

Universidad de las Ciencias Informáticas

Facultad # 8



Título: “Sistema para la Gestión de Servidores de Salvas de los Proyectos Productivos en la Universidad de las Ciencias Informáticas.”

Trabajo para optar por el Título de Ingeniería en Ciencias Informáticas

Autores: Odalis Bermúdez López

Yasser Fontaine Macías

Tutor: Ing. Frank David Avalos Palomo

Co-Tutor: Ing. Armando Ortiz Cabrera

La ciencia es universal y tenemos que aprender del mundo, de la misma manera que debemos estar siempre dispuestos a mostrar el aporte que nosotros podemos obtener para ellos.

Fidel Castro Ruz

*A mis padres queridos con todo mi amor,
Por haber dedicado toda su vida a mi formación,
Por haberme apoyado en todos los momentos buenos y difíciles de mi vida,
Por confiar plenamente en mí, en que el sueño de todos nosotros se hiciera realidad.*

Odalís

A mis padres y a todos los que me han apoyado.

Yasser

De Odalis:

En estos momentos me gustaría agradecerles a muchas personas que de una forma u otra han hecho posible que este sueño se hiciera realidad, aquellos que han confiado en mí y me han apoyado y ayudado hasta hoy. Me gustaría agradecerle a:

Mi hermano querido (**Orlandito**) que me ha ayudado a cumplir este maravilloso sueño, por ser una inspiración para el desarrollo de este trabajo que sé que es de ambos.

A Júa (**Julia Fernández**) y Pupa (**Julia Estela**) por estar siempre ahí desde que nací presente en todos los momentos buenos y difíciles de mi vida, a ellas que siempre me han apoyado y aconsejado.

A mi esposo (**Armando Ortiz Cabrera**) por todo su apoyo incondicional en estos cinco años y por haberme dado fuerzas para lograr lo que hasta ahora he alcanzado. Por co-tutoriarme la tesis, por mantenerse siempre a nuestro lado, pendientes de cada detalle de nuestra tesis. Muchas gracias por todo el apoyo incondicional que me has brindado, gracias por ser tan especial y por ayudarme a que este sueño se hiciera realidad.

A mis suegros (**Edelmira Cabrera y Armando Ortiz**) que les agradezco todo el cariño sincero que me han dado, por todo su apoyo y quererme como una hija.

A mi cuñada (**Yunelsy Ortiz Cabrera**), por todo la ayuda y apoyo que me ha dado, por ser tan buena conmigo y por sus consejos como una hermana.

A mi amiga mama (**Teresa**) que siempre me ha ayudado y aconsejado para seguir adelante para que este sueño se me hiciera realidad.

A mis compañeros de aula que han aportado su granito de arena en la realización de este trabajo.

A mis compañeras de cuarto por haber compartido estos cinco años y haber sido siempre como una familia.

Agradecer el apoyo infinito que nos ha brindado mi tutor Frank David Avalos Palomo, por guiarnos y estar siempre de nuestro lado, muchas gracias por ayudarnos a realizar este trabajo.

A Yulio Seriocha por estar pendiente al desarrollo de este trabajo y por siempre guiarnos.

A aquellas personas que no se encuentran en estos momentos con nosotros pero siempre quisieron que este día llegara.

Gracias a nuestro invencible comandante en jefe Fidel Castro Ruz y a la Revolución por ofrecernos la posibilidad de prepararnos profesionalmente logrando alcanzar mi sueño, el título de Ingeniera en Ciencias Informáticas.

En fin agradecerles a todos aquellos que me han apoyado durante este largo andar, a todos ustedes...,

Muchas gracias de todo corazón por hacer este sueño hoy realidad.

De Yasser:

Gracias al Comandante por sus ideas.

Gracias a la Revolución por las oportunidades.

Gracias a mi mamá por sus sacrificios.

Gracias a mi papá por su confianza.

Gracias a Indira, mi prima.

Gracias a Frank, mi tutor.

Gracias Tay.

A todos mis familiares, profesores y compañeros (los mejores del mundo).....

DEDICATORIA DE LA 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 _____.

Odalís Bermúdez López
Autor

Yasser Fontaine Macías
Autor

Ing. Frank David Avalos Palomo
Tutor

RESUMEN

En la Universidad de las Ciencias Informáticas (UCI), se lleva a cabo un proceso fundamental referente a los registros de proyectos: Solicitar Registro de Salvas. En la ejecución del mismo se manipula una cantidad de información muy variada y de carácter sensible.

Este trabajo analiza la propuesta de un sistema que informatize este proceso. Actualmente no se cuenta con un soporte informático en la UCI que brinde rapidez en el manejo de la información y consistencia en los datos.

En el presente trabajo se dan argumentos necesarios para demostrar que la situación problemática requiere de un sistema que cumpla con los requisitos propuestos para satisfacer las necesidades identificadas. Posteriormente se hace un análisis comparativo acerca de las tecnologías y herramientas existentes y se seleccionan las más apropiadas. Para su desarrollo se siguieron los pasos que proponen James Rumbaugh, Ivar Jacobson y Grady Booch autores del Proceso Unificado de Desarrollo de Software.

Se muestran los resultados del diseño de la propuesta del sistema, se desarrolla bajo el concepto de software libre y de fácil manejo. Algunas de estas tecnologías y herramientas son: PHP5 y Java Script como lenguaje de programación apoyándose en PostgreSQL como Gestor de Base de Datos, Entorno de Desarrollo Integrado Eclipse, Framework Symfony, Herramienta Case Visual Paradigm y como Metodología de Desarrollo, RUP. Este documento expone los resultados de todo el trabajo investigativo realizado.

ABSTRACT

In the Informatics Sciences University, a key process concerning to projects records is taking place: Save Records Request. Quite varied and sensitive information is manipulated when executing this process.

This job analyze the proposal of an system for computerizing this process. Because currently does not exist in the UCI an informatic support that provide a fast information management and consistent data.

This job provide the necessary arguments to demonstrate that the situation requires the development of a system that accomplish with proposed requirements in order to satisfy the identified needs, providing also a comparison among the different tools and technologies in favor of selecting the most appropriate. We had followed the following steps proposed by James Rumbaugh, Ivar Jacobson y Grady Booch, authors of Unified Process of Software Development.

The system proposal design results are shown; development is framed on free software concepts and easy management. Some of the selected technologies are: PHP5 and JavaScript as programming languages supporting in PostgreSQL as Database Manager, Eclipse IDE, Symfony Framework, Visual Paradigm as case tool and finally as development methodology, RUP. All investigative work results are cover in this document.

Índice

ÍNDICE DE TABLAS	XX
ÍNDICE DE FIGURAS	XXI
INTRODUCCIÓN.....	1
Situación problemática	2
Métodos teóricos.	3
Métodos Empíricos.	4
CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA	6
1.1. INTRODUCCIÓN.....	6
1.2. CONCEPTOS ESENCIALES	6
1.2.1. Aplicación Informática	6
1.2.2. Aplicación Web	8
1.2.3. Sistema.....	8
1.2.4. Sistema informático	9
1.2.5. Sistema de gestión	10
1.2.6. Servidor	10
1.2.7. Proyecto.....	10
1.2.8. Proyecto productivo	11
1.3. LA PÉRDIDA DE DATOS.....	11
1.4. ¿QUÉ SON LAS COPIAS DE SEGURIDAD?.....	12
1.5. HISTORIA DE LAS COPIAS DE SEGURIDAD.....	13
1.5.1. Lista de software para respaldo	13
1.5.1.1. Software libre.....	13
1.5.1.2. Software privativo	14

1.5.2. Descripción de algunas herramientas utilizadas para realizar respaldos.....	15
1.6. TENDENCIAS Y TECNOLOGIAS ACTUALES.....	17
1.6.1. LENGUAJES DE PROGRAMACIÓN WEB.....	17
1.6.1.1. Lenguajes del lado del servidor.....	17
1.6.1.2. Lenguajes del lado del cliente.....	20
1.6.2. SERVIDORES WEB.....	21
1.6.2.1. Servidor Apache.....	21
1.6.2.2. Servidor Web Internet Information Server(IIS).....	23
1.6.3. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	24
1.6.3.1. Proceso Unificado de Desarrollo de Software (RUP).....	24
1.6.3.2. Programación Extrema (XP).....	27
1.6.4. LENGUAJE UNIFICADO DE MODELO (UML).....	28
1.6.5. HERRAMIENTA CASE DE DESARROLLO DE SOFTWARE.....	30
1.6.5.1. Rational Rose.....	31
1.6.5.2. Visual Paradigm.....	31
1.6.6. SISTEMA GESTORES DE BASES DE DATOS.....	32
1.6.6.1. PostgreSQL.....	32
1.6.6.2. MySQL.....	35
1.6.6. FRAMEWORK.....	36
1.6.6.1. Framework Symfony.....	36
1.6.6.2. Prado Framework.....	38
1.6.7. ENTORNO INTEGRADO DE DESARROLLO (IDE).....	38
1.6.7.1. Zend Studio.....	39
1.6.7.2. Eclipse.....	39
1.6.8. JUSTIFICACIÓN DE LA SELECCIÓN.....	41

1.7. PROPUESTA DE SOLUCIÓN.....	43
1.8. CONCLUSIONES PARCIALES.....	43
CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	44
2.1. INTRODUCCIÓN.....	44
2.2. MODELO DEL NEGOCIO PROPUESTO	44
2.2.1. Reglas generales del negocio	45
2.2.2. Actores del Negocio	45
2.2.3. Trabajadores del Negocio	46
2.2.4. Diagrama de Caso de Uso del Negocio	46
2.3. DESCRIPCIÓN DE LOS PROCESOS DEL NEGOCIO PROPUESTO	47
2.3.1. Descripción del Caso de Uso de Negocio: Solicitar Registro de Salva.	47
2.4. MODELO DE OBJETO.....	48
2.5. REQUISITOS DEL SISTEMA.....	48
2.5.1. Requerimientos funcionales	49
2.5.2. Requerimientos No Funcionales	51
2.6. PROPUESTA DE SOLUCIÓN.....	51
2.6.1. Actores del Sistema.....	53
2.6.2. Diagrama de Casos de Uso del Sistema (DCUS).....	54
2.7. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA	55
2.7.1. Descripción del Caso de Uso: Autenticar Usuario	55
2.7.2. Descripción del Caso de Uso: Gestionar Usuario	57
2.7.3. Descripción del Caso de Uso: Gestionar Proyectos	59
2.7.4. Descripción del Caso de Uso: Gestionar Docentes	61
2.7.5. Descripción del Caso de Uso: Gestionar Facultades.....	64
2.7.6. Descripción del Caso de Uso: Gestionar Laboratorios	66

2.7.7. Descripción del Caso de Uso: Gestionar Plataformas	68
2.7.8. Descripción del Caso de Uso: Eliminar Proyecto.....	70
2.7.9. Descripción del Caso de Uso: Eliminar Docente.....	71
2.7.10. Descripción del Caso de Uso: Eliminar Facultad	72
2.7.11. Descripción del Caso de Uso: Eliminar Laboratorio	74
2.7.12. Descripción del Caso de Uso: Eliminar Plataforma.....	75
2.7.13. Descripción del Caso de Uso: Eliminar Registro de Salva	76
2.7.14. Descripción del Caso de Uso: Solicitar Registro de Salva.....	77
2.7.15. Descripción del Caso de Uso: Modificar Registro de Salva	79
2.7.16. Descripción del Caso de Uso: Generar Fichero Bacula.....	80
2.7.17. Descripción del Caso de Uso: Eliminar Fichero Bacula.....	81
2.7.18. Descripción del Caso de Uso: Reporte de Salva	82
2.8. Conclusiones Parciales	83
CAPÍTULO 3 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	84
3.1. INTRODUCCIÓN.....	84
3.2. MODELO DE ANÁLISIS	84
3.3. DIAGRAMAS DE CLASES DEL ANÁLISIS	85
3.3.1. Diagramas de Interacción del Análisis	85
3.4. MODELO DE DISEÑO	86
3.4.1. Diagrama de Clases del Diseño	88
3.5. DISEÑO DE LA BASE DE DATOS	88
3.5.1. Diagrama de Clases Persistentes	89
3.5.2. Diagrama Entidad Relación.....	90
3.6. DIAGRAMA DE DESPLIEGUE	90
3.7. MODELO DE IMPLEMENTACIÓN.....	91

3.8. CONCLUSIONES PARCIALES.....	94
CONCLUSIONES	¡Error! Marcador no definido.
RECOMENDACIONES	XIV
REFERENCIAS BIBLIOGRÁFICAS	XV
BIBLIOGRAFÍA CONSULTADA	XVII
ANEXOS	XVIII
ANEXO 1. REALIZACIÓN DE LOS CASOS DE USO DEL NEGOCIO: DIAGRAMAS DE ACTIVIDADES.	XVIII
Diagrama de Actividades del Caso de Uso: Solicitar Registro de Salva.	XVIII
ANEXO 2. DIAGRAMAS DE CLASES DEL ANÁLISIS	XVIII
Diagramas de Clases del Análisis del Caso de Uso: Autenticar Usuario.	XVIII
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Usuario.	XIX
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Docente.	XIX
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Facultad.....	XX
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Proyecto.	XX
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Laboratorio.	XXI
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Plataforma.	XXII
Diagramas de Clases del Análisis del Caso de Uso: Solicitar Registro de Salva.....	XXII
Diagramas de Clases del Análisis del Caso de Uso: Modificar Registro de Salva.	XXII
Diagramas de Clases del Análisis del Caso de Uso: Eliminar Registro de Salva.....	XXIII
Diagramas de Clases del Análisis del Caso de Uso: Reporte de Salva.	XXIII
ANEXO 3. DIAGRAMAS DE INTERACCIÓN DEL ANÁLISIS.	XXIII
Diagrama de Colaboración del Caso de Uso: Autenticar Usuario.	XXIII
Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Adicionar.	XXV
Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Modificar.	XXV

Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Buscar.....	XXVI
Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Eliminar.....	XXVI
Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Adicionar.....	XXVI
Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Modificar.....	XXVIII
Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Mostrar.....	XXVIII
Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Adicionar.....	XXIX
Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Modificar.....	XXIX
Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Mostrar.....	XXX
Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Adicionar.....	XXX
Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Modificar.....	XXX
Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Mostrar.....	XXXI
Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Adicionar.....	XXXI
Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Modificar.....	XXXII
Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Mostrar.....	XXXIII
Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Adicionar.....	XXXIII
Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Modificar.....	XXXIV
Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Mostrar.....	XXXIV
Diagrama de Colaboración del Caso de Uso: Eliminar Proyecto.....	XXXV
Diagrama de Colaboración del Caso de Uso: Eliminar Docente.....	XXXVI
Diagrama de Colaboración del Caso de Uso: Eliminar Facultad.....	XXXVI
Diagrama de Colaboración del Caso de Uso: Eliminar Laboratorio.....	XXXVII
Diagrama de Colaboración del Caso de Uso: Eliminar Registro de Salva.....	XXXVII
Diagrama de Colaboración del Caso de Uso: Eliminar Plataforma.....	XXXVIII

Diagrama de Colaboración del Caso de Uso: Solicitar Registro de Salva..... XXXVIII

Diagrama de Colaboración del Caso de Uso: Modificar Registro de Salva.....XXXIX

Diagrama de Colaboración del Caso de Uso: Reporte de Registro de Salva.....XXXIX

ANEXO 4. DIAGRAMAS DE CLASES DEL DISEÑO..... XLI

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

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

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

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

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

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

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

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

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Proyecto..... L

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

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

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

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Docente..... LV

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

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

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

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Laboratorio..... LIX

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

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

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

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Facultad. LXIII

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

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

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

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Plataforma. LXVII

Diagrama de Clases del Diseño del Caso de Uso: Solicitar Registro de Salva. LXVIII

Diagrama de Clases del Diseño del Caso de Uso: Modificar Registro de Salva. LXIX

Diagrama de Clases del Diseño del Caso de Uso: Reporte Registro de Salva. LXX

Diagrama de Clases del Diseño del Caso de Uso: Eliminar Registro de Salva. LXXI

GLOSARIO DE TÉRMINOS LXXII

ÍNDICE DE TABLAS

Tabla #1: Programas de respaldo de software libre.....	14
Tabla #2: Programas de respaldo de software privativo.....	15
Tabla #3: Descripción de los Actores del Negocio	46
Tabla #4: Descripción de los Trabajadores del Negocio	46
Tabla #5: Descripción textual del Caso de Uso: Solicitar Registro de Salva.....	48
Tabla #6: Descripción de los Actores del sistema a automatizar	54
Tabla #7: Descripción textual del Caso de Uso: Autenticar Usuario.....	57
Tabla #8: Descripción textual del Caso de Uso: Gestionar Usuario	59
Tabla #9: Descripción textual del Caso de Uso: Gestionar Proyectos	61
Tabla #10: Descripción textual del Caso de Uso: Gestionar Docentes.....	63
Tabla #11: Descripción textual del Caso de Uso: Gestionar Facultades	66
Tabla #12: Descripción textual del Caso de Uso: Gestionar Laboratorios.....	68
Tabla #13: Descripción textual del Caso de Uso: Gestionar Plataformas.....	70
Tabla #14: Descripción textual del Caso de Uso: Eliminar Proyecto.....	71
Tabla #15: Descripción textual del Caso de Uso: Eliminar Docente.....	72
Tabla #16: Descripción textual del Caso de Uso: Eliminar Facultad	74
Tabla #17: Descripción textual del Caso de Uso: Eliminar Laboratorio	75
Tabla #18: Descripción textual del Caso de Uso: Eliminar Plataforma	76
Tabla #19: Descripción textual del Caso de Uso: Eliminar Registro de Salva	77
Tabla #20: Descripción textual del Caso de Uso: Solicitar Registro de Salva	79
Tabla #21: Descripción textual del Caso de Uso: Modificar Registro de Salva	80
Tabla #22: Descripción textual del Caso de Uso: Generar Fichero Bacula.....	81
Tabla #23: Descripción textual del Caso de Uso: Eliminar Fichero Bacula	82
Tabla #24: Descripción textual del Caso de Uso: Reporte de Salva	83

ÍNDICE DE FIGURAS

Fig. #1: Aplicación Informática7

Fig. #2: Diagrama conceptual de sistema9

Fig. #3: Gráfica de uso de Servidores Web en el mundo.....21

Fig. #4: Disciplinas a través de las iteraciones26

Fig. #5: Modelo de Caso de Uso del Negocio.....47

Fig. #6: Modelo de Objetos del Negocio48

Fig. #7: Diagrama de Caso de Uso del Sistema55

Fig. #8: Diagrama de Clases Persistentes89

Fig. #9: Diagrama Entidad Relación.....90

Fig. #10: Diagrama de Despliegue91

Fig. #11: Diagrama de Componentes.....92

Fig. #12: Paquete Model93

Fig. #13: Paquete Módulo93

INTRODUCCIÓN

Desde hace algunos años la industria del software ha sufrido un gran número de cambios, dados esencialmente por el auge en el uso de las tecnologías, lo que ha implicado el aumento de proyectos informáticos a nivel mundial, así como las dimensiones de los mismos. Durante el desarrollo de software se genera un gran volumen de información por lo que se hace necesario realizar salvadas periódicas, que nos permitan posteriormente realizar estudios comparativos con otros proyectos de software.

A mediados de la década del 60 en Cuba se empezaron a desarrollar pequeñas aplicaciones informáticas que contribuían al desarrollo informático. Éstas han tenido pequeños incrementos con la aparición de la empresa DESOFT (Empresa de desarrollo de software); llegando así a una alta productividad en los últimos años con la creación de la Universidad de las Ciencias Informáticas (UCI), se creó con el concepto de: Universidad Productiva. Una de sus misiones es la creación de software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación, logrando una fuerte relación universidad-empresa. Esta universidad que convierte a la producción en sustento económico político y social; es productora desde pequeñas hasta grandes soluciones informáticas y esta comprometida a ser la vanguardia en el desarrollo de software en Cuba, de manera tal que convierta la industria de software en un renglón fundamental de la economía del país.

La UCI tiene retos importantes, dados sus compromisos productivos reales, vinculando la producción a todos sus estudiantes y a un alto porcentaje de sus educadores en proyectos de alto valor tanto para el mercado nacional como para el internacional.

Como entidad desarrolladora de software que pretende insertarse en el mercado internacional, la UCI debe lograr no sólo el desarrollo de soluciones comerciales, siendo esta su principal fuente de ingresos, sino de herramientas internas para poder llevar a cabo estas soluciones, y a medida que vaya creciendo esta gama de recursos, poder obtener productos de mejor calidad en menor tiempo y con un mayor control.

Los proyectos productivos de la UCI se desarrollan actualmente a nivel de facultad, recayendo sobre esta todo el peso y control del mismo. Cada facultad organiza sus actividades productivas de acuerdo a sus necesidades pudiendo encontrar allí todo lo referente al producto en desarrollo.

Al solo encontrarse en el local donde se desarrolla el producto toda la información referente al mismo, se corren muchos riesgos de seguridad. Una catástrofe podría conllevar a un daño parcial o total del mismo. La importancia de disponer de mecanismos preventivos para proteger los datos y poder reanudar las operaciones en caso de desastres, con el menor impacto posible, debido a que por acción de virus, de usuarios, ataques mal intencionados, fallos en el hardware o simplemente por accidentes (incendios, inundaciones, robos) o descuido, la información contenida en los servidores puede resultar dañada o incluso desaparecer.

Situación problemática

En la UCI las salvadas de los proyectos productivos se desarrollan de forma manual, lo que provoca deficiencia en la calidad de los mismos debido a lo engorroso que puede tornarse esto ante la cantidad de información existente lo que supone una alta complejidad y la aparición de tareas engorrosas que deben realizarse para mantener actualizada, consultar y/o modificar dicha información. Además de que está expuesto a desastres naturales lo que representa un alto riesgo para la pérdida de información.

Dada la explicación expuesta anteriormente el **problema a resolver** queda formulado de la siguiente forma:

¿Cómo lograr que la gestión de los Servidores de salvadas de los proyectos productivos en la Universidad de las Ciencias Informáticas se torne menos engorrosa y más rápida?

El **objeto de estudio** de esta investigación lo constituye el proceso de gestión de servidores de salvadas de los proyectos productivos.

El **campo de acción** que abarca esta investigación es la automatización del proceso de gestión de servidores de salvadas de los proyectos productivos en la Universidad de las Ciencias Informáticas.

Se plantea entonces como **objetivo general** desarrollar un sistema que permita la gestión de servidores de salvadas de los proyectos productivos en la Universidad de las Ciencias Informáticas de manera más sencilla y más rápida, y para darle solución al objetivo general se plantea los **objetivos específicos** siguientes:

1. Realizar un estudio del entorno de trabajo.
2. Aplicar la metodología, lenguaje de programación y herramientas adecuadas para la eficiente implementación de la aplicación.
3. Documentar la investigación con vista a la socialización de la información de la misma.

A partir de lo expuesto anteriormente surge la siguiente **hipótesis**:

Si, se desarrolla una aplicación informática que informatice los procesos de gestión de salvadas de los proyectos productivos de la UCI. Entonces, se logrará una mayor gestión y eficiencia de la información de los servidores de salvadas.

Para cumplir con el objetivo general propuesto y darle solución a la situación problemática planteada, se proponen las **tareas** siguientes:

1. Identificar las necesidades de la institución a través de la realización de entrevistas.
2. Identificar los conceptos fundamentales del negocio a informatizar.
3. Describir los procesos del negocio identificados.
4. Caracterizar a nivel mundial y nacional las soluciones existentes que soportan los procesos descritos en el negocio a informatizar.
5. Caracterizar las tecnologías a utilizar en el desarrollo del producto.
6. Diseñar la aplicación de forma modular haciendo uso de la metodología RUP como proceso iterativo e incremental y el modelado a través del lenguaje de UML.
7. Implementar el sistema para la gestión de salvadas de los proyectos productivos.

Métodos Científicos usados en la investigación.

Métodos teóricos.

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

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

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

Métodos Empíricos.

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

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

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

El presente documento está estructurado por 4 capítulos, a continuación se expone brevemente una descripción de cada uno.

Capítulo 1: Se abordará en detalle todo lo relacionado con la fundamentación teórica que sustenta la presente investigación, se hará un estudio del estado del arte del tema y se especificarán algunos conceptos asociados a la misma. Además de fundamentar la elección de las tecnologías y herramientas que se utilizarán para dar cumplimiento a los objetivos propuestos.

Capítulo 2: Se profundizará en el Negocio especificando los actores, trabajadores, casos de uso, descripciones y diagramas correspondientes, así como los requerimientos propios del sistema, tanto

funcionales como no funcionales, realizando también las descripciones y la documentación correspondientes.

Capítulo 3: Se desarrollarán los artefactos pertenecientes a los flujos de trabajo Análisis, Diseño e Implementación, dígame diagramas de clases, diagramas de interacción, diseño de la base de datos, modelo de despliegue y el diagrama de componentes.

1

CAPÍTULO FUNDAMENTACIÓN DEL TEMA

1.1. Introducción.

En este capítulo se abordará en detalle todo lo relacionado con la fundamentación teórica, se hará un estudio del estado del arte en el mundo y en nuestro país. Se especificarán algunos conceptos asociados a la investigación. A partir de todos estos elementos se desarrollará la investigación, haciéndose más sencilla la comprensión de temas que serán abordados en lo adelante.

1.2. Conceptos esenciales

1.2.1 Aplicación Informática

Son los programas con los cuales el usuario final interactúa, es decir, son aquellos programas que permiten la interacción entre el usuario y la computadora. Esta comunicación se lleva a cabo cuando el usuario elige entre las diferentes opciones o realiza actividades que le ofrece el programa.

Juan Salvador Castejón Garrido, Secretario del CIIRM (Colegio de Ingenieros en Informática de la Región de Murcia); define aplicación informática mediante la figura que se muestra a continuación. (CASTEJÓN GARRIDO 2004)

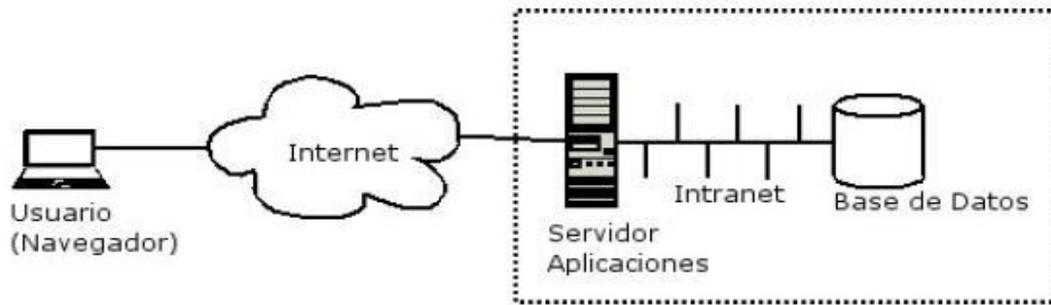


Fig. #1: Aplicación Informática

Es un programa de ordenador que se compra ya realizado y listo para usar. Las hay de muy diversos tipos, según para qué propósito se hayan diseñado: procesadores de texto, bases de datos, programas de contabilidad, de facturación, entre otras. (IGNACIO CABANES) Las aplicaciones por su uso se pueden dividir en dos grandes tipos:

- Aplicaciones de Escritorio
- Aplicaciones Web

A lo largo de la historia de las aplicaciones se ha demostrado con grandes resultados que las aplicaciones informáticas, ya sean de escritorio o Web han tenido un impacto crucial en la economía de un país, pues éstas han jugado un gran papel en el progreso de la humanidad en todas las esferas (educación, salud, cultura, deporte, economía, entre otras), estos avances se han demostrado con el perfeccionamiento intelectual humano que día a día se va alcanzando en el mundo, esto ha hecho al hombre más capaz pues ha puesto sus conocimientos en función de la evolución según las necesidades por las cuales han pasado.

Para poder manejar con eficiencia una Aplicación de escritorio, aquellos usuarios que vayan a hacer uso de una, deben ante todo ostentarla en su puesto de trabajo (PC), donde estará ocupando parte del espacio disponible de su PC para hacer uso de otros recursos y servicios.

Actualmente muchas empresas hacen uso de este tipo de aplicaciones y aunque en los últimos años la mayoría de las empresas están emigrando hacia las aplicaciones Web, les está solucionando de cierta forma los problemas que allí tienen.

Por otra parte las aplicaciones Web; para su uso sólo necesitan estar intercomunicadas con un servidor, los usuarios pueden acceder sin necesidad alguna de tener instalada una aplicación en su puesto de trabajo, con sólo sentarse en una PC cualquiera puede acceder con facilidad a las mismas.

La infraestructura productiva que tiene la universidad es muy dispersa, los clientes pueden estar en cualquier docente, oficina, etc. Por tanto para el desarrollo de este trabajo se seleccionó las Aplicaciones Web por las facilidades que nos brindan y por que es la más apropiada para darle cumplimiento a la situación problemática. Ya que se puede disponer de un servidor para atender las peticiones de un cliente, estando éste en cualquier parte.

1.2.2. Aplicación Web

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Es un sitio que se encuentra en la Red, al cual los usuarios acceden con el fin de beneficiarse con el uso de la información que posea. Aborda un determinado tema. Se dice que los usuarios se benefician con la información que éstos les brindan porque una vez que acceden a él, pueden interactuar con ella, actualizarla o modificarla según los privilegios de seguridad de la aplicación que éstos tengan.

1.2.3. Sistema

Ferdinand de Saussure define como sistema:

"Sistema es una totalidad organizada, hecha de elementos solidarios que no pueden ser definidos más que los unos con relación a los otros en función de su lugar en esa totalidad".

Según la IEEE Standard Dictionary of Electrical and Electronic Terms:

"Sistema es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por una cualquiera de las partes individuales. La complejidad de la combinación está implícita".

Dado estos planteamientos se puede definir sistema como el conjunto de procesos o elementos interrelacionados entre si para cumplir un objetivo específico y se define con el siguiente diagrama conceptual.

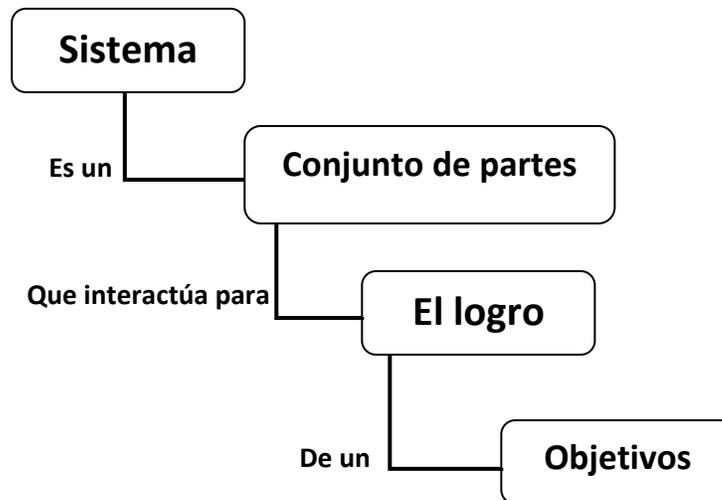


Fig. #2: Diagrama conceptual de sistema

1.2.4. Sistema informático

Según Andrés Berdasco Blanco un Sistema Informático es el conjunto formado por elementos hardware y software que constituyen los recursos a los cuales llegan las peticiones de los usuarios para ser atendidas, en otras palabras se puede decir que es uno o varios ordenadores con un sistema operativo y con los programas (software) necesarios por los usuarios (BERDASCO BLANCO 2000).

Es aquel sistema que se encarga del manejo de información en la computadora, a través de la cual el usuario controla las operaciones que realiza el procesador.

Los sistemas informáticos hay que mirarlos desde diferentes puntos de vista:

- Para una organización cualquiera un sistema informático puede ser: un departamento como cualquier otro, con recursos, que está a disposición de la organización.

- En el mundo de la informática: es un conjunto de servidores, redes y terminales de trabajo para "hacer software".
- Para los usuarios: es una herramienta más que les da la organización a la pertenecen para mejorar el funcionamiento de sus tareas.

Después de una investigación sobre los sistemas informáticos; se define que un sistema informático es aquella unión que se concibe entre los programas y los componentes físicos, ofreciéndoles respuestas a aquellos usuarios que han hecho de una forma o de otra cierto pedido.

1.2.5. Sistema de gestión

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. (BSI 2009)

1.2.6. Servidor

Es un tipo de software que realiza ciertas tareas en nombre de los usuarios, también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. (masadelante)

1.2.7. Proyecto

Los proyectos surgen de las necesidades individuales y colectivas de las personas. Son las personas las que importan, son sus necesidades las que deben ser satisfechas a través de una adecuada asignación de recursos, teniendo en cuenta la realidad social, cultural y política en la que el proyecto pretende desarrollarse.

Según J. Mónica Thompson define proyecto como "una herramienta o instrumento que busca recopilar, crear, analizar en forma sistemática un conjunto de datos y antecedentes, para la obtención de resultados esperados. Es de gran importancia porque permite organizar el entorno de trabajo".

Un proyecto es una ruta para el logro de conocimiento específico en una determinada área o situación en particular, a través de la recolección y el análisis de datos.

Por tanto, un proyecto es la búsqueda de una solución inteligente de forma sistemática al planteamiento de un problema.

1.2.8. Proyecto productivo

Se entiende por proyecto productivo al conjunto de actividades planeadas, encaminadas a desarrollar una actividad económica que genere beneficios económicos, de tal forma que justifique el uso de recursos financieros.

1.3. La pérdida de datos

Dentro del desarrollo tecnológico en el que estamos inmersos, la información se ha convertido en el elemento más importante de cualquier persona, empresa, institución o gobierno. Su disponibilidad es una de las características más fundamentales para la subsistencia del medio que la utilice, y su pérdida puede ocasionar grandes problemas de los cuales es muy difícil recuperarse.

Debido al gran cúmulo de información con la que estamos relacionados en nuestro país, las copias de seguridad han pasado a un plano indispensable para el buen funcionamiento de cualquier tipo de empresa, sin tener en cuenta sus dimensiones o perfil de negocio; la capacidad de reanudar sus servicios lo antes posible ante cualquier situación de pérdida o corrupción de la información es de vital importancia. En la conferencia a cargo de Andrés Sánchez, en junio de 2007, explicó que el 40 por ciento de las compañías que experimentan un desastre de gran magnitud quedan fuera del negocio si no pueden tener acceso a su información en 24 horas. (Sanchez, 2007).

