

Universidad de las Ciencias Informáticas

Facultad 5



**Título: Modelo para la gestión del conocimiento y la información
en el Polo Productivo de Realidad Virtual.**

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

Autor: Dailet Expósito Aguiar

Tutor: Ing. Liudmila Reyes Alvarez

Asesor: Ing. Manuel Martorell González

Junio 2009



“En la tierra hacen falta hombres que trabajen más y critiquen menos, que construyan más y destruyan menos, que prometan menos y resuelvan más, que esperen recibir menos y dar más, que digan mejor ahora que mañana.”

A small, stylized signature or logo in the bottom right corner, consisting of a few fluid, connected strokes.

DATOS DE CONTACTO

Liudmila Reyes Alvarez

Ingeniera en Ciencias Informáticas graduada el 19 de Julio del 2007. Ha trabajado desde 2do año de la carrera en proyectos de Realidad Virtual hasta el quinto año, primero como Diseñadora y luego como programadora hasta que se graduó y comenzó a trabajar en temas de calidad del proceso de desarrollo de software y en gestión de proyectos específicamente en la Gestión del conocimiento para un polo de la UCI. Actualmente se desempeña como Asesora de Calidad de la Facultad 5 y como profesora de Matemáticas, trabaja en el Vicedecanato y en el departamento de Ciencias Básicas de la facultad 5 de la Universidad de las Ciencias Informáticas. Se encuentra cursando la Maestría de Gestión de Proyectos.

Correo lreyes@uci.cu , teléfono del Departamento: 2471

AGRADECIMIENTOS

A mi padre y a mi madre por educarme como lo hicieron, por enseñarme lo que realmente es importante en la vida, por la confianza que depositaron en mí y por todo el apoyo que me han dado.

A mi hermano gracias por ser mi mejor amigo, por acompañarme siempre, en las buenas y las malas, y por cuidar a nuestros padres en mi ausencia.

A mis abuelas y al resto de mi familia que siempre estuvo pendiente de mi desarrollo universitario y me estimularon siempre a seguir adelante.

A Manuel, por todo lo que me ha enseñado y me ha ayudado en estos años. Y por todo lo bueno que logró en mí desde que nos conocemos. Y a su familia por estar pendiente de mis resultados.

A mi tutora por todo lo que hizo. A Yadira Brunet y a Jordanís Piñeiro por los momentos que me dedicaron los cuales fueron de mucha ayuda para el buen desarrollo de la tesis.

A mis viejas amigas Yily, Betty y Yado por comprenderme y animarme con cariño. Y a las nuevas, Kare, Mile y Leidy por estar ahí para mí.

A mis amigos de Código y Letra, y a mis compañeros de siempre del aula que a lo largo de estos 5 años de una forma u otra colaboraron para que yo lograra el gran sueño de mi vida, graduarme.

DEDICATORIA

*A mi madre y a mi padre que tanto esfuerzo hicieron durante
toda mi vida para que yo llegara a donde estoy.*

*A mi hermano que este trabajo sirva de estímulo para que
también se esfuerce al máximo para conseguir lo que se proponga.*

RESUMEN

La importancia del desarrollo de aplicaciones Web para la gestión radica en la necesidad de aumentar el nivel de acceso a los datos que se manejan. Producto al gran impacto que ha alcanzado la gestión del conocimiento a nivel mundial y las grandes ventajas que proporciona su uso, la facultad 5 de la Universidad de Ciencias Informáticas, está realizando un esfuerzo en el estudio y desarrollo de un sistema que permita identificar, agrupar y compartir continuamente conocimiento producido en el polo de Realidad Virtual (RV).

Debido a que algunos integrantes del polo de RV se les hace difícil el acceso a la información que se genera, el conocimiento adquirido a nivel de cada proyecto del polo no se socializa por lo que este es repetido en varias ocasiones y no es reutilizado, surge la necesidad de diseñar un modelo para la gestión del conocimiento y la información, en el polo productivo RV sobre la base de modelos existentes.

Este trabajo de diploma está estructurado en cuatro capítulos. En el Capítulo 1 se exponen los principales modelos de gestión del conocimiento y se propone un modelo específico para el polo. En el Capítulo 2 se hace un estudio de las tecnologías libres que existen en el mundo para el desarrollo de las aplicaciones web, así como de las metodologías que rigen el proceso de desarrollo de software. En los Capítulos 3 y 4 se desarrolla toda la ingeniería del software, desde el modelo de dominio hasta el análisis y diseño de la solución propuesta.

PALABRAS CLAVE

Conocimiento, gestión, gestión del conocimiento, modelo.

ÍNDICE

RESUMEN	III
ÍNDICE	IV
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema.....	5
1.2.1 Gestión del conocimiento	5
1.2.2 Conocimiento.....	6
1.2.3 Gestión.....	6
1.3 Modelos existentes hoy en día para la gestión del conocimiento.....	7
1.3.1 Proceso de creación del conocimiento (Nonaka, Takeuchi, 1995)	7
1.3.2 Modelo Balanced Scorecard (Kaplan, 1996)	8
1.3.3 Modelo de activos intangibles (Sveiby, 1997)	10
1.3.4 Modelo de gestión del conocimiento de KPMG Consulting (Tejedor y Aguirre, 1998)	11
1.3.5 Modelo Skandia Navigator (Edvinsson, 1999)	14
1.3.6 Modelo Andersen (Arthur Andersen, 1999)	14
1.3.7 Modelo EFQM de Excelencia (Rodríguez, 1999)	16
1.3.8 Modelo Social Capital Benchmarking System (Marti, 2001)	16
1.3.9 Modelo de Knowledge Practices Management (Díaz, 2005)	17
1.3.10 Knowledge Management Assessment Tool (KMAT) (Andersen y APQC)	18
1.4 El proceso de Gestión del conocimiento.....	19
1.5 Resultados	20
1.5.1 Modelo propuesto.....	21
1.6 Conclusiones parciales	23
CAPÍTULO 2: TECNOLOGÍAS Y TÉCNICAS UTILIZADAS EN EL DESARROLLO DE APLICACIONES WEB.....	24
2.1 Introducción.....	24

2.2 Software Libre y Código Abierto.....	24
2.3 Aplicaciones Web.....	25
2.4 Servidores Web.....	27
2.4.1 Apache.....	27
2.5 Tecnologías y técnicas utilizadas en el desarrollo de aplicaciones web.....	28
2.5.1 Tecnologías del lado del Cliente	29
2.5.1.1 Ajax	29
2.5.2 Tecnologías del lado del Servidor.....	30
2.5.2.1 PHP: Hypertext Preprocessor (PHP)	30
2.5.2.2 Symfony	31
2.5.3 Sistemas Gestores de Bases de Datos (SGBD).....	32
2.5.3.1 PostgreSQL	33
2.6 El Proceso Unificado de Desarrollo (RUP).....	35
2.7 El Lenguaje Unificado de Modelado (UML).....	36
2.8 Herramienta Case	37
2.8.1 Visual Paradigm	38
2.9 Entorno de desarrollo integrado (<i>IDE</i>).....	39
2.9.1 Eclipse	39
2.10 Justificación de la propuesta de las herramientas	40
2.11 Conclusiones parciales	41
CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA	42
3.1 Introducción.....	42
3.2 Propuesta del sistema	42
3.3 Modelo de dominio	42
3.3.1 Definición de objetos del dominio	43
3.3.2 Representación del Modelo de dominio.....	43
3.4 Requerimientos de Software	44
3.4.1 Requerimientos funcionales	44
3.4.2 Requerimientos no funcionales.....	45
3.5 Definición de los casos de uso.....	46
3.5.1 Definición de los actores	46

3.5.2 Diagrama de casos de uso.....	47
3.5.3 Descripción detallada de los casos de uso.....	48
3.6 Conclusiones parciales	55
CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA.....	56
4.1 Introducción.....	56
4.2 Modelo de Análisis	56
4.2.1 Diagramas de clases del análisis.....	56
4.3 Modelo de Diseño	57
4.3.1 Diagramas de clases del diseño	58
4.3.2 Diagramas de secuencia.....	64
4.3.3 Descripción de las clases.....	70
4.4 Diseño de la Base de datos	72
4.4.1 Descripción de las tablas	73
4.5 Definiciones de diseño que se apliquen.	75
4.6 Seguridad	76
4.7 Interfaz.....	76
4.8 Concepción de la ayuda	76
4.9 Conclusiones parciales	77
CONCLUSIONES	78
RECOMENDACIONES.....	79
REFERENCIAS BIBLIOGRÁFICAS	80
BIBLIOGRAFÍA	82
ANEXOS	84
GLOSARIO	102

ÍNDICE DE FIGURAS

Figura 1: Modelo de (Nonaka, Takeuchi, 1995).....	7
Figura 2: Modelo de (Kaplan, 1996)	10
Figura 3: Modelo de (Sveiby, 1997)	11
Figura 4: Modelo de (Tejedor y Aguirre, 1998)	11
Figura 5: Modelo de (Edvinsson, 1999)	14
Figura 6: Modelo de (Arthur Andersen, 1999).....	15
Figura 7: Modelo de (Rodríguez, 1999)	16
Figura 8: Modelo de (Marti, 2001)	17
Figura 9: Modelo de (Díaz, 2005)	17
Figura 10: Modelo KMAT (Andersen y APQC)	18
Figura 11: Proceso de gestión del conocimiento	19
Figura 12: Esquema de los componentes del modelo propuesto	21
Figura 13: Esquema de cómo se maneja el conocimiento y la información dentro del polo	22
Figura 14: Modelo de dominio.	44
Figura 15: Diagrama de casos de uso del sistema.....	47
Figura 16: Diagrama de clases del análisis del caso de uso "Autenticar usuario"	56
Figura 17: Diagrama de clases del análisis del caso de uso "Contener Foro"	57
Figura 18: Diagrama de clases del análisis del caso de uso "Gestionar información"	57
Figura 19: Diagrama de clases del diseño "Autenticar usuario".....	58
Figura 20: Diagrama de clases del diseño "Participar foro"	59
Figura 21: Diagrama de clases del diseño "Crear tema foro"	60
Figura 22: Diagrama de clases del diseño "Subir información"	61
Figura 23: Diagrama de clases del diseño "Bajar información"	62
Figura 24: Diagrama de clases del diseño "Modificar información"	63
Figura 25: Diagrama de clases del diseño "Eliminar información"	64
Figura 26: Diagrama de secuencia "Autenticar usuario"	65
Figura 27: Diagrama de secuencia "Participar foro"	66
Figura 28: Diagrama de secuencia "Crear tema foro"	67
Figura 29: Diagrama de secuencia "Subir información"	67

Figura 30: Diagrama de secuencia "Bajar información"	68
Figura 31: Diagrama de secuencia "Modificar información"	69
Figura 32: Diagrama de secuencia "Eliminar información"	70
Figura 33: Diagrama Entidad Relación de la Base de Datos	73
Figura 34: Diagrama de clases del análisis del caso de uso "Gestionar eventos"	84
Figura 35: Diagrama de clases del análisis del caso de uso "Mostrar información"	84
Figura 36: Diagrama de clases del análisis del caso de uso "Mostrar eventos"	85
Figura 37: Diagrama de clases del análisis del caso de uso "Gestionar proyectos"	85
Figura 38: Diagrama de clases del análisis del caso de uso "Mostrar proyectos"	85
Figura 39: Diagrama de clases del diseño "Insertar proyecto"	86
Figura 40: Diagrama de clases del diseño "Modificar proyecto"	87
Figura 41: Diagrama de clases del diseño "Eliminar proyecto"	88
Figura 42: Diagrama de clases del diseño "Insertar evento"	89
Figura 43: Diagrama de clases del diseño "Modificar evento"	90
Figura 44: Diagrama de clases del diseño "Eliminar evento"	91
Figura 45: Diagrama de clases del diseño "Mostrar información"	92
Figura 46: Diagrama de clases del diseño "Mostrar proyecto"	93
Figura 47: Diagrama de clases del diseño "Mostrar evento"	94
Figura 48: Diagrama de secuencia "Insertar proyecto"	95
Figura 49: Diagrama de secuencia "Modificar proyecto"	95
Figura 50: Diagrama de secuencia "Eliminar proyecto"	96
Figura 51: Diagrama de secuencia "Insertar evento"	97
Figura 52: Diagrama de secuencia "Modificar evento"	97
Figura 53: Diagrama de secuencia "Eliminar evento"	98

ÍNDICE DE TABLAS

Tabla 1: Relación entre datos, información y conocimiento.	6
Tabla 2: Definición de los actores del sistema.....	46
Tabla 11: Descripción detallada del caso de uso “Autenticar usuario”.	48
Tabla 12: Descripción detallada del caso de uso “Contener Foro”.	48
Tabla 13: Descripción detallada del caso de uso “Gestionar información”.	49
Tabla 14: Descripción detallada del caso de uso “Gestionar evento”.	51
Tabla 15: Descripción detallada del caso de uso “Mostrar información”.....	53
Tabla 16: Descripción detallada del caso de uso “Mostrar eventos”.....	53
Tabla 17: Descripción detallada del caso de uso “Gestionar proyecto”.	53
Tabla 18: Descripción detallada del caso de uso “Mostrar proyecto”.....	55
Tabla 19: Descripción de la clase AutenticarUsuarioAction.	70
Tabla 20: Descripción de la clase ForoAction.	70
Tabla 21: Descripción de la clase GestionarInformacionAction.	71
Tabla 22: Descripción de la clase GestionarProyectoAction.	71
Tabla 23: Descripción de la clase GestionarEventoAction.	72
Tabla 24: Descripción de la tabla Usuario.....	73
Tabla 25: Descripción de la tabla Proyecto.....	74
Tabla 26: Descripción de la tabla Información.....	74
Tabla 27: Descripción de la tabla Foro.....	74
Tabla 28: Descripción de la tabla Evento.....	75
Tabla 29: Descripción de la clase entidad Usuario.....	98
Tabla 30: Descripción de la clase entidad Foro.....	99
Tabla 31: Descripción de la clase entidad Información.....	99
Tabla 32: Descripción de la clase entidad Proyecto.....	100
Tabla 33: Descripción de la clase entidad Evento.....	101

INTRODUCCIÓN

El creciente auge y expansión del mundo de la información que se vive en la actualidad, ha traído consigo la necesidad del manejo y control de cada vez mayores cantidades de información. La era de la información y del creciente desarrollo de las tecnologías se encuentra en una etapa donde el factor esencial es el conocimiento. En esta nueva sociedad se puede observar como la información ha llegado a todos los espacios del planeta y como el uso de ordenadores y redes de comunicación se han convertido en parte del quehacer cotidiano y así colaborar con la socialización de ese conocimiento producido. Es por ello que en este mundo dominado por las grandes multinacionales, cada vez se destinan mayores cantidades de recursos a la creación de sistemas de información que les permita sacar el máximo provecho de sus datos.

Existen dos tipos de conocimiento el conocimiento tácito y conocimiento explícito. El conocimiento tácito es aquel que se encuentra en la mente de las personas y resulta difícil expresar o formalizar. Y El conocimiento explícito es el que podemos adquirir fácilmente mediante la comunicación y el intercambio porque este si puede ser plasmado o transmitido en libros o escritos.

El término Gestión del Conocimiento (GC), también conocido con el término Knowledge Management, se ha puesto de moda en los últimos años, muy especialmente en el mundo de los negocios. Pocas empresas habían comenzado a implantar estrategias de GC antes de 1999. Sin embargo, ya en el año 2000, muchas organizaciones se han planteado iniciativas o han arrancado proyectos en esa línea. Gestión del Conocimiento se puede definir como identificar, adquirir, almacenar, usar y participar del conocimiento generado en una organización. Se hace necesaria la creación de una vía para que se acumule el conocimiento y se desarrollen las habilidades mediante la socialización del mismo a los miembros de la empresa.

Cuba no está ajena al progreso y constante cambio como consecuencia del desarrollo de las Tecnologías de la informática y las comunicaciones (TIC). Nuestro país, se encuentra inmerso en un proceso de transformaciones desde el punto de vista tecnológico, dirigidas a desarrollarse tecnológicamente y lograr puestos a nivel mundial en este campo. Dentro de las muchas tecnologías que hoy en día se estudian, se

encuentran las relacionadas con la gestión del conocimiento. Existen empresas que disponen de grandes volúmenes de información, sobre sus propios procesos, sobre el entorno tecnológico y competitivo en el que se desarrollan, sobre los clientes con los que interactúan, etc.

Se han creado diversas instituciones a nivel nacional que contribuyen al crecimiento informático del país. Entre estas instalaciones se encuentra la Universidad de Ciencias Informáticas (UCI), una universidad de nuevo tipo en el país, donde se vincula la producción de software, la investigación y la docencia, para así convertirse en una empresa de producción de software en Cuba. La UCI no está exenta al proceso de cambio y de creación del conocimiento. Existe un exceso de información que está favorecido por la explosión de Internet y por el uso que a veces se hace de la tecnología informática. La Universidad de Ciencias Informáticas, es la universidad mejor equipada tecnológicamente que existe en Cuba sin embargo no se cuenta con un modelo propio para la gestión de todo ese conocimiento generado que viene envuelto con el desarrollo.

La Universidad de las Ciencias informáticas está conformada por 10 facultades las cuales en la parte productiva se dividen por Polos Productivos (PP) que son los encargados de agrupar proyectos que responden a una misma línea de investigación. Específicamente la facultad 5 de la UCI cuenta con dos PP, uno de Realidad Virtual y el otro de Hardware y Automática. En el polo productivo de Realidad Virtual a algunos integrantes del polo se les hace difícil el acceso a la información que se genera, se dificulta la toma de decisiones y el conocimiento adquirido a nivel de cada proyecto del polo no se socializa por lo que este conocimiento es repetido en varias ocasiones y no es reutilizado.

De lo anteriormente plantado se deriva el siguiente **Problema a resolver:**

¿Cuál sería el modelo de Gestión del Conocimiento apropiado para el Polo Productivo de Realidad Virtual?

El problema anterior está enmarcado en el **Objeto de estudio:** la gestión del conocimiento. Y como **Campo de acción:** Gestión del conocimiento en el polo realidad virtual.

Como **idea a defender** se tiene que la aplicación de un modelo de Gestión del Conocimiento (GC) brinda la posibilidad de crear un ambiente en el que el conocimiento y la información disponibles en el polo

productivo sean accesibles y puedan ser usados por las personas que integran los proyectos que conforman el polo.

Este trabajo de diploma tiene como **Objetivo General:** Diseñar un modelo para la gestión del conocimiento y la información, en el Polo Productivo de Realidad Virtual de la facultad 5 sobre la base de modelos existentes.

Para el logro del objetivo de esta investigación se plantean las **Tareas investigativas:**

- Analizar los modelos de gestión del conocimiento orientados al aprovechamiento de las posibilidades del polo de Realidad Virtual en las áreas de investigación y producción.
- Proponer un modelo para la gestión del conocimiento en el polo productivo Realidad Virtual.
- Seleccionar la metodología de análisis y diseño que facilite la creación y garantice la calidad de la aplicación web.
- Seleccionar las herramientas para llevar a cabo el sistema y la elección de la plataforma en la que se desarrollará la aplicación, fundamentando dicha elección.
- Realizar el análisis y diseño del sistema para la gestión del conocimiento, para una futura implementación.

Para el desarrollo de las tareas investigativas se utilizan los diferentes métodos.

Entre los **Métodos Teóricos**

- Analítico – sintético: se analizarán las teorías y documentos que permitan la extracción de elementos fundamentales que se relacionen con el objeto de estudio, es decir, con la Gestión del conocimiento.
- Inductivo – deductivo: partiendo del análisis de los procesos básicos que se llevan a cabo en la gestión del conocimiento hasta aquellos generalizadores que permitan llegar a una solución dentro de la investigación.
- Análisis Histórico – lógico: partiendo de las tareas de investigación que se plantean, se emplea este método para realizar un análisis de los modelos de gestión del conocimiento y para examinar las herramientas a utilizar en el desarrollo de la aplicación.

- **Modelación:** Se utiliza en la elaboración de diagramas y modelos que posibilitarán el desarrollo del sistema.

Y entre los **Métodos Empíricos**

- **Observación:** se observará el desarrollo de todos los procesos relacionados con la generación del conocimiento.
- **Entrevistas:** se planificarán conversaciones con las personas relacionadas con el desarrollo del software.

El presente trabajo de diploma esta organizado en cuatro capítulos.

En el **capítulo 1**, “Fundamentación Teórica”, se hace un análisis bibliográfico donde se investigan las características de los modelos para la gestión del conocimiento, así como las tendencias actuales de la gestión del conocimiento y como se desarrolla el proceso de gestión del conocimiento.

En el **capítulo 2**, “Tecnologías y técnicas utilizadas en el desarrollo de aplicaciones web”, donde se exponen algunas de las herramientas y tecnologías que se emplearán en el desarrollo de la aplicación web. Además se justifica por qué se escogieron esas tecnologías.

En el **capítulo 3**, “Características del Sistema”, se exponen las características técnicas que presentará el sistema como solución al problema planteado, se analiza con mayor profundidad el objeto de estudio, y se define el modelo de dominio. Además se hace la captura de requisitos y se realiza el modelo de casos de uso del sistema.

En el **capítulo 4**, “Análisis y diseño del sistema”, se presenta el análisis y el diseño de la aplicación a través de los diferentes diagramas existentes, incluyendo el diseño de la base de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En la actualidad los avances en la informática y las telecomunicaciones han mejorado y continúan mejorando la capacidad de almacenamiento y representación de la información. Por otro lado están permitiendo que esa información transite con fluidez entre los usuarios. En el presente capítulo se dan algunos de los principales conceptos asociados al objeto de estudio. Además se hace un análisis de las características de los modelos de gestión del conocimiento más conocidos y aplicados en el mundo. Para finalizar se propone un modelo aplicable al polo productivo de Realidad Virtual.

1.2 Conceptos asociados al dominio del problema

Con el objetivo de que el lector pueda tener una mejor comprensión del tema que será abordado en esta investigación, se describen a continuación un grupo de conceptos asociados al dominio del problema, relacionados con el objeto de estudio de la investigación.

1.2.1 Gestión del conocimiento

Podemos definir la gestión del conocimiento como el proceso de identificar, agrupar, ordenar y compartir continuamente conocimiento de todo tipo para satisfacer necesidades presentes y futuras, para identificar y explotar recursos de conocimiento tanto existentes como adquiridos y para desarrollar nuevas oportunidades. La práctica habitual es crear un foro virtual donde las experiencias individuales y los conocimientos se suman en un espacio que puede ser accesible a todos sus miembros. (1)

El Proceso de Gestión del Conocimiento es una interacción continuada, persistente y determinada entre agentes humanos a través de agentes participando con el objetivo de administrar (manipular, dirigir, gobernar, controlar, coordinar, planificar y organizar) otros agentes, componentes y actividades que participan en el proceso de conocimiento básico (la producción de conocimiento y la integración de conocimiento) en un ente planificado, dirigido y unificado, produciendo, manteniendo, reforzando, adquiriendo y transmitiendo la base de conocimiento de la organización.(Ph.D. Joseph Firestone)(2)

Después del análisis de los diferentes conceptos se concluye que la gestión del conocimiento no es más que la manipulación del conocimiento explícito producido por un grupo de personas para la utilización de

todos los miembros de esa organización. Es lograr que toda la información que se genere en un determinado organismo sea accesible por todos sus miembros. Para así lograr el incremento del capital intelectual.

1.2.2 Conocimiento

Si analizamos la trilogía datos – información – conocimiento, podemos ver que el conocimiento surge de la adecuada gestión de los datos y la información.

Tabla 1: Relación entre datos, información y conocimiento.

Datos	Información	Conocimiento
<p>Observaciones sencillas de distintos sucesos.</p> <ul style="list-style-type: none"> • Se capturan con facilidad en las máquinas. • Se estructuran fácilmente. • A menudo se cuantifican. • Se transfieren con facilidad. 	<p>Datos dotados de pertenencia y propósito.</p> <ul style="list-style-type: none"> • Requiere una unidad de análisis. • Necesita consenso sobre el significado. • La intermediación humana es indispensable. 	<p>Información valiosa de mente humana.</p> <ul style="list-style-type: none"> • Difícil de estructurar. • Difícil de capturar en las máquinas. • A menudo es tácito. • La transferencia es complicada.

El diccionario define conocimiento como: “El producto o resultado de ser instruido, el conjunto de cosas sobre las que se sabe o que están contenidas en la ciencia”. El diccionario acepta que la existencia de conocimiento es muy difícil de observar y reduce su presencia a la detección de sus efectos posteriores. Los conocimientos se almacenan en la persona (o en otro tipo de agentes). Esto hace que sea casi imposible observarlos. (3)

1.2.3 Gestión

Del latín *gestio*, el concepto de gestión hace referencia a la acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera. Administrar, por otra parte, consiste en gobernar, dirigir, ordenar, disponer u organizar.

El término gestión, por lo tanto, implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una empresa o de un negocio. (4)

1.3 Modelos existentes hoy en día para la gestión del conocimiento

A continuación se explican algunos de los modelos que existen en el mundo para la gestión del conocimiento.

1.3.1 Proceso de creación del conocimiento (Nonaka, Takeuchi, 1995)

El proceso de creación del conocimiento para Nonaka y Takeuchi (1995) es a través de un modelo de generación de conocimiento mediante dos espirales de contenido epistemológico y ontológico.

Es un proceso de interacción entre conocimiento tácito y explícito que tiene naturaleza dinámica y continua. Se constituye en una espiral permanente de transformación ontológica interna de conocimiento, desarrollada siguiendo 4 fases que podemos ver de forma gráfica en la siguiente figura:

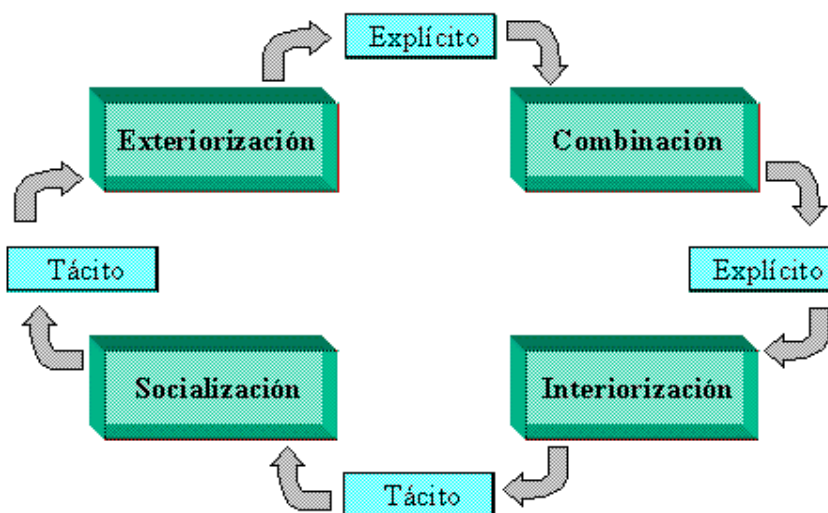


Figura 1: Modelo de (Nonaka, Takeuchi, 1995)

- **La Socialización:** es el proceso de adquirir conocimiento tácito a través de compartir experiencias por medio de exposiciones orales, documentos, manuales y tradiciones y que añade el conocimiento novedoso a la base colectiva que posee la organización.

- **La Exteriorización:** es el proceso de convertir conocimiento tácito en conceptos explícitos que supone hacer tangible mediante el uso de metáforas el conocimiento de por sí difícil de comunicar, integrándolo en la cultura de la organización; es la actividad esencial en la creación del conocimiento.
- **La combinación:** es el proceso de crear conocimiento explícito al reunir conocimiento explícito proveniente de cierto número de fuentes, mediante el intercambio de conversaciones telefónicas, reuniones, correos, etc., y se puede categorizar, confrontar y clasificar para formar bases de datos para producir conocimiento explícito.
- **La Interiorización:** es un proceso de incorporación de conocimiento explícito en conocimiento tácito, que analiza las experiencias adquiridas en la puesta en práctica de los nuevos conocimientos y que se incorpora en las bases de conocimiento tácito de los miembros de la organización en la forma de modelos mentales compartidos o prácticas de trabajo.

Para Nonaka y Takeuchi, lo expresado por Peter Drucker en el sentido de que, la esencia de la dirección es, cómo se puede aplicar de la mejor forma un conocimiento existente para poder crear otro conocimiento nuevo o reciclado, es justificado ya que sus estudios en compañías japonesas respaldan el proceso de creación del conocimiento que ambos japoneses han sostenido. (5)

1.3.2 Modelo Balanced Scorecard (Kaplan, 1996)

Trata de incorporar a los sistemas tradicionales de medición para la gestión, algunos aspectos no financieros que condicionan la obtención de resultados económicos. Ofrece un marco conceptual para conocer si se están utilizando los procesos y personas adecuadas para obtener un mejor rendimiento empresarial. Suministra una lista de recursos intangibles susceptibles de gestionarse y de tratarse desde el punto de vista del conocimiento. Propone dos campos de reflexión: uno de ellos de base: pretensión estratégica de la formación, y el otro operativo: cómo establecer la jerarquía de los vacíos de formación.

Introduce en el sistema de información a disposición de los que toman decisiones, variables estratégicas a considerar más allá de las convencionales y que pueden indicar vacíos de formación sustanciales antes

olvidados o difíciles de justificar. El modelo integra los indicadores financieros (de pasado) con los no financieros (de futuro), y los integra en un esquema que permite entender las interdependencias entre sus elementos, así como la coherencia con la estrategia y la visión de la empresa. Dentro de cada bloque, se distinguen dos tipos de indicadores: indicadores driver (factores condicionantes de otros) e indicadores output (indicadores de resultado). (5). Ver figura 2.

El modelo presenta cuatro bloques:

- **Perspectiva financiera:** contempla los indicadores financieros como el objetivo final; considera que éstos no deben sustituirse, sino complementarse con otros que reflejan la realidad empresarial.
- **Perspectiva de cliente:** identifica los valores relacionados con los clientes. Para ello, es necesario definir previamente los segmentos de mercado objetivo y realizar un análisis del valor y calidad de éstos.
- **Perspectiva de procesos internos de negocio:** Analiza la adecuación de los procesos internos de la empresa de cara a la obtención de la satisfacción del cliente y a conseguir altos niveles de rendimiento financiero. Para alcanzar este objetivo se propone un análisis de los procesos internos desde una perspectiva de negocio y una predeterminación de los procesos clave por medio de la cadena de valor.

