema

Cada usuario del sistema representa un rol y sus responsabilidades, luego de asignar las responsabilidades de cada rol, el diagrama de casos de uso del sistema quedó conformado de la siguiente manera:

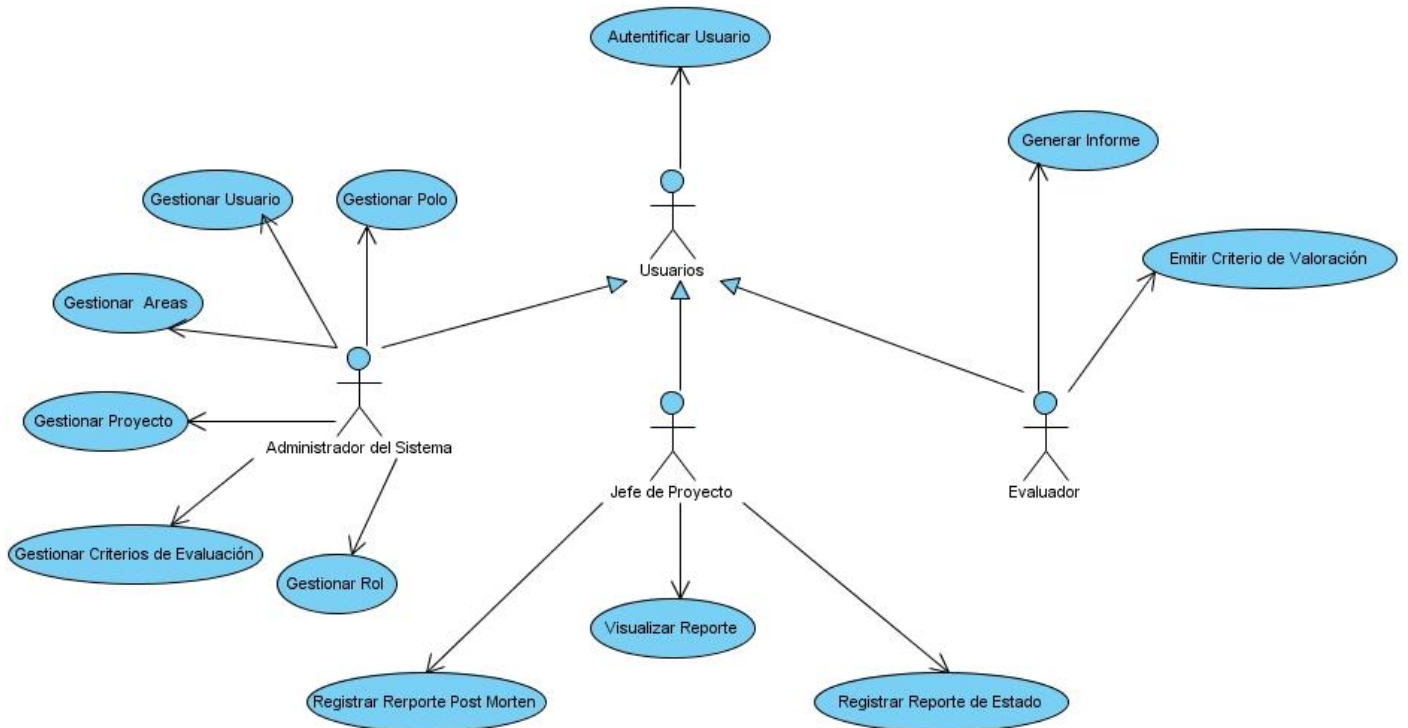


Figura 5 Diagrama de casos de uso del sistema

Descripción de los casos de uso

Tabla 1 Descripción del CU Autenticar Usuario

Caso de uso:	Autenticar Usuario	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el usuario accede por primera vez a la aplicación y entra sus datos.	
Precondiciones:	El usuario debe estar autenticado.	
Referencias:	RF 1.1	
Prioridad:	Critico	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor introduce los datos: <ul style="list-style-type: none"> • Usuario • Contraseña Selecciona la opción "Aceptar".	2. El sistema verifica que los datos introducidos sean correctos.	
	3. El sistema muestra la interfaz principal. Terminando así el caso de uso.	
Flujos Alternos		
Flujo Alterno al paso 2 "Verificar datos"		
Acción del Actor	Respuesta del Sistema	
	a. En caso de que los datos introducidos	

	<p>sean incorrectos, el sistema muestra el siguiente mensaje: “Verifique los datos introducidos”.</p>
--	---

Tabla 2 Descripción del CU Gestionar Área

Caso de uso:	Gestionar Área	
Actores:	Administrador del sistema	
Resumen:	<p>El caso de uso inicia cuando el actor necesita gestionar los datos de un área. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la adición, también puede que seleccione un área existente ya sea para eliminarla o modificarla. El caso de uso termina con la adición, modificación o eliminación de un área.</p>	
Precondiciones:	El usuario debe ser el Administrador del sistema	
Referencias:	RF 2.1, RF 2.2, RF 2.3	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Sección “Adicionar Área”		
	Acción del actor	Respuesta del sistema
	<p>1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Área”.</p>	<p>2. El sistema muestra la interfaz correspondiente a gestionar área donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.</p>

3. El actor selecciona la opción “Adicionar”.	4. El sistema muestra la interfaz “Adicionar Área”.
5. El actor introduce los datos correspondientes al área: <ul style="list-style-type: none"> • Nombre • Descripción Selecciona la opción “Aceptar”.	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema crea una nueva área y regresa al paso 2 de la sección “Adicionar Área” del flujo normal de eventos, terminando así el caso de uso.
Flujos Alternos	
Flujo Alterno al paso 5 “Cancelar Adición”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Área” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Editar Área”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Área”.	2. El sistema muestra la interfaz correspondiente a gestionar área donde aparecen un conjunto de propuestas a las

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

	que tiene acceso el actor administrador del sistema.
3. El actor elige el área que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz “Editar Área”.
5. El actor introduce los datos correspondientes al área: <ul style="list-style-type: none"> • Nombre • Descripción Selecciona la opción “Aceptar”.	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema modifica el área y regresa al paso 2 de la sección “Modificar Área” del flujo normal de eventos, terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Edición”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Editar Área” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Eliminar Área”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Área”.	2. El sistema muestra la interfaz correspondiente a gestionar área donde

	aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el área que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: “¿Está seguro que desea eliminar el área?”.
5. El actor selecciona la opción “Aceptar”.	6. El sistema elimina el área y regresa al paso 2 de la sección “Eliminar Área” del flujo normal de eventos. Terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Eliminación”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Eliminar Área” del flujo normal de eventos. Terminando así el caso de uso.

Tabla 3 Descripción del CU Gestionar Polo

Caso de uso:	Gestionar Polo
Actores:	Administrador del sistema
Resumen:	El caso de uso inicia cuando el actor necesita gestionar los datos de un polo. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la creación, también puede que seleccione un polo existente ya sea para eliminarlo o modificarlo. El caso de uso termina con la adición, modificación o eliminación de un polo.

Precondiciones:	El usuario debe ser el Administrador del sistema	
Referencias:	RF 3.1, RF 3.2, RF 3.3	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Sección “Adicionar Polo”		
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Polo”.	2. El sistema muestra la interfaz correspondiente a gestionar polo donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.	
3. El actor selecciona la opción “Adicionar”.	4. El sistema muestra la interfaz “Adicionar Polo”.	
5. El actor introduce los datos correspondientes al polo: <ul style="list-style-type: none"> • Área • Nombre • Descripción Selecciona la opción “Aceptar”.	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.	
	7. El sistema crea un nuevo polo y regresa al paso 2 de la sección “Adicionar Polo” del flujo normal de eventos, terminando así el	

	caso de uso.
Flujos Alternos	
Flujo Alterno al paso 5 “Cancelar Adición”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Polo” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Editar Polo”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Polo”.	2. El sistema muestra la interfaz correspondiente a gestionar polo donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el polo que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz “Editar Polo”.
5. El actor introduce los datos correspondientes al polo: <ul style="list-style-type: none"> • Área • Nombre • Descripción 	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.

Selecciona la opción "Aceptar".	
	7. El sistema modifica el polo y regresa al paso 2 de la sección "Editar Polo" del flujo normal de eventos, terminando así el caso de uso.
Flujo Alternativo al paso 5 "Cancelar Edición"	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción "Cancelar".	b. El sistema cancela la operación y regresa al paso 2 de la sección "Editar Polo" del flujo normal de eventos. Terminando así el caso de uso.
Sección "Eliminar Polo"	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Gestionar Polo".	2. El sistema muestra la interfaz correspondiente a gestionar polo donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el polo que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: "¿Está seguro que desea eliminar el polo?".
5. El actor selecciona la opción "Aceptar".	6. El sistema elimina el polo y regresa al paso 2 de la sección "Eliminar Polo" del flujo normal de eventos. Terminando así el caso de uso.

Flujo Alterno al paso 5 “Cancelar Eliminación”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la eliminación y regresa al paso 2 de la sección “Eliminar Polo” del flujo normal de eventos. Terminando así el caso de uso.

Tabla 4 Descripción del CU Gestionar Proyecto

Caso de uso:	Gestionar Proyecto
Actores:	Administrador del sistema
Resumen:	El caso de uso inicia cuando el actor necesita gestionar los datos de un proyecto. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la creación, también puede que seleccione un proyecto existente ya sea para eliminarlo o modificarlo. El caso de uso termina con la adición, modificación o eliminación de un proyecto.
Precondiciones:	El usuario debe ser el Administrador del sistema
Referencias:	RF 4.1, RF 4.2, RF 4.3
Prioridad:	Crítico
Flujo Normal de Eventos	
Sección “Adicionar Proyecto”	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor	2. El sistema muestra la interfaz correspondiente a gestionar proyecto

selecciona la opción “Gestionar Proyecto”.	donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor selecciona la opción “Adicionar”.	4. El sistema muestra la interfaz “Adicionar Proyecto”.
5. El actor introduce los datos correspondientes al proyecto: <ul style="list-style-type: none"> • Polo • Nombre • Lista de usuarios • Descripción Selecciona la opción “Aceptar”.	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema crea un nuevo proyecto y regresa al paso 2 de la sección “Adicionar Proyecto” del flujo normal de eventos, terminando así el caso de uso.
Flujos Alternos	
Flujo Alterno al paso 5 “Cancelar Adición”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Proyecto” del flujo normal de eventos. Terminando así el caso de uso.

Sección “Editar Proyecto”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Proyecto”.	2. El sistema muestra la interfaz correspondiente a gestionar proyecto donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el proyecto que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz “Editar Proyecto”.
5. El actor introduce los datos correspondientes al proyecto: <ul style="list-style-type: none"> • Polo • Nombre • Lista de usuarios • Descripción Selecciona la opción “Aceptar”.	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema modifica el proyecto y regresa al paso 2 de la sección “Editar Proyecto” del flujo normal de eventos, terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Edición”	
Acción del Actor	Respuesta del Sistema

a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Editar Proyecto” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Eliminar Proyecto”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Proyecto”.	2. El sistema muestra la interfaz correspondiente a gestionar proyecto donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el proyecto que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: “¿Está seguro que desea eliminar el proyecto?”.
5. El actor selecciona la opción “Aceptar”.	6. El sistema elimina el proyecto y regresa al paso 2 de la sección “Eliminar Proyecto” del flujo normal de eventos. Terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Eliminación”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la eliminación y regresa al paso 2 de la sección “Eliminar Proyecto” del flujo normal de eventos. Terminando así el caso de uso.

Tabla 5 Descripción del CU Gestionar Rol

Caso de uso:	Gestionar Rol	
Actores:	Administrador del sistema	
Resumen:	El caso de uso inicia cuando el actor necesita gestionar los datos de un rol. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la creación, también puede que seleccione un rol existente ya sea para eliminarlo o modificarlo. El caso de uso termina con la adición, modificación o eliminación de un rol.	
Precondiciones:	El usuario debe ser el Administrador del sistema	
Referencias:	RF 5.1, RF 5.2, RF 5.3	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Sección “Adicionar Rol”		
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Rol”.	2. El sistema muestra la interfaz correspondiente a gestionar rol donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.	
3. El actor selecciona la opción “Adicionar”.	4. El sistema muestra la interfaz “Adicionar Rol”.	
5. El actor introduce los datos correspondientes al rol: <ul style="list-style-type: none"> • Nombre 	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.	

<ul style="list-style-type: none"> • Descripción <p>Selecciona la opción “Aceptar”.</p>	
	7. El sistema crea un nuevo rol y regresa al paso 2 de la sección “Adicionar Rol” del flujo normal de eventos, terminando así el caso de uso.
Flujos Alternos	
Flujo Alterno al paso 5 “Cancelar Adición”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Rol” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Editar Rol”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Rol”.	2. El sistema muestra la interfaz correspondiente a gestionar rol donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el rol que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz “Editar Rol”.
5. El actor introduce los datos correspondientes al	6. El sistema valida que los datos introducidos son correctos y/o que no hay

rol: <ul style="list-style-type: none"> • Nombre • Descripción Selecciona la opción "Aceptar".	campos obligatorios vacíos.
	7. El sistema modifica el rol y regresa al paso 2 de la sección "Editar Rol" del flujo normal de eventos, terminando así el caso de uso.
Flujo Alternativo al paso 5 "Cancelar Edición"	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción "Cancelar".	b. El sistema cancela la operación y regresa al paso 2 de la sección "Editar Rol" del flujo normal de eventos. Terminando así el caso de uso.
Sección "Eliminar Rol"	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Gestionar Rol".	2. El sistema muestra la interfaz correspondiente a gestionar rol donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el rol que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: "¿Está seguro que desea eliminar el rol?".
5. El actor selecciona la opción "Aceptar".	6. El sistema elimina el rol y regresa al paso 2

	de la sección “Eliminar Rol” del flujo normal de eventos. Terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Eliminación”	
Acción del Actor	Respuesta del Sistema
c. El actor selecciona la opción “Cancelar”.	d. El sistema cancela la eliminación y regresa al paso 2 de la sección “Eliminar Rol” del flujo normal de eventos. Terminando así el caso de uso.

Tabla 6 Descripción del CU Gestionar Usuario

Caso de uso:	Gestionar Usuario
Actores:	Administrador del sistema
Resumen:	El caso de uso inicia cuando el actor necesita gestionar los datos de un usuario. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la creación, también puede que seleccione un usuario existente ya sea para eliminarlo o modificarlo. El caso de uso termina con la adición, modificación o eliminación de un usuario.
Precondiciones:	El usuario debe ser el Administrador del sistema
Referencias:	RF 6.1, RF 6.2, RF 6.3
Prioridad:	Crítico
Flujo Normal de Eventos	
Sección “Adicionar Usuario”	

Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Gestionar Usuario".	2. El sistema muestra la interfaz correspondiente a gestionar usuario donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor selecciona la opción "Adicionar".	4. El sistema muestra la interfaz "Adicionar Usuario".
5. El actor introduce los datos correspondientes al usuario: <ul style="list-style-type: none"> • Área • Rol • Usuario • Lista de Proyecto Selecciona la opción "Aceptar".	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema crea un nuevo usuario y regresa al paso 2 de la sección "Adicionar Usuario" del flujo normal de eventos, terminando así el caso de uso.
Flujos Alternos	
Flujo Alterno al paso 5 "Cancelar Adición"	
Acción del Actor	Respuesta del Sistema

a. El actor selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Usuario” del flujo normal de eventos. Terminando así el caso de uso.
Sección “Editar Usuario”	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Usuario”.	2. El sistema muestra la interfaz correspondiente a gestionar usuario donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el usuario que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz “Editar Usuario”.
<p>5. El actor introduce los datos correspondientes al usuario:</p> <ul style="list-style-type: none"> • Área • Rol • Usuario • Lista de Proyecto <p>Selecciona la opción “Aceptar”.</p>	6. El sistema valida que los datos introducidos son correctos y/o que no hay campos obligatorios vacíos.
	7. El sistema modifica el usuario y regresa al paso 2 de la sección “Editar Usuario” del flujo normal de eventos, terminando así el

	caso de uso.
Flujo Alternativo al paso 5 "Cancelar Edición"	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción "Cancelar".	b. El sistema cancela la operación y regresa al paso 2 de la sección "Editar Usuario" del flujo normal de eventos. Terminando así el caso de uso.
Sección "Eliminar Usuario"	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Gestionar Usuario".	2. El sistema muestra la interfaz correspondiente a gestionar usuario donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
3. El actor elige el usuario que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: "¿Está seguro que desea eliminar el usuario?".
5. El actor selecciona la opción "Aceptar".	6. El sistema elimina el usuario y regresa al paso 2 de la sección "Eliminar Usuario" del flujo normal de eventos. Terminando así el caso de uso.
Flujo Alternativo al paso 5 "Cancelar Eliminación"	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción "Cancelar".	b. El sistema cancela la eliminación y regresa al paso 2 de la sección "Eliminar Usuario"

	del flujo normal de eventos. Terminando así el caso de uso.
--	---

Tabla 7 Descripción del CU Gestionar Criterio

Caso de uso:	Gestionar Criterio	
Actores:	Administrador del sistema	
Resumen:	El caso de uso inicia cuando el actor necesita gestionar los criterios de un usuario. Consiste en que el administrador del sistema selecciona la opción adicionar y llena los campos requeridos para la creación, también puede que seleccione un criterio existente ya sea para eliminarlo o modificarlo. El caso de uso termina con la adición, modificación o eliminación de un criterio.	
Precondiciones:	El usuario debe ser el Administrador del sistema	
Referencias:	RF 7.1, RF 7.2, RF 7.3	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Sección “Adicionar Criterio”		
	Acción del actor	Respuesta del sistema
	1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Criterio”.	2. El sistema muestra la interfaz correspondiente a gestionar criterio donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del sistema.
	3. El actor selecciona la opción “Adicionar”.	4. El sistema muestra la interfaz “Adicionar

	Criterio”.
<p>5. El actor introduce los siguientes datos:</p> <ul style="list-style-type: none"> • Rol del usuario que va a entrar el criterio • Valor del criterio • Criterio <p>Selecciona la opción “Adicionar”.</p>	<p>6. El sistema adicionar el criterio y muestra los datos.</p>
<p>7. El actor selecciona la opción “Aceptar”.</p>	<p>8. El sistema crea un nuevo criterio y regresa al paso 2 de la sección “Adicionar Criterio” del flujo normal de eventos, terminando así el caso de uso.</p>
Flujos Alternos	
Flujo Alterno al paso 5 “Cancelar Adición”	
Acción del Actor	Respuesta del Sistema
<p>a. El actor selecciona la opción “Cancelar”.</p>	<p>b. El sistema cancela la operación y regresa al paso 2 de la sección “Adicionar Criterio” del flujo normal de eventos. Terminando así el caso de uso.</p>
Sección “Editar Criterio”	
Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso inicia cuando el actor selecciona la opción “Gestionar Criterio”.</p>	<p>2. El sistema muestra la interfaz correspondiente a gestionar criterio donde aparecen un conjunto de propuestas a las</p>

	que tiene acceso el actor administrador del sistema.
3. El actor elige el criterio que desea modificar y selecciona la opción Editar.	4. El sistema muestra la interfaz "Editar Criterio".
5. El actor introduce los datos correspondientes al criterio: <ul style="list-style-type: none"> • Rol del usuario que va a entrar el criterio • Valor del criterio • Criterio Selecciona la opción "Aceptar".	6. El sistema modifica el criterio y regresa al paso 2 de la sección "Editar Criterio" del flujo normal de eventos, terminando así el caso de uso.
Flujo Alternativo al paso 5 "Cancelar Edición"	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción "Cancelar".	b. El sistema cancela la operación y regresa al paso 2 de la sección "Editar Criterio" del flujo normal de eventos. Terminando así el caso de uso.
Sección "Eliminar Criterio"	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Gestionar Criterio".	2. El sistema muestra la interfaz correspondiente a gestionar criterio donde aparecen un conjunto de propuestas a las que tiene acceso el actor administrador del

	sistema.
3. El actor elige el criterio que desea eliminar y selecciona la opción Eliminar.	4. El sistema muestra el mensaje: “¿Está seguro que desea eliminar el criterio?”.
5. El actor selecciona la opción “Sí”.	6. El sistema elimina el criterio y regresa al paso 2 de la sección “Eliminar Criterio” del flujo normal de eventos. Terminando así el caso de uso.
Flujo Alternativo al paso 5 “Cancelar Eliminación”	
Acción del Actor	Respuesta del Sistema
a. El actor selecciona la opción “No”.	b. El sistema cancela la eliminación y regresa al paso 2 de la sección “Eliminar Criterio” del flujo normal de eventos. Terminando así el caso de uso.

Tabla 8 Descripción del CU Registrar Reporte de Estado

Caso de uso:	Registrar Reporte de Estado
Actores:	Jefe de Proyecto
Resumen:	El caso de uso inicia cuando el Jefe de Proyecto desea llenar los datos del Reporte de Estado de su Proyecto.
Precondiciones:	El Jefe de Proyecto debe tener los permisos necesarios para llenar el Reporte de Estado.
Referencias:	RF 8.1
Prioridad:	Critico

Flujo Normal de Eventos	
Sección “Entrar datos”	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Reporte de Estado”.	2. El sistema muestra la interfaz Reporte de Estado donde aparecen todos los campos a llenar en el reporte.
3. El actor llena los campos: <ul style="list-style-type: none"> • Fecha inicial • Fecha del alcance • Estado del proyecto 	
4. El actor entra un Hito/Entregable y selecciona la opción “Adicionar”.	5. El sistema adiciona el hito.
6. Si desea entrar otro hito, regresa al paso 4 de la sección “Entrar datos” del flujo normal de eventos.	
7. El actor selecciona un entregable y entra los datos de éste: <ul style="list-style-type: none"> • Entrega • Fecha Prevista • % completado • Estado de entrega 	8. El sistema adiciona los datos.

Luego selecciona la opción “Adicionar”.	
9. El actor llena los campos correspondientes a “Afectaciones sobre el proyecto”: <ul style="list-style-type: none">• Cambios sobre hitos o alcance del proyecto.• Breve descripción sobre esos cambios.	
10. El actor llena los campos correspondientes a “Presupuesto del proyecto/Estado financiero”: <ul style="list-style-type: none">• Elemento del presupuesto• Presupuesto planeado• Costo actual• Varianza	
11. Para llenar los datos correspondientes a los riesgos del proyecto el actor selecciona la opción “Adicionar” y luego llena los siguientes campos: <ul style="list-style-type: none">• Riesgo• Descripción• Probabilidad• Impacto• Prioridad	

<ul style="list-style-type: none"> • Cambio desde la última descripción 	
12. Si desea entrar otro riesgo, regresa al paso 11 de la sección “Entrar datos” del flujo normal de eventos.	
<p>13. Para llenar los datos correspondientes a las tareas del proyecto el actor selecciona la opción “Adicionar” y luego llena los siguientes campos:</p> <ul style="list-style-type: none"> • Tarea • Descripción • Impacto • Fecha • Estado • Solución 	
14. Si desea entrar otra tarea, regresa al paso 13 de la sección “Entrar datos” del flujo normal de eventos.	
15. El actor selecciona la opción “Salvar.”	16. El sistema guarda el Reporte de Estado, terminando así el caso de uso.

Tabla 9 Descripción del CU Registrar Reporte Postmortem

Caso de uso:	Registrar Reporte Postmortem
---------------------	------------------------------

Actores:	Jefe de Proyecto
Resumen:	El caso de uso inicia cuando el Jefe de Proyecto desea llenar los datos del Reporte Postmortem de su Proyecto.
Precondiciones:	El Jefe de Proyecto debe tener los permisos necesarios para llenar el Reporte Postmortem.
Referencias:	RF 9.1
Prioridad:	Critico
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción "Reporte Postmortem".	2. El sistema muestra la interfaz Reporte Postmortem donde aparecen todos los campos a llenar en el reporte.
3. El actor llena los campos: <ul style="list-style-type: none"> • Ejecución Contra Objetivos • Ejecución Contra Programa • Ejecución Contra Calidad • Ejecución Contra Presupuesto • Planificación de proyecto • Investigación y Desarrollo • Gestión del Proyecto • Aseguramiento de la Calidad & Soporte • Comercialización • Ventas • Lecciones aprendidas • Recomendaciones 	4. El sistema guarda el Reporte Postmortem, terminando así el caso de uso.

<ul style="list-style-type: none"> • Preguntas & Comentarios 	
---	--

Tabla 10 Descripción del CU Emitir Criterio de Valoración

Caso de uso:	Emitir Criterio de Valoración	
Actores:	Evaluador	
Resumen:	El caso de uso inicia cuando el evaluador desea emitir criterios de valoración sobre un proyecto.	
Precondiciones:	El evaluador debe tener los permisos necesarios para emitir los criterios.	
Referencias:	RF 10.1	
Prioridad:	Critico	
Flujo Normal de Eventos		
	Acción del actor	Respuesta del sistema
	1. El caso de uso inicia cuando el actor selecciona la opción Criterio de Usuario.	2. El sistema muestra la interfaz Criterio de Usuario con los criterios a los que tiene acceso el rol que desempeñe el usuario.
	3. El actor selecciona el proyecto sobre el cual va a emitir criterios. Luego selecciona los criterios que corresponden al proyecto que va a evaluar y selecciona la opción "Guardar".	4. El sistema guarda los criterios de valoración emitidos sobre el proyecto, terminando así el caso de uso.
Flujos Alternos		
Flujo Alterno al paso 2 "Cancelar"		
	Acción del Actor	Respuesta del Sistema

a. El usuario selecciona la opción “Cancelar”.	b. El sistema cancela la operación y regresa a la interfaz principal, terminando así el caso de uso.
--	--

Tabla 11 Descripción del CU Visualizar Reporte

Caso de uso:	Visualizar Reporte	
Actores:	Jefe de Proyecto	
Resumen:	El caso de uso inicia cuando el Jefe de Proyecto desea visualizar un reporte.	
Precondiciones:	El Jefe de Proyecto debe tener los permisos necesarios para visualizar un reporte.	
Referencias:	RF 11.1	
Prioridad:	Critico	
Flujo Normal de Eventos		
Sección “Visualizar”		
	Acción del actor	Respuesta del sistema
	1. El caso de uso inicia cuando el actor selecciona la opción “Visualizar Reporte”.	2. El sistema muestra la interfaz Visualizar Reporte.
	3. El Actor selecciona la fecha del reporte que desea visualizar.	4. El sistema muestra el reporte solicitado por el actor, terminando así el caso de uso.

Tabla 12 Descripción del CU Generar Informe

Caso de uso:	Generar Informe
---------------------	-----------------

Actores:	Evaluador
Resumen:	El caso de uso inicia cuando el Evaluador desea generar un informe.
Precondiciones:	El Evaluador debe tener los permisos necesarios para generar el informe.
Referencias:	RF 12.1, 12.2, 12.3, 12.4, 12.5
Prioridad:	Critico
Flujo Normal de Eventos	
Sección “Entrar datos”	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Generar Informe”.	2. El sistema muestra la interfaz Generar Informe donde aparecen todos los campos a llenar en el informe.
3. El actor llena los campos: <ul style="list-style-type: none"> • Área • Polo • Proyecto • Fecha desde(Reporte Estado) • Fecha hasta(Reporte Estado) • Fecha (Reporte Estado) Luego selecciona la opción “Aceptar”.	4. El sistema muestra el informe, terminando así el caso de uso.

Conclusiones del Capítulo 2

Luego de realizar un estudio del negocio existente, se determinó que para conocer el estado y el rendimiento de los proyectos de la UCI durante su desarrollo, es necesario que se automaticen aquellos procesos que forman parte del monitoreo y control del proyecto. Se hizo un análisis del negocio, que permitió determinar qué funcionalidades deberían ser implementadas, así como los actores que intervendrían en el sistema. De esta forma, se pudo establecer las características que tendrá el sistema que será implementado.

CAPÍTULO 3: ANÁLISIS Y DISEÑO.

En este capítulo se presenta el análisis y diseño para el sistema desarrollado. El análisis, a través de sus diagramas de clases, dará una visión general de lo que se hará en la implementación del sistema; mientras que el diseño, mediante los diagramas de clases correspondientes, dará como resultado una proyección de la implementación. Como parte del diseño se expondrán además los patrones de diseño y arquitectura que se aplican en el desarrollo de la aplicación, y se mostrarán también los diagramas de secuencia del diseño. Finalmente se describirán las clases que se obtienen en el diseño y el diseño de la Base de Datos.

Análisis

En esta etapa se refinan y estructuran los requisitos funcionales descritos en el capítulo anterior, con el objetivo de conseguir una mejor comprensión y una descripción más detallada que ayude a estructurar el sistema. En esta fase se realizan los diagramas de clases del análisis del sistema propuesto, donde se representan los conceptos en un dominio del problema. Estos diagramas muestran las clases que participan en las realizaciones de los distintos casos de usos planteados anteriormente. En el [Anexo 5](#) se muestran los diagramas de clases del análisis especificados por casos de uso.

Diseño

Lo esencial en esta etapa es la construcción de los diagramas de clases del diseño, que muestra las clases que participan en la realización de un caso de uso con todos sus atributos. Este diagrama es una descripción de los modelos de objetos, contiene clases y las relaciones estructurales y de herencia existentes entre ellas. Los diagramas de clases del diseño para cada caso de uso se muestran en el [Anexo 6](#).

En esta etapa se elaboran también los diagramas de interacción, que pueden ser de colaboración o de secuencia. Estos diagramas muestran gráficamente cómo los objetos se comunican entre ellos con el objetivo de cumplir los requerimientos. En el presente trabajo se muestran los diagramas de secuencia, que describen las secuencias de interacciones ordenadas en el tiempo. El diagrama de secuencia muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo. Este diagrama es el núcleo

de un modelo dinámico, y muestra todos los cursos alternos que pueden tomar los casos de uso. Los diagramas de secuencias se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos. Estos diagramas se muestran en el [Anexo 7](#).

Aplicación de patrones en Symfony

Para la implementación de Symfony se utilizan varios patrones, de los cuales se mencionan a continuación algunos ejemplos, situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC (ZANINOTTO and POTENCIER 2009).

Patrones GRASP

Experto: Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Propel para mapear la BD. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos (Peer del Modelo) poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

Creador: En la clase `tIActions` se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase `tIActions` es “creador” de dichas entidades. Ejemplos de algunas funciones utilizadas en la clase `tIActions` son: `doSelect ()`, `retrieveByPK ()`, `doSelectOne ()`.

Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase `tIActions`, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

Bajo Acoplamiento: La clase `tIActions` hereda únicamente de `sfActions` para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Controlador: Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases sfFrontController, sfWebFrontController, sfContext, los “actions” y el index.php del ambiente.

Patrones GoF

Patrón Singleton

Clase sfRouting – método getInstance

Esta clase la utiliza el controlador frontal (sfWebFrontController) y se encarga de enrutar todas las peticiones que se hagan a la aplicación. El singleton sfRouting precisa otros métodos muy útiles para la gestión manual de las rutas: ClearRoutes (), hasRoutes (), getRoutesByName ().

Patrón Command

Este patrón se observa en la clase sfWebFrontController, en el método dispatch (). Esta clase está por defecto y es la encargada de establecer el módulo y la acción que se va a usar según la petición del usuario. Este patrón se aplica además en la clase sfRouting, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el framework, la cual se puede activar o desactivar.

En este método es parseada la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el Actions que debe responder a la petición.

Patrón Decorador

Este método pertenece a la clase abstracta sfView, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente.

El archivo nombrado layout.php es el que contiene el Layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este procedimiento es una implementación del patrón Decorator.

Patrón Registry

Este patrón es muy útil para los desarrolladores en la Programación Orientada a Objetos. Este patrón es un medio sencillo y eficiente de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales. Este patrón se aplica en la clase `sfConfig`, que es la encargada de acumular todas las variables de uso global en el sistema.

Symfony aplica además el patrón “Front Controller” (Controlador frontal), por lo que posee una estructura bien organizada de controladores, que comienza desde el “`index.php`” del ambiente y termina en los “Actions”. Cada clase de esta capa tiene su responsabilidad y es única, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros YML, y otros que se encargan de identificar mediante algunos datos las clases que deben realizar determinadas tareas (Patrón GoF Command, clase `sfRouting`) y las clases relacionadas con la configuración del sistema (`sfConfig`, y `sfConfigHandler`).

Patrón MVC

El uso del framework que utiliza MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil además de restrictivo. La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

- `sfController`: Es la clase del controlador y se encarga de decodificar la petición y transferirla a la acción correspondiente.
- `sfRequest`: Guarda todos los elementos que integran la petición (parámetros, cookies, cabeceras, etc.)
- `sfResponse`: Posee las cabeceras de la respuesta y los contenidos. El contenido de este objeto se convierte en la respuesta HTML que se remite al usuario.
- El singleton de contexto (que se obtiene mediante `sfContext::getInstance()`): Guarda una referencia a todos los objetos que constituyen el núcleo de Symfony y puede ser accedido desde cualquier parte de la aplicación.

Symfony toma lo mejor de la arquitectura MVC y la realiza de modo que el desarrollo de aplicaciones sea rápido y sencillo.

En el controlador se encuentran las acciones, las cuales son el núcleo de la aplicación, pues contienen toda la lógica de la aplicación. Estas acciones utilizan el modelo y precisan las variables para la vista. Al realizarse una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.

La vista es la encargada de originar las páginas que son mostradas como resultado de las acciones, donde se encuentra el layout, que es común para todas las páginas de la aplicación. La vista en Symfony está conformada por varias partes, preparadas cada una de ellas especialmente para ser fácilmente transformable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

En el Modelo se encuentran las clases, que son generadas de forma automática según la estructura de la BD. En Symfony, el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos. Propel es el motor generador que se encarga de esta generación automática para construir sus clases, creando la estructura y generando el código de las mismas.

A medida que el desarrollo de un proyecto va avanzando, puede ser necesario agregar métodos y propiedades personalizadas en los objetos del modelo, esto trae consigo que se aumenten las tablas o columnas. Asimismo, cada vez que se modifica se deben regenerar las clases del modelo de objetos. Si se añaden los métodos personalizados en las clases que se generan, cada vez que se vuelvan a generar esas clases estos métodos se borrarían.

Las clases con nombre Base son generadas directamente a partir del esquema. Estas clases no deberían modificarse, ya que cada vez que se genera el modelo, todas las clases se borran. Por otro lado, las clases de objetos propias que heredan de las clases con nombre Base no se modifican, es por ello que son estas clases en las que se añaden los métodos propios. Las mismas heredan todos los métodos de la clase padre correspondiente, pero no son afectadas por las modificaciones en el esquema.

Descripción de las clases

La descripción de las clases proporciona detalles de las clases Base, Actions y Peer que están involucradas en la realización de los casos de uso del sistema y se encuentran en el [Anexo 8](#).

Diseño de la BD

El almacenamiento de la información es uno de los aspectos más importantes para cualquier aplicación; por lo cual en este epígrafe se modela la base de datos que será utilizada en el sistema propuesto. Aquí se exponen los distintos diagramas Entidad Relación agrupados por paquetes de funcionalidad. Ver [Anexo 9](#).

Conclusiones del Capítulo 3.