A lo largo de la historia, la humanidad ha sido víctima en determinados momentos de tragedias provocadas por la ira de la naturaleza, el 12 de julio de 2006, la Universidad del Sur de Chile, Octava Región, fue víctima del desborde del río Biobío, que además de verse obligada a cancelar sus actividades, se enfrentó al problema de perder todos los datos de un servidor, producto de la completa inundación de la pieza donde se alojaba. Los computadores se encontraban sumergidos bajo el agua. (Cooperativa, 2006)

En las oficinas Centrales del Credit Lyonnais, banco francés de mucho prestigio, un incendio hizo que los administradores entraran en el edificio en busca de las Cintas de copias de seguridad, y todo porque no tenían copias fuera del lugar. (Hoy, 2006)

Las copias de seguridad en los sistemas informáticos tienen por objetivo mantener cierta capacidad de recuperación de la información ante posibles pérdidas o desastres. Todas y cada una de las situaciones anteriormente planteadas ponen sobre la mesa la necesidad de poseer un sistema de respaldo de datos y sobre todo, en un lugar geográficamente distante de donde se encuentran los posibles afectados, y así de esta manera poder restaurar las operaciones en el menor tiempo posible.

1.4. ¿Qué son las copias de seguridad?

Existen varias definiciones acerca de lo que son las copias de respaldo, aunque todas convergen en la misma idea. A continuación se muestran varias de estas definiciones obtenidas de diferentes fuentes:

Copia de ficheros o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales. Esta sencilla acción evita numerosos y a veces irremediables problemas si se realiza de forma habitual y periódica. (Definicion, 2009)

Hacer una copia de seguridad o copia de respaldo (Backus en Ingles, el uso de este está ampliamente extendido) se refiere a la copia de datos de tal forma que estas copias adicionales pueden restaurar un sistema después de una pérdida de información. (Wikipedia, 2009)

Copia de seguridad es la actividad de copiar archivos a bases de datos a fin de que puedan ser conservadas en casos de fallo del equipo o de otra catástrofe. La copia de seguridad suele ser una parte rutinaria de la operación de grandes empresas con supercomputadoras, así como los administradores de las empresas más pequeñas con menos recursos. Para los usuarios de computadoras personales, la copia de seguridad también es necesaria pero no se realiza muy a menudo. La recuperación de los archivos de la copia de seguridad se denomina: restablecimiento. (Searchstorage, 2007)

Archivo de copia de seguridad: Copia de un archivo para efectos de la posterior reconstrucción del mismo, en caso de ser necesario. (Glossary, 1996)

Una copia de un programa o archivo se almacena por separado del original. (Dictionary, 2006)

1.5. Historia de las copias de seguridad

Las copias de seguridad, hoy día, son un campo rápido en vías de desarrollo. Las nuevas tendencias y soluciones aparecen, los métodos de respaldo y las tecnologías se convierten en más complejas. El respaldo es la copia de un archivo para efectos de la posterior reconstrucción del mismo. Así como podemos enfatizar que los primeros dos aspectos a tener en cuenta son: *el soporte lógico informático de almacenamiento para los datos y los depositarios para soporte lógico informático del respaldo*. Otro aspecto importante es la *necesidad del respaldo* causado por el desarrollo tecnológico y la expansión de volúmenes de datos.

Las primeras copias de seguridad estaban hechas de grandes carretes de cinta magnetofónica y hasta en papel, como las cintas perforadas. En la siguiente era se guardaron en su mayor parte en discos flexibles de tamaños diversos. Los respaldos son actualmente escritos en discos compactos, unidades de disco duro, dispositivos flash y hasta por vía remota. Pero algunas tecnologías, como cinta, aún son muy populares continuándose su desarrollo y fabricación.

1.5.1. Lista de software para respaldo

En el mundo existe una gran variedad de estos tipos de sistemas dedicados al respaldo de datos. En la tabla 1 se reflejan las herramientas más destacadas de este tipo en el software libre así como la licencia bajo la cual fueron liberadas, en la tabla 2 se reflejan las más destacadas en el mundo privativo.

1.5.1.1. Software libre

Redes Extensas	Licencias	
	Bacula	GPL
	BackupPc	GPL
	AMANDA	BSD
	Restore	GPL
Redes pequeñas		
	RSync	GPL

	Duplicity	GPL
	Dump	GPL
	Tar	GPL
	CPIO	GPL
	DAR	GPL
	Rdiff-Backup	GPL
	Rsnapshot	GPL
	Dirvish	OSL
Sistemas Locales		
	Areca Backup	GPL
	Cobian Backup	MPL
	FlyBack	GPL
	Mondo	GPL

Tabla #1: Programas de respaldo de software libre

1.5.1.2. Software privativo

Redes Extensas	
	Atempo TIMEnavigator
	Bakbone NetVault
	CommVault Systems Galaxy
	EMC Legato Networker
Redes pequeñas	

	CommVault Systems
	EMC Corporation Retrospect
	Symantec Backup Exec
	Unitrends
	VisionWorks Solutions
Sistemas Locales	
	Mac Backup
	Backup4all
	Genie Backup Manager
	Macrium Reflect

Tabla #2: Programas de respaldo de software privativo

1.5.2. Descripción de algunas herramientas utilizadas para realizar respaldos.

Rsnapshot

Se compone de una colección de scripts escritos en Perl que automatizan el proceso de crear copias de seguridad incrementales, la clave de este programa es que sólo copia el contenido que ha cambiado desde la última recopilación hecha por él. Rsnapshot utiliza rsync un conocido y popular programa de sincronización y transferencia de archivos. Trabaja en cualquier Sistema Operativo de la familia UNIX. (Gahona, 2008)

Rsync

Es un programa de respaldo muy fácil de utilizar. Detecta archivos y directorios que necesitan ser copiados cuando algún atributo de ellos ha sido modificado (por ejemplo, la fecha/hora de la última

modificación, o el tamaño del archivo), en cualquier caso, algo ha cambiado desde el último respaldo. Rsync utiliza para el proceso de copiado la compresión de la información, comprimiéndola antes de enviarla y descomprimiéndola en el destino. Rsync puede copiar archivos desde:(Gahona, 2008)

- Un sistema local a otro sistema local.
- Un sistema local a un sistema remoto.
- Un sistema remoto a un sistema local.

Bacula

Bacula es un sistema de respaldo de código abierto que se puede usar en entornos a gran escala para cubrir las necesidades profesionales. La herramienta de respaldo es sin duda adecuada para usarse con sistemas en producción, pero aún existen algunos elementos que deben mejorarse. Respecto a la documentación es bastante extensa y muy bien estructurada, de hecho se escribe antes de que exista el código y es posible encontrar características que todavía no se han implementado. Todas las tareas que realiza Bacula se han modularizado y repartido, entre varios dominios y servicios. Trabaja bajo Linux, la familia BSD, Solaris y Windows, tiene soporte para base de datos tanto MySQL, PostgreSQL y SQLite. (Gahona, 2008) (Bacula)

El servicio de almacenamiento de bacula consiste en un software que permite almacenar y recuperar los atributos de los archivos y datos al backup físico u otro volumen. En otras palabras, el bacula storage es el responsable de leer y escribir en las cintas (u otro dispositivo de almacenamiento que se utilice).

Después de hacer un estudio de estas tres herramientas de respaldo, se ha llegado a la conclusión que la herramienta a utilizar es Bacula para el desarrollo de la aplicación Web propuesta para la gestión de los servidores de salva.

1.6. Tendencias y tecnologías actuales

A continuación se describen algunas de las tendencias y tecnologías actuales posibles a utilizar para darle solución a los problemas planteados anteriormente de manera eficiente, teniendo en cuenta las necesidades existentes y el entorno donde se aplicará el sistema que se va a construir.

1.6.1. Lenguajes de Programación Web.

Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas comunicarse con las computadoras. Existen por lo menos varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. (Lenguaje de Programación, 2009).

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

1.6.1.1. Lenguajes del lado del servidor.

PHP

PHP es un acrónimo ("Hypertext Preprocessor"), es un lenguaje de alto nivel embebido en páginas HTML y ejecutado en servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí misma. La meta del lenguaje es permitir rápidamente a la comunidad de desarrolladores la generación dinámica de páginas Web". (ADRFORMACIÓN.COM, 2004)

Características del lenguaje

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierte en la herramienta ideal para la creación de páginas Web dinámicas:(RODAS HINOSTROZA, 2007)

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se realiza con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Ventajas que proporciona la utilización del lenguaje

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y Postgres.
- Posee una amplia documentación en Internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel, facilitando un poco la programación.

En PHP los scripts del lado del servidor se insertan dentro del código HTML. Es gratuito y multiplataforma, y está ampliamente difundido en el mundo entre la comunidad de programadores.

Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.

Pero tiene las desventajas siguientes:

Al ser un lenguaje interpretado hace al programa más lento.

Al no tener que compilar, por un lado se gana tiempo, pero por otro, te pasas por alto un montón de chequeos que hace el compilador.

1.6.1.2. Lenguajes del lado del cliente

Java Script

Lenguaje que permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web mayor dinamismo. Es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

El advenimiento de Java Script ha resuelto de manera fácil y elegante la mayoría de los problemas con que se enfrenta el diseñador de páginas Web referidos a la programación. Sus requerimientos son relativamente sencillos, es un lenguaje cuyos códigos se interpretan en el navegador del cliente, sin tener que ir y venir del cliente al servidor actualizando la información.

Visual Basic Script

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspiradas en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas web es muy similar al utilizado en Java script y los recursos a los que se puede acceder también son los mismos: El navegador.

Tendría sentido la construcción de una Intranet donde sepamos con toda seguridad que los navegadores que se van a conectar serán siempre Internet Explorer. En este caso, un programador habitual de Visual Basic tendría más facilidades para realizar los scripts utilizando Visual Basic Script en lugar de Java script.

1.6.2. Servidores Web.

El Servidor Web es el programa encargado de gestionar las peticiones de los usuarios que visitan su página Web. Es el programa encargado de mostrar sus páginas cuando un visitante realiza una petición en su dominio.

Teniendo en cuenta que el 80% de las operaciones que realizará un usuario en el servidor es visionar páginas Web, la elección del programa encargado para tal cometido es clave para el buen funcionamiento general del servidor. A continuación se expondrán algunas de las características del Servidor Apache y el IIS con vista a seleccionar el que se utilizará para el desarrollo de la aplicación.

1.6.2.1. Servidor Apache

Apache es considerado el Servidor Web por excelencia, no solo por su gran aceptación, puesto que casi el 70% de los servidores Internet confía en él, sino porque desde su nacimiento ha demostrado con creces su estabilidad, solidez y su mayor rendimiento ante sus competidores.

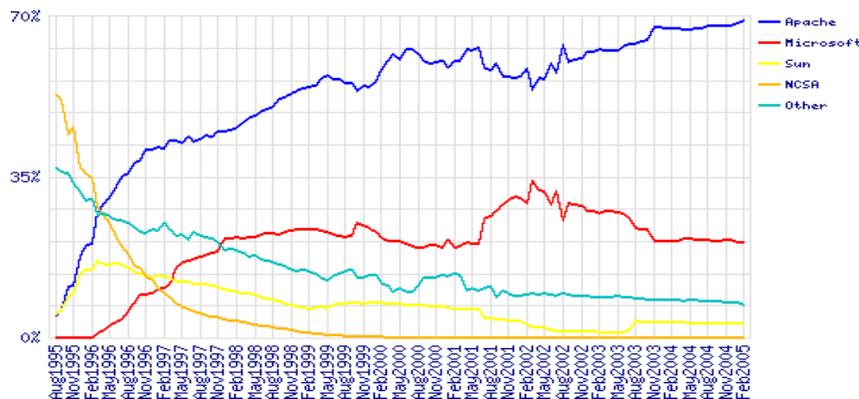


Fig. #3: Gráfica de uso de Servidores Web en el mundo

Principales ventajas:

Fiabilidad: Alrededor del 90% de los servidores con mas alta disponibilidad funcionan bajo servidores Apache.

Software Libre: Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License,

que permite la modificación del código.

Extensibilidad: Se pueden añadir módulos para ampliar las ya de por sí amplias capacidades de Apache.

También dispone de una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python, etc.), monitorizar el rendimiento del servidor, atender peticiones encriptados por SSL, crear servidores virtuales por IP o por nombre (varias direcciones web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso modificados por cualquier persona con conocimientos de programación.

Apache es un servidor Web que permite acceder a las páginas Web que están alojadas en una computadora. Es de código abierto y actualmente es el servidor Web que más se utiliza en el mundo, encontrándose por encima de sus competidores, tanto gratuitos como comerciales. Funciona sobre cualquier plataforma. Actualmente es uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.

Este servidor tiene capacidad para servir tanto páginas estáticas como dinámicas a través de otras herramientas soportadas que facilitan la actualización de los contenidos usando bases de datos, ficheros u otras fuentes de información.

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

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

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no sea necesario volver a instalar el software.

1.6.2.2. Servidor Web Internet Information Server(IIS)

Internet Information Server, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

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

Dentro de las ventajas agregadas en seguridad se aprovechan:

De las últimas tecnologías de cifrado y métodos de autenticación mediante certificados de cliente y servidor. Una de las formas que tiene IIS de asegurar los datos es mediante SSL (Secure Sockets Layer). Esto proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario. Otra es la autenticación implícita que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy.

IIS también es capaz de impedir que aquellos usuarios con direcciones IP conocidas obtengan acceso no autorizado al servidor, permitiendo especificar la información apropiada en una lista de restricciones. Volviendo de nuevo a la seguridad. El almacenamiento de certificados se integra ahora con el almacenamiento CryptoAPI de Windows. Se puede utilizar el administrador de certificados de Windows para hacer una copia de seguridad, guardar y configurar los certificados.

Además, la administración de la seguridad del servidor IIS es una tarea fácilmente ejecutable a base de asistentes para la seguridad. Se pueden definir permisos de acceso en directorios virtuales e incluso en archivos, de forma que el asistente actualizará los permisos NTFS para reflejar los cambios. Si se trabaja con entidades emisoras de certificados, es posible gestionar la lista de certificados de confianza (CTL, Certificate Trust List) con el asistente para CTL.

Microsoft ha mejorado sustancialmente su software estrella en el campo de los servicios Web. Los avances vienen motivados sobre todo por la seguridad y el rendimiento pero todavía adolece de algunos agujeros de seguridad además pues de ser código cerrado, IIS solo puede operar en plataformas Windows.

1.6.3. Metodología De desarrollo de Software

Uno de los principales problemas en la actualidad en el desarrollo de software es seleccionar la metodología más adecuada que posibilite obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos a nivel mundial. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Se analizará seguidamente dos de las más conocidas, sus características.

1.6.3.1. Proceso Unificado de Desarrollo de Software (RUP)

Actualmente el proceso unificado lleva a la construcción de sistemas de software mucho más complejos y grandes. Debido al auge de las computadoras y a la vez el aumento del rigor de los usuarios, esto ha traído un rápido incremento en Internet para el intercambio de todo tipo de información. James Rumbaugh, Grady Booch e Ivar Jacobson, autores de “El proceso unificado de desarrollo de software”, opinan que “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las

múltiples cadenas de trabajo de un gran proyecto de software” (PRESSMAN 2002; RUMBAUHG et al. 2000b)

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular. (PRESSMAN 2002; RUMBAUHG et al. 2000a)

- 1) Desarrollo de software de forma iterativa.
- 2) Manejo de requerimientos.
- 3) Utiliza arquitectura basada en componentes.
- 4) Modela el software de forma visual, usando UML.
- 5) Verifica la calidad del software.
- 6) Controla los cambios.
- 7) Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- 8) Especifica los artefactos que deben desarrollarse en cada fase de desarrollo del software.

RUP consta de cuatro etapas o fases por donde el software tiene que transitar para que obtenga la calidad requerida.(RUMBAUHG et al. 2000b)

- 1) Fase de Inicio
- 2) Fase de Elaboración
- 3) Fase de Construcción
- 4) Fase de Transición

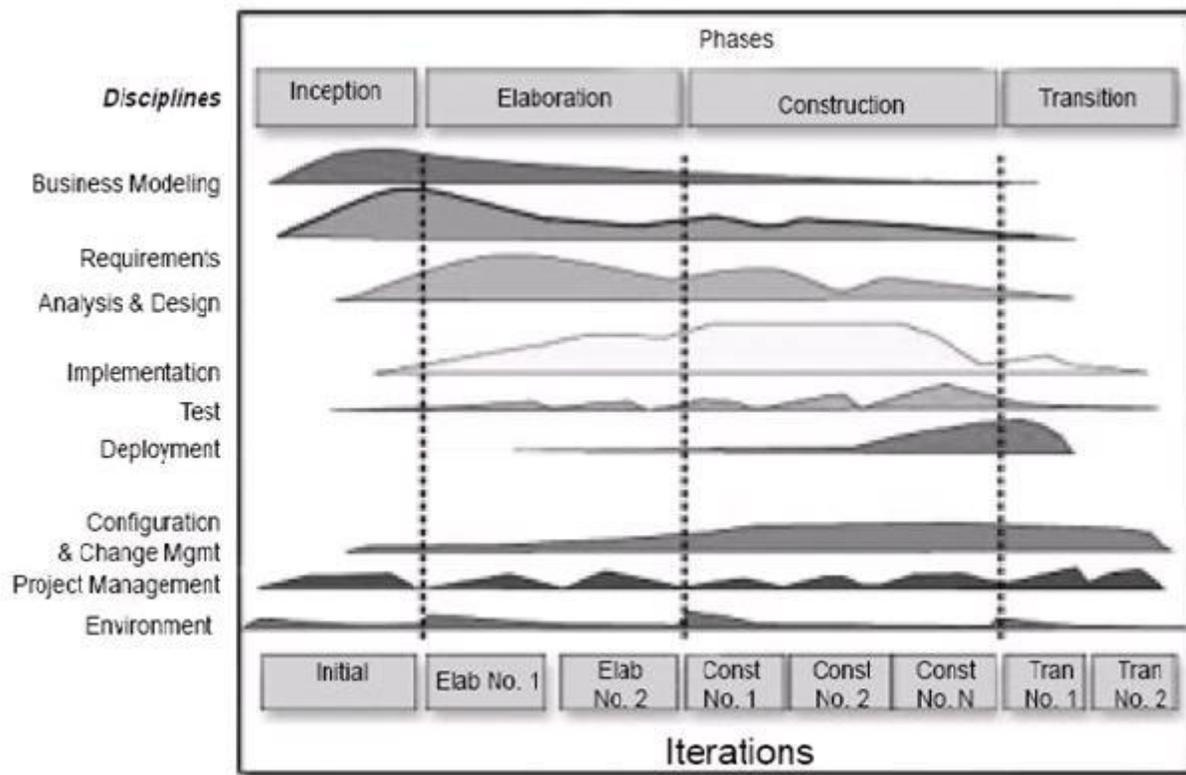


Fig. #4: Disciplinas a través de las iteraciones

En el caso específico de las aplicaciones Web, las actividades establecidas por dicho proceso son suficientes para garantizar cubrir todos los aspectos de los entornos de este tipo de aplicación. El Proceso Unificado Rational presenta una serie de características que lo convierte en único, al proceso unificado, ciclo de vida de RUP. (RUMBAUHG et al. 2000a)

1. Dirigido por Casos de Uso
2. Centrado en la Arquitectura
3. Iterativo e Incremental

Esta metodología utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software. Es una parte esencial del Proceso Unificado de desarrollo de software y fueron desarrollados paralelamente por las mismas personas, haciendo que su integración sea un éxito.

1.6.3.2. Programación Extrema (XP)

A diferencia de RUP como metodología pesada, la Programación Extrema (XP) es una metodología ágil que intenta reducir la complejidad del software por medio de un trabajo orientado al objeto, basado en las relaciones interpersonales y la velocidad de reacción. Intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debe estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones. (MOLPECERES 2003)

XP define UserStories como base del software a desarrollar. A partir de los UserStories y de la arquitectura perseguida se crea un plan de release entre el equipo de desarrollo y el cliente. Para cada release se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración). (MOLPECERES, 2003)

Cada iteración tendrá como resultado un programa que se le entrega al cliente para que lo juzgue, si el cliente no está de acuerdo se definen las próximas iteraciones del proyecto y se someten a la aprobación del cliente hasta que el cliente esté completamente de acuerdo y el software cumpla con todos sus requisitos. A diferencia de otros métodos, en XP la codificación pertenece al equipo completo, de forma que el conocimiento de la aplicación lo posea el equipo entero y no unos pocos miembros. (MOLPECERES, 2003)

En esta metodología se sigue un diseño evolutivo con la siguiente premisa: Conseguir la funcionalidad deseada de la forma más sencilla posible. Este diseño hace que apenas se le de importancia al análisis como fase independiente, debido a que se trabaja exclusivamente en función de las necesidades del momento. (MOLPECERES, 2003)

Ventajas

- ✓ Al trabajar de dos en dos, el código será de mayor calidad desde el mismo momento de crearlo y tendrá menos fallos.
- ✓ Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- ✓ Si una pareja realiza un trozo de código susceptible de ser reutilizado en el proyecto, hay dos programadores que lo saben y que lo reutilizarán cuando puedan (ya que saben cómo funciona), enseñándolo a sus nuevos compañeros. De esta manera el conocimiento del código ya hecho se propaga de forma natural entre todos los programadores del equipo.
- ✓ El estilo de programación tiende a unificarse.

Inconvenientes

- ✓ Para un programador experto puede resultar tedioso tener a un novato a su lado permanentemente.
- ✓ El programador experto no aprende y su trabajo se ve ralentizado.
- ✓ La mejora o cambios en el estilo de programación puede resultar más complejo

1.6.4. Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) fue adquirido en 1997 por la Object Management Group (OMG) para luego convertirse en un estándar de facto, que permite visualizar, especificar y documentar modelos que se crean durante la aplicación de un proceso de software.

Es, probablemente, una de las innovaciones conceptuales en el mundo tecnológico del desarrollo de software que más expectativa ha generado a lo largo de muchos años, comparable con la aparición e implantación de los lenguajes COBOL, Basic, Pascal, C++, y actualmente con los más recientes Java, XML, C#.

UML es ya un estándar de la industria del software, pero no solo de la industria sino, que en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posteriormente para la construcción de prototipos. (HERNÁNDEZ ORALLO 2001)

UML ha nacido como un lenguaje, pero es mucho más que un lenguaje de programación. En realidad se ha diseñado y construido un lenguaje que ha nacido con una madurez sólida si se le compara, incluso con los últimos desarrollos de HTML, C#, Java, Ajax, Xml, los lenguajes por excelencia del mundo de la Internet. (LARMAN 1999)

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software.(RUMBAUHG et al. 2000a)

El Lenguaje Unificado de Modelado (UML) ha ganado en utilización actualmente, por ser la mezcla eficiente de una gran cantidad de estándares internacionales. Su base esta soportada por tres metodologías procedentes de la unión de tres grandes creadores, James. Rumbaugh, Grady. Boosh e Ivar. Jacobson. Logrando así un lenguaje de excelencia para modelar, que es el procesamiento que realizan los ingenieros para el diseño de software previo a su construcción.(RUMBAUHG et al. 2000b)¹

Tiene una gran cantidad de propiedades que han sido las que, realmente, ha contribuido a hacer de UML el estándar de la industria en la actualidad.

Algunas de las propiedades de UML como lenguaje de modelado son:(RUMBAUHG et al. 2000b)

1. Es un lenguaje distribuido y adecuado a las necesidades de conectividades actuales y futuras. Ampliamente utilizado por la industria del software.
2. Reemplaza a decenas de notaciones empleadas por otros lenguajes.
3. Modela estructuras complejas.

4. Las estructuras más importantes que soporta tienen su fundamento en la tecnología orientada a objeto, tales como objetos, clases, componentes y nodos.

5. Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirve para evaluar el estado de las máquinas.

El modelar sirve, no solamente para los grandes sistemas, sino en aplicaciones de pequeño tamaño se obtienen beneficios del modelado, sin embargo es un hecho que entre más grande y complejo es el sistema, más importante es el papel que juega el modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad". (RUMBAUHG et al. 2000a)

Se utilizará como notación el Lenguaje Unificado de Modelado para lograr un mayor entendimiento ya que se logra modelar y describir secuencialmente por pasos todos los procesos que se llevan a cabo según la problemática planteada. Sirve porque es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

1.6.5. Herramienta CASE de Desarrollo de Software

Hoy días, se ha hecho habitual que cada vez que se desarrolle un software por pequeño o grande que se sea, se utilizan las herramientas CASE, con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio hasta el final e incrementando su calidad.

Estas herramientas son de gran utilidad pues intervienen en todos los aspectos del ciclo de vida del desarrollo del software, permitiendo a los desarrolladores menos complejidad a la hora de realizar diferentes actividades del proceso de software como son la realización del diseño del proyecto, la documentación o detección de errores.

Algunos de los objetivos de las herramientas CASE son:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software en construcción.
- Perfeccionar la planificación del proyecto.

1.6.5.1. Rational Rose

Es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática, otra dinámica de los modelos del sistema, una lógica y otra física; que permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.(LARMAN 1999)

A continuación los tipos de modelos:

1. Desarrollo Iterativo
2. Generador de Código
3. Ingeniería Inversa
4. Trabajo en Grupo

1.6.5.2. Visual Paradigm

Visual Paradigm es una herramienta multiplataforma para el modelado de aplicaciones, sobre todo en proyectos donde se vaya a aplicar intensivamente los conceptos avanzados de orientación a objeto, ejemplo de ello es el uso del lenguaje Java, para el cual es vital realizar un buen diseño del sistema.(CANALES MORA, 2004)

Facilita a los ingenieros de software diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software. Para construir sistemas de software a gran escala de manera confiable a través del uso de un enfoque Orientado al Objeto. Presenta un generador de código que soporta más de 10 lenguajes y proporciona la ingeniería inversa:(VISUAL PARADIGM 2007)

1. Java.
2. C++.
3. CORBA.
4. IDL.
5. PHP.
6. Esquema de XML.

1.6.6. Sistema Gestores de Bases de Datos.

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta (Maestrosdelweb.com, 2008)

La bibliografía especializada a menudo se refiere a estos sistemas como SGBD o DBMS, siendo ambos equivalentes y acrónimos respectivamente. En la actualidad existe una gran variedad de SGBD, tanto de tipo comercial como libre. Entre los más usados dentro del grupo de los comerciales se encuentran:

1.6.6.1. PostgreSQL

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

Postgres ofrece una alternativa a los demás sistemas de administración de bases de datos. Al igual que otros proyectos de software libre como Apache, GNU Linux y Media Wiki, PostgreSQL no está controlado por una sola compañía, sino que cuenta con comunidad global de desarrolladores y compañías para su evolución.

PostgreSQL posee una serie de características positivas respecto a otros. Entre estas cabe mencionar que es escalable, ajustable al número de procesadores y a la cantidad de memoria que posee el sistema

de la forma más óptima. Por este motivo es capaz de soportar una mayor cantidad de peticiones simultáneas.

Otra característica relevante de este gestor es la capacidad de almacenar procedimientos (funciones) en la propia base de datos, además de ser multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso; posee tipos internos que han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales; la velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20% - 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

Ventajas de PostgreSQL

PostgreSQL ofrece muchas ventajas para su compañía o negocio respecto a otros sistemas de bases de datos:

Instalación ilimitada

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto tiene varias ventajas adicionales:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

Mejor soporte que los proveedores comerciales

Además de nuestras ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

Ahorros considerables en costos de operación

Nuestro software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Además de esto, nuestros programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

Estabilidad y confiabilidad legendarias

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

Extensible

El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

Multiplataforma

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

Diseñado para ambientes de alto volumen

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mucho mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

Herramientas gráficas de diseño y administración de bases de datos

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin , pgAccess) y para hacer diseño de bases de datos (Tora , Data Architect).

1.6.6.2. MySQL

Es un sistema de administración de base de datos, multi - hilo, multiusuario, con más de seis millones de instalaciones por todo el planeta. MySQL es propietario y patrocinado a partir de la versión 5 por la compañía Sun Microsystems que lo adquirió recientemente. La empresa sueca MySQL AB que fue fundada por David Amar, Allan Lar son y Michael “Monti” Adenias, era la propietaria de MySQL hasta hace muy poco y vende otro producto llamado Max, que está formado por otra Base de Código no relacionada con la de MySQL. Existen muchas API (Aplicacion Programan Interface – Interfaz de Programación de Aplicaciones) que permiten que aplicaciones escritas en varios lenguajes de programación accedan a bases de datos.

MySQL es muy popular en aplicaciones Web y actúa como un componente de bases de datos para las plataformas LAMP, MAMP y WAMP (Linux/MAC/Windows-Apache-MySQL-PHP/Perl/Chitón) y para herramientas de búsqueda e identificación de errores como Bugirla.

Para administrar bases de datos en MySQL pueden usarse herramientas de línea de comandos como MySQL y mysqladmin. Descargables desde el sitio de MySQL están las herramientas basadas en Interfaz gráfica de usuario (GUI), MySQL Administrador y MySQL Query Browser, estas dos incluidas en un paquete llamado MySQL GUI Tools.

MySQL trabaja en numerosas plataformas como AIX, HP-UX, GNU/Linux, Mac OS X, Novell NetWare, OpenBSD, OS/2, Solaris, SunOS, y todas las versiones de Windows. Su mayor desempeño se logra cuando se combina con el lenguaje de programación PHP.

Desventajas

- ✓ Mysql no tiene un panel de control gráfico y detallado, pero existen varias herramientas libres que juntas dan la misma utilidad.
- ✓ No permite el modo de autenticación local (seguridad integrada de windows), sólo el modo estándar.
- ✓ No sincroniza los datos con otras bases de datos réplicas.

Postgresql como base de datos es más maduro que mysql, en lo único que falla es que puede no ser tan rápido como mysql en cierto rango de datos. Postgresql simplemente deja atrás a MySQL, ya que la caída de rendimiento de MySQL cae exponencialmente, mientras que PostgreSQL si bien cae el rendimiento, pero no tanto como exponencialmente.

1.6.6. Framework.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes; proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener; facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (librosweb.es, 2008).

1.6.6.1. Framework Symfony.

Symfony es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

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

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

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

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

Symfony proporciona una solución muy factible a la calidad de su código fuente y a la gran cantidad de documentación disponible, dos ventajas muy importantes sobre otros frameworks disponibles. Los colaboradores han aparecido enseguida proponiendo parches y mejoras, detectando los errores de la documentación y realizando otras tareas muy importantes.

Patrón Arquitectónico.

Symfony está basado en un patrón clásico del diseño Web conocido como arquitectura

Modelo – Vista – Controlador (MVC), que está formado por tres niveles:

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

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

1.6.6.2. Prado Framework.

Prado es un Framework bastante orientado a objetos con similitudes con lenguajes como Microsoft ASP .NET. Entre sus principales características están:

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

Prado es un framework para PHP basado en componentes y en eventos. Inicialmente inspirado en Apache Tapestry, la primera versión se realizó para PHP4, pero se reescribió completamente para PHP5.

Entre las características que ofrece se pueden encontrar la separación entre la presentación y la lógica, su arquitectura modular configurable, componentes web, internalización y localización, manejo de errores, logs, caché, ACL, prevención de XSS y mucho más.

1.6.7. Entorno Integrado de Desarrollo (IDE).

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

De manera general un entorno de desarrollo integrado o en inglés Integrated Development Environment, es un programa compuesto por un conjunto de herramientas para un programador (Maestrosdelweb.com, 2008). Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, puede utilizarse para varios.

1.6.7.1. Zend Studio.

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

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

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda, permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración (Maestrosdelweb.com, 2008).

Si se desea aumentar la productividad en los desarrollos PHP no cabe duda que este programa puede resultar útil y prestar gran ayuda en la realización de los mismos ofreciendo un entorno agradable y funcionalidades que simplifican el trabajo y optimizan el resultado. Todas las opciones que dispone están pensadas con acierto por personas capacitadas que conocen a fondo la tecnología. Zend Studio incorpora suficientes ayudas como para que su utilización sea confiable y resulte idóneo para desarrollar aplicaciones Web.

1.6.7.2. Eclipse

En la web oficial de eclipse, se define como "An IDE for everything and nothing in particular"(un IDE para todo y para nada en particular). Eclipse es en el fondo, únicamente un armazón sobre el que se puede montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plugins adecuados.

La arquitectura de plugins de Eclipse permite además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, entre otros. Existen versiones de Eclipse instalables para cualquier plataforma que incluyen el código fuente y los plugins más habituales.

Una de las características más curiosas del IDE Eclipse es el modo en que se compilan los proyectos. No existe en Eclipse ningún botón que permita compilar individualmente un fichero concreto. La compilación es una tarea que se lanza automáticamente al guardar los cambios realizados en el código.

La principal diferencia entre un simple editor y un buen entorno de desarrollo es que este se integre con una buena herramienta visual para depurar los programas escritos. Eclipse incluye un depurador potente, sencillo y muy cómodo de utilizar. Permite de una forma muy simple, generar automáticamente la documentación del propio proyecto.

Eclipse ofrece servicios que permiten el trabajo entre desarrolladores de forma más eficiente y sincronizada, pues cuenta con las tareas, que posibilitan una comunicación directa entre los miembros del equipo, permitiendo dar órdenes de trabajo de manera online en forma de correo electrónico. Esta opción hace de Eclipse un IDE ideal para el trabajo en equipo.

1.6.8. Justificación de la selección.

Después de haber analizado las características fundamentales de las tecnologías, lenguajes de programación y sistemas gestores de bases de datos candidatos para la implementación de la propuesta de este trabajo se tuvo como principal elemento a considerar en cada selección el que sean herramientas y tecnologías libres, o de código abierto, motivada por las características que presentan, que los hacen factibles para países subdesarrollados como Cuba.

A partir de estas consideraciones se usará PHP como lenguaje del lado del servidor, al darse su creación y desarrollo en el ámbito de los sistemas libres, bajo la licencia GNU. PHP es un lenguaje orientado a objetos, simple, elegante y con seguridad en el tratamiento de tipos, cuenta con todas las funciones suficientes y necesarias para brindar una solución óptima.

Como lenguaje del lado del cliente se propone el uso de Java Script, por su universal aceptación y la limitante de portabilidad de Visual Basic Script al ser compatible solamente con Internet Explorer.

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

Como framework a utilizar, Symfony deviene candidato ideal porque, entre otras razones, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web; proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja; así como automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

A partir de lo seleccionado hasta el momento es conveniente utilizar como Servidor Web, el Apache primeramente por su aceptación universal y gran potencialidad, además de que como se analizó en el

epígrafe correspondiente IIS es propietario, ideal para aplicaciones sobre Windows lo que no constituye un aspecto a considerar según los objetivos propuestos.

Como sistema de gestión de base de datos, se seleccionó PostgreSQL en su versión 8.2. PostgreSQL es un potente gestor de Base de Datos. Es software libre, corre en un gran número de sistemas operativos, siendo multiplataforma. Tiene todas las características de los SGBD modernos, como son las llaves foráneas, vistas, procedimientos almacenados, varios tipos modernos de datos, entre otros.