Se distinguen tres tipos de procesos:

- Procesos de innovación (difícil de medir).
 - Procesos de operaciones. Se desarrollan mediante los análisis de calidad y reingeniería.
 - Procesos de servicio postventa. Critica la concepción de la formación como un gasto, no como una inversión.
- **Perspectiva del aprendizaje y mejora que clasifica los activos relativos al aprendizaje y mejora en:** Capacidad y competencia de las personas (gestión de los empleados); Sistemas de información; así como Cultura-clima-motivación para el aprendizaje y la acción.



Figura 2: Modelo de (Kaplan, 1996)

1.3.3 Modelo de activos intangibles (Sveiby, 1997)

Se basa en la importancia de los activos intangibles. Identifica:

- **Competencias de las personas:** Incluye las competencias de la organización como son planificar, producir, procesar o presentar productos o soluciones (que sería el capital humano).
- **Estructura interna:** Es el conocimiento estructurado de la organización como las patentes, procesos, modelos, sistemas de información, cultura organizativa, las personas que se encargan de mantener dicha estructura (que sería el capital estructural).
- **Estructura externa:** Comprende las relaciones con clientes y proveedores, las marcas comerciales y la imagen de la empresa (que sería el capital relacional). Estos activos intangibles, forman lo que se conoce como el balance invisible.

Se propone para la medición y evaluación de estos, tres tipos de indicadores dentro de cada uno de los tres bloques: **indicadores de crecimiento e innovación:** recogen el potencial futuro de la empresa; **indicadores de eficiencia:** informan hasta qué punto los intangibles son productivos (activos), e **indicadores de estabilidad:** indican el grado de permanencia de estos activos en la empresa. (5). Ver figura 3.



Figura 3: Modelo de (Sveiby, 1997)

1.3.4 Modelo de gestión del conocimiento de KPMG Consulting (Tejedor y Aguirre, 1998)

El modelo parte de la siguiente pregunta: ¿qué factores condicionan el aprendizaje de una organización y qué resultados produce dicho aprendizaje?

Para responder a esta pregunta KPMG realiza un esfuerzo que produce un modelo cuya finalidad es la exposición clara y práctica de los factores que condicionan la capacidad de aprendizaje de una organización, así como los resultados esperados del aprendizaje.

Una de las características esenciales del modelo es la interacción de todos sus elementos, que se presentan como un sistema complejo en el que las influencias se producen en todos los sentidos.

La estructura organizativa, la cultura, el liderazgo, los mecanismos de aprendizaje, las actitudes de las personas, la capacidad de trabajo en equipo, etc., no son independientes, sino que están conectados entre sí. (5). Ver figura 4.

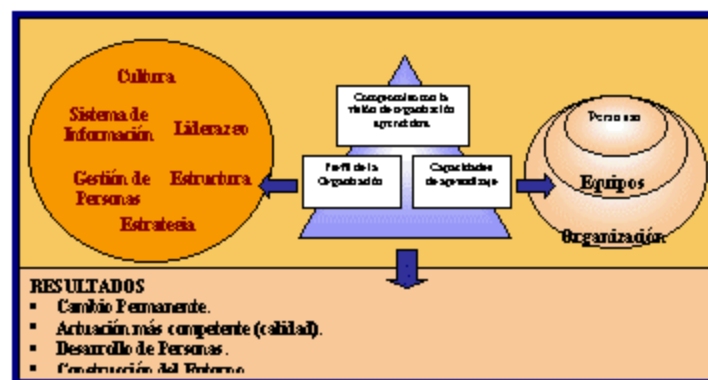


Figura 4: Modelo de (Tejedor y Aguirre, 1998)

Los factores condicionantes del aprendizaje.

Los factores que configuran la capacidad de aprender de una empresa han sido estructurados en los tres bloques siguientes, atendiendo a su naturaleza:

1.- Compromiso firme y consciente de toda la empresa, en especial de sus líderes, con el aprendizaje generativo, continuo, consciente y a todos los niveles.

El primer requisito para el éxito de una iniciativa de gestión del conocimiento es reconocer explícitamente que el aprendizaje es un proceso que debe ser gestionado y comprometerse con todo tipo de recursos.

2.- Comportamientos y mecanismos de aprendizaje a todos los niveles. La organización como ente no humano sólo puede aprender en la medida en que las personas y equipos que la conforman sean capaces de aprender y deseen hacerlo.

Disponer de personas y equipos preparados es condición necesaria pero no suficiente para tener una organización capaz de generar y utilizar el conocimiento mejor que las demás. Para lograr que la organización aprenda es necesario desarrollar mecanismos de creación, captación, almacenamiento, transmisión e interpretación del conocimiento, permitiendo el aprovechamiento y utilización del aprendizaje que se da en el nivel de las personas y equipos.

Los comportamientos, actitudes, habilidades, herramientas, mecanismos y sistemas de aprendizaje que el modelo considera son:

- La responsabilidad personal sobre el futuro (pro actividad de las personas).
- La habilidad de cuestionar los supuestos (modelos mentales).
- La visión sistémica (ser capaz de analizar las interrelaciones existentes dentro del sistema, entender los problemas de forma no lineal y ver las relaciones causa-efecto a lo largo del tiempo).
- La capacidad de trabajo en equipo.
- Los procesos de elaboración de visiones compartidas.
- La capacidad de aprender de la experiencia.
- El desarrollo de la creatividad.
- La generación de una memoria organizacional.
- Desarrollo de mecanismos de aprendizaje de los errores.
- Mecanismos de captación de conocimiento exterior.
- Desarrollo de mecanismos de transmisión y difusión del conocimiento.

Si se consigue que las personas aprendan, pero no convierten ese conocimiento en activo útil para la organización, no se puede hablar de aprendizaje organizacional. La empresa inteligente practica la comunicación a través de diversos mecanismos, tales como reuniones, informes, programas de formación internos, visitas, programas de rotación de puestos, creación de equipos multidisciplinares.

3.- Desarrollo de las infraestructuras que condicionan el funcionamiento de la empresa y el comportamiento de las personas y grupos que la integran, para favorecer el aprendizaje y el cambio permanente.

Pero no debemos olvidar que las condiciones organizativas pueden actuar como obstáculos al aprendizaje organizacional, bloqueando las posibilidades de desarrollo personal, de comunicación, de relación con el entorno, de creación, etc.

Las características de las organizaciones tradicionales que dificultan el aprendizaje:

- Estructuras burocráticas.
- Liderazgo autoritario y/o paternalista.
- Aislamiento del entorno.
- Autocomplacencia.
- Cultura de ocultación de errores.
- Búsqueda de homogeneidad.
- Orientación a corto plazo.
- Planificación rígida y continuista.
- Individualismo.

En definitiva, la forma de ser de la organización no es neutra y requiere cumplir una serie de condiciones para que las actitudes, comportamiento y procesos de aprendizaje descritos puedan desarrollarse.

El modelo considera los elementos de gestión que afectan directamente a la forma de ser de una organización: cultura, estilo de liderazgo, estrategia, estructura, gestión de las personas y sistemas de información y comunicación. (5)

Los resultados del aprendizaje.

Una vez analizados los factores que condicionan el aprendizaje, el modelo refleja los resultados que debería producir ese aprendizaje. La capacidad de la empresa para aprender se debe traducir en:

- La posibilidad de evolucionar permanentemente (flexibilidad).
- Una mejora en la calidad de sus resultados.
- La empresa se hace más consciente de su integración en sistemas más amplios y produce una implicación mayor con su entorno y desarrollo.
- El desarrollo de las personas que participan en el futuro de la empresa.

1.3.5 Modelo Skandia Navigator (Edvinsson, 1999)

No está estructurado en tipos de capital sino que se compone de cinco áreas de enfoques. Proporciona un equilibrio entre: **el pasado** (enfoque financiero); **el presente** (enfoque de cliente: mide un tipo distinto de capital intelectual), **el enfoque humano** en el centro (la primera mitad del modelo de capital intelectual), y **el enfoque de proceso** (mide una gran parte del capital estructural); así como **el futuro** (enfoque de innovación y desarrollo: la otra parte del capital estructural). (5). Ver figura 5.

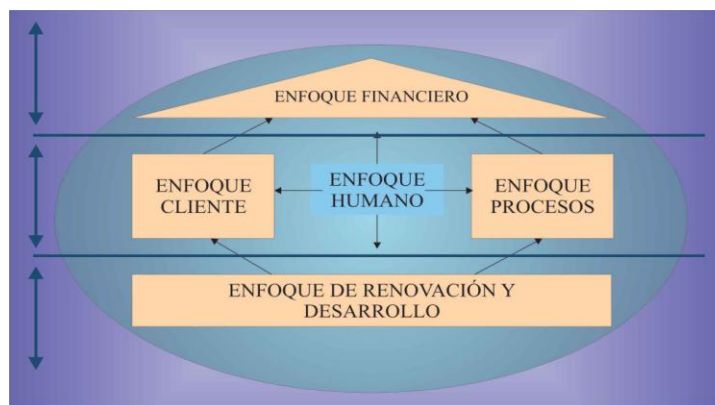


Figura 5: Modelo de (Edvinsson, 1999)

1.3.6 Modelo Andersen (Arthur Andersen, 1999)

Andersen (1999) reconoce la necesidad de acelerar el flujo de la información que tiene valor, desde los individuos a la organización y de vuelta a los individuos, de modo que ellos puedan usarla para crear valor para los clientes. Ver figura 6.

¿Qué hay de nuevo en este modelo?

Desde la perspectiva individual: la responsabilidad personal de compartir y hacer explícito el conocimiento para la organización.

Desde la perspectiva organizacional: la responsabilidad de crear la infraestructura de soporte para que la perspectiva individual sea efectiva, creando los procesos, la cultura, la tecnología y los sistemas que permitan capturar, analizar, sintetizar, aplicar, valorar y distribuir el conocimiento. (5)

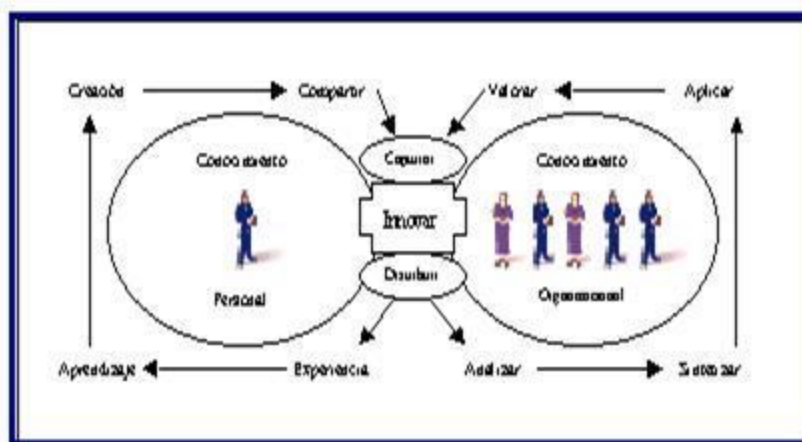


Figura 6: Modelo de (Arthur Andersen, 1999)

Se han identificado dos tipos de sistemas necesarios para el propósito fijado:

1.- Sharing Networks

- Acceso a personas con un propósito común a una comunidad de práctica. Estas comunidades son foros virtuales sobre los temas de mayor interés de un determinado servicio o industria. Existen más de 80 comunidades de prácticas.
- Ambiente de aprendizaje compartido
 - Virtuales: AA on line, bases de discusiones, etc.
 - Reales: Workshops, proyectos, etc.

2.- Conocimiento “empaquetado”

La espina dorsal de esa infraestructura se denomina “Arthur Andersen Knowledge Space”, que contiene:

- Global best practices.
- Metodologías y herramientas.
- Biblioteca de propuestas, informes, etc.

1.3.7 Modelo EFQM de Excelencia (Rodríguez, 1999)

Aunque se creó en 1988, es un año después que se modificó para incluir aspectos relacionados con la gestión del conocimiento, que subrayan la importancia de la innovación y el aprendizaje. Fue en 1999 que se presentó la versión actual del modelo EFQM de Excelencia.

En el criterio **Agentes colaboradores** y recursos se incluyó la gestión de la información y los conocimientos, y en el criterio **Procesos** se enfatizó la mejora y la innovación como son el liderazgo, la estrategia, la estructura, los procesos, las personas, los resultados y la medición. (5). Ver figura 7.

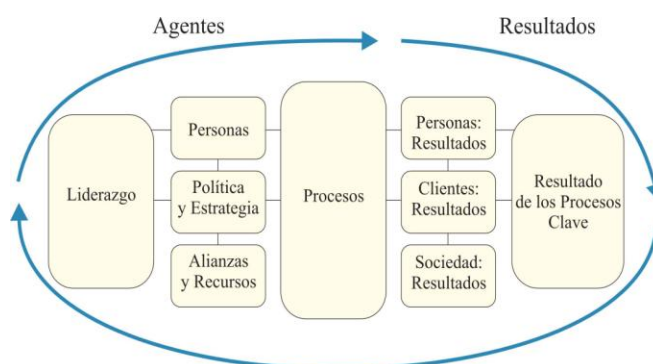


Figura 7: Modelo de (Rodríguez, 1999)

1.3.8 Modelo Social Capital Benchmarking System (Marti, 2001)

Posibilita utilizar el capital intelectual de las empresas, organizaciones e instituciones del entorno geográfico próximo (clúster, micro-clústeres o territorio) para construir la mejor organización posible en forma de red que necesita un modelo de negocio determinado, complementa así el capital intelectual interno con este capital intelectual externo de naturaleza relacional.

Sirve para seleccionar y evaluar las diferentes localizaciones alternativas que una empresa determinada elige para desarrollar sus actividades con el fin de aprovechar al máximo el capital intelectual del entorno en el proceso de construcción de la organización en forma de red.

Muestra como la empresa inteligente que se organiza en forma de red construye sus competencias esenciales gracias a las relaciones que le permiten acceder a las competencias y a los recursos de otras empresas, organizaciones e instituciones; algunas de ellas situadas dentro de un clúster específico y otras fuera de este.

Parte del principio de que las relaciones con las empresas, organizaciones e instituciones situadas en el clúster tienen un carácter primordial porque permiten realizar operaciones conjuntas o complementarias

con la empresa inteligente en cuestión, así como la transmisión de conocimientos tácitos que aportan un valor superior. (5). Ver figura 8.

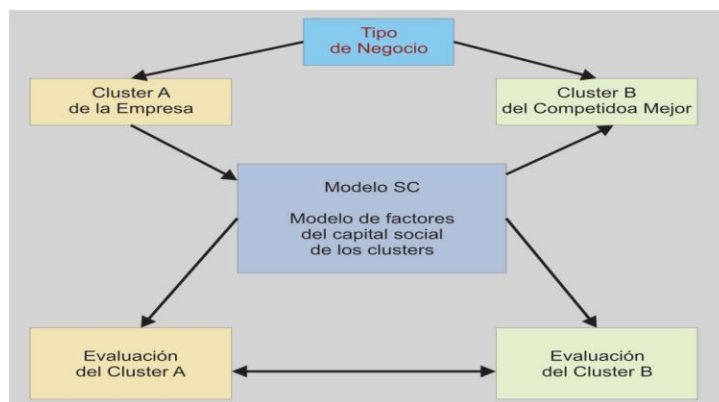


Figura 8: Modelo de (Marti, 2001)

1.3.9 Modelo de Knowledge Practices Management (Díaz, 2005)

Expone los factores que condicionan la capacidad de aprendizaje de una organización, así como los resultados esperados. Una de las características esenciales del modelo es la interacción de todos sus elementos, que se presentan como un sistema complejo en el que las influencias se producen en todos los sentidos. La estructura organizativa, la cultura, el liderazgo, los mecanismos de aprendizaje, las actitudes de las personas, la capacidad de trabajo en equipo, etc., no son independientes, sino que se conectan entre ellos. (5). Ver figura 9.

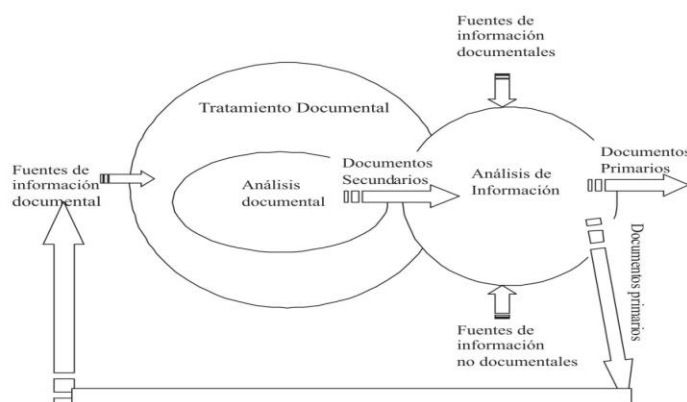


Figura 9: Modelo de (Díaz, 2005)

1.3.10 Knowledge Management Assessment Tool (KMAT) (Andersen y APQC)

El KMAT es un instrumento de evaluación y diagnóstico construido sobre la base del Modelo de Administración del Conocimiento Organizacional desarrollado conjuntamente por Arthur Andersen y APQC (American Productivity and Quality Center).

El modelo propone cuatro facilitadores (liderazgo, cultura, tecnología y medición) que favorecen el proceso de administrar el conocimiento organizacional. (5). Ver figura 10.

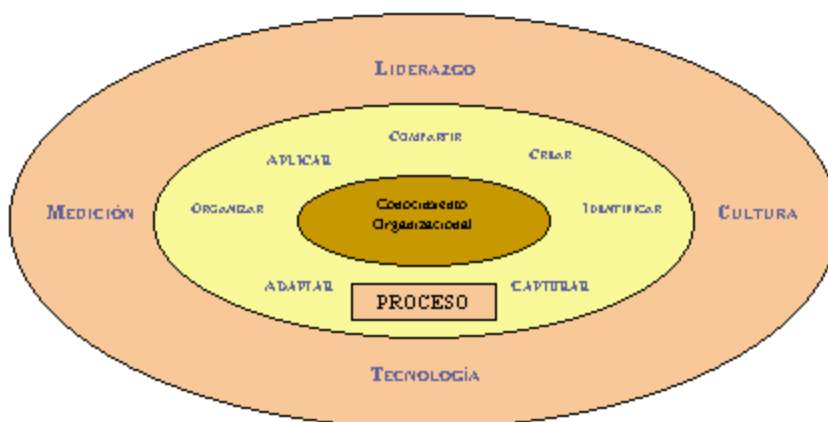


Figura 10: Modelo KMAT (Andersen y APQC)

- **Liderazgo:** Comprende la estrategia y cómo la organización define su negocio y el uso del conocimiento para reforzar sus competencias críticas.
- **Cultura:** Refleja cómo la organización enfoca y favorece el aprendizaje y la innovación incluyendo todas aquellas acciones que refuerzan el comportamiento abierto al cambio y al nuevo conocimiento.
- **Tecnología:** Se analiza cómo la organización equipa a sus miembros para que se puedan comunicar fácilmente y con mayor rapidez.
- **Medición:** Incluye la medición del capital intelectual y la forma en que se distribuyen los recursos para potenciar el conocimiento que alimenta el crecimiento.
- **Procesos:** Incluyen los pasos mediante los cuales la empresa identifica las brechas de conocimiento y ayuda a capturar, adoptar y transferir el conocimiento necesario para agregar valor al cliente y potenciar los resultados.

1.4 El proceso de Gestión del conocimiento

El proceso de Gestión del conocimiento debe entenderse como los subprocesos necesarios para el desarrollo de soluciones orientadas a generar las bases del conocimiento de valor para la organización. El proceso presentado en la figura 11 representa la cadena de agregación de valor a cada una de las instancias de conocimiento existentes en la organización. Cabe destacar que el proceso de Gestión del conocimiento se centra en el concepto de generación de valor asociado al negocio, el cual ayudará a descartar las instancias de conocimiento que sean no relevantes.

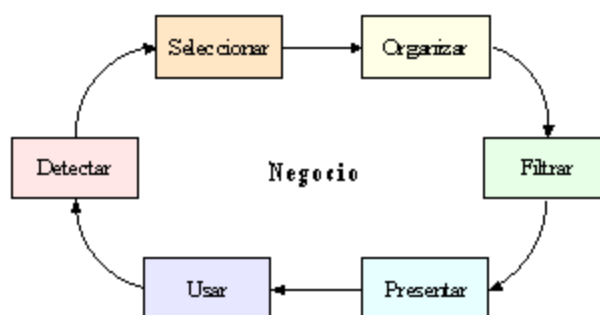


Figura 11: Proceso de gestión del conocimiento

Tal como se representa en la figura 11, la gestión del conocimiento puede ser descrita como el proceso sistemático de detectar, seleccionar, organizar, filtrar, presentar y usar la información por parte de los participantes de la organización, con el objeto de explotar cooperativamente los recursos de conocimiento basados en el capital intelectual propio de las organizaciones, orientados a potenciar las competencias organizacionales y la generación de valor, donde:

- **Detectar:** Es el proceso de localizar modelos cognitivos y activos (pensamiento y acción) de valor para la organización, el cual radica en las personas. Son ellas, de acuerdo a sus capacidades cognitivas (modelos mentales, visión sistémica, etc.), quienes determinan las nuevas fuentes de conocimiento de acción.
- **Seleccionar:** Es el proceso de evaluación y elección del modelo en torno a un criterio de interés. Los criterios pueden estar basados en criterios organizacionales, comunales o individuales, los cuales estarán divididos en tres grandes grupos: Interés, Práctica y Acción.

- **Organizar:** Es el proceso de almacenar de forma estructurada la representación explícita del modelo.
- **Filtrar:** Una vez organizada la fuente, puede ser accedida a través de consultas automatizadas en torno a motores de búsquedas. Las búsquedas se basarán en estructuras de acceso simples y complejas, tales como mapas de conocimientos, portales de conocimiento o agentes inteligentes.
- **Presentar:** Los resultados obtenidos del proceso de filtrado deben ser presentados a personas o máquinas. En caso que sean personas, las interfaces deben estar diseñadas para abarcar el amplio rango de comprensión humana. En el caso que la comunicación se desarrolle entre máquinas, las interfaces deben cumplir todas las condiciones propias de un protocolo o interfaz de comunicación.
- **Usar:** El uso del conocimiento reside en el acto de aplicarlo al problema objeto de resolver. De acuerdo con esta acción es que es posible evaluar la utilidad de la fuente de conocimiento a través de una actividad de retroalimentación.

Sobre el proceso descrito anteriormente, es posible desarrollar el concepto de proyecto de Gestión del conocimiento, el cual tiene como objetivo generar las instancias que reflejen de manera práctica cada una de las etapas del proceso. (6)

1.5 Resultados

Después de analizar los diferentes modelos de gestión del conocimiento anteriores se llega a la conclusión de que ellos fueron pensados según el momento histórico de la gestión del conocimiento que se estaba viviendo. Además de que cada uno estudia disímiles elementos o se enmarca en estructuras que no son las necesidades que presenta el polo productivo de Realidad virtual de la facultad 5. Algunas de las partes que se estudian en algunos de estos modelos se pudieran aplicar al polo, pero ningún modelo es aplicable completamente al polo de Realidad virtual. Por estas razones se propone un modelo específico que se adecua a las necesidades y la estructura del polo productivo.

1.5.1 Modelo propuesto

El modelo que se propone en este trabajo de diploma se obtiene del análisis de algunos de los modelos anteriores y de cómo se gestiona eficientemente el conocimiento y la información. Constituye una representación de la gestión del conocimiento en el polo productivo de Realidad Virtual enfocado a la gestión interna del conocimiento de cada proyecto que lo conforma, su necesaria retroalimentación a través de interacciones mutuas y su relación con el entorno. Ver figuras 12 y 13.

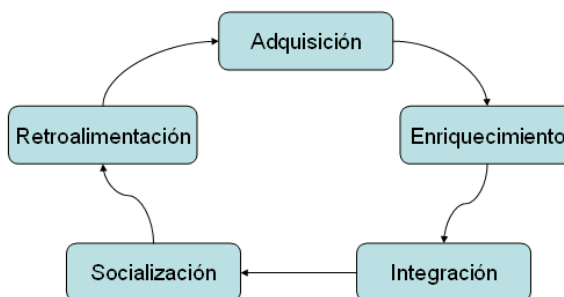


Figura 12: Componentes del modelo propuesto

Los componentes del modelo son:

- **Adquisición:** La adquisición del conocimiento es el objetivo de este proceso. Para una buena gestión del conocimiento se debe contar con que el conocimiento recopilado presenta un mínimo de calidad, esto puede lograrse mediante la validación de las fuentes de datos, información o conocimiento con que se trabaja, las cuales deben ser confiables, además de la verificación de su actualización. Después de realizada esta selección la última fase de este proceso será la estructuración y almacenamiento de ese conocimiento en bases de datos.
- **Enrichimiento:** El objetivo de este proceso es el de que a partir del conocimiento adquirido, es decir, recopilado, seleccionado, estructurado y almacenado, este sea enriquecido con un valor añadido, producto del procesamiento del mismo. En este proceso es fundamental la calidad del individuo que va a estar trabajando en el proyecto, pues este es el punto donde la creatividad es la clave del éxito y se muestra el potencial innovador del proyecto y a su vez del polo productivo.
- **Integración:** El conocimiento enriquecido no debe quedar en su fuente sino que debe ser puesto a disposición de todo el proyecto por lo que debe ser integrado al ya existente. Por eso el objetivo de

este componente es la combinación del nuevo conocimiento, adquirido y enriquecido al existente en el proyecto.

- **Socialización:** Este es el proceso en que mas interés se tiene en el polo, pues es el que divulga y disemina el conocimiento de un proyecto dentro de si, a través de todo el polo de Realidad Virtual y fuera de este. De esta manera se facilita y enriquece el proceso de adquisición de información a todos los integrantes de los proyectos del polo productivo. La Socialización puede ser lograda a través de foros de discusión, eventos, intranet, listas de correos, entre otros.
- **Retroalimentación:** Este componente no es más que el proceso de adquisición del conocimiento que ha sido producido por el proyecto y que ha sido enriquecido en el proceso de socialización. Tiene como objetivo obtener esa información para mejorar o modificar diversos aspectos del funcionamiento del polo de Realidad virtual. La realimentación tiene que ser bidireccional de modo que la mejora continua sea posible.

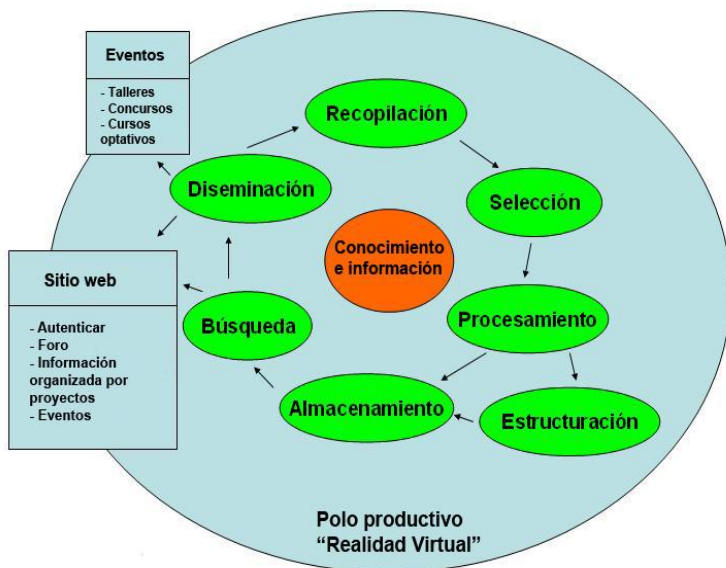


Figura 13: Cómo se maneja el conocimiento y la información dentro del polo

1.6 Conclusiones parciales

La aparición y creciente importancia del conocimiento como un nuevo factor de producción hace que el desarrollo de tecnologías, metodologías y estrategias para su medición, creación y difusión se convierta en una de las principales prioridades de las organizaciones en la sociedad del conocimiento.

En el modelo propuesto la gestión del conocimiento primero se lleva a cabo en el interior de cada proyecto a través de un ciclo de procesos y luego, como parte de ese ciclo, se socializa no solo a los otros proyectos, sino también al entorno general del polo y se pueden incluir otros polos productivos, utilizando los mismos mecanismos. Cada modelo tiene sus particularidades y sus ventajas, pero el modelo propuesto se acomoda perfectamente a las necesidades del polo Realidad virtual de la facultad 5 y a lo que se desea lograr en dicho polo.

CAPÍTULO 2: TECNOLOGÍAS Y TÉCNICAS UTILIZADAS EN EL DESARROLLO DE APLICACIONES WEB

2.1 Introducción

Actualmente la web ha evolucionado y se desarrollan aplicaciones que permiten explotar y utilizar todas las ventajas que nos brinda. Muchas empresas, instituciones y demás han comenzado a migrar sus servicios y aplicaciones al entorno web para permitir a sus usuarios y clientes el acceso a ellas desde Internet. En este capítulo se hace un análisis de las tecnologías a utilizar para la implementación del software. Se enfatiza en las conocidas como software libre y código abierto, pues la facultad 5 esta migrando hacia Linux y desarrollando la utilización del software libre. Además se justifica la propuesta de las herramientas escogidas.

2.2 Software Libre y Código Abierto.

El software libre es un asunto de calidad no de precio. Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Parte de un concepto creado por Richard Stallman y se basa en 4 libertades de los usuarios del software:

- Libertad de usar el programa, con cualquier propósito (Libertad 0).
- Libertad de estudiar como funciona el programa y adaptarlo a tus necesidades (Libertad 1).
- Libertad de distribuir copias (Libertad 2).
- Libertad de mejorar el programa y hacer públicas las mejoras (Libertad 3).

Un programa es software libre si cumple con todas estas libertades para con los usuarios.

