

Universidad de las Ciencias Informáticas
Facultad 7



Título: *Gestión del Evento Científico*
NeuroRehabana 2010

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Ramón A. Sánchez Santos

Juan Pablo Rodríguez Figueroa

Tutor: Ing. Alexis Turruella

Ing. Yisel Reyes Cardoso

Ciudad de La Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, con carácter exclusivo.

Ramón A. Sánchez Santos

Juan Pablo Rodríguez Figueroa

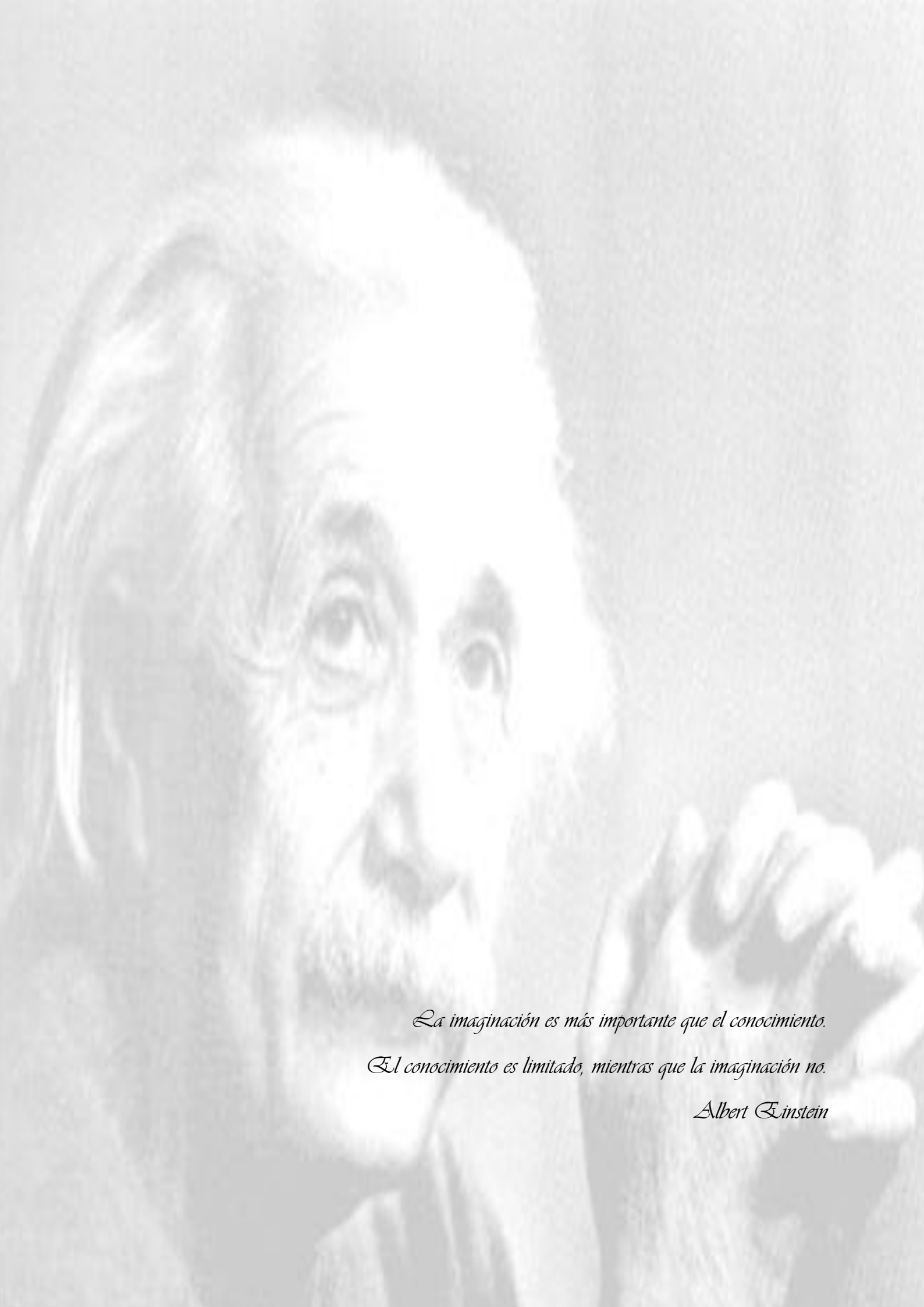
Ing. Alexis José Turruella Sánchez

Ing. Yisel Reyes Cardoso

Datos de contacto

Ing. Yisel Reyes Cardoso (ycardoso@uci.cu): Recién graduada con Título de Oro de Ingeniera en Ciencias Informáticas en la UCI. Posee Categoría Docente de Profesor Adiestrado. Ha participado en procesos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Imparte la asignatura de Álgebra Lineal. Actualmente se desempeña como líder y analista principal del proyecto Colaboración Médica del Área Temática Sistemas de Apoyo a la Salud.

Ing. Alexis José Turruella Sánchez (ajturruella@uci.cu): Recién graduado en Ciencias Informáticas en la UCI. Posee Categoría Docente de Profesor instructor. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Imparte la asignatura de Programación WEB y Gráfico por computadora. Actualmente se desempeña como miembro del grupo de Desarrollo de Negocio del Área Temática Sistemas de Apoyo a la Salud.



*La imaginación es más importante que el conocimiento.
El conocimiento es limitado, mientras que la imaginación no.*

Albert Einstein

A mis padres **Susana Sofía Santos Maury** y **Ramón Santiago Sánchez Santos**, y a mi hermana **Susana Llama Santos** por el apoyo, el ejemplo y por saberme guiar siempre por el camino correcto.

A mi abuela **Ana Luisa Maury Yumar** por haberme criado y estar siempre para mí, tú eres y serás siempre mi luz en la oscuridad, mi faro y mi apoyo, para ti va este triunfo.

A su entender para la persona más importante del mundo, "la vieja", **Margarita Sánchez Blanco**, siempre te tendré presente.

A mis amigos y hermanos de Camagüey **Carrasco, el enano, Tamayo, el Capo** y otros que no se encuentra aquí que les debo mucho en mi carrera y en la vida.