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

Como ya se ha explicado en epígrafes anteriores de este capítulo, la correcta selección de la metodología de desarrollo a utilizar en un proyecto es crucial para el buen desarrollo del mismo. Después de analizar las características fundamentales de las metodologías de desarrollo RUP y XP se concluyó que para un mejor desarrollo del proyecto, era necesario utilizar RUP.

A pesar de que pudiese parecer un proyecto de corto plazo, es decir pequeño, representa un sub-sistema de un gran proyecto que necesita de una metodología robusta para su desarrollo, y además de que debe existir uniformidad entre todos los sub-sistema, por lo que no puede utilizarse una metodología ligera en algunos módulos y una pesada en otros.

Con la comparación de las herramientas de modelado CASE se llegó a la conclusión de que se utilizaría Visual Paradigm, esta herramienta soporta hasta la fecha UML 2.1 completo como mínimo y BPMN, entre otros. La principal razón de esta selección es que es una herramienta libre, multiplataforma, y en calidad de la migración que pretende llevar a cabo Cuba unánimemente se está adoptando como herramienta de modelado en la universidad, además de que indudablemente posee funcionalidades que la hacen óptima para el desarrollo de cualquier proyecto.

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

1.7. Propuesta de solución

Con la puesta en práctica de este sistema de salvos, la información se encontraría en los servidores locales de cada proyecto y también estaría disponible en los servidores centrales donde sólo tenga acceso el personal adecuado. Teniendo una salva de toda la información que nos interesa no hay daño que no pueda ser reparado.

La Universidad, además de su servidor central destinado a las salvos de los proyectos, ubicado en la infraestructura productiva (IP), cuenta con un servidor de respaldo a este, ubicado en el Nodo Central, por lo que, para garantizar que el proceso de salva cumpla con las exigencias del desarrollo de la UCI, éste debe dar la posibilidad de replicar la información del servidor central hacia otros servidores. El sistema de salvos en general debe ser automatizado a través de un mecanismo que posea la capacidad de disminuir el esfuerzo que empleará la organización en la labor.

Con este mecanismo de salva implementado, la información de todos los productos de la universidad se encontraría centralizada y actualizada, lo que traería consigo varias mejoras dentro del proceso de toma de decisiones por parte de los directivos. Se podría conocer el avance real de cada proyecto si se cuenta con la autorización para hacerlo y sin necesidad de ir al laboratorio de desarrollo ni reunirse con los implicados con este. También sería una gran ventaja para el área de calidad pues podrían descargar el producto en cualquier momento, confiando en la actualidad del mismo, y realizar las pruebas pertinentes incluso antes de que el producto sea liberado por sus desarrolladores.

1.8. Conclusiones Parciales

Con el desarrollo de este capítulo, se llegó a la conclusión de que los conceptos asociados al objeto de estudio son indispensables para comprender realmente el problema descrito y tener una mayor claridad para encaminar dicha investigación. Además de la selección de las herramientas a utilizar para dar cumplimiento a la propuesta de solución.

2

CAPÍTULO DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1. Introducción

En el presente capítulo se describen los procesos del negocio que tiene que ver con el objeto de estudio, primeramente se modela el negocio propuesto, se identifican los actores, trabajadores, los Casos de Uso correspondientes y la descripción de los mismos. Se especifican las reglas asociadas al dominio del problema.

También se enumeran los requisitos; tanto Funcionales como No funcionales que debe cumplir el sistema que se propone (La aplicación Web a desarrollar), permitiendo así hacer una mejor concepción de forma general del sistema, se identificará mediante un Diagrama de Casos de Uso las relaciones de los actores que interactúan con el sistema y las secuencias de acciones con las que interactúan.

El desarrollo de la aplicación Web que se propone se centra en la puesta en práctica del Proceso Unificado de Desarrollo de Software (RUP), con el apoyo del Lenguaje Unificado de Modelado (UML) para la modelación (generación) de los distintos artefactos que involucran el desarrollo del trabajo y, haciendo uso de la herramienta Case Visual Paradigm para darle cumplimiento a la solución propuesta.

2.2. Modelo del negocio propuesto

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo es necesario centrarse en los procesos de gestión de salvados de los proyectos productivos. Para un mayor y mejor entendimiento de cómo se lleva actualmente a cabo las salvados de los proyectos productivos es mejor hacer uso de las técnicas de modelado que propone UML. El paso más importante dentro de dicho modelado es la identificación de los procesos del negocio de la producción de la UCI. La

obtención de los procesos del negocio es crucial ya que establece las metas y límites de modelado del negocio.

Dada la estructuración del negocio que se está estudiando se propone un Modelo del Negocio ya que este permite de forma visual mostrar a los usuarios los principales conceptos que se manejan. Esto se hace con el objetivo de ayudar a los usuarios, clientes, desarrolladores y otros a tener un mejor entendimiento de los procesos de negocio y, a utilizar un vocabulario común entre todos para un mejor entendimiento del contexto en que se pone el sistema. Para una mejor captura de los requisitos y así construir un sistema correcto se necesita tener un seguro conocimiento de las actividades y del funcionamiento del objeto de estudio.

2.2.1. Reglas generales del negocio

A continuación se definirán las políticas que deben cumplirse o condiciones que deben satisfacerse para que se ejecuten los casos de uso, de manera que regulen los procesos de negocio de acuerdo a las especificidades de cada uno.

- ✓ El Solicitante debe llevar todos los documentos necesarios y correctos al Especialista GSD del IP para la solicitud de un registro de salva que desee realizar.
- ✓ El Solicitante es la persona autorizada a solicitar un registro de salvas.
- ✓ El Especialista GSD es el encargado de crear la salva.
- ✓ El Especialista GSD solo puede tener acceso a la información que se maneja para la confección de las salvas de la UCI.
- ✓ El Especialista GSD es el encargado de revisar las solicitudes que le hace un Solicitante y es quien elabora la solicitud a nivel UCI.

2.2.2. Actores del Negocio

Los actores del negocio son aquellas personas o sistemas que obtienen un resultado observable de gran valor de los procesos del negocio. A continuación se definen los actores del negocio estudiados en la siguiente tabla.

Actor	Descripción
 Solicitante (Asesor de Arquitectura, Especialista y Líder de Proyecto Productivo).	Interviene inicializando el caso de uso del negocio, Solicitar registro de salva. Representa al actor encargado de brindar la información referente a la Gestión de Infraestructura de los proyectos productivos. Es el encargado de realizar las planillas de los Proyectos Productivos.

Tabla #3: Descripción de los Actores del Negocio

2.2.3. Trabajadores del Negocio

Los trabajadores del negocio son aquellas personas o sistemas que están vinculadas en uno o más procesos del negocio, participando en ellos, pero que a diferencia de los Actores, no obtienen de valor alguno. Los trabajadores del negocio estudiado se definen en la siguiente tabla.

Trabajador	Descripción
 Especialista Grupo de Soporte al Desarrollo(GSD)	El trabajador encargado de revisar todas las planillas recibidas de las facultades y de elaborar las planillas UCI y de procesar toda la información proveniente de las facultades.

Tabla #4: Descripción de los Trabajadores del Negocio

2.2.4. Diagrama de Caso de Uso del Negocio

El Diagrama de Caso de Uso del Negocio describe los procesos de negocio de una empresa en términos de casos de uso y actores, que se corresponden con los procesos del negocio y los clientes, respectivamente. Este modelo describe los procesos relacionados con el trabajo de solicitar un registro de salvas.

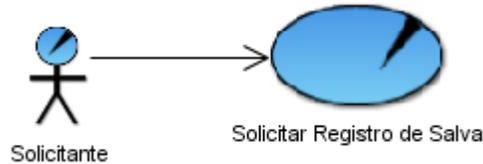


Fig. #5: Modelo de Caso de Uso del Negocio

2.3. Descripción de los procesos del negocio propuesto

Se expondrán a continuación las descripciones de cada caso de uso, en vista a una mayor comprensión del desarrollo de los procesos, puesto que los mismos describen en detalle los pasos y estados por los que transitan los casos de uso. Los diagramas de actividades pueden encontrarse en los [Anexos1](#).

2.3.1. Descripción del Caso de Uso de Negocio: Solicitar Registro de Salva.

Caso de Uso:	Solicitar Registro de Salvas.	
Actores:	Solicitante (Asesor de Arquitectura, Especialista, Líder Proyecto Productivo).	
Trabajadores	Especialistas del GSD de la IP	
Resumen:	El caso de uso se inicia cuando el actor Solicitante llena la planilla de solicitud de registro de salva y la envía al los Especialistas del GSD de la IP los cuales las procesan terminando así el caso de uso.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El caso de uso se inicia cuando un Solicitante realiza una solicitud de registro de salva para el proyecto productivo a través de la PRSPP y la envía a un Especialista del GSD de la IP.	2. El Especialista del GSD de la IP recibe la planilla.	
	3. Verifica que todos los datos de la planilla del registro de salva del proyecto productivo estén correctos.	

	4. Las solicitudes una vez aprobadas por el GSD se procede a la creación de la solicitud a nivel UCI. Registrándose toda la información de la salva.
	5. Se le informa al Solicitante de la aprobación de la salva.
6. Recibe la Información y termina el caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1. La solicitud no es aprobada por el Especialista del GSD de la IP.
7. Recibe la Información y termina el caso de uso.	
Prioridad:	Crítico

Tabla #5: Descripción textual del Caso de Uso: Solicitar Registro de Salva

2.4. Modelo de Objeto



Fig. #6: Modelo de Objetos del Negocio

2.5. Requisitos del Sistema

Desde ahora, ya conocido los conceptos asociados al objeto de estudio del problema comenzaremos a modelar el sistema que se va construir, y de paso se analizará ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo? Para lo mismo identificamos los requisitos Funcionales (RF) y No Funcionales (RNF), y modelar los RF en representaciones de Casos de Uso del

sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

2.5.1. Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, por tal motivo a continuación se presenta un conjunto de ellos:

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

- ✓ Cerrar sesión de usuario.
- ✓ Mostrar tipo y usuario autenticado.

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

- ✓ Adicionar usuario.
- ✓ Eliminar Usuario.
- ✓ Buscar Usuario.
- ✓ Modificar Usuario.
- ✓ Mostrar Usuarios.

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

- ✓ Adicionar Docente.
- ✓ Eliminar Docente.
- ✓ Modificar Docente.
- ✓ Mostrar Docentes.

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

- ✓ Adicionar Proyectos Productivos.
- ✓ Eliminar Proyectos Productivos.
- ✓ Modificar Proyectos Productivos.
- ✓ Mostrar Proyectos Productivos.

RF5. El sistema debe ser capaz de: Gestionar Facultad.

- ✓ Adicionar Facultades.

- ✓ Eliminar Facultades.
- ✓ Modificar Facultades.
- ✓ Mostrar Facultades.

RF6. El sistema debe ser capaz de: Gestionar Laboratorios

- ✓ Adicionar Laboratorios.
- ✓ Eliminar Laboratorios.
- ✓ Modificar Laboratorios.
- ✓ Mostrar Laboratorios.

RF7. El sistema debe ser capaz de insertar datos sobre Solicitudes de Registro de Salvas.

RF8. El sistema debe ser capaz de visualizar reportes sobre las Salvas existentes en la UCI.

RF 9. El sistema debe ser capaz de listar todas las salvas existentes.

RF10. El sistema debe ser capaz de eliminar cualquier registro de salva de las existentes.

RF11. El sistema debe ser capaz de generar los ficheros de configuración de salva del bacula.

RF12. El sistema debe ser capaz de eliminar los ficheros de configuración de salva del bacula.

RF13. El sistema debe ser capaz de modificar un registro de salva.

RF14. El sistema debe ser capaz de: Gestionar Plataformas

- ✓ Adicionar Plataformas.
- ✓ Eliminar Plataformas.
- ✓ Modificar Plataformas.
- ✓ Mostrar Plataformas.

2.5.2. Requerimientos No Funcionales

Para que el sistema que se propone logre ser óptimo y eficaz en el campo de acción en que se utilice, no solo es importante definir las capacidades que debe cumplir sino también aquellas cualidades que lo harán más atractivo al cliente, usable, rápido y confiable.

Los requisitos no funcionales pueden ser más críticos que los funcionales, puesto que si un requisito funcional no se cumple, el sistema se degrada, pierde eficacia, y puede no responder a la totalidad de los requerimientos del usuario, pero en cambio si un requisito no funcional no se cumple, el sistema puede inutilizarse.

A continuación se expondrán los requisitos no funcionales que el sistema propuesto debe cumplir.

Usabilidad

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

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

Rendimiento

El sistema debe ser capaz de ejecutarse con todos sus componentes instalados en una sola estación de trabajo.

Portabilidad

El sistema debe ser Multiplataforma.

Confiabilidad y Seguridad

El sistema debe tener la capacidad de darle la seguridad al usuario de que la información solo será vista por personal autorizado.

La información manejada por el sistema deberá estar protegida de acceso no autorizado. Se logra a través de la implementación del mecanismo de autenticación del sistema.

El sistema de gestión de bases de datos debe tener soporte para recuperación ante fallos y errores.

Mantenimiento y actualización

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

Desempeño y escalabilidad

El tiempo de respuestas debe ser de 3.0 segundos como máximo.

El sistema podrá soportar hasta 150 clientes online al mismo tiempo.

Requerimiento de software

Cliente

- ✓ Navegador Mozilla Firefox 3.0 ó superior, ó compatible.

Servidor

- ✓ Sistema operativo Linux (Debian, Ubuntu) ó windows server (2000 ó superior).
- ✓ Apache Web Server para Php5 2.2.
- ✓ SGBD PostgreSQL 8.2.
- ✓ Framework Symfony 1.0.

Requerimiento de hardware

Servidor

- ✓ Procesador Pentium 2.8 GHz ó superior.
- ✓ 256 MB de memoria RAM mínima, recomendable (1GB ó mayor).
- ✓ Mínimo 80 GB de disco duro, recomendable (120 GB ó mayor).

Requerimiento de ayuda y documentación

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

2.6. Propuesta de solución

Haciendo uso de las habilidades y facilidades que nos brinda UML, vamos a capturar los requisitos funcionales (RF) del sistema y así lo representaremos a través de un Diagrama de Casos de Uso. Para lo mismo teniendo en cuenta lo definido anteriormente cuales serán los actores que van a interactuar con el sistema, y a la vez los Casos de Uso que me van a representar las diferentes funcionalidades.

Un Caso de Uso es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. Por otra parte un Actor no es como tal parte del sistema, sino que es un Rol de un usuario, donde éste puede intercambiar información y representa a un ser humano, software o a una máquina que interactúa con el sistema. En nuestro caso interactúan cuatro actores.

2.6.1. Actores del Sistema

Actor	Descripción
Especialista GSD	Es el actor encargado de configurar los perfiles de los usuarios del sistema.
Usuario	Es el actor encargado de generalizar el acceso dentro del sistema y debe autenticarse para entrar al sistema. Un usuario puede ser el Asesor de

	Arquitectura y tecnología de la facultad que es el actor encargado de interactuar con el sistema con el objetivo de gestionar la información referente a su facultad. Y un Líder del proyecto que es el actor encargado de interactuar con el sistema con el objetivo de gestionar los datos referentes a su proyecto productivo.
--	---

Tabla #6: Descripción de los Actores del sistema a automatizar

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

Un DCUS representa gráficamente a los procesos y su interacción con los actores, permite comprender qué actor interviene en qué proceso.

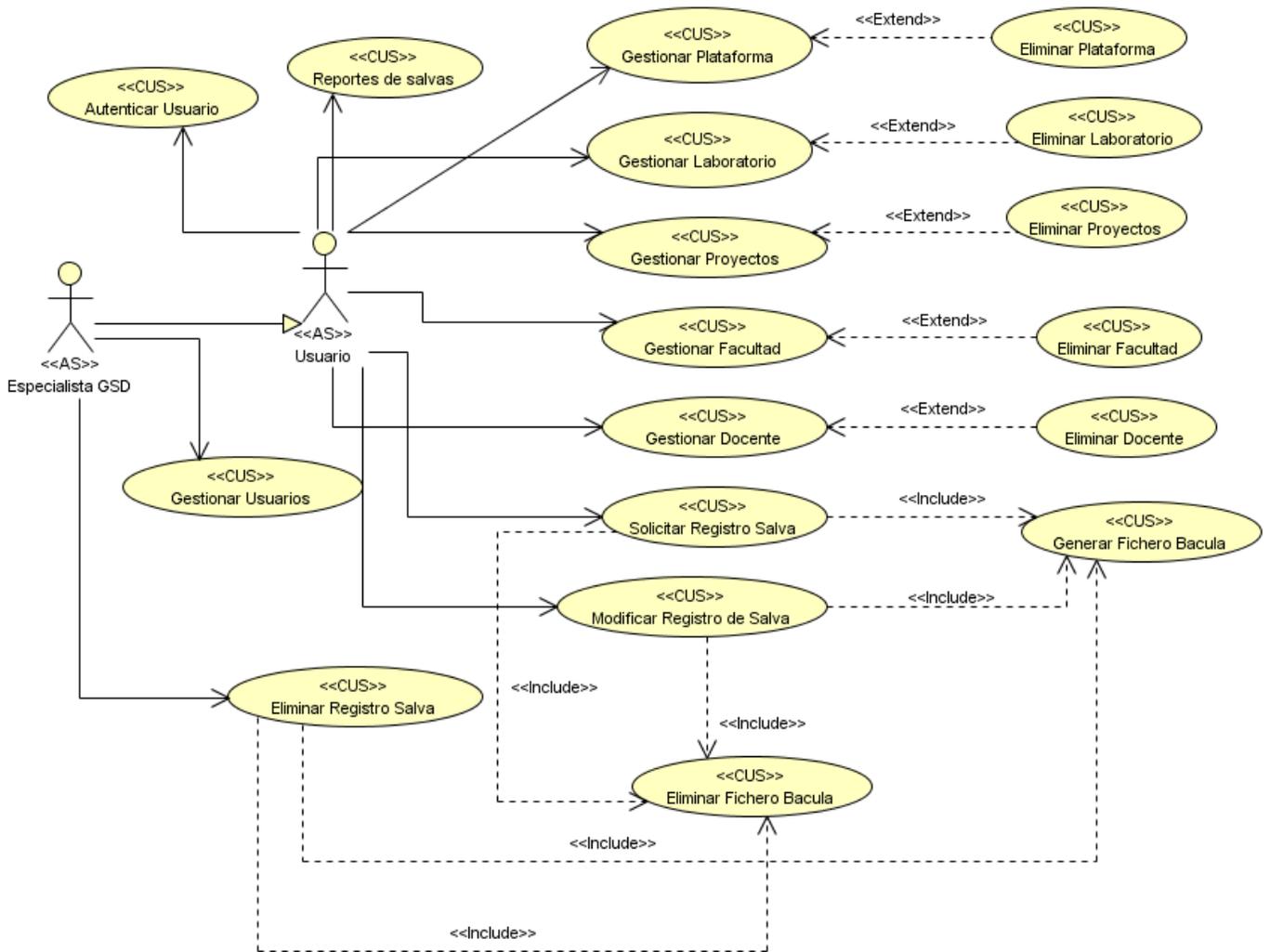


Fig. #7: Diagrama de Caso de Uso del Sistema

1.7. Descripción de los Casos de Uso del Sistema

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

2.7.1. Descripción del Caso de Uso: Autenticar Usuario

Caso de Uso:	Autenticar Usuario.
Actores:	Usuario(inicia)

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



Tabla #7: Descripción textual del Caso de Uso: Autenticar Usuario

2.7.2. Descripción del Caso de Uso: Gestionar Usuario

Caso de Uso:	Gestionar Usuario
Actores:	Especialista GSD(inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Insertar, Buscar, Modificar, Mostrar y Eliminar un Usuario del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el especialista gsd accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF2
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	Información de usuario adicionada a la Base de Datos. Información de usuario modificada en la Base de Datos. Información de usuario eliminada de la Base de Datos. Mostrar la información que contiene un usuario determinado.
Flujo Normal de Eventos	
Sección Adicionar Usuario	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción de <u>Adicionar Usuario</u> .	2. El sistema le muestra una interfaz con los datos a completar. (Usuario y contraseña).
3. El Especialista GSD ingresa los datos solicitados por el sistema.	4. El sistema verifica que los campos del formulario no estén vacíos.

	5. El sistema verifica que el usuario no esté registrado en la base de datos.
	6. El sistema procede al registro del usuario en la base de datos correspondiente y termina el CUS.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Campos requeridos"). Retorna a la acción 2.
	5.1. Muestra un mensaje de error indicando que la cuenta ya existe. ("El nombre de usuario especificado ya existe").
Sección Modificar Usuario	
1. El Especialista GSD selecciona la opción <u>Modificar Usuario</u>	2. El sistema busca todos los usuarios existentes y los muestra en una pantalla para que el Especialista GSD elija el usuario a modificar.
3. El Especialista GSD elige el usuario y pincha en el botón <u>Modificar Usuario</u> .	4. El sistema muestra una pantalla con todos los datos del usuario elegido de forma editable.
5. El Especialista GSD modifica los datos necesarios y pincha el botón <u>Aceptar</u> .	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos del usuario seleccionado y termina el CUS.
Flujos Alternos de los Eventos	
	6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Buscar Usuario	
1. El Especialista GSD selecciona la opción <u>Buscar Usuario</u> .	2. El sistema muestra una pantalla con los identificadores (nombre usuario) de los usuarios existentes.
3. El Especialista GSD elige el usuario que desea buscar.	4. El sistema realiza la búsqueda y muestra los datos del mismo.
Sección Eliminar Usuario	
1. El Especialista GSD selecciona la opción <u>Eliminar Usuario</u> .	2. El sistema busca todos los usuarios existentes y los muestra en una pantalla para que el Especialista GSD elija el usuario a eliminar.

3. El Especialista GSD elige el usuario y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.
5. El Especialista GSD confirma que desea eliminar el usuario.	6. El sistema procede a la eliminación de los datos del usuario seleccionado y termina el CUS.
Flujos Alternos de los Eventos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Sección Mostrar Usuarios	
1. El Usuario selecciona la opción de <u>Mostrar Usuarios</u> .	2. El sistema busca todos los usuarios existentes en la BD y los muestra en una pantalla y termina el CUS.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla #8: Descripción textual del Caso de Uso: Gestionar Usuario

2.7.3. Descripción del Caso de Uso: Gestionar Proyectos

Caso de Uso:	Gestionar Proyectos
Caso de Uso Asociado	Eliminar Proyecto
Actores:	Usuario (inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar proyectos productivos del menú

	de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF4
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	Información del proyecto adicionada a la Base de Datos. Información del proyecto modificada en la base de Datos. Mostrar los proyectos productivos existentes.
Flujo Normal de Eventos	
Sección Adicionar Proyecto Productivos	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Proyecto.</u>	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo proyecto productivo.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del proyecto no estén vacío.
	5. El sistema verifica que ese proyecto no esté registrado en la base de datos.
	6. El sistema registra los datos del nuevo proyecto productivo y muestra el nuevo proyecto productivo ya insertado.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. (“Llene todos los campos”).
	5.1. El sistema muestra un mensaje de error indicando que ese proyecto ya existe en la base de datos.
Sección Modificar Proyecto Productivos	
1. El Usuario selecciona la opción <u>Modificar Proyecto Productivo.</u>	2. El sistema busca todos los Proyectos Productivos de la UCI y los muestra en una

		pantalla para que el Usuario elija el Proyecto Productivo a Modificar.
3. El Usuario elije el Proyecto Productivo a Modificar y presiona el botón Modificar.		4. El sistema muestra una pantalla con todos los datos del proyecto productivo de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.		6. El sistema verifica que todos lo datos modificados estén correctos.
		7. El sistema modifica todos lo datos del proyecto productivo seleccionado.
Flujos Alternos de los Eventos		
		6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Mostrar Proyecto Productivos		
1. El Usuario selecciona la opción <u>Listar</u>		2. El sistema busca los proyectos Productivos existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico	
Prototipo de Interfaz:		

Tabla #9: Descripción textual del Caso de Uso: Gestionar Proyectos

2.7.4. Descripción del Caso de Uso: Gestionar Docentes

Caso de Uso:	Gestionar Docentes
Caso de Uso Asociado	Eliminar Docente

Actores:	Usuario (inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a las opciones de Adicionar, Modificar, o Mostrar los Docentes de la UCI. El caso de uso termina cuando el sistema realiza la operación elegida, y el Usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF3
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	Información del docente adicionada a la Base de Datos. Información del docente modificada en la base de Datos. Mostrar los docentes existentes.
Flujo Normal de Eventos	
Sección Adicionar Docente	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Docente</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo Docente.
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que ese docente no esté registrado en la base de datos.
	6. El sistema registra los datos del nuevo docente y muestra el nuevo docente ya insertado.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Llene todos los campos").
	5.1. El sistema muestra un mensaje de error indicando que ya existe ese docente en la base de datos.
Sección Modificar Docente	

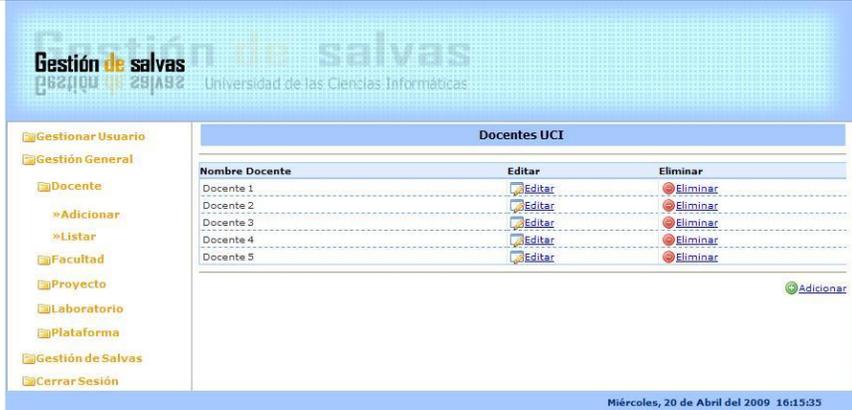
1. El Usuario selecciona la opción <u>Modificar Docente</u> .	2. El sistema busca todos los Docentes de la UCI y las muestra en una pantalla para que el Usuario elija el Docente a Modificar.
3. El Usuario elije el Docentes a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos del docente de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos lo datos del docente seleccionado.
Flujos Alternos de los Eventos	
	6.2. El sistema muestra un mensaje de error indicando el problema.
Sección Mostrar Docentes	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca los docentes existentes y los muestra en una nueva pantalla y termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	 <p>The screenshot shows a web application interface for 'Gestión de salvas' at the 'Universidad de las Ciencias Informáticas'. On the left is a navigation menu with options like 'Gestionar Usuario', 'Gestión General', 'Docente', 'Proyecto', and 'Laboratorio'. The main content area displays a table titled 'Docentes UCI' with five rows of data. Each row includes a 'Nombre Docente' (Docente 1 to 5), an 'Editar' button with a pencil icon, and an 'Eliminar' button with a red circle and minus icon. At the bottom right of the table is an 'Adicionar' button with a plus icon. The footer of the page shows the date and time: 'Miércoles, 20 de Abril del 2009 16:15:35'.</p>

Tabla #10: Descripción textual del Caso de Uso: Gestionar Docentes

2.7.5. Descripción del Caso de Uso: Gestionar Facultades

Caso de Uso:	Gestionar Facultades	
Caso de Uso Asociado	Eliminar Facultad	
Actores:	Usuario(inicia)	
Resumen:	El caso de uso comienza cuando un Usuario accede a las opciones de Adicionar, Modificar, o Mostrar las Facultades de la UCI. El caso de uso termina cuando el sistema realiza la operación elegida, y el Usuario accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF5	
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.	
Poscondiciones:	Información de la facultad adicionada a la Base de Datos. Información de la facultad modificada en la base de Datos. Mostrar las facultades existentes.	
Flujo Normal de Eventos		
Sección Adicionar Facultad		
Acción del Actor	Respuesta del Sistema	
1. El Usuario selecciona la opción de <u>Adicionar Facultad</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos de la nueva Facultad.	
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.	
	5. El sistema verifica que esa facultad no esté registrada en la base de datos.	
	6. El sistema registra los datos de la nueva facultad y muestra la nueva facultad ya insertado.	
Flujos Alternos		
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Llene todos los campos").	

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

Tabla #11: Descripción textual del Caso de Uso: Gestionar Facultades

2.7.6. Descripción del Caso de Uso: Gestionar Laboratorios

Caso de Uso:	Gestionar Laboratorios	
Caso de Uso Asociado	Eliminar Laboratorio	
Actores:	Usuario(inicia)	
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar laboratorios del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el usuario accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF6	
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.	
Poscondiciones:	Información del laboratorio adicionada a la Base de Datos. Información del laboratorio modificada en la base de Datos. Mostrar los laboratorios existentes.	
Flujo Normal de Eventos		
Sección Adicionar Laboratorio		
Acción del Actor	Respuesta del Sistema	
1. El Usuario selecciona la opción de <u>Adicionar Laboratorio</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos del nuevo laboratorio.	
3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.	
	5. El sistema verifica que ese laboratorio no esté registrado en la base de datos.	
	6. El sistema registra los datos del nuevo laboratorio y muestra el nuevo laboratorio ya insertado.	

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



Tabla #12: Descripción textual del Caso de Uso: Gestionar Laboratorios

2.7.7. Descripción del Caso de Uso: Gestionar Plataformas

Caso de Uso:	Gestionar Plataformas
Caso de Uso Asociado	Eliminar Plataforma
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Adicionar, Modificar y Mostrar plataformas del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF14
Precondiciones:	El usuario con el rol de Usuario se debe haber autenticado con anterioridad.
Poscondiciones:	Información de la plataforma adicionada a la Base de Datos. Información de la plataforma modificada en la base de Datos. Mostrar las plataformas existentes.
Flujo Normal de Eventos	
Sección Adicionar Plataforma	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de <u>Adicionar Plataforma</u> .	2. El sistema muestra la interfaz para que el Usuario introduzca los datos de la nueva plataforma.

3. El Usuario ingresa los datos solicitados por el sistema y presiona el botón aceptar.	4. El sistema verifica que los campos del formulario no estén vacíos.
	5. El sistema verifica que esa plataforma no esté registrada en la base de datos.
	6. El sistema registra los datos de la nueva plataforma y muestra la nueva plataforma ya insertada.
Flujos Alternos	
	4.1. Muestra un mensaje de error indicando que debe llenar todos los campos. ("Llene todos los campos").
	5.1. El sistema muestra un mensaje de error indicando que esa plataforma ya existe en la base de datos.
Sección Modificar Plataforma	
1. El Usuario selecciona la opción <u>Modificar Plataforma</u> .	2. El sistema busca todas las Plataformas existentes en la UCI y las muestra en una pantalla para que el Usuario elija la Plataforma a Modificar.
3. El Usuario elige la Plataforma a Modificar y presiona el botón Modificar.	4. El sistema muestra una pantalla con todos los datos de la Plataforma de forma editable.
5. El Usuario modifica los datos necesarios y pincha el botón Aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos de la Plataforma seleccionada.
Flujos Alternos de los Eventos	
	6.1. El sistema muestra un mensaje de error indicando el problema.
Sección Mostrar Plataformas	
1. El Usuario selecciona la opción <u>Listar</u>	2. El sistema busca las plataformas existentes y los muestra en una nueva pantalla y

	termina el caso de uso.
Prioridad:	Crítico
Prototipo de Interfaz:	

Tabla #13: Descripción textual del Caso de Uso: Gestionar Plataformas

2.7.8. Descripción del Caso de Uso: Eliminar Proyecto

Caso de Uso:	Eliminar Proyecto <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un proyecto del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF4
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	Información del proyecto eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Proyecto</u> Productivo.	2. El sistema busca todos los proyectos productivos existentes y los muestra en una pantalla para que el Especialista GSD elija el proyecto productivo a eliminar.

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

Tabla #14: Descripción textual del Caso de Uso: Eliminar Proyecto

2.7.9. Descripción del Caso de Uso: Eliminar Docente

Caso de Uso:	Eliminar Docente <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar docente del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF3
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.

Poscondiciones:	Información del docente eliminado de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Docente</u> .	2. El sistema busca todos los docentes existentes y los muestra en una pantalla para que el Especialista GSD elija el docente a eliminar.
3. El Especialista GSD elige el docente y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.
5. El Especialista GSD confirma que desea eliminar el docente.	6. El sistema procede a la eliminación de los datos del docente seleccionado y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

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

2.7.10. Descripción del Caso de Uso: Eliminar Facultad

Caso de Uso:	Eliminar Facultad <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar una facultad del menú de operaciones. El caso de

	uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF5
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	Información de la facultad eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Facultad</u> .	2. El sistema busca todas las facultades existentes y los muestra en una pantalla para que el Especialista GSD elija la facultad a eliminar.
3. El Especialista GSD elige la facultad y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.
5. El Especialista GSD confirma que desea eliminar la facultad.	6. El sistema procede a la eliminación de los datos de la facultad seleccionada y termina el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #16: Descripción textual del Caso de Uso: Eliminar Facultad

2.7.11. Descripción del Caso de Uso: Eliminar Laboratorio

Caso de Uso:	Eliminar Laboratorio <<extend>>	
Actores:	Especialista GSD (inicia)	
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un Laboratorio del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF6	
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.	
Poscondiciones:	1. Información del laboratorio eliminada de la base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Especialista GSD selecciona la opción <u>Eliminar Laboratorio</u> .	2. El sistema busca todos los laboratorios existentes y los muestra en una pantalla para que el Especialista GSD elija el laboratorio a eliminar.	
3. El Especialista GSD elige el laboratorio y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.	
5. El Especialista GSD confirma que desea eliminar el laboratorio.	6. El sistema procede a la eliminación de los datos del laboratorio seleccionado y termina el CUS.	
Flujos Alternos		
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.		
Prioridad:	Secundario	



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

2.7.12. Descripción del Caso de Uso: Eliminar Plataforma

Caso de Uso:	Eliminar Plataforma <<extend>>
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar una Plataforma del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF14
Precondiciones:	El usuario con el rol de Especialista GSD se debe haber autenticado con anterioridad.
Poscondiciones:	Información de la plataforma eliminada de la base de Datos.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Plataforma</u> .	2. El sistema busca todas las plataformas existentes y los muestra en una pantalla para que el Especialista GSD elija la plataforma a eliminar.
3. El Especialista GSD elige la plataforma y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que lo desea eliminar.
5. El Especialista GSD confirma	6. El sistema procede a la eliminación de los

que desea eliminar la plataforma.	datos de la plataforma seleccionada y termina el CUS.															
Flujos Alternos																
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.																
Prioridad:	Secundario															
Prototipo de Interfaz:	 <p>The screenshot shows a web application interface for 'Gestión de salvas' (Backup Management) at the 'Universidad de las Ciencias Informáticas'. On the left is a navigation menu with options like 'Gestionar Usuario', 'Gestión General', 'Docente', 'Facultad', 'Proyecto', 'Laboratorio', 'Plataforma', 'Adicionar', 'Listar', 'Gestión de Salvas', and 'Cerrar Sesión'. The main content area is titled 'Plataformas' and contains a table with the following data:</p> <table border="1"> <thead> <tr> <th>Nombre Plataforma</th> <th>Editar</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>Linux</td> <td></td> <td></td> </tr> <tr> <td>Solaris</td> <td></td> <td></td> </tr> <tr> <td>Unix</td> <td></td> <td></td> </tr> <tr> <td>Windows</td> <td></td> <td></td> </tr> </tbody> </table> <p>At the bottom right of the table is an 'Adicionar' button. The footer of the page shows the date and time: 'Miércoles, 20 de Abril del 2009 16:49:25'.</p>	Nombre Plataforma	Editar	Eliminar	Linux			Solaris			Unix			Windows		
Nombre Plataforma	Editar	Eliminar														
Linux																
Solaris																
Unix																
Windows																

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

2.7.13. Descripción del Caso de Uso: Eliminar Registro de Salva

Caso de Uso:	Eliminar Registro de Salva
Caso de Uso Asociado	Generar Fichero Bacula y Eliminar Fichero Bacula.
Actores:	Especialista GSD (inicia)
Resumen:	El caso de uso comienza cuando un Especialista GSD accede a la opción de Eliminar un Registro de Salva del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Especialista GSD accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF10
Precondiciones:	El Especialista GSD debe haberse autenticado con anterioridad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Ficheros eliminados. 2. Ficheros creados.