Existen buenas razones como para decir que el software libre es más confiable que el software propietario. Una de las razones es que el software libre consigue involucrar a toda la comunidad para que trabaje unida para arreglar problemas, no solo informar acerca de ellos. Otra de las razones que se exponen es que los desarrolladores se preocupan realmente por la fiabilidad, se compite por una reputación y no de forma comercial. (7)

Muchos son las personas que confunden software libre con Open Source o código abierto. Open Source significa el mero acceso al código fuente. Los términos para la distribución del Software de Código Fuente Abierto tienen que cumplir los siguientes criterios:

1. Redistribución libre: la licencia no debe impedir la venta o el ofrecimiento del software como componente de una distribución de software agregado. La licencia no debe requerir el pago de los derechos de autor u otra tasa por dicha venta.
2. Código fuente: el programa tiene que incluir el código fuente y tiene que permitir la distribución tanto en código fuente como en forma compilada. El código fuente deliberadamente ofuscado no está permitido.
3. Trabajos derivados: tiene que permitir modificaciones y trabajos derivados, y debe permitir que ellos sean distribuidos bajo los mismo términos que la licencia del software original.
4. Integridad del código fuente del autor: la licencia tiene que permitir explícitamente la distribución del software construido a partir del código fuente modificado. La licencia puede requerir que los trabajos derivados tengan un nombre distinto o un número de versión distinto al del software original.
5. No a la discriminación de personas o grupos: la licencia no tiene que discriminar a ninguna persona o grupo de personas.
6. No a la discriminación de campos laborales: la licencia no tiene que restringir a nadie que haga uso del programa de un campo laboral específico.
7. Distribución de la licencia: los derechos adjuntos al programa tienen que aplicarse a todos aquellos que reciben el programa sin la necesidad de ejecutar una licencia adicional para estas partes.
8. La licencia no tiene que ser específica de un producto: los derechos adjuntos al programa no tienen que depender de que el programa forme parte de una distribución particular del software.
9. La licencia no tiene que restringir a otro software: la licencia no tiene que colocar restricciones en otro software que es distribuido junto con el software licenciado.
10. La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

2.3 Aplicaciones Web

Una página web es un documento de la World Wide Web (el cual intenta ser estandarizado por el W3C), normalmente en formato HTML o XHTML. Un Sitio Web es un conjunto de páginas Web

relacionadas entre sí que tratan un tema en común y cuyo contenido es visualizado en un navegador web a través de una petición hecha a un servidor web mediante la arquitectura cliente servidor.

La arquitectura cliente/servidor es una arquitectura de red que separa el cliente del servidor. El cliente es el encargado de iniciar un requerimiento de servicio. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente. El servidor proporciona un servicio al cliente y devuelve los resultados. Para que el cliente y el servidor se puedan comunicar se necesita de una infraestructura de comunicación que proporcione los mecanismos básicos de direccionamiento y transporte.

Una aplicación web es un sistema Web (servidor web, navegador, HTTP, red) donde la entrada de los datos por parte del usuario afecta el estado del negocio. Esta definición permite exponer que de no haber lógica de negocio en el servidor no puede llamarse aplicación web a dicho sitio web. (8)

Las principales características de las aplicaciones web son:

- Se accede a ella mediante una red y utilizando un navegador web.
- Se aloja en un servidor
- El cliente puede tener instalado cualquier sistema operativo
- No se requiere instalación o actualización de la aplicación, las mismas son transparentes para el usuario.
- La lógica del negocio se ejecuta completamente en el servidor.

Las aplicaciones web poseen varias ventajas, entre ellas se encuentran las siguientes:

1. Son fáciles y poco costosas de desarrollar, solo se necesita un computador con una navegador web y conexión a Internet o a una Intranet.
2. El costo de mantenimiento de las aplicaciones web son ligeras, tan pronto como la aplicación en el servidor sea actualizada los clientes podrán contar con la ultima versión.
3. Puede acceder a ella desde cualquier sistema operativo y si la aplicación es correctamente desarrollada funcionará de igual manera en cualquier navegador existente.
4. Permite la posesión del almacenamiento de los datos de una forma centralizada.
5. Permite que múltiples usuarios utilicen el sistema a la misma vez.

2.4 Servidores Web

Un servidor web es un servidor que atiende pedidos siguiendo las especificaciones del protocolo HTTP o HTTPS (la versión segura, cifrada y autenticada de HTTP). Un servidor web ejecuta de forma cíclica los siguientes pasos:

1. Espera peticiones en el puerto TCP asignado (el estándar para HTTP es el 80).
2. Recibe petición.
3. Busca el recurso en la cadena de la petición.
4. Envía el recurso por la misma conexión por donde ha recibido la petición.
5. Vuelve al segundo punto.

Un servidor web debe cumplir con aspectos como: servicio de ficheros estáticos, seguridad y autenticación, contenido dinámico. Existen otros aspectos que algunos de los servidores existentes también poseen como son: servidores virtuales, prestaciones extras, proxy servers y protocolos adicionales. (8)

En la actualidad existen servidores web de código libre entre los que podemos encontrar el AOLServer, Roxen, thttpd, Jetty y Apache.

2.4.1 Apache

En 1995 el servidor web de NCSA (National Center for Super Computer Applications) era el más utilizado. Sin embargo, el principal desarrollador de servidores web de la NCSA abandonó el centro y el proyecto comenzó a bloquearse. Mientras, las personas que utilizaban este servidor comenzaron a intercambiar paquetes y de esa forma nace el Grupo Apache. Hoy en día es uno de los servidores más utilizados a nivel mundial. La versión 1.0 se estrenó el 1ro de diciembre de 1995.

Apache es un servidor web de código libre robusto. El proyecto está dirigido y controlado por el Grupo Apache, que son un grupo de personas voluntarias de todo el mundo que planifican y desarrollan el servidor y la documentación relacionada. Además de este grupo, cientos de personas han contribuido con el proyecto. (9)

Apache muestra una serie de características por las cuales ha sido uno de los principales servidores web utilizados. A continuación se mencionan algunas características de este servidor web:

- Funciona en la mayoría de las versiones Unix, en Windows y en otros sistemas operativos.
- Elaborado índice de directorios.
- Directorio de alias.
- Negociación de contenidos.
- Informe de errores HTTP configurables.
- Gestión de recursos para procesos hijos.
- Integración de imágenes del lado del servidor.
- Reescritura de las URL.
- Comprobación de la ortografía de las URL.
- Soporte del último protocolo HTTP 1.1.
- Soporte para CGI.
- Soporte de host virtuales.
- Soporte de autenticación HTTP.
- Perl integrado.
- Soporte de scripts PHP.
- Soporte de servlets de Java.
- Servidor proxy integrado.
- Soporte de Secured Socket Layer (SSL).

Hoy en día Apache se encuentra en la versión 2.2. Esta versión hace de Apache una solución web más flexible, transportable y escalable.

2.5 Tecnologías y técnicas utilizadas en el desarrollo de aplicaciones web.

En computación, un programa es una secuencia de instrucciones que permiten a un ordenador procesar información conocida como datos de entrada para producir una información de salida o resultados. Esas instrucciones pertenecen a un lenguaje de programación determinado. Un lenguaje de programación es una construcción mental del ser humano para expresar programas.

2.5.1 Tecnologías del lado del Cliente

- Scripts del lado del cliente (Java script, JScript, Vbscript): lenguajes de programación interpretados por el navegador que se utiliza para controlar su apariencia y manipular los eventos que ocurran en la ventana.
- Document Object Model (DOM): Conjunto de objetos y colecciones que permiten el acceso desde varios lenguajes, como Java script y Java, a cada uno de los elementos que forman parte del documento.
- Cascade Style Sheets (CSS): Tecnología que permite controlar la presentación de los documentos en la web estableciendo un formato uniforme para todo el sitio.

2.5.1.1 Ajax

Ajax es un acrónimo de Asynchronous Javascript + XML. Surge por primera vez con la publicación del artículo “Ajax: A New Approach to Web Applications” publicado por Jesse Garrett el 18 de febrero del 2005.

Este artículo expresa que:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan en forma autónoma y que se unen de formas nuevas y sorprendentes.”

Las tecnologías que conforman Ajax son:

- XHTML y CSS, para crear la presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y manipulación de la información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- Javascript, para unir todas las demás tecnologías.

En las aplicaciones web tradicionales, las acciones del usuario en la página desencadenan llamadas al servidor. Una vez que es procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador.

Esta técnica funciona de manera correcta pero al realizar continuas peticiones al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados.

Ajax permite mejorar la interacción del usuario con la aplicación, evitando la recargas constantes de la página, pues el intercambio de la información con el servidor se realiza en un segundo plano. Estas recargas constantes se eliminan mediante la creación de un elemento intermedio entre el usuario y el servidor. Esta capa mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. (10)

2.5.2 Tecnologías del lado del Servidor

El lado del servidor está formado por un programa o script que es ejecutado por el servidor web, y cuya salida se envía al navegador del cliente.

Dentro de las tecnologías del lado del servidor se encuentran:

- CGI: estándar que especifica como se va a comunicar un servidor HTTP con una aplicación ejecutándose en la misma computadora.
- SERVLETS: Tecnología Java del lado del servidor. Presenta todas las características del lenguaje Java, es multiplataforma, orientado a objetos y basado en un fuerte modelo de seguridad.
- Scripts del lado del servidor (ASP, PERL, PHP, JSP): lenguajes de scripts que son interpretados en el servidor web.

2.5.2.1 PHP: Hypertext Preprocessor (PHP)

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML.

La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

PHP fue concebido en otoño de 1994 por Rasmus Lerdorf. Las primeras versiones no distribuidas al público fueron usadas en un sus páginas web para mantener un control sobre quien consultaba su currículum. La primera versión disponible para el público a principios de 1995 fue conocida como "Herramientas para paginas web personales" (Personal Home Page Tools).

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP. (11)

El PHP se encuentra en su versión 5. En la actualidad existen una gran cantidad de *frameworks* que están desarrollados en PHP y que permiten un desarrollo rápido y eficiente de aplicaciones, pues encapsulan operaciones complejas en instrucciones sencillas. Ejemplos de estos frameworks son: Prado, CakePHP y Symfony.

Ventajas de PHP

- 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.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una muy buena documentación en su página oficial.
- Es Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

2.5.2.2 Symfony

Symfony es un completo 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. (12)

Características de Symfony

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

2.5.3 Sistemas Gestores de Bases de Datos (SGBD).

Una base de datos es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. Puede considerarse una colección de datos en el tiempo.

El conjunto de programas que permite a los usuarios crear y mantener una base de datos es denominado Sistema Gestor de Bases de Datos (SGBD). El SGBD es un software con el propósito de definir, construir y manipular la base de datos para diversas aplicaciones.

Las bases de datos pueden ser organizadas de diferentes maneras, pero existen un conjunto de objetivos que deben cumplir todos los SGBD para facilitar el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios. Estos objetivos son:

- Independencia de los datos y los programas aplicación: es la inmunidad de las aplicaciones a los cambios en la estructura de almacenamiento y en la estrategia de acceso y constituye el objetivo fundamental de los SGBD.
- Minimización de la redundancia: se trata de minimizar la redundancia de los datos, es decir, disminuirla no eliminarla. Se trata de eliminar la redundancia superflua.
- Integración y sincronización de las bases de datos: garantiza una respuesta a los requerimientos de diferentes aspectos de los mismos datos por diferentes usuarios, de forma que, aunque el sistema almacene la información con cierta estructura y cierto tipo de representación, debe garantizar entregar al programa de aplicación datos que solicita y en la forma en que lo solicita.
- Integridad de los datos: garantiza la no contradicción entre los datos almacenados de modo que, en cualquier momento del tiempo, sean correctos, es decir, que no detecte inconsistencia entre ellos.
- Seguridad y recuperación: garantiza el acceso autorizado a los datos y dispone métodos que garanticen la restauración de las bases de datos al producirse alguna falla técnica, interrupción de la energía eléctrica, etc.
- Facilidad de manipulación de la información: el SGBD debe contar con la capacidad de una búsqueda rápida por diferentes criterios, permitir que los usuarios planteen demandas de una forma simple, aislándolo de las complejidades de tratamiento de los ficheros y del direccionado de datos.
- Control centralizado: permite controlar de manera sistemática y única de datos que se almacenan en la base de datos, así como el acceso a ella.

Existen otros objetivos que deben cumplir los SGBD que en muchos casos dependen de las condiciones o requerimientos específicos de utilización del sistema. Entre los SGBD conocidos se encuentran MySQL, Oracle, Microsoft SQL Server y PostgreSQL.

2.5.3.1 PostgreSQL

En 1986, en la Universidad de California en Berkeley, el profesor y también reconocido experto en tecnologías de bases de datos, Michael Stonebraker, decide construir un mejor SGBD. De esta forma surge POSTGRES, comenzando a ganar en popularidad en los siguientes 8 años. El proyecto fue oficialmente terminado en la versión 4.2, sin embargo, fue retomado por Andrew Yu y Jolly Chen

quienes le adicionan un analizador gramatical de SQL y lo lanzan como Postgres95. En las posteriores versiones se le comienza a llamar PostgreSQL haciendo alusión al proyecto inicial POSTGRES y a sus nuevas capacidades SQL. (13)

Actualmente PostgreSQL se encuentra en su versión 8.3 y es uno de los proyectos *Open-Source* más populares de Internet. Algunas de las más grandes y famosas organizaciones utilizan PostgreSQL (*Apple Computers, BASF, Cisco Systems y The World Health Organization*).

Los principales atributos que hacen a PostgreSQL una buena opción para ser escogido a la hora de desarrollar una aplicación son:

- *Integridad de Datos:* si algún nuevo aspecto compromete la integridad de los datos no es incluida hasta que no cumpla con este requisito.
- *Altamente escalable:* PostgreSQL evita el problema que pueden crear una gran cantidad de usuarios conectados a él, leyendo y escribiendo simultáneamente en la base de datos.
- *Características Completas:* soporta un número elevado de lenguajes incluyendo C, SQL, PL/pgSQL y PL/Perl. No importa lo que quieras hacer, PostgreSQL posee una forma de hacerlo, y si no, probablemente tu mismo le puedas adicionar alguna funcionalidad.
- *Extensible:* te provee de herramientas como tipos de datos, dominios y operadores para añadir tus propias extensiones.
- *Soporte para plataformas:* PostgreSQL siempre ha avanzado para tratar de ser lo más funcional posible a través de diferentes plataformas. Es usado en docenas de plataformas Unix y Linux, desde sistemas populares como FreeBSD y Red Hat Linux hasta oscuras plataformas como QNX y BeOS. También lo utilizan plataformas de juegos como Sony PlayStation 2 y Nintendo GameCube. Desde la versión 8.0.0 puede ser ejecutado en Windows.
- *Opciones de seguridad flexibles:* posee un extenso conjunto de protocolos de seguridad y opciones de configuración como rasgos internos, que ayudan a tener el control sobre quienes y que accede a los datos dentro de la base de datos.

2.6 El Proceso Unificado de Desarrollo (RUP)

Hoy en día, se requiere la creación de sistemas más grandes y complejos en un tiempo mínimo. Sin embargo, se necesitan métodos que permitan cumplir con estos requerimientos, una forma coordinada para trabajar, un procedimiento que integre las múltiples facetas del desarrollo. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. RUP es un proceso de desarrollo de software.

RUP es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (14)

RUP se basa en componentes y utiliza UML para preparar todos los esquemas de un software. UML es una parte esencial de RUP. Los aspectos que hacen de RUP un proceso de desarrollo único son:

- Dirigido por casos de uso: un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Todos los casos de uso juntos constituyen el modelo de casos de uso. Ellos guían el diseño, implementación y prueba del sistema, es decir, el proceso de desarrollo. Dirigido por casos de uso significa que el proceso de desarrollo sigue un hilo que avanza a través de una serie de flujos y parte de los casos de uso.
- Centrado en la arquitectura: el concepto de arquitectura de software incluye aspectos estáticos y dinámicos más significativos del sistema. Surge de las necesidades de la empresa y se refleja en los casos de uso. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando de lado los detalles. Debe haber integración entre los casos de uso y la arquitectura. La arquitectura debe permitir el desarrollo de todos los casos de uso requeridos y debe diseñarse para permitir que el sistema evolucione, no solo en el desarrollo inicial, sino a lo largo de las futuras generaciones.
- Desarrollo iterativo e incremental: el desarrollo de productos software puede durar meses o más, por lo que es práctico dividir el trabajo en partes más pequeñas. Cada parte es una iteración que resulta ser un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos al crecimiento del producto. Para una mayor efectividad estas

iteraciones deben ser controladas, es decir, deben seleccionarse y ejecutarse de una forma planificada.

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema y al concluir cada ciclo se concluye una versión del producto. Cada ciclo cuenta con 4 fases: inicio, elaboración, construcción y transición. En la fase de inicio se desarrolla una descripción del producto final y se presenta el análisis del negocio para el producto. Durante la fase de elaboración se especifican la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. En la fase de elaboración se crea el producto. La fase de transición cubre el período donde el producto se convierte en versión beta.

Los principales elementos de RUP son:

- Trabajadores: definen el comportamiento y responsabilidades de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades: una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por las actividades.
- Flujo de actividades: secuencia de actividades realizadas por los trabajadores y que produce un resultado de valor observable.

2.7 El Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas.

UML fue desarrollado en un esfuerzo para simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que habían surgido. Fue creado por Grady Booch, James Rumbaugh e Ivar Jacobson.

UML combina conceptos aceptados por muchos métodos orientados a objetos, seleccionando una definición clara para cada concepto, así como una notación y una terminología. No tiene saltos ni discontinuidades desde los requisitos a la implantación, está pensado para la mayoría de los dominios de la aplicación y para ser usado en sistemas desarrollados en varios lenguajes de implementación y plataformas. UML es un lenguaje que permite modelar sistemas con tecnología orientada a objetos, no es una guía para realizar el análisis y diseño orientado a objetos.

UML fue desarrollado con varios objetivos. El más importante es que UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores, no tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Otro de los objetivos finales de UML era ser tan simple como fuera posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.

UML está compuesto por una variedad de elementos gráficos que se combinan para formar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. (15)

2.8 Herramienta Case

Las herramientas Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador (CASE) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Las herramientas CASE se han desarrollado como una de las soluciones para afrontar los problemas de una calidad de software pobre y una documentación inadecuada. Cuanto mayor es un proyecto, más importante es el uso de tecnología CASE.

A principios de los años 80 se comenzaron a desarrollar las Herramientas CASE, con una documentación y un diagramado asistido por computadoras. Se siguió desarrollando durante la década de los 80, hasta llegar a los años 90 donde se aprecia una conducción inteligente de la metodología, tiene una interfaz amigable con el usuario y la reutilización como metodología de desarrollo.

Objetivos de las Herramientas Cases:

- Las herramientas CASE aseguran que se alcanzan las tres C's:

- Consistencia
- Completitud
- Cumplimiento de los estándares.
- Mejora de la productividad y de la calidad, mediante un entorno interactivo.
- Aceleración del proceso de desarrollo, favoreciendo el proceso de prototipo (espiral).
- Automatizar e integrar las tareas de las distintas etapas del ciclo de vida.
- Asistencia en la gestión de proyectos software.
- Mejora de la calidad del software (automatización comprobación de errores).
- Automatizar la generación de documentación.

2.8.1 Visual Paradigm

Visual Paradigm es una poderosa herramienta CASE que utiliza UML para el modelado. Es la herramienta por excelencia utilizada en un ambiente de software libre. Las principales características del Visual Paradigm son:

- Licencia gratuita y comercial.
- Producto de calidad.
- Soporta aplicaciones web.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

También ofrece:

- Un entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, para cada necesidad.

- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, actividad, estado, entre otros. Identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, y, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. (16)

2.9 Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado (IDE) es un programa compuesto por un conjunto de herramientas para un programador.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes, proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, Visual Basic, etc.

Un IDE puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder funcionar con varios lenguajes de programación. Este es el caso de Eclipse, que mediante pluggins se le puede añadir soporte de lenguajes adicionales

2.9.1 Eclipse

Es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. (17)

Características

La versión actual de Eclipse dispone de las siguientes características:

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

Asimismo, a través de "plugins" libremente disponibles es posible añadir:

- Control de versiones con Subversion, vía Subclipse.
- Integración con Hibernate, vía Hibernate Tools.

2.10 Justificación de la propuesta de las herramientas

Como resultado del estudio realizado para la elaboración de la propuesta de solución se desarrollará una aplicación web, lo cual posibilita que el sistema pueda ser utilizado por todos los integrantes del polo productivo de Realidad Virtual que necesiten acceder al conocimiento generado en el polo, mientras se posea una computadora con un navegador web y conexión de red.

Para el desarrollo de la aplicación se utilizará como lenguaje de programación del lado del cliente PHP 5 con el framework Symfony 1.2.7 para un desarrollo rápido y eficiente, y como SGBD se propone PostgreSQL 8.3. La aplicación estará montada sobre un servidor web Apache 2.2.

Se escogió como lenguaje de programación el PHP 5 debido a la extensa documentación existente sobre él. Es un lenguaje fácil de aprender y mantiene una buena colaboración con el servidor Apache,

además de que permite el paradigma de programación orientado a objetos. Es uno de los lenguajes scripts del lado del servidor libre de licencias para la creación de aplicaciones y productos. Se eligió usar el framework Symfony debido a que tiene buena documentación en español, conjuntamente a su madurez y flexibilidad para adaptarse a diferentes problemas a solucionar. Posee además una implementación de arquitectura web y un Mapeador Relacional de Objetos (ORM) que le facilitan al desarrollador su trabajo.

Como SGBD se optó por el PostgreSQL 8.3 por ser un sistema Open Source robusto que posee una buena seguridad e integridad de los datos y altamente escalable. PostgreSQL posee un plugin PDT que permite programar en PHP.

Para el desarrollo del lado del cliente se utilizara la técnica de Ajax para lograr un menor intercambio de información con el servidor, de esta manera se aumenta la interacción, la velocidad y facilidad de uso.

La metodología de desarrollo que se escoge es RUP, pues permite dirigir el sistema por casos de uso centrándonos en la arquitectura bajo un desarrollo iterativo e incremental. RUP permite llevar una documentación adecuada de todas las fases y flujos de trabajo del producto para futuras versiones de los productos a desarrollar. Además es recomendable para proyectos grandes donde se genere un gran nivel de documentación.

2.11 Conclusiones parciales

Se realizó el estudio de las tecnologías a emplear en el desarrollo de la solución propuesta. Se fundamentó la elección de las tecnologías a utilizar en la implementación de la aplicación como son: el lenguaje de programación, sistema gestor de bases de datos y la metodología de desarrollo. Se ha hecho alusión también a las herramientas escogidas para el desarrollo de la aplicación web.

CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA

3.1 Introducción

En todo proyecto de desarrollo de software es necesario cuando se va a comenzar a trabajar, definir como se desenvuelven hasta ese momento los procesos que se desean informatizar. Así mismo de importante es precisar qué se desea lograr y que es específicamente lo que se quiere obtener.

En el presente capítulo se describen las características del sistema, para esto se estudiarán los procesos del negocio relacionados con el objeto de estudio. A partir de un análisis realizado se puede percibir que debido a la poca estructuración de los procesos se necesita puntualizar conceptos que se pueden agrupar en un Modelo de dominio para que sea más entendible el contexto en el que se ubica el sistema. Además se especifican los requerimientos funcionales y no funcionales de la aplicación web a diseñar y se conforma el diagrama de casos de uso del sistema.

3.2 Propuesta del sistema

Anteriormente se mostró que la gestión del conocimiento constituye el objeto de estudio de este trabajo de diploma, y que dentro de este objeto como campo de acción se encuentra la gestión del conocimiento en el polo productivo de Realidad Virtual.

Este trabajo de diploma se centra en la propuesta de un modelo para la gestión del conocimiento a través de una aplicación web, que sirva como herramienta de apoyo en el proceso de toma de decisiones. Con una serie de funcionalidades responde a la necesidad de que no se pierda la información que se genera en el polo productivo de Realidad Virtual y que todos los integrantes del polo puedan tener acceso a ese conocimiento producido. Conjuntamente permite el desarrollo de una aplicación para el manejo de la información de forma centralizada y que se logre reutilizar dicha información; además de contribuir a que los productos de software salgan con una mejor calidad.

La plataforma manejará principalmente el trabajo con información, mediante un foro, además se podrá subir la información, modificarla, descargarla, eliminarla, entre otras operaciones.

3.3 Modelo de dominio

Un Modelo de Dominio o Modelo Conceptual captura los tipos más importantes de objetos en el contexto del problema que se está analizando. Los objetos del dominio representan las entidades que existen o los

eventos que suceden en el entorno en el que trabaja el sistema. Las clases del dominio aparecen de diferentes formas: como objetos del entorno, objetos del mundo real, ó, como sucesos que ocurrirán o han ocurrido.

Después de un profundo análisis se llegó a la conclusión de que existe una poca estructuración de los procesos del negocio y por tanto se hace necesario realizar un modelo de dominio. A continuación se definen una serie de conceptos para hacer más entendible el problema.

3.3.1 Definición de objetos del dominio

Polo productivo: Unidad organizativa dentro de la cual se enmarca el contexto del sistema.

Área temática: Módulo organizativo sobre un tema del cual se trabaja en el proyecto productivo.

Proyecto productivo: Elemento organizativo a través del cual se lleva a cabo el desarrollo del software.

Jefe de polo: Persona encargada de dirigir todos los eventos y acciones que se lleven a cabo en el polo productivo, además de regir a los integrantes de los proyectos.

Jefe de área temática: Persona encargada de manejar el área temática.

Líder de proyecto: Persona encargada de guiar el trabajo y las actividades que se llevan a cabo dentro del proyecto.

Asesor de la calidad: Persona encargada de revisar todos los documentos que se generan en el polo.

Profesor: Profesores vinculados al proyecto productivo y que interactúan con el repositorio y los documentos.

Estudiante: Estudiantes escogidos que pertenecen al proyecto productivo y que interactúan con el repositorio y los documentos.

Expediente de proyecto: Documentación asociada al proyecto productivo.

Repositorio: Lugar donde se almacenan los documentos generados.

Documentos: Documentos que se generan en el proyecto productivo.

3.3.2 Representación del Modelo de dominio

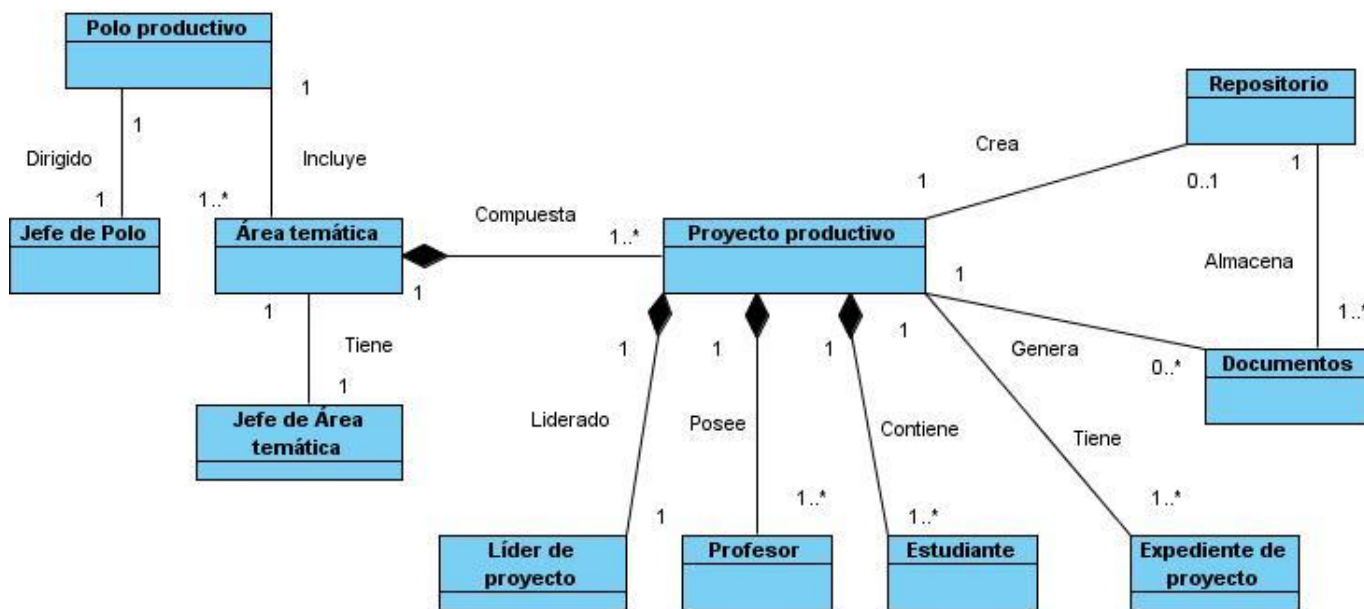


Figura 14: Modelo de dominio.

3.4 Requerimientos de Software

La correcta especificación de requisitos determina en gran medida el éxito de un proyecto de desarrollo de software. A continuación se muestra la especificación de los requisitos tanto funcionales como no funcionales de la plataforma web que se propone.

3.4.1 Requerimientos funcionales

Los requerimientos funcionales son las capacidades o condiciones que el sistema debe cumplir. En este caso se debe:

RF1 Autenticar usuario

RF2 Contener un foro

- RF 2.1 Participar en el foro
- RF 2.2 Crear nuevo tema de discusión

RF3 Gestionar información

- RF 3.1 Subir información
- RF 3.2 Bajar información
- RF 3.3 Modificar Información

RF 3.4 Eliminar Información

RF4 Gestionar eventos

RF4.1 Insertar eventos

RF4.2 Modificar eventos

RF4.3 Eliminar eventos

RF5 Mostrar información

RF6 Mostrar eventos

RF7 Gestionar proyecto

RF 7.1 Insertar proyecto

RF 7.2 Modificar proyecto

RF 7.3 Eliminar proyecto

RF8 Mostrar proyecto

3.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido o confiable.

Requerimientos de Software

- **PC Cliente:** Puede poseer Sistemas Operativos Linux (Debian, Ubuntu) ó Windows (98, Server, XP). Se recomienda que los navegadores web sean Internet Explorer versión 6.0 o superiores y Mozilla Firefox con versiones superiores a la 1.5.
- **PC Servidor:** Debe contar con Sistema Operativo Linux, Servidor de aplicaciones Apache, Sistema Gestor de Base de Datos PostgreSQL.

Requerimientos de Hardware

- **PC Cliente:** Debe contar con un microprocesador con una velocidad de procesamiento superior a 1GHz, 128 MB o más de memoria RAM y una tarjeta de red.

- **PC Servidor:** Debe contar con un microprocesador con una velocidad de procesamiento superior a 3 GHz, 512 MB o más de memoria RAM y una tarjeta de red.

Requerimientos de apariencia o interfaz externa

- La interfaz debe ser sencilla y de fácil entendimiento para el usuario.
- Los colores deben ser representativos de la UCI y la facultad 5 logrando el equilibrio entre estos para obtener una interfaz agradable para el usuario.
- Diseño encuadrado para resoluciones de 800X600 aunque también preparado para otro tipo de resoluciones.

Requerimientos de Seguridad

- Identificación por el dominio del usuario antes de realizar cualquier acción en el sistema.
- Protección contra acciones no autorizadas que pueda afectar la disponibilidad, integridad y confiabilidad de la información.