A mi tutores, por ayudarme siempre que lo necesitaba.

A mi hermano y compañero de tesis, **Juan Pablo**.

A mis hermanos que vienen conmigo desde primer año **San Juan, el Gordo, la marmota, Rosmery, la vaca** y otros, por toda la ayuda que me han prestado.

A los profesores **Ramos y Rotse** por su ayuda incondicional.

A todo el que de una forma u otra ha hecho posible la realización de este sueño.

Ramón.

A mi **mamá** y mi **papá**, porque les debo todo lo que soy y no me alcanzarían las palabras para poder decirles todo lo que siento por ustedes y todo lo que les agradezco.

A **Cusita, Danay y familia**, porque me han tenido como si fuera su hijo y para mi son parte de mi familia.

A mi hermano y compañero de tesis, **Ramón**.

A mis hermanos de la zona, **Jose, Yáriek y Nelsito**.

A mis amigos de la universidad, en especial **San Juan, Jorgito, Raúl, La marmota, Ale, Alberto, Leshter, Félix, Karel** entre otros.

A mi **tutores**, por ayudarme siempre que lo necesitaba.

A **Ramos** por su importantísima ayuda.

A mis vecinos y amigos del barrio.

A todas las personas que de una forma u otra hicieron este sueño realidad.

Juan Pablo.

A nuestros padres, hermanos, familiares y amigos...

Resumen

Con el desarrollo de las tecnologías a nivel mundial, la utilización de aplicaciones web y portales para empresas, instituciones sociales y demás entidades, se ha convertido en una necesidad. Ya que les permite entregar información actualizada a sus clientes. El presente trabajo de diploma tiene como objetivo el desarrollo de un portal web que permita gestionar toda la información científica, publicitaria y los recursos humanos del evento NeuroRehabana 2010, para optimizar el proceso organizativo del mismo.

Para el desarrollo de la aplicación se utilizó el lenguaje de programación PHP en su versión 5.2 y como metodología de desarrollo, RUP, que rige todo el proceso de ingeniería de software. Apoyándose en la herramienta Enterprise Architect y el lenguaje UML 2.0 para la generación de los diferentes artefactos que propone esta metodología. El sistema se implementó bajo la arquitectura Modelo Vista Controlador (MVC).