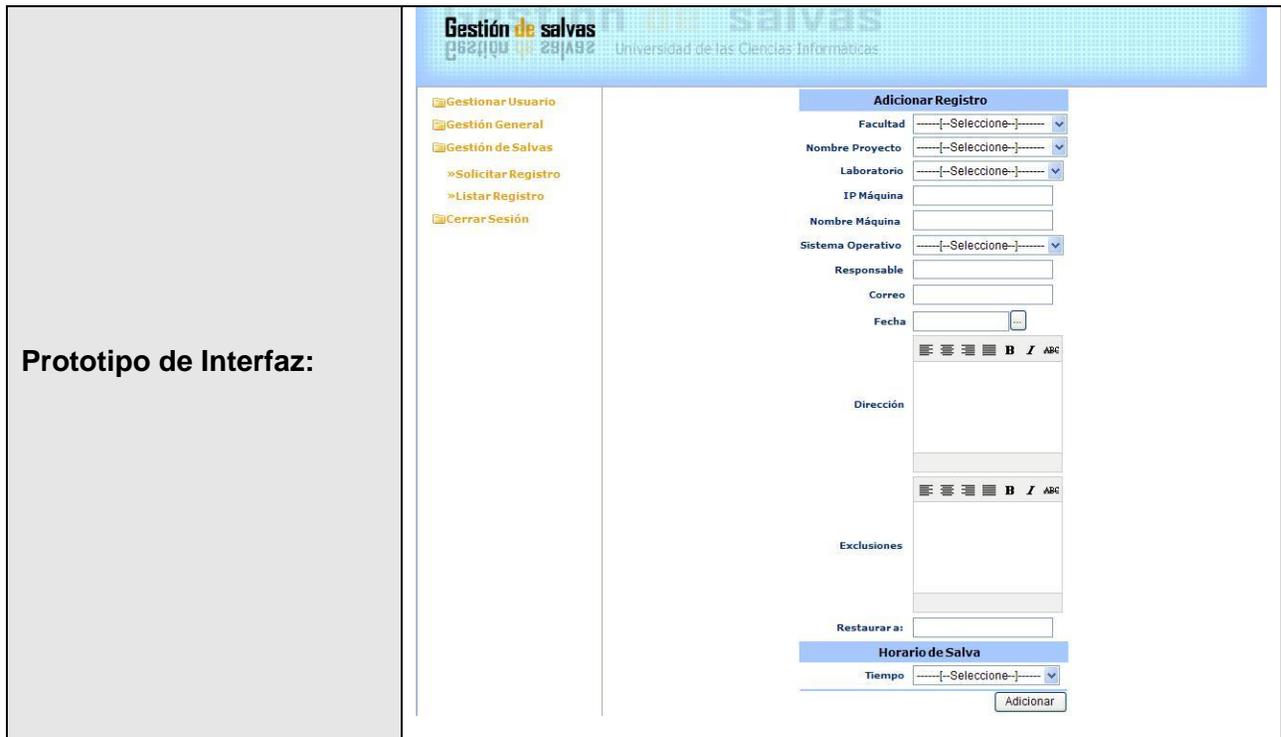
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Especialista GSD selecciona la opción <u>Eliminar Registro de Salvas</u> .	2. El sistema busca todas las salvas existentes y los muestra en una pantalla para que el Especialista GSD elija la salva a eliminar.
3. El Especialista GSD selecciona la salva y pincha en el botón Eliminar.	4. El sistema muestra una confirmación para estar seguro que la desea eliminar.
5. El Especialista GSD confirma que desea eliminar el registro de salva.	6. El sistema procede a la eliminación de los datos de la salva seleccionada. Invoca a los casos de uso Eliminar Fichero Bacula y Generar Fichero Bacula. Concluyendo así el CUS.
Flujos Alternos	
5.1. Si cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #19: Descripción textual del Caso de Uso: Eliminar Registro de Salva

2.7.14. Descripción del Caso de Uso: Solicitar Registro de Salva

Caso de Uso:	Solicitar Registro de Salva
Caso de Uso Asociado	Generar Fichero Bacula y Eliminar Fichero Bacula
Actores:	Usuario (inicia)

Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Solicitar Registro de Salvas del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el usuario accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF7	
Precondiciones:	El Usuario debe estar autenticado.	
Poscondiciones:	Información de la salva adicionada a la Base de Datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Usuario selecciona la opción <u>Solicitar Registro de Salvas</u> .	2. El sistema le muestra una interfaz con un formulario para que llene los campos a completar.	
3. El Usuario ingresa los datos en el formulario.	4. El sistema verifica que los campos requeridos no estén vacío.	
	5. El sistema procede a guardar los datos de la salva quedando ésta registrada. Invoca a los casos de uso Eliminar Fichero Bacula y Generar Fichero Bacula. Muestra la nueva salva ya insertada concluyendo así el CUS.	
Flujos Alternos		
	4.1. Si falta algún campo requerido muestra un mensaje de error.	
Prioridad:	Secundario	



Prototipo de Interfaz:

Tabla #20: Descripción textual del Caso de Uso: Solicitar Registro de Salva

2.7.15. Descripción del Caso de Uso: Modificar Registro de Salva

Caso de Uso:	Modificar Registro de Salva
Caso de Uso Asociado	Generar Fichero Bacula y Eliminar Fichero Bacula.
Actores:	Usuario(inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Modificar Registro de Salva del menú de operaciones. Y el sistema actualiza los fichero de configuración del bacula que en este se encuentra los datos del proyecto registrado. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	RF13
Precondiciones:	El usuario se debe haber autenticado con anterioridad.
Poscondiciones:	Ficheros de configuración actualizados.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de <u>Modificar Registro de Salva.</u>	2. El sistema le muestra un listado con todas las salvas para que el usuario seleccione la salva a modificar.
3. El usuario selecciona la salva y pincho el botón modificar.	4. El sistema le muestra los campos de formas editable para que modifique los datos deseados.
5. El usuario modifica los datos y pincha el botón aceptar.	6. El sistema verifica que los campos modificados estén correctos.
	7. El sistema modifica todos los datos de la salva seleccionada. Invoca a los casos de uso Eliminar Fichero Bacula y Generar Fichero Bacula. Concluyendo así el CUS.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. El sistema muestra un mensaje de error indicando el problema
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #21: Descripción textual del Caso de Uso: Modificar Registro de Salva

2.7.16. Descripción del Caso de Uso: Generar Fichero Bacula

Caso de Uso:	Generar Fichero Bacula <<include>>
Caso de Uso Base	Eliminar Registro de Salva, Solicitar Registro de Salva y Modificar Registro de Salva

Actores:	
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de realizar un Registro de Salva, Modificar un Registro de Salva o Eliminar un Registro de Salva del menú de operaciones. Y el sistema genera un fichero de configuración del bacula que en éste se encuentra los datos de la salva registrado. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Usuario accede a otra opción o simplemente sale de la aplicación.
Referencia:	R11
Precondiciones:	Que se haya ejecutado el caso de uso Solicitar Registro de Salva, Modificar Registro de Salva o Eliminar Registro de Salva.
Poscondiciones:	Ficheros de configuración creados.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. Crear ficheros Clients, Devices, Sd y Dir.
	2. Abrir cada fichero como escritura.
	3. Crear la estructura de cada fichero.
	4. Registra los datos en cada fichero.
	5. Cierro los ficheros y termina el caso de uso.
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #22: Descripción textual del Caso de Uso: Generar Fichero Bacula

2.7.17. Descripción del Caso de Uso: Eliminar Fichero Bacula

Caso de Uso:	Eliminar Fichero Bacula <<include>>
Caso de Uso Base	Eliminar Registro de Salva, Solicitar Registro de Salva y Modificar Registro de Salva
Actores:	

Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de realizar un Registro de Salva, Modificar Registro de Salva o Eliminar Registro de Salva del menú de operaciones. Y el sistema elimina los ficheros de configuración del bacula que en éste se encuentra los datos de la salva registrado. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Usuario accede a otra opción o simplemente sale de la aplicación.	
Referencia:	RF12	
Precondiciones:	Que se haya ejecutado el caso de uso Solicitar Registro de Salva, Modificar Registro de Salva o Eliminar Registro de Salva.	
Poscondiciones:	Ficheros de configuración eliminados.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
		<ol style="list-style-type: none"> 1. El sistema localiza los ficheros .dir y .sd del directorio de configuración del bacula. 2. El sistema procede a eliminarlo definitivamente de la carpeta de configuración y termina el CUS.
Prioridad:	Secundario	
Prototipo de Interfaz:		

Tabla #23: Descripción textual del Caso de Uso: Eliminar Fichero Bacula

2.7.18. Descripción del Caso de Uso: Reporte de Salva

Caso de Uso:	Reporte de Salva
Actores:	Usuario(Inicia)
Resumen:	El caso de uso comienza cuando un Usuario accede a la opción de Reporte de Salva del menú de operaciones. El caso de uso termina cuando el sistema devuelve la información solicitada, y el Usuario accede a otra opción o simplemente sale de la aplicación.

Referencia:	RF8
Precondiciones:	Que el usuario haya sido autenticado.
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario accede a la opción <u>Reporte de Salva.</u>	2. El sistema busca las salvas existentes y las muestra para que el usuario selecciona la salva de la cual quiere el reporte.
3. El Usuario selecciona la salva y pincha el botón de reporte.	4. El sistema genera el reporte en formato .pdf y lo muestra en pantalla y termina el CUS.
Prioridad:	Secundario
Prototipo de Interfaz:	

Tabla #24: Descripción textual del Caso de Uso: Reporte de Salva

1.8. Conclusiones Parciales

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

3

CAPÍTULO CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

3.1. Introducción

El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica: cómo el sistema satisface los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo.

En este capítulo se realiza el análisis y diseño (construcción) de la propuesta de solución, a través de los flujos de Análisis y Diseño e Implementación, se presenta el diagrama de clases del análisis y del diseño de los diferentes casos de usos definidos en el capítulo anterior, a demás de los diagramas de interacción correspondientes a cada uno de ellos, se muestra el modelo de datos. Finalmente se realiza el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación y el de componentes para una mejor descripción de la solución propuesta.

3.2. Modelo de Análisis

En la construcción del Modelo de Análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas, es el resultado de la actividad de analizar los casos de uso. Entre las principales ventajas que aporta al proceso de desarrollo de un software es que suaviza la transición al diseño, sirve para tener una visión general de la propuesta de sistema, y entre otras cosas apoya el cambio a otra plataforma de programación puesto que no está ligado a un lenguaje en particular.

El objetivo principal de este flujo son los diagramas de clases de análisis, los cuales muestran que clases participan en las realizaciones de los distintos casos de usos, constituyendo la base sobre la cual debe realizarse el Modelo de Diseño.

3.3. Diagramas de Clases del Análisis

Estos diagramas se encuentran en los [Anexos2](#).

3.3.1. Diagramas de Interacción del Análisis

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si consideramos el “interior” del sistema, un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje a algún otro objeto, y de esa forma los objetos implicados interactuarán para llevar a cabo el caso de uso (RUMBAUHG et al. 2000a).

Los diagramas de interacción están constituidos por dos tipos: los diagramas de Colaboración y los diagramas de Secuencia. Ambos expresan información similar, pero en una forma diferente.

Diagrama de Secuencia

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones.

Diagrama de Colaboración

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. Representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un Procedimiento.

Estos diagramas de interacción se encuentran en los [Anexos3](#).

3.4. Modelo de Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación.(RUMBAUHG et al. 2000b).

Para un mayor entendimiento del diseño de las clases realizado, es conveniente hacer una breve descripción del funcionamiento del framework que se utilizó, pues al estar el diseño completamente ligado al lenguaje de programación como ya se ha dicho, sin conocer como funciona al menos de manera general el Symfony es muy difícil poder entender el flujo que describen los diagramas.

Descripción general del funcionamiento del Symfony

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

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. En primer lugar, el controlador frontal y el layout son comunes para todas las acciones de la aplicación. Se pueden tener varios controladores y varios layouts, pero solamente es obligatorio tener uno de cada tipo. El controlador frontal es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática, es además el único punto de entrada a la aplicación, carga la configuración y determina la acción a ejecutarse. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita (o pinchada) por el usuario.

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

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

Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. La librería Propel se encarga de esta generación automática, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario. Cuando Propel encuentra restricciones de claves foráneas (o externas) o cuando encuentra datos de tipo fecha, crea métodos especiales para acceder y modificar esos datos.

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

La vista se encarga de producir las páginas que se muestran como resultado de las acciones. La vista en Symfony está compuesta por diversas partes, estando cada una de ellas especialmente preparada para que pueda ser fácilmente modificable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

Normalmente se trabaja con las plantillas (que son la presentación de los datos de la acción que se está ejecutando) y con el layout (que contiene el código HTML común a todas las páginas). Estas partes están formadas por código HTML que contiene pequeños trozos de código PHP, que normalmente son llamadas a los diversos helpers disponibles. Los helpers son funciones de PHP que devuelven código HTML y que se utilizan en las plantillas.

El Layout, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en él, o si se mira desde el otro punto de vista, decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “decorator”. Normalmente se utiliza el layout para mostrar la navegación, el logotipo del sitio, etc. Incluso es posible definir más de un layout y decidir en cada acción cuál se va a utilizar.

3.4.1. Diagrama de Clases del Diseño

Estos diagramas se encuentran en los [Anexos4](#).

3.5. Diseño de la Base de Datos

El diseño de la Base de Datos ocupa un lugar fundamental dentro del proceso de desarrollo del software, con el auge de las tecnologías los sistemas informáticos en casi su totalidad se apoyan en bases de datos para el manejo y almacenamiento de la información. Luego de haber definido las clases del diseño que intervienen en los casos de uso, corresponde analizar cuales poseen un carácter permanente y a partir de ello realizar el diagrama de clases persistentes, para luego desarrollar el modelo de datos del sistema.

3.5.1. Diagrama de Clases Persistentes

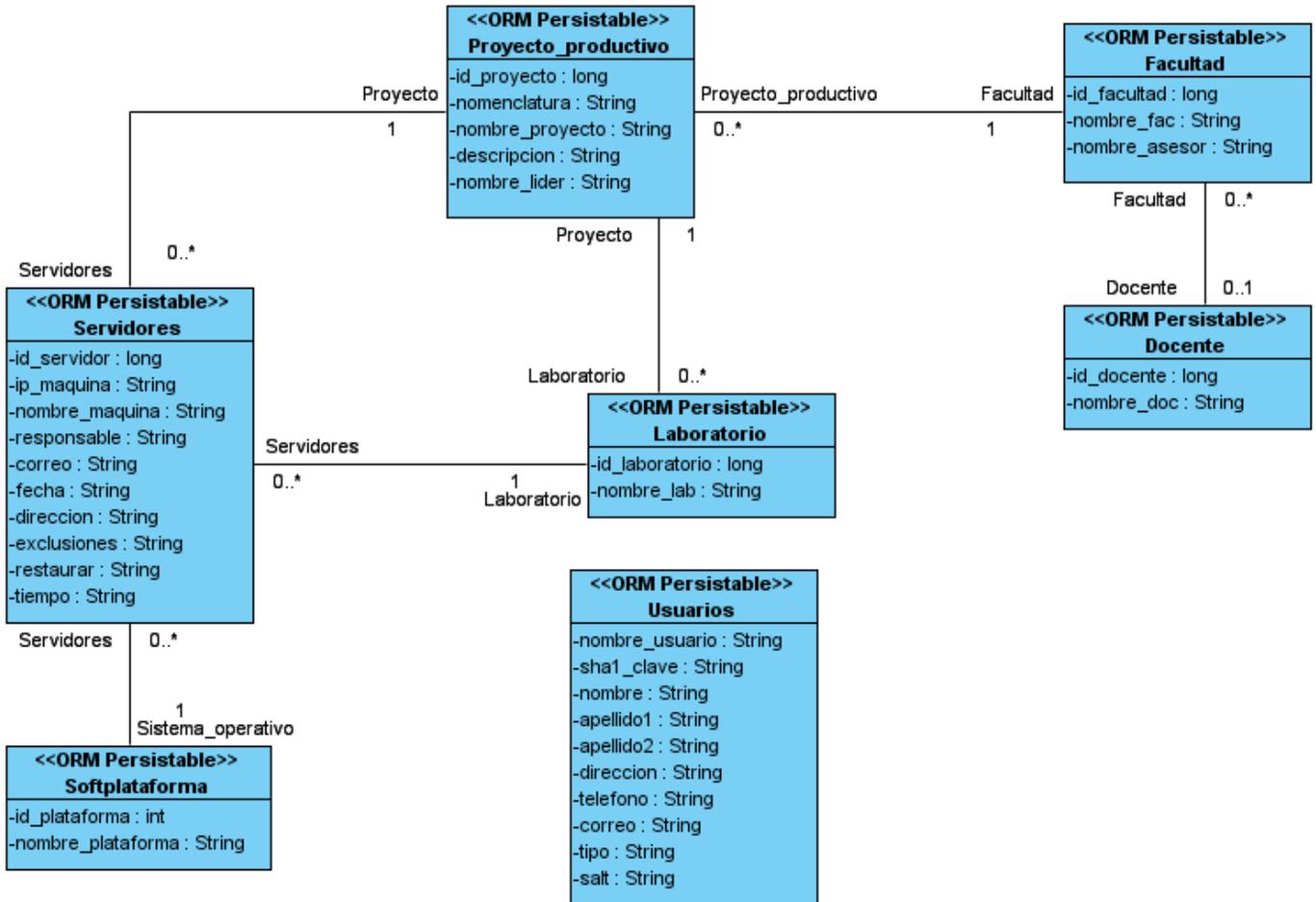


Fig. #8: Diagrama de Clases Persistentes

3.5.2. Diagrama Entidad Relación

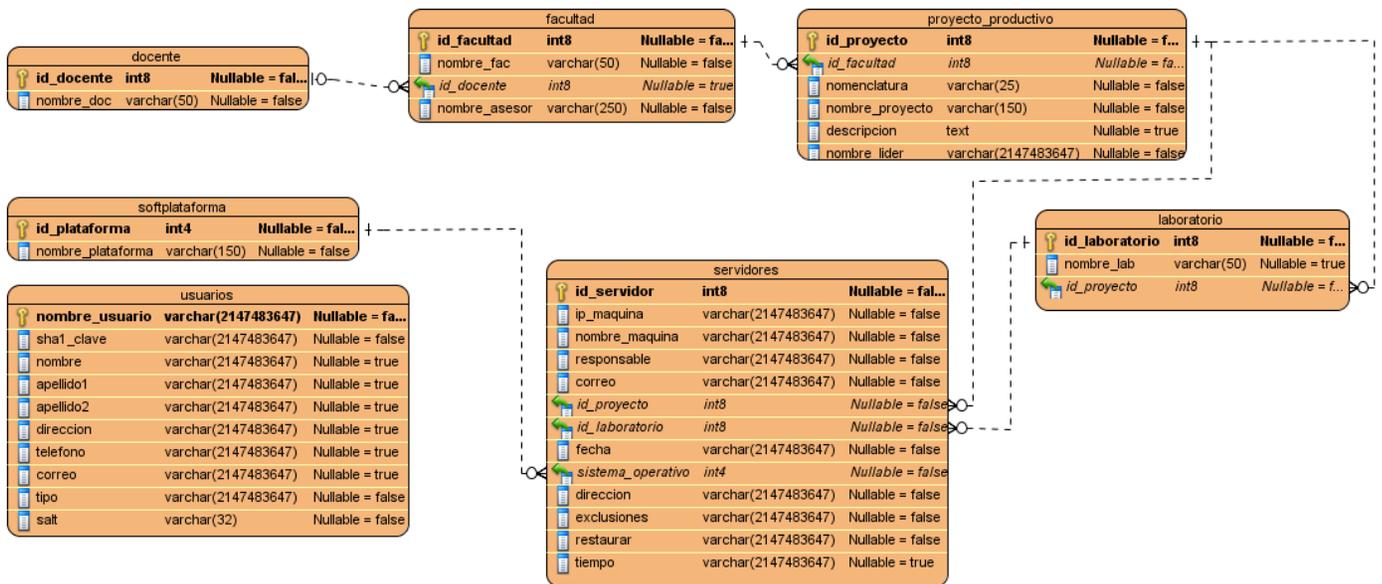


Fig. #9: Diagrama Entidad Relación

3.6. Diagrama de Despliegue

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

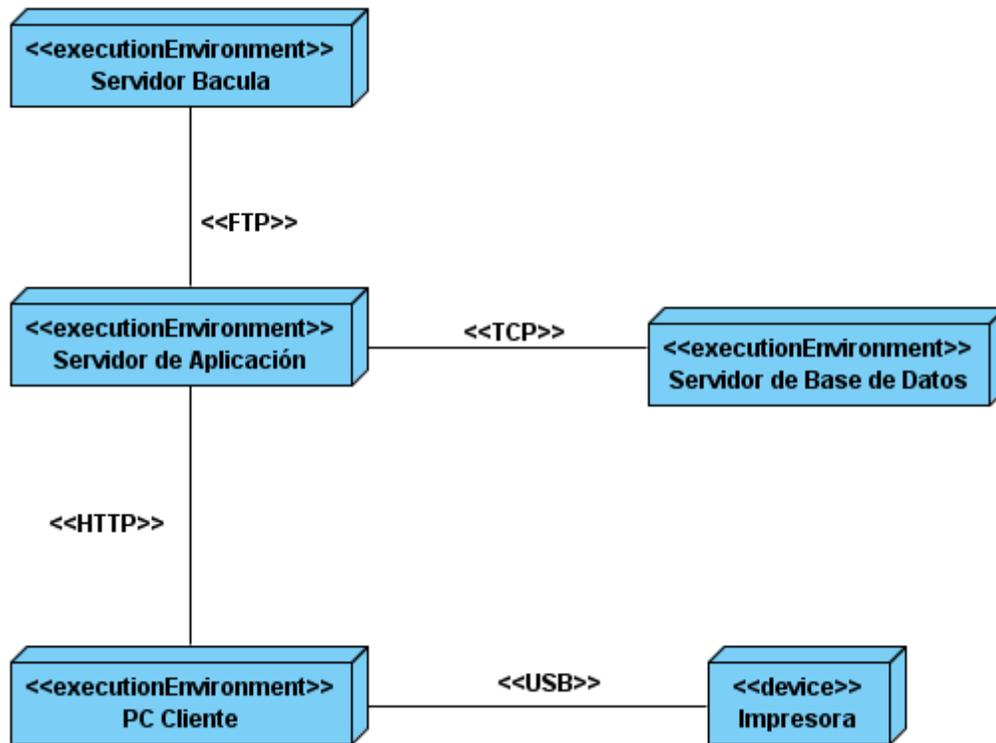


Fig. #10: Diagrama de Despliegue

3.7. Modelo de Implementación

Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc.

Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. (PRESSMAN 2002; RUMBAUHG et al. 2000b)

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas. (RUMBAUHG et al. 2000a)

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. A continuación se muestra el diagrama de componentes correspondiente al sistema que se propone en la presente investigación.

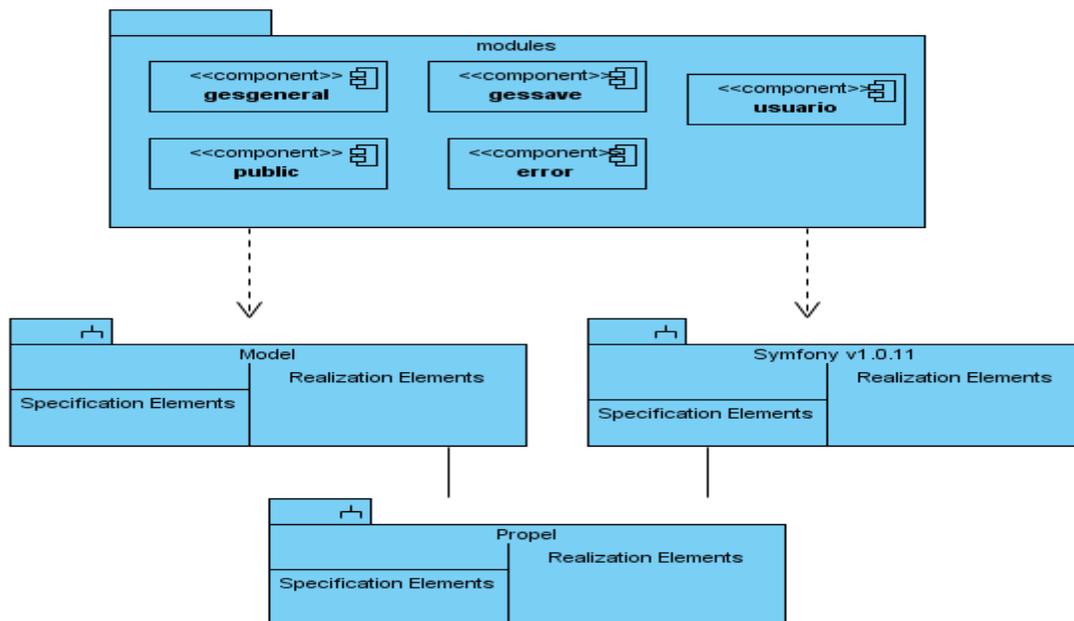


Fig. #11: Diagrama de Componentes

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

Vista Detallada: *Paquete Model.*

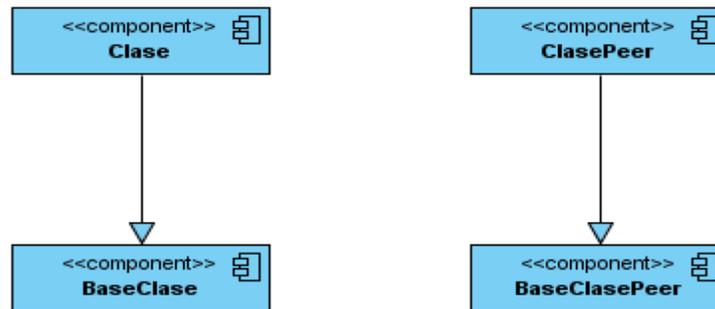


Fig. #12: Paquete Model

Vista Detallada: *Paquete Módulo.*

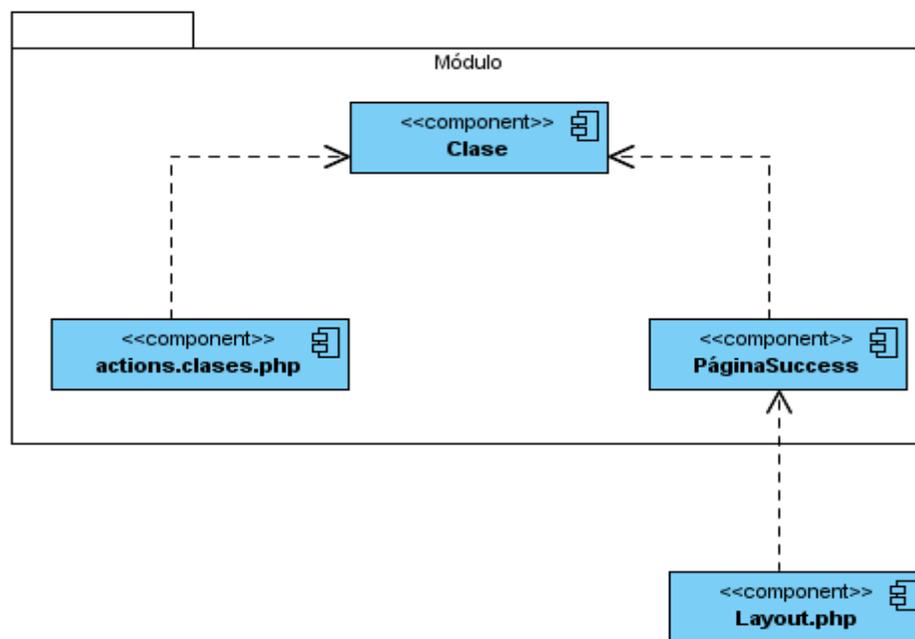


Fig. #13: Paquete Módulo

3.8. Conclusiones Parciales

Con el desarrollo de este capítulo han quedado realizados los principales artefactos de los flujos de trabajo de Análisis, Diseño e Implementación, constituyendo la base fundamental sobre la que ha de sustentarse la construcción del software.

CONCLUSIONES

Durante el desarrollo de la presente investigación los autores de este trabajo han arribado a las siguientes conclusiones:

Con la implantación del sistema propuesto se logra optimizar la gestión de los servidores de salvadas de los proyectos productivos de la UCI. Con la utilización del sistema propuesto se logran realizar con mayor rapidez y menor esfuerzo. Además de que la información se encuentra protegida, ya que solo pueden acceder a ella aquellos que tienen los privilegios requeridos y la gestión de la misma se torna menos engorrosa; todo ello contribuye a que los datos sean más consistentes y por ende más confiables.

La aplicación informática desarrollada, brinda las posibilidades inherentes a la informatización, y hace que el trabajo se vuelva más simple, y a la vez que sean más reales los resultados que se obtengan a partir de su buen uso.

Se desarrolló una aplicación Web que:

- Permite una navegación placentera a través de las diferentes secciones de trabajo.
- Cuenta con una interfaz cómoda, amigable, de fácil entendimiento y sobre todo de facilidad de uso por los usuarios.
- Mejora la calidad de los procesos de salvadas de los proyectos productivos de la UCI.
- Aumenta considerablemente la velocidad de acceso a la información y búsqueda de la misma tanto por los especialistas de gsd como por los usuarios de cada uno de los proyectos productivos.
- La integridad, disponibilidad, autenticidad y confidencialidad de los datos se logró, realizando un control de las acciones llevadas a cabo por los usuarios en el sistema y estableciendo niveles de acceso dada las responsabilidades de los usuarios.

RECOMENDACIONES

Sobre la presente investigación los autores recomiendan:

- Someter el sistema a pruebas de calidad de software para su posterior instalación.
- Ampliar las funcionalidades del sistema de manera que pueda utilizarse en otras direcciones que se involucran en los procesos a los que responde.
- Que este material sea tomado como consulta por el personal (dígase técnicos o profesionales) que vayan a desarrollar un sistema similar.
- Continuar desarrollando el sistema con el objetivo de hacer su uso más flexible, posibilitando su aplicación en otras instituciones del país donde existan infraestructura productiva de software.
- Realizar un sistema que posibilite el proceso de mantenimiento y actualización de forma periódica y continua, logrando de esta forma que se mantenga la confidencialidad y funcionamiento insuperable del sistema y de la información que se gestiona mediante el mismo.

REFERENCIAS BIBLIOGRÁFICAS

- Cooperativa Radio cooperativa.cl [Online]. - 2006. - 2009.
http://www.cooperativa.cl/p4_noticias/antialone.html?page=http://www.cooperativa.cl/p4_noticias/site/artic/20060712/pags/20060712053348.html.
- Definicion definicion.org [Online]. - 2009.
- Dictionary The Free thefreedictionary.com [Online]. - 2006. - <http://www.thefreedictionary.com/backup..>
- Glossary glossaty.its.bldrdoc.gov [Online]. - agosto 23, 1996. - febrero 9, 2009. - http://glossaty.its.bldrdoc.gov/fs-1037/dir004/_0509.htm..
- Hoy diariohoy.net [Online]. - 2006. - febrero 9, 2009. - <http://pdf.diariohoy.next/2000/01/13/pdf/u0405.pdf>.
- librosweb.es librosweb.es [Online]. - 2008. - http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.
- Maestrosdelweb.com Maestrosdelweb.com [Online]. - 2008. - [http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/..](http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/)
- Masadelante masadelante.com [Online]. - 2008. - www.masadelante.com/faq-servidor-web.html..
- Sanchez Andres HP Storage Mirroring [Book Section]. - 2007.
- Searchstorage searchstorage.techtarget.com [Online]. - marzo 7, 2007. - febrero 9, 2009. - http://searchstorage.techtarget.com/sDefinition/0,,sid5_gci211633,00.html..
- Wikipedia wikipwdia.org [Online]. - 2009. - http://es.wikipwdia.org/wiki/Copia_de_Seguridad..
- Lenguaje de Programación. [En línea] [Citado el: 20 de Enero de 2009.] <http://entren.dgsca.unam.mx/introduccion/lenguajes.html>.
- CANALES MORA. Adictos al Trabajo. 2004, nº Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.
- PRESSMAN, R. S. *Ingeniería del software. Un enfoque práctico*, 5ta ed. McGraw-Hill, 2002. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>

MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*, javaHispano, 2003. [2007]. Disponible en: <http://www.javahispano.org/articles.article.action?id=76>

RUMBAUHG, J.; I. JACOBSON, *et al.* *El Lenguaje Unificado de Modelado. Manual de referencia*. Madrid, Pearson Educación, 2000a. p. 84-7829-037-0
---. *El Proceso Unificado de Desarrollo de Software*, Pearson Educación, 2000b. [2007]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>

HERNÁNDEZ ORALLO, E. *El Lenguaje Unificado de Modelado (UML)*, Universidad Politécnica de Valencia, 2001. [2007]. Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, Prentice Hall, 1999. [2007]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00061.pdf>

VISUAL PARADIGM. *Visual Paradigm for UML*, 2007. [2007]. Disponible en: <http://www.visualparadigm.com/product/vpuml/>

ADRFORMACIÓN.COM. *Curso de PHP*, Disponible en: http://www.adrformacion.com/curso/php/leccion1/Introduccion_php.htm.