Requerimientos de Usabilidad

- El sistema podrá ser utilizado por estudiantes y profesores que se encuentren en el dominio UCI y que posean conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Portabilidad

- Sistema multiplataforma.

Confiabilidad y disponibilidad

- El sistema debe estar disponible durante las 24 horas del día.

3.5 Definición de los casos de uso

3.5.1 Definición de los actores

Tabla 2: Definición de los actores del sistema.

Actores	Justificación
---------	---------------

Usuario (Líder de Proyecto, Asesor de la calidad, Profesor, Estudiante)	Actor general que interactúa con el sistema. Tiene acceso a todas las secciones de la aplicación web, menos a Gestionar evento y proyecto.
Jefe de Polo	Es un usuario que tiene acceso a todas las secciones de la aplicación menos a Gestionar proyecto, y es el único que puede Gestionar evento.
Jefe de Área temática	Es un usuario que tiene acceso a todas las secciones del sitio web menos a Gestionar eventos, además es el único que puede Gestionar proyecto.

3.5.2 Diagrama de casos de uso

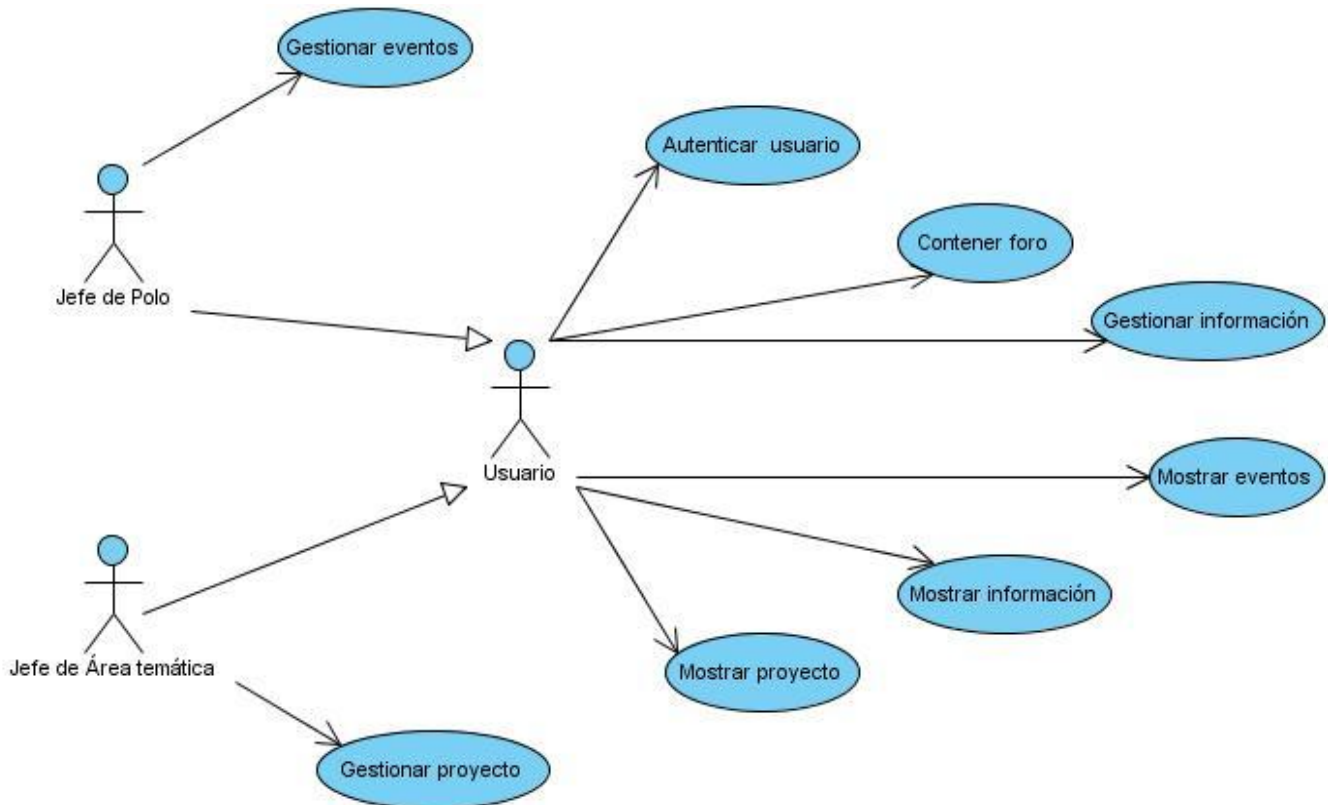


Figura 15: Diagrama de casos de uso del sistema

3.5.3 Descripción detallada de los casos de uso

Tabla 3: Descripción detallada del caso de uso “Autenticar usuario”.

Caso de uso	Autenticar usuario	
Propósito	Que el usuario se autentique para poder entrar al sistema.	
Actores	Usuario	
Resumen	El caso de uso de inicia cuando el usuario intenta acceder al sistema y se autentica para hacerlo.	
Referencias		
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción autenticar usuario. 3. Introduce nombre de usuario y contraseña. 	<ol style="list-style-type: none"> 1. El sistema muestra el formulario a llenar. 4. Valida los datos. 4.1 Si los datos son correcto accede al sistema. 	
Flujo alternativo		
Acción del actor	Respuesta del sistema	
	4.2 Si los datos no son correctos muestra un mensaje de error.	

Tabla 4: Descripción detallada del caso de uso “Contener Foro”.

Caso de uso	Contener Foro
Propósito	Que el usuario pueda participar en el foro.
Actores	Usuario
Resumen	El caso de uso comienza cuando el usuario ya autenticado en el sistema selecciona la opción foro.
Referencias	RF1, RF 2.1, RF 2.2

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción foro. 3. Escoge una opción. <ol style="list-style-type: none"> a) En caso de querer participar en un tema: Sección “Participar en el foro”. b) En caso de querer crear un nuevo tema de discusión: Sección “Crear nuevo tema de discusión” 	<ol style="list-style-type: none"> 2. Muestra los temas de discusión y la opción crear un nuevo tema de discusión.
Sección “Participar en el foro”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona el tema en el que quiere participar. 3. Lee el tema de discusión escogido. 	<ol style="list-style-type: none"> 2. Visualiza la venta de discusión del tema.
Flujo alternativo	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 3.1 Participa en la discusión del tema escogido. 	<ol style="list-style-type: none"> 4. Guarda la opinión introducida. 5. Actualiza el tema en el foro.
Sección “Crear nuevo tema de discusión”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona crear un nuevo tema de discusión. 3. Llena los campos del formulario. 	<ol style="list-style-type: none"> 2. Muestra un formulario con los datos a llenar. 4. Guarda los datos en memoria. 5. Muestra el nuevo tema de discusión en el foro.

Tabla 5: Descripción detallada del caso de uso “Gestionar información”.

Caso de uso	Gestionar información	
Propósito	Subir, bajar, modificar y eliminar información.	
Actores	Usuario	
Resumen	El caso de uso comienza cuando el usuario autenticado desea gestionar información.	
Referencias	RF1, RF 3.1, RF 3.2, RF 3.3, RF 3.4	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Selecciona la opción gestionar información. 2. Escoge una opción. <ol style="list-style-type: none"> a) En caso de subir información: Sección “Subir información”. b) En caso de bajar información: Sección “Bajar información”. c) En caso de modificar información: Sección “Modificar información”. d) En caso de eliminar información: Sección “Eliminar información”. 	<ol style="list-style-type: none"> 2. Muestra las opciones subir, bajar, modificar y eliminar información. 	
Sección “Subir información”		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Selecciona la opción subir información. 3. Llena el formulario con la información asociada al documento y busca el documento que desea subir. 	<ol style="list-style-type: none"> 2. Muestra el formulario para realizar la operación solicitada (título del documento, autor, comentario, proyecto). 4. Guarda la información en la base de datos. 5. Muestra la información insertada. 	
Sección “Bajar información”		
Acción del actor	Respuesta del sistema	

1. Selecciona la opción bajar información. 3. Descarga la información deseada.	2. Muestra la información por proyectos.
Sección “Modificar información”	
Acción del actor	Respuesta del sistema
1. Selecciona la opción modificar información. 3. Escoge la información que desea modificar. 4. Modifica la información deseada y acepta la acción.	2. Muestra la información por proyectos. 5. Guarda la nueva información en la base de datos. 6. Muestra la información actualizada.
Sección “Eliminar información”	
Acción del actor	Respuesta del sistema
1. Selecciona la opción eliminar información. 3. Escoge la información que desea eliminar. 4. Elimina la información escogida.	2. Muestra la información por proyectos. 5. Actualiza la base de datos.

Tabla 6: Descripción detallada del caso de uso “Gestionar evento”.

Caso de uso	Gestionar evento
Propósito	Insertar, modificar y eliminar un evento
Actores	Jefe de polo
Resumen	El caso de uso inicia cuando el jefe de polo desea gestionar un evento.
Referencias	RF1,RF 4.1, RF 4.2, RF 4.3
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

<ol style="list-style-type: none"> 1. Selecciona la opción gestionar eventos. 3. Escoge una opción. a) En caso de insertar evento: Sección “Insertar evento” b) En caso de modificar evento: Sección “Modificar evento” c) En caso de eliminar evento: Sección “Eliminar evento” 	<ol style="list-style-type: none"> 1. Muestra las acciones insertar, modificar y eliminar evento.
Sección “Insertar evento”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción insertar evento. 3. Introduce los datos y selecciona insertar. 	<ol style="list-style-type: none"> 2. Muestra un formulario con los datos a llenar (Nombre del evento, Fecha, Lugar, Descripción y Categoría). 4. Guarda los datos en memoria. 5. Actualiza los eventos y los muestra.
Sección “Modificar evento”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción modificar evento. 3. Escoge el evento a modificar. 5. Modifica los datos. 	<ol style="list-style-type: none"> 2. Muestra todos los eventos existentes. 4. Muestra los datos del evento. 6. Guarda los cambios en memoria y muestra los eventos.
Sección “Eliminar evento”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción eliminar evento. 3. Escoge el evento a eliminar y da clic en eliminar. 	<ol style="list-style-type: none"> 2. Muestra todos los eventos existentes. 4. Actualiza los eventos y los muestra.

Tabla 7: Descripción detallada del caso de uso “Mostrar información”.

Caso de uso	Mostrar información	
Propósito	Visualizar la información.	
Actores	Usuario	
Resumen	El caso de uso se inicia cuando el Usuario selecciona la opción mostrar información.	
Referencias		
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Selecciona la opción mostrar información. 3. Escoge el proyecto para ver la información asociada a el. 	<ol style="list-style-type: none"> 2. Muestra los proyectos para que el usuario escoja el proyecto deseado. 4. Muestra la información contenida en el proyecto seleccionado. 	

Tabla 8: Descripción detallada del caso de uso “Mostrar eventos”.

Caso de uso	Mostrar eventos	
Propósito	Visualizar los eventos.	
Actores	Usuario	
Resumen	El caso de uso se inicia cuando el Usuario selecciona la opción mostrar eventos.	
Referencias		
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Selecciona la opción mostrar eventos. 3. Escoge la categoría de evento. 	<ol style="list-style-type: none"> 2. Muestra todos los eventos por categoría. 4. Muestra los eventos de la categoría escogida. 	

Tabla 9: Descripción detallada del caso de uso “Gestionar proyecto”.

Caso de uso	Gestionar proyecto	
Propósito	Insertar, modificar y eliminar un proyecto	
Actores	Jefe de Área temática	
Resumen	El caso de uso inicia cuando el jefe de área temática desea gestionar un proyecto.	
Referencias	RF1,RF 7.1, RF 7.2, RF 7.3	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
<ul style="list-style-type: none"> 1. Selecciona la opción gestionar proyecto. 3. Escoge una opción. d) En caso de insertar proyecto: Sección "Insertar proyecto" e) En caso de modificar proyecto: Sección "Modificar proyecto" f) En caso de eliminar proyecto: Sección "Eliminar proyecto" 	<ul style="list-style-type: none"> 2. Muestra las acciones insertar, modificar y eliminar proyecto. 	
Sección "Insertar proyecto"		
Acción del actor	Respuesta del sistema	
<ul style="list-style-type: none"> 1. Selecciona la opción insertar proyecto. 3. Introduce los datos y selecciona insertar. 	<ul style="list-style-type: none"> 2. Muestra un formulario con los datos a llenar (Nombre del proyecto, Área temática, líder). 4. Guarda los datos en memoria. 5. Actualiza los proyectos y los muestra. 	
Sección "Modificar proyecto"		
Acción del actor	Respuesta del sistema	
<ul style="list-style-type: none"> 1. Selecciona la opción modificar proyecto. 3. Escoge el proyecto a modificar. 5. Modifica los datos. 	<ul style="list-style-type: none"> 2. Muestra todos los proyectos existentes. 4. Muestra los datos del proyecto. 6. Guarda los cambios en memoria y 	

	muestra los proyectos.
Sección “Eliminar proyecto”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción eliminar proyecto. 3. Escoge el proyecto a eliminar y da clic en eliminar. 	<ol style="list-style-type: none"> 2. Muestra todos los proyectos existentes. 4. Actualiza los proyectos y los muestra.

Tabla 10: Descripción detallada del caso de uso “Mostrar proyecto”.

Caso de uso	Mostrar proyecto
Propósito	Visualizar los proyectos.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el Usuario selecciona la opción mostrar proyecto.
Referencias	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción mostrar proyecto. 	<ol style="list-style-type: none"> 2. Muestra todos los proyectos existentes.

3.6 Conclusiones parciales

En este capítulo se han obtenido resultados importantes para el posterior desarrollo de la aplicación y flujos de trabajo de la metodología RUP. Se han identificado conceptos del dominio ya que los procesos del negocio no estaban bien estructurados. Se definieron los requisitos funcionales y no funcionales encontrados en el levantamiento de requerimientos, con los cuales se pudieron identificar los casos de uso del sistema y realizar la descripción textual de los mismos.

CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA

4.1 Introducción

La realización del análisis y diseño ayudan a definir los aspectos lógicos del sistema. En un proyecto de desarrollo de software se puede omitir el análisis si se tienen bien definidos los procesos para desarrollar correctamente el diseño. En este capítulo se presentan los diagramas de clases del análisis de los diferentes casos de uso del sistema. Además se realiza el diseño de la aplicación a través de los diagramas de clases del diseño, los de secuencia y la descripción de las clases. También se efectúa el modelo entidad relación y la descripción de las tablas.

4.2 Modelo de Análisis

El modelo de análisis es una aproximación al modelo de diseño. Hay un refinamiento de los requisitos, sin embargo no se tiene en cuenta el lenguaje de programación que se va a utilizar en la construcción de la aplicación, debido a que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar como se implementará la solución.

4.2.1 Diagramas de clases del análisis

Los diagramas clases del análisis son utilizados para representar los conceptos en un dominio del problema. Representan las cosas del mundo real, no de la implementación automatizada de estas cosas. A continuación se representan las clases del análisis de los casos de uso Autenticar usuario, Contener foro y Gestionar información, el resto de los diagramas se encuentra en el anexo 1.



Figura 16: Diagrama de clases del análisis del caso de uso “Autenticar usuario”



Figura 17: Diagrama de clases del análisis del caso de uso “Contener Foro”

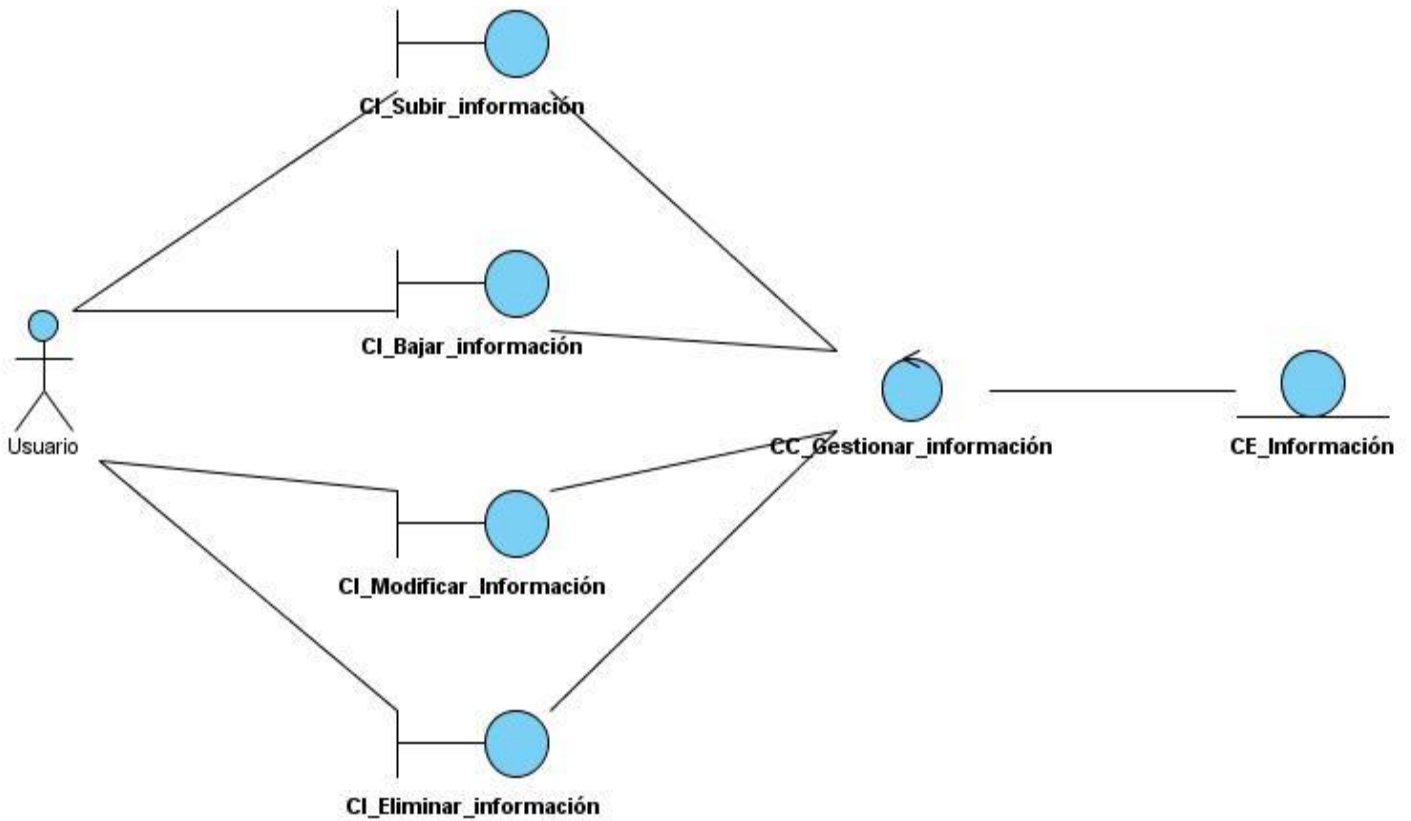


Figura 18: Diagrama de clases del análisis del caso de uso “Gestionar información”

4.3 Modelo de Diseño

En el modelo de diseño estructuramos el sistema y encontramos su forma, incluida la arquitectura, para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones.

4.3.1 Diagramas de clases del diseño

Para tener una visión general del diseño de la aplicación se realizan los diagramas de clases del diseño. Como se decidió usar Symfony como framework los diagramas de clases del diseño se realizaron según el patrón de diseño Modelo Vista Controlador (MVC) ya que el Symfony utiliza este patrón arquitectónico. A continuación se figuran los diagramas de clases de los casos de uso Autenticar usuario, Contener foro y Gestionar información; cada uno de ellos desglosados en sus respectivas secciones. El resto de los diagramas se encuentran en el anexo 2.

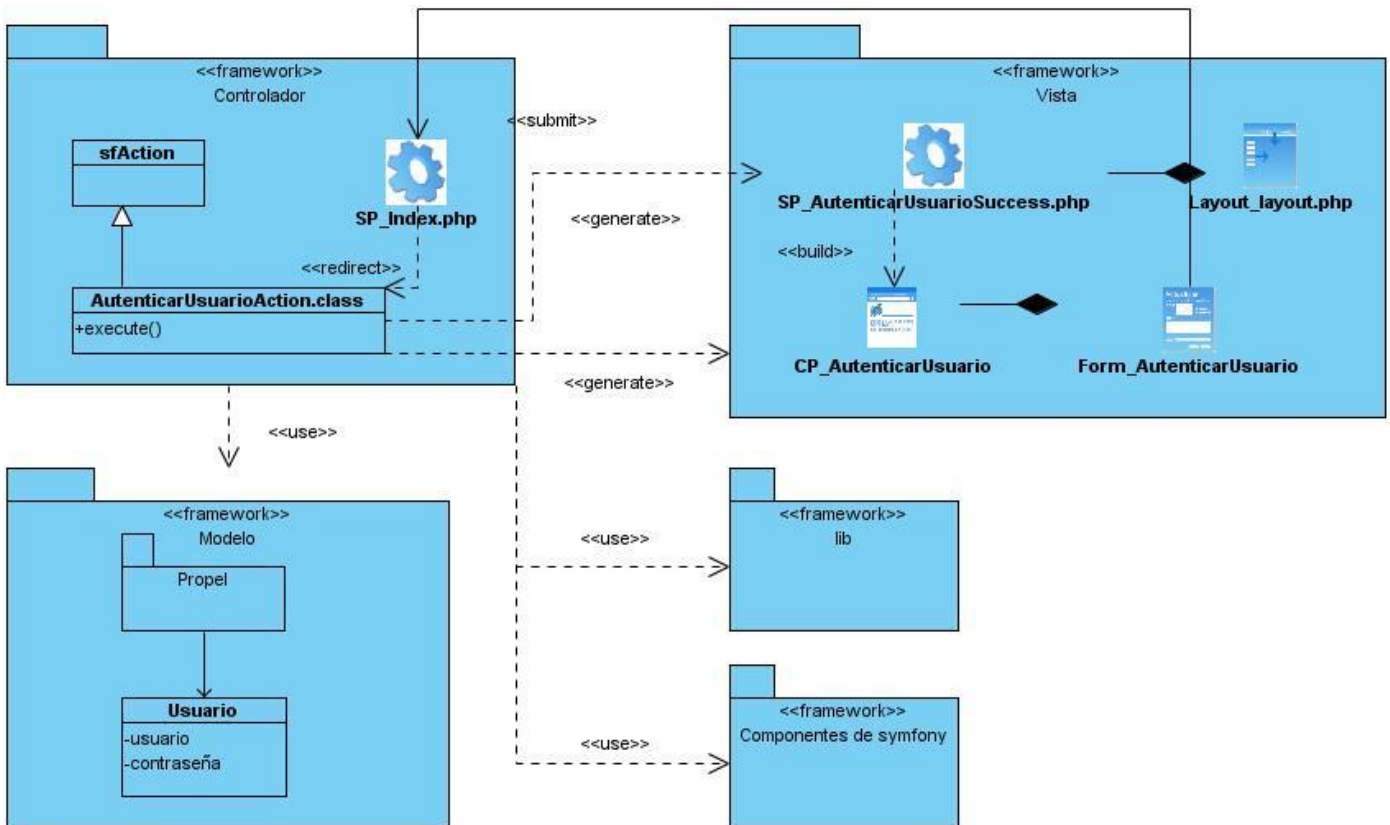


Figura 19: Diagrama de clases del diseño "Autenticar usuario"

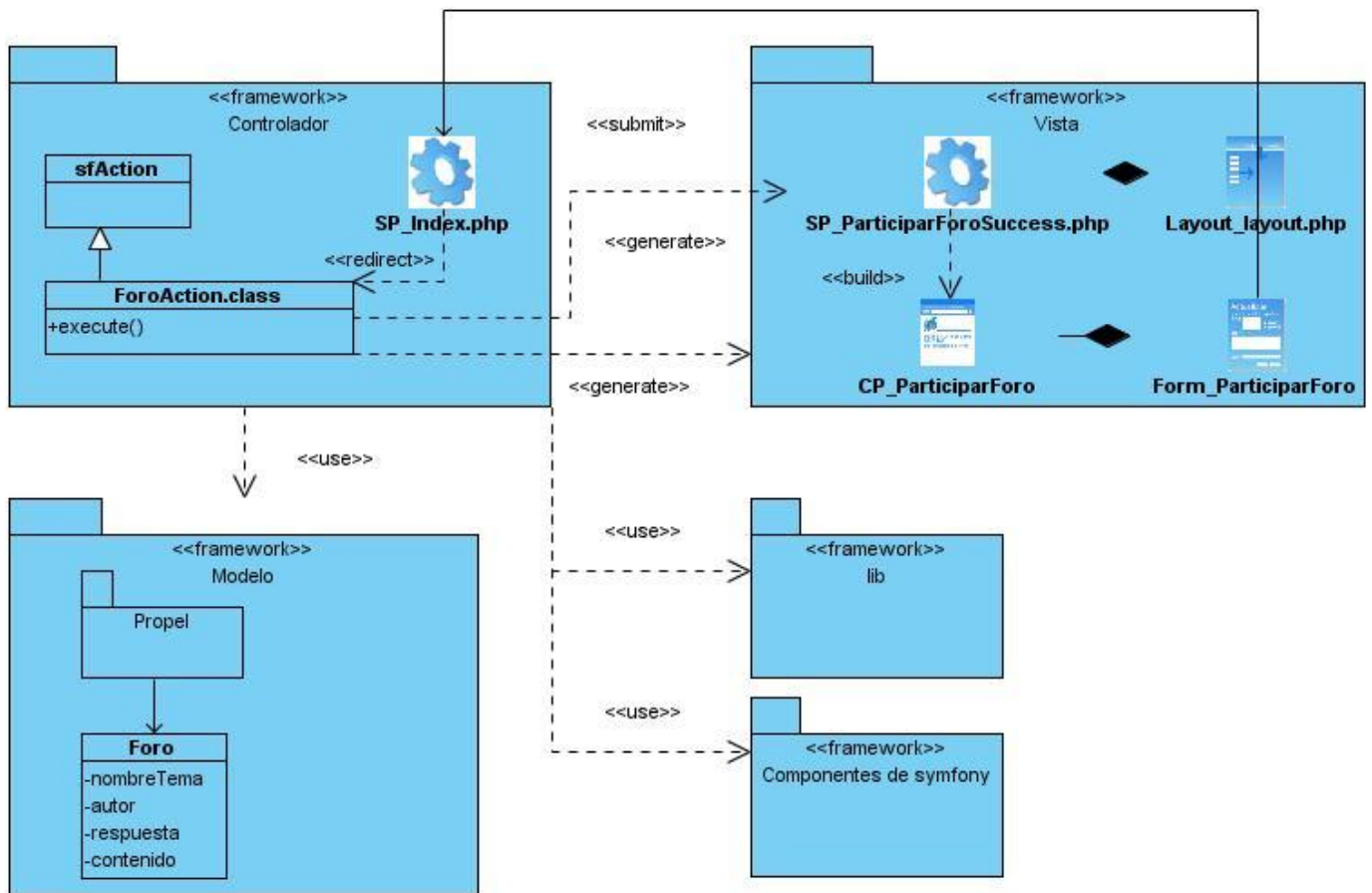


Figura 20: Diagrama de clases del diseño "Participar foro"

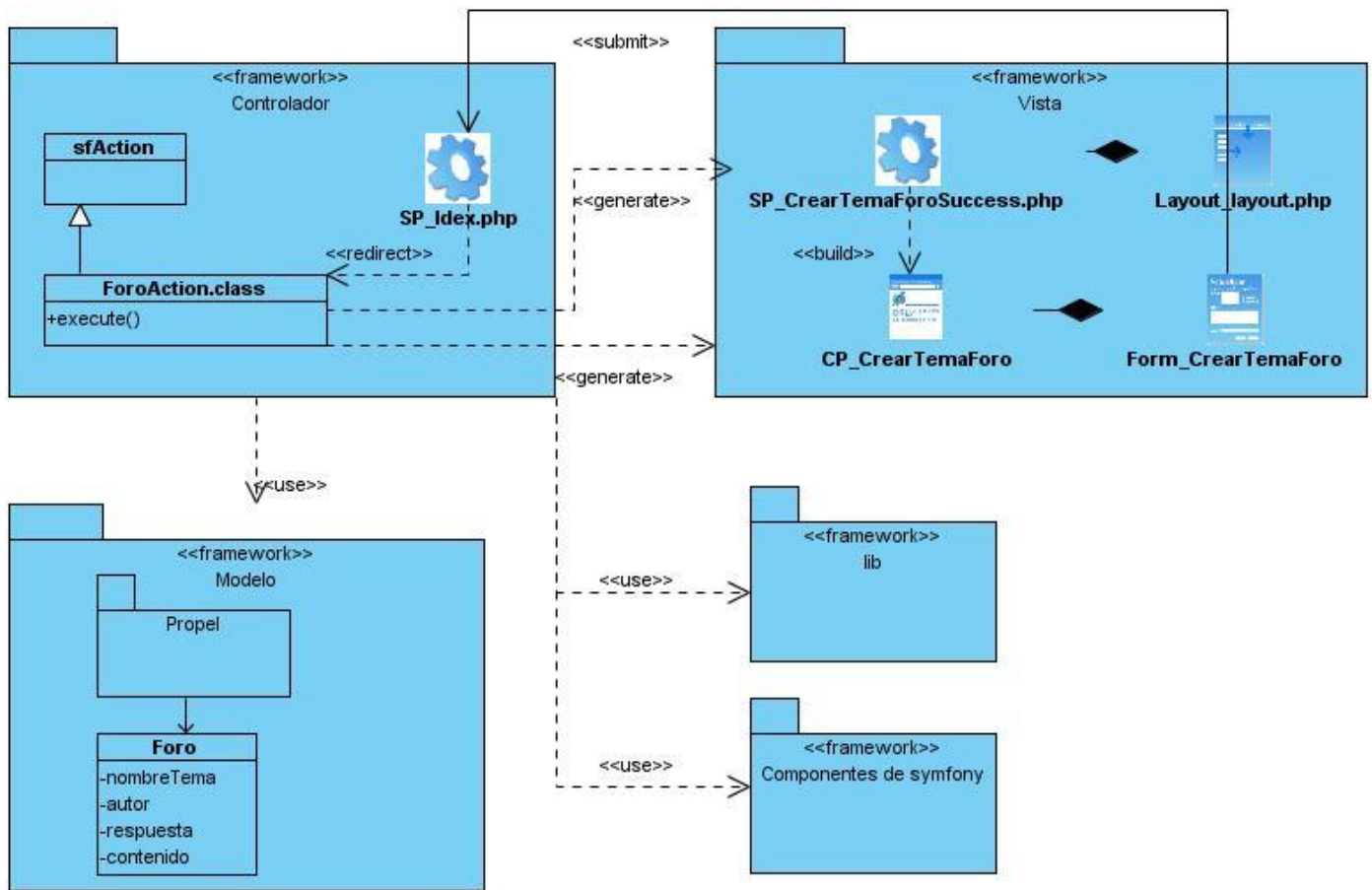


Figura 21: Diagrama de clases del diseño "Crear tema foro"