A partir del uso de esta aplicación se facilitará la organización y gestión de los recursos del evento científico NeuroRehabana 2010. Los usuarios podrán contar con un portal que les proporcione la información relacionada con los trabajos presentados, participantes y cronogramas de actividades del mismo. Además, tendrán la posibilidad de descargar materiales distribuidos por el administrador, así como realizar diferentes reportes dinámicos en formato PDF para futuros análisis.

Palabras claves:

Aplicaciones web, portal web, información, organización, gestión, recursos.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	4
1.1 Conceptos asociados al problema.....	4
1.1.1 Aplicaciones Web	4
1.1.2 Portal Web.....	4
1.2 CMS	5
1.2.1 Necesidad de un CMS	6
1.3 Aplicaciones web de gestión de eventos	8
1.3.1 Internacional	8
1.3.1.1 ConfWiz.....	8
1.3.1.2 Gestkal XXI.....	8
1.3.1.3 Eventtual	8
1.3.2 Nacional	8
1.3.2.1 Eventos en Cuba	9
1.3.2.2 Portal NeuroRehabana2008	9
1.3.3 UCI.....	9
1.4 Tendencias, tecnologías y metodologías actuales a considerar.....	9
1.4.1 Servidor Web.....	10
1.4.2 Lenguajes de programación web y protocolos	10
1.4.2.1 HTML.....	10
1.4.2.2 PHP	11
1.4.2.3 JavaScript.....	12
1.4.2.4 CCS.....	12
1.4.2.5 XML.....	13
1.4.2.6 SOAP (Simple Object Access Protocol).....	13
1.4.3 Framework.....	14
1.4.3.1 Zapatec	14
1.4.4 Gestor de base de datos (SGBD)	14
1.4.4.1 MySQL	15
1.4.5 Arquitectura de Software	16
1.4.5.1 Modelo Vista Controlador (MVC)	16
1.4.5.2 Arquitectura Orientada a Servicios (SOA).....	17

1.4.6	Desarrollo basado en RUP	17
1.4.7	Tecnologías de desarrollo.....	18
1.4.7.1	AJAX	18
1.4.8	Servicios Web.....	19
1.4.9	Herramientas	19
1.4.9.1	Embarcadero ER/Studio 7.0	19
1.4.9.2	Zend Studio 6.1.1	20
1.4.9.3	Macromedia Dreamweaver:.....	20
1.4.9.4	Enterprise Architect	21
1.4.9.5	EMS MySQL Manager.....	21
1.4.9.6	CMS a Utilizar.....	21
Capítulo 2: Características del sistema.....		23
2.1	Modelo de Dominio	23
2.1.1	Conceptos fundamentales	23
2.1.2	Modelo de dominio	24
2.2	Captura de requisitos.....	25
2.2.1	Requerimientos funcionales.....	25
2.2.2	Requerimientos no funcionales del sistema	28
2.3	Modelo de sistema	29
2.3.1	Definición de los actores del sistema.....	29
2.3.2	Diagrama de casos de uso del sistema	30
2.4	Descripción textual de los casos de uso críticos del sistema.....	31
Capítulo 3: Diseño y construcción del sistema		38
3.1	Modelo de diseño	38
3.1.1	Patrones de diseño utilizados	38
3.1.1.1	Bajo acoplamiento	38
3.1.1.2	Alta cohesión	39
3.1.2	Estructura del diseño	39
3.1.3	Diagrama de clases del diseño con estereotipos web.....	40
3.1.4	Descripción de las clases.	45
3.1.5	Diagramas de interacción	50
3.1.6	Diseño de la base de datos	50
3.1.7	Descripción de tablas y atributos	51

3.2 Principios de diseño gráfico y estándares de codificación	60
3.2.1 Principios de diseño gráfico	60
3.2.2 Estándares	61
3.2.2.1 Identación.....	61
3.2.2.2 Comentarios	61
3.2.2.3 Líneas en blanco	62
3.2.2.4 Clases y Objetos.....	62
3.3 Modelo de implementación	62
3.3.1 Diagrama de componentes	63
3.4 Diagrama de despliegue	66
Conclusiones	68
Beneficios	69
Recomendaciones	70
Referencias Bibliográficas	71
Bibliografía	72
Anexos	74
Glosario de Términos	82