Con este capítulo se ha dado fin al flujo de trabajo de análisis y diseño que RUP plantea. El mismo sirvió para establecer los artefactos necesarios que facilitarían el trabajo a los desarrolladores, proporcionándoles los elementos necesarios para la implementación del sistema.

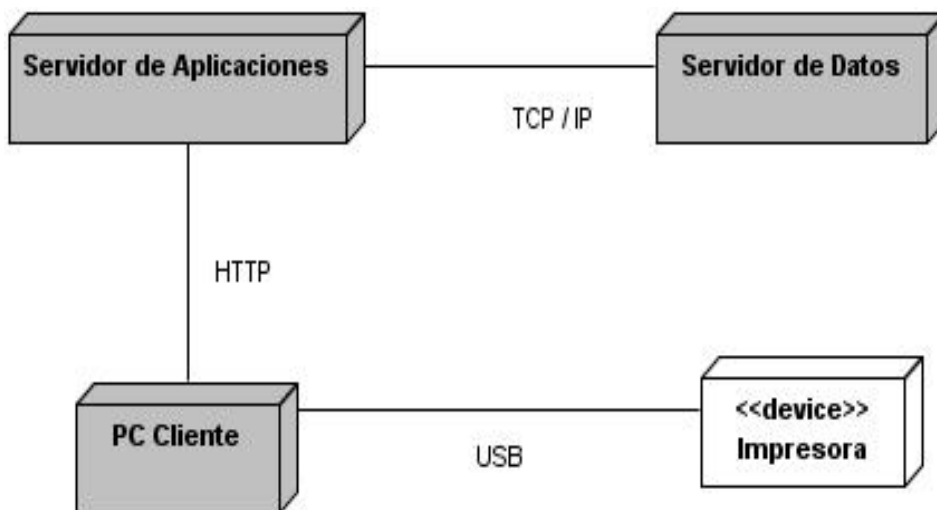
CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA

En este capítulo se presentan los diagramas construidos en la fase de elaboración: el diagrama de despliegue y los diagramas de componentes. Se comienza además con la validación de la solución propuesta, a través de diferentes métodos de prueba. Para ello se utilizan algunas métricas técnicas orientadas a objetos, con el objetivo de mejorar la calidad del trabajo llevado a cabo y evaluar la efectividad del proceso. Se realizan también pruebas de rendimiento (stress) al sistema y se realizan pruebas de caja negra.

Implementación

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo éstos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

El diagrama de despliegue representa la distribución física de la aplicación. La estructura de nodos que en éste se utiliza, está diseñado para permitir que el usuario tenga comodidad en el trabajo y proporcionarle una seguridad confiable.



El diagrama de componentes representa los elementos físicos del sistema y sus relaciones. El mismo muestra las dependencias lógicas entre los componentes de software, ya sean componentes, fuentes, binarios o ejecutables. En el [Anexo 10](#) se muestra el diagrama de componentes del sistema.

Estándar de codificación

Estándar de codificación.

Para el desarrollo del sistema y asegurar la calidad del mismo, uno de los instrumentos que facilitan estas tareas es la adopción de estándares de estilo y codificación.

El uso de estos estándares permite:

- Asegurar la legibilidad del código entre distintos programadores.
- Proveer una guía para el encargado de mantenimiento o actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

Es por esto que la codificación del sistema a desarrollar debe cumplir ciertos requisitos, detallados en el presente documento. Estos requisitos están basados en el estándar que propone la comunidad Extensión de PHP y Repositorio de Aplicaciones (Comunidad PEAR). Pueden aplicarse a cualquier lenguaje de programación, exceptuando algunas reglas específicas correspondientes a PHP.

Los estándares se resumen en los siguientes puntos:

Estándar 1: La indentación debe ser a cuatro espacios sin caracteres de tabulación. Esto es debido a que ciertos IDE's de desarrollo introducen caracteres de tabulación cuando indentan un texto automáticamente. Se recomienda el uso de herramientas o editores generales.

Estándar 2: Las estructuras de control deben tener un espacio entre el **keyword** de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones, el signo de llaves debe estar sobre la línea de la estructura.

Estándar 3: Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

Estándar 4: El estilo de los comentarios debe ser como el estilo de comentarios para C (`/* */` ó `//`), no debe utilizarse el estilo de comentarios de Perl (`#`).

Estándar 5: Cuando se incluya un archivo de dependencia incondicionalmente utilice `require_once` y cuando sea condicionalmente, utilice `include_once`.

Estándar 6: Siempre se debe utilizar las etiquetas `<?php ?>` para abrir un bloque de código. No se debe utilizar el método de etiquetas cortas (`<? y ?>` o `<% y %>`) porque esto depende de las directivas de configuración en el archivo `PHP.INI` y hace que el script no sea tan portable.

Estándar 7: Los nombres de las clases deben de iniciar con letra mayúscula. Los nombres de las variables y de las funciones pueden iniciar con letra minúscula, pero si estas tienen más de una palabra, cada nueva palabra debe iniciar con letra mayúscula (el nombre puede escribirse separado por signos de guión mayor). Si una función, en una clase, es privada; deberá comenzar con el signo de guión mayor para una fácil identificación. Las constantes deben de escribirse siempre en mayúsculas y tanto estas como las variables globales deben de tener como prefijo el nombre de la clase a la que pertenecen.

Estándar 8: Los archivos con código PHP, deben de ser guardados en formato ASCII utilizando la codificación ISO-8859-1. El formato ASCII con codificación ISO-8859-1 es el formato en que se guardan los archivos de texto plano (`.txt`). La razón de este estándar es que determinados editores HTML (en especial Dreamweaver), agregan códigos de carácter extraño de salto de línea (como si se tratara de un archivo binario) y esto puede ocasionar que el intérprete de PHP, encuentre problemas a la hora de leer el script.

Modelo de Prueba

El objetivo de las pruebas es verificar si el software es ejecutado bajo las condiciones o requerimientos especificados. Para comprobar si el sistema desarrollado cumple con las expectativas del cliente, se

utilizaron métricas de validación, se aplicaron pruebas de rendimiento al sistema y se diseñaron los casos de prueba.

Métricas asociadas al diseño

Con el propósito de determinar el grado de calidad y fiabilidad del diseño propuesto, se aplicaron algunas métricas orientadas a objetos: Métrica Tamaño de clase (TC) y Métrica Árbol de profundidad de herencia (APH), las cuales se explican a continuación:

Métrica Tamaño de clase (TC)

Esta métrica, propuesta por Lorenz y Kidd, está centrada en el recuento de operaciones y atributos para cada clase individual y los valores promedio para el sistema Orientado a Objetos como un todo.

Para calcular el tamaño general de una clase se puede determinar el total de operaciones que se encapsulan dentro de la clase, ya sean operaciones heredadas o privadas de la instancia, o se pueden determinar el número de atributos encapsulados por la clase, pueden ser atributos tanto heredados como privados de la instancia.

Luego de calcular los valores para el Tamaño de Clase, si son altos, significa que la clase debe tener bastante responsabilidad, esto disminuiría la reutilización de dicha clase y complicaría la implementación y las pruebas. Se puede calcular además los promedios para el número de atributos y operaciones de clase, y entre más pequeño sea este promedio más reutilizables podrán ser las clases dentro del sistema.

No existe actualmente una medida fija para calcular el umbral para los parámetros de calidad en el diseño de sistemas, no obstante, la mayoría de los especialistas proponen umbrales como se muestra en la siguiente tabla.

Números de Operaciones y Atributos	
TC	Umbral
Pequeño	≤ 20
Medio	> 20 y ≤ 30

Grande	>30
--------	-----

Tabla 47 Umbrales para la Métrica TC

Resultado de la aplicación de esta métrica al sistema.

No.	Nombre de la clase	Cantidad de atributos	Cantidad de operaciones	Tamaño de la clase
1	Areas	3	8	Pequeño
2	Polo	4	10	Pequeño
3	Actividad	6	14	Pequeño
4	CriterioDeUsuario	6	14	Pequeño
5	CriterioEvaluacion	5	12	Pequeño
6	Hitos	3	8	Pequeño
7	Proyecto	4	10	Pequeño
8	Riesgo	8	18	Medio
9	Rol	3	8	Pequeño
10	Tareas	8	18	Medio
11	Usuario	4	10	Pequeño
12	ReporteEstado	8	18	Medio
13	ReportePostMortem	9	20	Medio

Tabla 48 Clases del sistema y su tamaño.

De un total de 13 clases que fueron calculadas, 9 resultaron tener un tamaño pequeño, mientras que las otras 4 tienen un tamaño medio, esto demuestra un buen resultado en la aplicación de la métrica seleccionada.

Métrica Árbol de profundidad de herencia (APH)

Esta métrica es definida como la máxima longitud del nodo a la raíz del árbol. A medida que el valor del APH crece, se hace posible que las clases de más bajos niveles hereden más métodos, lo que trae

consigo que ocurran dificultades al intentar predecir el comportamiento de una clase. Una jerarquía de clases profundas conduciría además a una mayor complejidad en el diseño. Por otro lado, estos altos valores del APH implicarían un gran número de métodos que se reutilizarían.

Luego de aplicarse la métrica se logró un valor de APH igual a 5. Algunos autores como Lorenz y Kidd proponen que un umbral de 6 niveles como indicador es un abuso en la herencia en distintos lenguajes de programación. Por tanto, el valor obtenido en esta métrica demuestra que la solución propuesta es compleja, pues las clases de más bajos niveles heredarían un gran número de métodos, lo que implicaría una gran cantidad de métodos que se reutilizarán.

Pruebas de rendimiento (stress)

Para realizar las pruebas de rendimiento se seleccionó la herramienta Apache Jmeter. Esta herramienta fue desarrollada por la Apache Software Foundation. Es una aplicación 100% Java utilizada habitualmente para realizar pruebas funcionales y medir rendimientos. Se puede utilizar para simular condiciones de carga muy elevada en servidores, redes o aplicaciones concretas, para comprobar su capacidad o analizar el rendimiento general bajo diferentes condiciones de carga.

El objetivo principal de estas pruebas es el de comprobar que el sistema es estable y funciona correctamente a distintos niveles de carga. Además, comprueba que el sistema es capaz de dar servicio simultáneamente a un número suficiente de usuarios.

Las pruebas fueron realizadas al sistema bajo las siguientes características de software:

Servidor web: Apache 2.2.4

Servidor de Base de Datos: PostgreSQL 8.2

Lenguaje utilizado: php 5.2.5.

Con el siguiente hardware de servidor:

Disco Duro: ST380817AS (80 GB)

Memoria del sistema: 256 MB

Procesador: Pentium (R) 4 CPU, 3.00 MHz ()

Para la ejecución satisfactoria de las pruebas automatizadas se realizaron las tareas definidas en la estrategia de pruebas que definen una serie de pasos a seguir como parte del aseguramiento de la calidad del proceso.

- Se comprobó que el Sistema Web estuviera disponible y funcionando de forma estable.
- Se comprobó que las condiciones de configuración no fueron cambiadas.
- Se ejecuta el test de pruebas teniendo como cantidad de usuarios límites 25 usuarios. Con esta cantidad de hilos se grabó un resumen de los resultados de las peticiones del test y se obtienen las condiciones máximas en las que puede funcionar de manera correcta el sistema. Los resultados se muestran en el epígrafe a continuación.

Análisis de los resultados de las Pruebas

La cantidad de peticiones que generó la grabación de la navegación del Sistema Web fue de 175 peticiones mostrando los resultados que aparecen en la figura 5. Los resultados se mostraron por debajo de lo que se podría alcanzar debido a que el sistema estaba instalado en un servidor con deficientes recursos, lo cual limita la cantidad de conexiones y velocidad de respuesta de las peticiones realizadas. Analizando los resultados generales de las pruebas al Sistema Web se obtiene un total de 175 peticiones para un total de 25 usuarios reportando los datos que se muestran a continuación.

Los resultados obtenidos se pueden observar en los informes generados por la herramienta Jmeter. En estas pruebas se utilizaron para visualizar los mismos, un informe agregado, que muestra los siguientes datos:

- **# Muestras:** es la cantidad de veces que se realizó la actividad (una vez por cada usuario).
- **Media:** el promedio o media aritmética del tiempo en milisegundos.
- **Mediana:** mediana del tiempo en milisegundos.
- **Línea de 90 %:** Tiempo de respuesta en el que se encuentra el 90% de las respuestas del servidor.
- **Min:** tiempo mínimo de todas las peticiones de este tipo.
- **Max:** tiempo máximo de todas las peticiones de este tipo.
- **Porcentaje de error:** muestra el porcentaje de las peticiones fallidas.
- **Rendimiento:** está medido en request / (segundo/minuto).

- **KB/ces:** medida de velocidad de Kilobytes/sec.



Figura 5 Concurrencia de 25 usuarios

De esta prueba se destacan las siguientes observaciones a tener en cuenta por el equipo de desarrollo del sistema para lograr una mayor eficiencia.

- El sistema muestra un soporte de concurrencia aproximadamente 175 peticiones simultáneas. Esto significa que con las prestaciones del servidor en el que se encuentra instalado puede soportar un máximo de 25 usuarios concurrentes ya que con 50 no responde las peticiones. Mostrando un tiempo de recuperación de alrededor de 60 segundos restableciéndose de manera eficiente sin intervención del recurso humano.
- La petición:/template/image/imagenes/logoleft.gif no respondió de manera satisfactoria. Por lo que esa petición es fallida. Esta fue la causa principal del 14.29 % de error.
- El Sistema fue probado en condiciones inferiores a las reales en las que se debe desplegar. Para la cantidad de usuarios que soporta en estos momentos con las prestaciones anteriormente mencionadas, se considera satisfactorio el rendimiento de la aplicación. En general las peticiones tienen un tiempo de respuesta de 0.36666 peticiones por segundos lo cual demuestra que ha sido implementado siguiendo los requerimientos establecidos en los que se plantean buenas prácticas de programación.

Para la ejecución de las pruebas de carga se simuló el envío de datos de 25 usuarios utilizando un loop (número de iteraciones que puede realizar el Jmeter que se configura para las repeticiones de peticiones al server) de 30 repeticiones. Las que posibilitaron simular suficiente carga en el sistema.

Los resultados muestran:

- El comportamiento ascendente del rendimiento del sistema mientras disminuye el tiempo de respuesta de las peticiones de manera lineal.
- El comportamiento de la media del tiempo de respuesta de las peticiones de en condiciones extremas de carga.
- El sistema muestra un comportamiento uniforme y estable.

Todo ello teniendo en cuenta que al instalar el servidor en condiciones superiores estos resultados deben verse favorecidos en gran medida, pues estos han afectado la cantidad de usuarios permisibles para estas pruebas de Carga. Estos resultados se muestran en la gráfica siguiente.

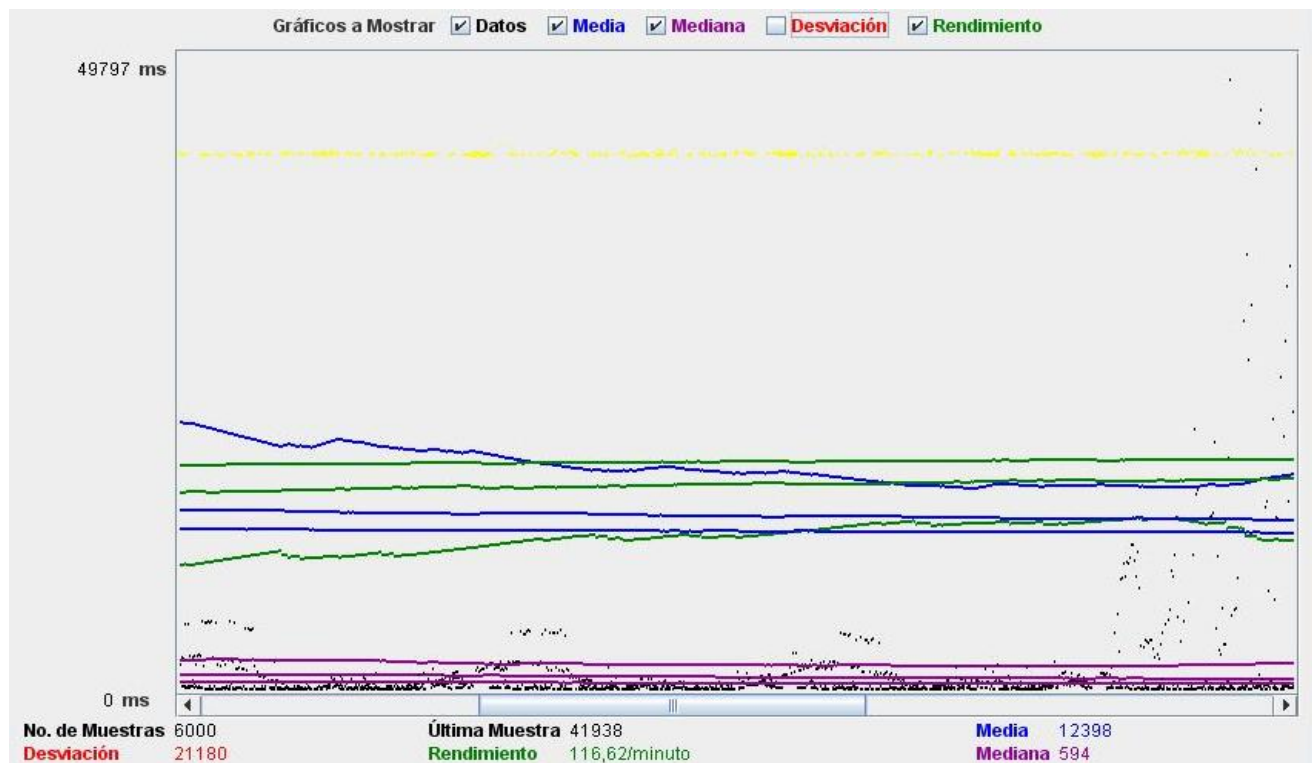


Figura 6 Carga de 300 usuarios

De manera general se pueden observar los distintos comportamientos que presentó el sistema en el [Anexo 11](#).

Pruebas de Caja Negra

Para comprobar la funcionalidad del sistema se diseñaron Casos de Prueba de Funcionalidad. Estos casos de prueba tuvieron resultados satisfactorios, lo que demuestra que el sistema cumple con las expectativas esperadas y con la calidad requerida. En el [Anexo 12](#) se muestran algunos ejemplos.

Conclusiones del Capítulo 4.

Con el desarrollo de este capítulo se termina la implementación del sistema que da solución al problema planteado inicialmente. Las métricas aplicadas sirvieron para determinar el grado de calidad y fiabilidad del diseño propuesto para el sistema. Las pruebas de carga y estrés realizadas permitieron detectar anomalías como los errores por fallo de respuesta de una petición que incurren en el rendimiento del sistema. Finalmente, los diseños de casos de prueba permitieron validar la solución y encontrar los errores ocultos en el sistema.

CONCLUSIONES GENERALES

El desarrollo de este Trabajo de Diploma permitió a los autores la adquisición de importantes conocimientos para la creación del sistema implementado. Para ello se hizo necesario estudiar las tendencias y tecnologías actuales, así como una extensa documentación relacionada con el tema de investigación. De esta forma, al concluir el trabajo, los autores arribaron a las siguientes conclusiones:

- El sistema desarrollado gestionará información sobre el estado y el rendimiento de los proyectos de la UCI durante el proceso de desarrollo de software.
- El sistema permitirá efectuar un seguimiento y control sobre los proyectos a fin de lograr el éxito en los mismos.
- Permitirá a los responsables evaluar el desempeño del proyecto y tomar decisiones efectivas sobre el mismo.
- Facilitará además una explotación más eficiente de los recursos asignados a la producción.

RECOMENDACIONES

- El uso de este material como bibliografía para investigaciones futuras en esta área de desarrollo.
- Continuar un estudio sobre el seguimiento y control de las actividades de los proyectos, con el objetivo de agregarle nuevas funcionalidades al sistema a medida que surjan las necesidades de los usuarios.
- Someter a prueba el sistema en un período de tiempo determinado, de forma tal que se pueda comprobar su desempeño y que las funcionalidades del sistema se correspondan con la necesidad del usuario.
- La implantación del sistema en todas las facultades de la Universidad, para proporcionar a las direcciones de la producción y a las facultades la gestión de la información del estado y el rendimiento de los proyectos durante su desarrollo.

BIBLIOGRAFÍA

ADOBE. 2008. Usos comunes de las aplicaciones Web. *Usos comunes de las aplicaciones Web*. [En línea] 2008. http://help.adobe.com/es_ES/Dreamweaver/10.0_Using/WS1356B4A6-E516-4621-9D1E-5F946C3C0E2B.html..

ÁLVAREZ, M. A. 2004. Introducción a PHP 5. *Introducción a PHP 5*. [En línea] 2004. <http://www.desarrolloweb.com/articulos/1696.php>.

APACHE. 2007. Apache. *Apache*. [En línea] 2007. from <http://httpd.apache.org/>..

BERTINO, E. A. y MARTINO, L. A. 1995. *Sistemas de bases de datos orientadas a objetos*. s.l. : Ediciones Díaz de Santos, 1995.

Boletín del Instituto Andaluz de Patrimonio Histórico. **BUSTELO RUESTA, CARLOTA y AMARILLA IGLESIAS, RAQUEL. 2001.** 34, s.l. : INFORAREA S.L, marzo de 2001, págs. 226-230.

COBO, ÁNGEL y GÓMEZ, PATRICIA. 2005. *PHP y MySQL- tecnologías para el desarrollo de aplicaciones web*. s.l. : Ediciones Díaz de Santos, 2005. 8479787066.

DAVIS, DUANE L. 2000. *Business Research for Decision Making, 5th ed.* s.l. : Duxbury Resource Center, 2000. 0534373976.

ESTEBAN ALBERDI, CRISTINA, y otros. 2007. *Gestión del tiempo y evolución en los usos del tiempo*. s.l. : Vision Libros, 2007. 8498215897.

ESTERKIN, JOSÉ. 2008. IAAP. *Qué es y cómo se hace un reporte de estado del proyecto*. [En línea] 2008. <http://iaap.wordpress.com/2008/06/13/que-es-y-como-se-hace-un-reporte-de-estado-del-proyecto/>.

FOUNDATION, ECLIPSE. 2004. About the Eclipse Foundation. *About the Eclipse Foundation*. [En línea] 2004. <http://www.eclipse.org/org/#Unique..>

FREDERICK, S. y RAMSAY, C. 2008. *Learning Ext JS*. s.l. : Packt Publishing Ltd, 2008.

FRESNO, V. D. 2006. *Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios*. Departamento de Ingeniería Telemática y Tecnología Electrónica, Universidad Rey Juan Carlos, Escuela Superior de Ciencias Experimentales y Tecnología. España : s.n., 2006. Tesis Doctoral.

GAMMA, ERICH, y otros. 1995. *Design Patterns. Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wesley, 1995.

GÓMEZ SÁNCHEZ S, RUBÉN. 2007. *Necesidad de lograr proyectos exitosos.* 2007.

Gutmans, Andi, Sæther Bakken, Stig y Rethans, Derick. 2004. *PHP 5 Power Programming.* 2004. 0-131-47149-X.

HIDALGO, M. 2001. SQL MAX Connections. *SQL MAX Connections.* [En línea] 2001. http://www.sqlmax.com/reportin_services1.asp..

HUMPHREY, W. S. 1999. *Introduction to the team software process(sm).* s.l. : Addison-Wesley, 1999.

JACOBSON, IVAR, BOOCH, GRADY y RUMBAUGH, JAMES. 2000. *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison-Wesley, 2000. 0-201-89535-8.

LARMAN, C. 1999. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos.* s.l. : PRENTICE HALL, 1999. 970-17-0261-1.

LARMAN, CRAIG. 1999. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos.* s.l. : PRENTICE HAL, 1999. 970-17-0261-1.

LLORENS FABREGAS, JUAN. 2005. *Gerencia de proyectos de tecnología de información: Cómo organizar, planificar, estimar, evaluar y controlar exitosamente proyectos de tecnología de información. Nociones y criterios fundamentales para la gestión de riesgos, alcance y calidad.* s.l. : El Nacional, 2005. 9803881868.

LOCKHART, T. 1996. *Tutorial de PostgreSQL.* 1996.

MANSO, MARYLYN. 2000. *Aplicación del Formato Dublin Core para la descripción de los recursos en el Sistema de Gestión Multimedia desarrollado en el CDICT- UCLV.* 2000.

MEYER, E. A. 2001. *Cascading Style Sheets 2.0 Programmer's Reference.* 2001.

MIKKONEN, T. y TAIVALSAARI, A. 2007. *Using JavaScript as a Real Programming Language.* s.l. : Sun Microsystems , 2007.

OROZCO, S. 2009. LíderDeProyecto.com. *El seguimiento del proyecto: En Dios confiamos, los demás traigan sus datos.* [En línea] 2009. http://www.liderdeproyecto.com/articulos/el_seguimiento_del_proyecto.html..

PARADIGM, V. 2007. Visual Paradigm for UML. *Visual Paradigm for UML.* [En línea] 2007. http://206.222.18.10/media/documents_download/vpuml61ug1/vpuml_user_guide.pdf..

PÉREZ, J. E. 2009. Introducción a AJAX. *Introducción a AJAX.* [En línea] 2009. http://www.sqlmax.com/reportin_services1.asp..

PETKOVIĆ, DUŠAN. 2005. *Microsoft SQL Server 2005: a beginner's guide.* s.l.: McGraw-Hill Professional, 2005. 0072260939.

PMI. 2004. *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®).* s.l.: Project Management Institute, Inc, 2004.

PONJUÁN DANTE, GLORIA. 2004. *Gestión de información: Dimensiones e implementación para el éxito organizacional.* s.l.: Nuevo Paradigma, 2004. 9879653661.

POSTGRESQL. 2006. What is PostgreSQL? *What is PostgreSQL?* [En línea] 2006. <http://www.postgresql.org/docs/8.2/interactive/intro-what-is.html>.

RODRÍGUEZ, JOSÉ RAMÓN, GARCÍA MÍNGUEZ, JORDI y LAMARCA OROZCO, IGNACIO. 2007. *Gestión de proyectos INFORMATICOS/ Information Project Management: Metodos, herramientas y casos/ Methods, Tools And Cases.* s.l.: Editorial UOC, 2007.

SÁNCHEZ, M. A. M. 2004. *Metodologías De Desarrollo De Software.* 2004.

SERER FIGUEROA, MARCOS. 2001. *Gestión integrada de proyectos.* s.l.: Edicions UPC, 2001. 848301453X.

TURNER, JOHN RODNEY. 1993. *The Handbook of Project-based Management: Improving the Processes for Achieving Strategic Objectives.* s.l.: Illustrated, 1993. 0077076567.

USERO, J. A. M. 2007. *Nuevas tecnologías para nuevas bibliotecas: Desarrollo de servicios de información electrónica.* s.l.: Alfagrama Ediciones, 2007.

W3C. 1997. What is HTML. *What is HTML.* [En línea] 1997. <http://www.w3.org/TR/html401/intro/intro.html#h-2.2>.

ZANINOTTO, F. y POTENCIER, F. 2009. *librosweb.es. Symfony 1.1, la guía definitiva.* [En línea] 2009. http://www.librosweb.es/symfony_1_1.

ANEXOS

Anexo 1: Plantilla de Reporte de Estado

Reporte de Estado del Proyecto

Fecha del reporte:

Proyecto:

Facultad:

Polo Productivo:

Preparado Por:

Encargado del documento	Rol que desempeña

Control del Reporte de Estado

Versión	Fecha	Autor	Descripción de cambio
1.0	[dd/mm/aa]	[Encargado del documento]	Documento creado
[Versión #]	[dd/mm/aa]	[Encargado del cambio]	<ul style="list-style-type: none"> • [Cambio 1] • [Cambio 2] • [Cambio n]

TABLA DE CONTENIDOS

1	OBJETIVO DEL REPORTE DE ESTADO.....	¡Error!
	Marcador no definido.	
2	REPORTE DE ESTADO.....	¡Error!
	Marcador no definido.	
	2.1 Resumen Ejecutivo del Reporte de Estado.....	¡Error!
	Marcador no definido.	
	2.2 Plantilla del Reporte de Estado.....	4
3	APROBACIONES DEL REPORTE DE ESTADO.....	6

Objetivo del Reporte de Estado

[Reemplazar este texto con un resumen del objetivo del reporte referente a las necesidades específicas del proyecto]

Reporte de Estado

Resumen Ejecutivo del Reporte de Estado

[Reemplazar este texto con detalles del estado del proyecto para sus principales jefes, centrando en la entrega de los hitos. Este Informe debe incluir los siguientes puntos:]

- [Estado del proyecto]
 - [Descripción del proyecto]
 - [Hitos a entregar para el último período reportado]
 - [Éxitos o fracasos del proyecto en la entrega de los hitos en el periodo actual del proyecto]
- [Informe de presupuesto—Escribe con respecto a gasto planeado, déficit y gasto real/superávit]
- Informe de Administración de riesgos—Especifica cualquier cambio en los riesgos más importantes identificados desde el informe anterior y cualquier modificación en las estrategias para abordarlos, si lo considera necesario.
- [Informe de tareas—Incluye las tareas del proyecto, el estado en que se encuentran, la fecha prevista y la solución de las mismas.]

Plantilla del Reporte de Estado

Nombre del Proyecto:		
Preparado Por: [Nombre]	Fecha: [dd/mm/aa]	Período del reporte: [dd/mm/aa] a [dd/mm/aa]

Estado total del proyecto:			
[Reemplaza este texto con una breve declaración de la situación del proyecto]			
Resumen del proyecto:			
[Reemplazar este texto con una breve descripción de los resultados del proyecto no incluidos en el resto del informe]			
Hitos a entregar en el último período			
Resultados (Entregables)	Fecha Prevista	% Completado	Estado de la Entrega
Hito 1			
• [Entrega 1]	[dd/mm/aa]	[n%]	[En tiempo]
• [Entrega 2]	[dd/mm/aa]	[n%]	[Adelantado]
• [Entrega n]	[dd/mm/aa]	[n%]	[Retrasado]
Hito 2			
• [Entrega 1]	[dd/mm/aa]	[n%]	[En tiempo]
• [Entrega 2]	[dd/mm/aa]	[n%]	[Adelantado]
• [Entrega n]	[dd/mm/aa]	[n%]	[Retrasado]
Hitos a entregar en el próximo período			
Resultados (Entregables)	Fecha Prevista	% Completado	Estado de la Entrega

Hito 1				
• [Entrega 1]	[dd/mm/aa]	[n%]	[En tiempo]	
• [Entrega 2]	[dd/mm/aa]	[n%]	[Adelantado]	
• [Entrega n]	[dd/mm/aa]	[n%]	[Retrasado]	
Hito 2				
• [Entrega 1]	[dd/mm/aa]	[n%]	[En tiempo]	
• [Entrega 2]	[dd/mm/aa]	[n%]	[Adelantado]	
• [Entrega n]	[dd/mm/aa]	[n%]	[Retrasado]	
Afectaciones sobre el proyecto				
[Cambios sobre hitos o alcance del proyecto]		[Breve descripción sobre esos cambios]		
Presupuesto del Proyecto/Estado Financiero				
Elemento del Presupuesto	Presupuesto Planeado	Costo Actual	Varianza/Explicación	
Estado de los Riesgos del Proyecto				
Riesgos y	Probabilidad	Impacto de	Prioridad	Cambio desde la

Descripción	de Riesgo	Riesgo	de los Riesgos	última revisión
Descripción del riesgo <ul style="list-style-type: none"> [Elemento] [Elemento] 	[Alto/Medio/Bajo]	[Alto/Medio/Bajo]	[Alto/Medio/Bajo]	[Descripción]
Descripción del riesgo <ul style="list-style-type: none"> [Elemento] [Elemento] 	[Alto/Medio/Bajo]	[Alto/Medio/Bajo]	[Alto/Medio/Bajo]	[Descripción]

Estado de las tareas del Proyecto

Tarea y Descripción	Impacto en el proyecto	Fecha Prevista	Estado de la tarea	Solución de la tarea
[Descripción de la tarea] <ul style="list-style-type: none"> [Elemento] [Elemento] 	[Alto/Medio/Bajo]	[dd/mm/aa]	[Abierto/Cerrado]	[Descripción]
[Descripción de la tarea] <ul style="list-style-type: none"> [Elemento] [Elemento] 	[Alto/Medio/Bajo]	[dd/mm/aa]	[Abierto/Cerrado]	[Descripción]

APROBACIONES DEL REPORTE DE ESTADO

Preparado por _____

Jefe del Proyecto

Aprobado por _____

Gerente del Proyecto

Director

Cliente Patrocinador



Anexo 2: Plantilla de Reporte Postmortem

[Nombre del Proyecto]

[Redactado por:]

Ejecución Contra Objetivos

- Objetivo: objetivos iniciales del proyecto u objetivos del proyecto
 - Lista de métricas claves (elementos para medir éxito)

Ejecución Contra Programa

- Plan: resumen del programa original de proyecto
 - Lista de hitos claves

Ejecución Contra Calidad

- Objetivo de calidad: objetivos iniciales de la calidad del proyecto
 - Lista de métricas claves (elementos para medir éxito)

Ejecución Contra Presupuesto

- Presupuesto: objetivos iniciales del presupuesto del proyecto
 - Lista de costos principales por objetivos, límites de gasto

Reporte Por Áreas

Planificación de proyecto

¿Cómo estaba planeado el proyecto?

- ¿Quién era el responsable de la planificación original?
 - ¿Cómo hizo este trabajo? ¿Participaron las personas adecuadas?
- ¿Estaba el proyecto bien definido desde el principio?
 - ¿Había un plan escrito real?

- ¿Cómo era distribuido el plan de proyecto?
- ¿Cómo funcionó el área de planificación?

¿Era un cronograma correcto?

- ¿Era un buen cronograma?
 - ¿Qué estaba bien? ¿Qué faltó?
- ¿Era el cronograma realista?
- ¿Cómo evolucionó el cronograma con el tiempo?
 - ¿El cambio fue bueno o malo?
 - ¿Cómo estos cambios afectaron al proyecto?
- Áreas claves para mejorar:
 - Hacer recomendaciones muy específicas.

Investigación y Desarrollo

¿Cómo era I+D Dirigido?

- ¿Qué papel jugaba I+D dentro del proyecto?
 - ¿Cuántos equipos, número de personas, estructuras de supervisión, etc.?
 - ¿Cuán bien funcionó? ¿Mejoras posibles?
- ¿Cuán bien se comunicaban los equipos de I+D?
 - Qué métodos, tiempos, etc.
 - ¿Cuán bien funcionó la comunicación?

¿Cuán Efectiva fue el área de Investigación y Desarrollo?

- Identificando y solucionando problemas técnicos
 - ¿Se identificaban a tiempo los problemas?
 - ¿Las respuestas a los problemas eran correctas?
 - ¿Qué funcionó? ¿Qué no funcionó? ¿Pudo ser mejor?
- Estimación & ejecución
 - ¿Se cumplieron los estimados?
 - ¿Qué ayudó a estimar correctamente?
 - ¿Qué provocó que se estimara incorrectamente?

Gestión del Proyecto

¿Cómo era dirigido el proyecto?