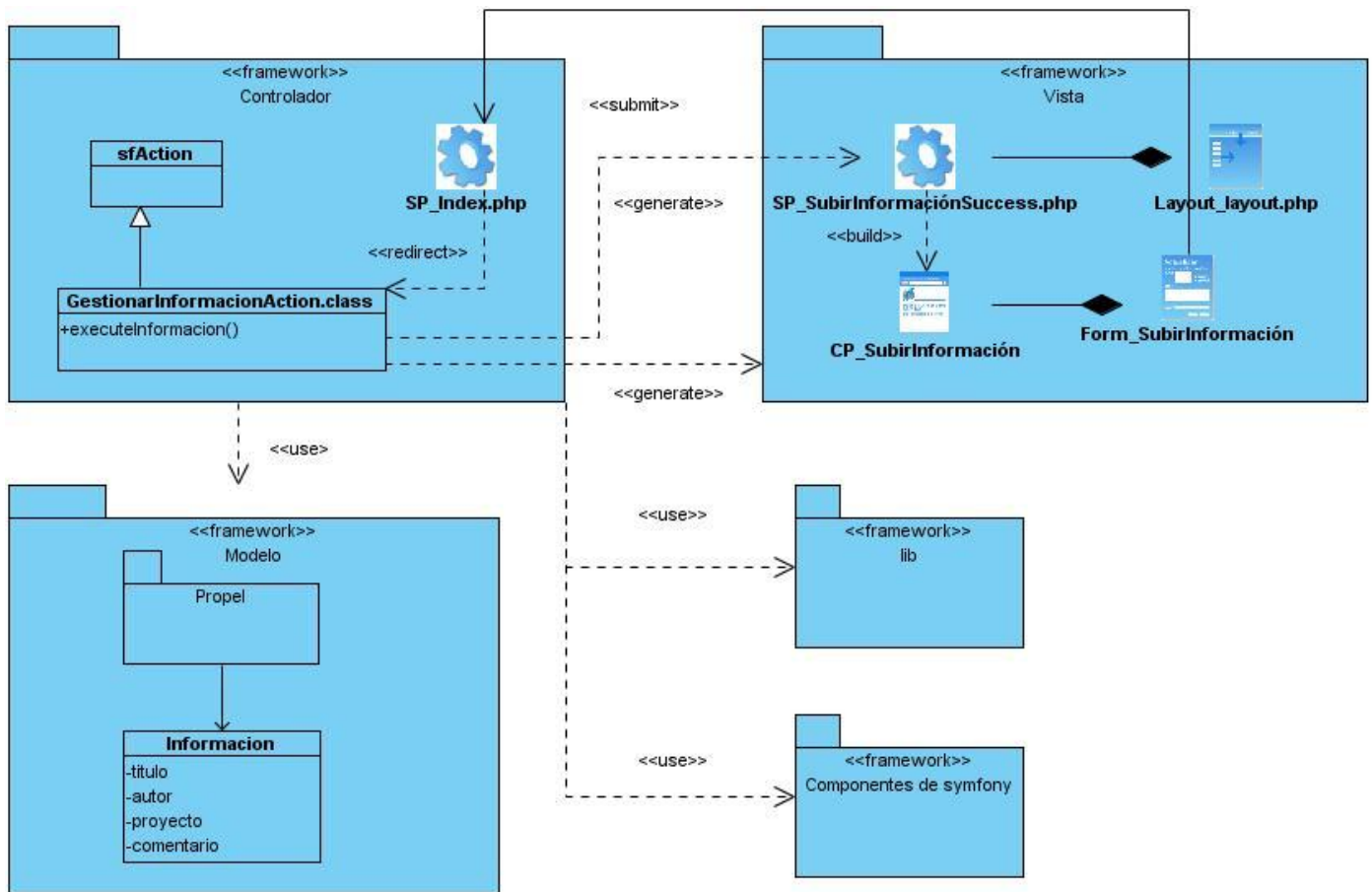


Figura 22: Diagrama de clases del diseño "Subir información"

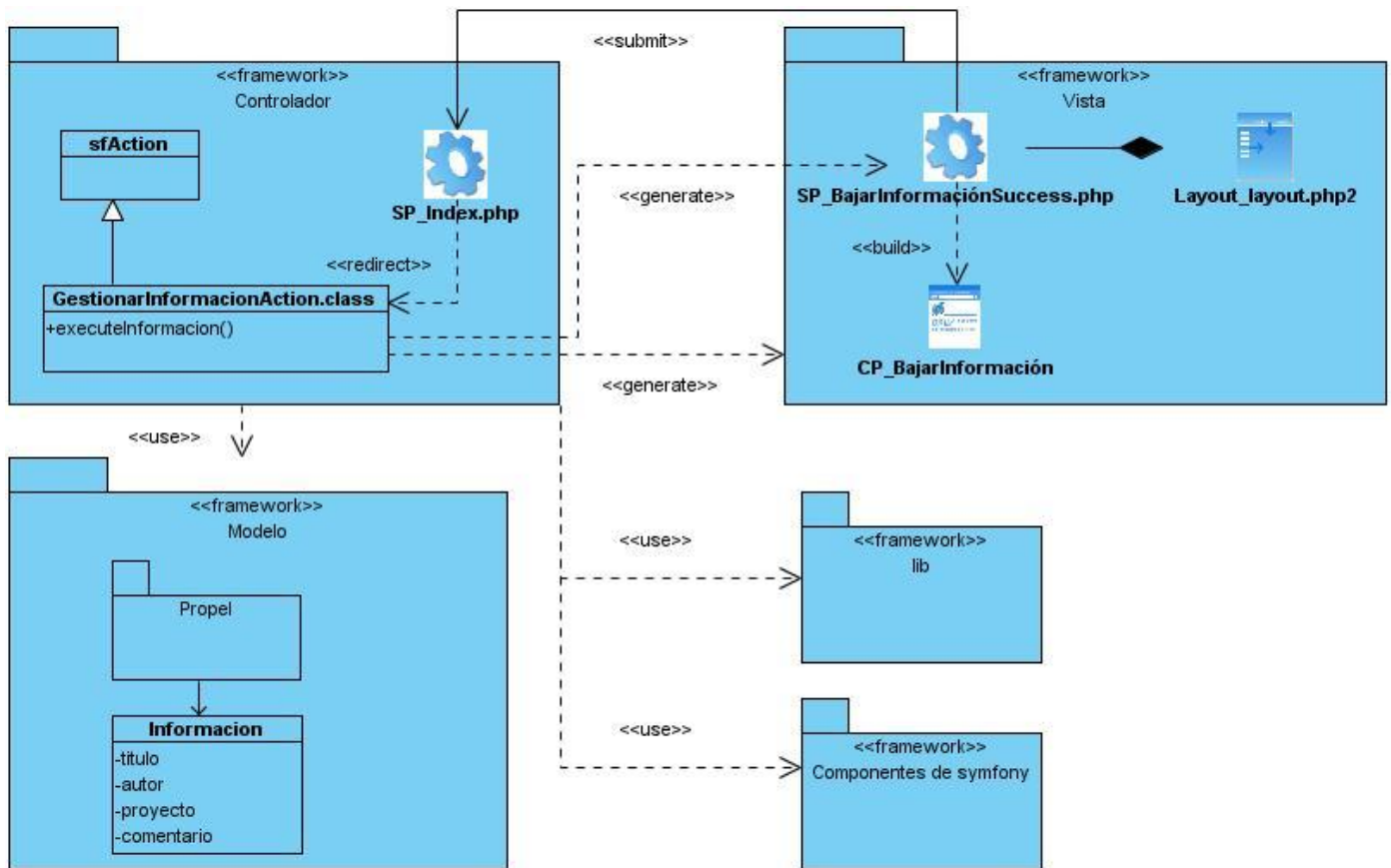


Figura 23: Diagrama de clases del diseño "Bajar información"

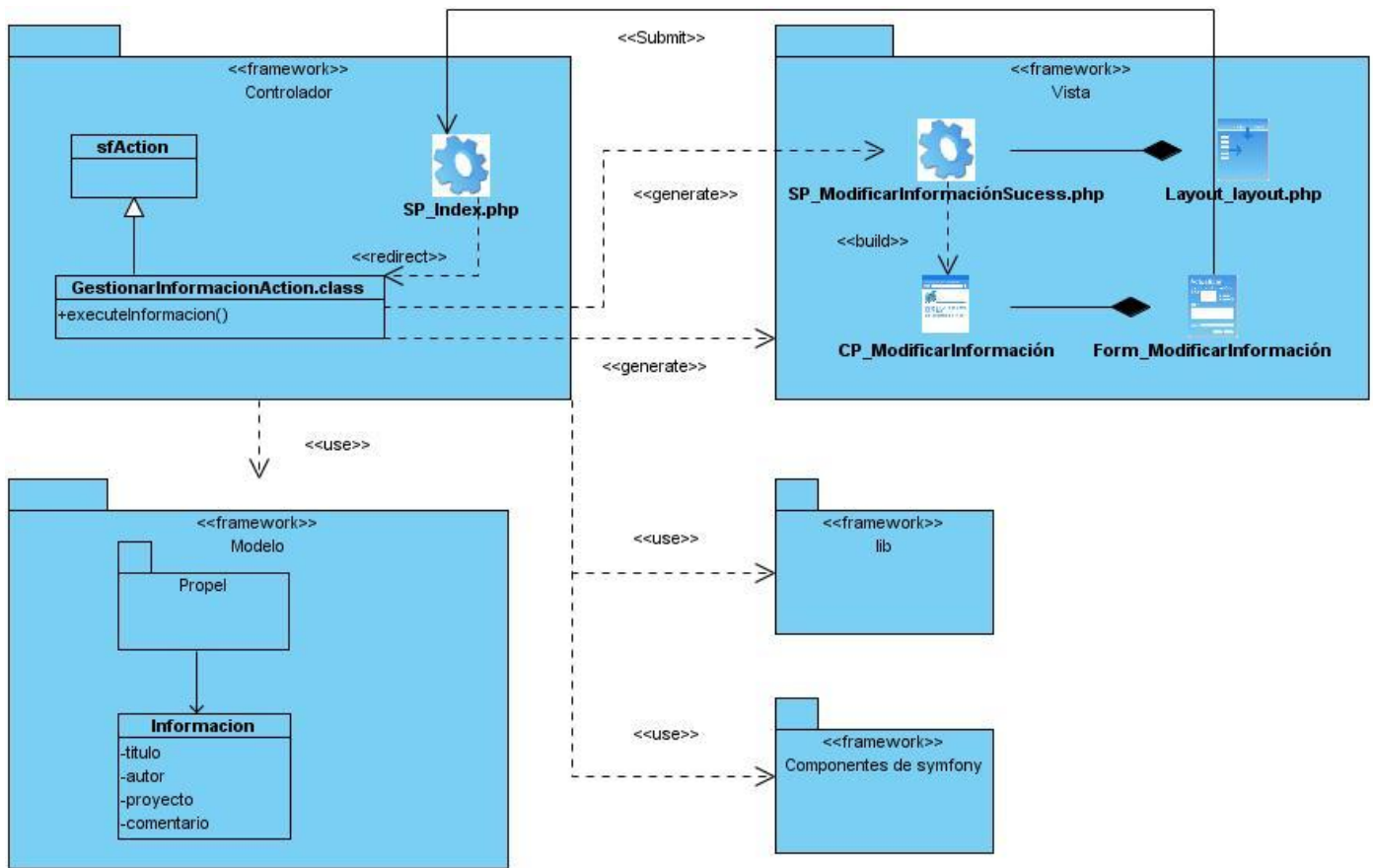


Figura 24: Diagrama de clases del diseño "Modificar información"

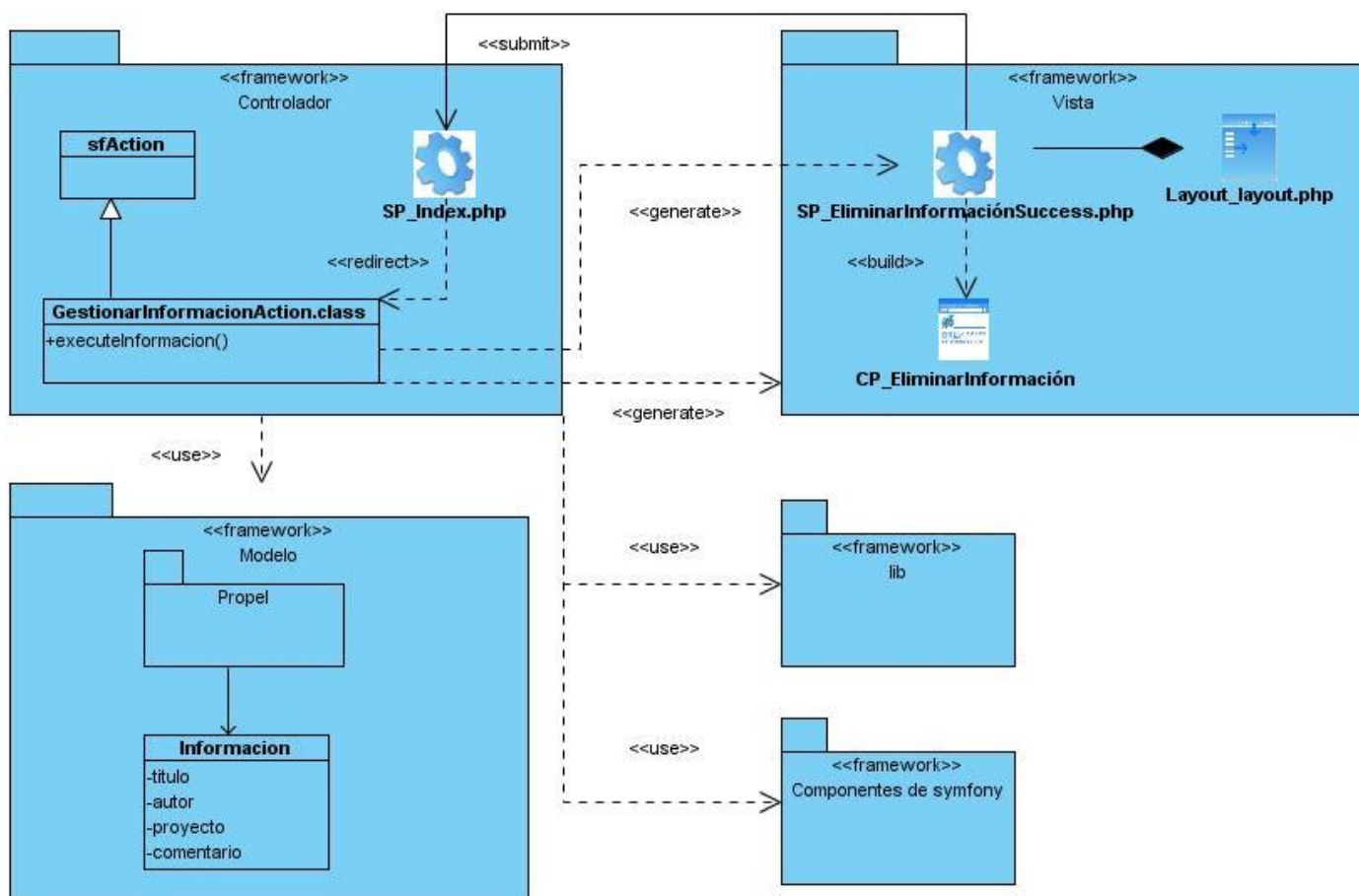


Figura 25: Diagrama de clases del diseño "Eliminar información"

4.3.2 Diagramas de secuencia

Para modelar los aspectos dinámicos de este sistema se confeccionaron los diagramas de secuencia por cada sección de los casos de uso. A continuación se incorporan los diagramas de secuencia de los casos de uso Autenticar usuario, Contener foro y Gestionar información; cada uno de ellos desglosados en sus respectivas secciones. El resto de los diagramas se pueden localizar en el anexo 3.

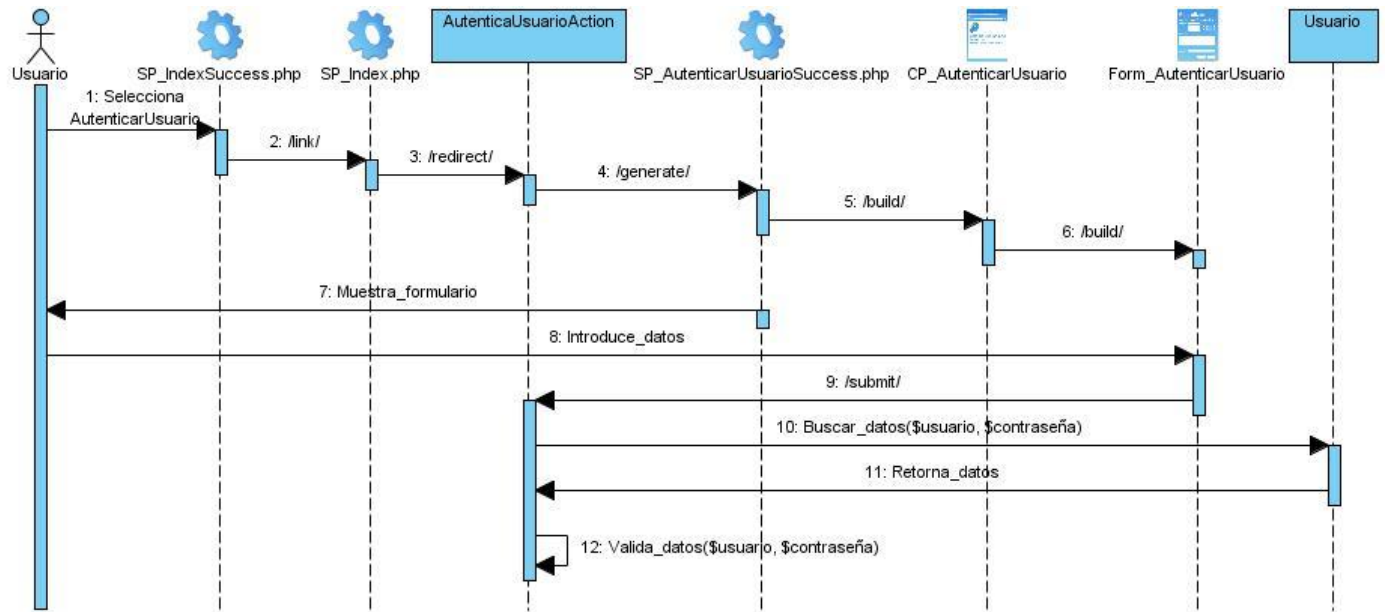


Figura 26: Diagrama de secuencia "Autenticar usuario"

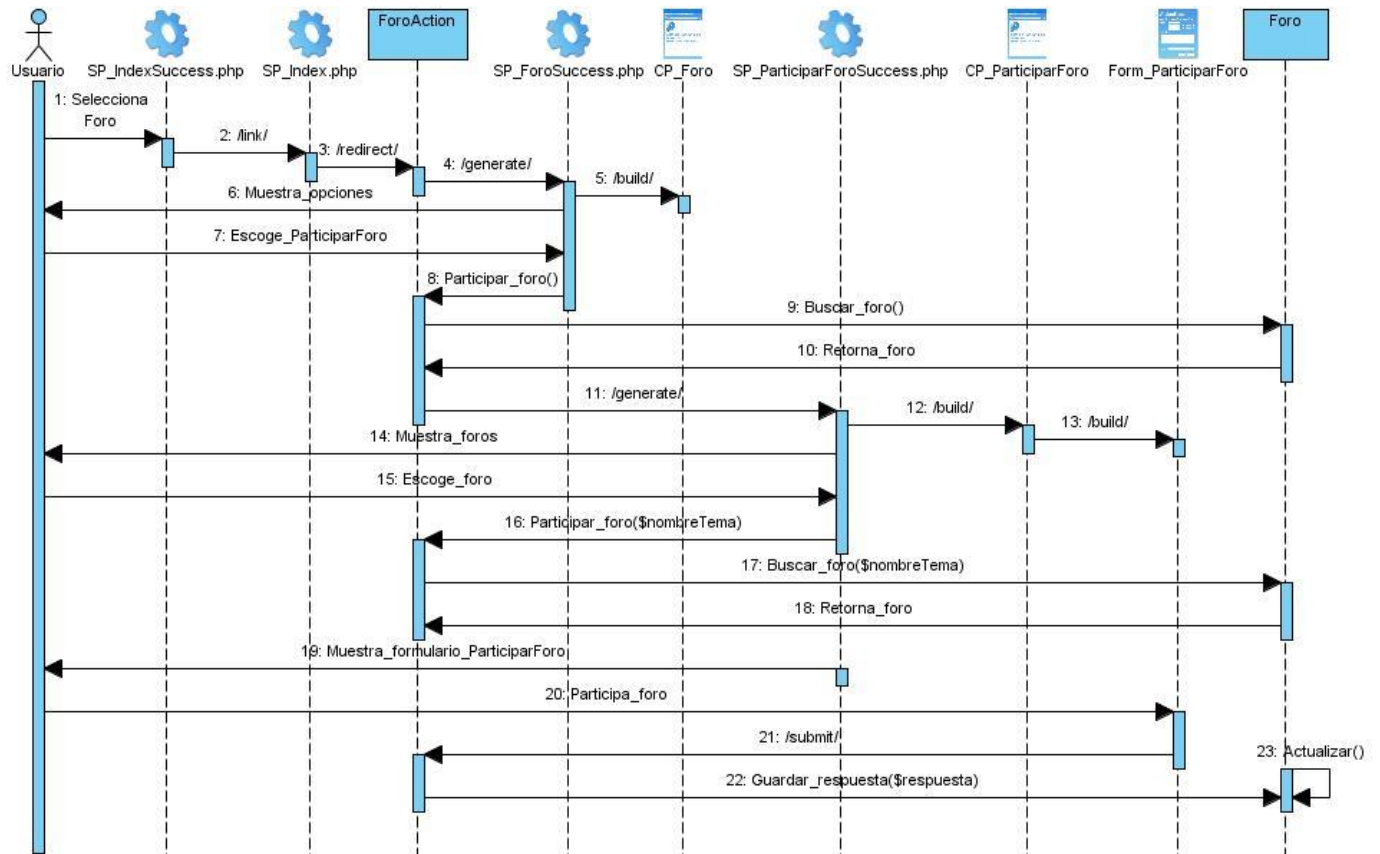


Figura 27: Diagrama de secuencia "Participar foro"

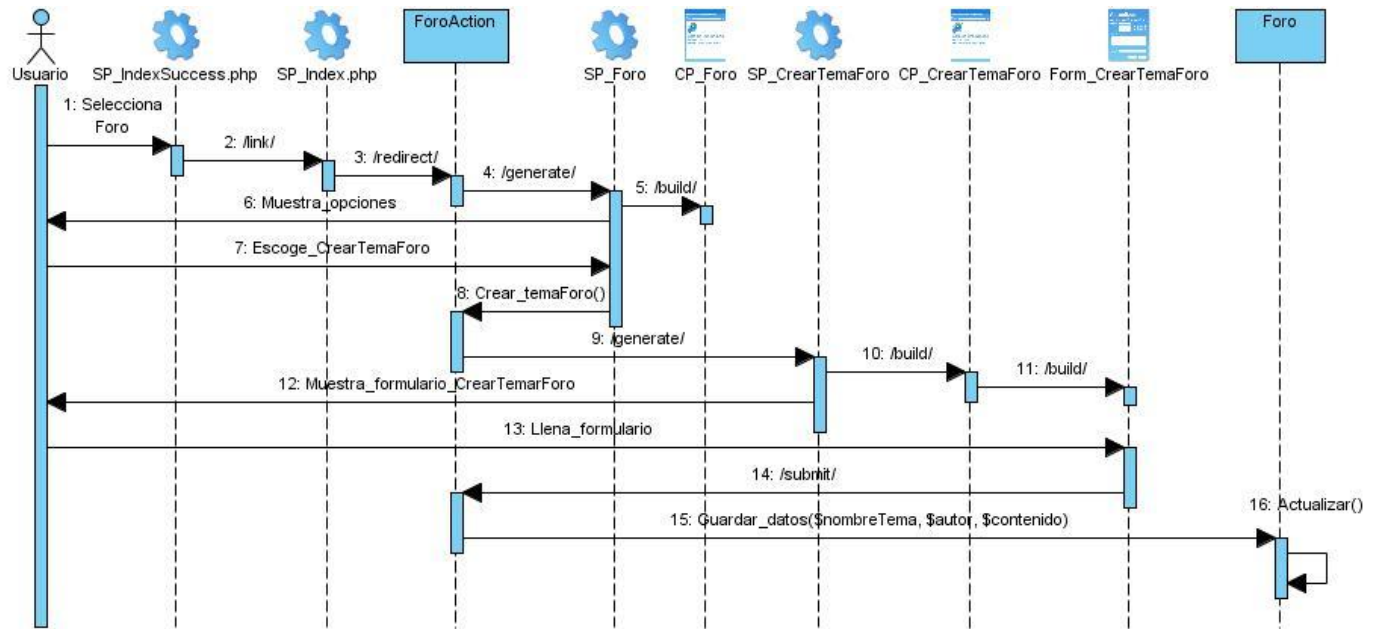


Figura 28: Diagrama de secuencia "Crear tema foro"

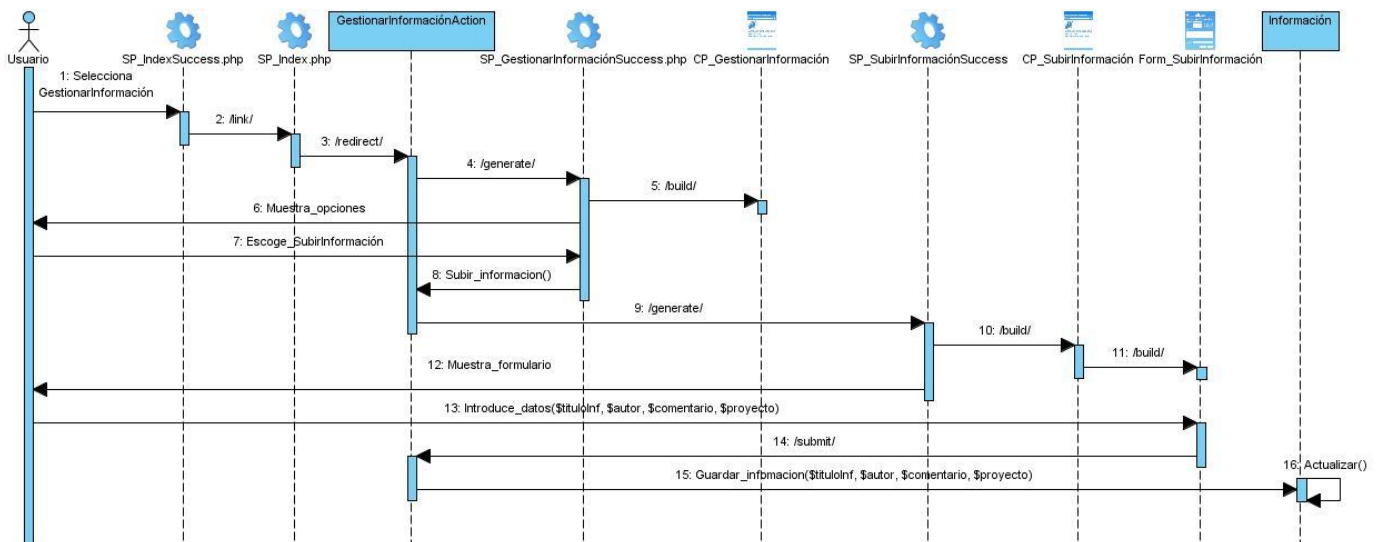


Figura 29: Diagrama de secuencia "Subir información"

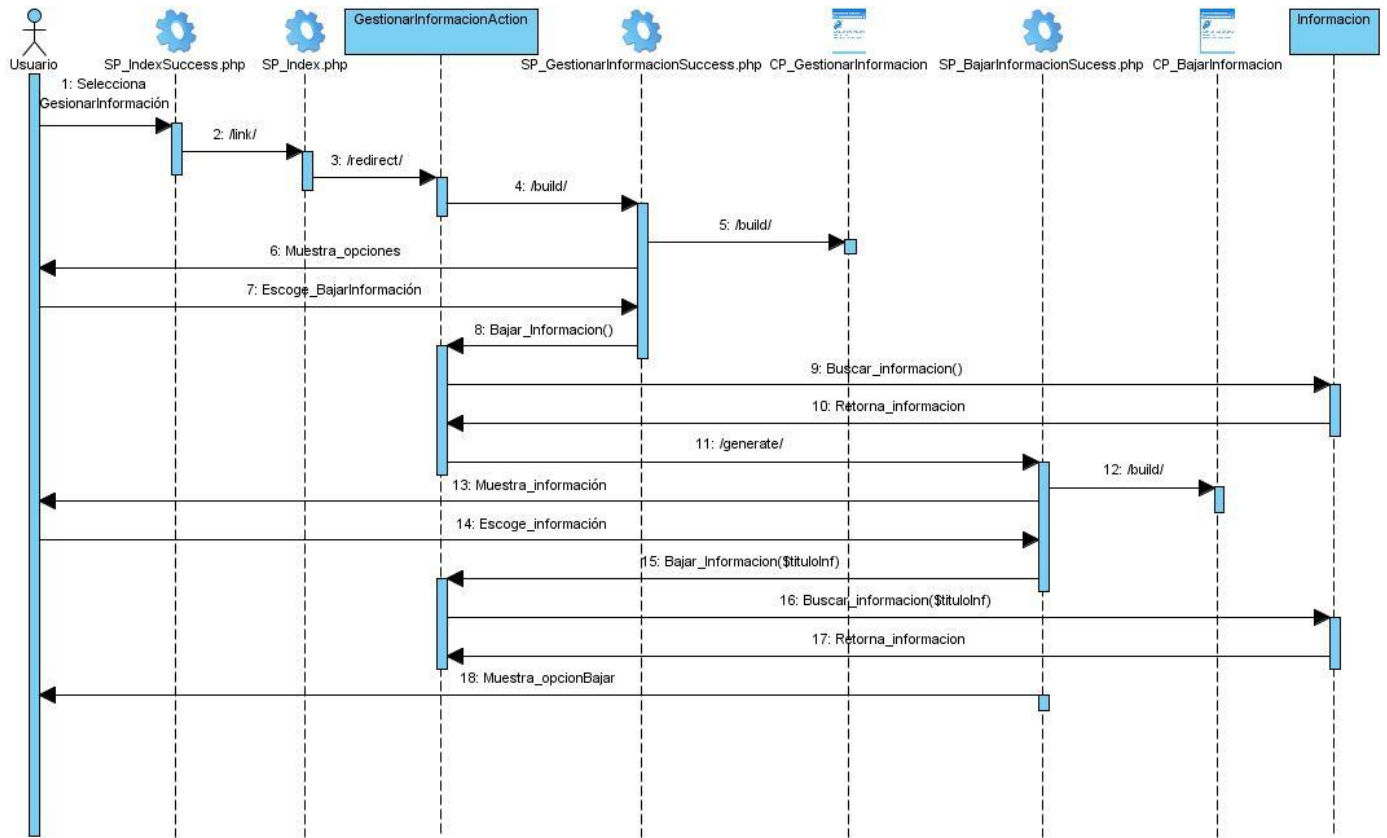


Figura 30: Diagrama de secuencia "Bajar información"