Introducción

En la actualidad no se puede hablar de educación, cultura y desarrollo si no se aplican eficientemente y en todo el ámbito social, las Tecnologías de la Información y Comunicaciones (TIC). Cuba tiene como proyecto de desarrollo alcanzar la informatización de su sociedad. Para ello ha diseñado e iniciado la aplicación de tácticas que permiten convertir los conocimientos y las TICs en herramientas a disposición del avance y las profundas alternativas revolucionarias. Actualmente se sigue perfeccionando el trabajo y ampliando el radio de acción de las nuevas tecnologías en beneficio de todo el país.

Existen varias empresas cubanas de producción de software que gestionan prácticamente todo el desarrollo informático. Entre ellas: la Empresa Nacional de Software DESOFT y la Empresa de Soluciones Informáticas (SOFTEL). La sociedad cubana ha estado inmersa en un profundo proceso de transformaciones, denominado Batalla de Ideas. Al calor de ésta, surge la Universidad de las Ciencias Informáticas (UCI), en el año 2002, con un novedoso programa de formación. El cual combina la producción de software y la docencia, aportando diferentes soluciones informáticas a empresas dentro y fuera de la isla.

La UCI tiene una estructura productiva orientada a informatizar diferentes ramas de la economía y la sociedad cubana. La Facultad 7, como parte de la estructura docente-productiva-investigativa de la UCI, mantiene un perfil dirigido a la informatización del sector de la salud. Teniendo en cuenta el número de proyectos que lo anterior ha generado, fue necesario agruparlos en una estructura llamada Áreas temáticas. Entre ellas, la de Sistema de Apoyo a la Salud (SAS), que se encarga del desarrollo de software especializado en actividades no asistenciales, es decir, no ligadas directamente con la atención médica a pacientes.

Las afecciones neurológicas son causas fundamentales de discapacidad. Esta se traduce en una persona con necesidades especiales que se expresan como una amplia gama de deficiencias o desventajas que afectan el equilibrio y la integridad bio-psico-social del individuo y sus relaciones con el entorno, provocando importantes repercusiones en las estructuras dinámicas de la sociedad. Esta relación de acontecimientos ofrece plena justificación para abordar la discapacidad en toda su dimensión y dedicar urgentes esfuerzos a la búsqueda de posibles recursos, en primer lugar, para prevenir su aparición y en segundo lugar, una vez presente, reducir al mínimo sus efectos.

La neurorehabilitación es una disciplina médica integrada por un equipo interdisciplinario (médicos neurólogos, fisiatras, terapeutas físicos, del lenguaje, psicopedagogos) con una visión de tratamiento selectivo de acuerdo a los diferentes métodos clásicos y al ejercicio físico terapéutico. El programa de neurorehabilitación es un proceso terapéutico estructurado como programa altamente individualizado. Específicamente, desarrollado para resolver las necesidades del paciente después de lesiones del cerebro u otro tipo de origen neurológico. Es por esta razón, que la prevención de la discapacidad de causa neurológica, se asume como el lema central del Evento NeuroRehabana 2010.

Especialistas de distintas latitudes del planeta se reúnen en este evento para publicar y exponer diversos trabajos científicos relacionados con éste tema. Estos trabajos son de vital importancia para la comunidad internacional de lucha contra las afecciones neurológicas debido a que constituyen una fuente de material de estudio. Es una información valiosa, que constituye una importante base bibliográfica para todas aquellas personas (especialista o no) que se acercan a este encuentro. Las que pueden aumentar sus conocimientos mediante la consulta de esta información en el momento requerido.

En la pasada edición se desarrolló un portal web para la gestión de la información de este evento. El mismo no resuelve las exigencias actuales estipuladas por el colectivo organizador, debido a que la información se encuentra solamente de manera estática y brinda a los usuarios solo algunos aspectos. En la actualidad toda la información de este evento de envergadura internacional y con un alto nivel científico se gestiona de forma manual, provocando pérdida de información valiosa para la prevención y reducción de daños de enfermedades neurológicas, así como un arduo trabajo por parte de la comisión organizadora, que en víspera del mismo trabaja hasta altas horas de la noche para ultimar detalles organizativos y de gestión de personal del evento.