- Reuniones: quién/cuándo/con qué frecuencia
- Comunicación: quién/cuándo/con qué frecuencia
 - ¿Cuán bien funcionó?
- Cambios: ¿cómo eran seguidos, comunicados?
 - ¿Cuán bien funcionó?
- Otros métodos: email, programas, bases de datos, informes, etc.

Aseguramiento de la Calidad & Soporte

- ¿Cómo se medía la calidad del producto?
 - ¿Era efectivo? ¿Eficiente?
- ¿Cómo se compara el producto obtenido con los objetivos planteados?
- ¿Se solucionaron los problemas de calidad?
- ¿Los equipos de soporte estaban correctamente preparados?
- ¿Es la calidad del producto compatible con los recursos de soporte?

Comercialización

- ¿Se realizó un posicionamiento ajustado al producto?
- ¿Fue exitosa? ¿Apropiada? ¿Efectiva?
- ¿Fue efectivo el lanzamiento del producto?
- ¿Los programas de comercialización estaban eficazmente implementados?
- ¿Estaban el producto y su lanzamiento en correspondencia con los objetivos de comercialización?

Ventas

- ¿Estaban los canales de comunicación al cliente y los equipos de venta adecuadamente informados acerca del producto?
- ¿Estaban el producto y la comunicación en función de las necesidades del cliente?
- ¿Tiempos de entrega adecuados? ¿Costos?

- ¿Cómo se comparan las ventas con los objetivos?
- ¿Cómo fue recibido el producto?

Lecciones aprendidas

Qué Estuvo Bien

- Enumere por puntos aquellas cosas que estuvieron bien.
 - Uso específico de ejemplos: “Fue muy útil la reunión diaria. matutina de 15 minutos” en lugar de “buena comunicación en el equipo”.
 - Procedimientos, informes, etc. que fueron particularmente útiles.

Qué Estuvo Mal

- Enumere por puntos aquellas cosas que estuvieron mal.
 - Trate de aislar actitudes específicas, procedimientos, métodos, etc. que causaron problemas.
- ¿Cómo el equipo respondió ante los problemas?

Recomendaciones

- Registro de recomendaciones críticas por área para productos futuros de este tipo.
- Recomendaciones acerca de la distribución de este documento y sobre las presentaciones que de este se deben hacer

Preguntas & Comentarios

Anexo 3: Encuesta sobre los Reporte de Estado y Reporte Postmortem realizada a las facultades

1. Alcance del proyecto: Informatización UCI Nacional Internacional
2. ¿Usted realiza un Reporte de Estado de su proyecto y lo envía a sus superiores? Si No
 - 2.1. ¿A quién(es) le entrega ese Reporte de Estado del Proyecto?

 - 2.2. ¿Con qué frecuencia lo realiza?
 1 semana 15 días 1 mes otro (especifique)

 - 2.3. Especifique el medio en que lo hace:
 Documento enviado por correo.
 Sistema en línea desarrollado con ese propósito.
 Documento depositado en sistema de gestión documental o repositorio.
 - 2.4. De los siguientes campos, diga cuales están presentes en su informe de reporte:
 Próximos Hitos o entregas (Resultados, Fecha Prevista, % completado).
 Afectaciones sobre el proyecto.
 Estado del Presupuesto del Proyecto o Estado Financiero.
 Estado de los Riesgos sobre Proyecto (Riesgos y Descripción, Probabilidad, Impacto, otros).
 Estado de las tareas del Proyecto (Tarea y Descripción, Fecha Prevista, Estado, otros).
3. Diga qué campos usted cree que deben estar presentes en el Reporte de Estado de un proyecto:
 - a) _____

b) _____

c) _____

d) _____

4. ¿Ha elaborado y entregado alguna vez un reporte al concluir su función al frente de un proyecto?

___ Si ___ No

4.1. Diga qué campos usted cree que deben estar presentes en el mismo:

a) _____

b) _____

c) _____

d) _____

4.2. ¿A quién(es) le entregó ese Reporte?

4.3. Especifique el medio en que lo hizo:

___ Documento enviado por correo.

___ Sistema en línea desarrollado con ese propósito.

___ Documento depositado en sistema de gestión documental o repositorio.

5. De los siguientes campos, diga cuáles estaban presentes en su informe

a) ___ Objetivos iniciales del proyecto u objetivos del proyecto.

b) ___ Plan (resumen del programa original de proyecto).

c) ___ Objetivo de calidad (objetivos iniciales de la calidad del proyecto).

d) ___ Presupuesto (objetivos iniciales de la calidad del proyecto).

- e) ____ Planificación de proyecto (¿Cómo estaba planeado el proyecto?, ¿Era un cronograma correcto?).
- f) ____ Investigación y Desarrollo (¿Cómo era I+D Dirigido?, ¿Cuán Efectiva fue el área de Investigación y Desarrollo?).
- g) ____ Gestión del Proyecto (¿Cómo era dirigido el proyecto?).
- h) ____ Aseguramiento de la Calidad & Soporte.
- i) ____ Comercialización.
- j) ____ Ventas.
- k) ____ Lecciones aprendidas (Qué Estuvo Bien, Qué Estuvo Mal).

Anexo 4: Plantilla propuesta por ALBET

RESUMEN SEMANAL DEL PROYECTO

Fecha

2009-06-11

Proyecto:

Cierre: jueves, 11 de junio de 2009 Hora: 4:58:45

Ministerio

Proyectos VIII Comisión Mixta

Resumen
del
Proyecto

- Incidencias:
- Temas pendientes:
- Cumplimiento de Hitos:
- Actividades realizadas y otras observaciones:

Área
Económica
del
Proyecto.

○

Anexo 5: Diagramas de clases del análisis

Diagrama de clases del análisis del CU: Autenticar Usuario

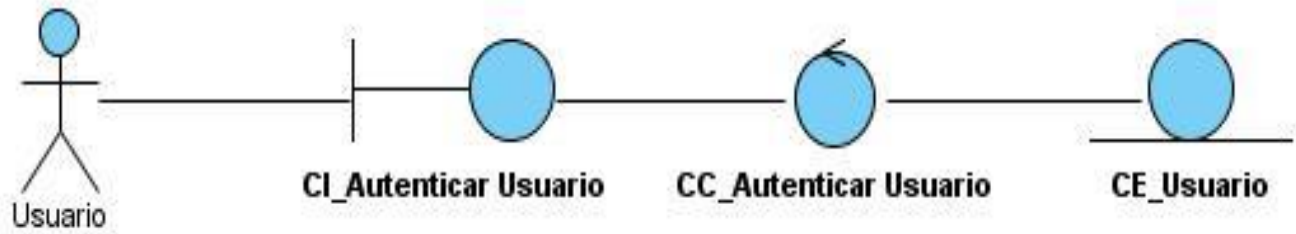


Diagrama de clases del análisis del CU: Gestionar Área

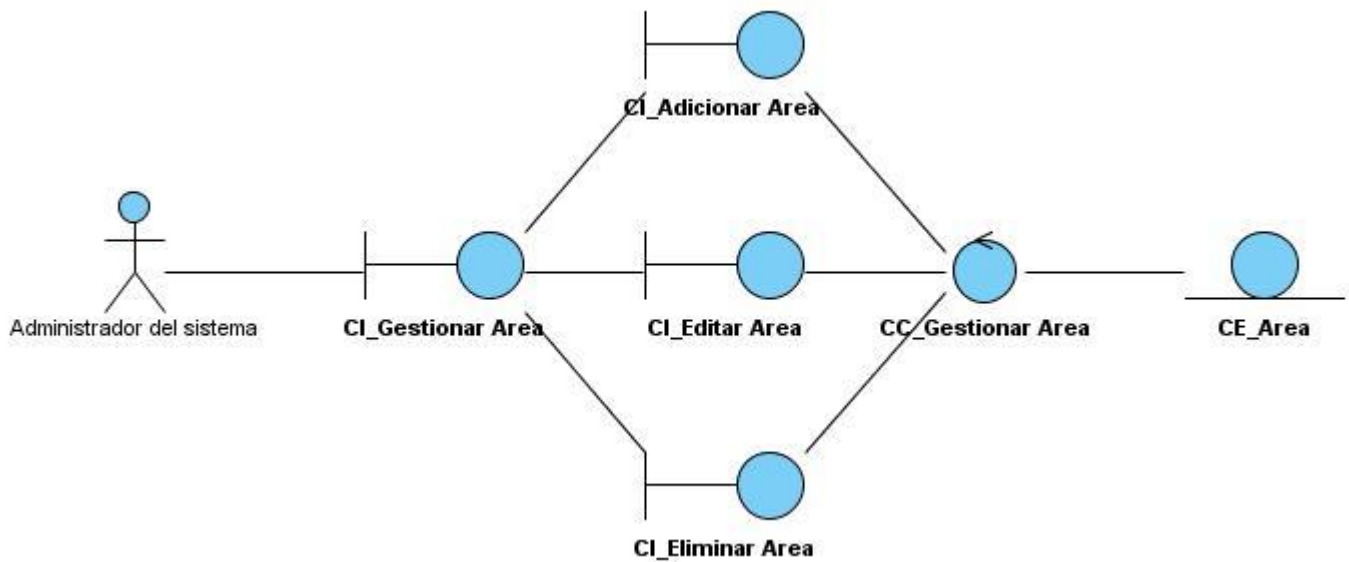


Diagrama de clases del análisis del CU: Gestionar Polo

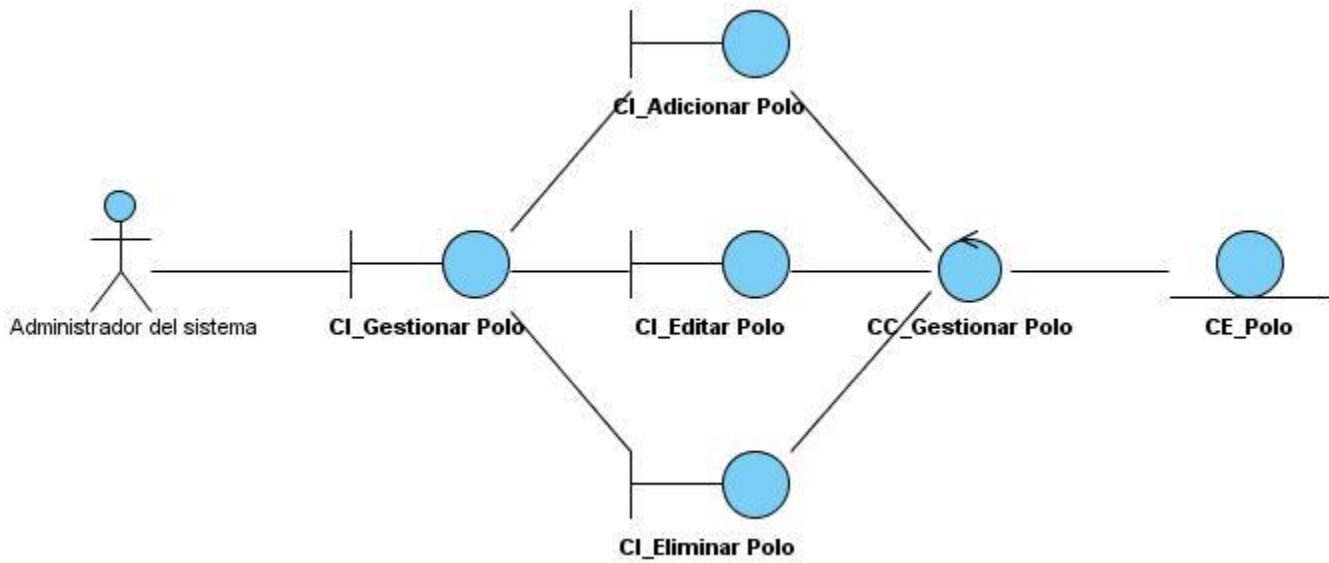


Diagrama de clases del análisis del CU: Gestionar Proyecto

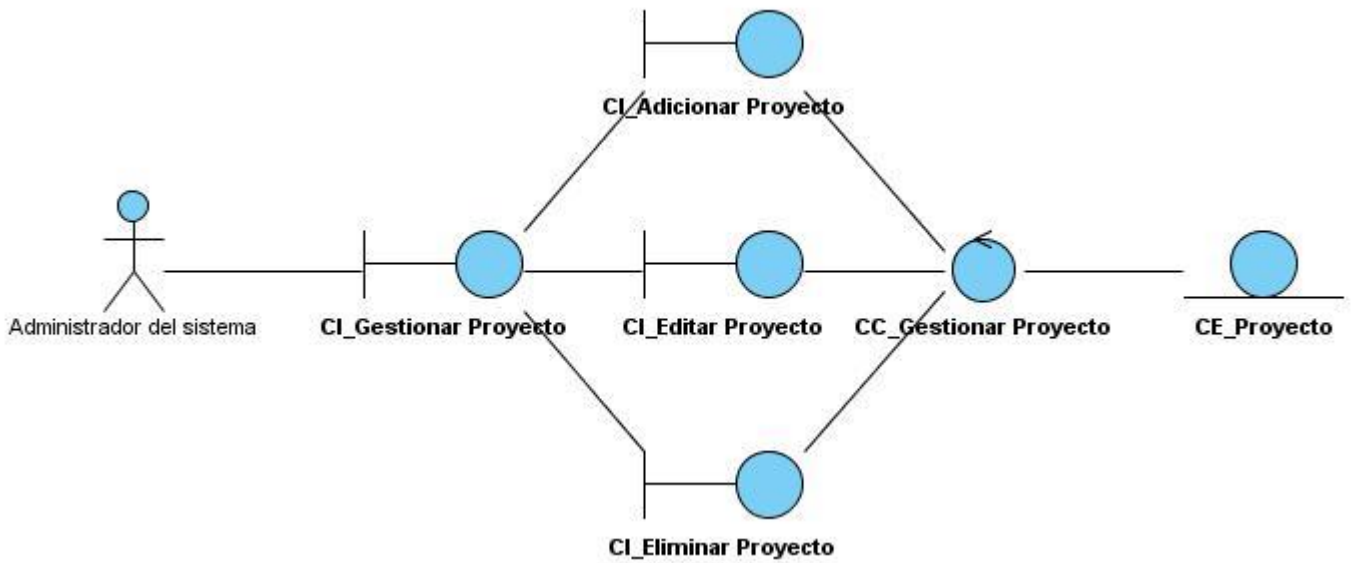


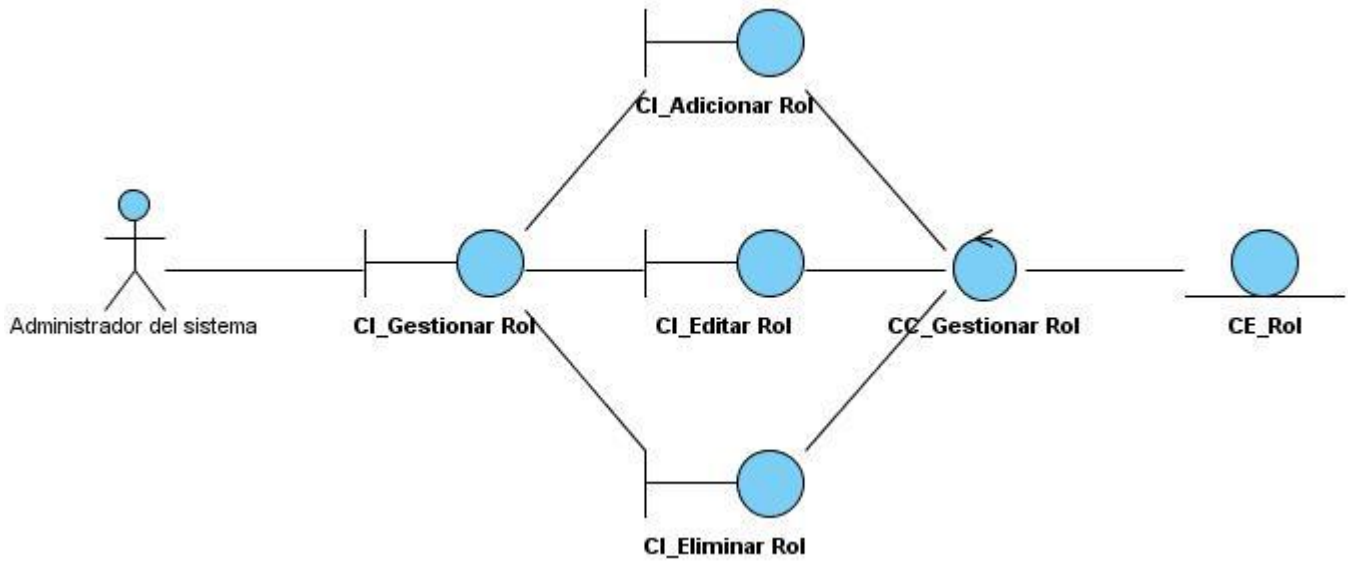
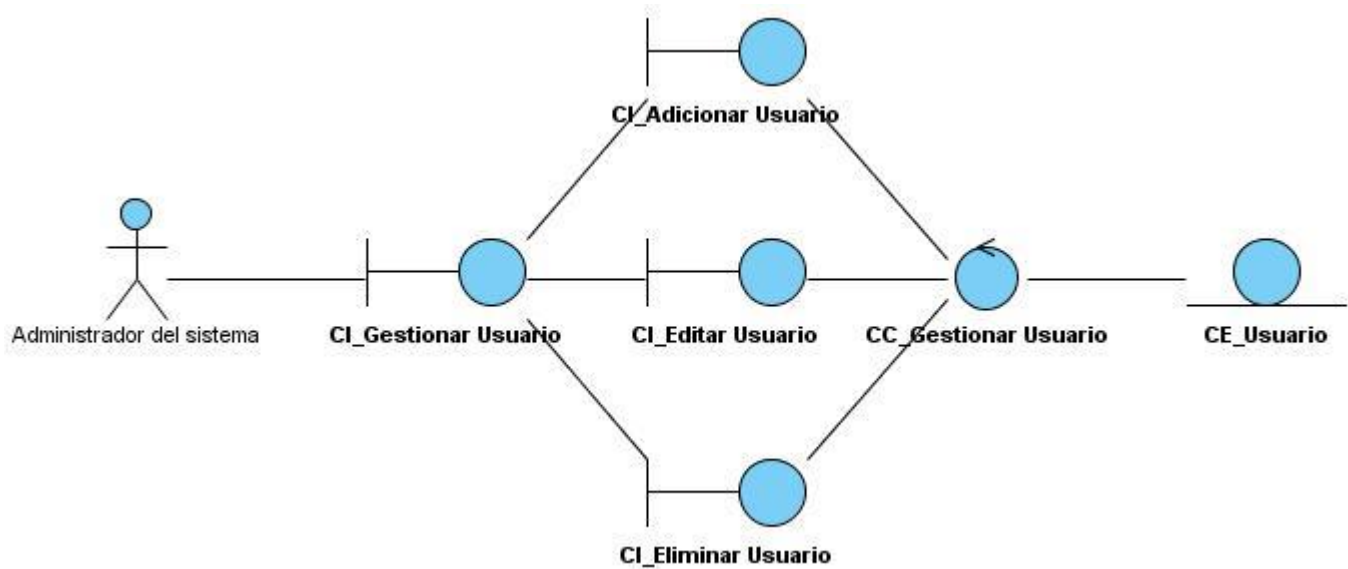
Diagrama de clases del análisis del CU: Gestionar Rol**Diagrama de clases del análisis del CU: Gestionar Usuario**

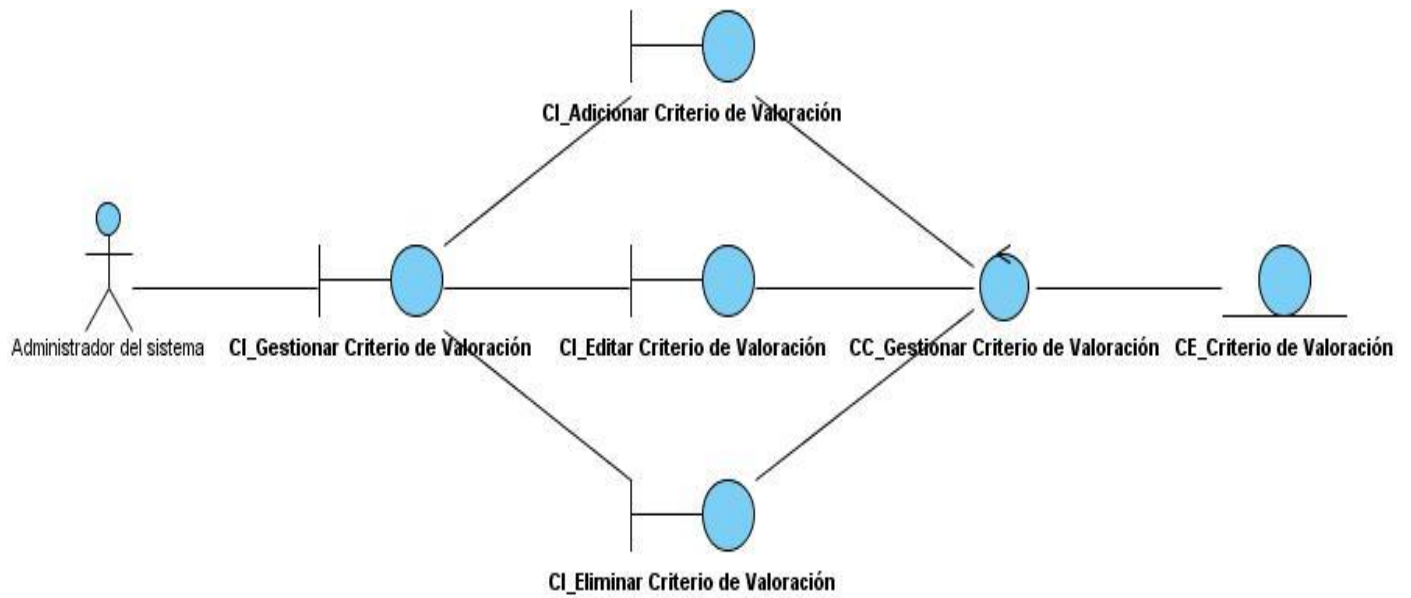
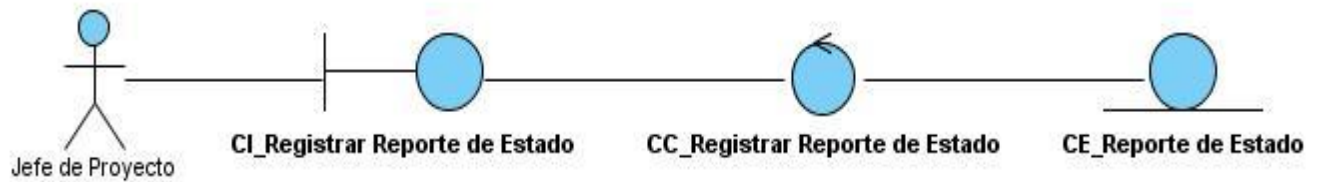
Diagrama de clases del análisis del CU: Gestionar Criterio de Valoración**Diagrama de clases del análisis del CU: Registrar Reporte de Estado**

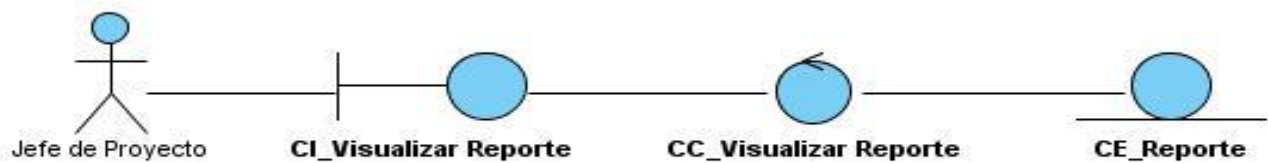
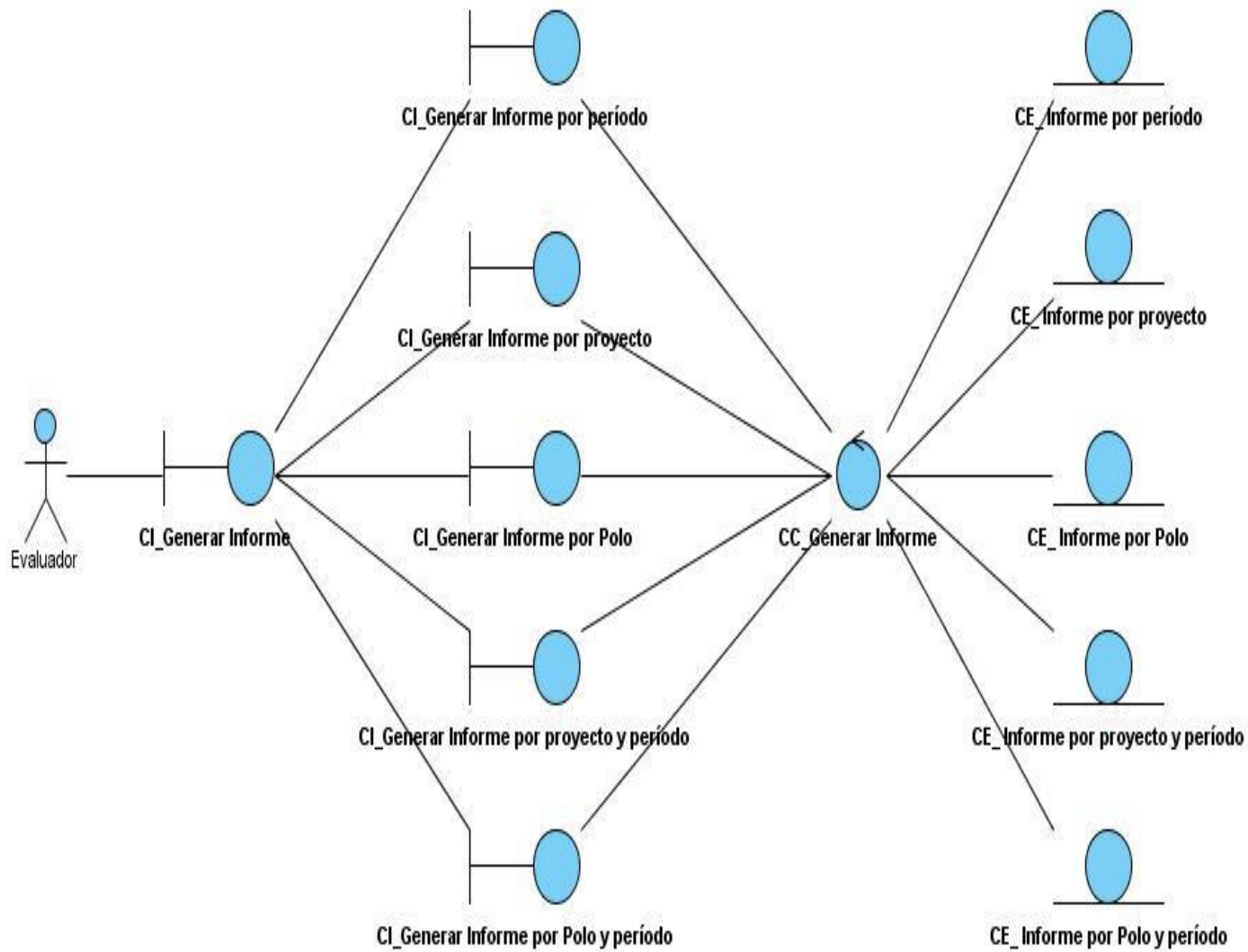
Diagrama de clases del análisis del CU: Registrar Reporte Postmortem**Diagrama de clases del análisis del CU: Emitir Criterio de Valoración****Diagrama de clases del análisis del CU: Visualizar Reporte**