RODAS HINOSTROZA. Características de PHP. 2007, nº Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

2009, The British Standards Institution. [Online] BSI. [Cited: 01 30, 2009.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>. 1.

masadelante, sitio. masadelante.com. [Online] [Cited: 01 30, 2009.] <http://www.masadelante.com/faq-servidor.htm>.

BIBLIOGRAFÍA CONSULTADA

2006. igmus.org. [Online] Febrero 14 , 2006. [Cited: Enero 20, 2009.] <http://old.igmus.org/code/>.

Nathan Rosenquist, David Cantrell, and others. 2003-2007. rsnapshot. *Website hosted at Cybersource.* [Online] 2003-2007. [Cited: Enero 23, 2009.] www.rsnapshot.org.

2005 . storebackup. *sourceforge.* [Online] Agosto 29, 2005 . [Cited: Enero 18, 2009.] <http://sourceforge.net/projects/storebackup>.

2008. The Advanced Maryland Automatic Network Disk Archiver . *Amanda.* [Online] Diciembre 24, 2008. [Cited: Enero 18, 2009.] <http://www.amanda.org/>.

GROUP, P. G. D. Sitio oficial de PostgreSQL, 2007 [Disponible en: <http://www.postgresql.org/>]

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

HOJTSY, G.; Á. PARDO, et al. Manual de PHP, 2006. [2008]. Disponible en: <http://www.php.net/manual/es/>

DESARROLLOWEB. Programación en PHP, 2007. [2008]. Disponible en: <http://www.desarrolloweb.com/manuales/12/>

NAVARRA, U. D. Tutorial de JavaScript 2007. [2008]. Disponible en: <http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>

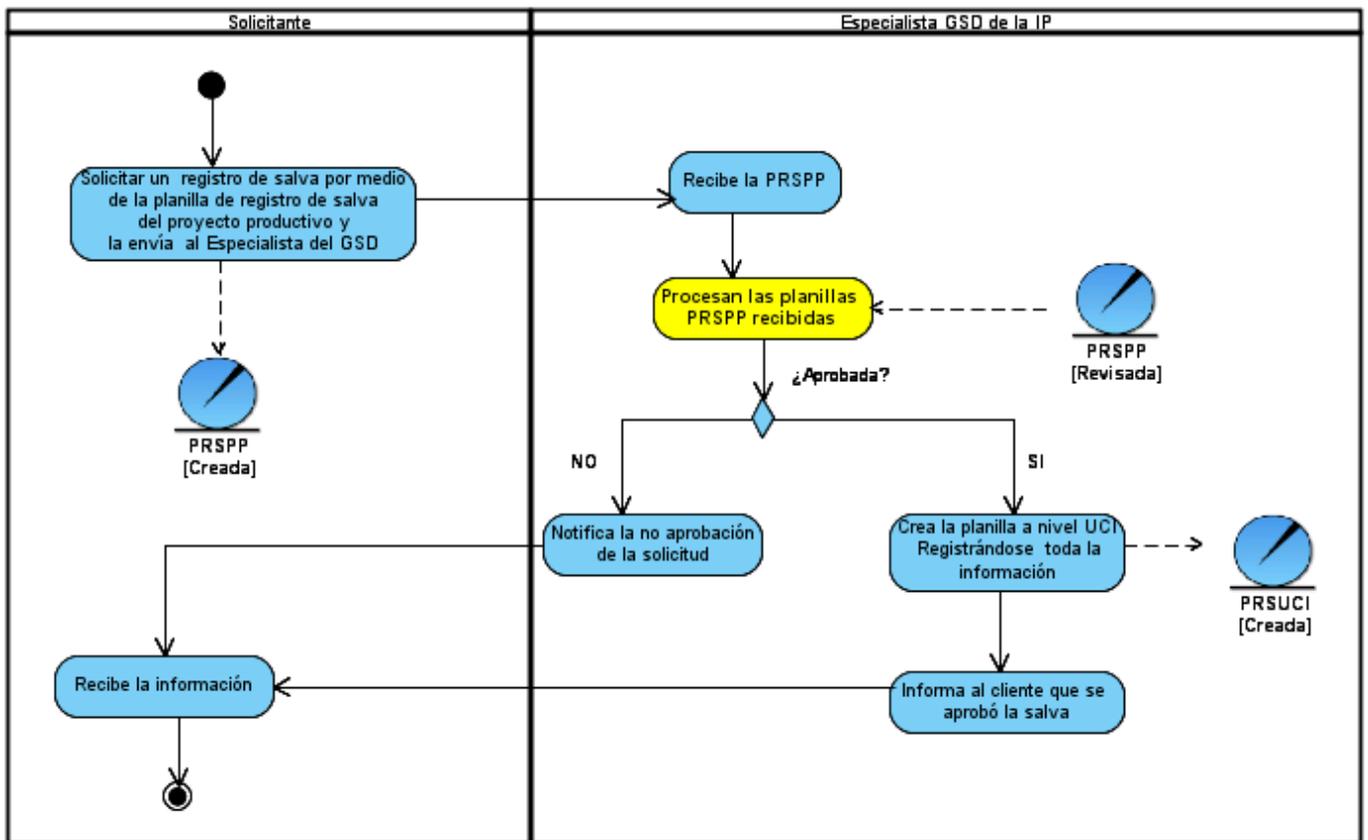
Potencier, F., & Zaninotto, F. (2007). *Symfony la guia definitiva.*

Programación extrema. [cited 2008 marzo]; Available from: www.programacionextrema.org.

ANEXOS

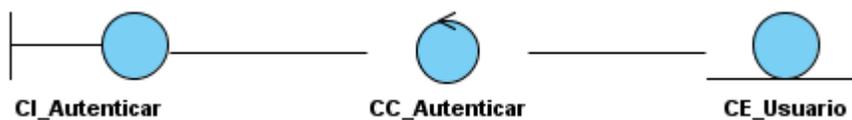
Anexo1. Realización de los casos de uso del negocio: Diagramas de Actividades.

Diagrama de Actividades del Caso de Uso: Solicitar Registro de Salva.

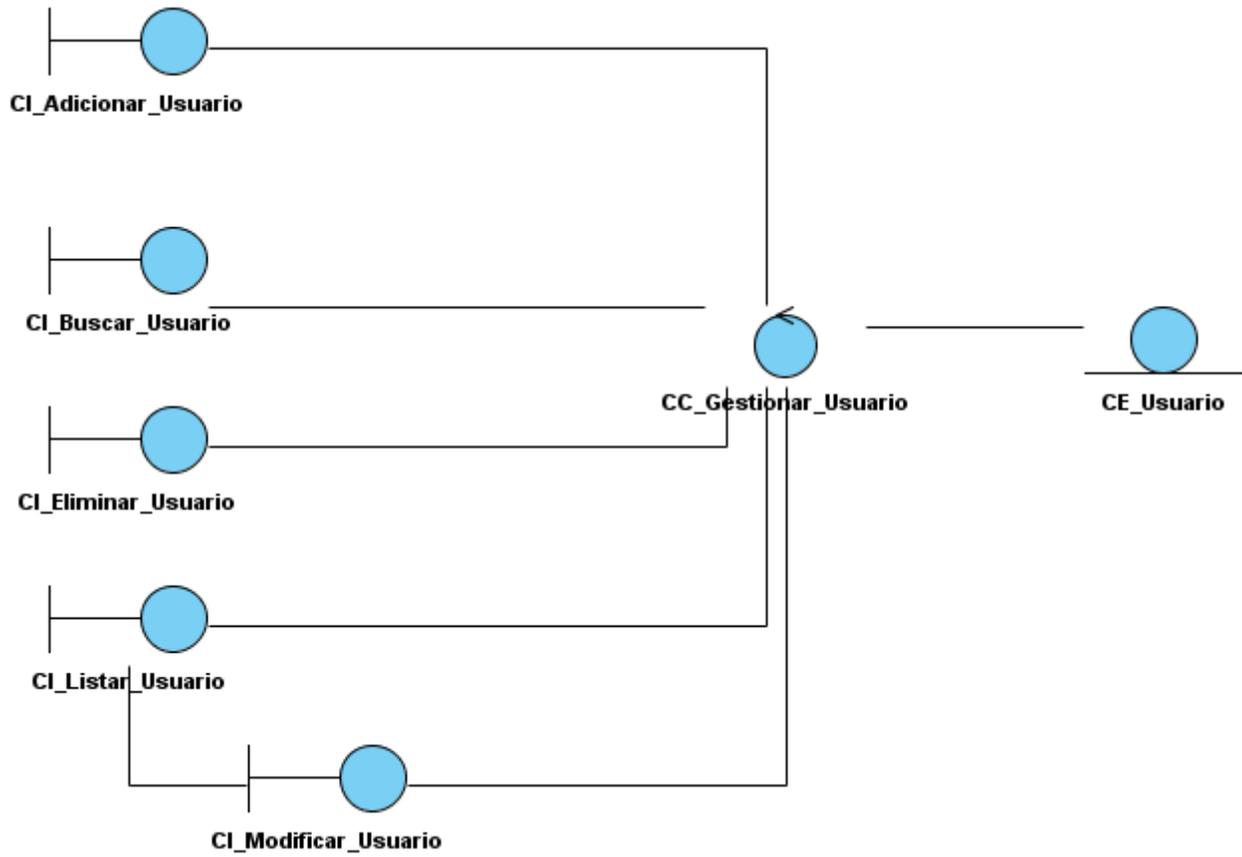


Anexo 2. Diagramas de Clases del Análisis

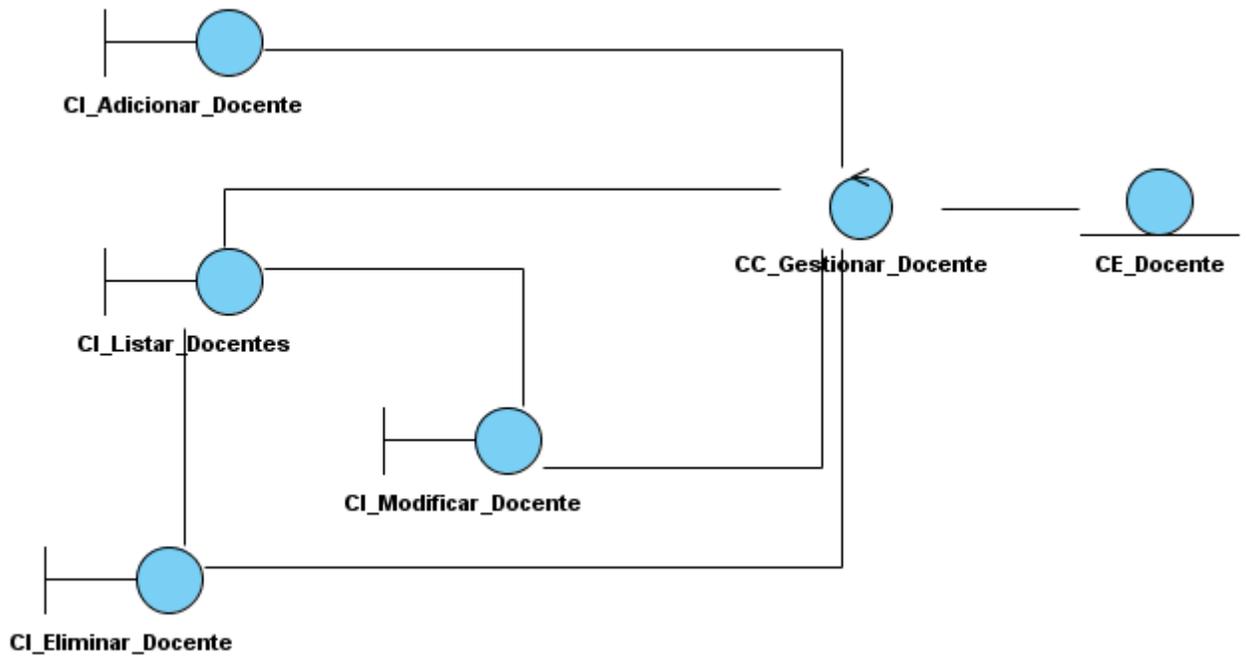
Diagramas de Clases del Análisis del Caso de Uso: Autenticar Usuario.



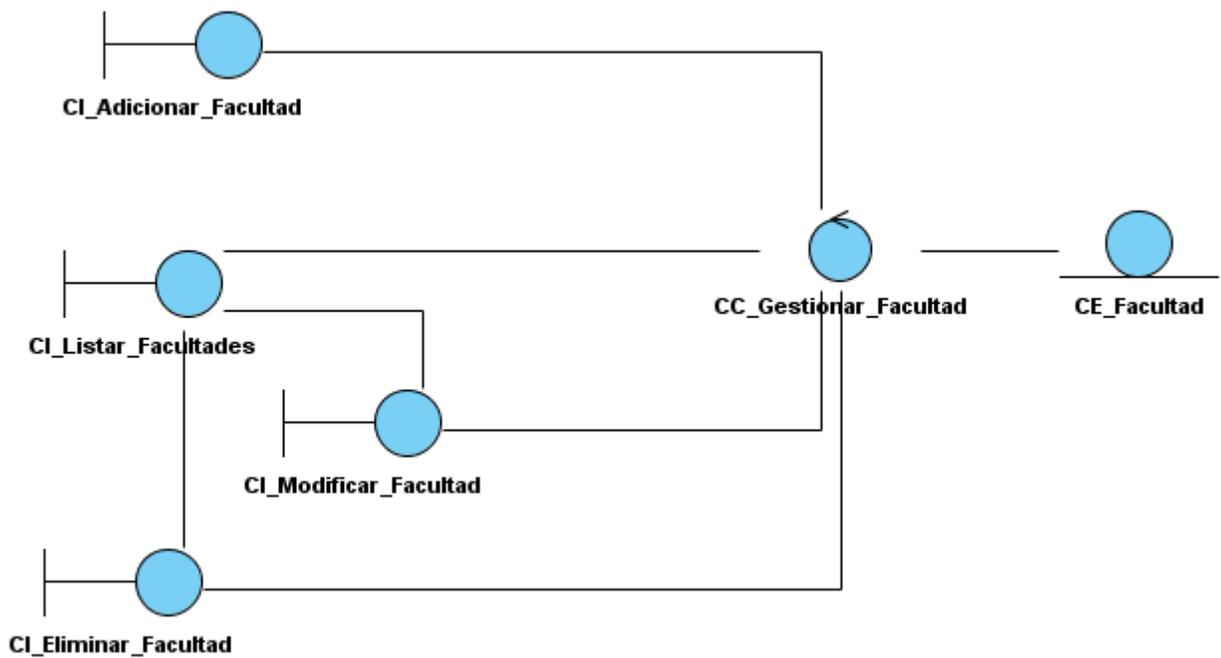
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Usuario.



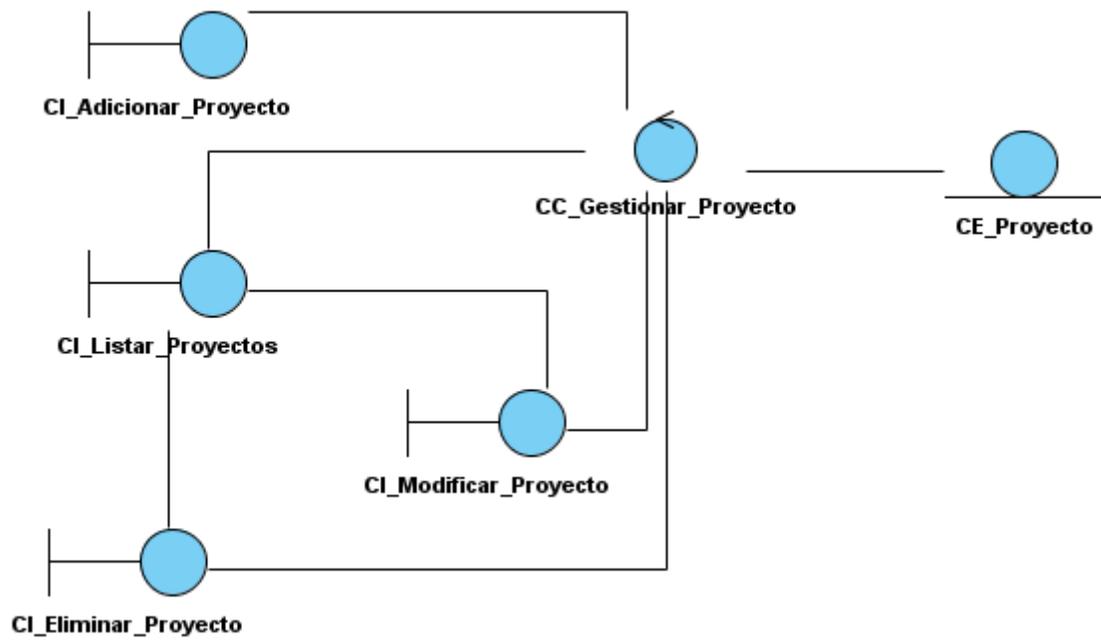
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Docente.



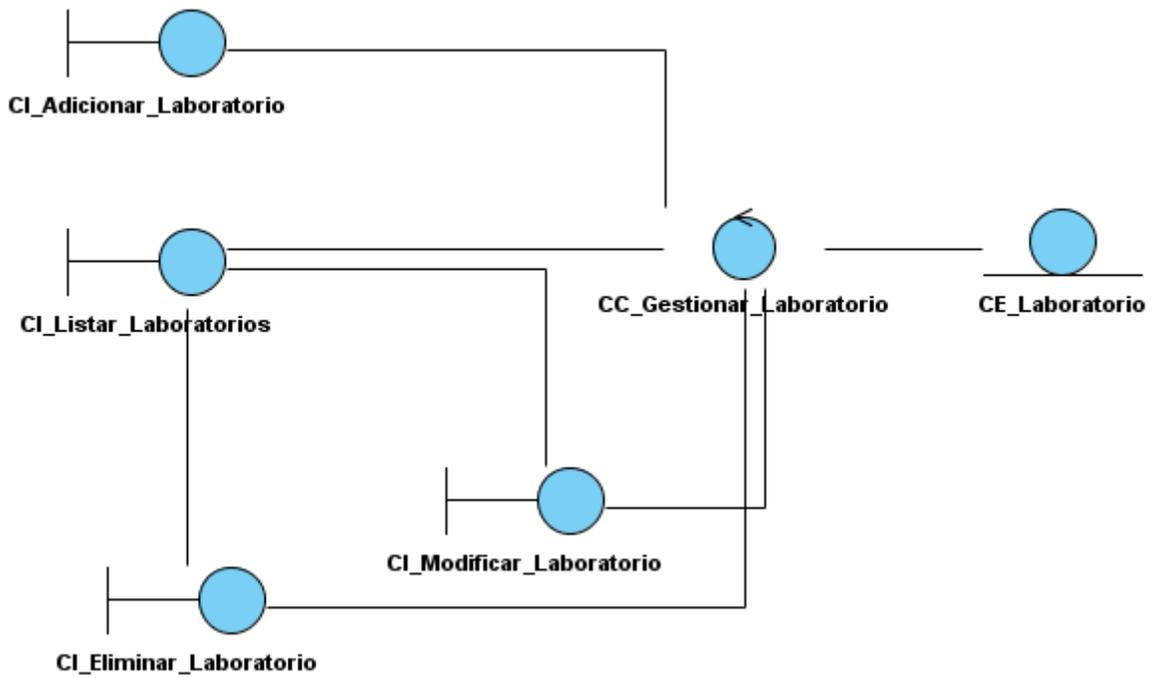
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Facultad.



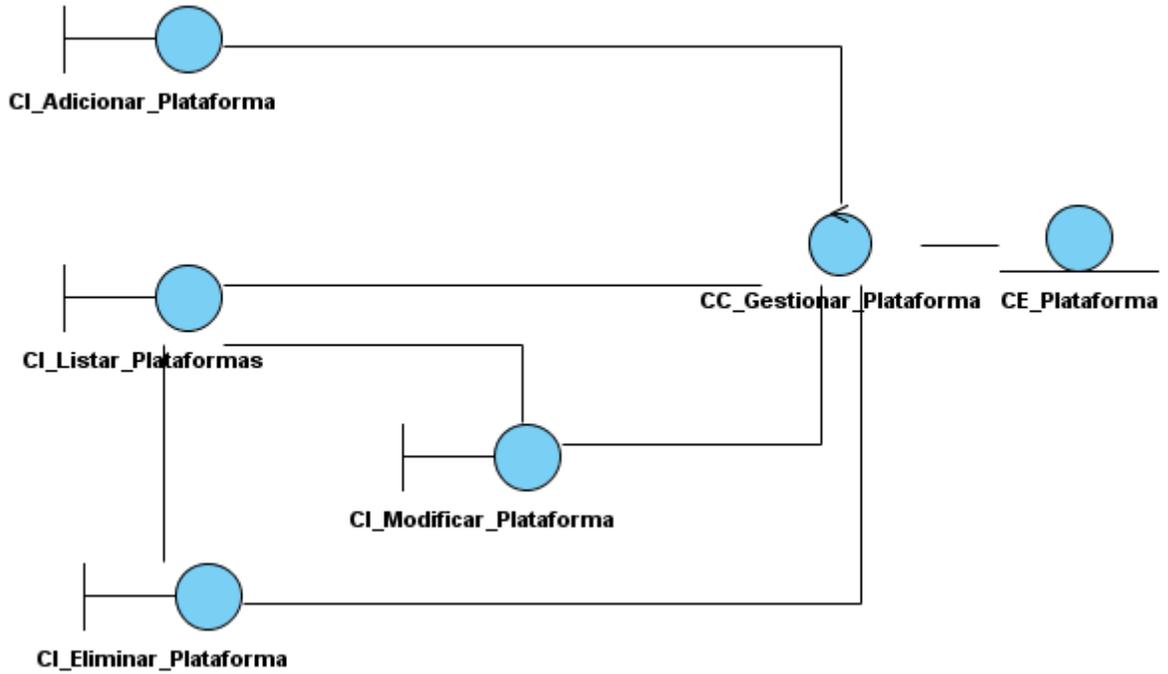
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Proyecto.



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



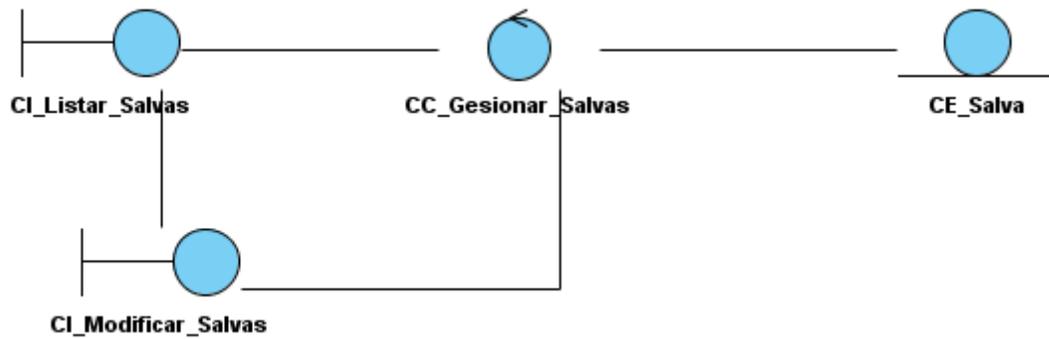
Diagramas de Clases del Análisis del Caso de Uso: Gestionar Plataforma.



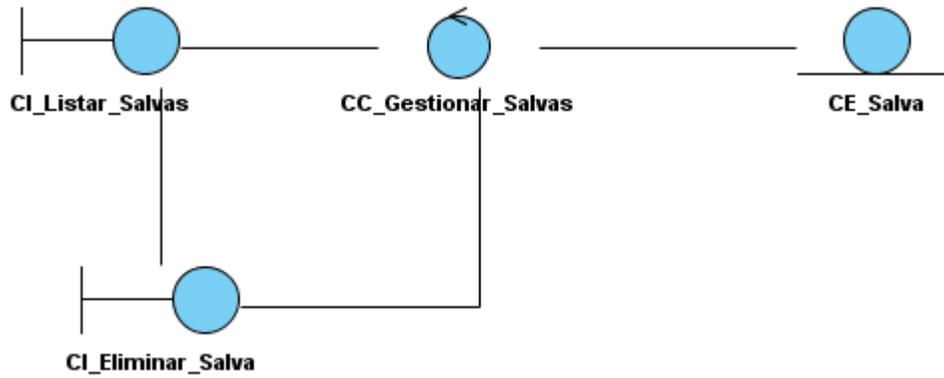
Diagramas de Clases del Análisis del Caso de Uso: Solicitar Registro de Salva.



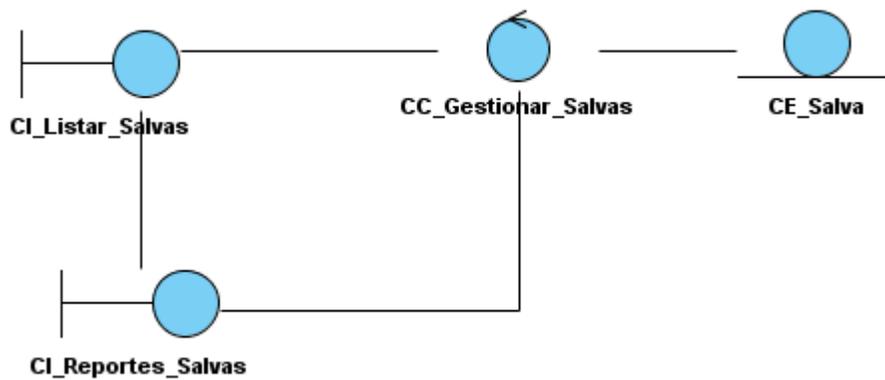
Diagramas de Clases del Análisis del Caso de Uso: Modificar Registro de Salva.



Diagramas de Clases del Análisis del Caso de Uso: Eliminar Registro de Salva.



Diagramas de Clases del Análisis del Caso de Uso: Reporte de Salva.



Anexo3. Diagramas de Interacción del Análisis.

Diagrama de Colaboración del Caso de Uso: Autenticar Usuario.

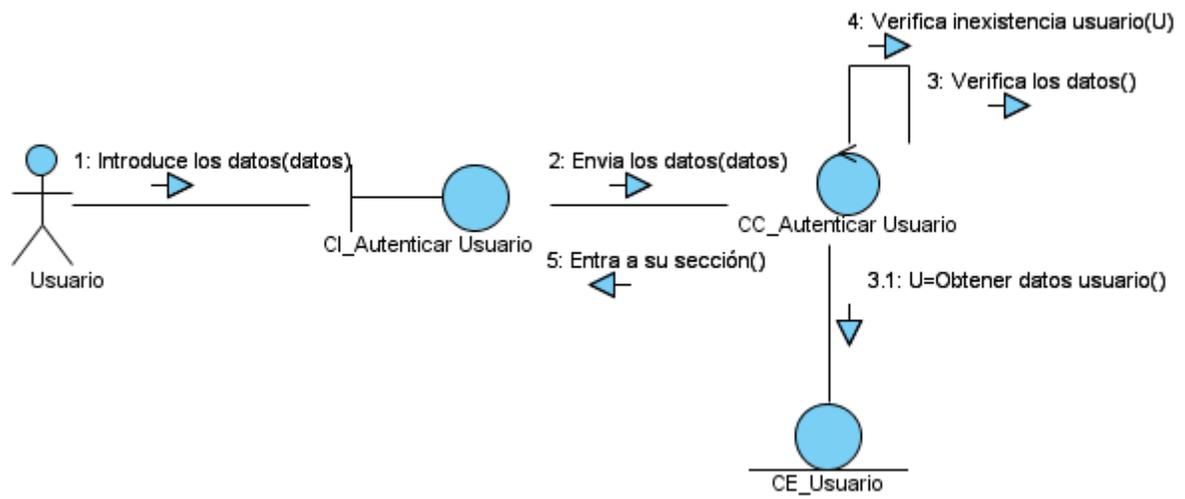


Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Adicionar.

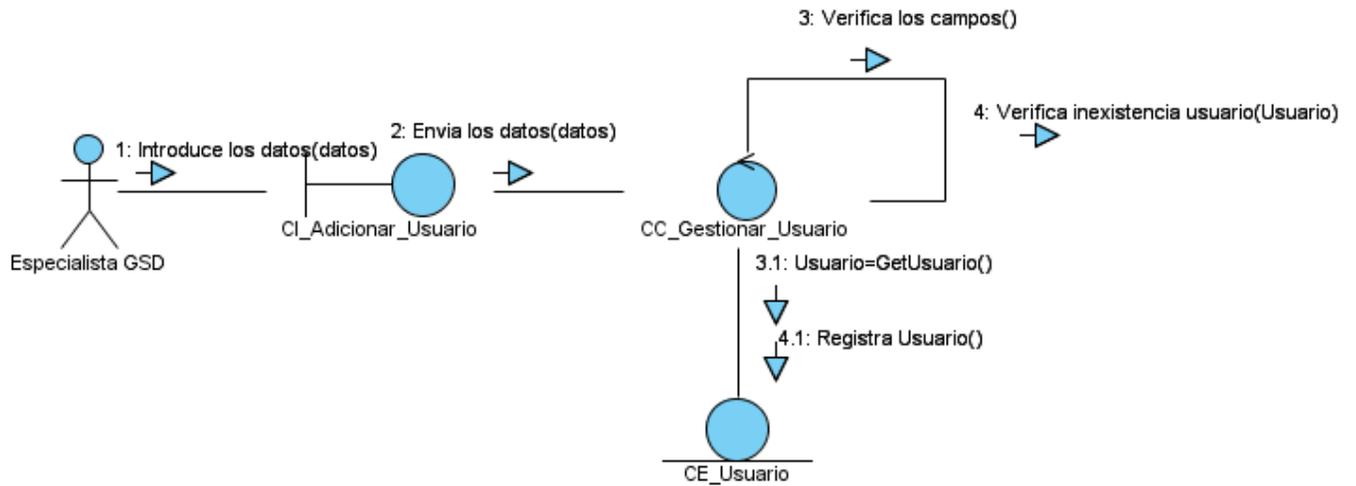


Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Modificar.

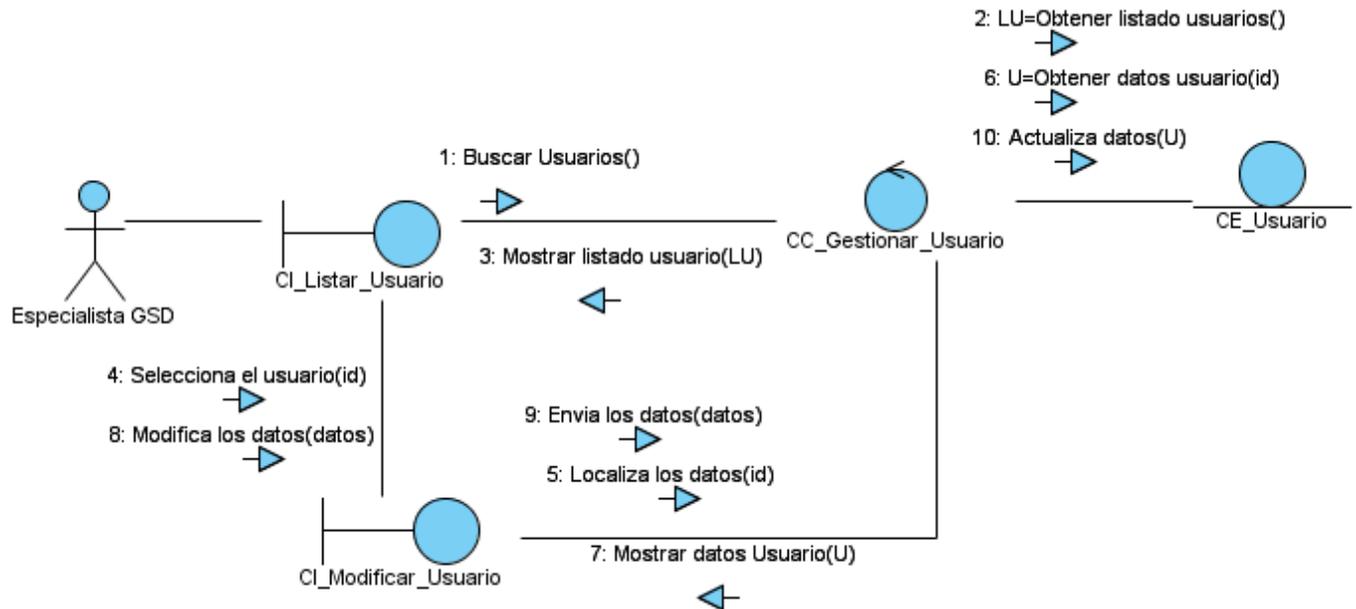


Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Buscar.

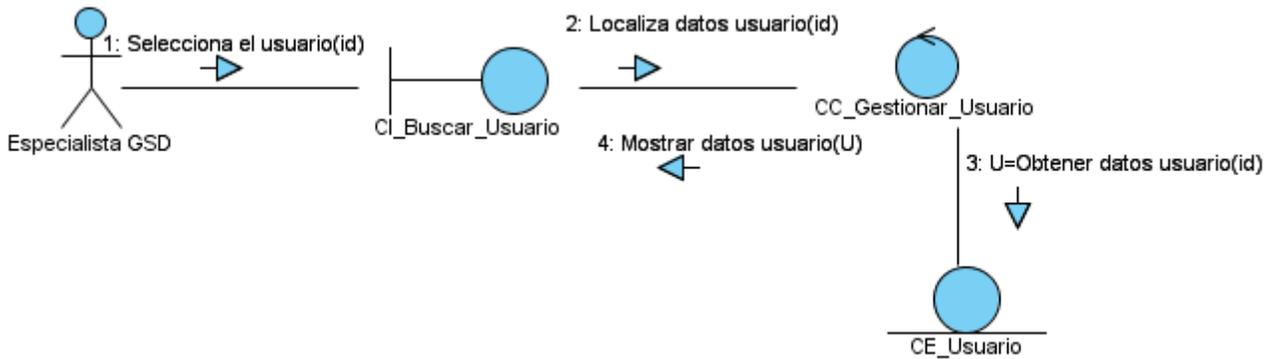


Diagrama de Colaboración del Caso de Uso: Gestionar Usuario. Sección Eliminar.