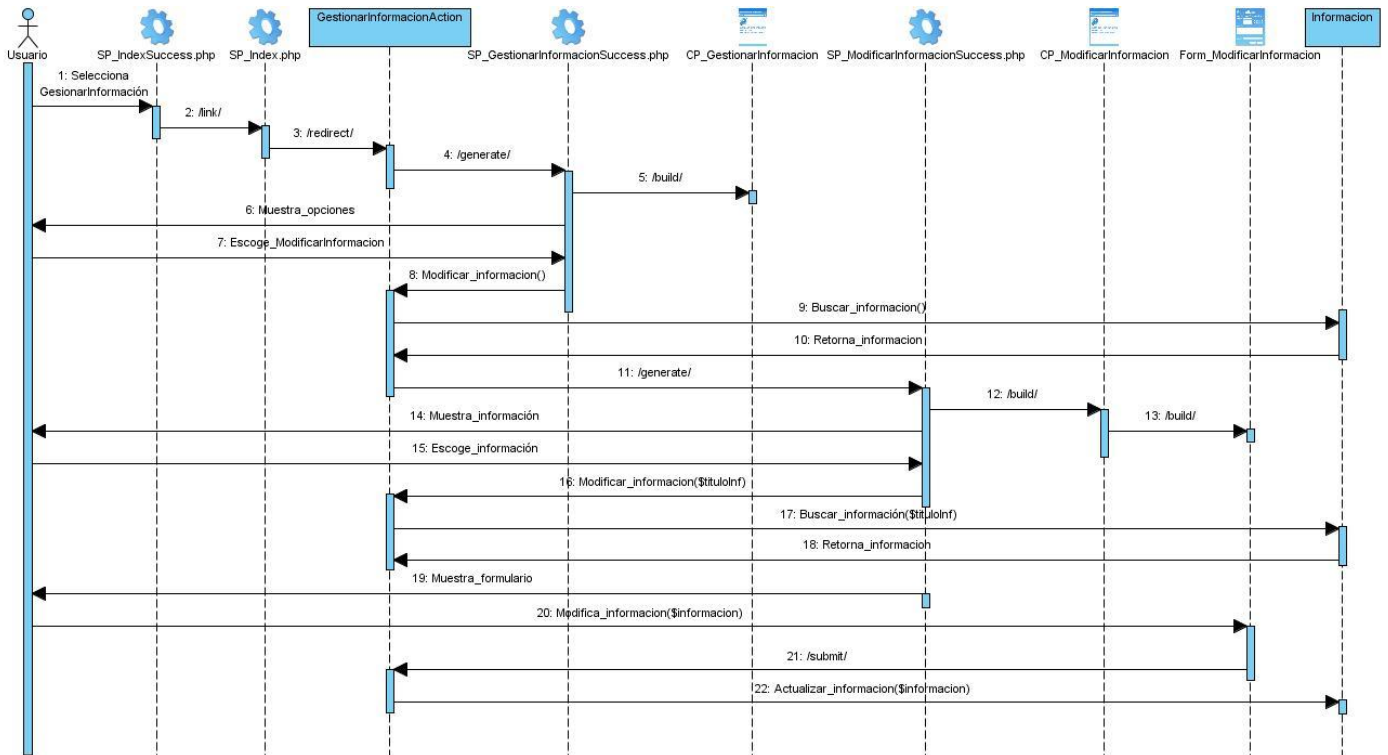


Figura 31: Diagrama de secuencia "Modificar información"

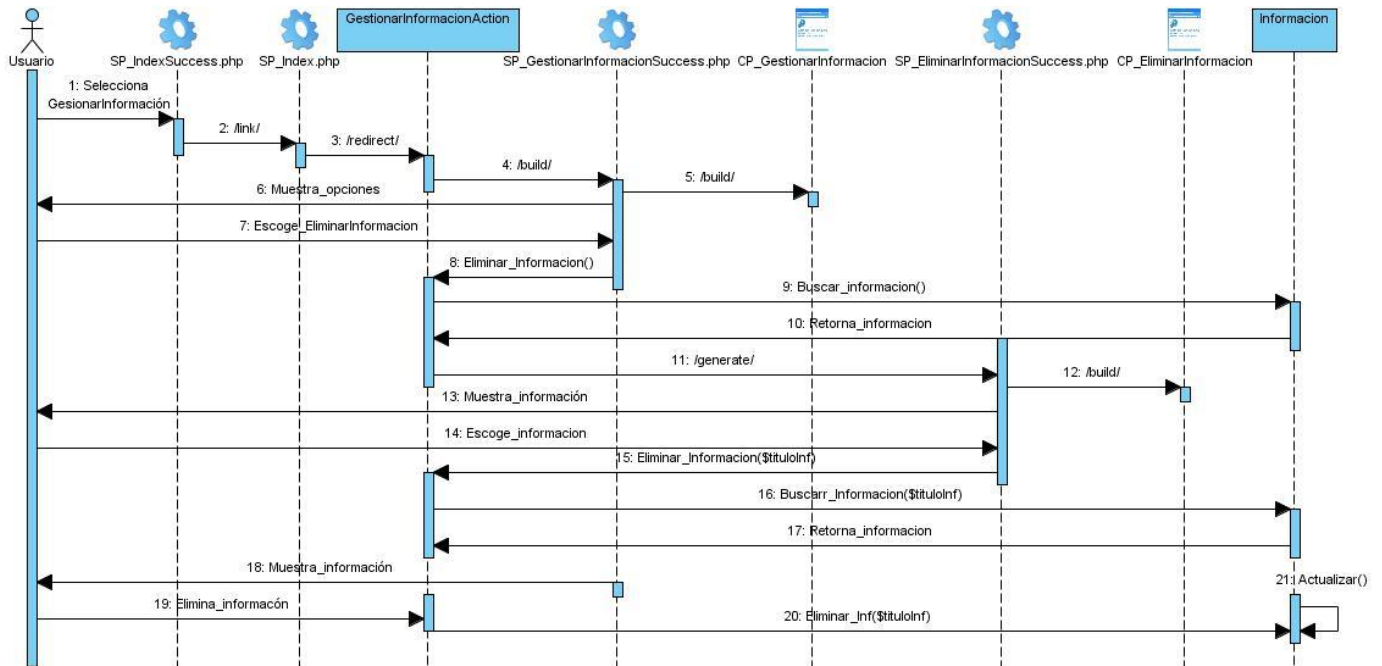


Figura 32: Diagrama de secuencia "Eliminar información"

4.3.3 Descripción de las clases

En este epígrafe se describen las clases action que tendrá el sistema, las descripciones de las clases entidad se encuentran en el anexo 4.

Tabla 11: Descripción de la clase AutenticarUsuarioAction.

Nombre	AutenticarUsuarioAction
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	Validar_datos(\$usuario, \$contraseña)
Descripción:	Validar que el usuario y contraseña introducidos sean correctos.

Tabla 12: Descripción de la clase ForoAction.

Nombre	ForoAction
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	Participar_foro(\$nombreTema)
Descripción:	El usuario puede participar en el foro cuyo nombre es pasado por parámetro.
Nombre:	Crear_tema(\$nombreTema, \$autor, \$contenido)
Descripción:	Crear un nuevo tema de discusión en el foro.

Tabla 13: Descripción de la clase GestionarInformacionAction.

Nombre	GestionarInformacionAction
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	Subir_Informacion(\$tituloInf, \$autor, \$comentario, \$proyecto)
Descripción:	Se encarga de subir la nueva información a la aplicación.
Nombre:	Bajar_Informacion(\$tituloInf)
Descripción:	Brinda la posibilidad de descargar la información deseada.
Nombre:	Modificar_Informacion(\$tituloInf)
Descripción:	Se modifican los datos de la información escogida.
Nombre:	Eliminar_informacion(\$tituloInf)
Descripción:	Elimina la información que se pasa por parámetro.

Tabla 14: Descripción de la clase GestionarProyectoAction.

Nombre	GestionarProyectoAction
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	Insertar_proyecto(\$nombreProy, \$areaTematica, \$liderProy)
Descripción:	Inserta un nuevo proyecto con los datos pasados por parámetro.

Nombre:	Modificar_proyecto(\$nombreProy)
Descripción:	Modifica datos del proyecto que su nombre coincida con el nombre pasado por parámetro.
Nombre:	Eliminar_proyecto(\$nombreProy)
Descripción:	Elimina el proyecto cuyo nombre es pasado por parámetro.

Tabla 15: Descripción de la clase GestionarEventoAction.

Nombre	GestionarEventoAction
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	Insertar_Evento(\$nombreEven, \$fecha, \$lugar, \$descripcion, \$categoria)
Descripción:	Inserta un nuevo evento con los datos pasados por parámetro.
Nombre:	Modificar_Evento(\$nombreEven)
Descripción:	Modifica datos del evento cuyo nombre es pasado por parámetro.
Nombre:	Eliminar_Evento(\$)
Descripción:	Elimina el evento que su nombre coincida con el nombre pasado por parámetro.

4.4 Diseño de la Base de datos

Los sistemas de base de datos almacenan información y permiten a los usuarios recuperarla y actualizarla en base a sus peticiones, por lo que es de gran importancia el diseño de la base de datos. A partir del diagrama de clases persistentes se obtuvo el siguiente diagrama Entidad Relación de la Base de Datos.

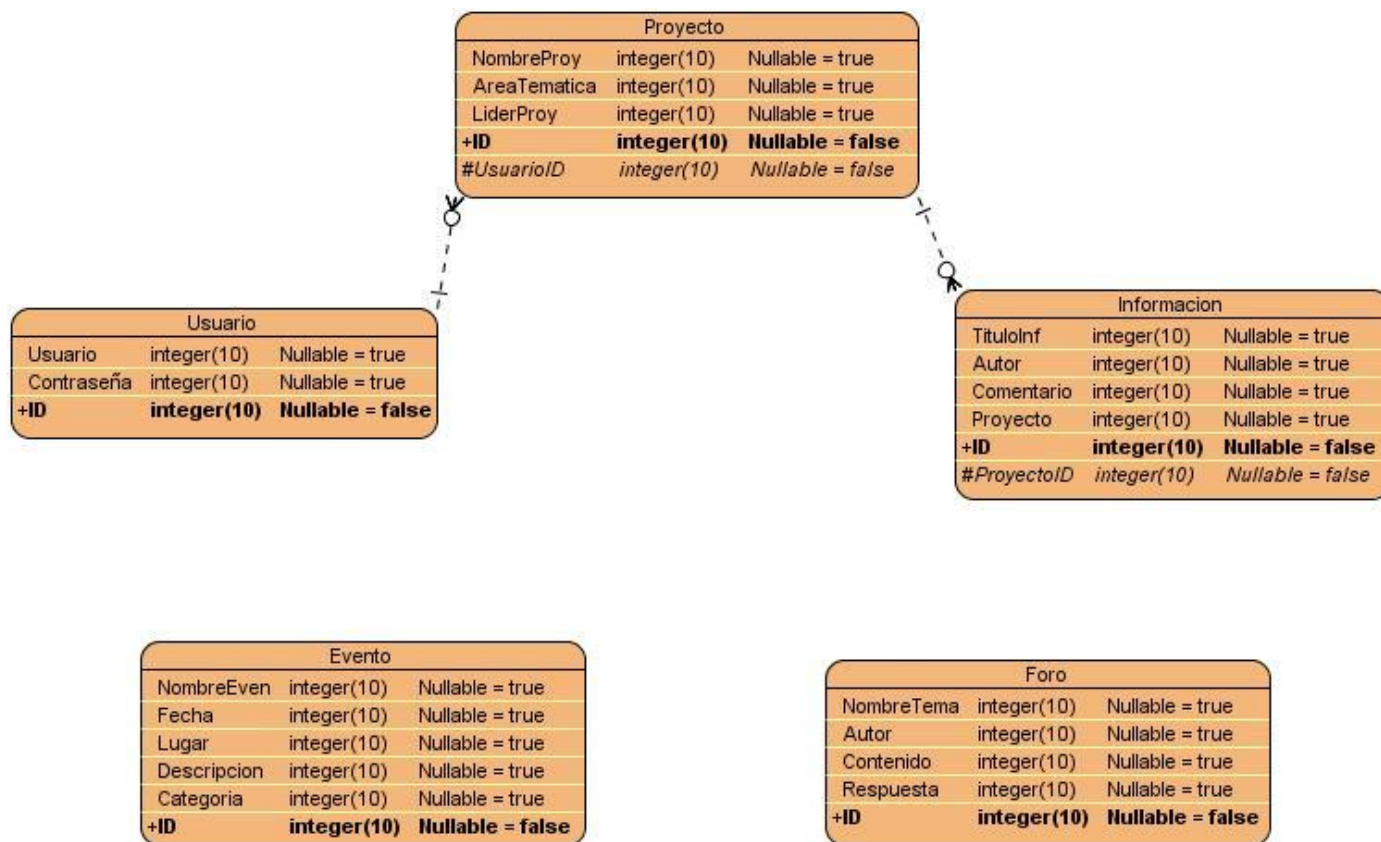


Figura 33: Diagrama Entidad Relación de la Base de Datos

4.4.1 Descripción de las tablas

Tabla 16: Descripción de la tabla Usuario.

Nombre	Usuario
Descripción	Esta tabla contiene todos los usuarios del polo productivo realidad virtual.
Atributo	Descripción
usuario	Usuario del polo
contraseña	Contraseña del usuario
ID	El identificador de usuario

Tabla 17: Descripción de la tabla Proyecto.

Nombre	Proyecto	
Descripción	Esta tabla contiene todos los proyectos por áreas temáticas del polo.	
Atributo	Descripción	
nombreProy	Nombre del proyecto	
areaTematica	A que área temática pertenece	
liderProy	Nombre del líder del proyecto	
ID	Identificador de proyecto	
usuarioID	El identificador de usuario que pasó como llave foránea.	

Tabla 18: Descripción de la tabla Información.

Nombre	Información	
Descripción	Esta tabla almacena las informaciones que se suben al sitio web.	
Atributo	Descripción	
tituloInf	Título de la información	
autor	Autor de la información	
comentario	Comentario acerca de que trata la información	
proyecto	Proyecto la que pertenece la información	
ID	Identificador de información	
proyectoID	El identificador de proyecto que pasó como llave foránea.	

Tabla 19: Descripción de la tabla Foro.

Nombre	Foro	
Descripción	Esta tabla guarda los datos de los temas de discusión del foro.	
Atributo	Descripción	
nombreTema	Nombre del tema de discusión	
autor	Autor del tema de discusión	
contenido	Pregunta o comentario del tema de discusión	

respuesta	Respuesta al tema de discusión
ID	Identificador de foro

Tabla 20: Descripción de la tabla Evento.

Nombre	Evento
Descripción	Esta tabla contiene todos los eventos que se planifican en el polo de realidad virtual.
Atributo	Descripción
nombreEven	Nombre del evento
fecha	Fecha en la que se va a realizar el evento
lugar	Lugar en el que se va a desarrollar el evento
descripcion	Breve descripción del evento
categoria	Categoría del evento, si es un concurso, un curso optativo o un taller.
ID	Identificador de evento

4.5 Definiciones de diseño que se apliquen.

Estándares en la interfaz de la aplicación.

El diseño de la interfaz es una de las cosas fundamentales a tratar en la presentación de una aplicación, teniendo en cuenta que esa es la capa que se presentará al usuario y por lo tanto debe ser lo más amigable y comprensible posible.

El color y contraste que se recomienda por la mayoría de los diseñadores es la utilización de letras oscuras sobre fondo claro. Solo es adecuado usar fondo oscuro y texto claro para títulos, cabeceras o celdas. El problema radica en que los fondos oscuros con texto claro son muy empleados en publicidad y ello puede provocar que contenidos no publicitarios sean afectados por la ceguera a los banners y por tanto sean ignorados por los usuarios.

Barra de Menú.

La barra del menú debe ser horizontal. La denominación de las opciones principales debe referirse o encapsular a un grupo de acciones afines y no a una acción en particular.

Diseño Principal de la Página.

La página principal del sistema debe llevar una cabecera que lo identificará, imagen no muy grande representativa del sitio web y del polo para el cual fue diseñado, y una barra de menú con sus principales opciones.

4.6 Seguridad

Se cuenta con políticas de seguridad bien establecidas. Para interactuar con el sistema el usuario debe autenticarse. Se estableció un nivel de acceso a las distintas funcionalidades que brinda la aplicación, según el rol desempeñado por cada integrante del polo productivo de Realidad Virtual.

4.7 Interfaz

Mediante el uso del lenguaje de programación del lado del servidor PHP y la tecnología AJAX se logra desarrollar una interfaz cómoda para el usuario creando una experiencia interactiva que no requiere recargar la página cada vez que el mismo realiza una petición al servidor, la misma es además, sencilla, con tonalidades de colores claros predominando el blanco, y fácil de usar.

4.8 Concepción de la ayuda

La ayuda es un elemento necesario para que los usuarios tengan un apoyo al trabajar con el sistema, pues les permite tener conocimiento de cómo funciona cada una de las opciones del mismo. Para esto se mostrarán mensajes explicativos en los campos de datos a llenar en las distintas páginas del sistema, cuando se pase el puntero por encima de un signo de ayuda que se muestra al lado de los campos a llenar. Existirá una ayuda en línea que ilustrará las funcionalidades del sistema, mostrándose en ella detalladamente cada aspecto de dichas funcionalidades. Esto permitirá que el usuario esté en todo momento informado y orientado cuando se encuentre interactuando en el sistema.

4.9 Conclusiones parciales

En el flujo de trabajo de análisis y diseño se realiza un análisis del sistema ya en términos de solución. En este capítulo quedo definida la estructura del análisis, donde se proporciona una comprensión detallada de los requisitos con los cuales tiene que cumplir el sistema. Se le dio continuidad al análisis realizando el diseño del sistema a través de los diagramas de clases del diseño con estereotipos Web y baja el patrón de diseño Modelo Vista Controlador. Además se especificó más el sistema mediante los diagramas de secuencia y se diseñó la base de datos. Siempre teniendo en cuenta las características del sistema.

CONCLUSIONES

Durante el desarrollo de esta investigación se realizó un estudio y análisis de los modelos y conceptos fundamentales de la Gestión del Conocimiento (GC). Se pudo observar que ninguno de los modelos existentes para la GC era aplicable al polo productivo de Realidad Virtual y se propuso un modelo específico para el polo. Además se investigaron las principales herramientas tecnológicas actuales para el desarrollo de aplicaciones web. Se modelaron todos los diagramas y representaciones necesarias para completar el diseño del sistema.

El objetivo general planteado en este trabajo de diploma fue cumplido satisfactoriamente, así como las tareas descritas para llevar a cabo dicho objetivo. La solución propuesta se ajusta a los estándares internacionales para el desarrollo de software. La utilización de la metodología RUP avala que el desarrollo del software quede estructurado por fases, con la calidad requerida.

Se concluye que la aplicación de un modelo de gestión del conocimiento brinda la posibilidad de crear un acercamiento a la calidad que se quiere alcanzar en el proceso y en los productos de software que se desarrollan en cada uno de los proyectos que conforman el polo, logrando transformar el conocimiento tácito y explícito y de esa forma aumentar la capacidad de novedad y el trabajo en equipo a nivel del polo.

RECOMENDACIONES

Se recomienda:

- Que se realice la implementación del sistema diseñado.
- Añadir más funcionalidades al sistema que se implemente de acuerdo con las facilidades que brinda Symfony.
- La utilización del presente trabajo de diploma como bibliografía para posibles investigaciones referentes al tema desarrollado en el mismo.
- Una vez implementada la aplicación web se ponga a funcionar en el polo productivo Realidad Virtual de la facultad 5 para el cual fue diseñado.

REFERENCIAS BIBLIOGRÁFICAS

1. GSI. (s.f.). Recuperado el Febrero de 2009, de http://www.gsi.dit.upm.es/~fsaez/intl/capitulos/14%20-Capital%20humano%20_y%20II_.pdf
2. *Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gestiondelconocimiento.com/pdf-art-gc/00218pepelara.pdf>
3. *Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de http://www.gestiondelconocimiento.com/conceptos_conocimiento.htm
4. *Definicion.de*. (s.f.). Recuperado el Febrero de 2009, de <http://definicion.de/gestion/>
5. *Gestio del Conocimiento.com*. (s.f.). Recuperado el Diciembre de 2008, de <http://www.gestiondelconocimiento.com/modelos.htm>
6. Pavez, A (s.f.). *La gestión del conocimiento en las organizaciones*. Recuperado el Diciembre de 2008, de <http://www.gestiondelconocimiento.com/documentos2/apavez/gdc.htm>
7. *Gnu.org*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gnu.org/philosophy/free-sw.es.html>
8. *Entorno Virtual de Aprendizaje*. (s.f.). Recuperado el Febrero de 2009, de <http://teleformacion.uci.cu/mod/resource/view.php?id=10530>
9. Mateu, C. (2004). *Desarrollo de aplicaciones web*.
10. Pérez, J. E. (s.f.). *Introducción a AJAX*. Recuperado el Diciembre de 2008, de <http://www.librosweb.es/ajax/>
11. Stig Sæther Bakken, A. A. (2001). *Manual de PHP*.
12. Fabien Potencier, F. Z. (s.f.). *Symfony la guía definitiva*. Recuperado el Febrero de 2009, de <http://www.librosweb.es>
13. W. Jason Gilmore, R. T. (2006). *Beginning PHP and PostgreSQL 8 from Novice to Professional*. Apress.
14. Ivar Jacobson, G. B. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley.
15. Ivar Jacobson, J. R. *El Lenguaje Unificado de Modelado*. Addison Wesley.

16. Martorell, M. (2 de Julio de 2008). Subsistema automatizado de recuperación de datos del Balance Nacional de Recursos y Reservas de Minerales Sólidos en la Oficina Nacional de Recursos Minerales. Ciudad Habana, Cuba.
17. WIKIPROD. (s.f.). Recuperado el Febrero de 2009, de <http://ucipedia.uci.cu/index.php/Eclipse>

BIBLIOGRAFÍA

- 12MANAGE*. (s.f.). Recuperado el Diciembre de 2008, de http://www.12manage.com/i_ki_es.html
- Definicion.de*. (s.f.). Recuperado el Febrero de 2009, de <http://definicion.de/gestion/>
- Enciclopedia.us*. (s.f.). Recuperado el Febrero de 2009, de http://enciclopedia.us.es/index.php/C%C3%B3digo_abierto
- Entorno Virtual de Aprendizaje*. (s.f.). Recuperado el Febrero de 2009, de <http://teleformacion.uci.cu/mod/resource/view.php?id=10530>
- Fabien Potencier, F. Z. (s.f.). *Symfony la guía definitiva*. Recuperado el Febrero de 2009, de <http://www.librosweb.es>
- Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de http://www.gestiondelconocimiento.com/conceptos_conocimiento.htm
- Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gestiondelconocimiento.com/pdf-art-gc/00218pepelara.pdf>
- Gestio del Conocimiento.com*. (s.f.). Recuperado el Noviembre de 2008, de <http://www.gestiondelconocimiento.com/index.php3>
- Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gestiondelconocimiento.com/pdf-art-gc/00218pepelara.pdf>
- Gestio del Conocimiento.com*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gestiondelconocimiento.com/modelos.htm>
- Pavez, A (s.f.). *La gestión del conocimiento en las organizaciones*. Recuperado el Diciembre de 2008, de <http://www.gestiondelconocimiento.com/documentos2/apavez/gdc.htm>
- Gestion-conocimiento*. (s.f.). Recuperado el Noviembre de 2008, de <http://www.gestion-conocimiento.com>
- GestioPolis*. (s.f.). Recuperado el Enero de 2009, de <http://www.gestiopolis.com/canales/gerencial/articulos/56/56/gc.htm>
- GestioPolis*. (s.f.). Recuperado el Noviembre de 2008, de <http://www.gestiopolis.com/dirgp/adm/gestionconocimiento.htm>
- Gnu.org*. (s.f.). Recuperado el Febrero de 2009, de <http://www.gnu.org/philosophy/free-sw.es.html>
- GSI*. (s.f.). Recuperado el Febrero de 2009, de http://www.gsi.dit.upm.es/~fsaez/intl/capitulos/14%20-Capital%20humano%20_y%20II_.pdf

Ivar Jacobson, G. B. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley.

Ivar Jacobson, J. R. *El Lenguaje Unificado de Modelado*. Addison Wesley.

Martorell, M. (2 de Julio de 2008). Subsistema automatizado de recuperación de datos del Balance Nacional de Recursos y Reservas de Minerales Sólidos en la Oficina Nacional de Recursos Minerales. Ciudad Habana, Cuba.

Mateu, C. (2004). *Desarrollo de aplicaciones web*.

Pérez, J. E. (s.f.). *Introducción a AJAX*. Recuperado el Diciembre de 2008, de <http://www.librosweb.es/ajax/>

Ponce. (s.f.). Recuperado el Enero de 2009, de http://ponce.inter.edu/cai/manuales/HERRAMIENTAS_BUSQUEDA_files/frame.htm#slide0001.htm

SciELO. (s.f.). Recuperado el Febrero de 2009, de http://www.scielo.cl/scielo.php?pid=S0718-07642007000100017&script=sci_arttext

Stig Sæther Bakken, A. A. (2001). *Manual de PHP*.