Diagrama de clases del análisis del CU: Generar Informe



Anexo 6: Diagramas de clases del diseño

Diagrama de clases del diseño del CU: Autenticar Usuario

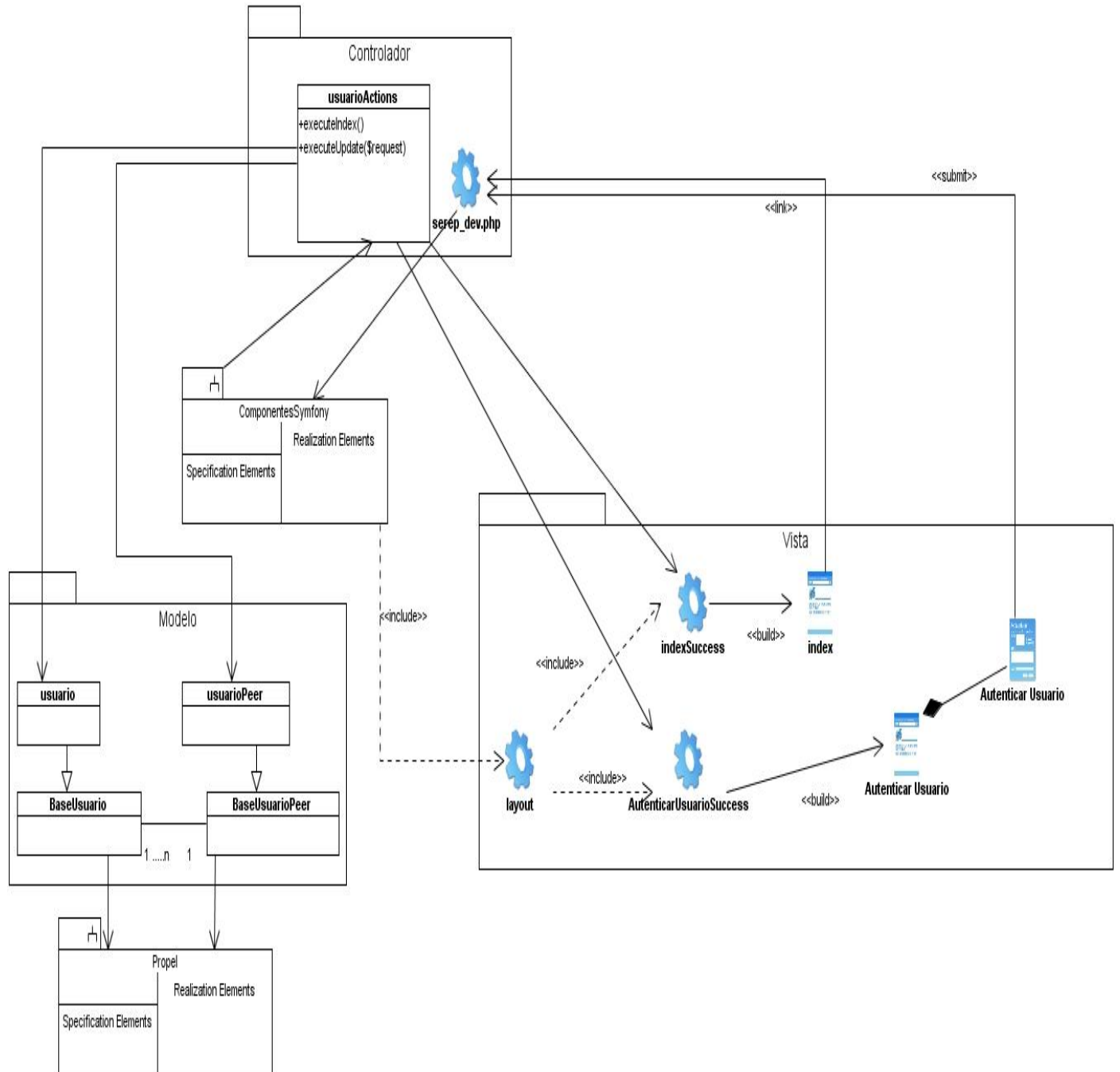


Diagrama de clases del diseño del CU: Gestionar Área

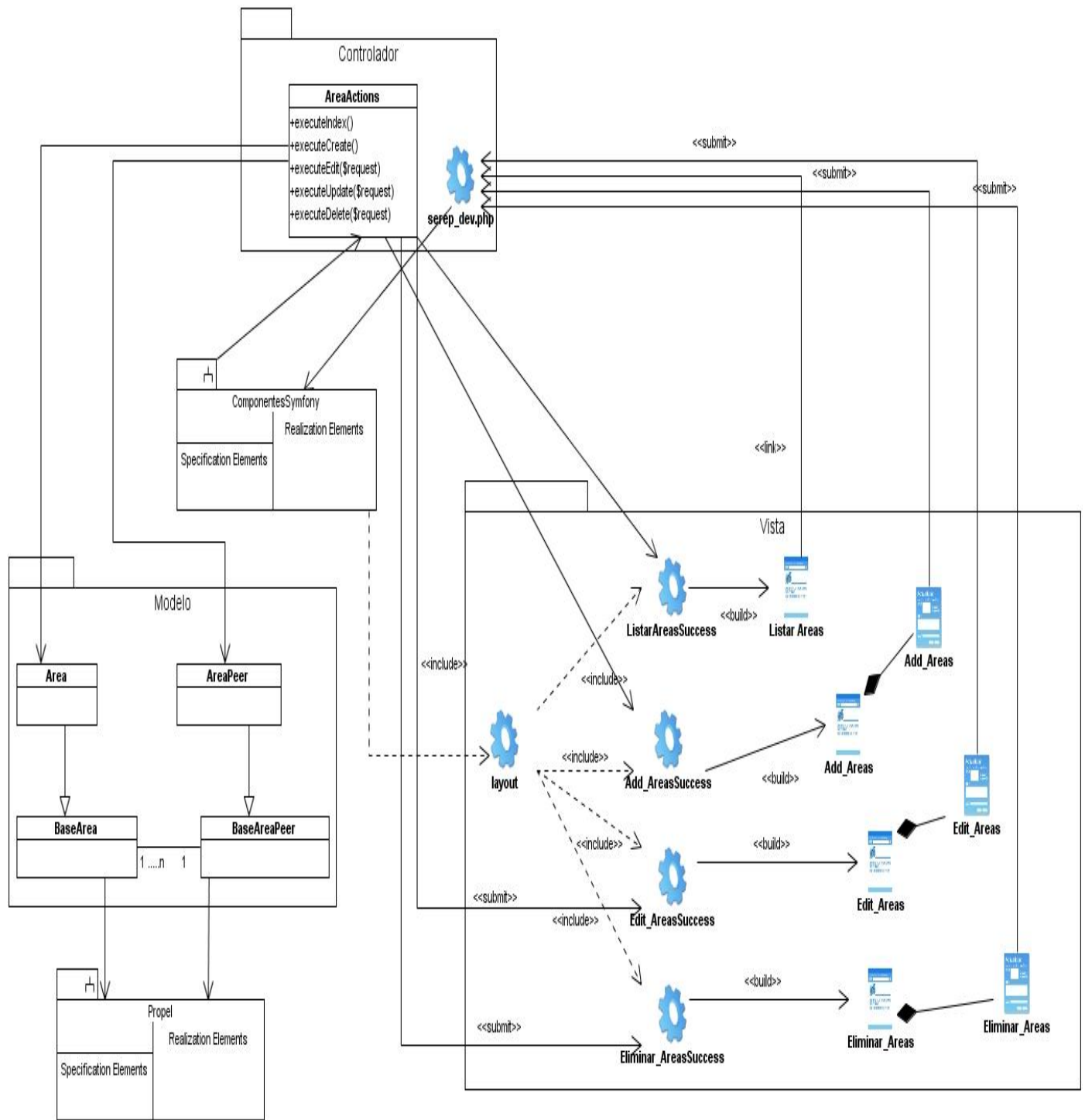


Diagrama de clases del diseño del CU: Gestionar Polo

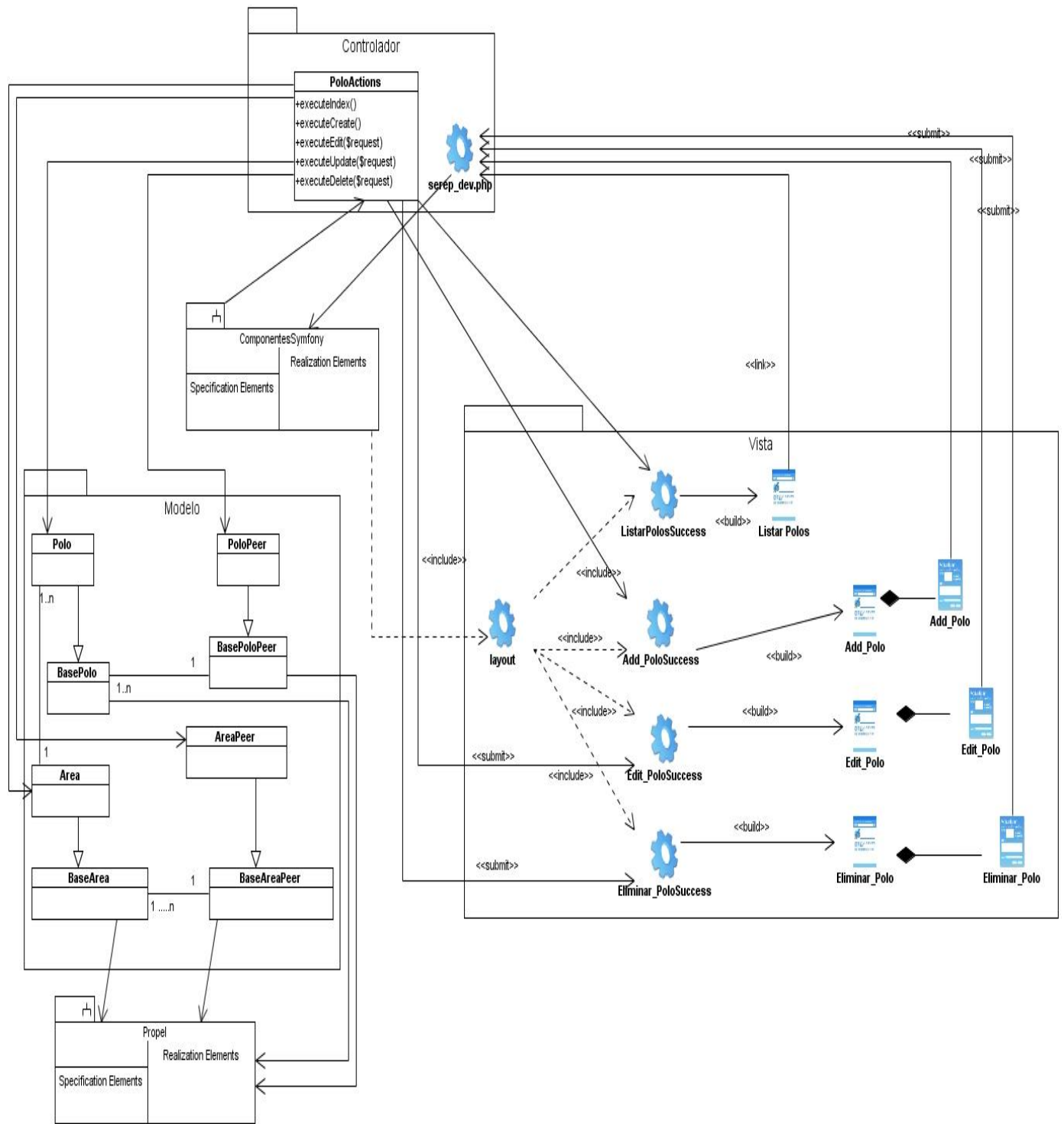


Diagrama de clases del diseño del CU: Gestionar Proyecto

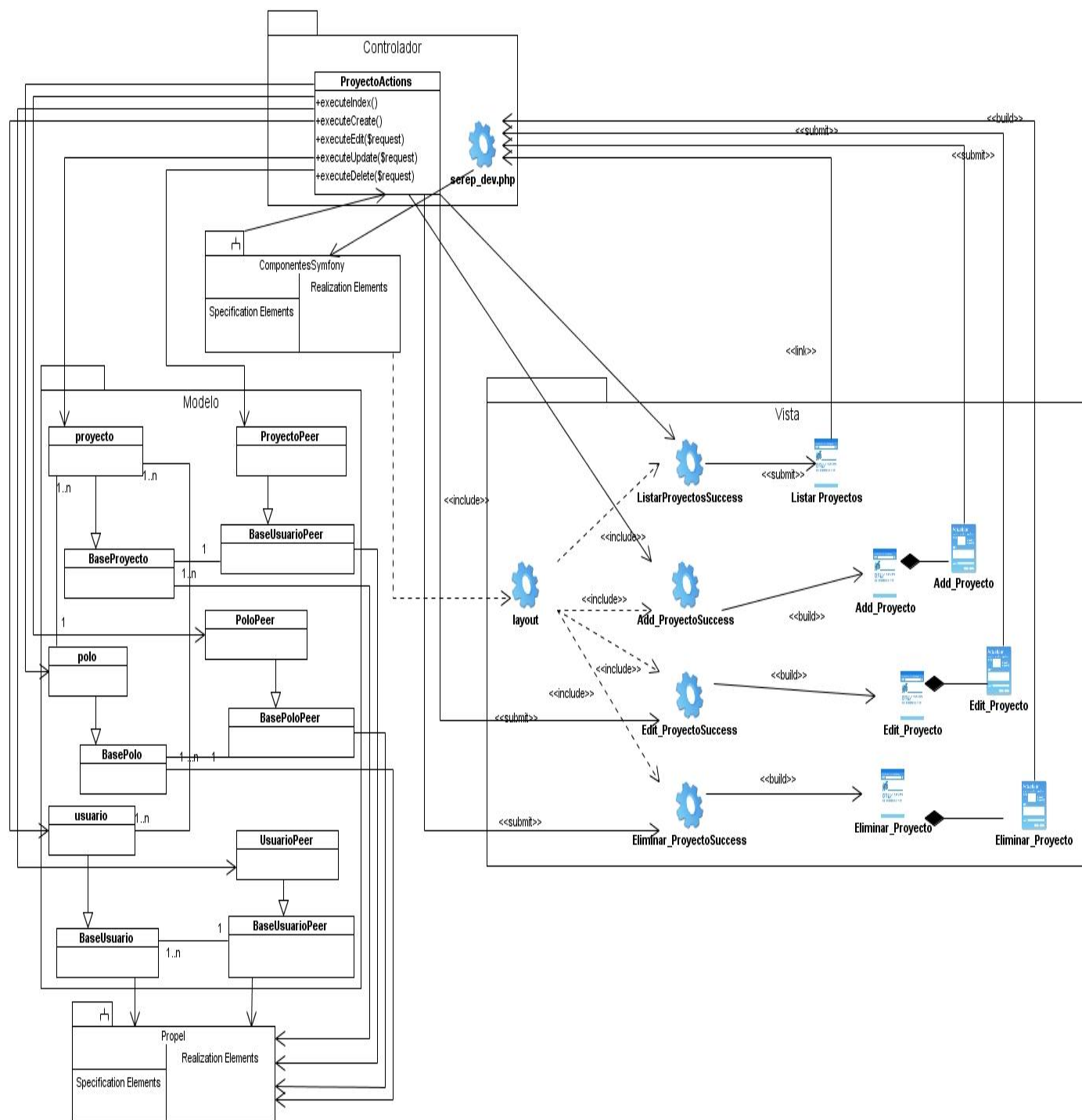


Diagrama de clases del diseño del CU: Gestionar Rol

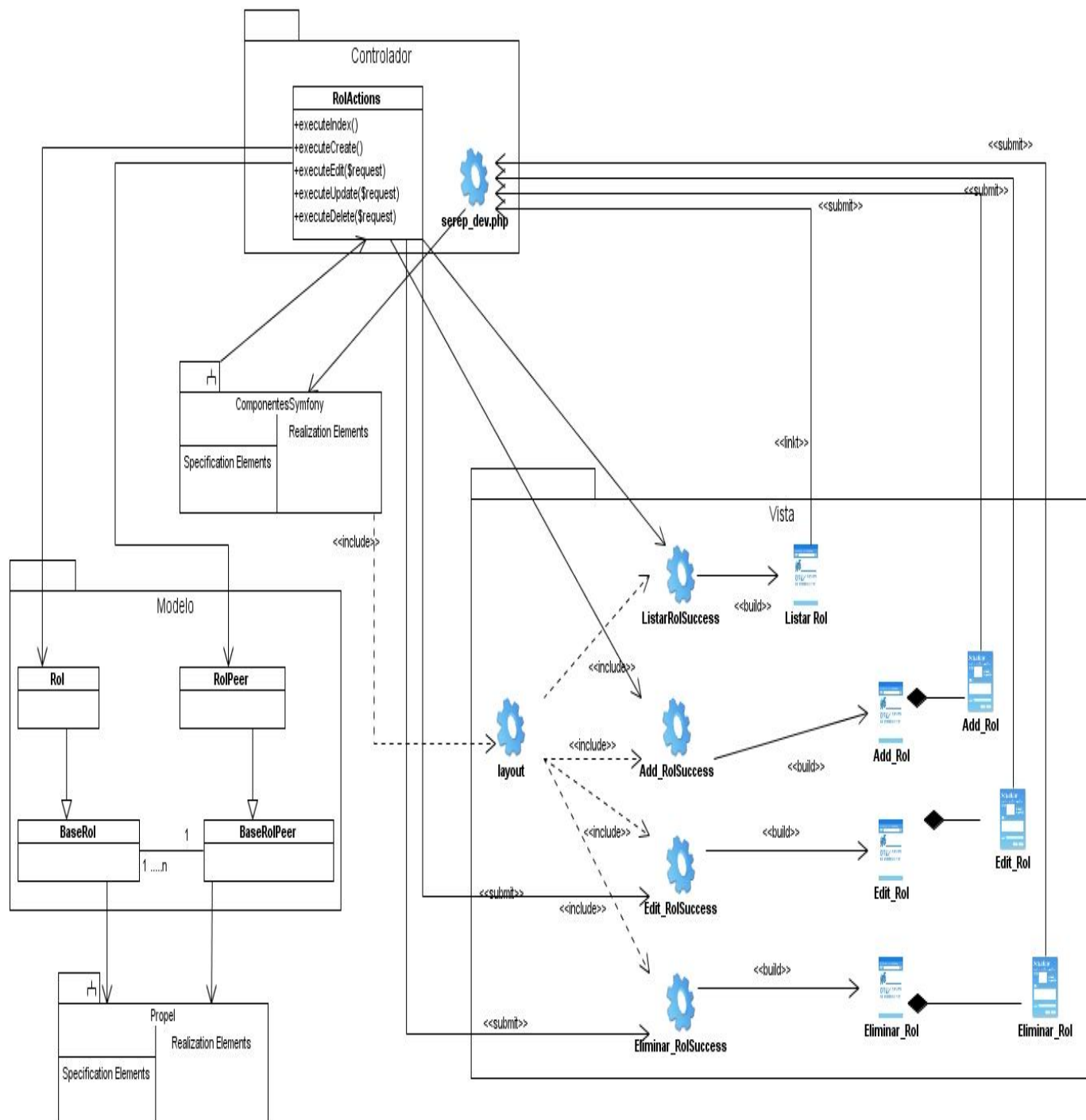


Diagrama de clases del diseño del CU: Gestionar Usuario

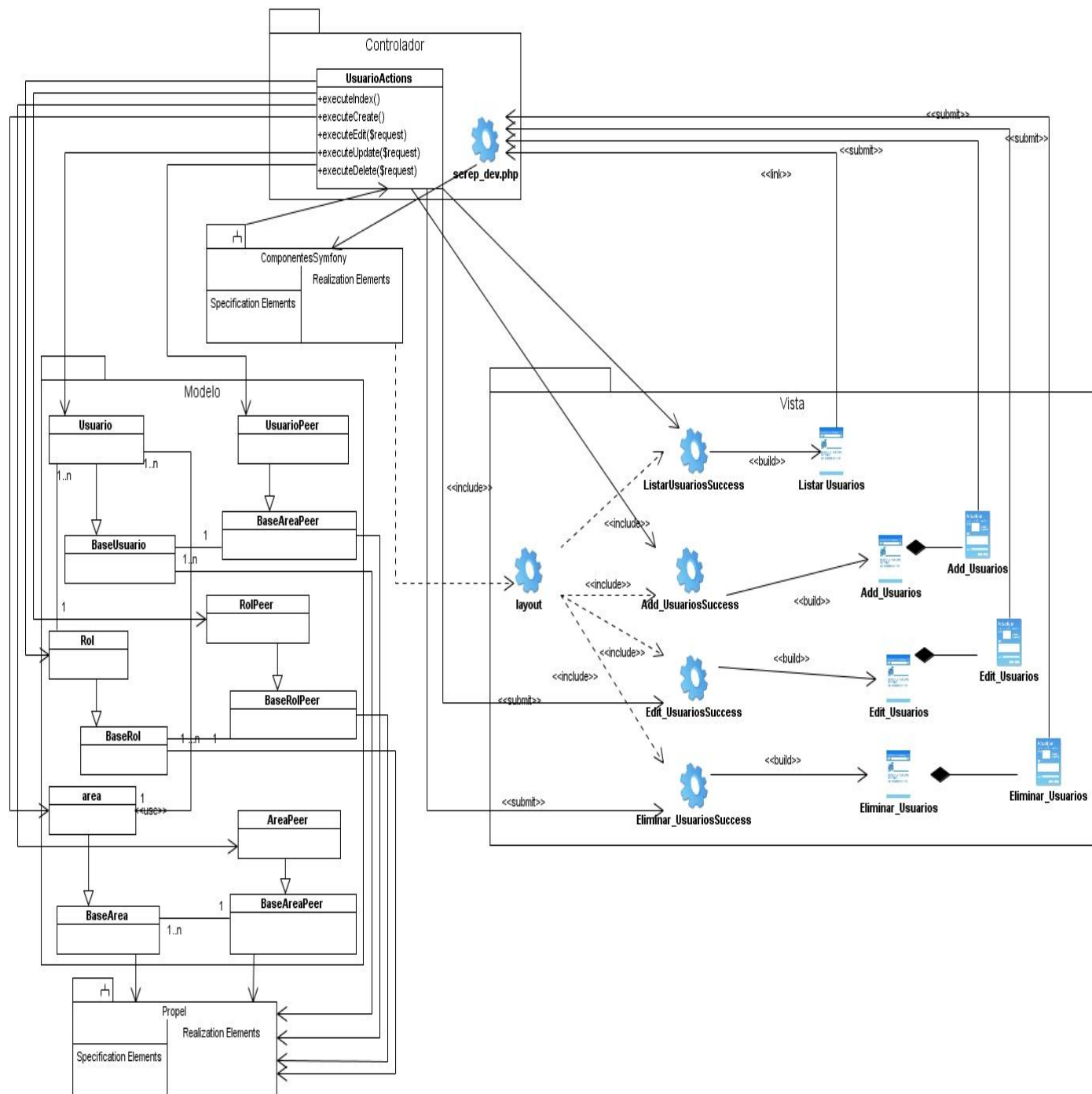


Diagrama de clases del diseño del CU: Gestionar Criterio de Valoración

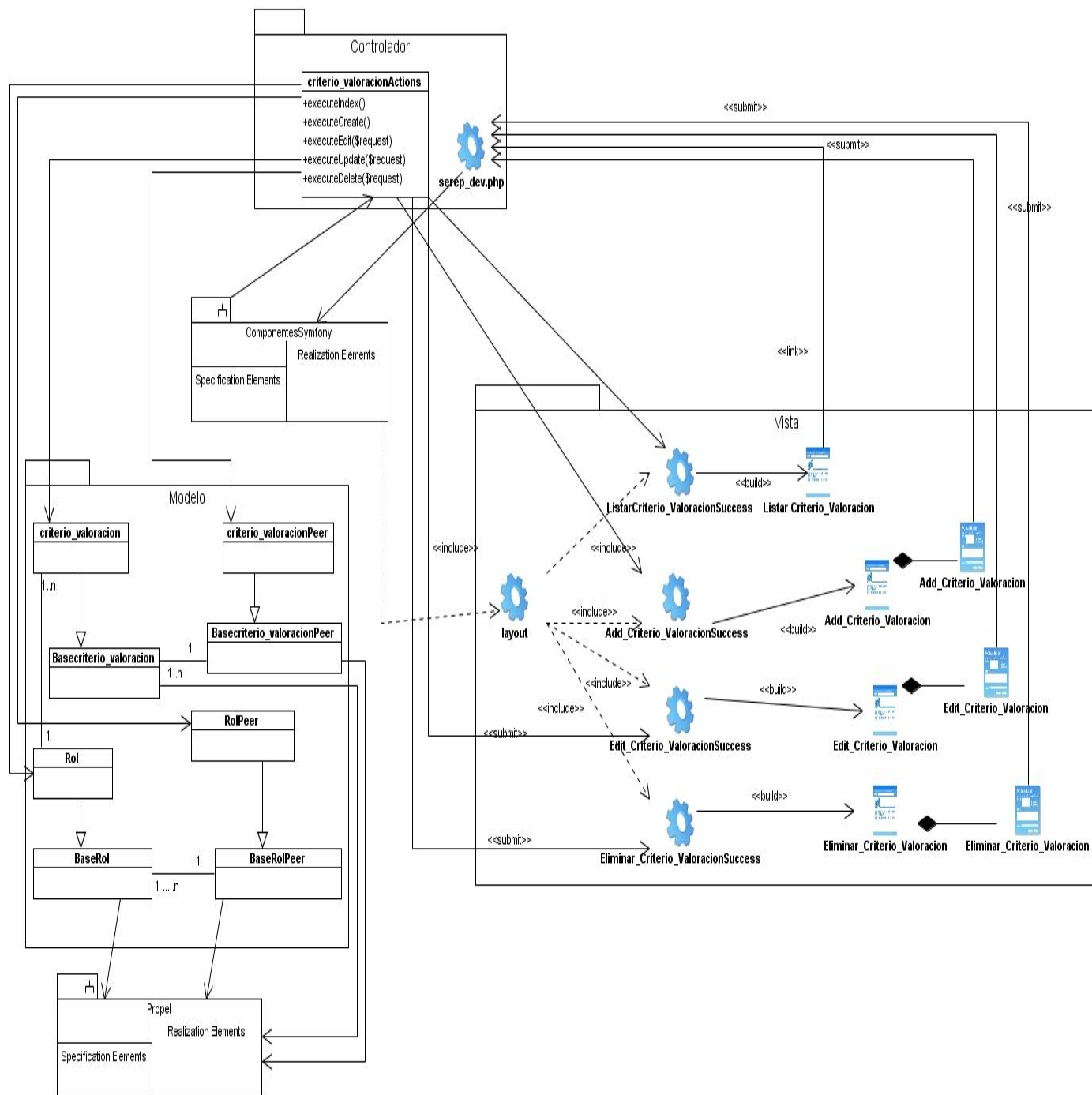


Diagrama de clases del diseño del CU: Registrar Reporte de Estado

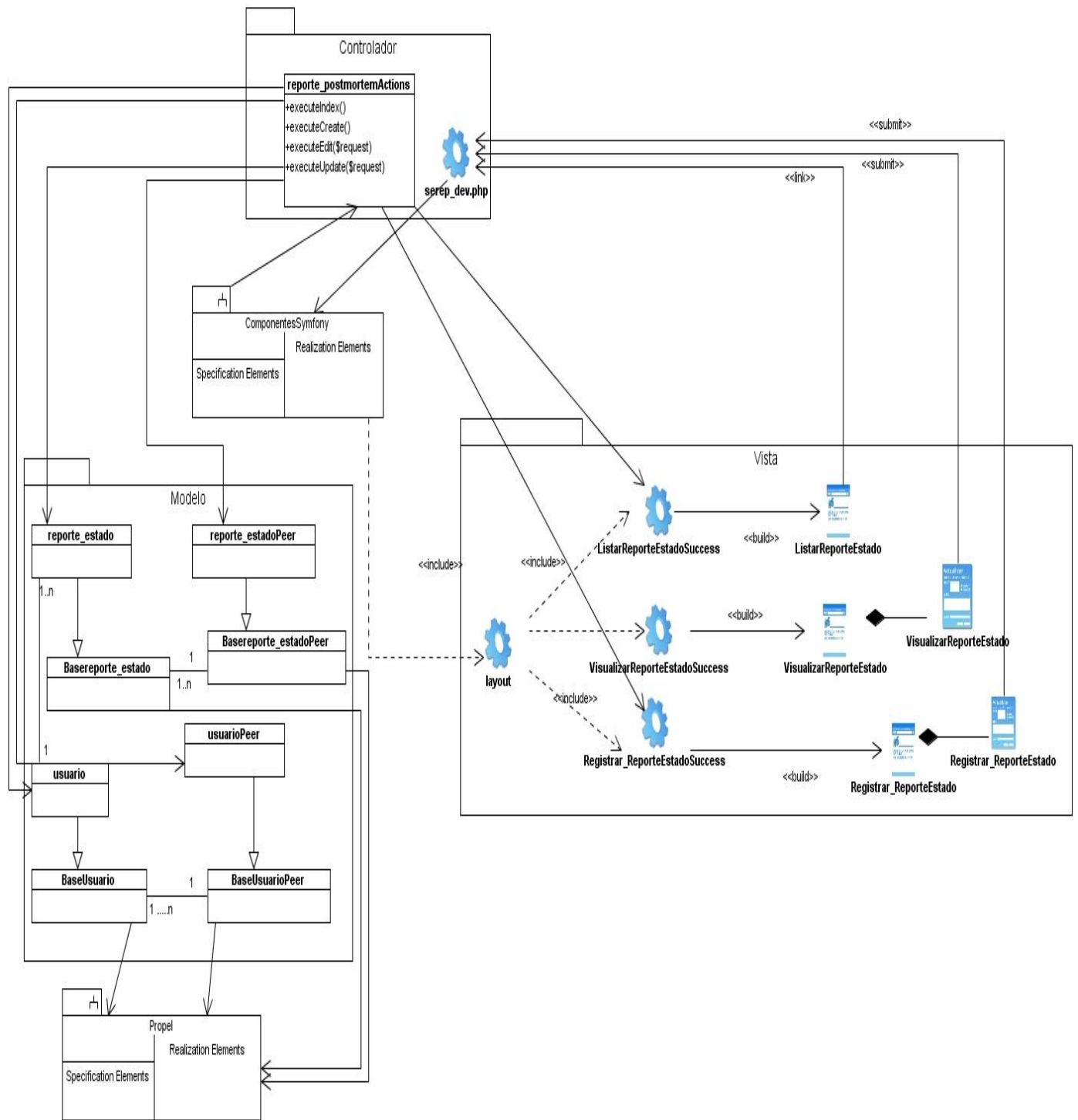


Diagrama de clases del diseño del CU: Registrar Reporte Postmortem

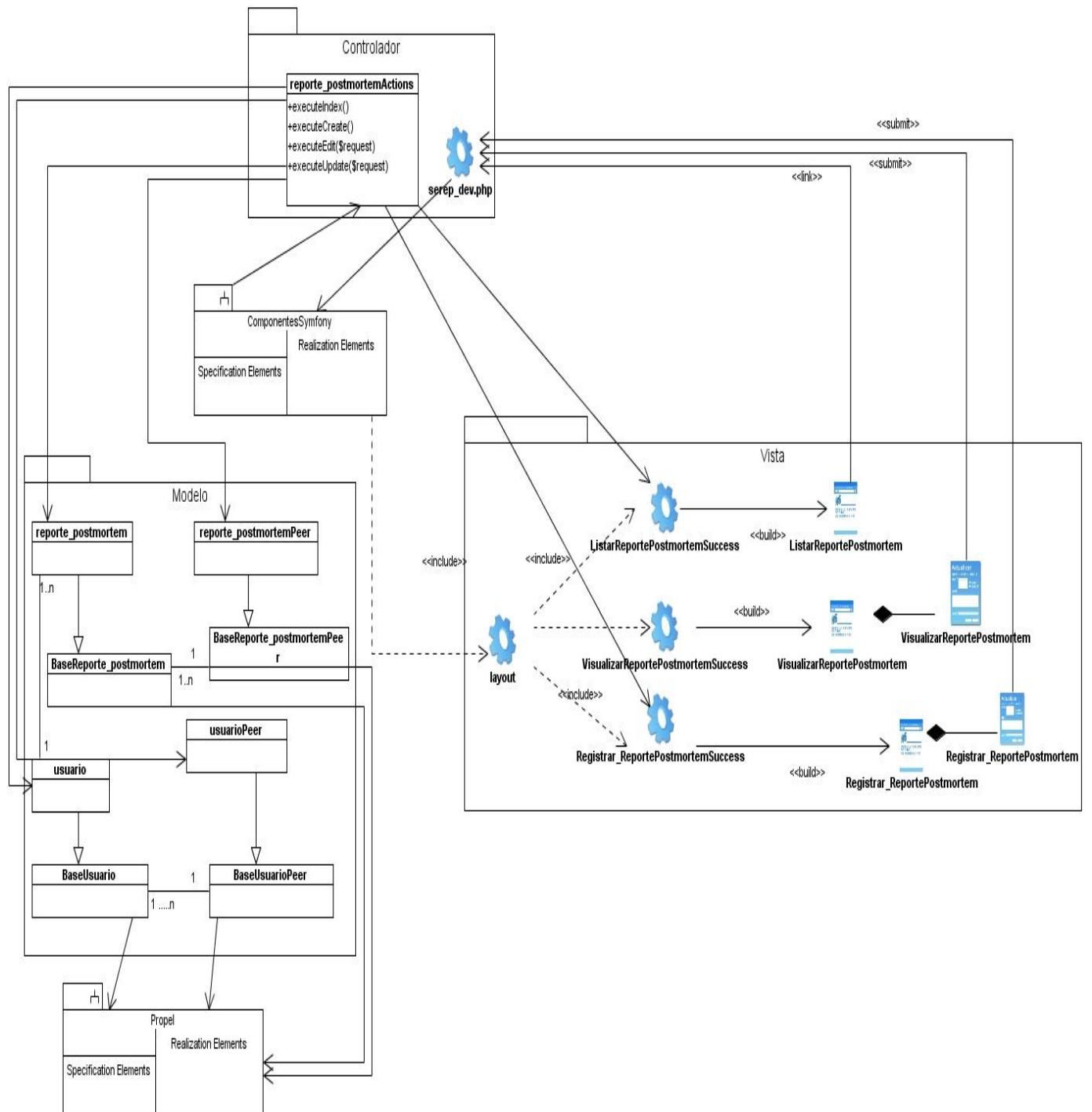


Diagrama de clases del diseño del CU: Emitir Criterio de Valoración

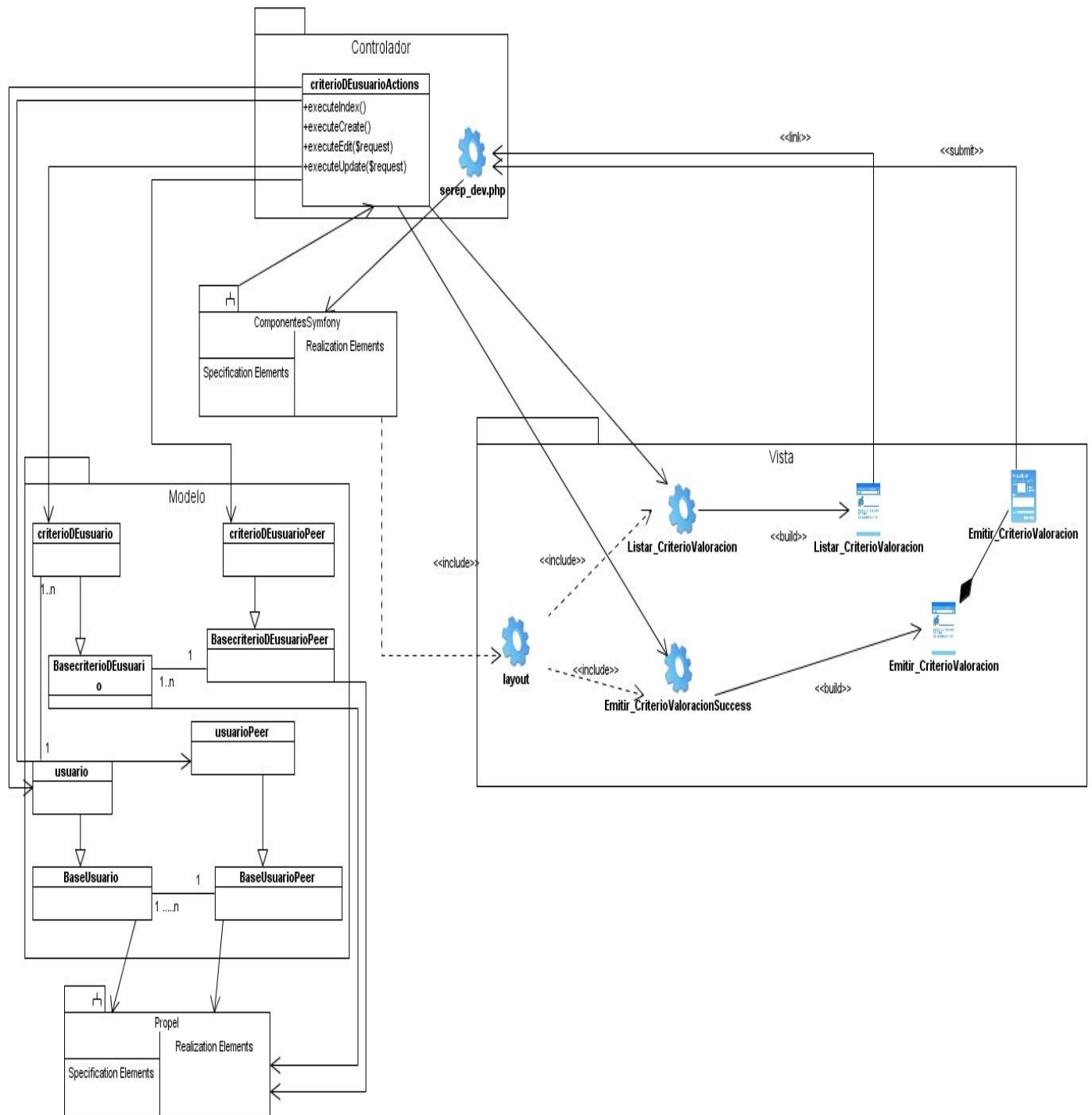
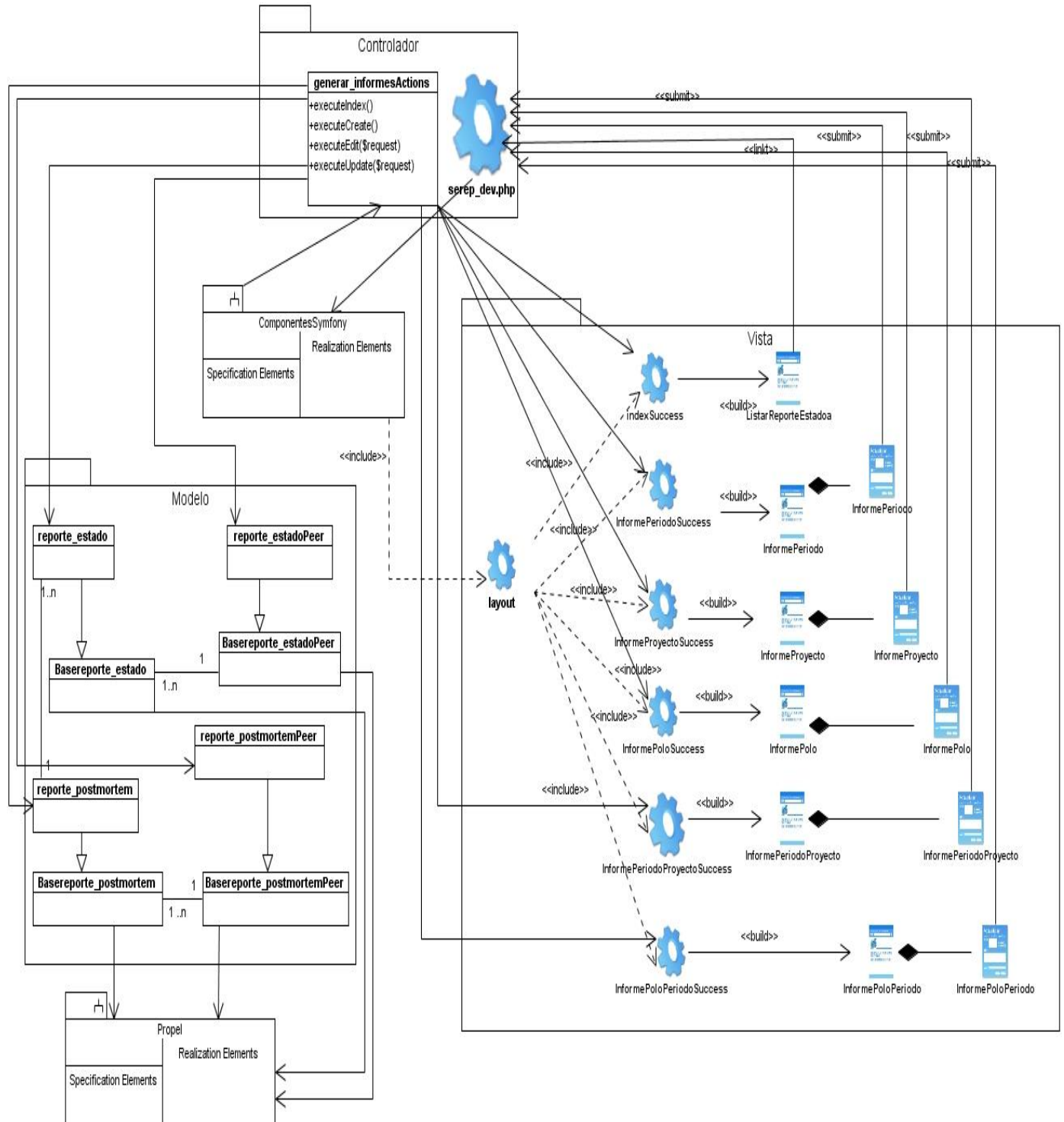


Diagrama de clases del diseño del CU: Generar Informe



Anexo 7: Diagramas de secuencia

Diagrama de secuencia del diseño del CU: Autenticar Usuario

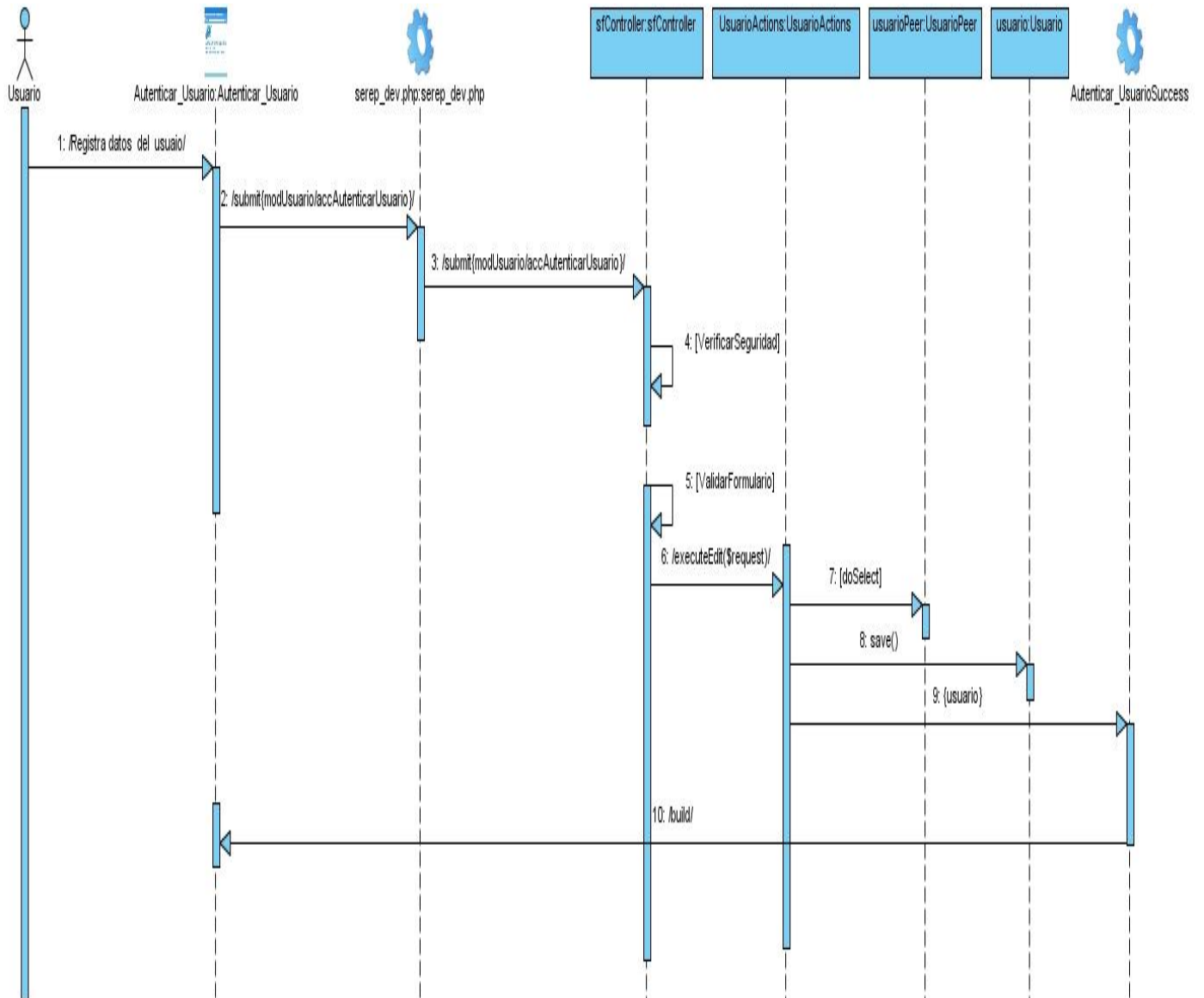


Diagrama de secuencia del diseño del escenario Adicionar Área del CU: Gestionar Área.