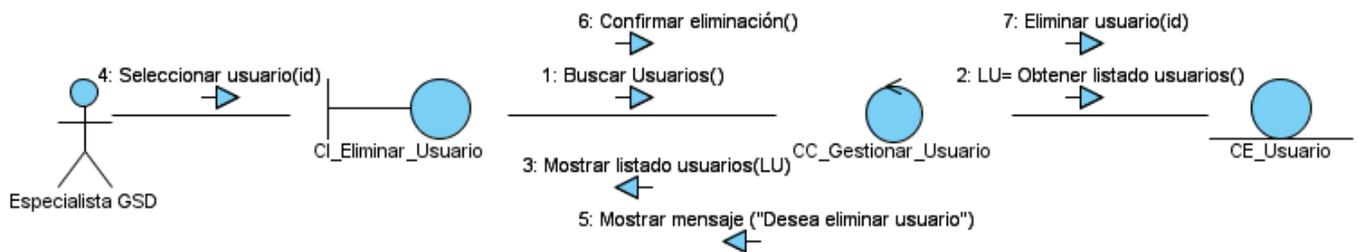


Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Adicionar.

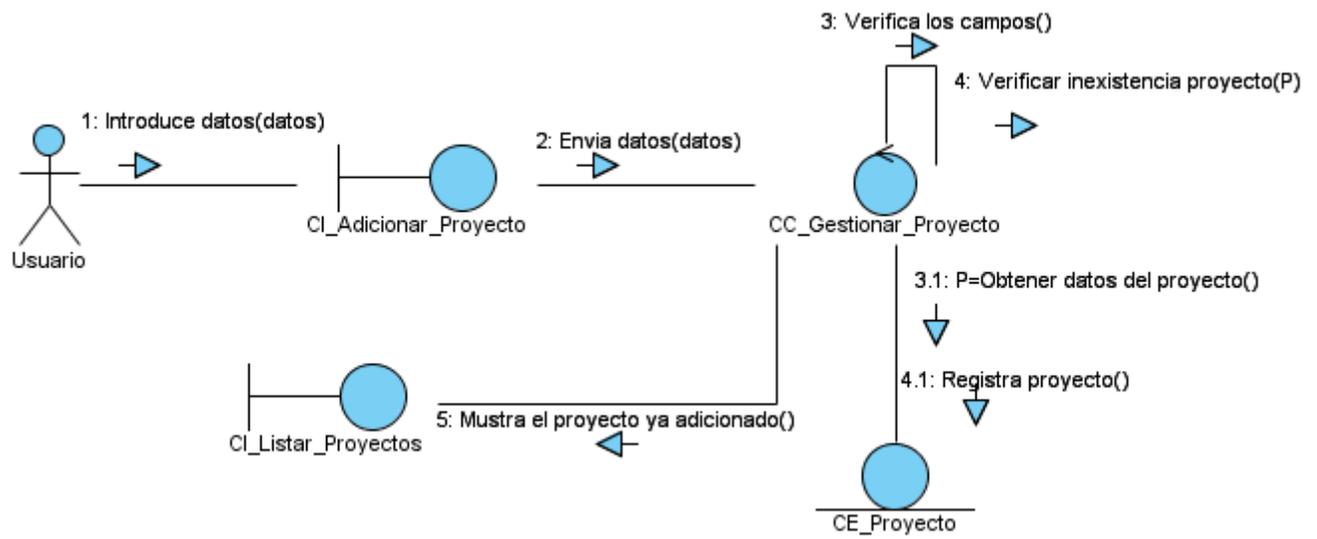


Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Modificar.

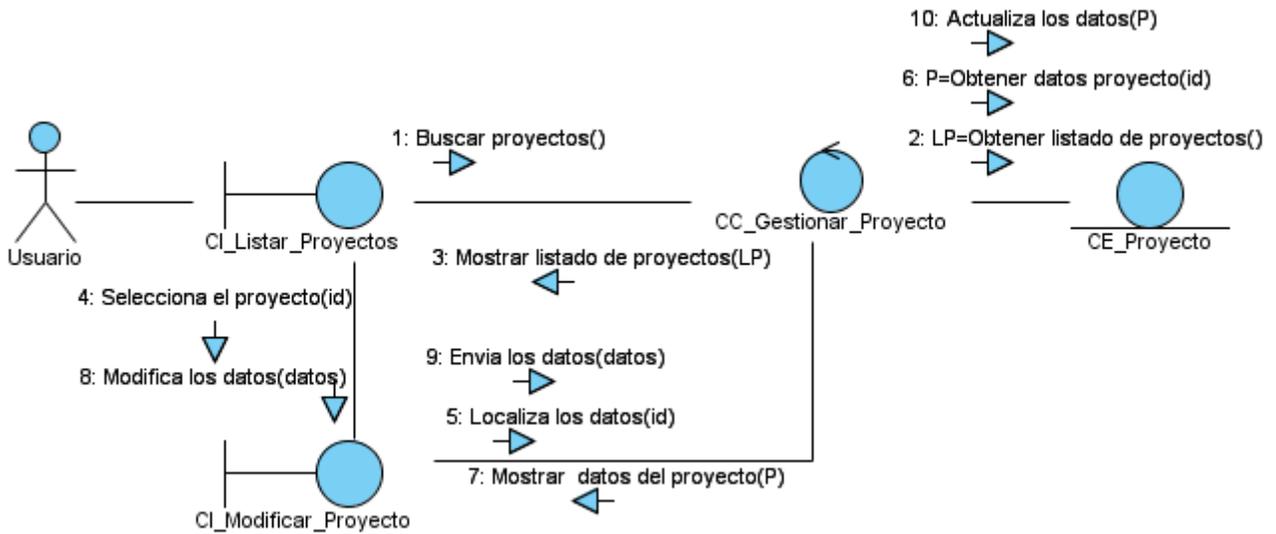


Diagrama de Colaboración del Caso de Uso: Gestionar Proyectos. Sección Mostrar.

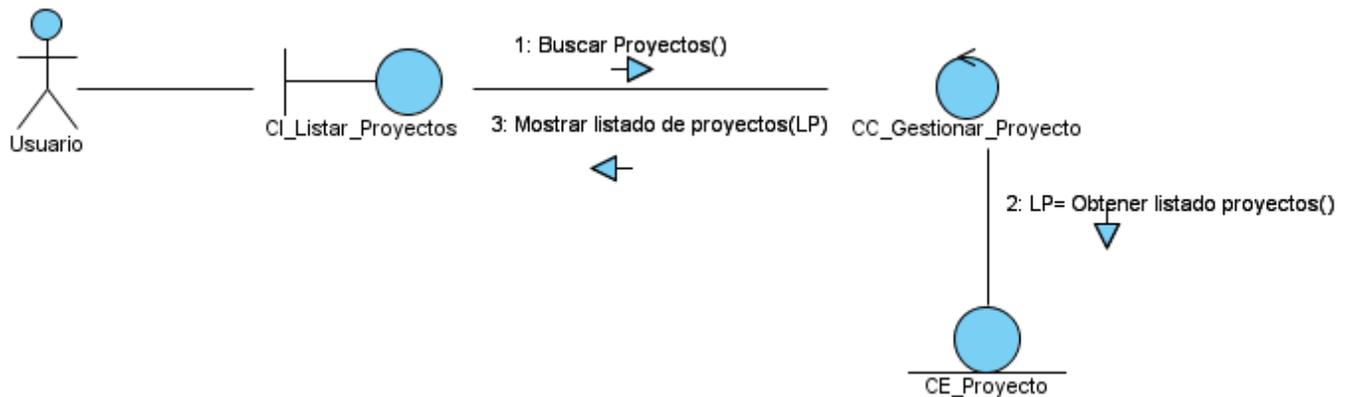


Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Adicionar.

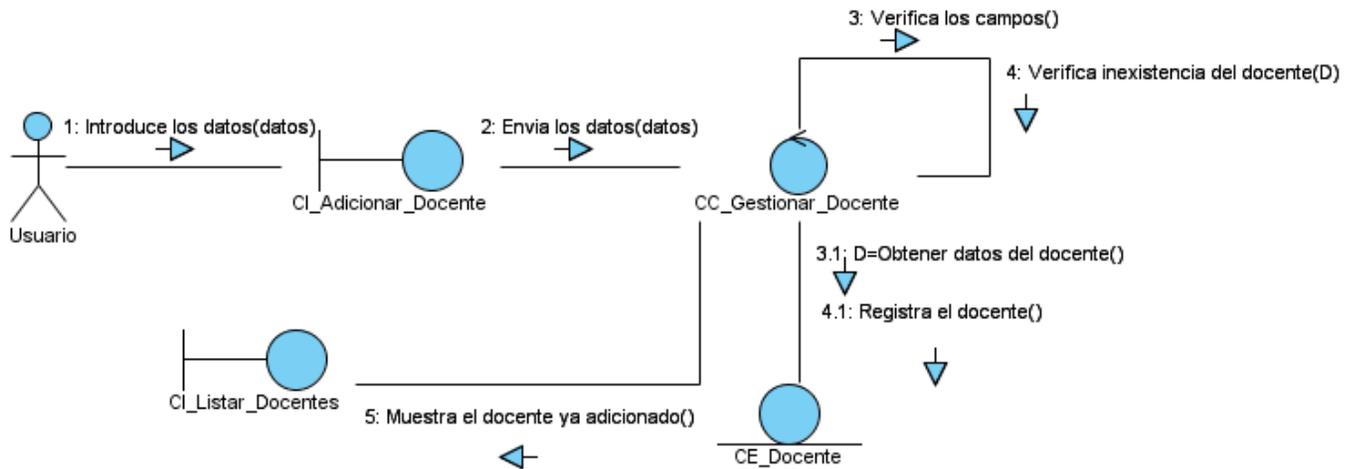


Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Modificar.

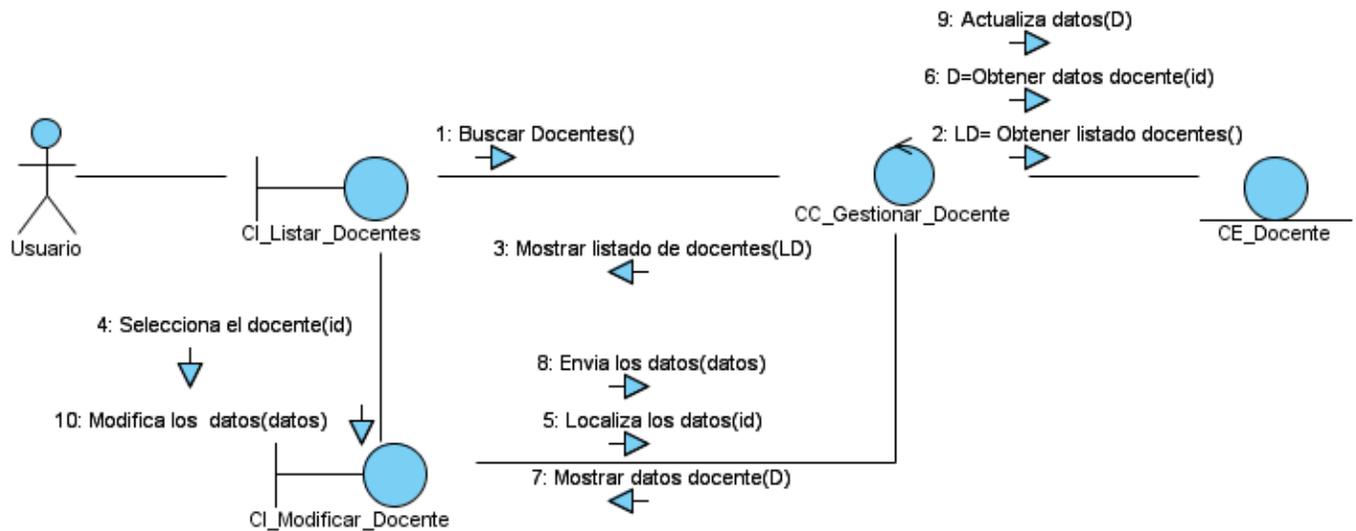


Diagrama de Colaboración del Caso de Uso: Gestionar Docentes. Sección Mostrar.

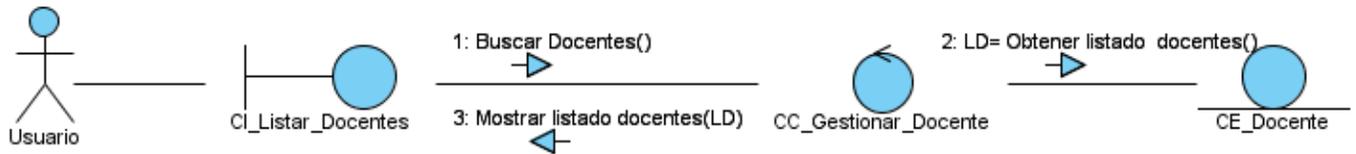


Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Adicionar.

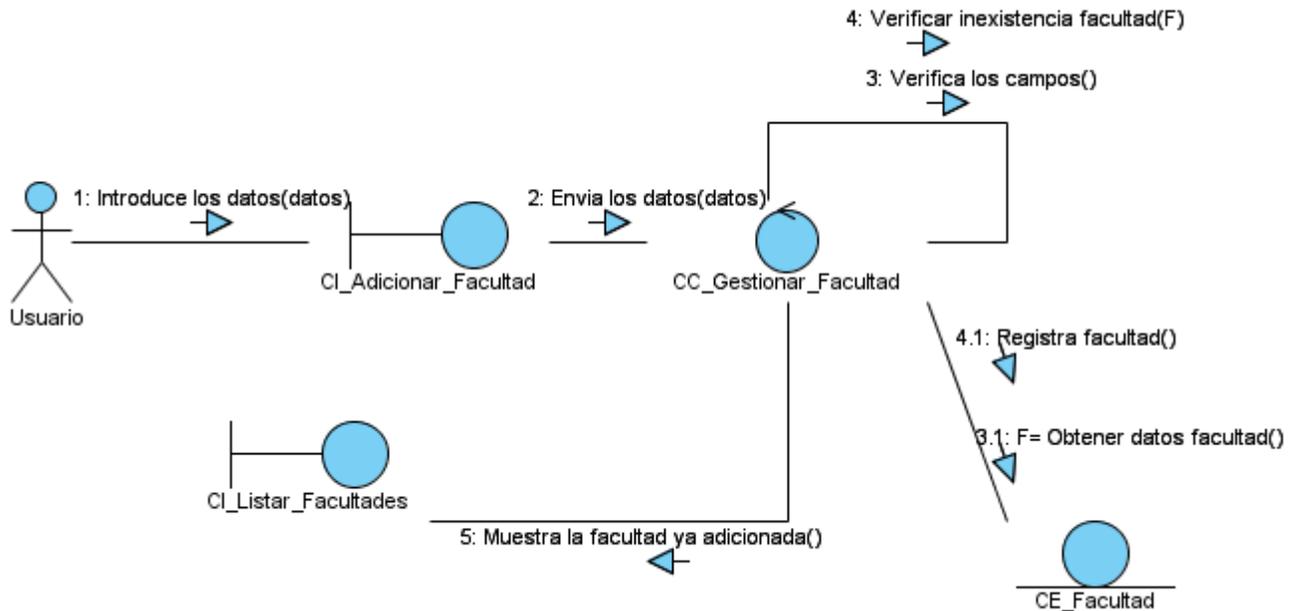


Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Modificar.

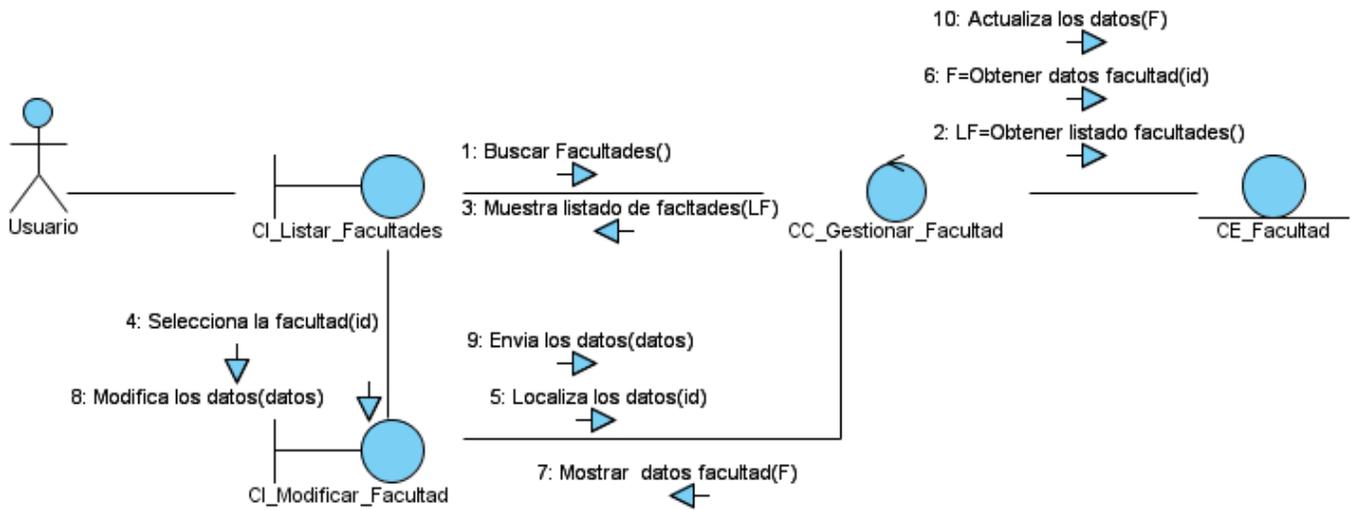


Diagrama de Colaboración del Caso de Uso: Gestionar Facultades. Sección Mostrar.

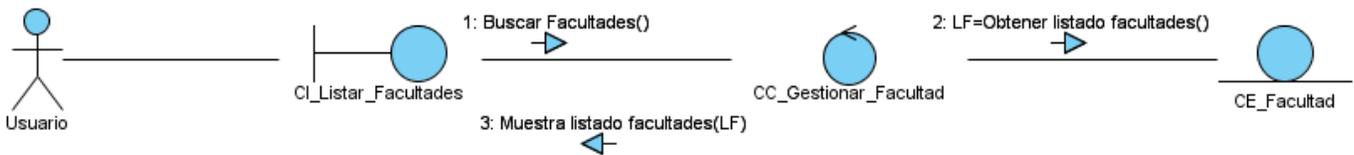


Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Adicionar.

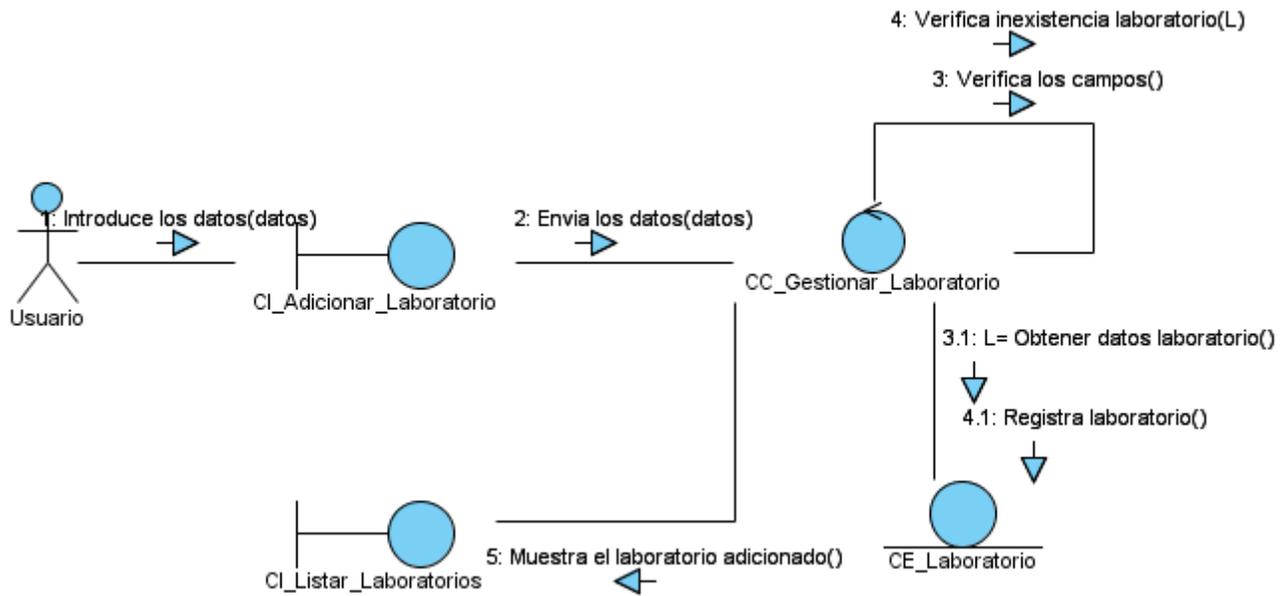


Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Modificar.

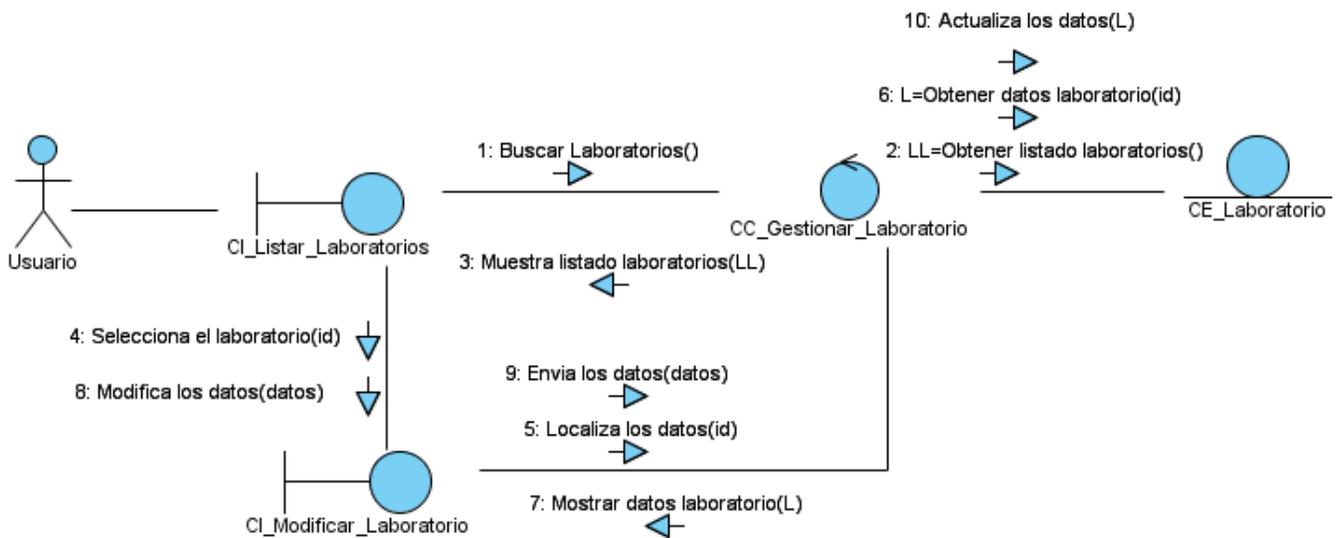


Diagrama de Colaboración del Caso de Uso: Gestionar Laboratorios. Sección Mostrar.

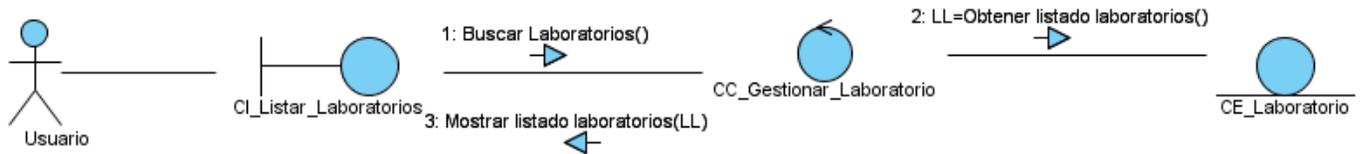


Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Adicionar.

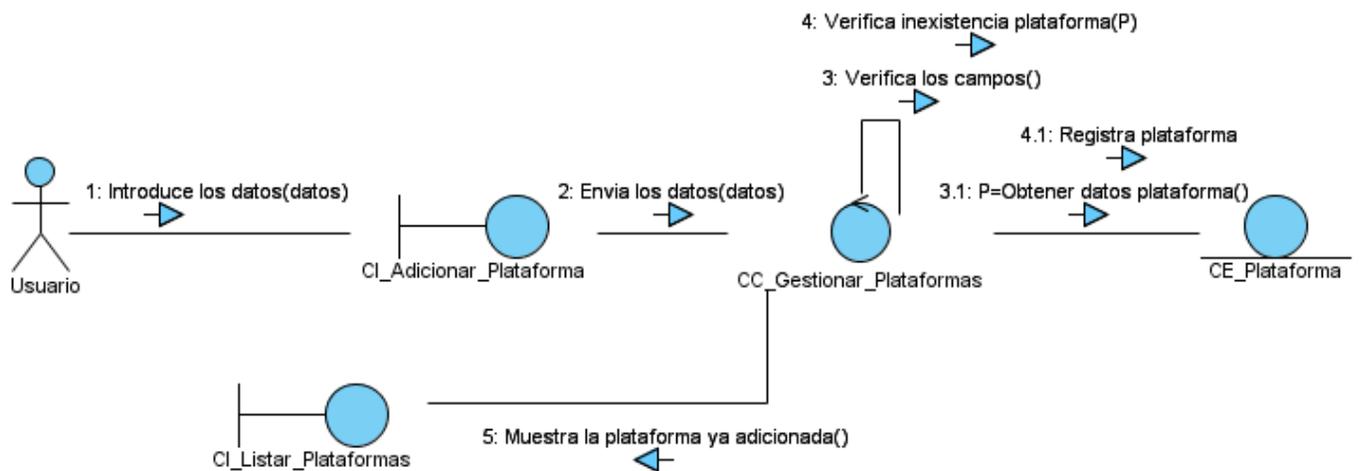


Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Modificar.

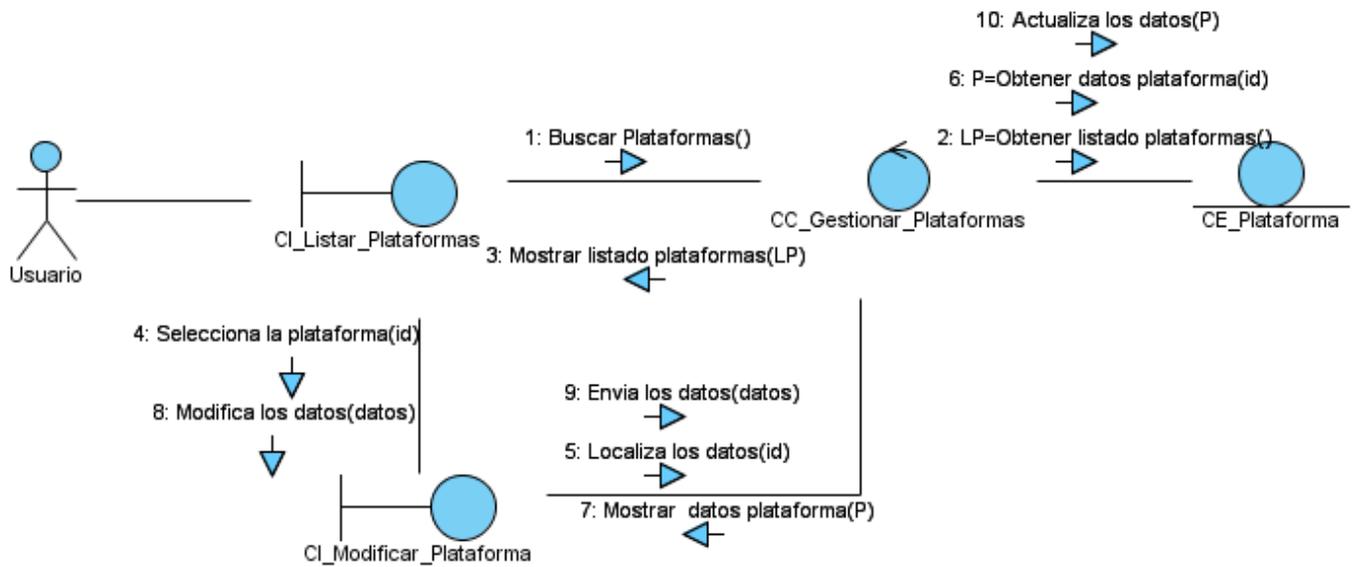


Diagrama de Colaboración del Caso de Uso: Gestionar Plataforma. Sección Mostrar.

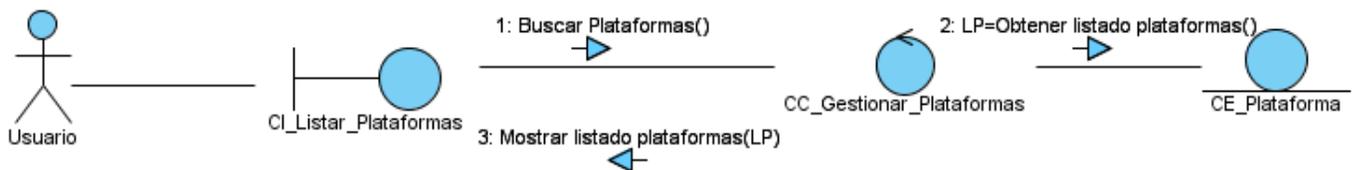


Diagrama de Colaboración del Caso de Uso: Eliminar Proyecto.

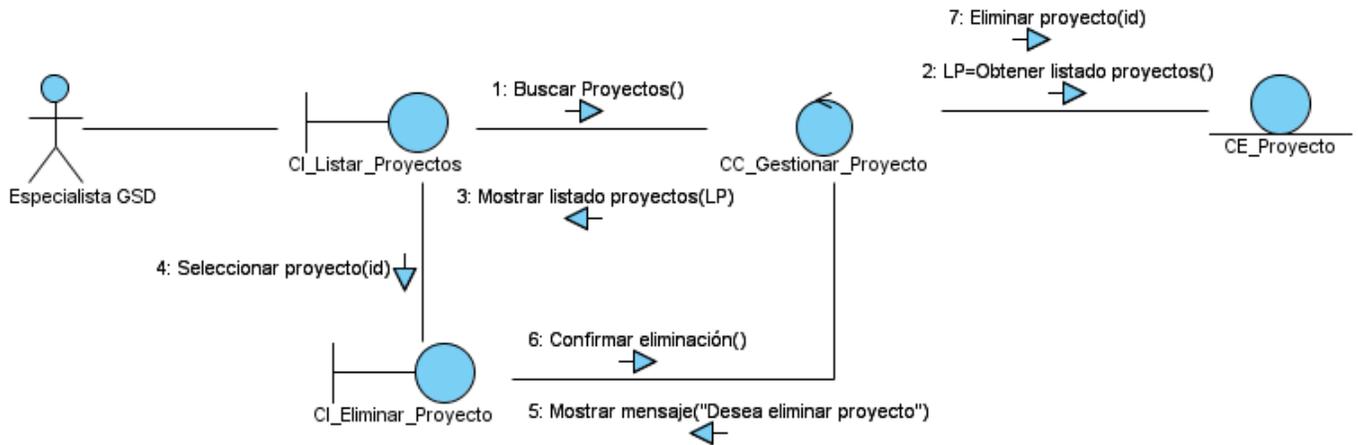


Diagrama de Colaboración del Caso de Uso: Eliminar Docente.

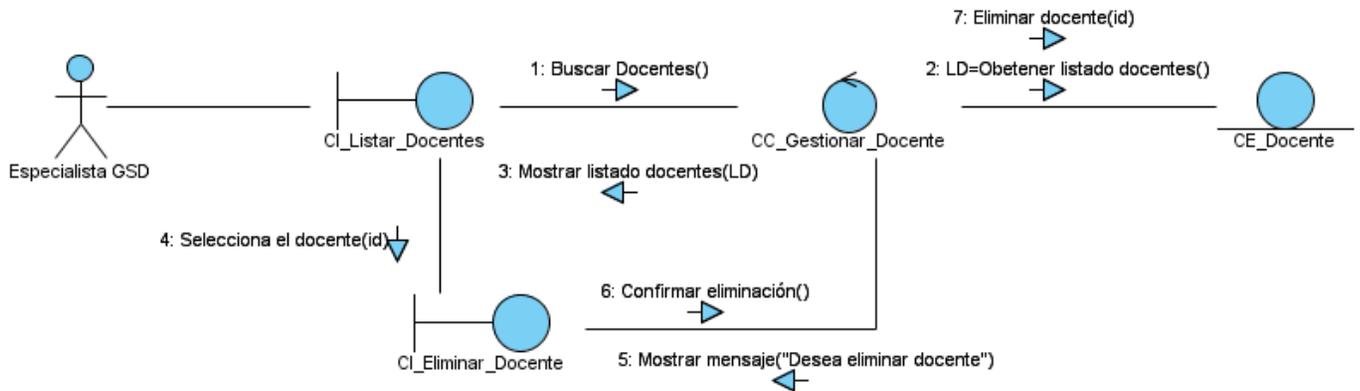


Diagrama de Colaboración del Caso de Uso: Eliminar Facultad.

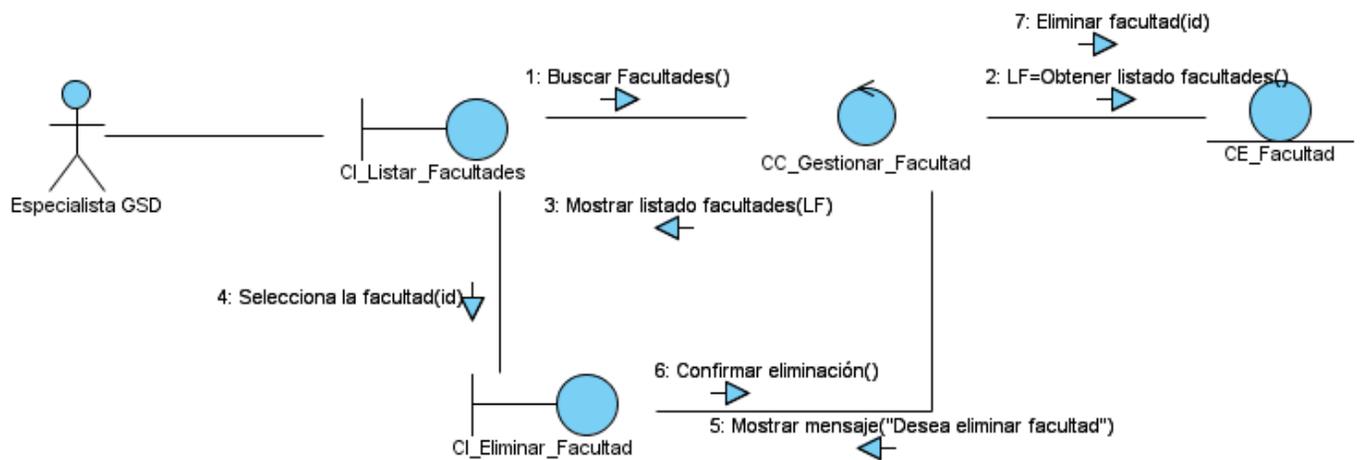


Diagrama de Colaboración del Caso de Uso: Eliminar Laboratorio.