W. Jason Gilmore, R. T. (2006). *Beginning PHP and PostgreSQL 8 from Novice to Professional*. Apress.

WIKIPROD. (s.f.). Recuperado el Febrero de 2009, de <http://ucipedia.uci.cu/index.php/Eclipse>

ANEXOS

Anexo 1: Diagramas de clases del análisis

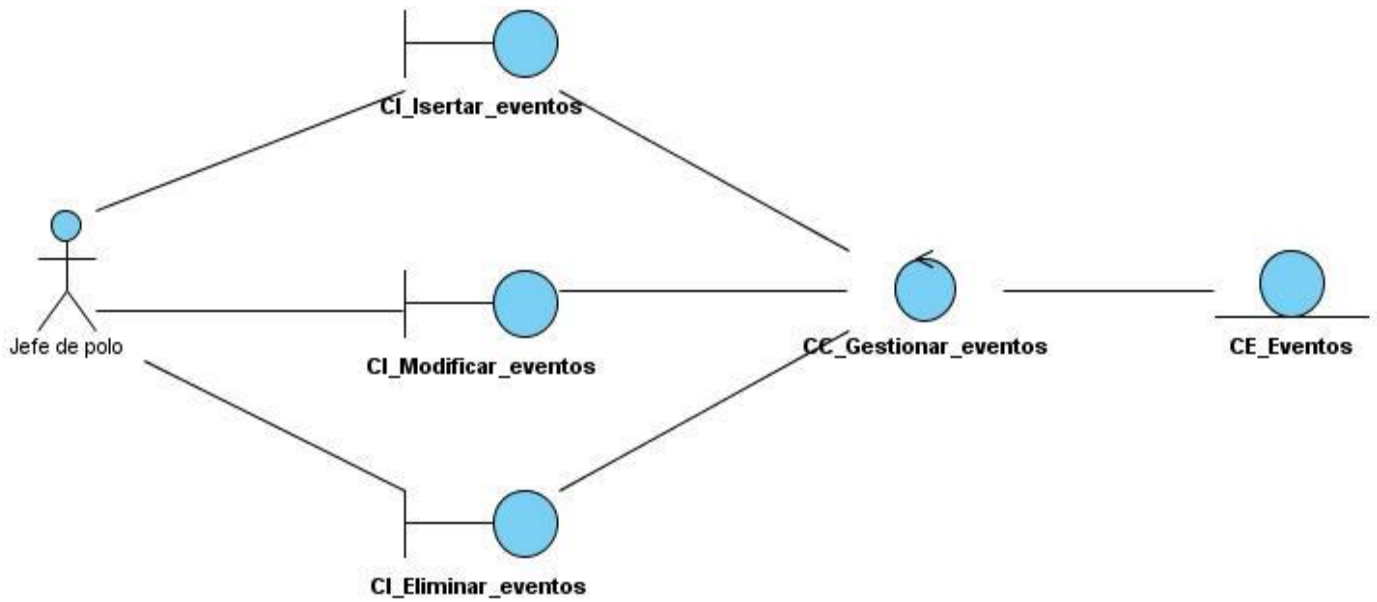


Figura 34: Diagrama de clases del análisis del caso de uso "Gestionar eventos"



Figura 35: Diagrama de clases del análisis del caso de uso "Mostrar información"



Figura 36: Diagrama de clases del análisis del caso de uso "Mostrar eventos"

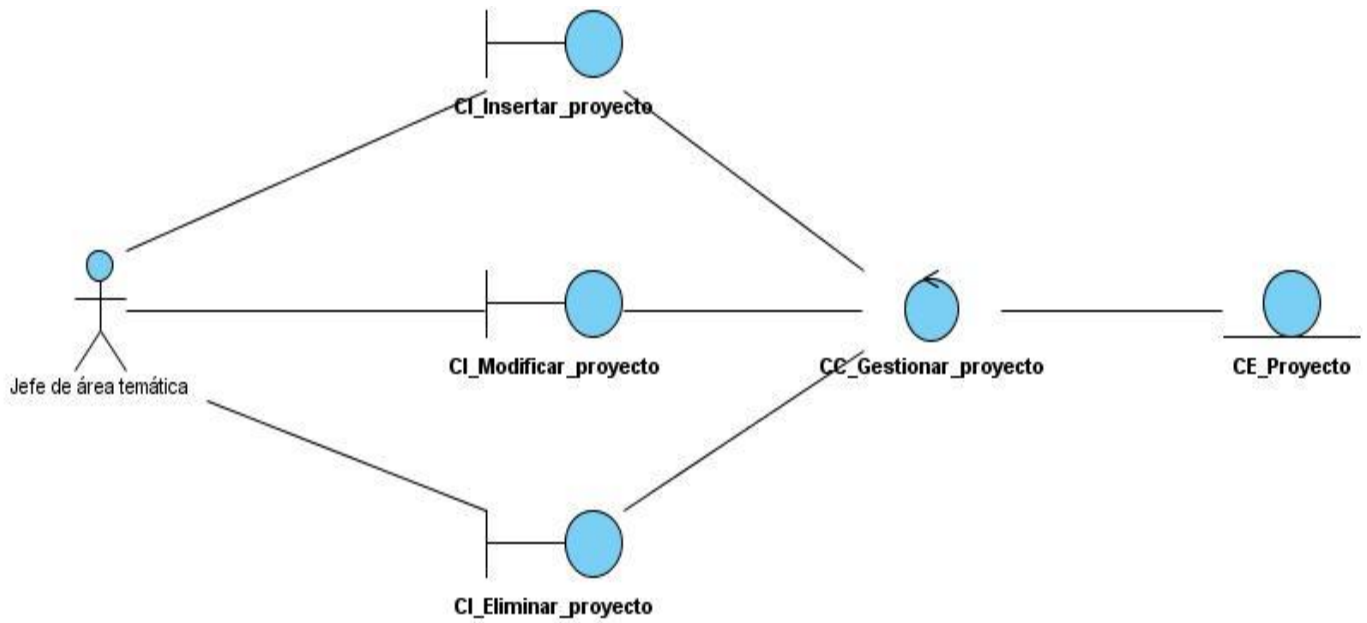


Figura 37: Diagrama de clases del análisis del caso de uso "Gestionar proyectos"



Figura 38: Diagrama de clases del análisis del caso de uso "Mostrar proyectos"

Anexo 2: Diagramas de clases del diseño

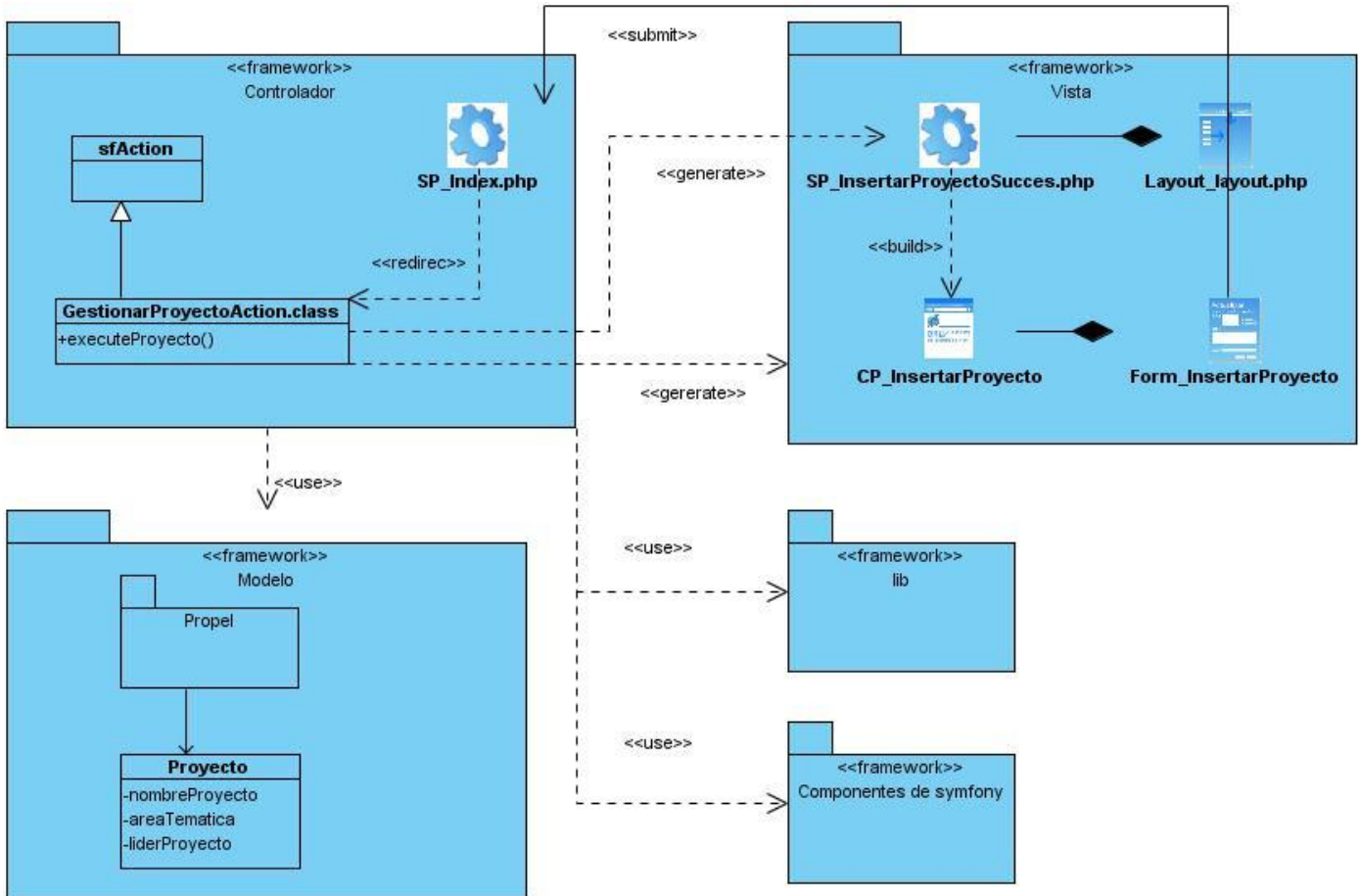


Figura 39: Diagrama de clases del diseño "Insertar proyecto"

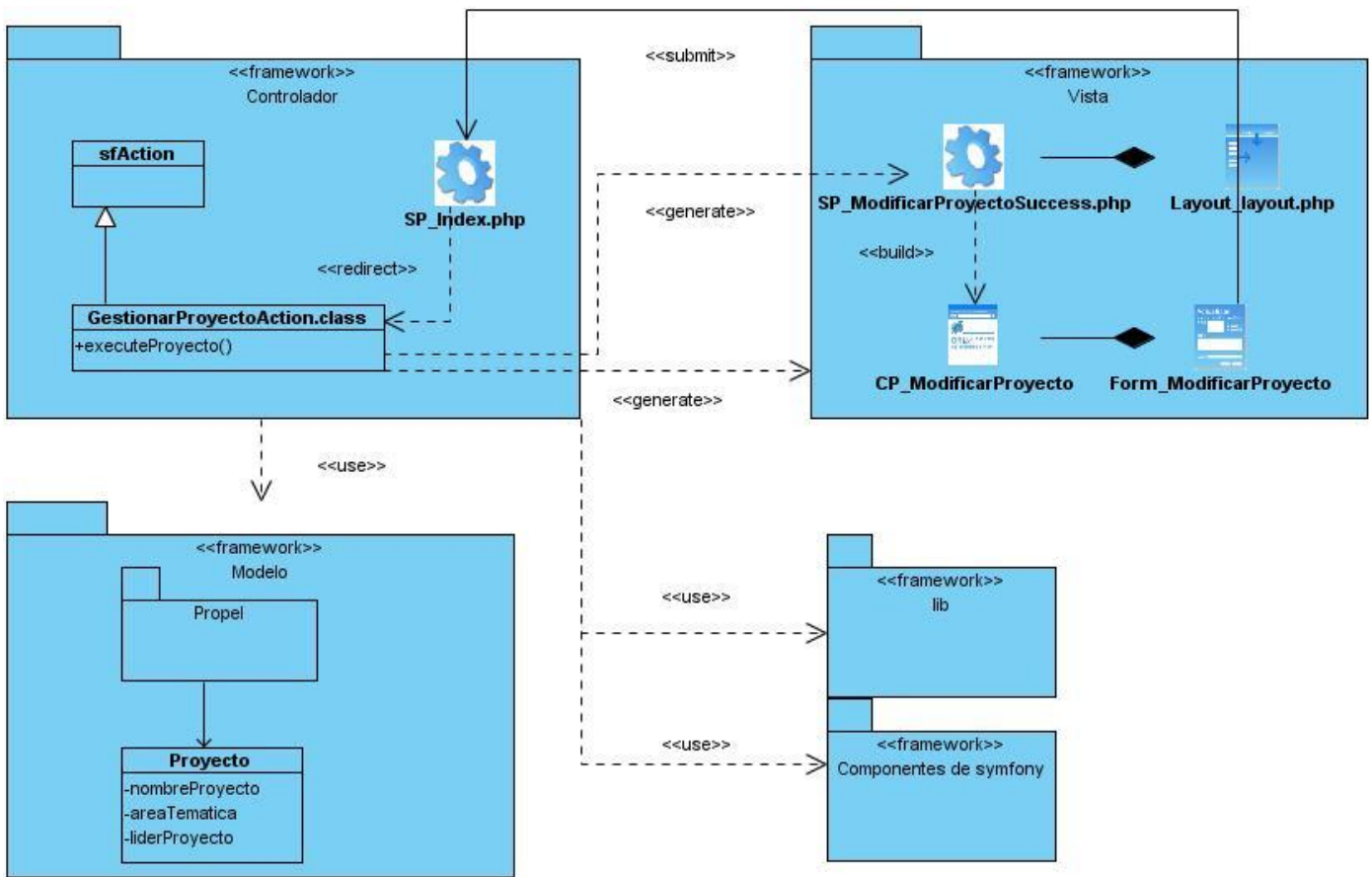


Figura 40: Diagrama de clases del diseño "Modificar proyecto"

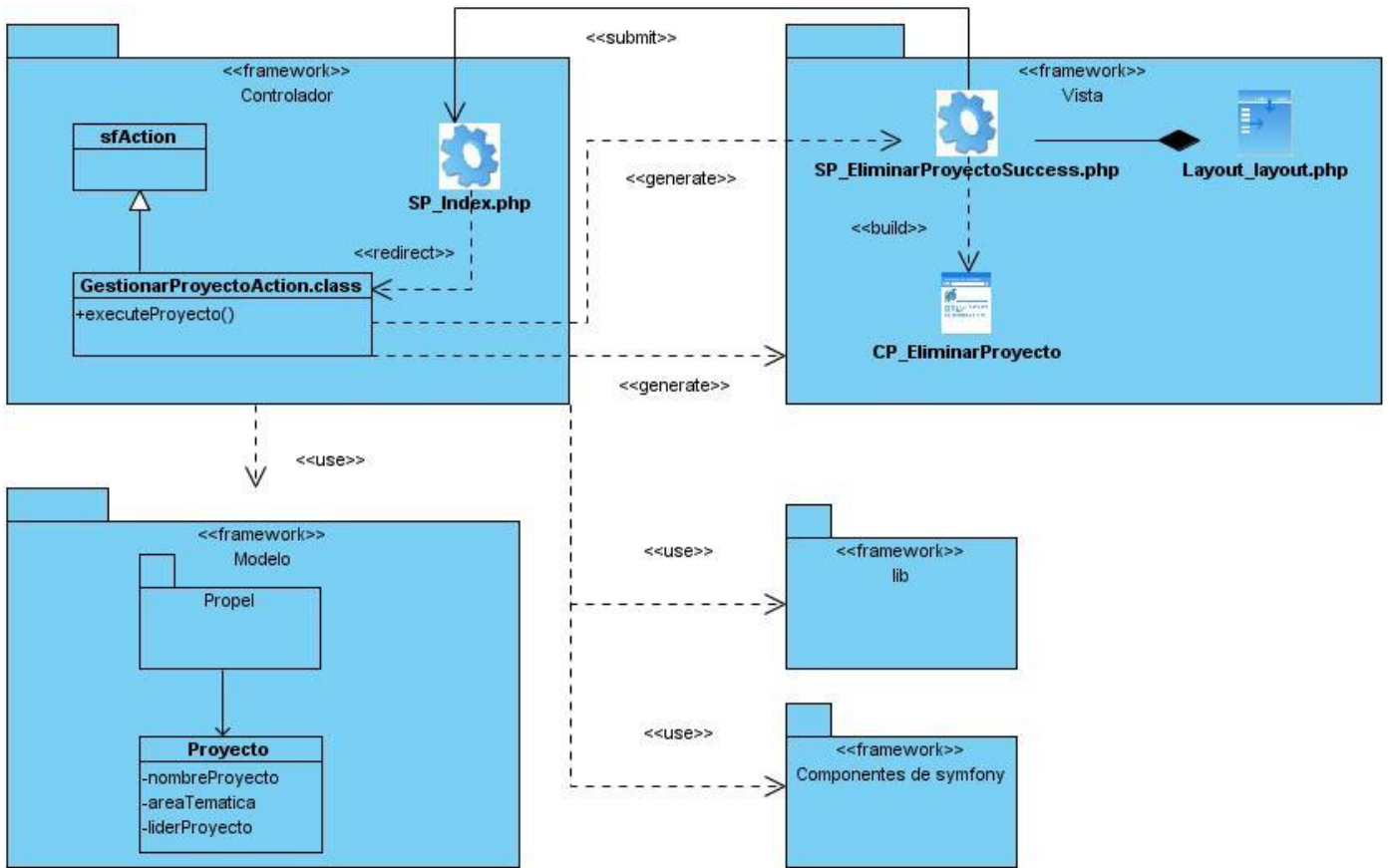


Figura 41: Diagrama de clases del diseño "Eliminar proyecto"

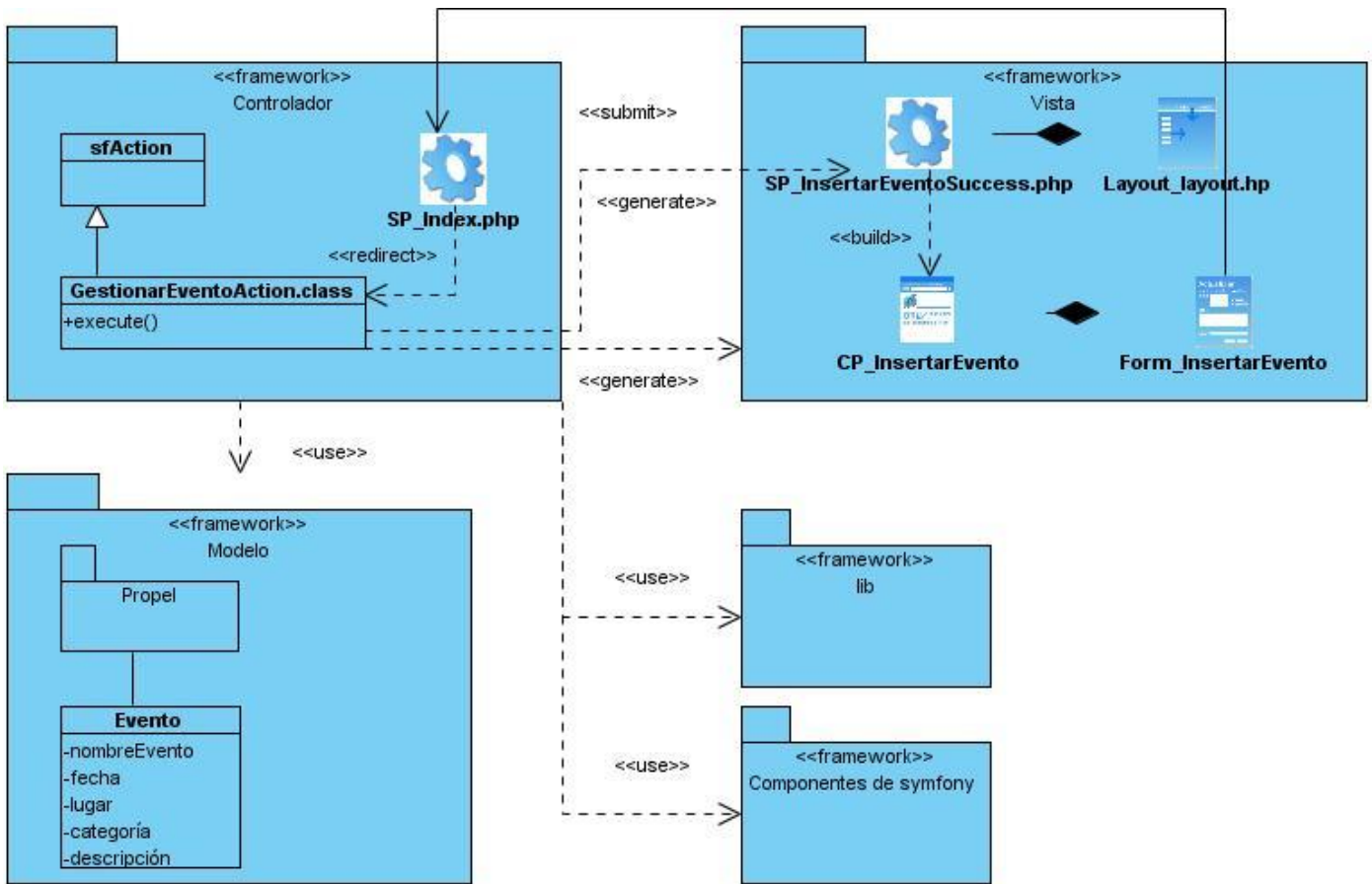


Figura 42: Diagrama de clases del diseño "Insertar evento"

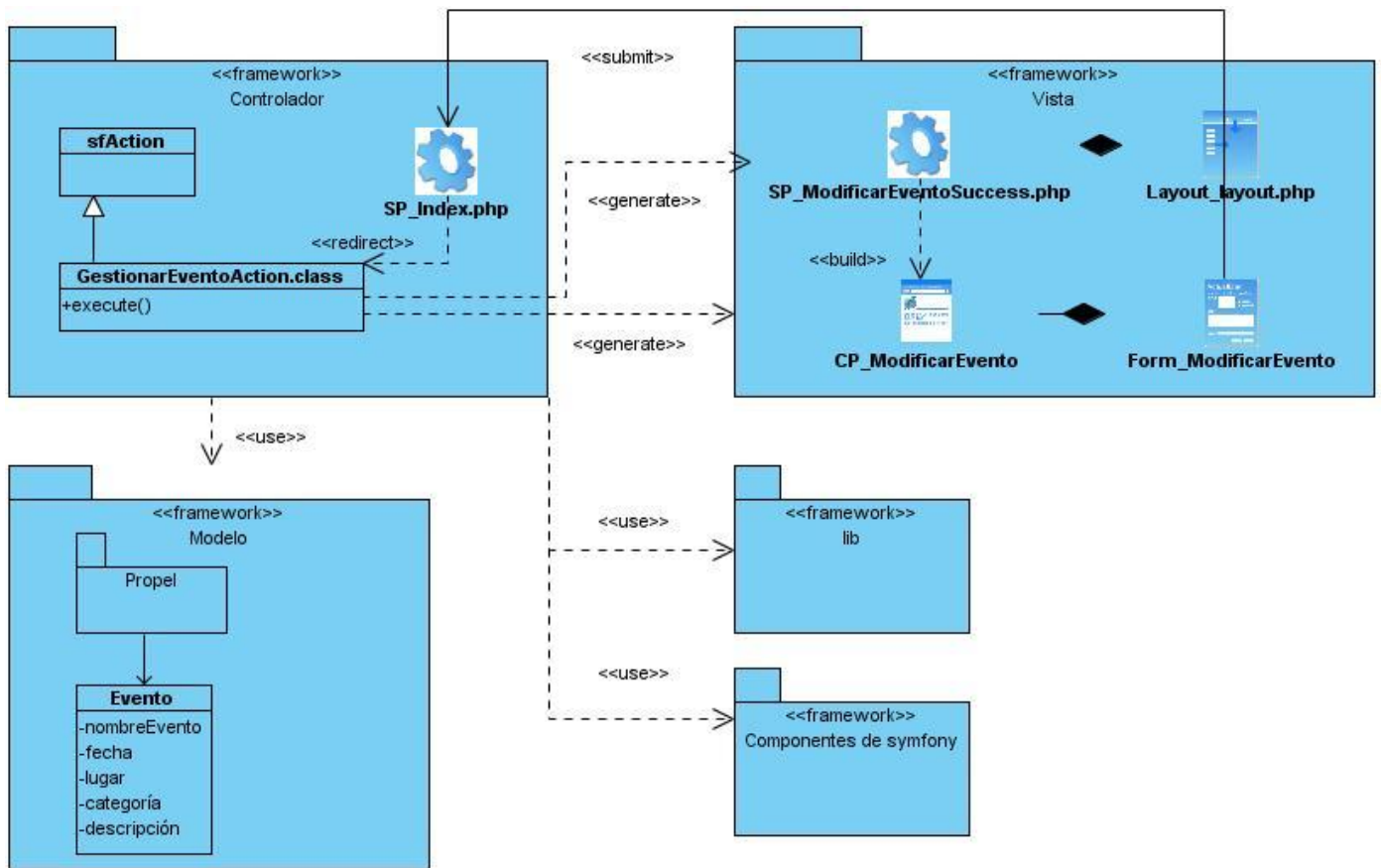


Figura 43: Diagrama de clases del diseño "Modificar evento"

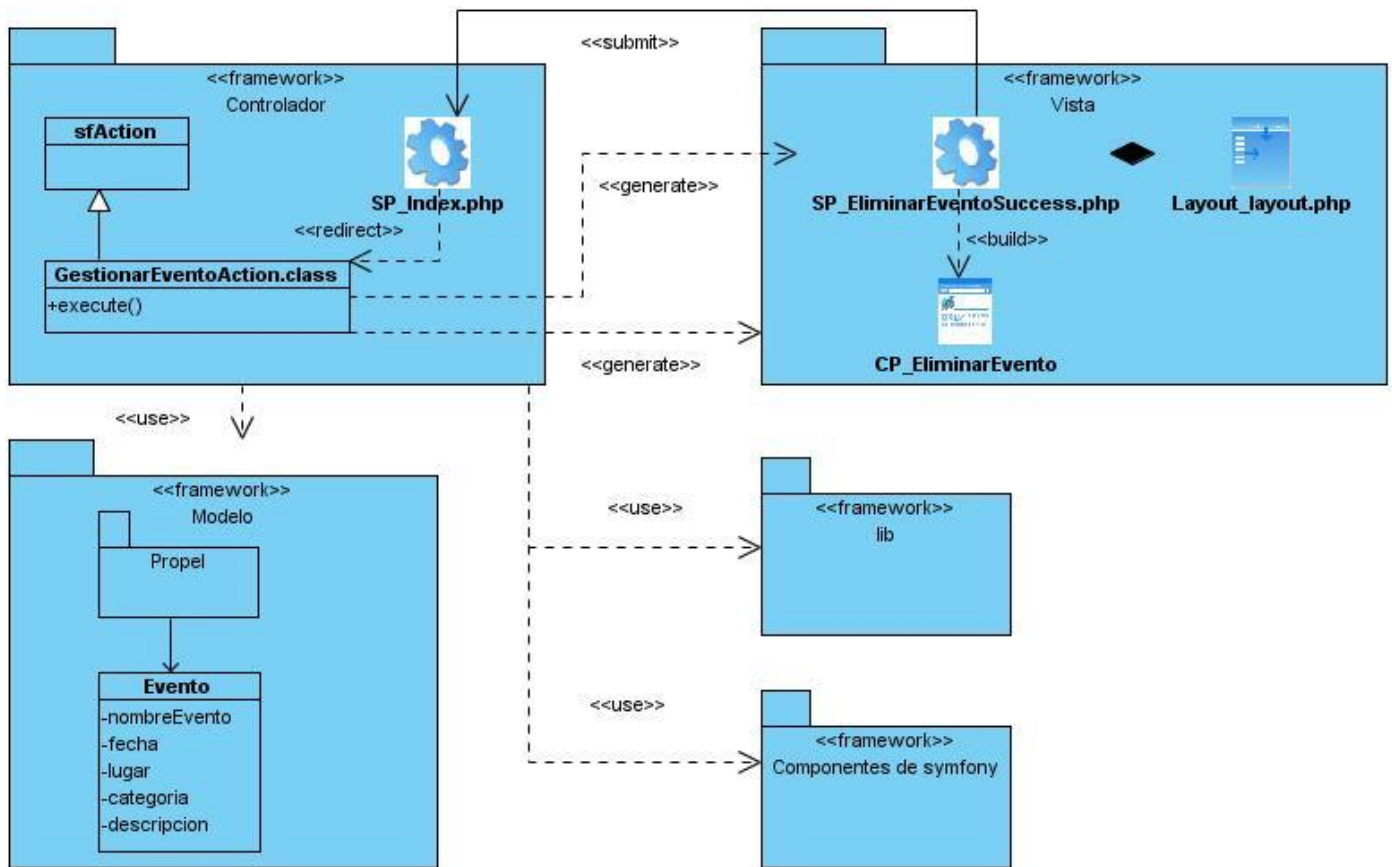


Figura 44: Diagrama de clases del diseño "Eliminar evento"

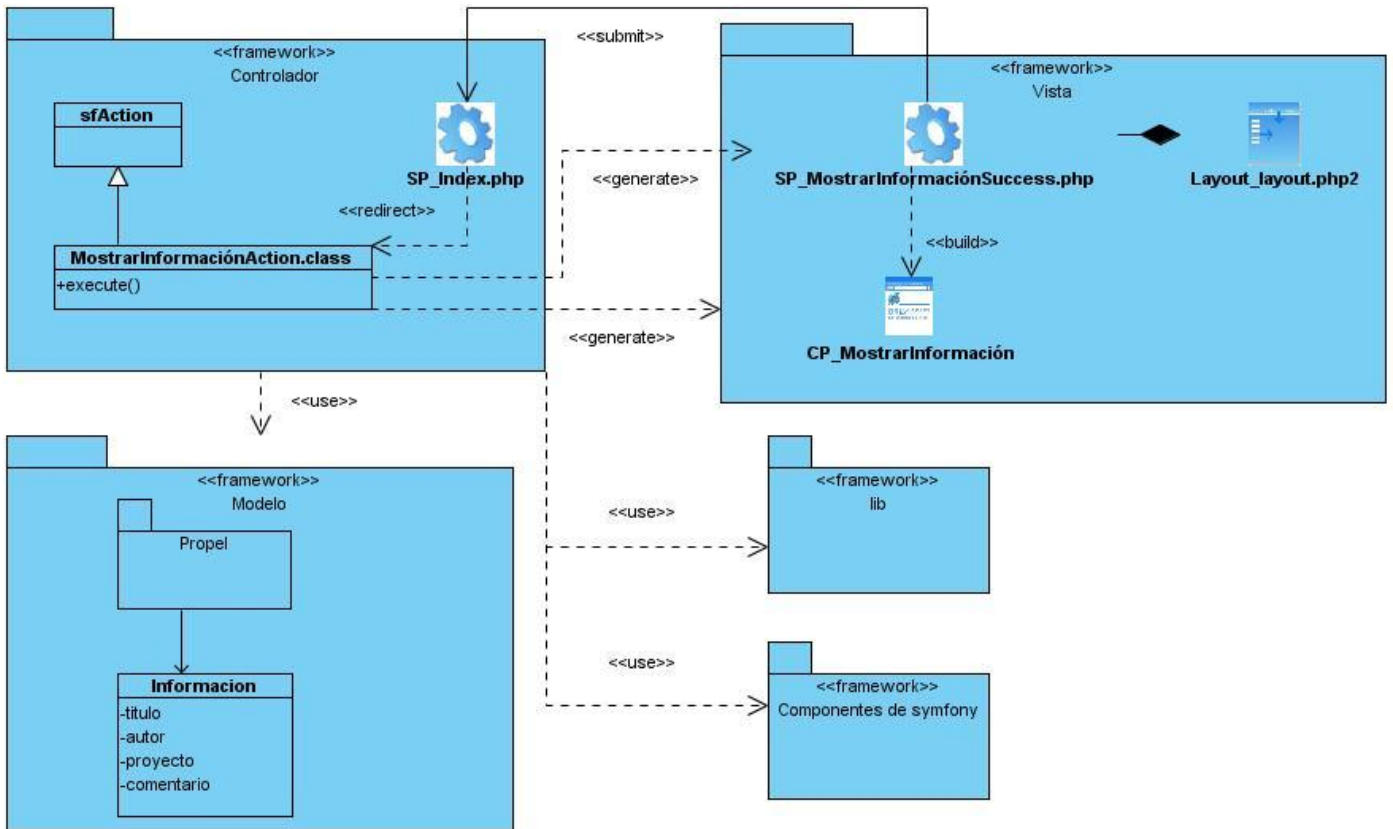


Figura 45: Diagrama de clases del diseño "Mostrar información"