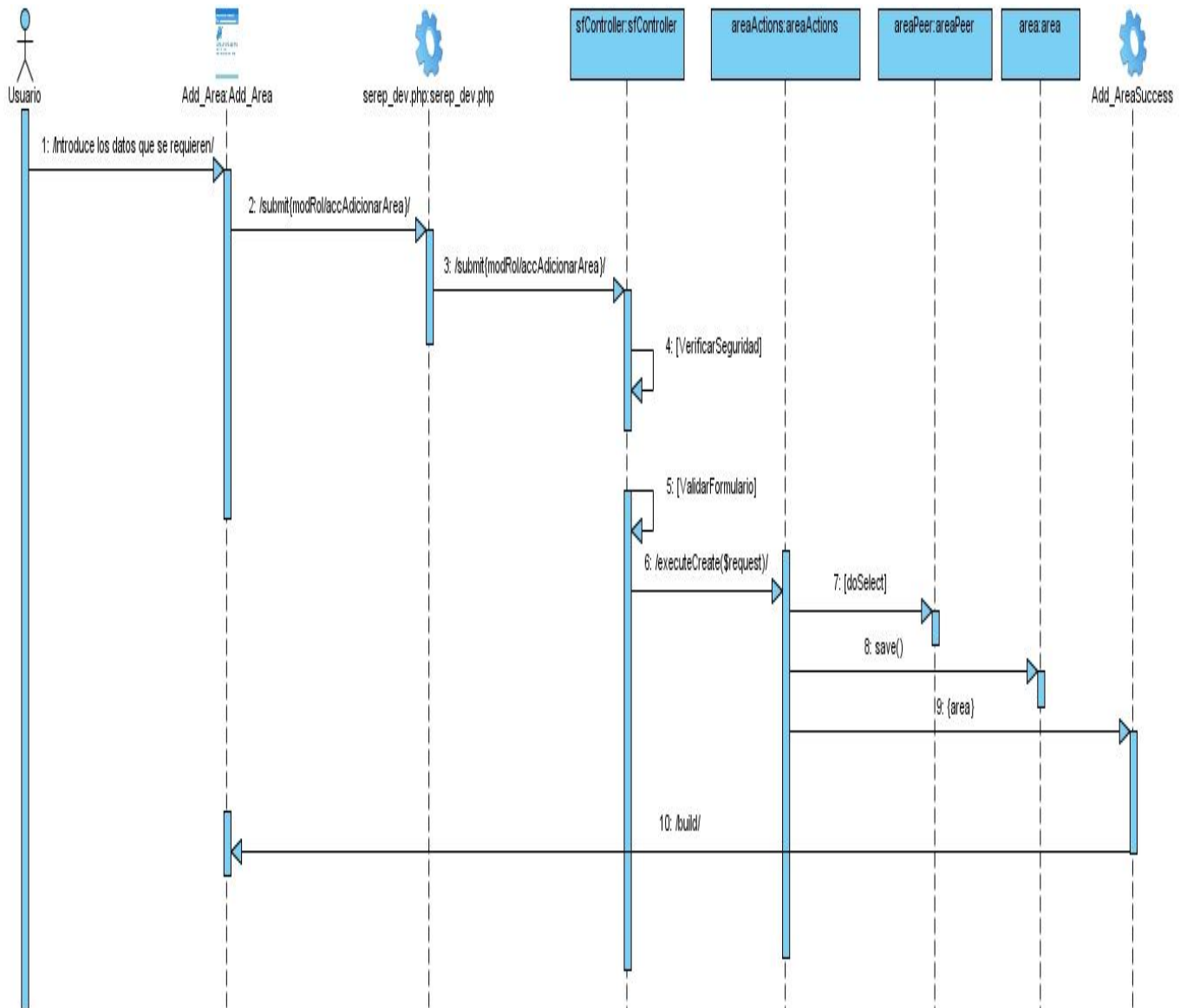


Diagrama de secuencia del diseño del escenario Editar Área del CU: Gestionar Área.

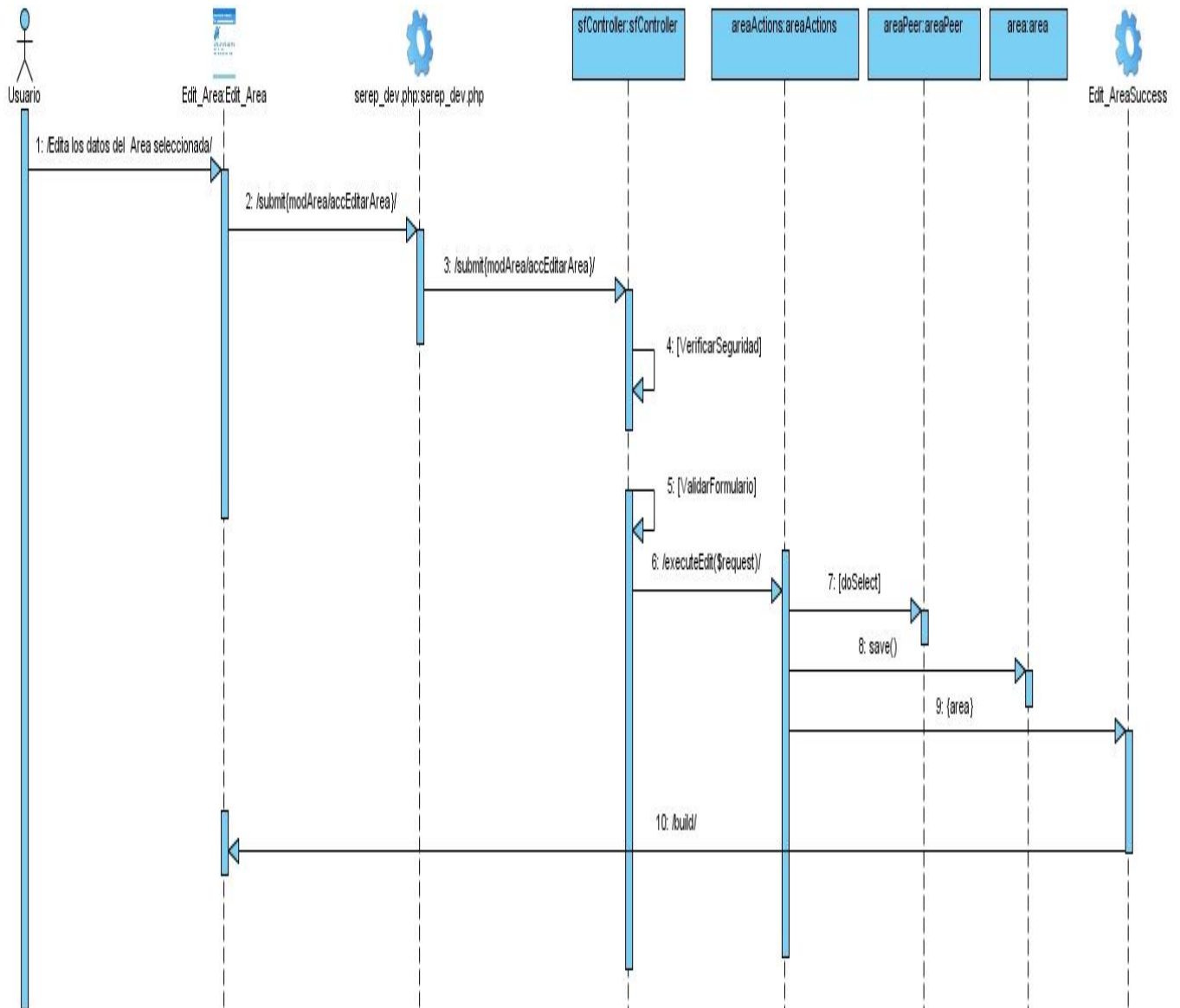


Diagrama de secuencia del diseño del escenario Eliminar Área del CU: Gestionar Área.

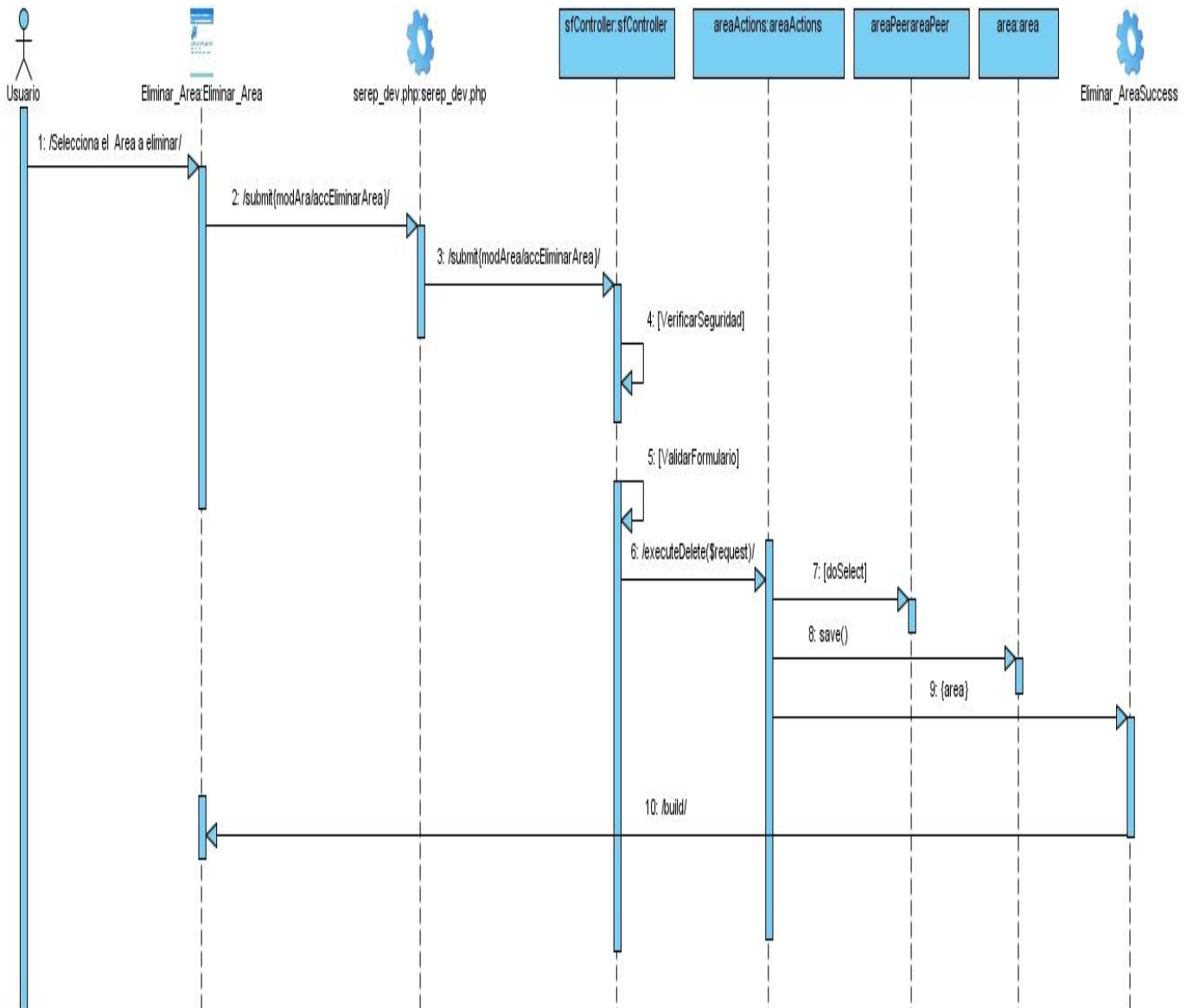


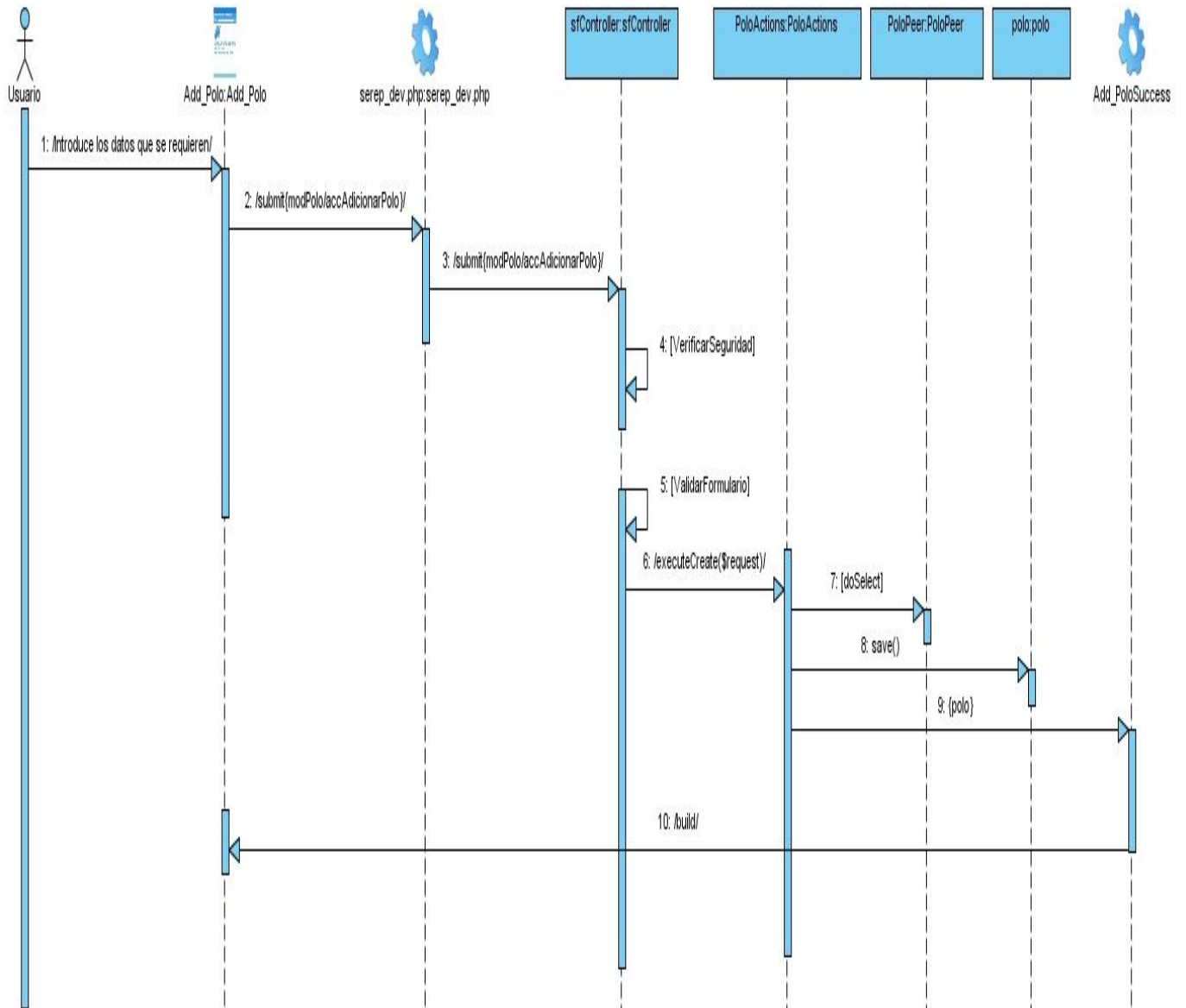
Diagrama de secuencia del diseño del escenario Adicionar Polo del CU: Gestionar Polo.

Diagrama de secuencia del diseño del escenario Editar Polo del CU: Gestionar Polo.

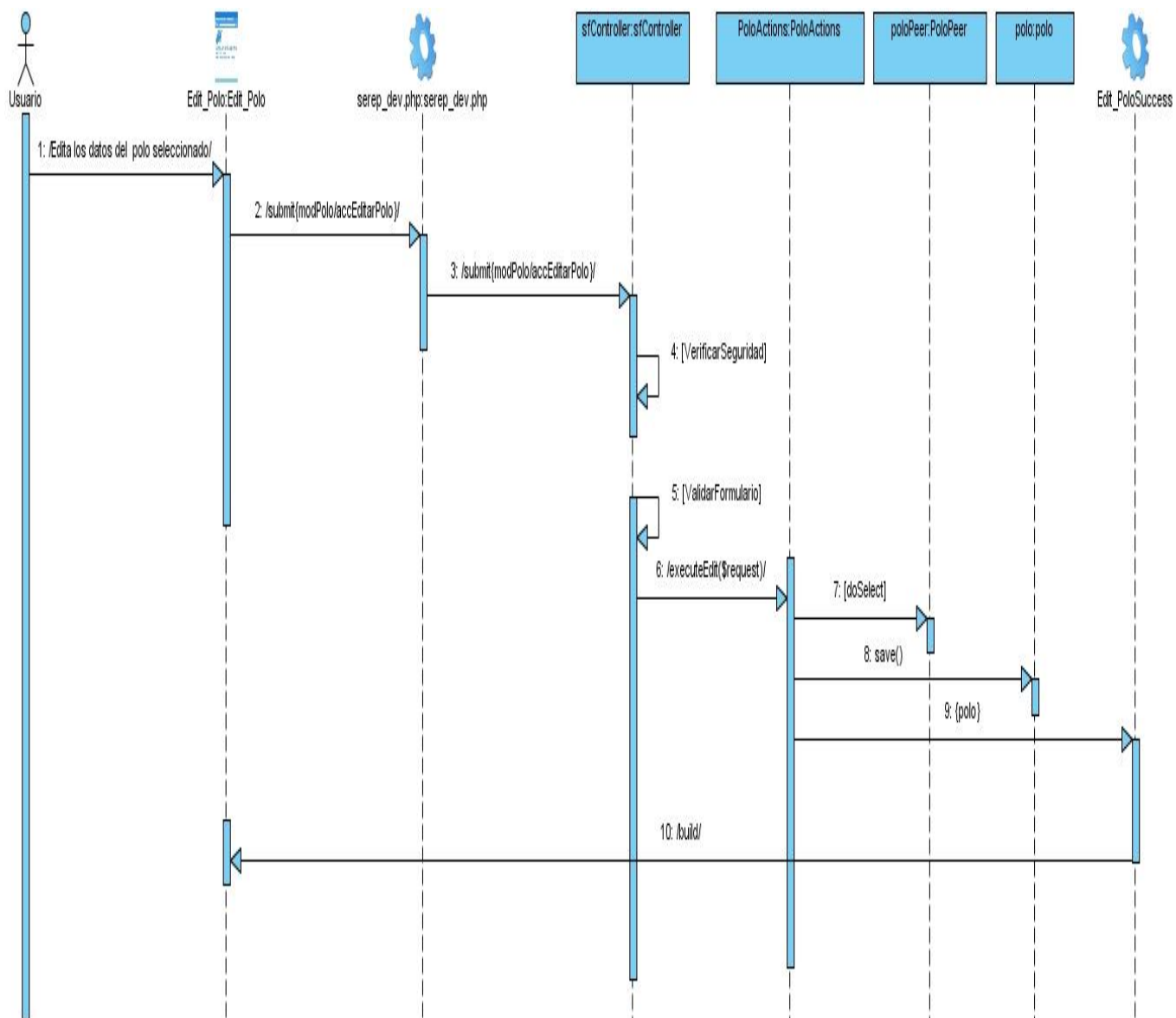


Diagrama de secuencia del diseño del escenario Eliminar Polo del CU: Gestionar Polo.

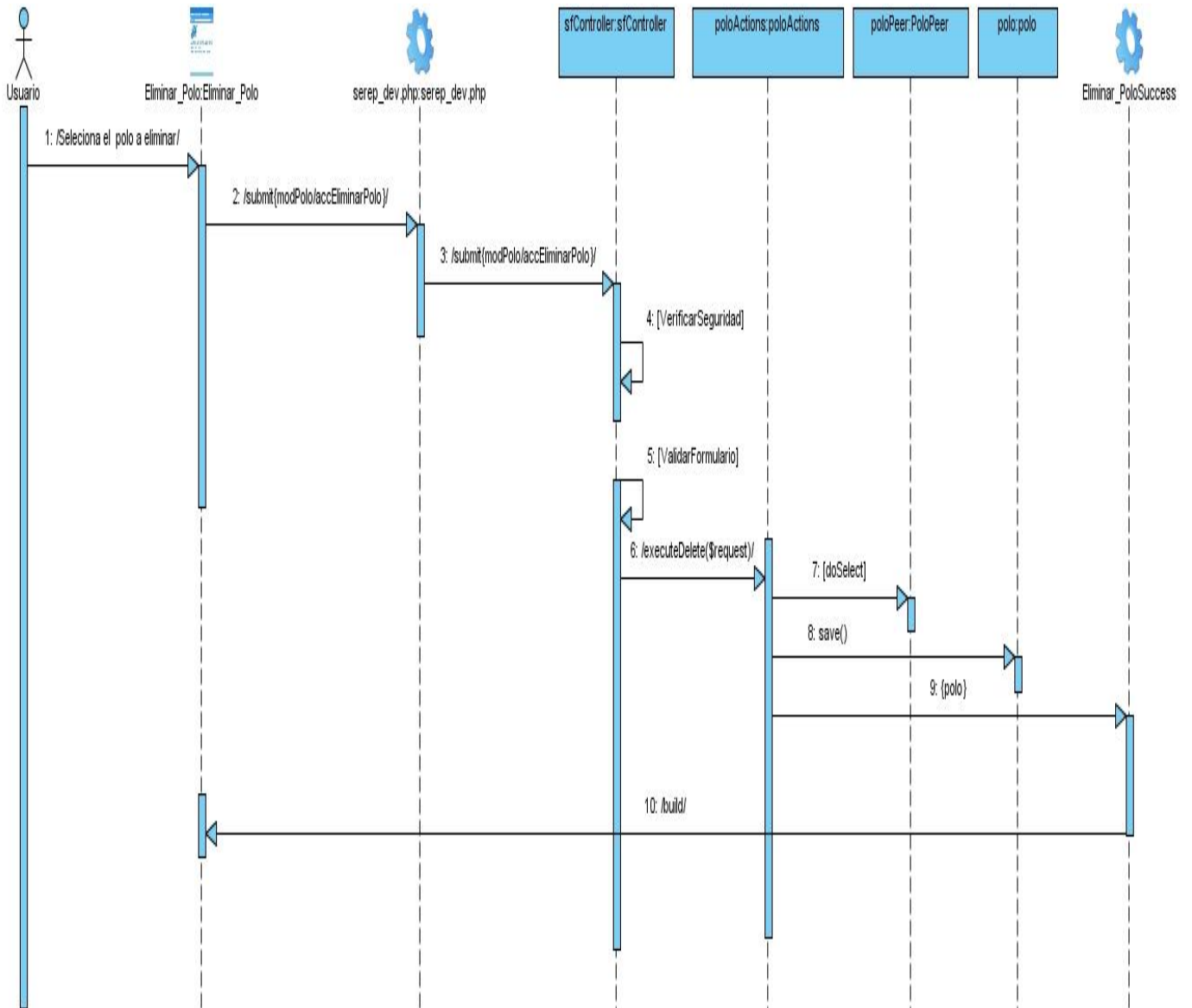


Diagrama de secuencia del diseño del escenario Adicionar Proyecto del CU: Gestionar Proyecto.

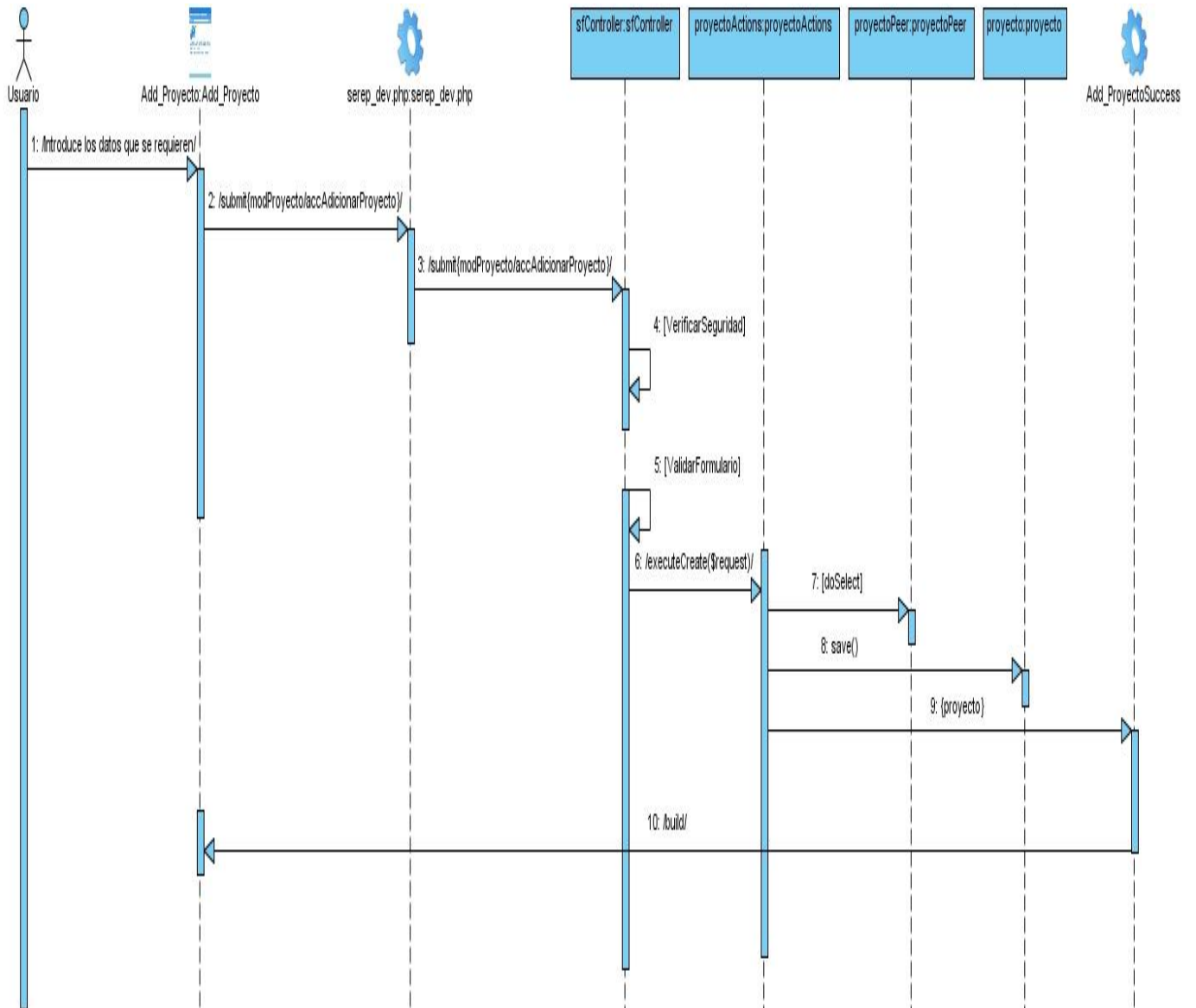


Diagrama de secuencia del diseño del escenario Editar Proyecto del CU: Gestionar Proyecto.

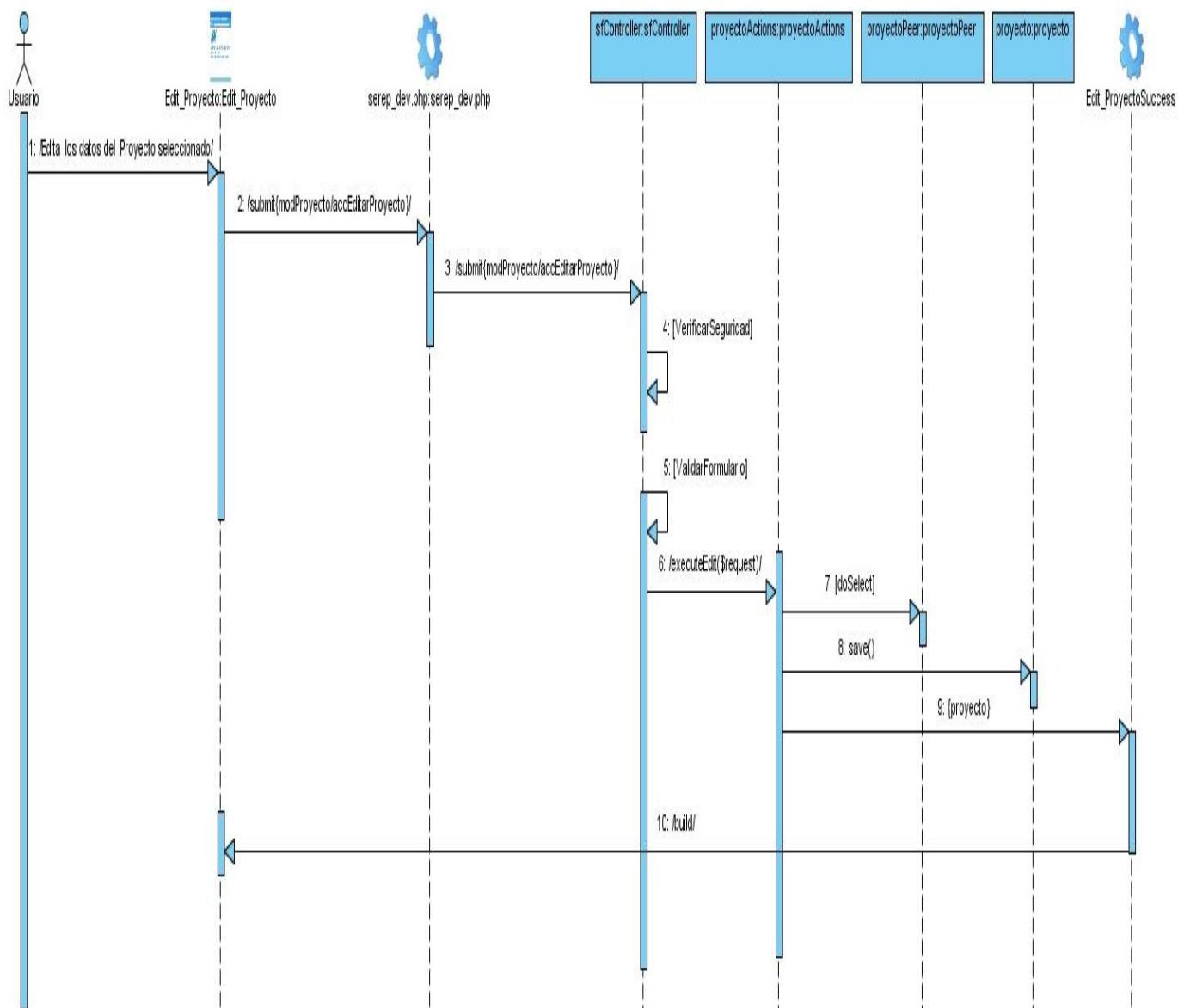


Diagrama de secuencia del diseño del escenario Eliminar Proyecto del CU: Gestionar Proyecto.