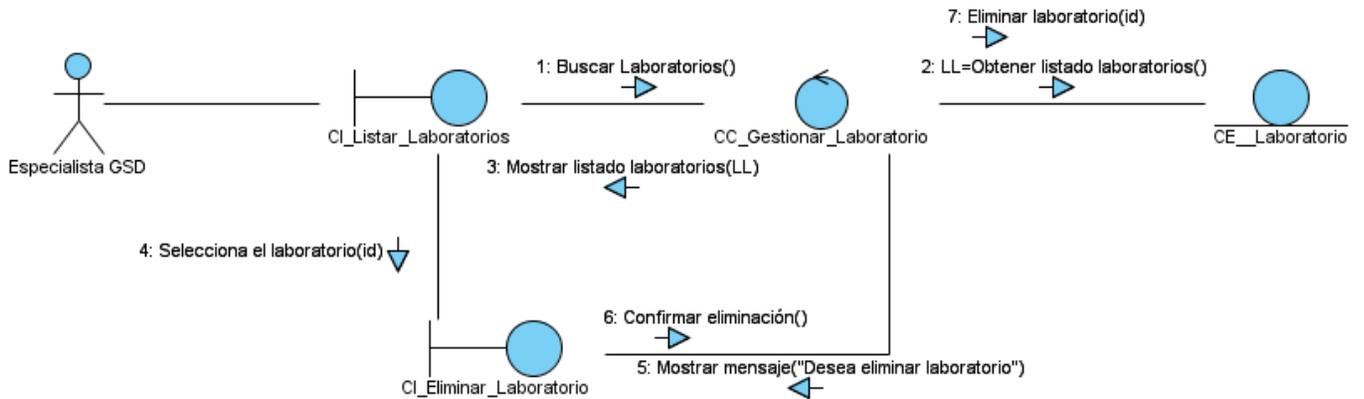


Diagrama de Colaboración del Caso de Uso: Eliminar Registro de Salva.

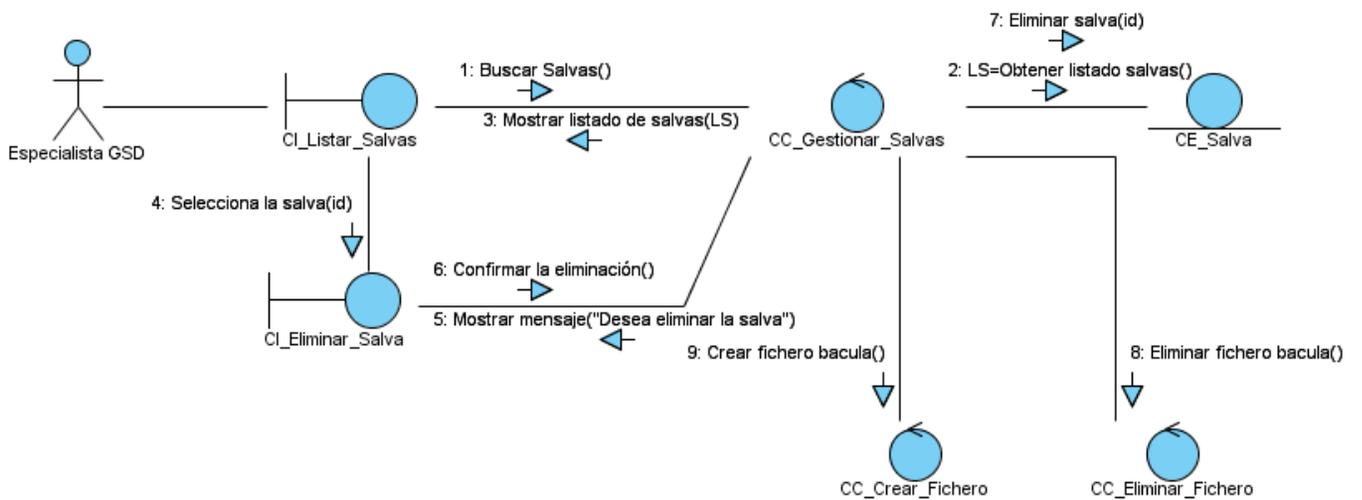


Diagrama de Colaboración del Caso de Uso: Eliminar Plataforma.

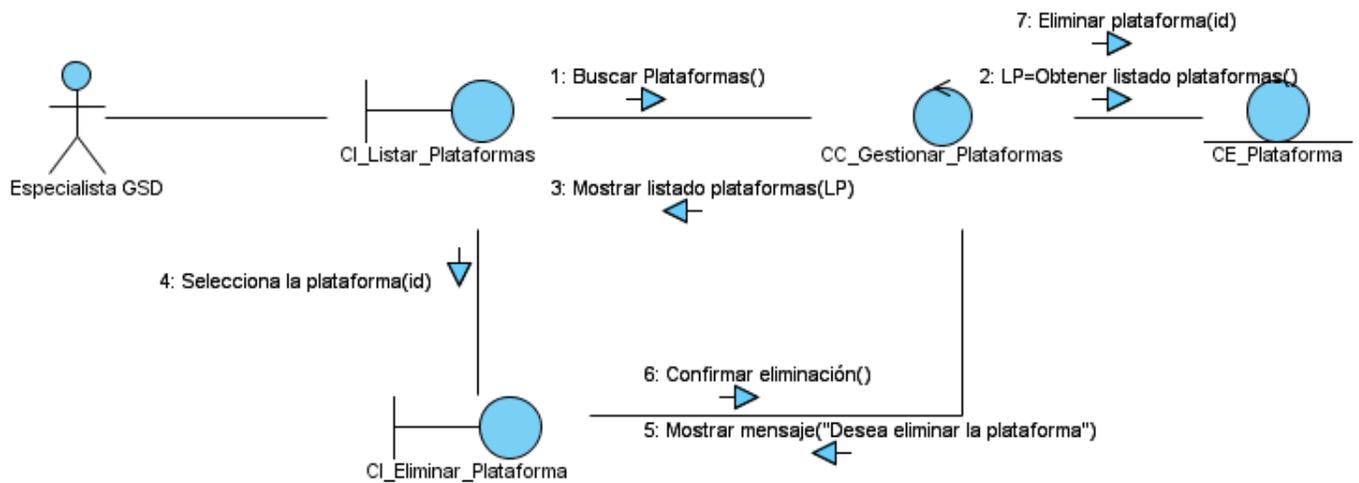


Diagrama de Colaboración del Caso de Uso: Solicitar Registro de Salva.

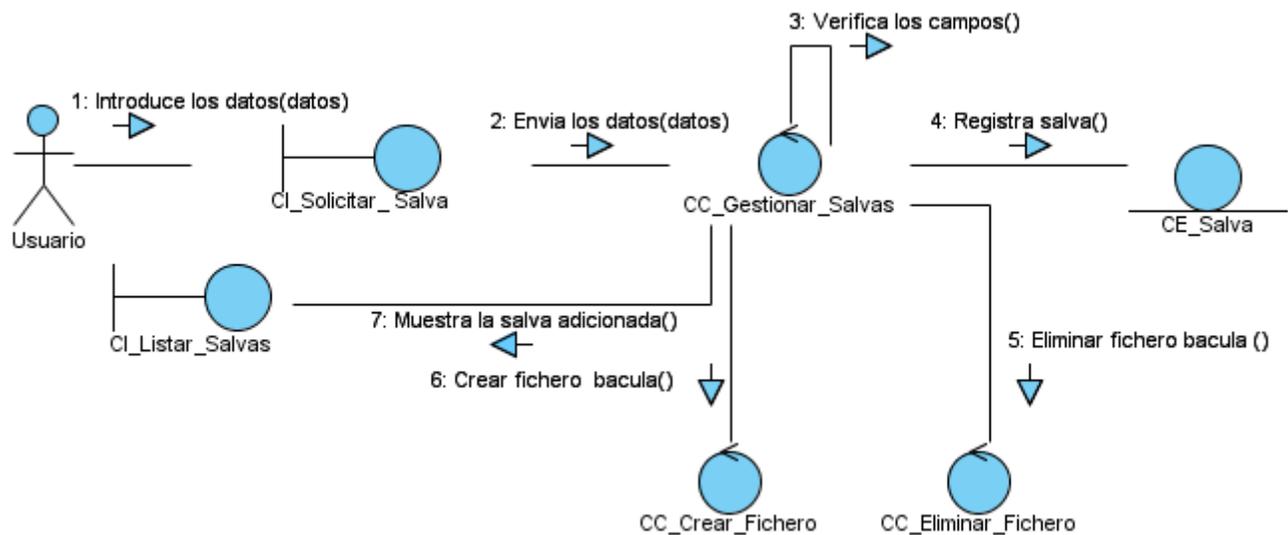


Diagrama de Colaboración del Caso de Uso: Modificar Registro de Salva.

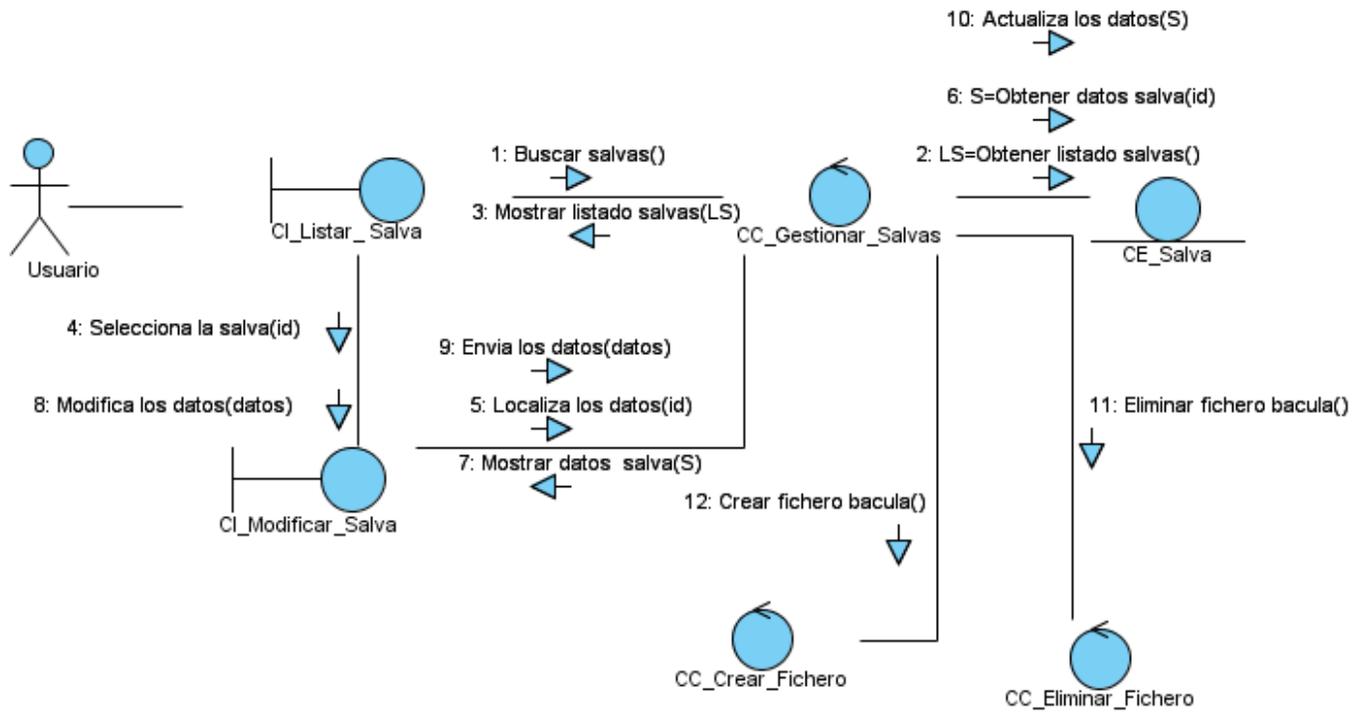


Diagrama de Colaboración del Caso de Uso: Reporte de Registro de Salva.

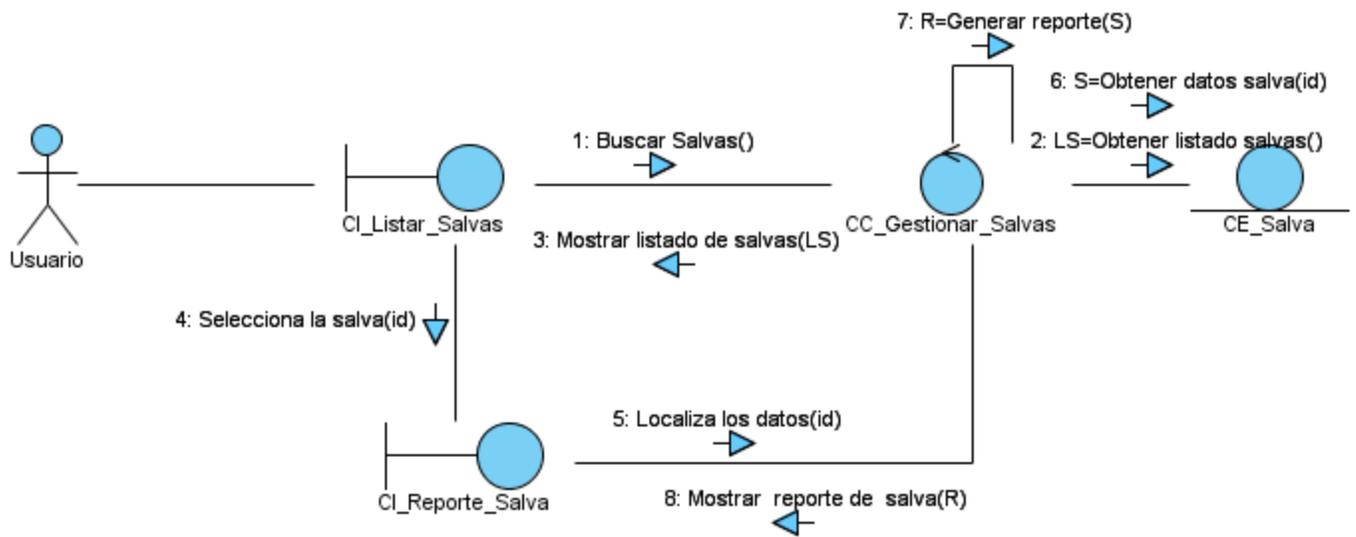


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Usuario. Sección Buscar.

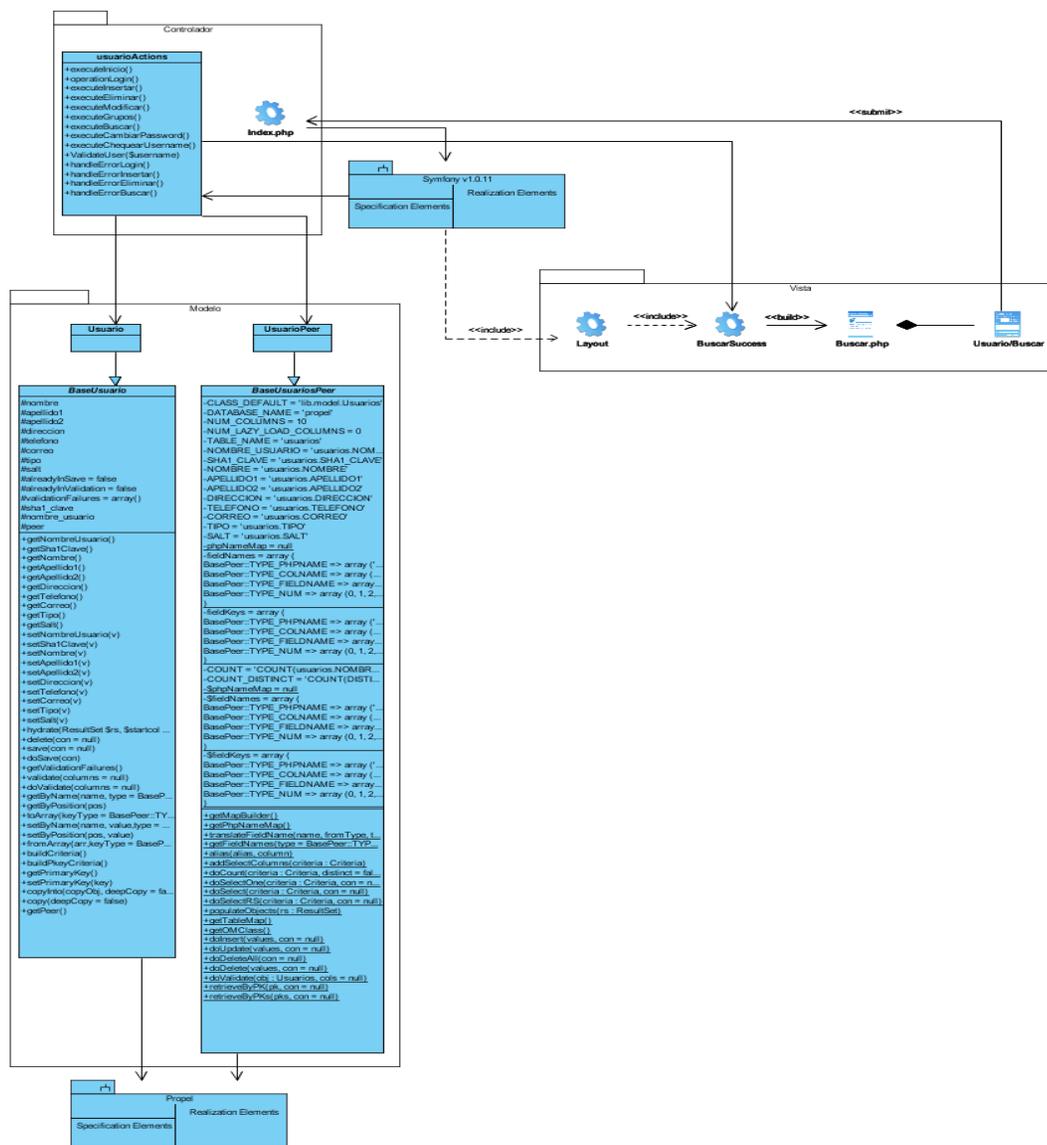


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto. Sección Adicionar.

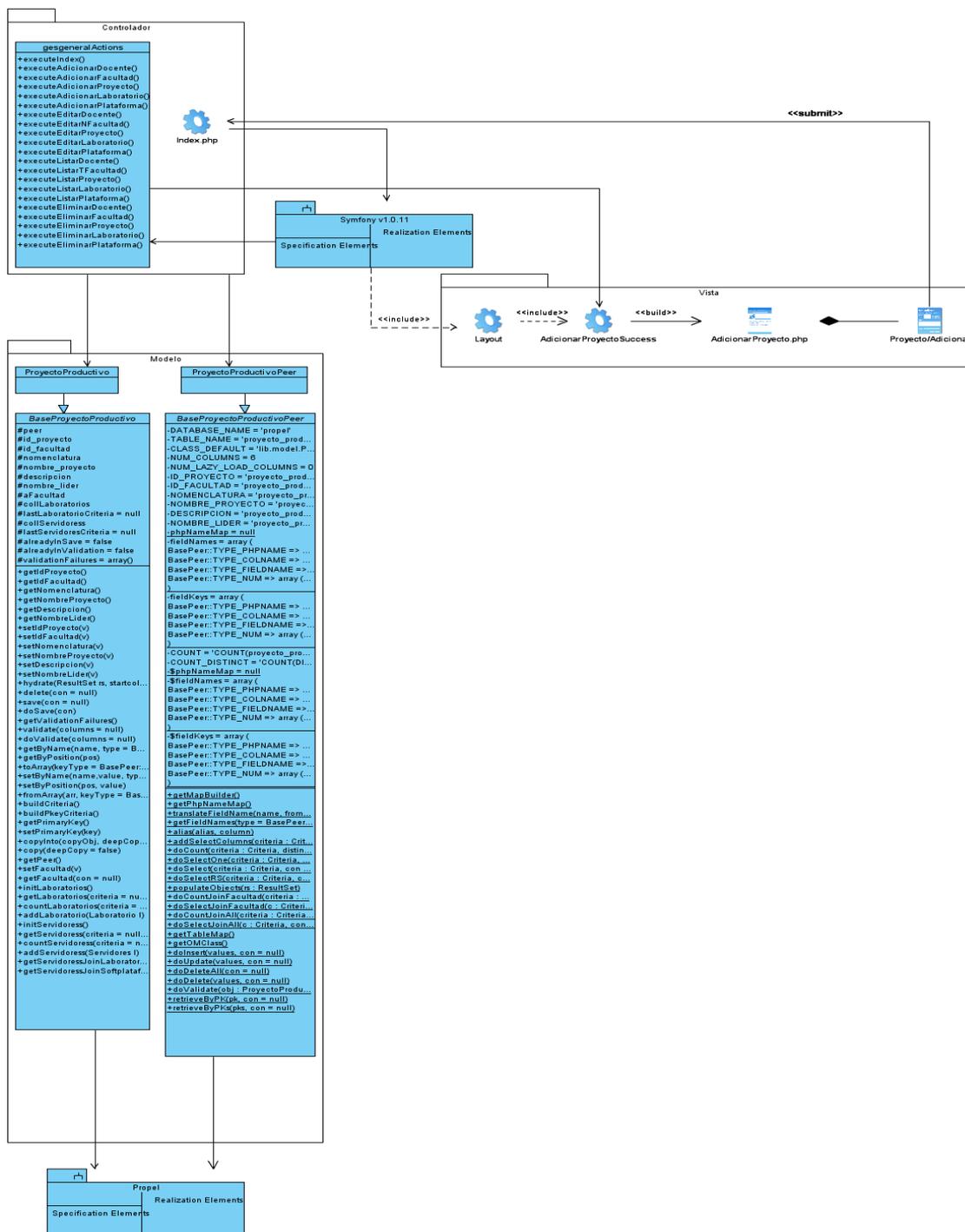


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Proyecto. Sección Modificar.

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

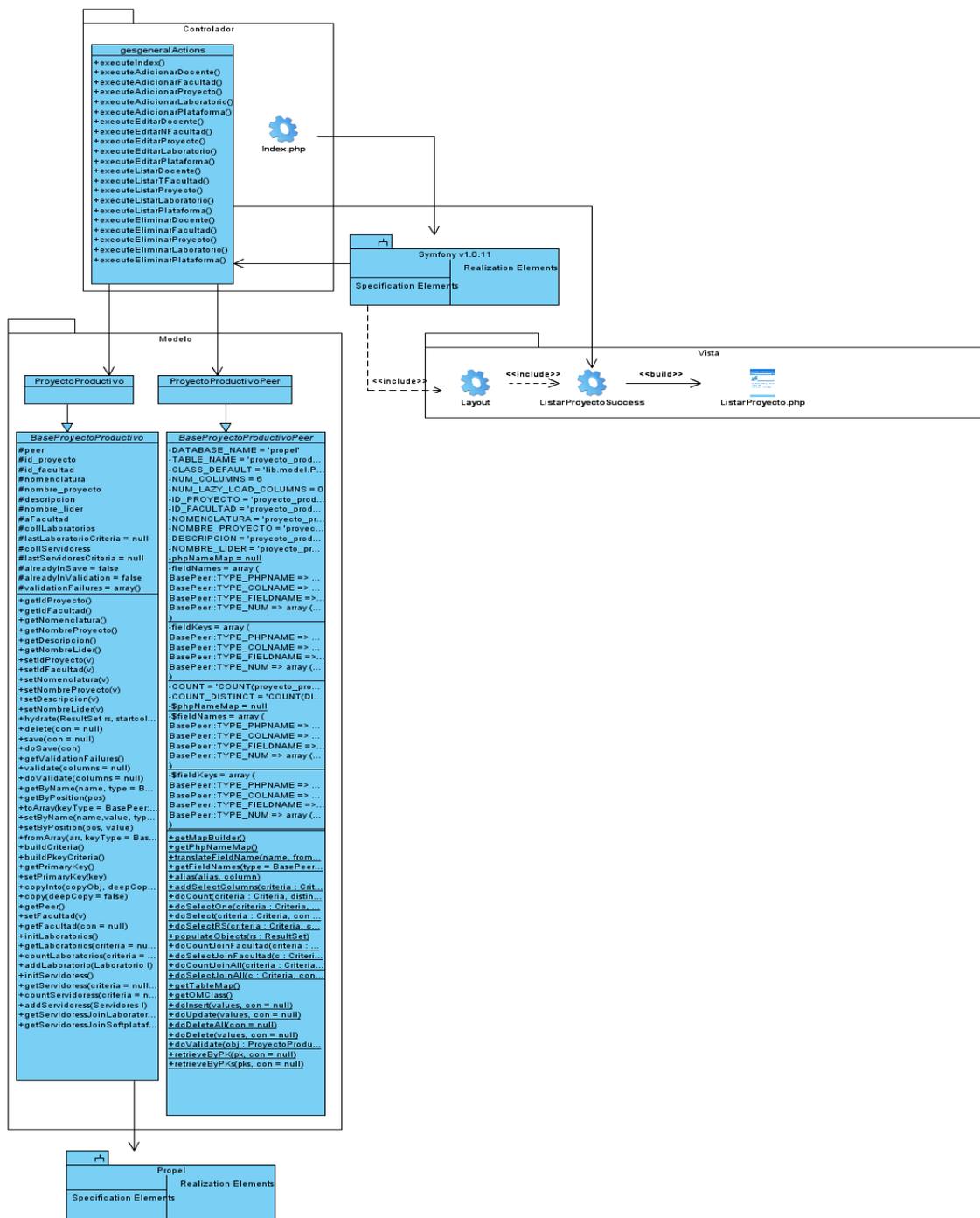


Diagrama de Clases del Diseño del Caso de Uso: Eliminar Proyecto.

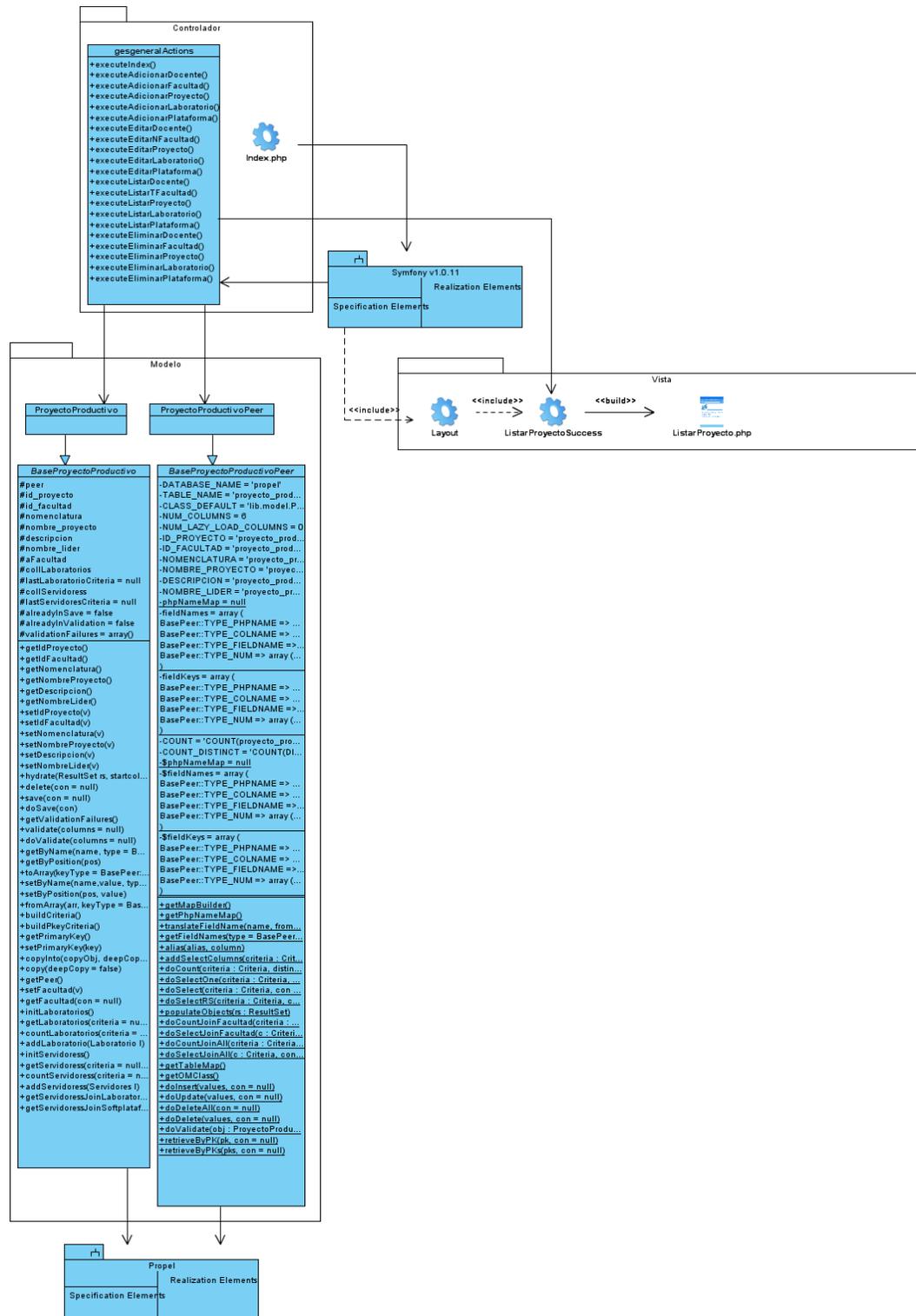


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Docente. Sección Mostrar.

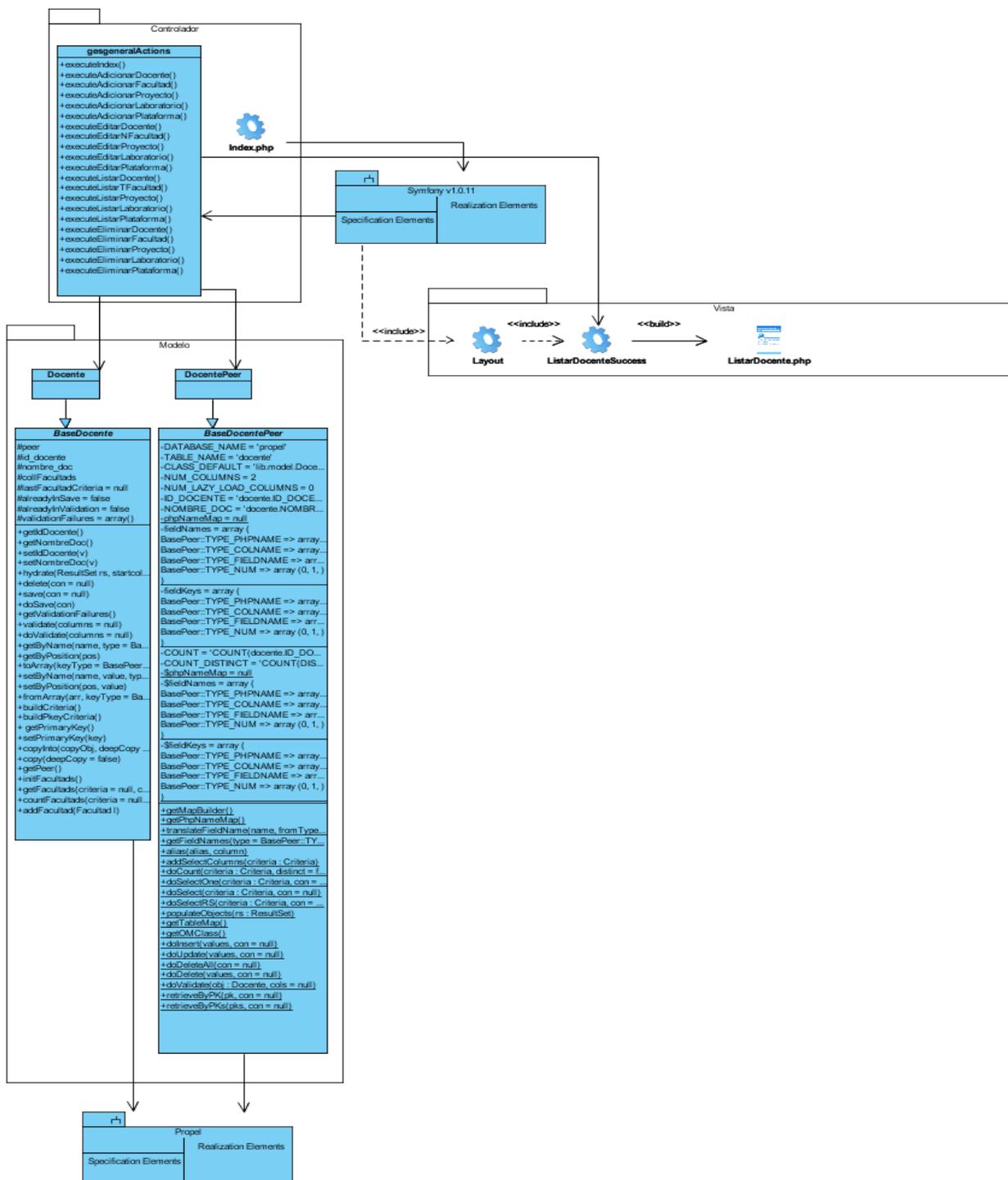


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Laboratorio. Sección Adicionar.