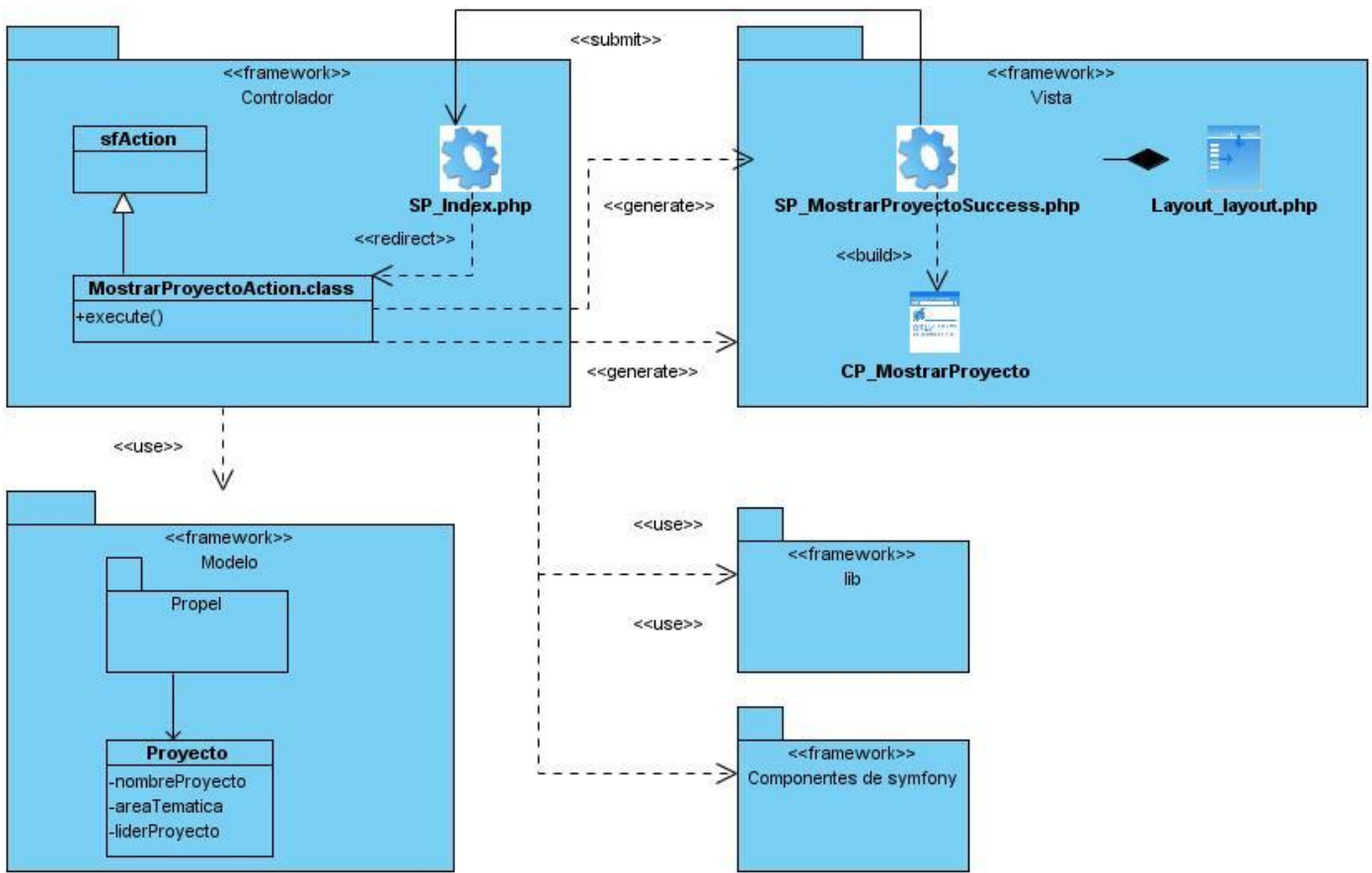


Figura 46: Diagrama de clases del diseño "Mostrar proyecto"

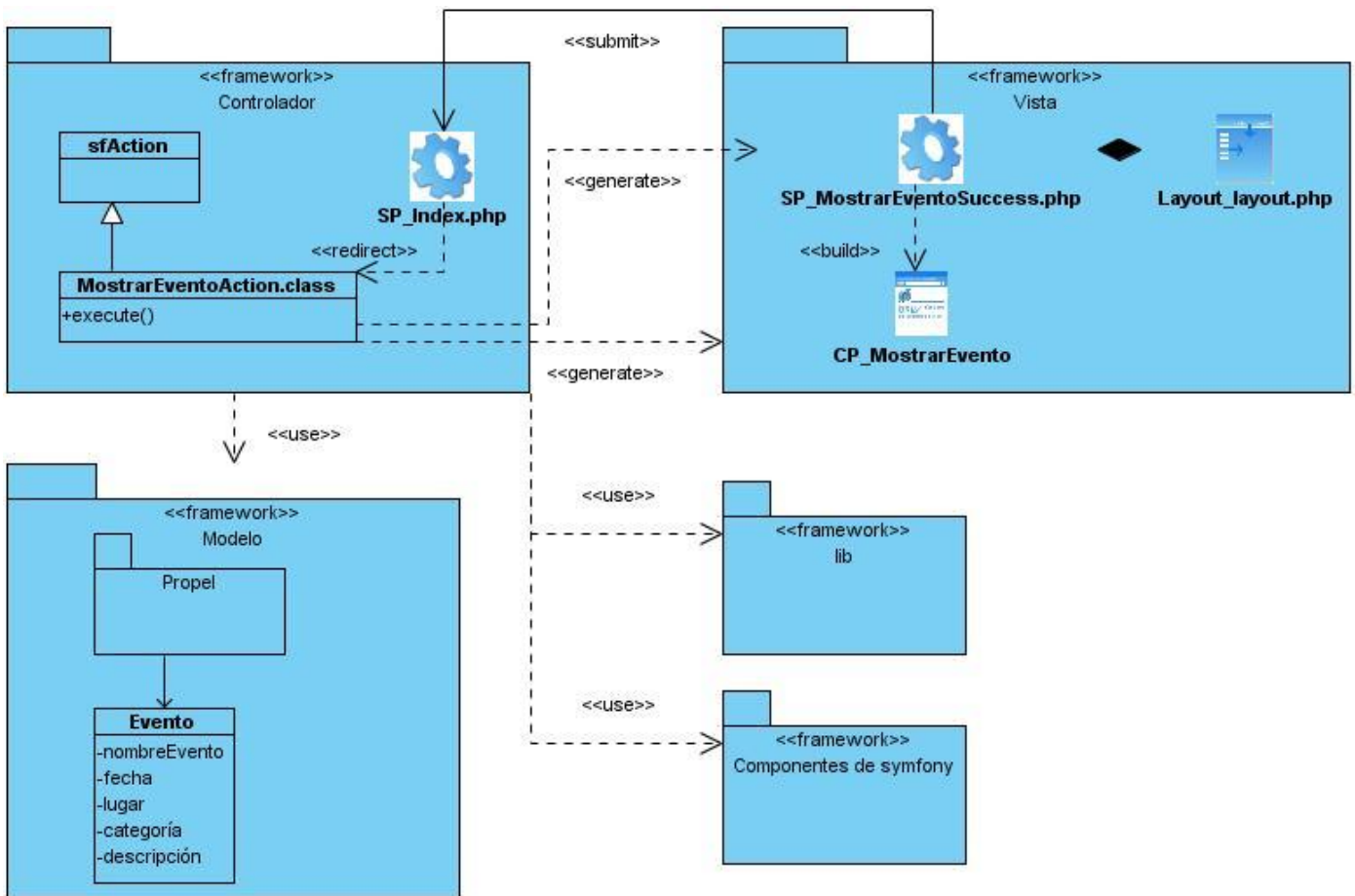


Figura 47: Diagrama de clases del diseño "Mostrar evento"

Anexo 3: Diagramas de secuencia

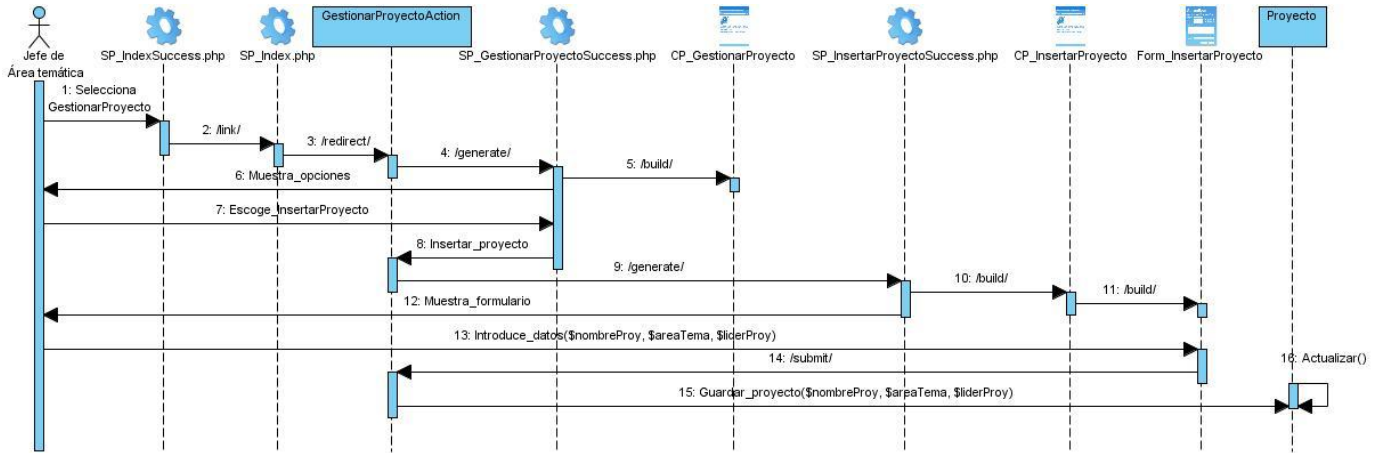


Figura 48: Diagrama de secuencia "Insertar proyecto"

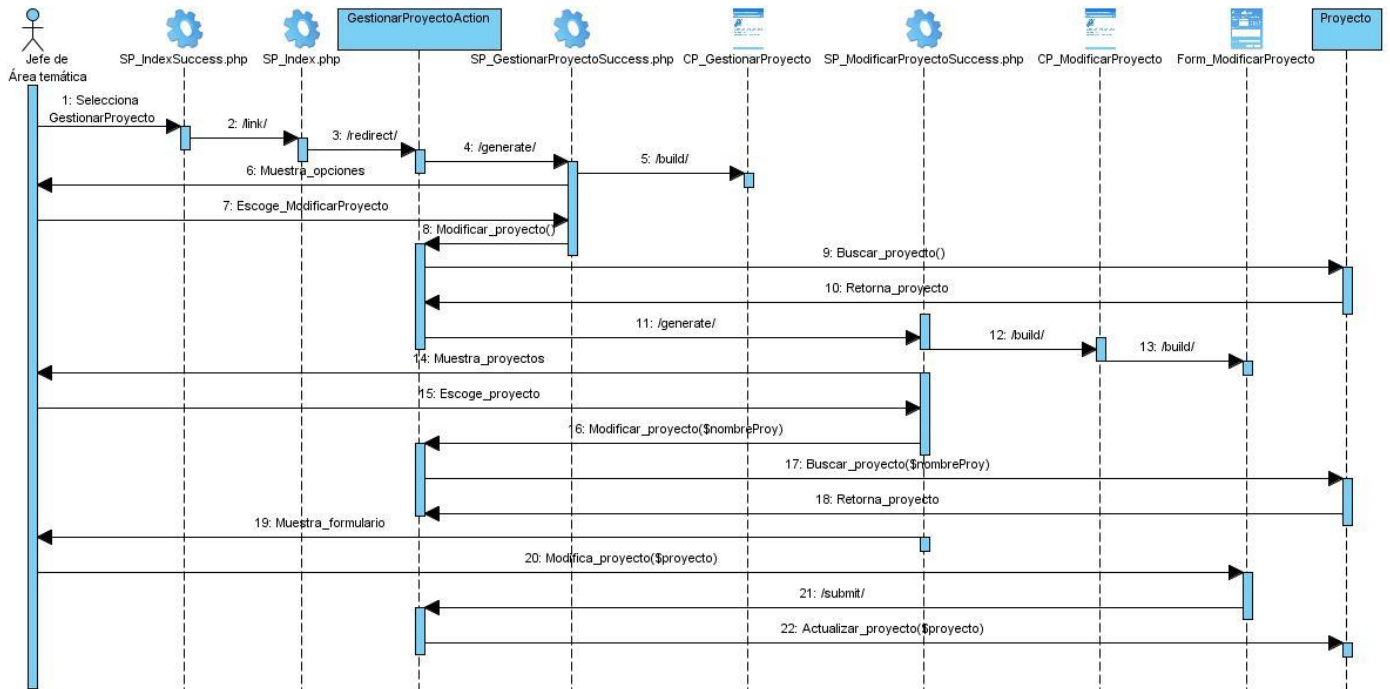


Figura 49: Diagrama de secuencia "Modificar proyecto"

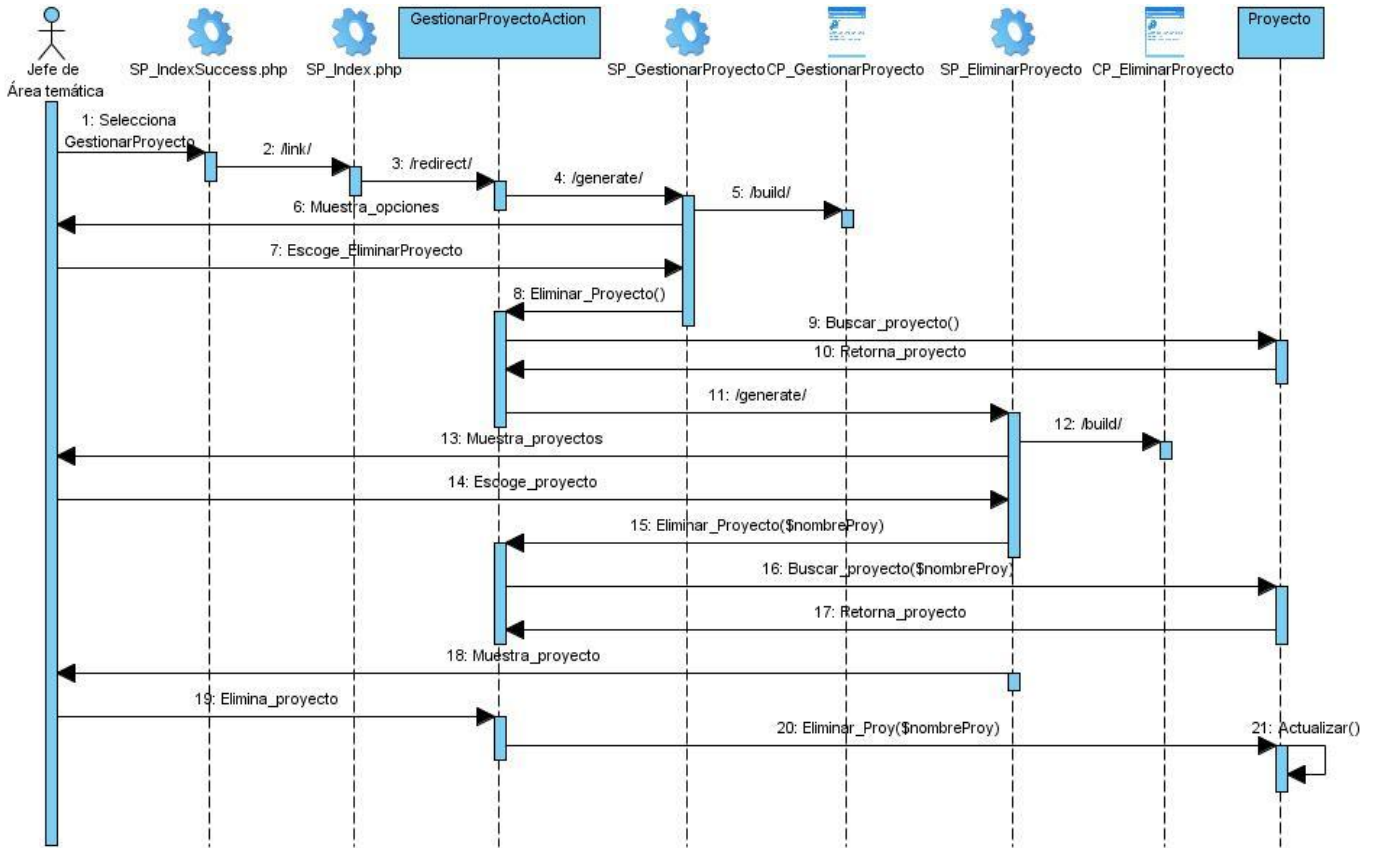


Figura 50: Diagrama de secuencia "Eliminar proyecto"

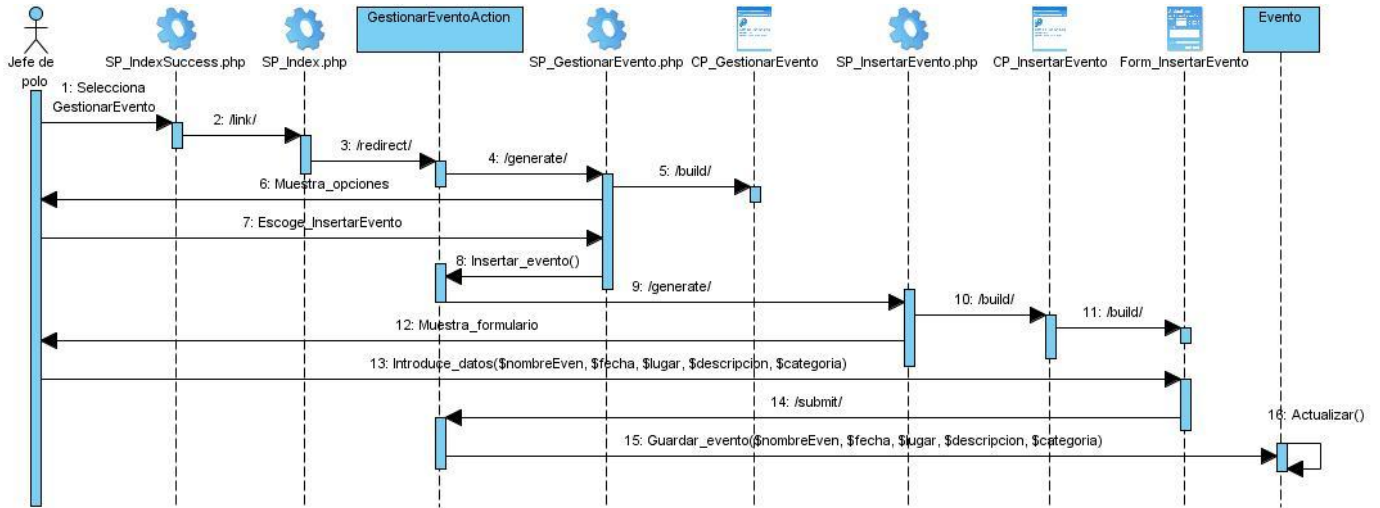


Figura 51: Diagrama de secuencia "Insertar evento"

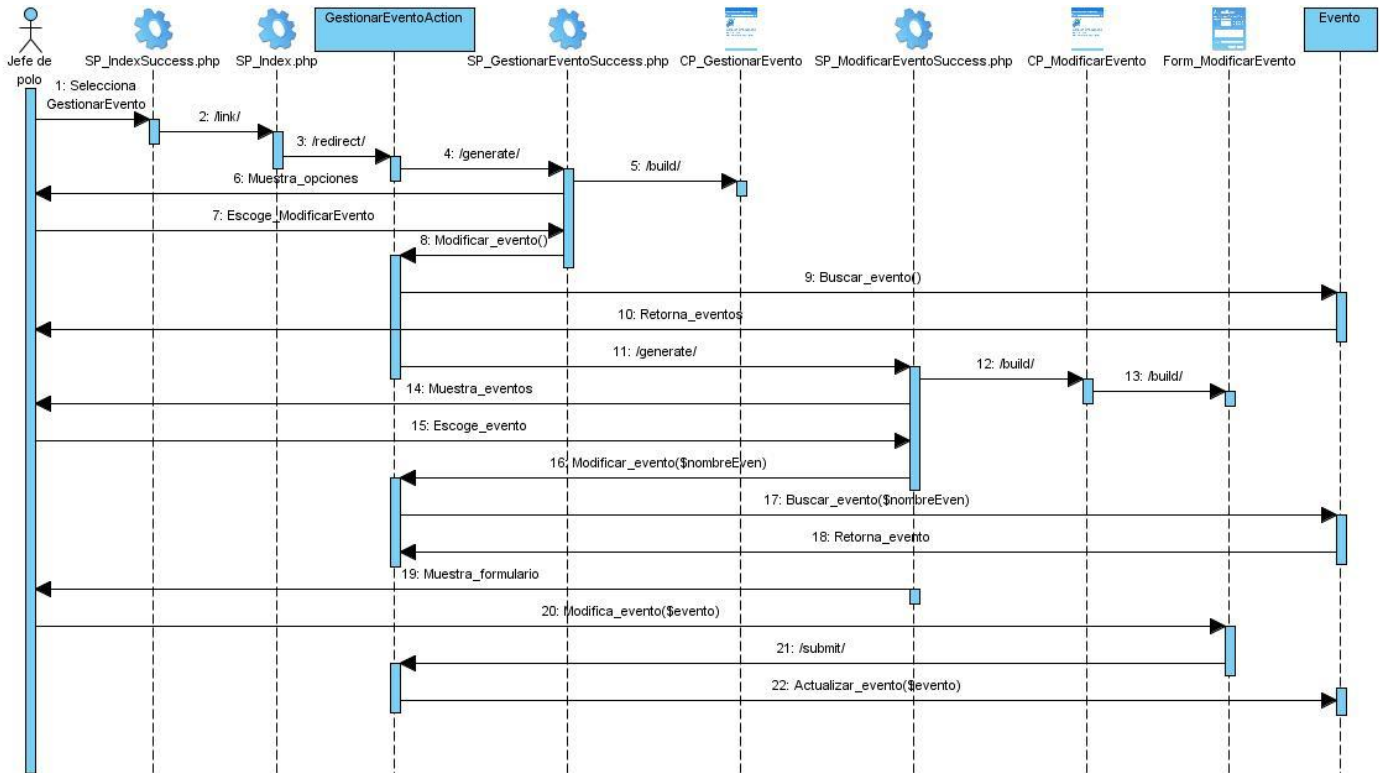


Figura 52: Diagrama de secuencia "Modificar evento"

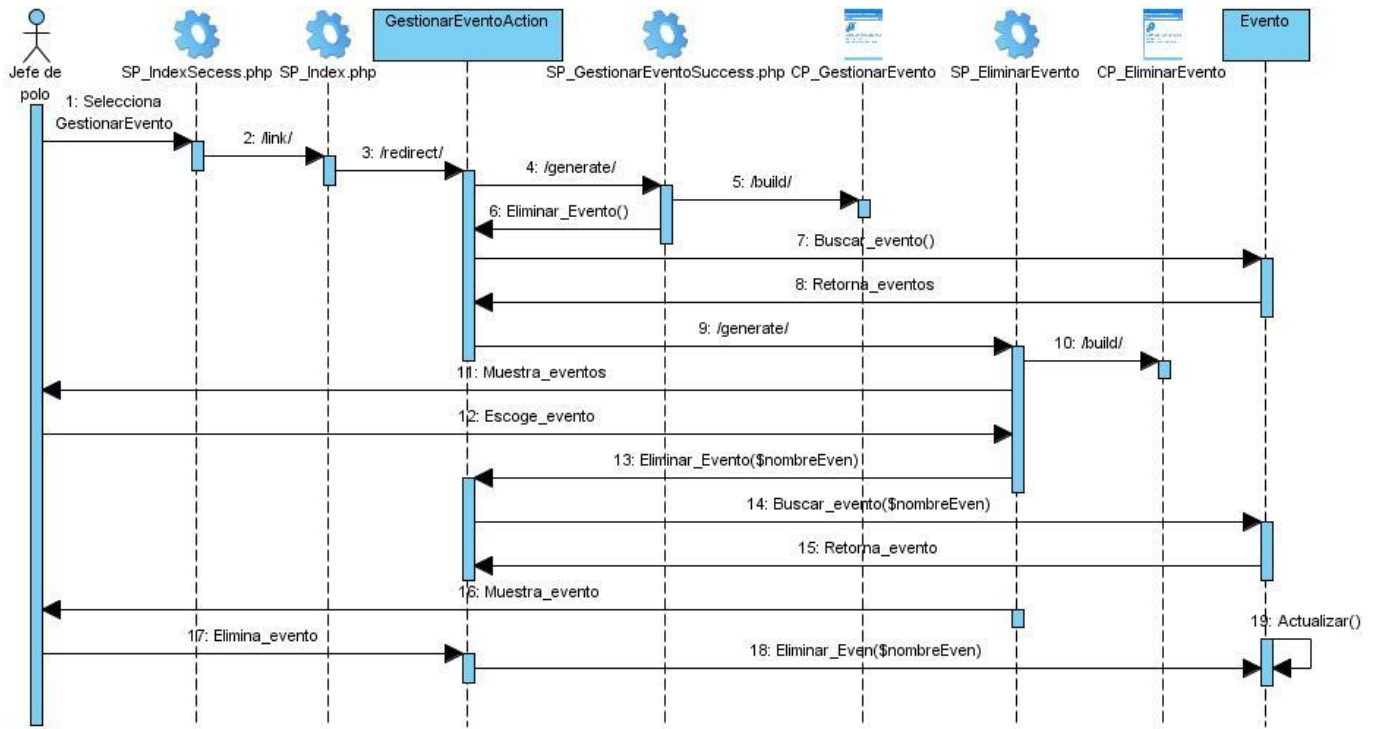


Figura 53: Diagrama de secuencia "Eliminar evento"

Anexo 4: Descripción de clases entidad

Tabla 21: Descripción de la clase entidad Usuario.

Nombre	Usuario
Tipo de clase	Entidad
Atributo	
\$usuario	
\$contraseña	
Para cada responsabilidad	
Nombre:	Buscar_datos(\$usuario, \$contraseña)
Descripción:	Busca el usuario y la contraseña en la base de datos para ver si existen.

Tabla 22: Descripción de la clase entidad Foro.

Nombre	Foro
Tipo de clase	Entidad
Atributo	
\$nombreTema	
\$autor	
\$contenido	
\$respuesta	
Para cada responsabilidad	
Nombre:	Guardar_datos(\$nombreTema, \$ autor, \$ contenido)
Descripción:	Guarda los datos cuando se crea un tema de discusión en el foro.
Nombre:	Actualizar()
Descripción:	Se actualizan los datos de la entidad.
Nombre:	Guardar_respuesta(\$respuesta)
Descripción:	Guarda la respues a un tema cuando se participa en el foro.
Nombre:	Buscar_foro(\$nombreTema)
Descripción:	Busca el tema del foro cuyo nombre es pasado por parámetro.

Tabla 23: Descripción de la clase entidad Información.

Nombre	Información
Tipo de clase	Entidad
Atributo	
\$tituloInf	
\$autor	
\$comentario	
\$proyecto	
Para cada responsabilidad	
Nombre:	Actualizar()

Descripción:	Se actualiza la entidad después de algún tipo de cambio ocurrido.
Nombre:	Guardar_informacion(\$tituloInf, \$autor, \$comentario, \$proyecto)
Descripción:	Guarda la nueva información subida al sitio web.
Nombre:	Buscar_informacion(\$tituloInf)
Descripción:	Busca la información cuyo titulo es pasado por parámetro.
Nombre:	Actualizar_informacion(\$informacion)
Descripción:	Cuando se modifica algún dato de la información se le pasa por parámetro un objeto información y este se actualiza.
Nombre:	Eliminar_inf(\$tituloInf)
Descripción:	Elimina la información de la cual el titulo es pasado por parámetro.

Tabla 24: Descripción de la clase entidad Proyecto.

Nombre	Proyecto
Tipo de clase	Entidad
Atributo	
	\$nombreProy
	\$areaTematica
	\$liderProy
Para cada responsabilidad	
Nombre:	Actualizar_proyecto(\$proyecto)
Descripción:	Se actualiza el proyecto después de habersele modificado algún dato.
Nombre:	Buscar_proyecto(\$nombreProy)
Descripción:	Busca el proyecto del cual el nombre sea \$nombreProy.
Nombre:	Guardar_proyecto(\$nombreProy, \$areaTematica, \$liderProy)
Descripción:	Guarda los datos cuando es insertado un nuevo proyecto.
Nombre:	Eliminar_proy(\$nombreProy)
Descripción:	Elimina el proyecto cuyo titulo coincide con el pasado por parámetro.
Nombre:	Actualizar()
Descripción:	Se actualiza la entidad proyecto después de algún cambio efectuado.

Tabla 25: Descripción de la clase entidad Evento.

Nombre	Evento
Tipo de clase	Entidad
Atributo	
\$nombreEven	
\$fecha	
\$lugar	
\$descripcion	
\$categoria	
Para cada responsabilidad	
Nombre:	Guardar_evento(\$nombreEven, \$fecha, \$lugar, \$descripcion, \$categoria)
Descripción:	Guarda en la base de datos el nuevo evento insertado con todos sus datos.
Nombre:	Actualizar_evento(\$evento)
Descripción:	Actualiza el evento que ha sido modificado.
Nombre:	Buscar_evento(\$nombreEven)
Descripción:	Busca el evento que su nombre coincida con \$nombreEven.
Nombre:	Eliminar_even(\$nombreEven)
Descripción:	Elimina el evento que tenga el nombre pasado por parámetro.
Nombre:	Actualizar()
Descripción:	Se actualiza la entidad evento después de los cambios efectuados.

GLOSARIO

Ajax: Acrónimo de Asynchronous JavaScript And XML. Es una técnica de desarrollo Web para crear aplicaciones interactivas.

Aplicación Web: Es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet.

Arquitectura: La arquitectura de computadoras es el diseño conceptual y la estructura operacional fundamental de un sistema de computadora.

Base de Datos (BD): También conocida como banco de datos es un conjunto de datos pertenecientes al un mismo contexto y almacenados sistemáticamente para su posterior uso.

Código abierto: Cualidad de algunos software de incluir el código fuente en la distribución del programa. En general se usa para referirse al software libre.

Clúster: Conjuntos o conglomerados de computadoras construidas mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Hoy en día juegan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Hardware: Es todo componente físico involucrado en el funcionamiento de un equipo informático.

HTTP: (Hiper-Text Transfert Protocol, Protocolo de Transferencia de Hipertexto), es usado para solicitar y recibir contenidos tales como ficheros, imágenes, flujos binarios, etc.

Interfaz: Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.

Lenguaje de Programación: Es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Plataforma: En informática, una plataforma es precisamente el principio, en el cual se constituye un hardware, sobre el cual un software puede ejecutarse/desarrollarse.

PHP: (Hypertext Preprocessor). Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

PostgreSQL: Es un motor de base de datos, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

RAM: (Random Access Memory, memoria de acceso aleatorio) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados. Es el área de trabajo para la mayor parte del software de un computador.

RUP: (Rational Unified Process, Proceso Unificado de Rational) es un proceso de desarrollo de software y junto con el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sistemas Gestores de Base de Datos (SGBD): 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.

Software: Programas, procedimientos y reglas para la ejecución de tareas específicas en un sistema de cómputo.

Software Libre: Aplicaciones informáticas que pueden ser libremente copiadas, distribuidas, estudiadas y modificadas por el usuario, según los parámetros establecidos por su creador.

UML: (**U**nified **M**odeling **L**anguage, Lenguaje Unificado de Modelo) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

URL: (**U**niform **R**esource **L**ocator, localizador uniforme de recurso). Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización. Es una dirección web.

Web: Sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.