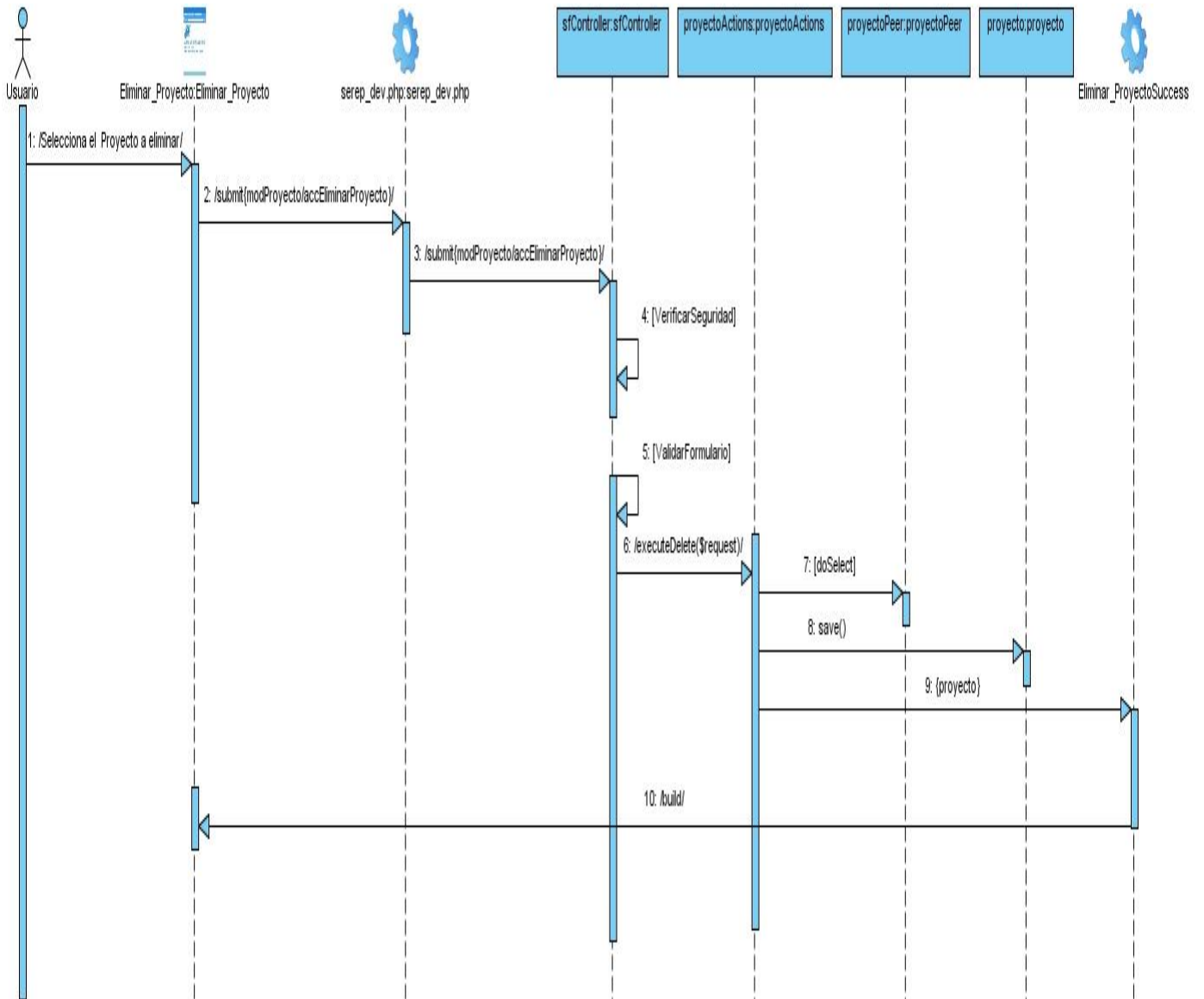


Diagrama de secuencia del diseño del escenario Adicionar Rol del CU: Gestionar Rol.

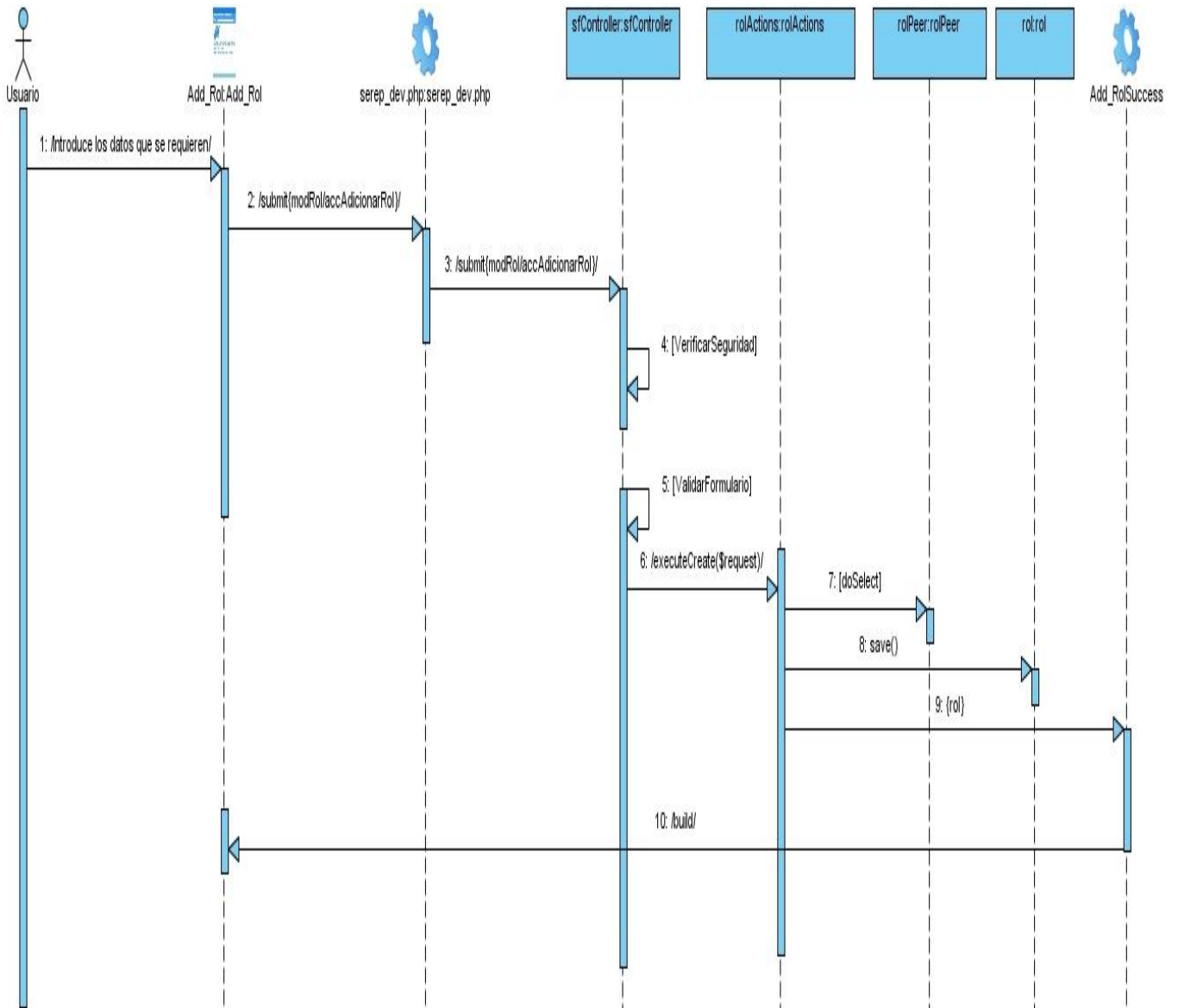


Diagrama de secuencia del diseño del escenario Editar Rol del CU: Gestionar Rol.

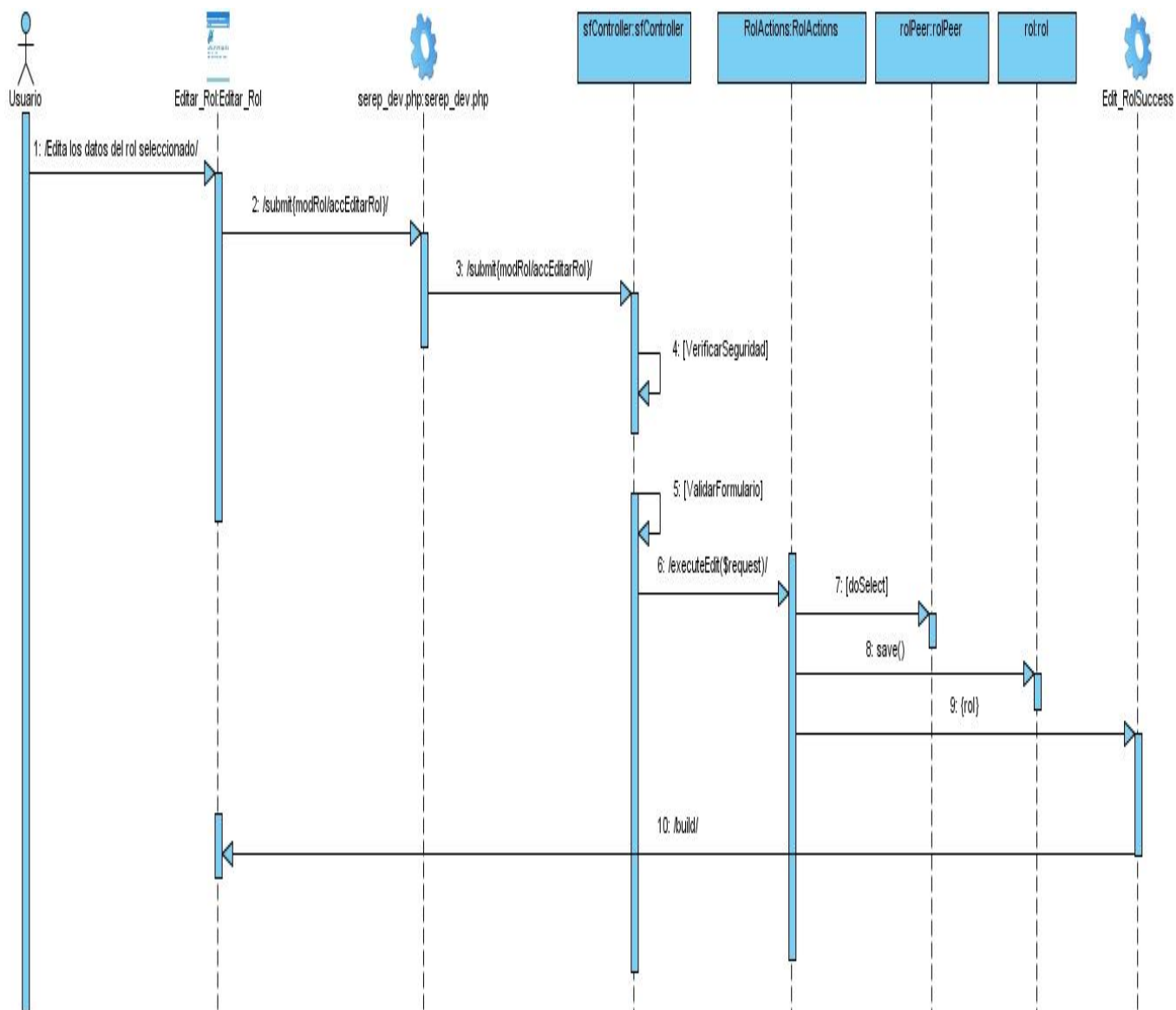


Diagrama de secuencia del diseño del escenario Eliminar Rol del CU: Gestionar Rol.

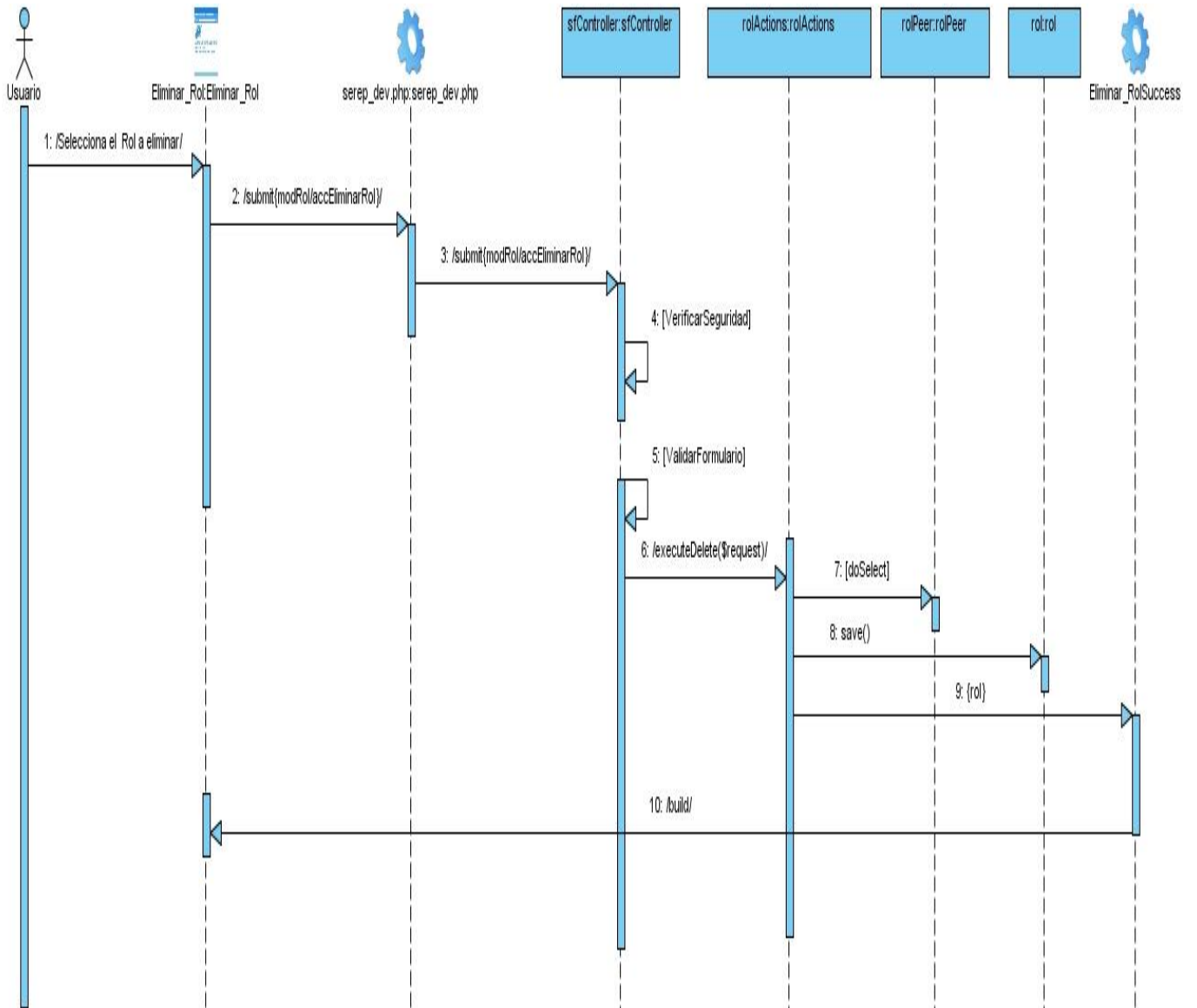


Diagrama de secuencia del diseño del escenario Adicionar Usuario del CU: Gestionar Usuario.

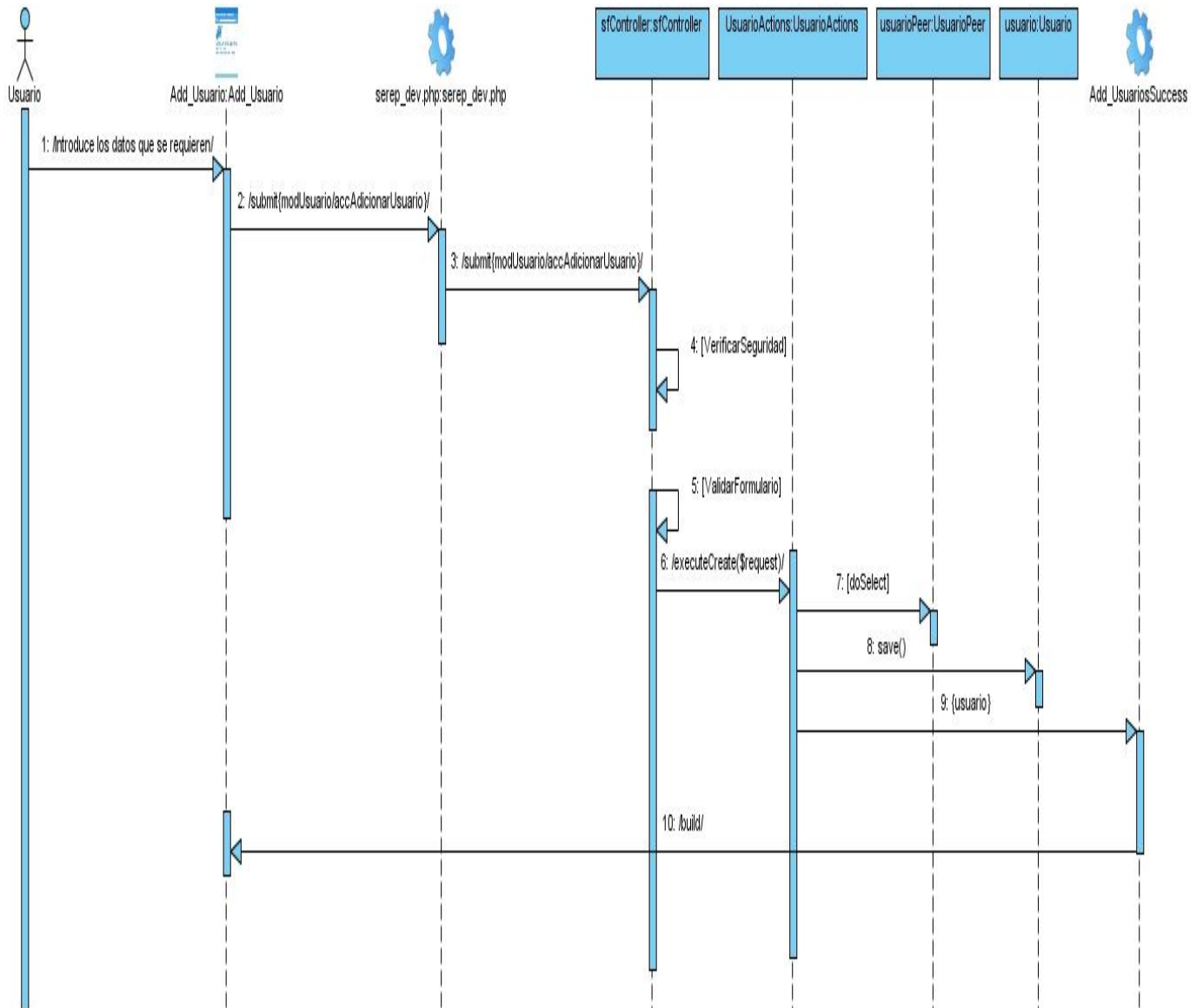


Diagrama de secuencia del diseño del escenario Editar Usuario del CU: Gestionar Usuario.

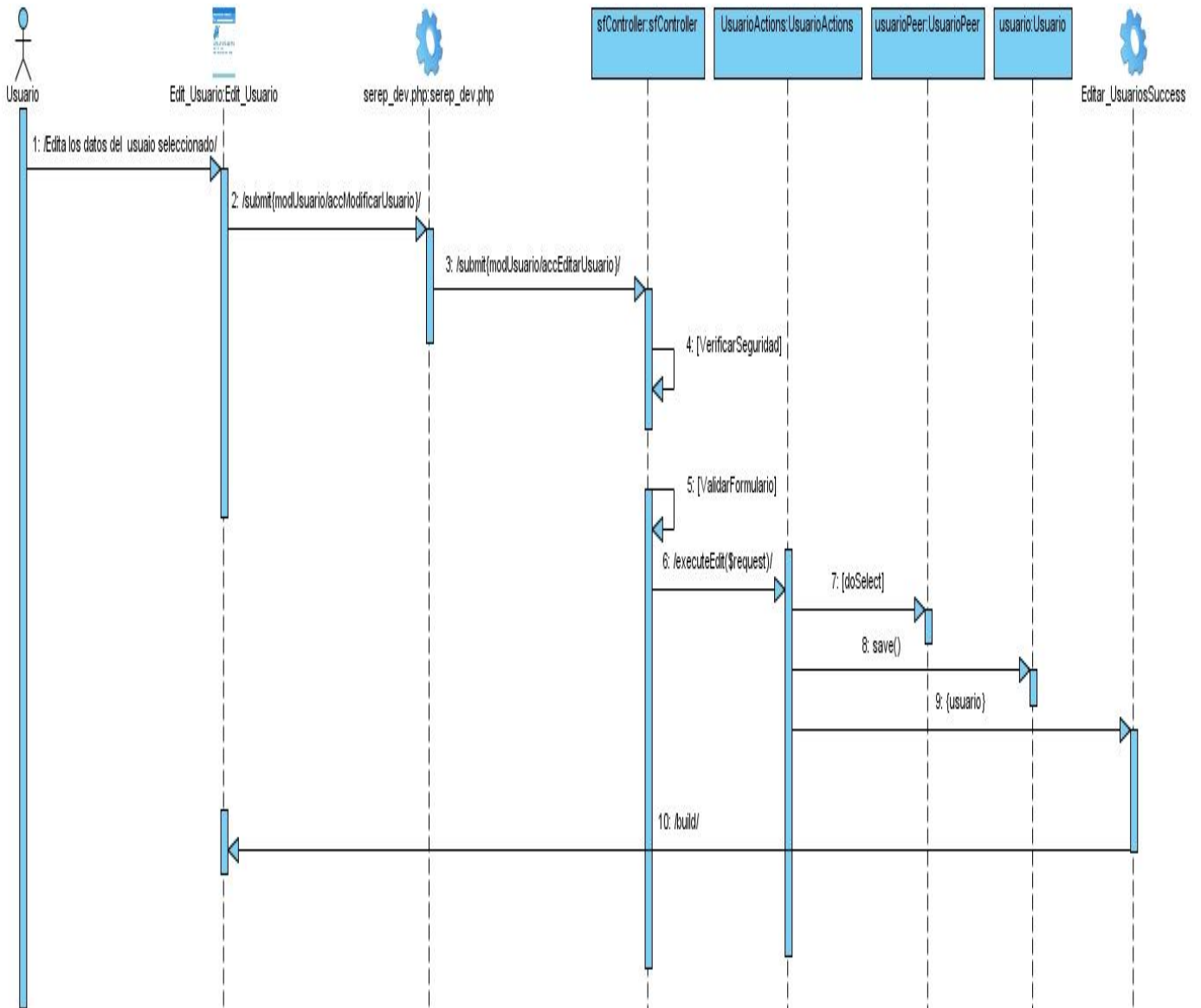


Diagrama de secuencia del diseño del escenario Eliminar Usuario del CU: Gestionar Usuario.

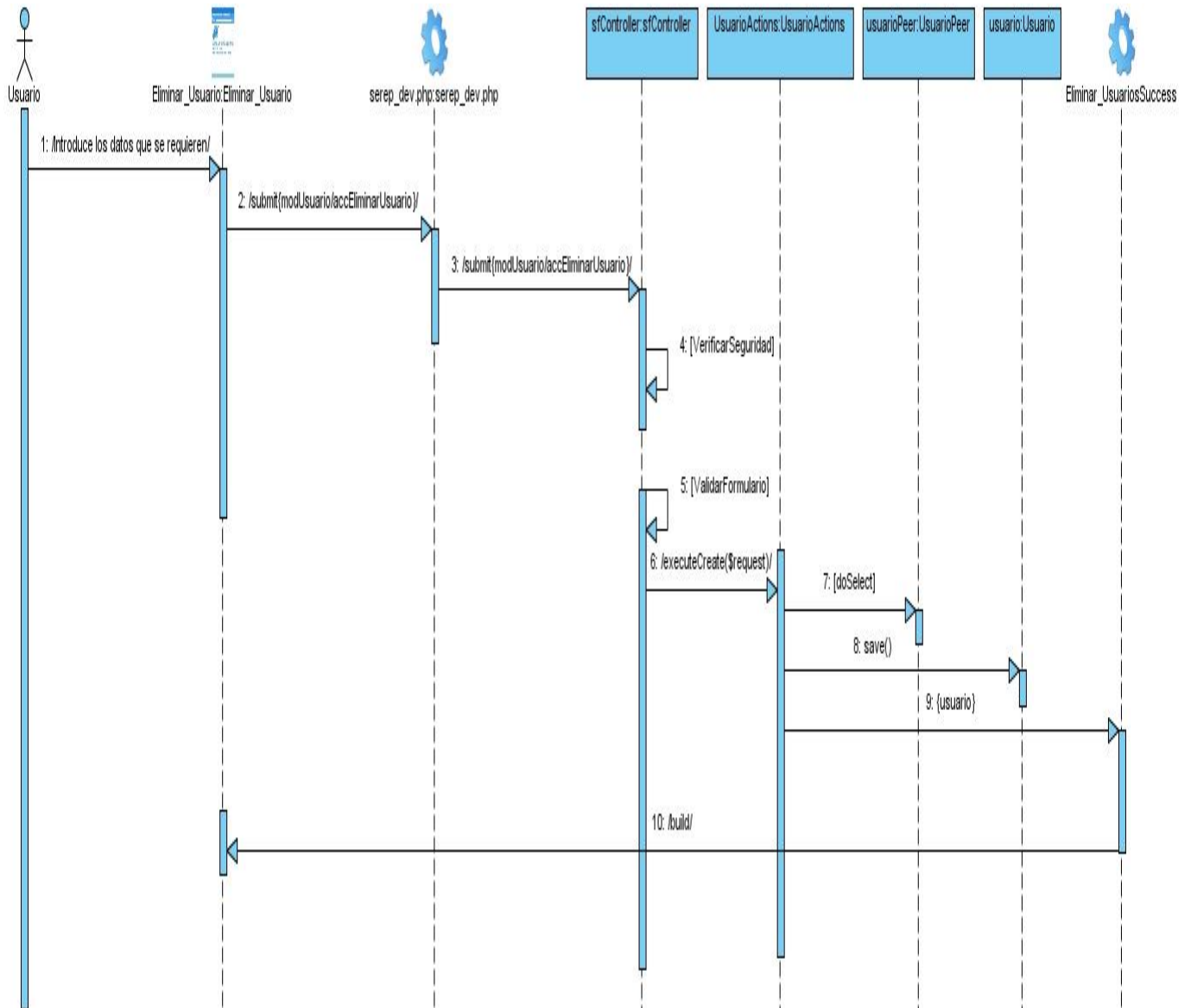


Diagrama de secuencia del diseño del escenario Adicionar Criterio de Valoración del CU: Gestionar Criterio de Valoración.

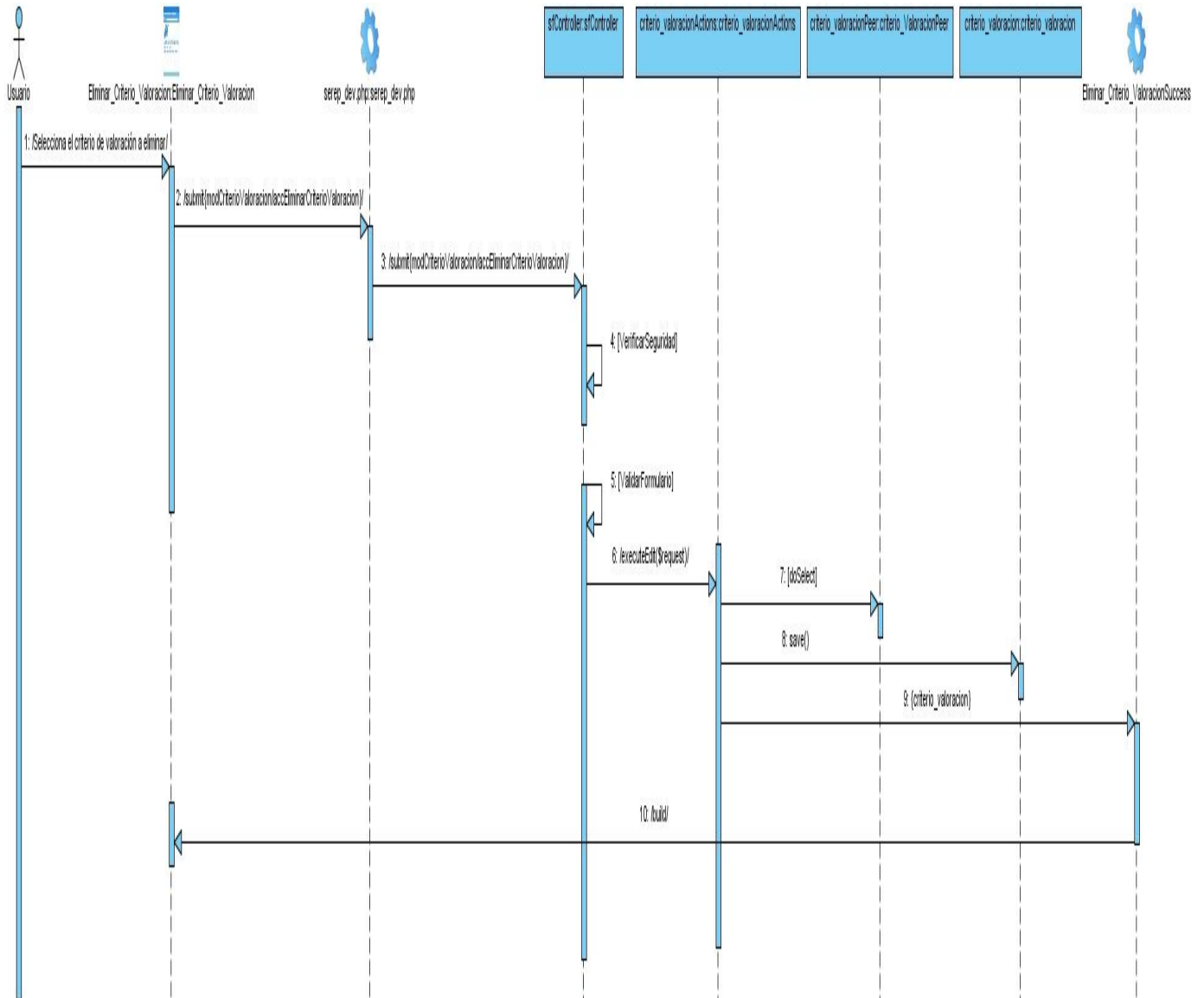


Diagrama de secuencia del diseño del escenario Editar Criterio de Valoración del CU: Gestionar Criterio de Valoración.

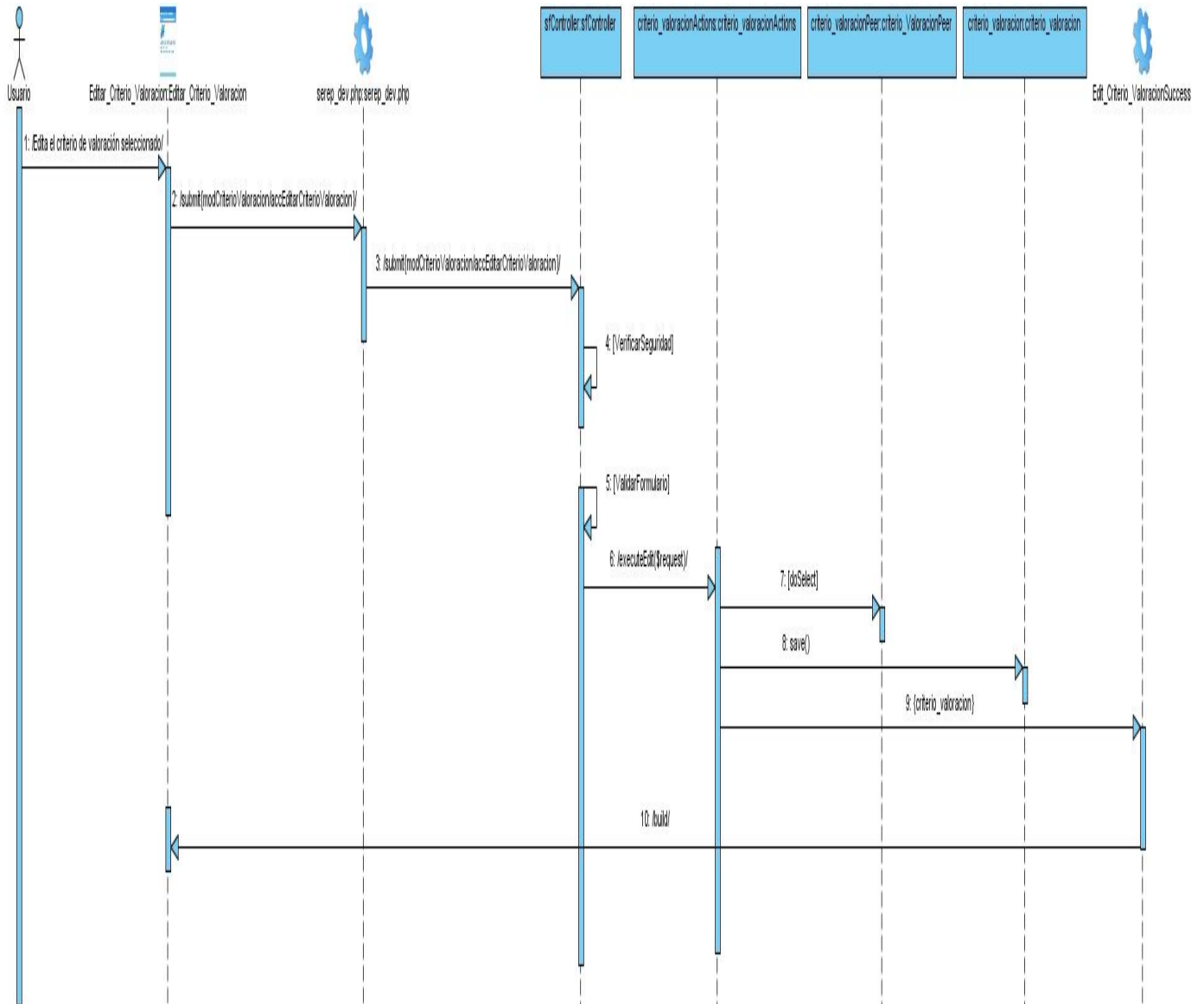


Diagrama de secuencia del diseño del escenario Eliminar Criterio de Valoración del CU: Gestionar Criterio de Valoración.