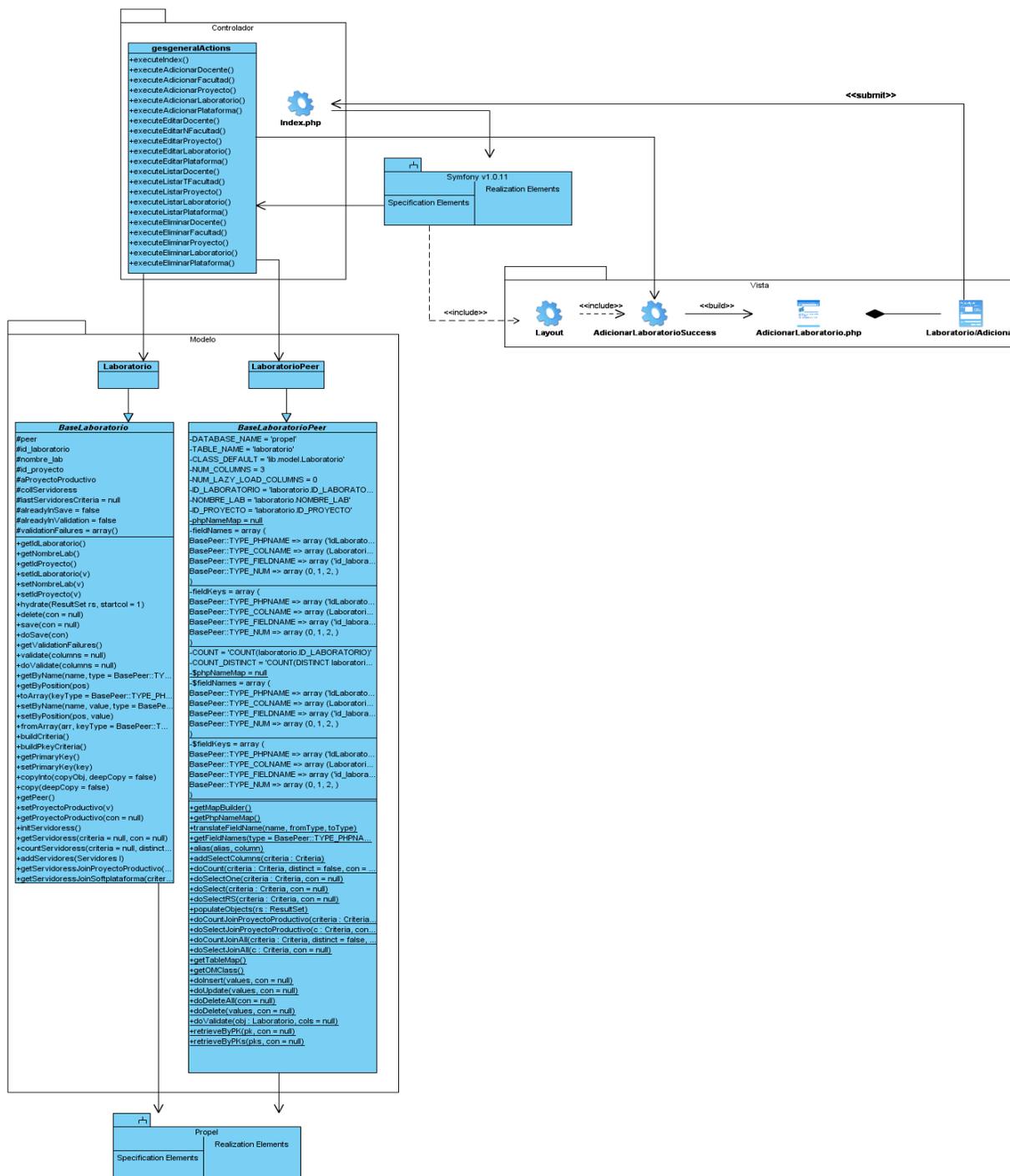


Diagrama de Clases del Diseño del Caso de Uso: Eliminar Laboratorio.

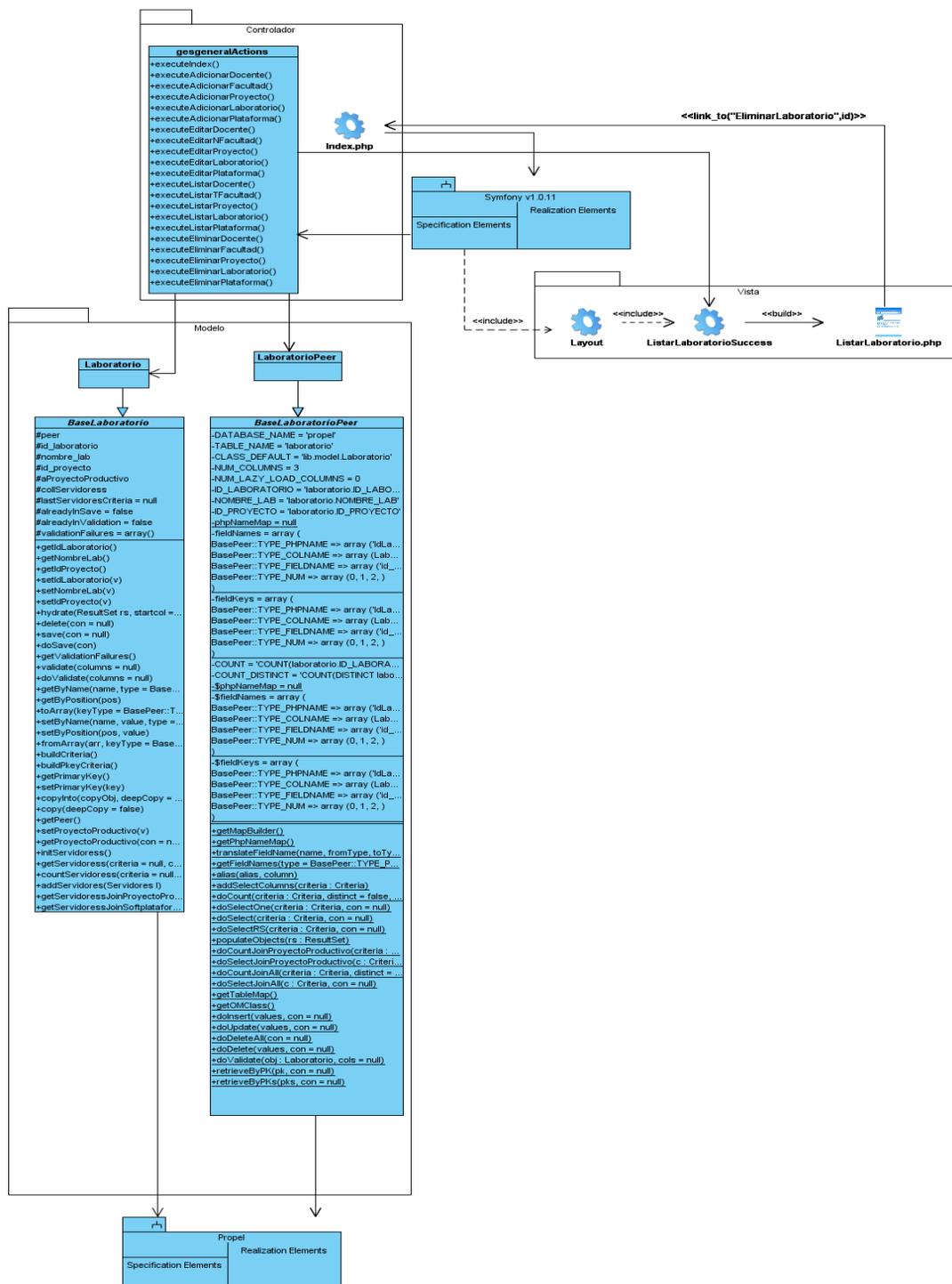


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Adicionar.

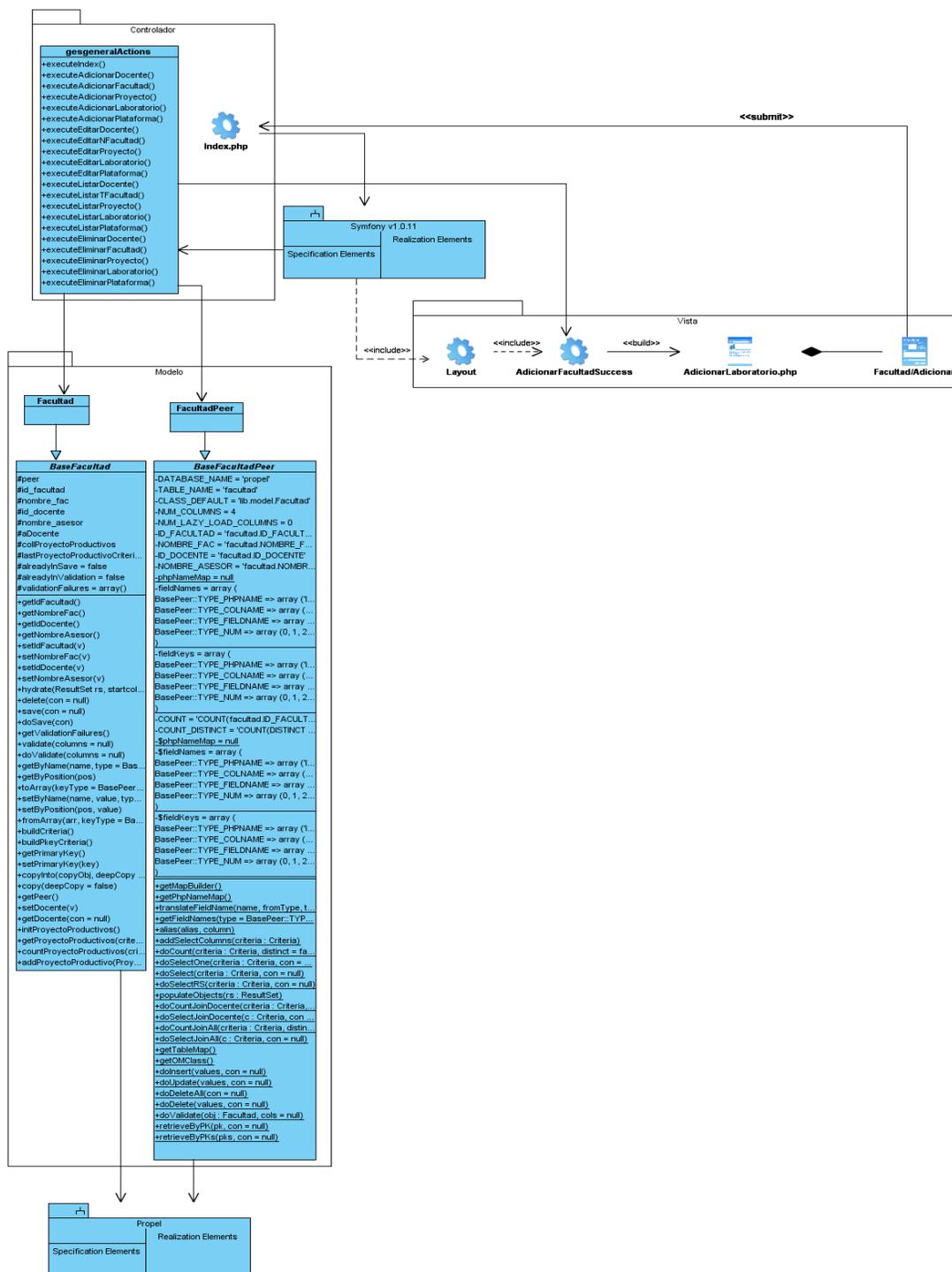


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Facultad. Sección Modificar.

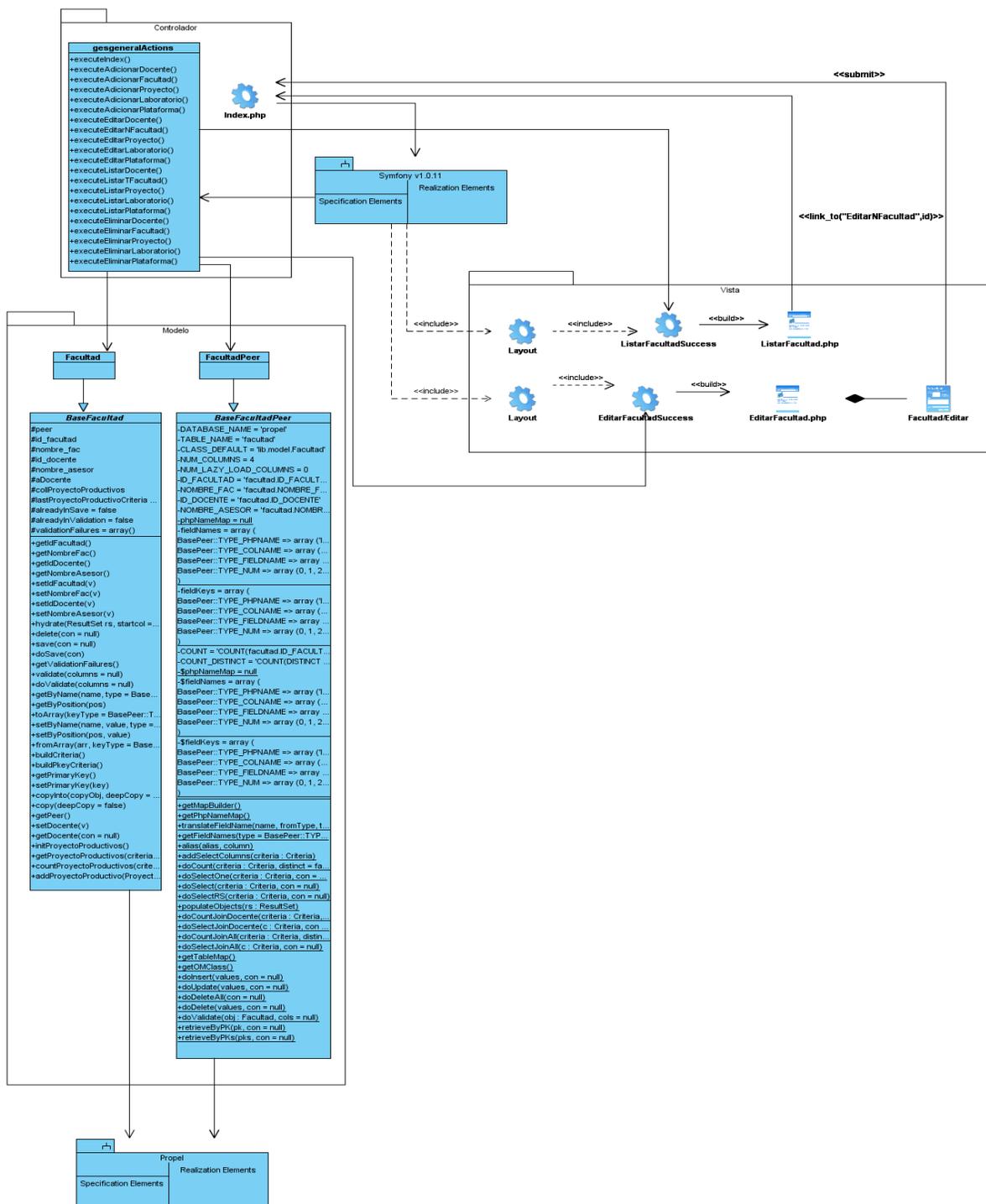


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Plataforma. Sección Adicionar.

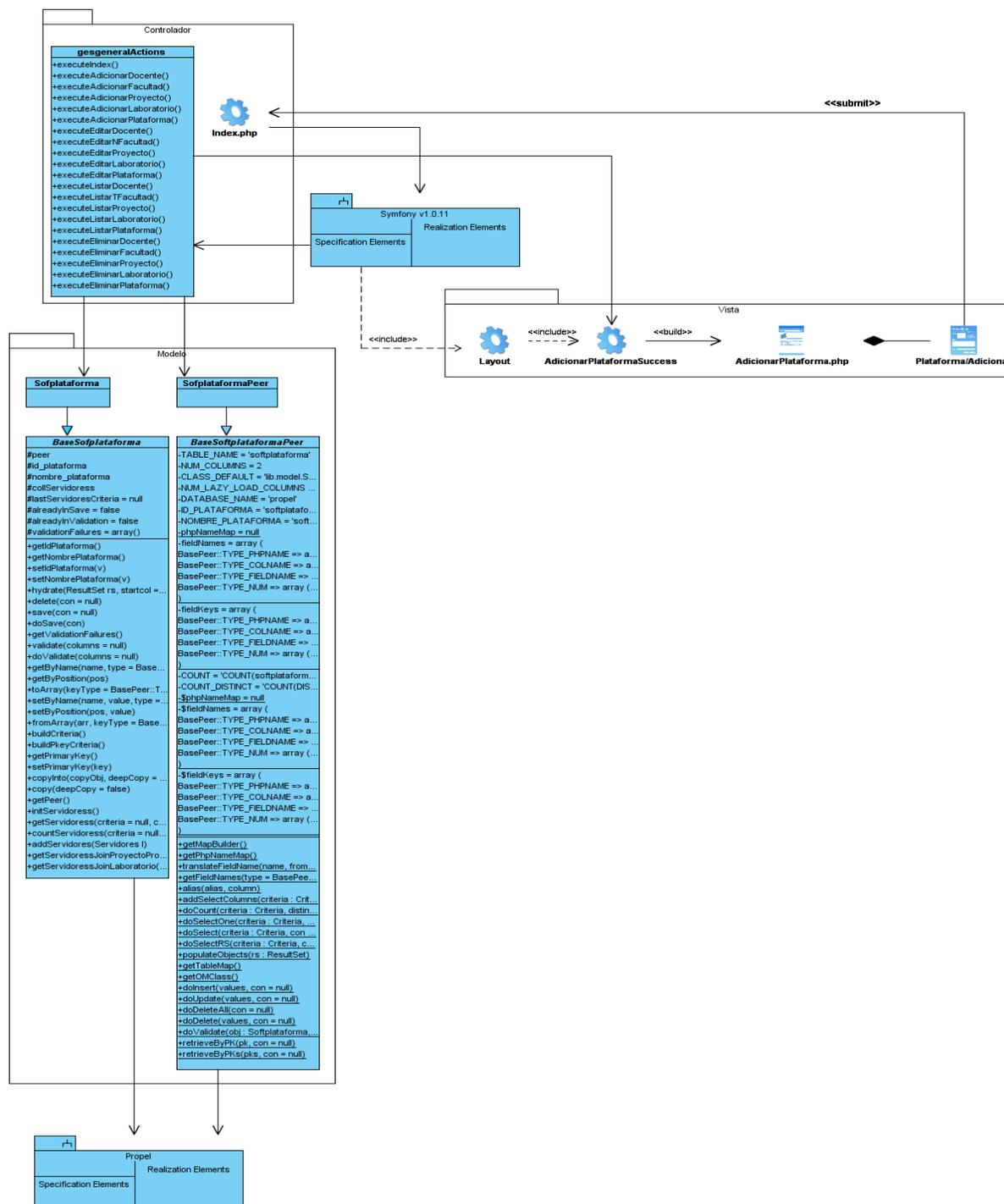


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Plataforma. Sección Modificar.

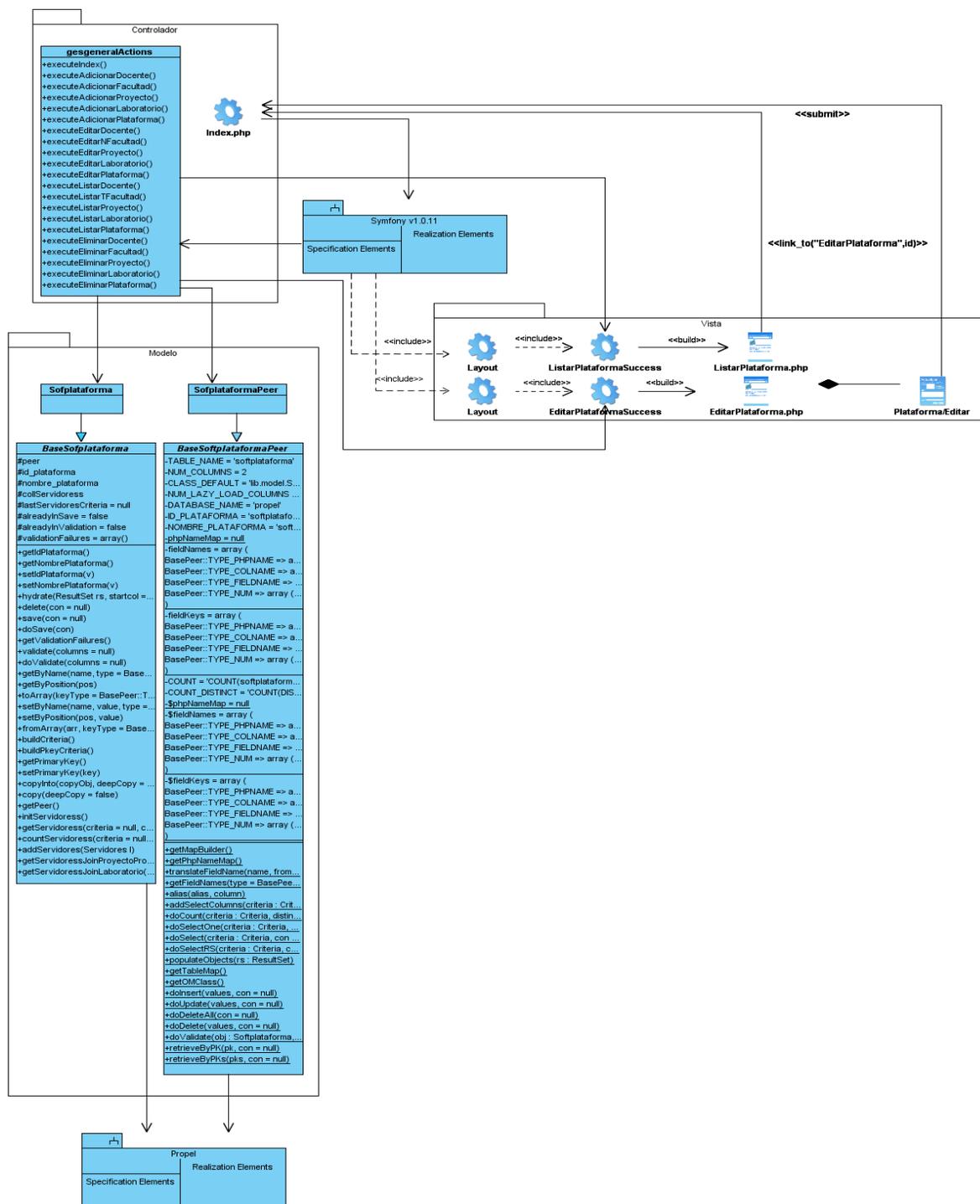


Diagrama de Clases del Diseño del Caso de Uso: Gestionar Plataforma. Sección Mostrar.

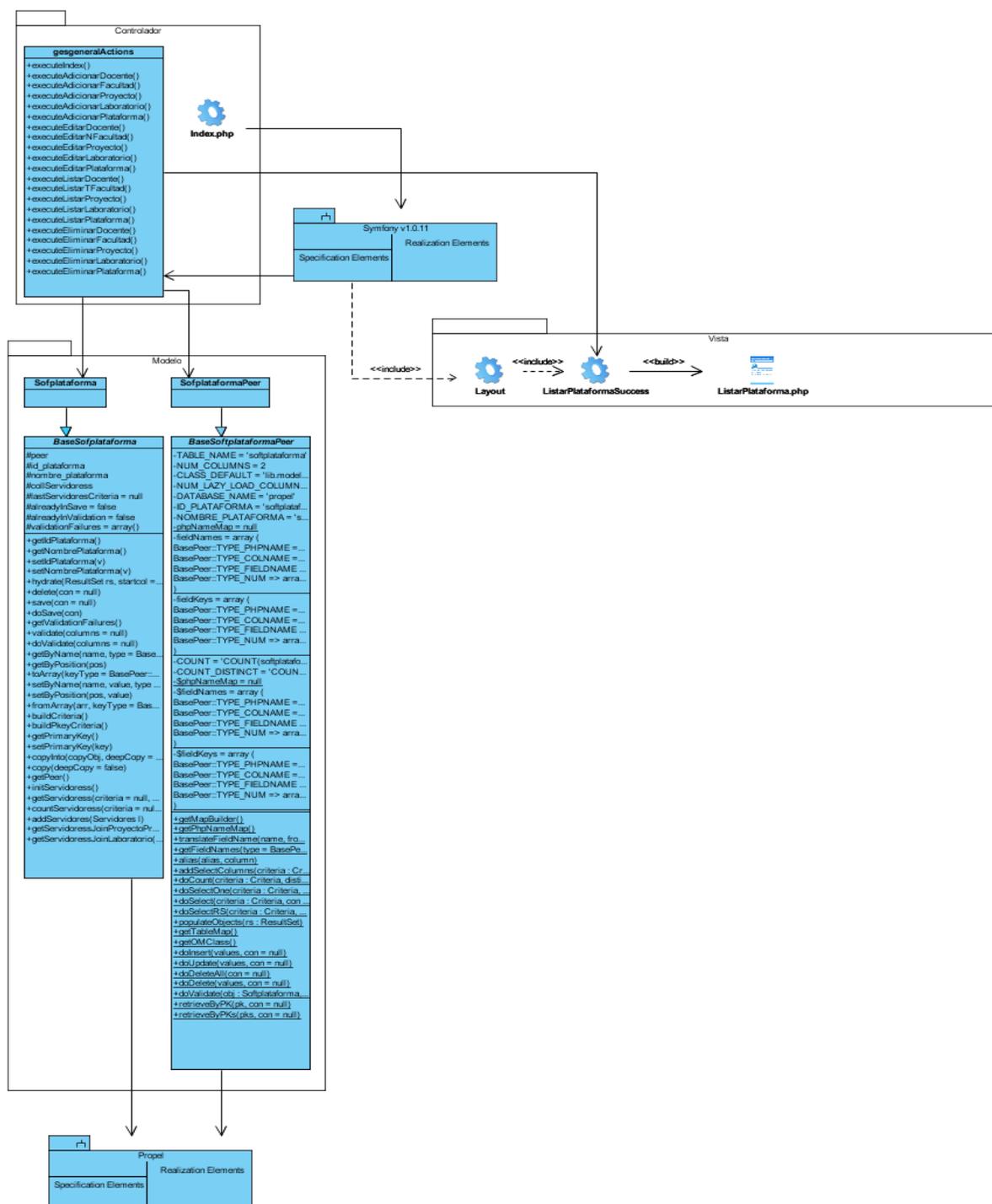
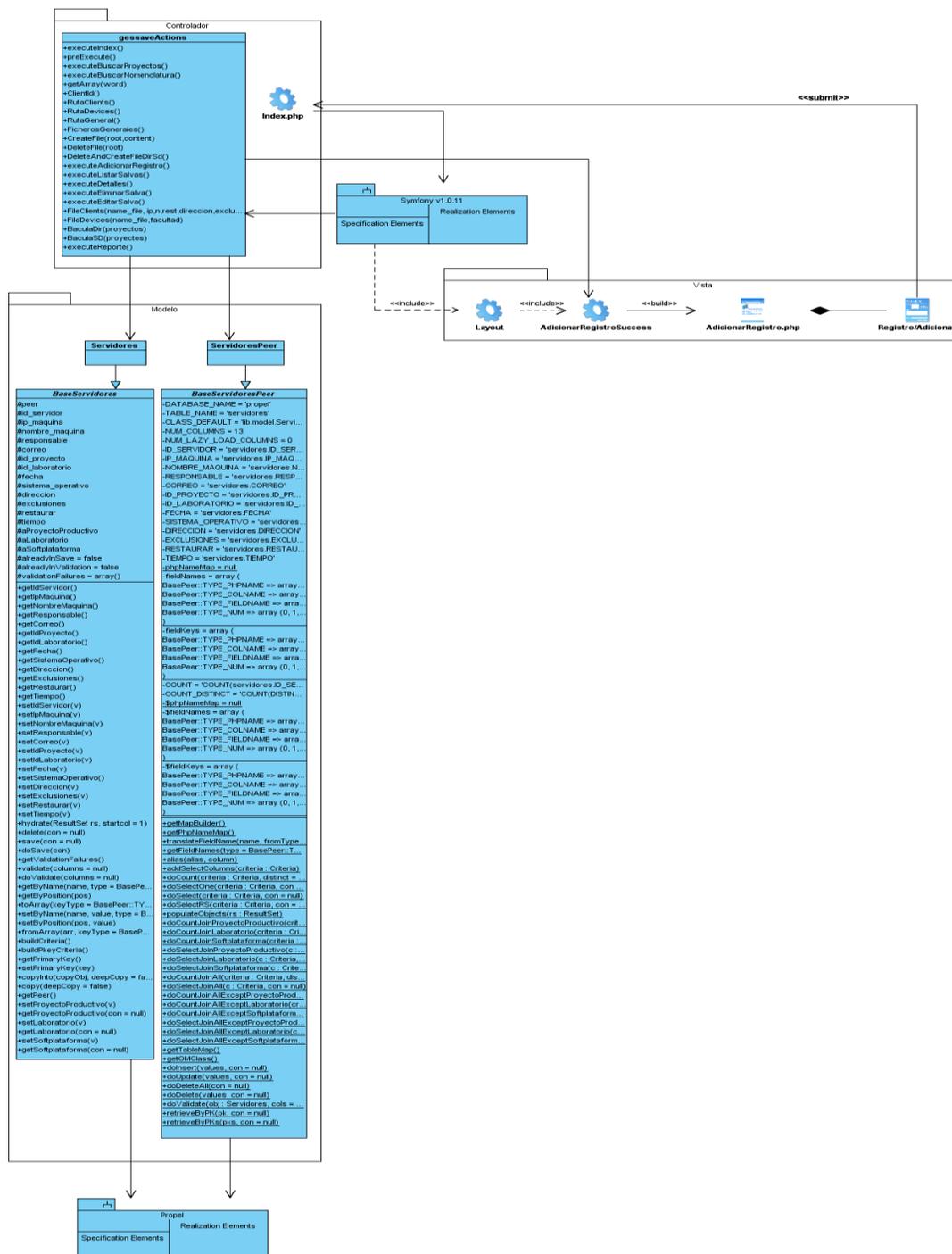


Diagrama de Clases del Diseño del Caso de Uso: Solicitar Registro de Salva.



GLOSARIO DE TÉRMINOS

RUP: Proceso Unificado de Desarrollo.

UML: Lenguaje Unificado de Modelado.

RF: Requisitos Funcionales.

RNF: Requisitos no Funcionales.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos, asociación estadounidense dedicada a la estandarización. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros de telecomunicaciones, ingenieros electrónicos, ingenieros en informática y otros (***The Institute of Electrical and Electronics Engineers***).

CUN: Caso de Uso del Negocio.

CUS: Caso de Uso del Sistema.

HTTP: Protocolo para transferir archivos o documentos hipertexto a través de la red. (Hyper Text Transmisión protocol).

UNIX: Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

PHP: Es acrónimo de Hypertext Pre-processor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación (Personal Home Page).

MySQL: Sistema de administración de base de datos, es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

COBOL: Lenguaje común Orientado a Negocios (COmmon Business- Oriented Language).

Basic: Es una familia de lenguajes de programación.

Pascal: Lenguaje de programación estructurado de alto nivel y propósito general, es un lenguaje orientado a procedimientos con usos varios, desarrollado por Niklaus Wirth hace ya más de 20 años; profesor del Instituto tecnológico de Zurich, Suiza.

C++: Es un lenguaje híbrido, que se puede compilar y resulta más sencillo de aprender para los programadores que ya conocen C. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Las principales características son abstracción (encapsulación), el soporte para programación orientada a objetos (polimorfismo) y el soporte de plantillas o programación genérica (templates). Es un lenguaje que abarca tres paradigmas de la

programación: La programación estructurada, la programación genérica y la programación orientada a objetos.

Java: Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

C#: Lenguaje de programación orientado a objetos, evolución del lenguaje C++, desarrollado por Microsoft.

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

HTML: Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el estándar de las páginas Web. (Hyper Text Markup).

Ajax: Unión de varias tecnologías que juntas puede lograr cosas realmente impresionantes. Es el acrónimo para Asynchronous JavaScript + XML y su concepto es cargar y renderizar una página

XP: Programación Externa.

CASE: Ingeniería del software asistida por computadora.

CORBA: Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. (Common Object Request Broker Architecture).

IDL: Lenguaje de especificación de interfaces que se usa como parte de la tecnología CORBA. Ofrece la sintaxis necesaria para definir los métodos que queremos invocar remotamente. (Interface Definition Language).

Rational Rose: Herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto.

GNU: Conjunto de programas desarrollados por miembros de la Fundación por el Software Libre, son de uso gratuito (FSF- Free Software Foundation).

GNU/LINUX: Es un sistema operativo, es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Es multitarea, multiusuario, multiplataforma y multiprocesador.

GPL: Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre (General Public License).

UNIX: Sistema operativo portable, multitarea y multiusuario.

SGBD: Sistema Gestor de Bases de Datos.

BD: Base de Datos.

DBMS: Es un Sistema Gestor de Bases de Datos, un tipo de software muy específico, es equivalente al término SGBD (Data Base Management System).

Oracle: Es un sistema de administración de base de datos, es una potente herramienta cliente/servidor para la gestión de bases de datos.

Apache: Proyecto nacido para crear un servidor Web estable, fiable y veloz para plataformas Unix.

Perl: Lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX.

PostgreSQL: Servidor de Base de Datos relacional libre, liberado bajo la licencia BSD, es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

MVCC: Control de la Concurrencia Multi Versión es una técnica avanzada para mejorar las prestaciones de una base de datos en un entorno multiusuario. (Multi-Version Concurrency Control).

Linux o GNU/Linux:

LAMP: Es un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o Servidores dinámicos con un esfuerzo reducido.

En las tecnologías LAMP esto se consigue mediante la unión de las siguientes tecnologías:

- **Linux**, el sistema operativo.
- **Apache**, el servidor web.
- **MySQL**, el gestor de bases de datos.
- **Perl**, **PHP**, o **Python**, lenguajes de programación.

WAMP: Es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- **Windows**, como sistema operativo;
- **Apache**, como servidor web;
- **MySQL**, como gestor de bases de datos;

- PHP (generalmente), Perl, o Python, como lenguajes de programación.

IDE: Entorno de desarrollo integrado.

Framework: Marco de trabajo.

SMTP: Es un protocolo de la capa de aplicación, protocolo simple de transferencia de correo. (Simple Mail Transfer Protocol).

FTP: Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

SSL: Es un protocolo que permite la autenticación de servidores, la codificación de datos y la integridad de los mensajes.(Secure Socket Layer).

CGI: Es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos web que por medio del HTML se presentan de forma estática (Common Gateway Interface).

NTFS: Sistema de archivo.

BSD: Berkeley Software Distribution.

Phyton: Es un lenguaje de programación orientado a objetos moderno, de muy alto nivel, y con el que realizar aplicaciones o scripts lleva muy poco tiempo.

NNTP: Es un protocolo utilizado por los servidores de noticias de Usenet y los clientes (lectores). (Network News Transfer Protocol).

Usenet es un gran mensaje de sistema compartido que se utiliza en Internet.

SQLite: Es una librería escrita en lenguaje C que implementa un manejador de base de datos SQL embebido.

Solaris: Es un sistema operativo de tipo Unix.

Informix: Gestor de Base de Datos.