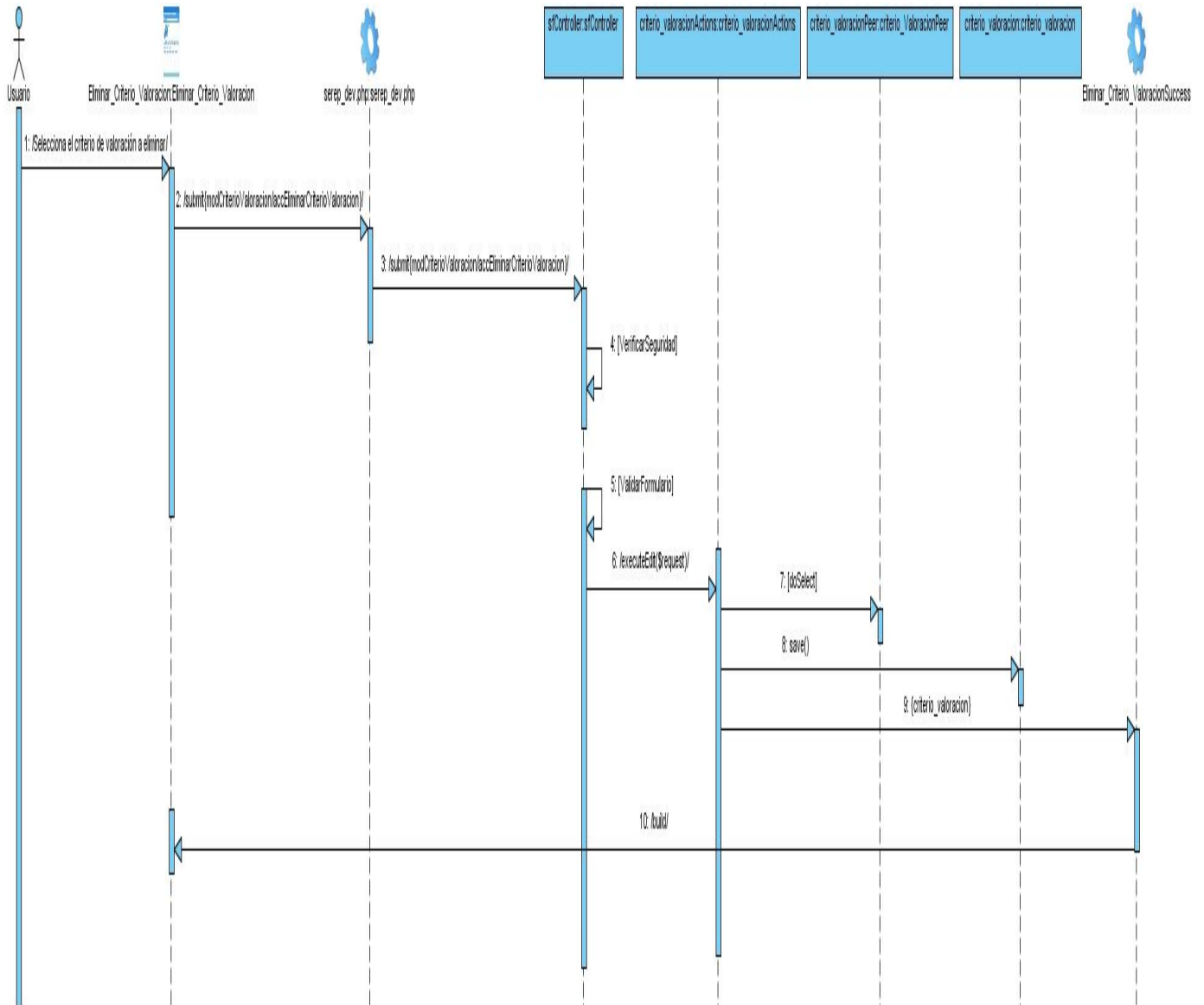


Diagrama de secuencia del diseño del CU: Registrar Reporte de Estado.

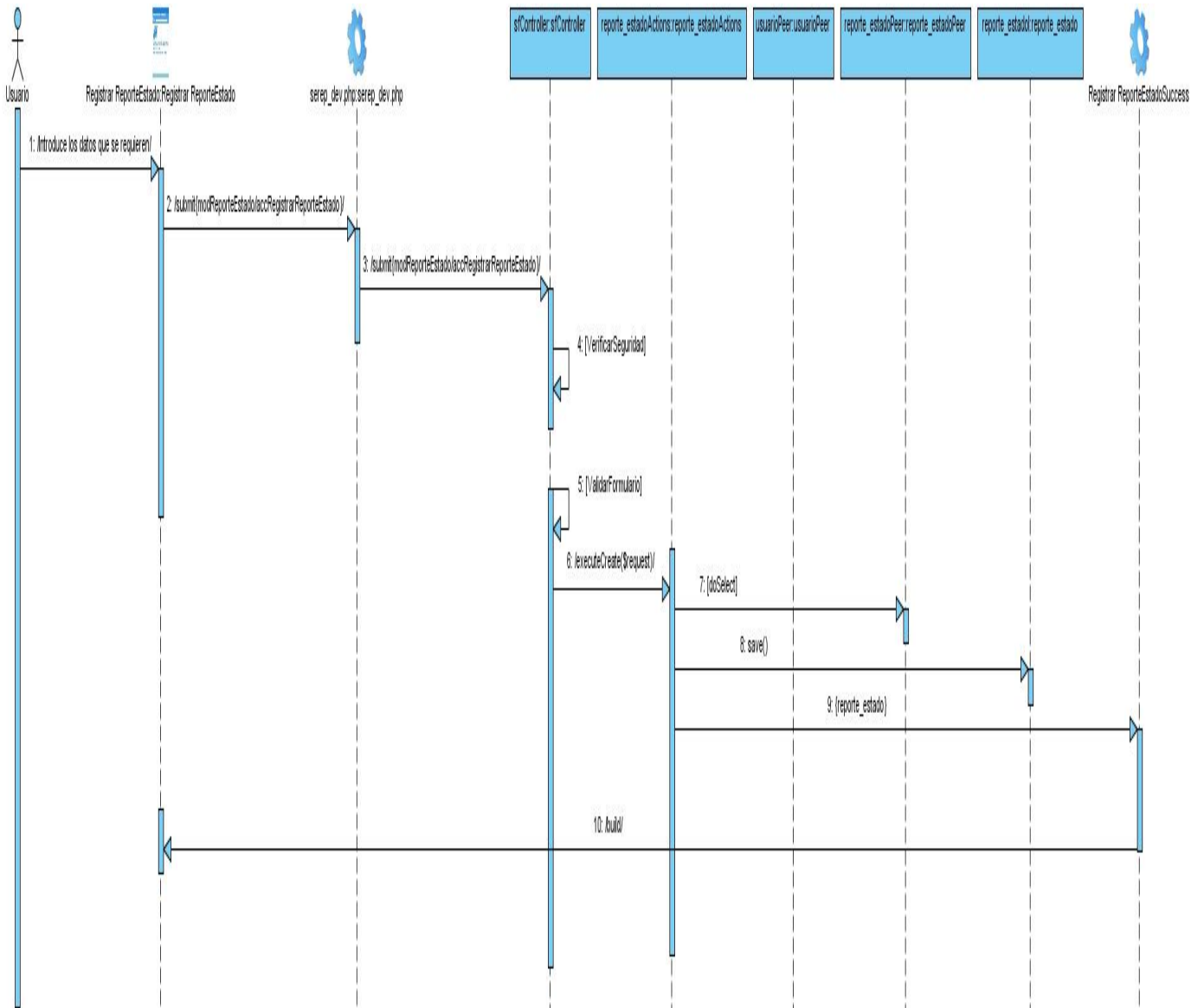


Diagrama de secuencia del diseño del CU: Registrar Reporte Postmortem

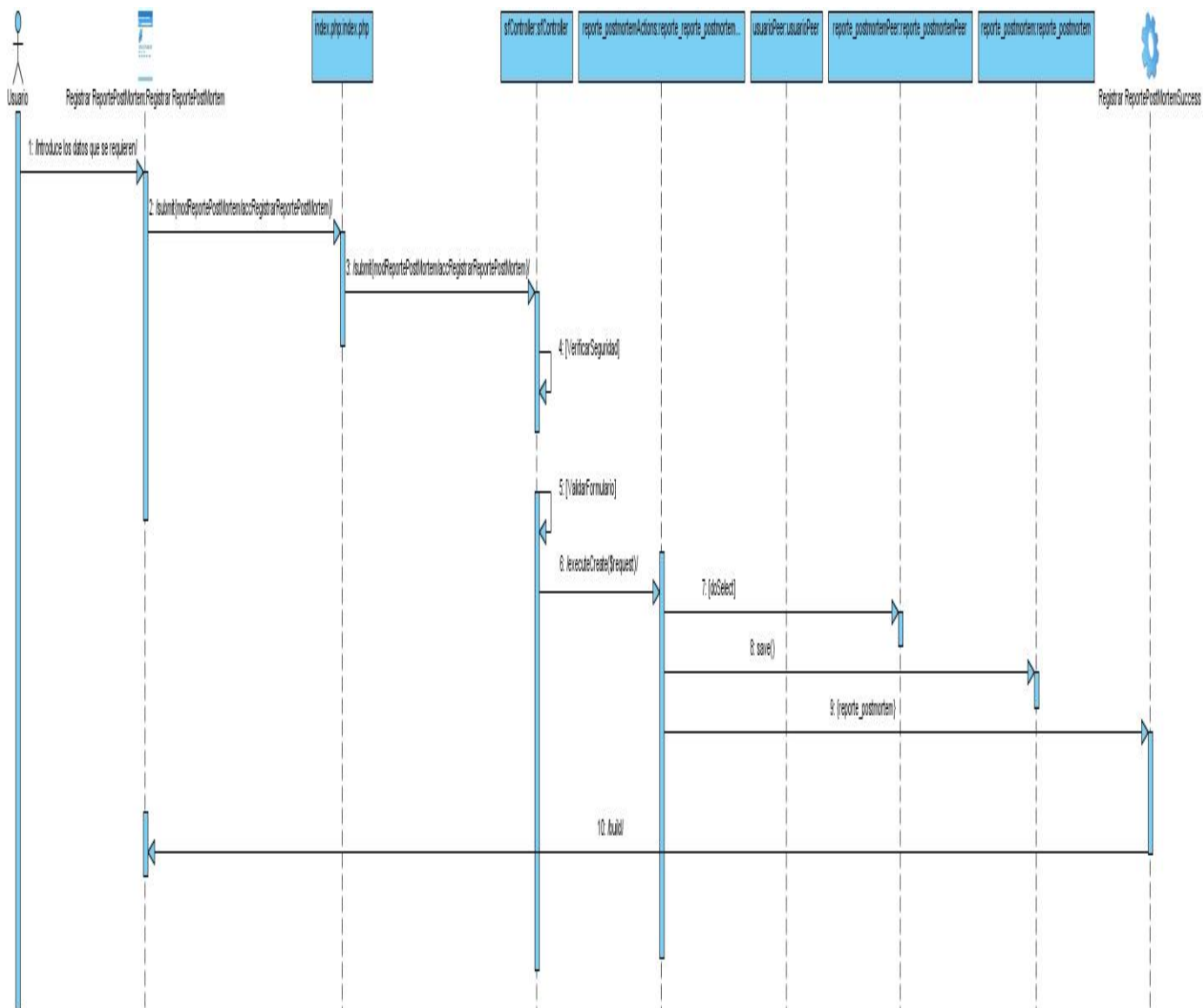


Diagrama de secuencia del diseño del CU: Emitir Criterio de Valoración

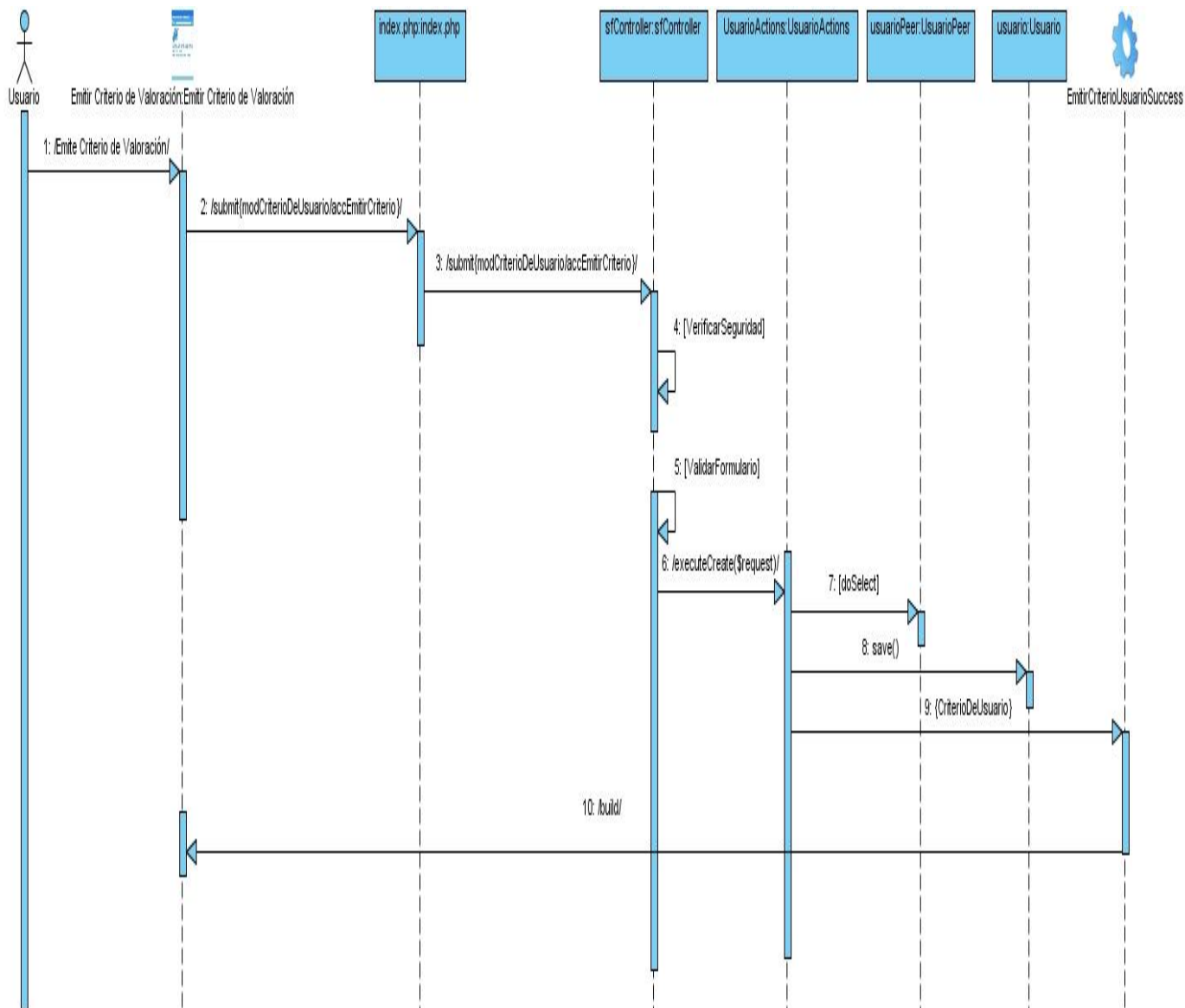
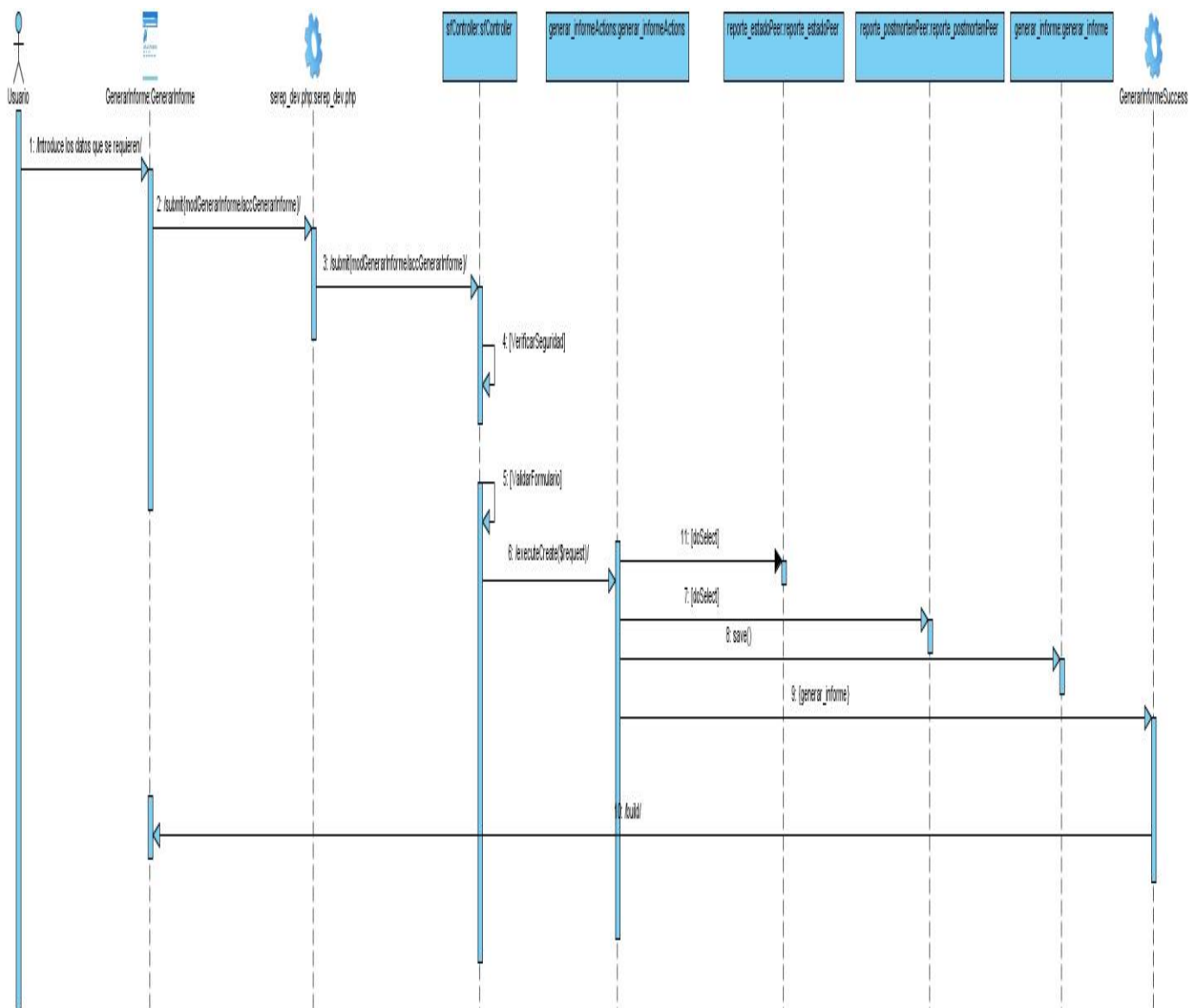


Diagrama de secuencia del diseño del CU: Generar Informe



Anexo 8: Descripción de las clases

Tabla 13 Descripción de la Clase del Diseño: BaseAreas

Nombre	BaseAreas
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_areas	
\$nombre	
\$descripcion	
Para cada Responsabilidad	
Nombre:	getIdAreas()
Descripción:	Devuelve el número identificativo del Área.
Nombre:	getNombre()
Descripción:	Devuelve el nombre del Área.
Nombre:	getDescripcion()
Descripción:	Devuelve la descripción del Área.
Nombre:	setIdAreas()
Descripción:	Cambia el número identificativo del Área.
Nombre:	setNombre()
Descripción:	Cambia el nombre del Área.
Nombre:	setDescripcion()
Descripción:	Cambia la descripción del Área.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	doSave()
Descripción:	Función llamada por la función save()

Tabla 14 Descripción de la Clase del Diseño: BasePolo

Nombre	BasePolo
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_polo	
\$id_areas	
\$nombre	
\$descripcion	
Para cada Responsabilidad	
Nombre:	getIdPolo()
Descripción:	Devuelve el número identificativo del Polo.
Nombre:	getIdAreas()

Descripción:	Devuelve número identificativo del Área.
Nombre:	getNombre()
Descripción:	Devuelve el nombre del Polo.
Nombre:	getDescripcion()
Descripción:	Devuelve la descripción del Polo.
Nombre:	setIdPolo()
Descripción:	Cambia el número identificativo del Polo.
Nombre:	setIdAreas()
Descripción:	Cambia el número identificativo del Área.
Nombre:	setNombre()
Descripción:	Cambia el nombre del Polo.
Nombre:	setDescripcion()
Descripción:	Cambia la descripción del Polo
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 15 Descripción de la Clase del Diseño: BaseActividad

Nombre	BaseActividad
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_actividad	
\$id_hitos	
\$actividad	
\$fecha	
\$completado	
\$estadoentrega	
Para cada Responsabilidad	
Nombre:	getIdActividad()
Descripción:	Devuelve el número identificativo de una Actividad.
Nombre:	getIdHitos()
Descripción:	Devuelve número identificativo del Hito al que pertenece.
Nombre:	getActividad()
Descripción:	Devuelve la actividad.
Nombre:	getFecha()
Descripción:	Devuelve la Fecha.
Nombre:	getCompletado()
Descripción:	Devuelve el % completado de la Actividad.
Nombre:	getEstadoentrega()

Descripción:	Devuelve el estado de entrega de la Actividad.
Nombre:	setIdActividad()
Descripción:	Cambia el número identificativo de una Actividad.
Nombre:	setIdHitos()
Descripción:	Cambia el número identificativo del Hito al que pertenece.
Nombre:	setActividad()
Descripción:	Cambia la Actividad.
Nombre:	setFecha()
Descripción:	Cambia la fecha de la Actividad.
Nombre:	setCompletado()
Descripción:	Cambia el % completado de la Actividad.
Nombre:	setEstadoentrega()
Descripción:	Cambia el estado de entrega de la Actividad.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 16 Descripción de la Clase del Diseño: BaseCriterioDeUsuario

Nombre	BaseCriterioDeUsuario	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_criterio_de_usuario		
\$id_proyecto		
\$id_usuario		
\$id_criterio_evaluacion		
\$grupo		
\$fecha		
Para cada Responsabilidad		
Nombre:	getIdCriterioDeUsuario()	
Descripción:	Devuelve el número identificativo de un CriterioDeUsuario.	
Nombre:	getIdProyecto()	
Descripción:	Devuelve el número identificativo del proyecto que se valoró.	
Nombre:	getIdUsuario()	
Descripción:	Devuelve el número identificativo del usuario que emite el criterio.	
Nombre:	getIdCriterioEvaluacion()	
Descripción:	Devuelve el número identificativo del criterio sobre el que se valoró.	
Nombre:	getGrupo()	
Descripción:	Devuelve el grupo al que pertenece.	
Nombre:	getFecha()	

Descripción:	Devuelve la Fecha.
Nombre:	setIdCriterioDeUsuario()
Descripción:	Cambia el número identificativo de un CriterioDeUsuario.
Nombre:	setIdProyecto()
Descripción:	Cambia el número identificativo del proyecto que se valoró.
Nombre:	setIdUsuario()
Descripción:	Cambia el número identificativo del usuario que emite el criterio.
Nombre:	setIdCriterioEvaluacion()
Descripción:	Cambia el número identificativo del criterio sobre el que se valoró.
Nombre:	setGrupo()
Descripción:	Cambia el grupo al que pertenece.
Nombre:	setFecha()
Descripción:	Cambia la Fecha.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 17 Descripción de la Clase del Diseño: BaseCriterioEvaluacion

Nombre	BaseCriterioEvaluacion	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_criterio_evaluacion		
\$id_rol		
\$criterio		
\$valor		
\$grupo		
Para cada Responsabilidad		
Nombre:	getIdCriterioValoracion()	
Descripción:	Devuelve el número identificativo del CriterioValoracion.	
Nombre:	getIdRol()	
Descripción:	Devuelve número identificativo del Rol.	
Nombre:	getCriterio()	
Descripción:	Devuelve el criterio.	
Nombre:	getValor()	
Descripción:	Devuelve el valor del criterio de valoración.	
Nombre:	getGrupo()	
Descripción:	Devuelve el grupo al que pertenece.	
Nombre:	setIdCriterioValoracion()	

Descripción:	Cambia el número identificativo de un CriterioValoracion.
Nombre:	setIdRol()
Descripción:	Cambia el número identificativo del Rol.
Nombre:	setCriterio()
Descripción:	Cambia el criterio.
Nombre:	setValor()
Descripción:	Cambia el valor del criterio de valoración.
Nombre:	setGrupo()
Descripción:	Cambia el grupo al que pertenece.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 18 Descripción de la Clase del Diseño: BaseHitos

Nombre	BaseHitos	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_hitos		
\$id_reporte_estado		
\$hitos		
Para cada Responsabilidad		
Nombre:	getIdHitos()	
Descripción:	Devuelve el número identificativo del hito.	
Nombre:	getIdReporteEstado()	
Descripción:	Devuelve el número identificativo del Reporte de Estado.	
Nombre:	getHitos()	
Descripción:	Devuelve el hito.	
Nombre:	setIdHitos()	
Descripción:	Cambia el número identificativo del Hito.	
Nombre:	setIdReporteEstado ()	
Descripción:	Cambia el número identificativo del Reporte de Estado.	
Nombre:	setHitos()	
Descripción:	Cambia el hito.	
Nombre:	save()	
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.	
Nombre:	doSave()	
Descripción:	Función llamada por la función save()	

Tabla 19 Descripción de la Clase del Diseño: BaseProyecto

Nombre	BaseProyecto	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_proyecto		
\$id_polo		
\$nombre		
\$descripcion		
Para cada Responsabilidad		
Nombre:	getIdProyecto()	
Descripción:	Devuelve el número identificativo del Proyecto.	
Nombre:	getIdPolo()	
Descripción:	Devuelve número identificativo del polo al que pertenece.	
Nombre:	getNombre()	
Descripción:	Devuelve el nombre de un Proyecto.	
Nombre:	getDescripcion()	
Descripción:	Devuelve la descripción de un Proyecto.	
Nombre:	setIdProyecto()	
Descripción:	Cambia el número identificativo de un Proyecto.	
Nombre:	setIdPolo()	
Descripción:	Cambia el número identificativo del polo al que pertenece.	
Nombre:	setNombre()	
Descripción:	Cambia el nombre del proyecto.	
Nombre:	setDescripcion()	
Descripción:	Cambia la descripción del proyecto.	
Nombre:	save()	
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.	
Nombre:	dosave()	
Descripción:	Función llamada por la función save()	

Tabla 20 Descripción de la Clase del Diseño: BaseRiesgo

Nombre	BaseRiesgo	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_riesgo		
\$id_reporte_estado		
\$riesgo		
\$elemento		
\$probabilidad		
\$impacto		
\$prioridad		
\$descripcion		

Para cada Responsabilidad	
Nombre:	getIdRiesgo()
Descripción:	Devuelve el número identificativo del riesgo.
Nombre:	getIdReporteEstado()
Descripción:	Devuelve el número identificativo del reporte de estado al que pertenece.
Nombre:	getRiesgo()
Descripción:	Devuelve el riesgo.
Nombre:	getElemento()
Descripción:	Devuelve el elemento que caracteriza al riesgo.
Nombre:	getProbabilidad()
Descripción:	Devuelve la probabilidad del riesgo.
Nombre:	getImpacto()
Descripción:	Devuelve el impacto del riesgo.
Nombre:	getPrioridad()
Descripción:	Devuelve la prioridad del riesgo.
Nombre:	getDescripcion()
Descripción:	Devuelve la descripción del riesgo.
Nombre:	setIdRiesgo()
Descripción:	Cambia el número identificativo del riesgo.
Nombre:	setIdReporteEstado
Descripción:	Cambia el número identificativo del reporte de estado al que pertenece.
Nombre:	setRiesgo()
Descripción:	Cambia el riesgo.
Nombre:	setElemento()
Descripción:	Cambia el elemento que caracteriza al riesgo.
Nombre:	setProbabilidad()
Descripción:	Cambia la probabilidad del riesgo.
Nombre:	setImpacto()
Descripción:	Cambia el impacto del riesgo.
Nombre:	setPrioridad()
Descripción:	Cambia la prioridad del riesgo.
Nombre:	setDescripcion()
Descripción:	Cambia la descripción del riesgo.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 21 Descripción de la Clase del Diseño: BaseRol

Nombre	BaseRol
--------	---------

Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_rol	
\$rol	
\$descripcion	
Para cada Responsabilidad	
Nombre:	getIdRol()
Descripción:	Devuelve el número identificativo del rol.
Nombre:	getRol()
Descripción:	Devuelve el rol.
Nombre:	getDescripcion()
Descripción:	Devuelve la descripción de un rol.
Nombre:	setIdRol()
Descripción:	Cambia el número identificativo del rol.
Nombre:	setRol()
Descripción:	Cambia el rol.
Nombre:	setDescripcion()
Descripción:	Cambia la descripción del rol.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	doSave()
Descripción:	Función llamada por la función save()

Tabla 22 Descripción de la Clase del Diseño: BaseTareas

Nombre	BaseTareas
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_tareas	
\$id_reporte_estado	
\$tarea	
\$elemento	
\$impacto	
\$fecha	
\$estado	
\$solucion	
Para cada Responsabilidad	
Nombre:	getIdTareas()
Descripción:	Devuelve el número identificativo de la tarea.
Nombre:	getIdReporteEstado()
Descripción:	Devuelve el número identificativo del reporte de estado al que pertenece.

Nombre:	getTarea()
Descripción:	Devuelve la tarea.
Nombre:	getElemento()
Descripción:	Devuelve el elemento que caracteriza a la tarea.
Nombre:	getImpacto()
Descripción:	Devuelve el impacto de la tarea.
Nombre:	getFecha()
Descripción:	Devuelve la fecha de la tarea.
Nombre:	getEstado()
Descripción:	Devuelve el estado de la tarea.
Nombre:	getSolucion()
Descripción:	Devuelve la solución a la tarea.
Nombre:	setIdTareas()
Descripción:	Cambia el número identificativo de la tarea.
Nombre:	setIdReporteEstado()
Descripción:	Cambia el número identificativo del reporte de estado al que pertenece.
Nombre:	setTarea()
Descripción:	Cambia la tarea.
Nombre:	setElemento()
Descripción:	Cambia el elemento que caracteriza a la tarea.
Nombre:	setImpacto()
Descripción:	Cambia el impacto de la tarea.
Nombre:	setFecha()
Descripción:	Cambia la fecha de la tarea.
Nombre:	setEstado()
Descripción:	Cambia el estado de la tarea.
Nombre:	setSolucion()
Descripción:	Cambia la solución de la tarea.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 23 Descripción de la Clase del Diseño: BaseUsuario

Nombre	BaseUsuario	
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)	
Atributo	Tipo	
\$id_usuario		
\$id_areas		
\$id_rol		

\$usuario	
Para cada Responsabilidad	
Nombre:	getIdUsuario()
Descripción:	Devuelve el número identificativo del usuario.
Nombre:	getIdAreas()
Descripción:	Devuelve el número identificativo del área al que pertenece.
Nombre:	getIdRol()
Descripción:	Devuelve el número identificativo del rol del usuario.
Nombre:	getUsuario()
Descripción:	Devuelve el usuario.
Nombre:	setIdUsuario()
Descripción:	Cambia el número identificativo del usuario.
Nombre:	setIdAreas()
Descripción:	Cambia el número identificativo del área al que pertenece.
Nombre:	setIdRol()
Descripción:	Cambia el número identificativo del rol del usuario.
Nombre:	setUsuario()
Descripción:	Cambia el usuario.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 25 Descripción de la Clase del Diseño: BaseReporteEstado

Nombre	BaseReporteEstado
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_reporte_estado	
\$id_usuario	
\$cambioshitos	
\$descripcioncambios	
\$elementopresupuesto	
\$presupuesto	
\$costoactual	
\$varianza	
Para cada Responsabilidad	
Nombre:	getIdReporteEstado()
Descripción:	Devuelve el número identificativo del reporte de estado.
Nombre:	getIdUsuario()
Descripción:	Devuelve el número identificativo del usuario que emite el reporte.

Nombre:	getCambioshitos()
Descripción:	Devuelve los cambios en los hitos.
Nombre:	getDescripcioncambios()
Descripción:	Devuelve la descripción de los cambios de los hitos.
Nombre:	getElementopresupuesto()
Descripción:	Devuelve el elemento del presupuesto.
Nombre:	getPresupuesto()
Descripción:	Devuelve el presupuesto.
Nombre:	getCostoactual()
Descripción:	Devuelve el costo actual del proyecto.
Nombre:	getVarianza()
Descripción:	Devuelve la varianza.
Nombre:	setIdReporteEstado()
Descripción:	Cambia el número identificativo del reporte de estado.
Nombre:	setIdUsuario()
Descripción:	Cambia el número identificativo del usuario que emite el reporte.
Nombre:	setCambioshitos()
Descripción:	Cambia los cambios en los hitos.
Nombre:	setDescripcioncambios()
Descripción:	Cambia la descripción de los cambios en los hitos.
Nombre:	setElementopresupuesto()
Descripción:	Cambia el elemento del presupuesto.
Nombre:	setPresupuesto()
Descripción:	Cambia el presupuesto del proyecto.
Nombre:	setCostoactual()
Descripción:	Cambia el costo actual del proyecto.
Nombre:	setVarianza()
Descripción:	Cambia la varianza.
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 26 Descripción de la Clase del Diseño: BaseReportePostMortem

Nombre	BaseReportePostmortem
Tipo de Clase	Clase Base(Clase que se genera a partir del esquema)
Atributo	Tipo
\$id_reporte_postMortem	
\$id_usuario	
\$ejeContraObjetivo	

\$ejeContraPrograma	
\$ejeContraCalidad	
\$ejeContraPresupuesto	
\$ReporteAreas	
\$lecciones_aprendidas	
\$recomendaciones	
Para cada Responsabilidad	
Nombre:	getIdReporte postMortem()
Descripción:	Devuelve el número identificativo del reporte postmortem.
Nombre:	getIdUsuario()
Descripción:	Devuelve el número identificativo del usuario que emite el reporte.
Nombre:	getEjeContraObjetivo()
Descripción:	Devuelve los objetivos del proyecto.
Nombre:	getEjeContraPrograma()
Descripción:	Devuelve el resumen del programa original del proyecto.
Nombre:	getEjeContraCalidad()
Descripción:	Devuelve los objetivos iniciales de la calidad del proyecto.
Nombre:	getEjeContraPresupuesto()
Descripción:	Devuelve los objetivos iniciales del presupuesto del proyecto.
Nombre:	getReporteAreas()
Descripción:	Devuelve el reporte por areas.
Nombre:	getLeccionesAprendidas()
Descripción:	Devuelve las lecciones aprendidas en el desarrollo del proyecto.
Nombre:	getRecomendaciones()
Descripción:	Devuelve las recomendaciones.
Nombre:	setIdReporte postMortem()
Descripción:	Cambia el número identificativo del reporte postmortem.
Nombre:	getIdUsuario()
Descripción:	Cambia el número identificativo del usuario que emite el reporte.
Nombre:	setEjeContraObjetivo()
Descripción:	Cambia los objetivos del proyecto.
Nombre:	setEjeContraPrograma()
Descripción:	Cambia el resumen del programa original del proyecto.
Nombre:	setEjeContraCalidad()
Descripción:	Cambia los objetivos iniciales de la calidad del proyecto.
Nombre:	setEjeContraPresupuesto()
Descripción:	Cambia los objetivos iniciales del presupuesto del proyecto.
Nombre:	setReporteAreas()
Descripción:	Cambia el reporte por areas.
Nombre:	setLeccionesAprendidas()
Descripción:	Cambia las lecciones aprendidas en el desarrollo del proyecto.

Nombre:	setRecomendaciones()
Descripción:	Cambia las recomendaciones.
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 27 Descripción de la Clase del Diseño: AreasPeer

Nombre:	AreasPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 28 Descripción de la Clase del Diseño: PoloPeer

Nombre:	PoloPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 29 Descripción de la Clase del Diseño: ActividadPeer

Nombre:	ActividadPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	

Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.
--------------------	---

Tabla 30 Descripción de la Clase del Diseño: CriterioDeUsuarioPeer

Nombre:	CriterioDeUsuarioPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 31 Descripción de la Clase del Diseño: CriterioEvaluacionPeer

Nombre:	CriterioEvaluacionPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 32 Descripción de la Clase del Diseño: HitosPeer

Nombre:	HitosPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 33 Descripción de la Clase del Diseño: ProyectoPeer

Nombre:	ProyectoPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 34 Descripción de la Clase del Diseño: RiesgoPeer

Nombre:	RiesgoPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 35 Descripción de la Clase del Diseño: RolPeer

Nombre:	RolPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Para cada responsabilidad:		
Nombre	doSelect()	
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.	
Nombre	doDelete()	
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.	

Tabla 36 Descripción de la Clase del Diseño: TareasPeer

Nombre:	TareasPeer	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo

Para cada responsabilidad:	
Nombre	doSelect()
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.
Nombre	doDelete()
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.

Tabla 37 Descripción de la Clase del Diseño: UsuarioPeer

Nombre:	UsuarioPeer
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo
Para cada responsabilidad:	
Nombre	doSelect()
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.
Nombre	doDelete()
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.

Tabla 38 Descripción de la Clase del Diseño: ReporteEstadoPeer

Nombre:	ReporteEstadoPeer
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo
Para cada responsabilidad:	
Nombre	doSelect()
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.
Nombre	doDelete()
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.

Tabla 39 Descripción de la Clase del Diseño: ReportePostMortemPeer

Nombre:	ReportePostMortemPeer
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo
Para cada responsabilidad:	
Nombre	doSelect()
Descripción	Permite obtener un listado de objetos con todos los datos del áreas que se encuentran en las tablas (en dependencia del objeto tipo Criterio que se le pase por parámetro.

Nombre	doDelete()
Descripción	Permite eliminar pasándole como parámetro un valor, de la entidad área.

Tabla 40 Descripción de la Clase del Diseño: areasActions

Nombre:	areasActions
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo
Para cada responsabilidad:	
Nombre	executeAdd_Area()
Descripción	Permite adicionar un área.
Nombre	executeModificar_Area()
Descripción	Permite modificar un área determinada.
Nombre	executeEliminar_Area()
Descripción	Permite eliminar un área determinada.
Nombre	executeIndex()
Descripción	Indica la creación de la página principal donde se muestra un listado de las áreas.

Tabla 41 Descripción de la Clase del Diseño: criterioEvaluacionActions

Nombre:	criterioEvaluacionActions
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo
Para cada responsabilidad:	
Nombre	executeAdd_CriterioValoracion()
Descripción	Permite adicionar un criterio de valoración.
Nombre	executeModificar_CriterioValoracion ()
Descripción	Permite modificar un criterio de valoración determinado.
Nombre	executeEliminar_CriterioValoracion ()
Descripción	Permite eliminar un criterio de valoración determinado.
Nombre	executeInsertar()
Descripción	Permite insertar un criterio de valoración.
Nombre	executeIndex()
Descripción	Indica la creación de la página principal donde se muestra un listado de los criterios de Valoración.

Tabla 42 Descripción de la Clase del Diseño: proyectoActions

Nombre:	proyectoActions
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)
Atributo	Tipo

Para cada responsabilidad:	
Nombre	executeAdd_Proyecto()
Descripción	Permite adicionar un proyecto.
Nombre	executeModificar_Proyecto()
Descripción	Permite modificar un proyecto determinada.
Nombre	executeEliminar_Proyecto()
Descripción	Permite eliminar un proyecto determinada.
Nombre	executeIndex()
Descripción	Indica la creación de la página principal donde se muestra un listado de los proyecto.

Tabla 43 Descripción de la Clase del Diseño: rolActions

Nombre:	rolActions	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo	Tipo	
Para cada responsabilidad:		
Nombre	executeAdd_Rol()	
Descripción	Permite adicionar un rol.	
Nombre	executeModificar_Rol()	
Descripción	Permite modificar un rol determinada.	
Nombre	executeEliminar_Rol()	
Descripción	Permite eliminar un rol determinada.	
Nombre	executeIndex()	
Descripción	Indica la creación de la página principal donde se muestra un listado de los roles.	

Tabla 44 Descripción de la Clase del Diseño: criterioDeUsuarioActions

Nombre:	criterioDeUsuarioActions	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo	Tipo	
Para cada responsabilidad:		
Nombre	executeAdd_CriterioDeUsuario()	
Descripción	Permite adicionar un criterio de usuario.	
Nombre	executeModificar_CriterioDeUsuario()	
Descripción	Permite modificar un criterio de usuario determinado.	
Nombre	executeEliminar_CriterioDeUsuario()	
Descripción	Permite eliminar un criterio de usuario determinado.	
Nombre	executeInsertar()	
Descripción	Permite insertar un criterio de usuario.	
Nombre	executeIndex()	

Descripción	Indica la creación de la página principal donde se muestra un listado de los criterios de usuario.
--------------------	--

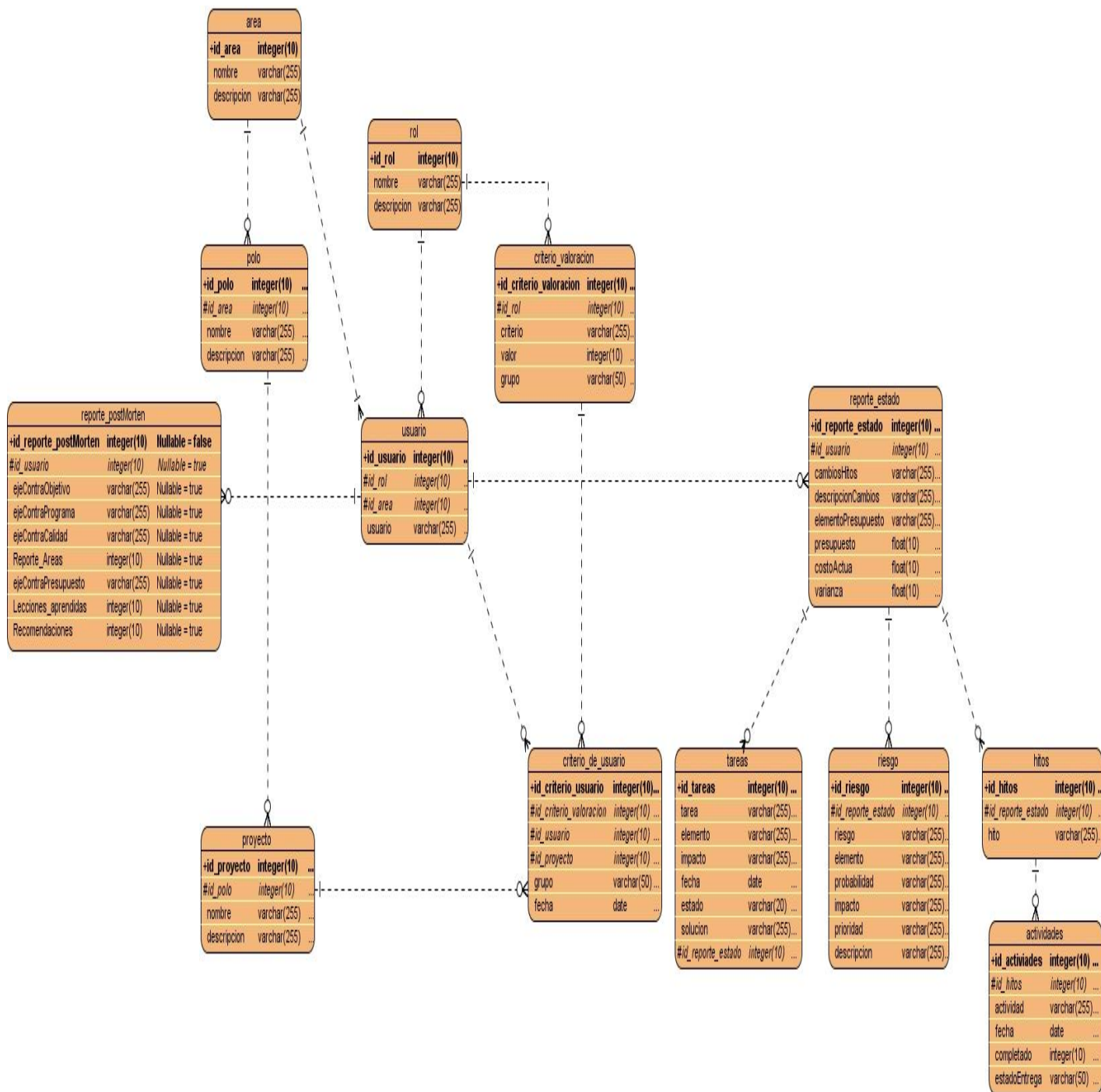
Tabla 45 Descripción de la Clase del Diseño: poloActions

Nombre:	poloActions	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo	Tipo	
Para cada responsabilidad:		
Nombre	executeAdd_Polo()	
Descripción	Permite adicionar un polo.	
Nombre	executeModificar_Polo()	
Descripción	Permite modificar un polo determinada.	
Nombre	executeEliminar_Polo()	
Descripción	Permite eliminar un polo determinada.	
Nombre	executeIndex()	
Descripción	Indica la creación de la página principal donde se muestra un listado de los polos.	

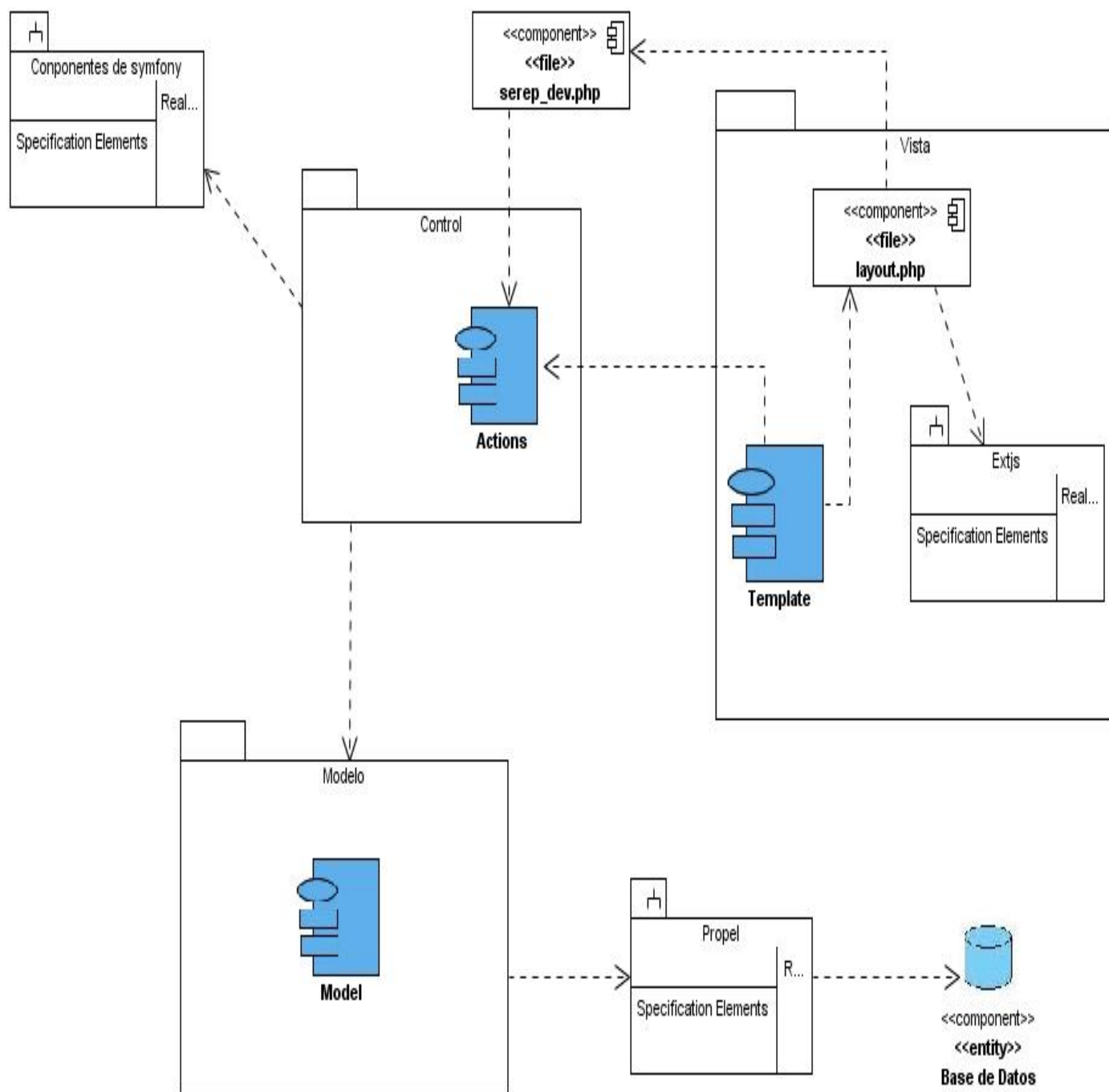
Tabla 46 Descripción de la Clase del Diseño: usuarioActions

Nombre:	usuarioActions	
Tipo de Clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo	Tipo	
Para cada responsabilidad:		
Nombre	executeAdd_Usuario()	
Descripción	Permite adicionar un usuario.	
Nombre	executeModificar_Usuario()	
Descripción	Permite modificar un usuario determinada.	
Nombre	executeEliminar_Usuario()	
Descripción	Permite eliminar un usuario determinada.	
Nombre	executeIndex()	
Descripción	Indica la creación de la página principal donde se muestra un listado de los usuarios.	

Anexo 9: Diseño de la BD



Anexo 10: Diagrama de componentes



Anexo 11: Resultados de Pruebas de rendimiento

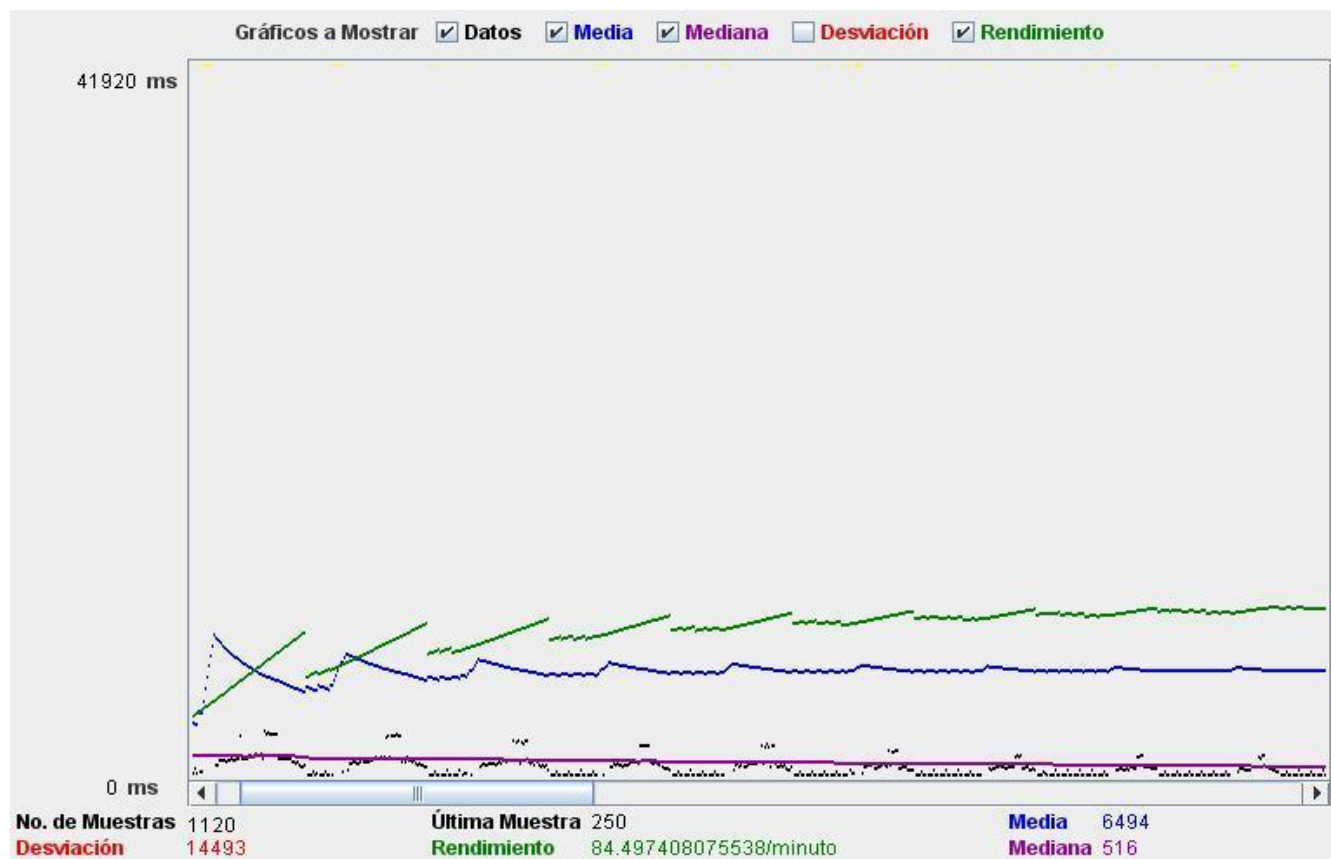


Figura 7 Carga de 150 usuarios

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Escribir en Log Sólo Errores

Label	# Muestras	Media	Mediana	Línea de 90...	Mín	Máx	% Error	Rendimiento	Kb/sec
/tesis/web/eval...	20	1233	1359	1469	297	1532	0,00%	2,6/sec	101073,6
/tesis/web/eval...	10	3007	3016	3203	2859	3203	0,00%	1,2/sec	128374,2
/tesis/web/eval...	10	860	859	1125	657	1125	0,00%	3,2/sec	181660,1
/safebrowsingf/...	10	42002	42002	42096	41956	42096	100,00%	7,0/min	228,0
/tesis/web/eval...	10	482	594	812	156	812	0,00%	3,3/sec	117871,3
/tesis/web/eval...	10	456	531	750	140	750	0,00%	3,3/sec	116467,8
TOTAL	70	7039	1281	41987	140	42096	14,29%	46,0/min	34352,0

Figura 8 Concurrencia de 10 usuarios

Label	# Muestras	Media	Mediana	Línea de 90...	Mín	Máx	% Error	Rendimiento	Kb/sec
/tesis/web/eval...	320	590	563	1266	171	1532	0,00%	22,0/min	14529,0
/tesis/web/eval...	160	1333	1234	2937	359	3203	0,00%	11,0/min	19145,1
/tesis/web/eval...	160	438	437	844	140	1188	0,00%	11,1/min	10477,2
/safebrowsing/...	160	41974	41968	42067	41841	42138	100,00%	10,1/min	328,2
/tesis/web/eval...	160	377	188	797	140	1047	0,00%	11,1/min	6534,2
/tesis/web/eval...	160	396	250	875	140	1125	0,00%	11,1/min	6591,3
TOTAL	1120	6528	562	41950	140	42138	14,29%	1,2/sec	52454,5

Figura 9 Concurrencia de 15 usuarios

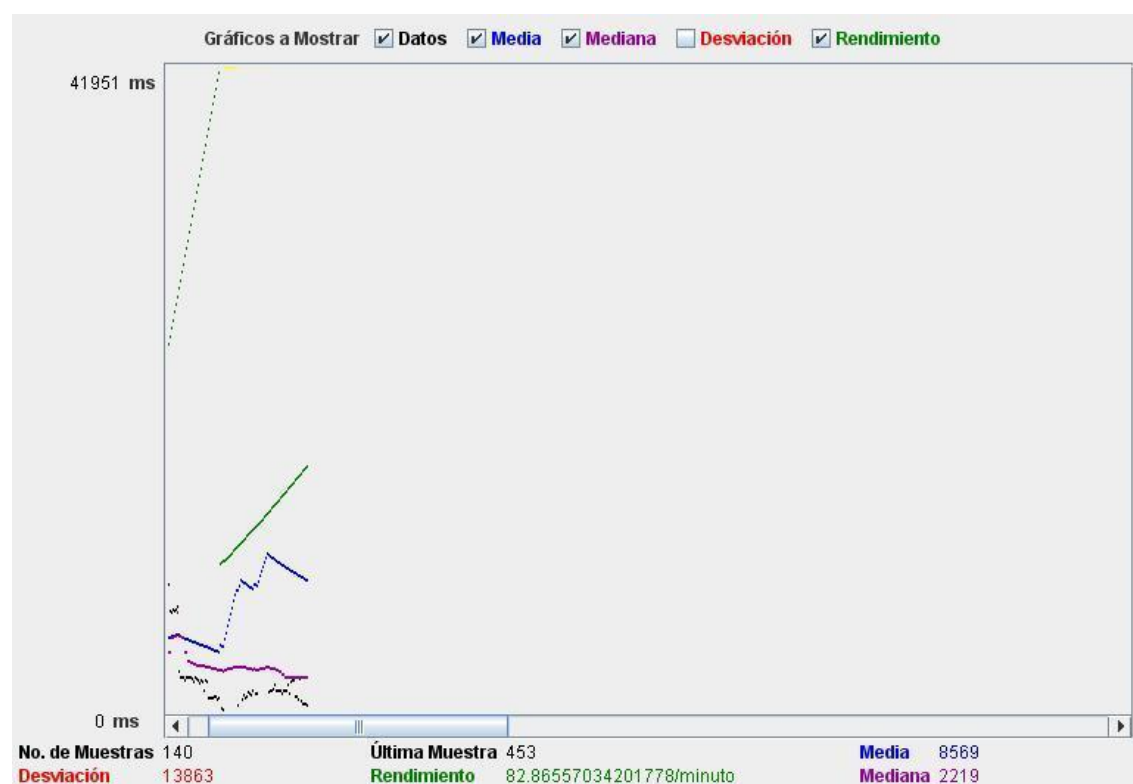


Figura 10 Concurrencia de 25 usuarios

Anexo 12: Diseño de Casos de Prueba

Secciones a probar en el Caso de Uso Autenticar Usuario:

SC 1: CU Autenticar Usuario

Id de escenario	Escenario	Usuario	Contraseña	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para el CU Autenticar Usuario.	V	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio
		F	V		

Secciones a probar en el Caso de Uso Gestionar Área:

SC 1: Adicionar Área del CU Gestionar Área

Id de escenario	Escenario	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Adicionar Área del CU Gestionar Área.	V	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio
		F	V		

SC 2: Modificar Área del CU Gestionar Área

Id de escenario	Escenario	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Modificar	V	F	No permite la inserción de	Satisfactorio

	Área del CU Gestionar Área.	F	V	datos inválidos o que se deje algún campo obligatorio sin llenar.	
--	--------------------------------	----------	----------	---	--

Secciones a probar en el Caso de Uso Gestionar Polo:

SC 1: Adicionar Polo del CU Gestionar Polo

Id de escenario	Escenario	Área	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Adicionar Polo del CU Gestionar Polo.	V	F	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio
		F	V	F		
		F	F	V		

SC 2: Modificar Polo del CU Gestionar Polo

Id de escenario	Escenario	Área	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Modificar Polo del CU Gestionar Polo.	V	F	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio
		F	V	F		
		F	F	V		

Secciones a probar en el Caso de Uso Gestionar Proyecto:

SC 1: Adicionar Proyecto del CU Gestionar Proyecto

Id de escenario	Escenario	Polo	Nombre	Lista de usuarios	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Adicionar Proyecto del CU Gestionar Proyecto.	V	F	F	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio.
		F	V	F	F		
		F	F	V	F		
		F	F	F	V		

SC 2: Modificar Proyecto del CU Gestionar Proyecto

Id de escenario	Escenario	Polo	Nombre	Lista de usuarios	Descripción	Respuesta del Sistema	Resultado de la Prueba
EC 1	Flujo básico para la sección Modificar Proyecto del CU Gestionar Proyecto.	V	F	F	F	No permite la inserción de datos inválidos o que se deje algún campo obligatorio sin llenar.	Satisfactorio
		F	V	F	F		
		F	F	V	F		
		F	F	F	V		

Anexo 13: Fotos del sistema

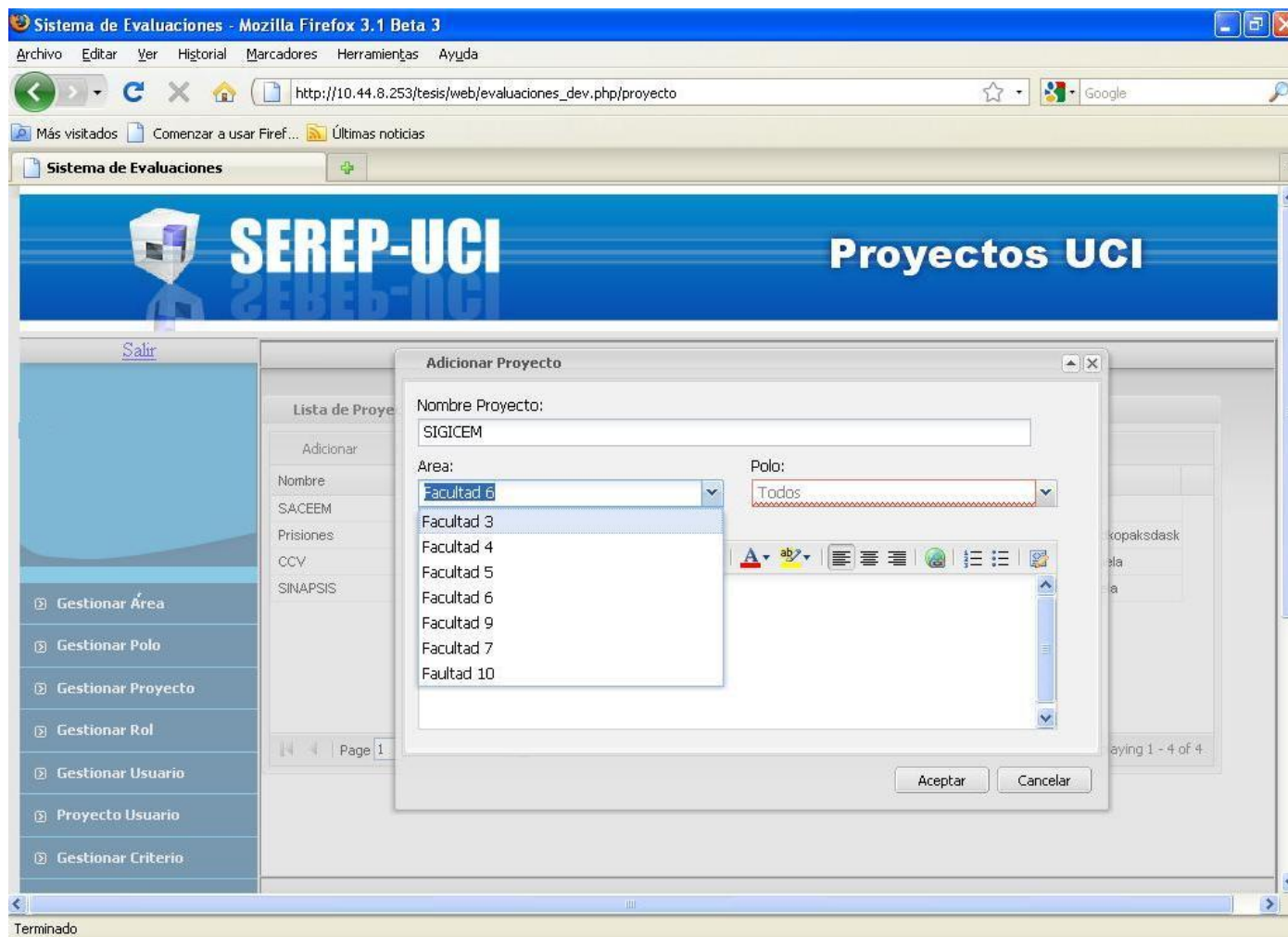


Figura 11 Funcionalidad Adicionar Proyecto

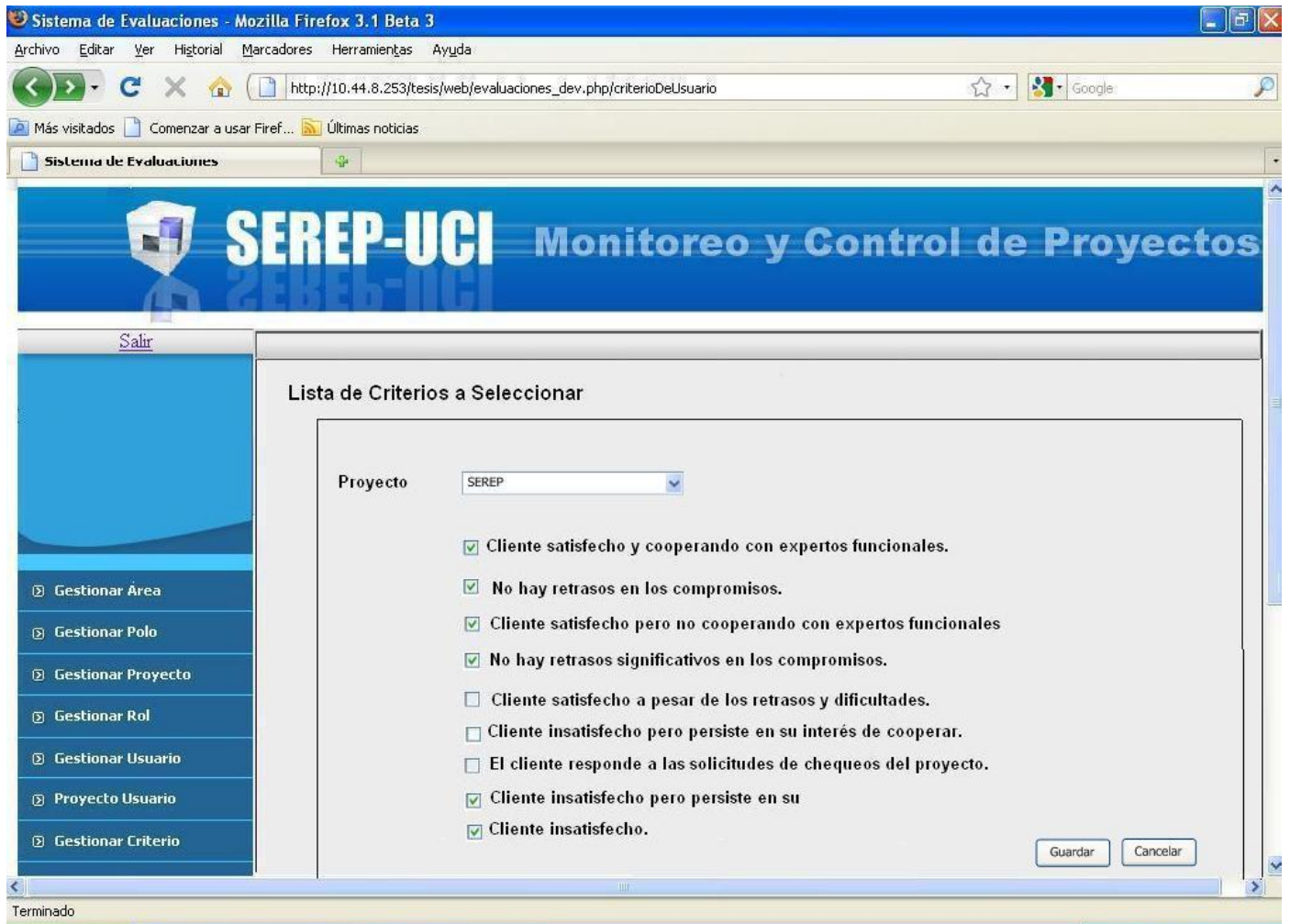


Figura 12 Funcionalidad Emitir Criterios de Valoración

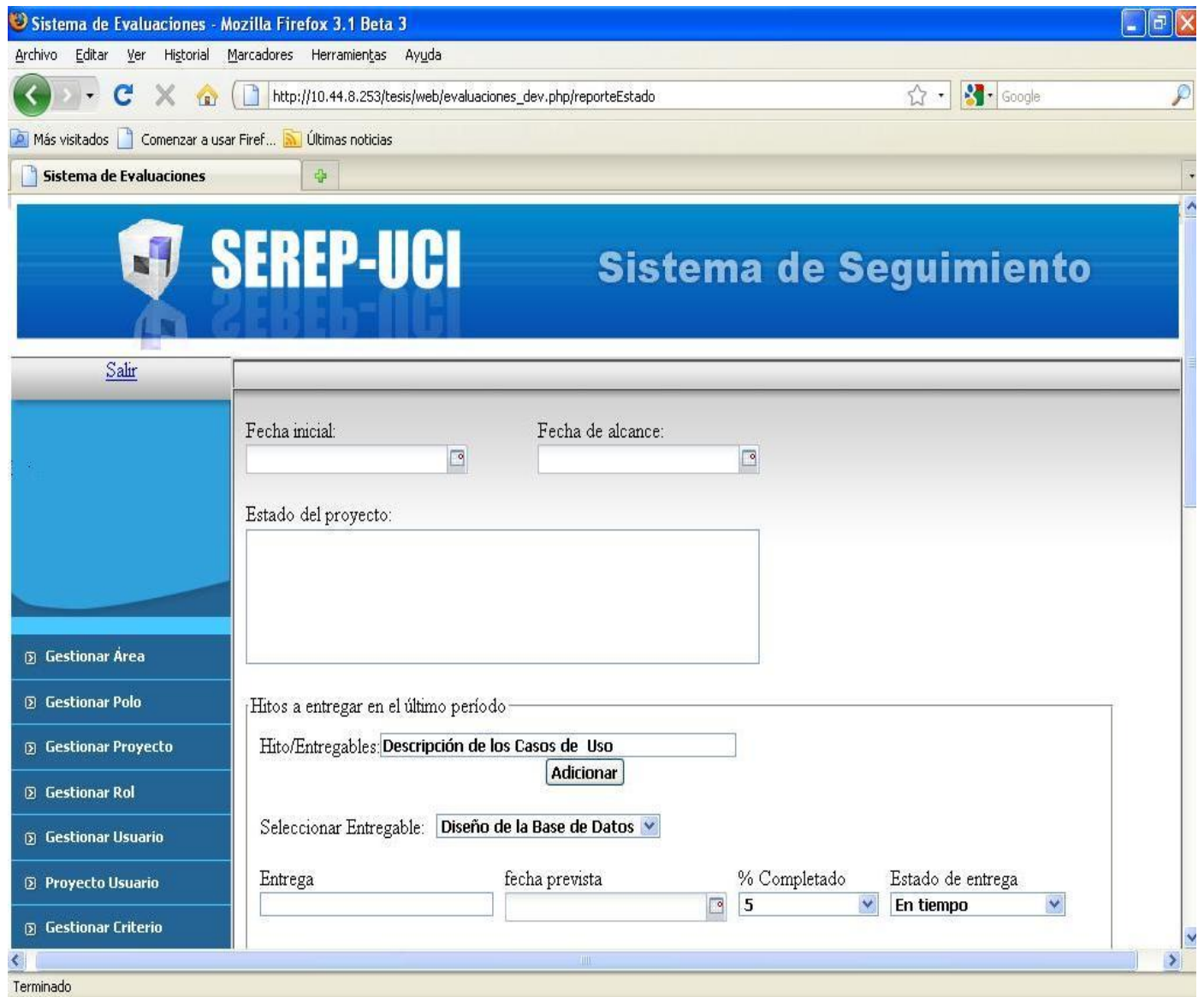


Figura 13 Funcionalidad Registrar Reporte de Estado



Figura 13 Funcionalidad Registrar Reporte Postmortem