El evento se encuentra en su segunda edición y no posee un historial de los trabajos presentados con anterioridad, lo que ha provocado la pérdida de muchos materiales con un alto valor como estudios complementarios por parte del personal invitado. Lo antes planteado, es consecuencia de que no existe un mecanismo que gestione toda la información de este encuentro.

En este sentido, el **Problema a resolver** es: ¿Cómo facilitar el manejo de la información científica, publicitaria y los recursos humanos que intervienen en el evento científico NeuroRehabana 2010?

Se plantea como **Objeto de estudio** el Proceso de gestión de información en eventos científicos. Mientras que el **campo de acción** es el Proceso de gestión de la información científica, publicitaria y los recursos humanos que intervienen en el evento científico NeuroRehabana 2010.

Objetivo General: Desarrollar una aplicación WEB que permita gestionar la información relacionada con el evento científico NeuroRehabana 2010.

Tareas de la Investigación:

1. Analizar las características de diferentes eventos científicos, haciendo énfasis en el Evento Científico NeuroRehabana 2010.
2. Investigar las características y servicios de un Portal Web.
3. Seleccionar las herramientas a utilizar.
4. Asimilar las tecnologías (PHP, XHTML, CMS JOOMLA, JavaScript y SOAP) y la arquitectura Modelo-Vista-Controlador.
5. Realizar Modelo de Sistema.
6. Realizar modelo de análisis y diseño.
7. Implementar el prototipo funcional.
8. Brindar Servicios WEB a otras instituciones o aplicaciones web interesadas en las informaciones del evento científico NeuroRehabana 2010.

El documento de tesis se encuentra estructurado en tres capítulos. En el Capítulo 1 Fundamentación Teórica, se describe el contexto de las nuevas tecnologías de la informática, las tendencias actuales de herramientas y tecnologías que serán utilizadas para el desarrollo de la aplicación.

En el Capítulo 2 Características del sistema, se abordan los elementos fundamentales de la arquitectura sobre la que se desarrolla el sistema, se detallan muy bien el objeto de estudio y el negocio.

El capítulo 3 Diseño y construcción del sistema, se centra en la modelación detallada y la construcción de la estructura de la aplicación. Así como también se fundamentan las diferentes funcionalidades exigidas por el cliente.

Capítulo 1: Fundamentación Teórica

El presente capítulo tiene como objetivo abordar los elementos que fundamentan la base teórica conceptual de la gestión de distintos eventos. Lo que permite la realización, en este caso, de la gestión del evento científico NeuroRehabana 2010. Se hace alusión además, a algunas aplicaciones que de una forma u otra gestionan eventos en Cuba y en la UCI, así como también a nivel internacional. Además se realiza un análisis de las técnicas, tecnologías, metodologías, herramientas de software y lenguajes de programación sobre los que se llevará a cabo el proceso de desarrollo.

1.1 Conceptos asociados al problema

1.1.1 Aplicaciones Web

Es una aplicación informática que los usuarios utilizan accediendo a un servidor web a través de internet o de una intranet. Las aplicaciones web se codifican con lenguajes como HTML, Java Script, Java y PHP, que son soportados por los navegadores web, quienes son los encargados de ejecutarlas.

Generalmente una aplicación web posee una estructura de tres capas:

Capa I: El navegador o Interfaz

Capa II: Lógica del negocio. Incluye alguna tecnología web dinámica (PHP, ASP.NET, etc.)

Capa III: Base de datos.

Entre estas capas se establece una fuerte relación de comunicación. De esta forma, desde la primera capa el navegador web o interfaz realiza peticiones a la que contiene la base de datos a través de la capa intermedia que contiene la lógica del negocio.

1.1.2 Portal Web

Un Portal Web es una aplicación web que brinda a los usuarios una interfaz amigable y fácil de usar, además de permitir el acceso a un conjunto de recursos como foros, documentación, buscadores de información, servicio de correo electrónico y otras.

La función de un portal web es ser el punto de partida para un usuario que pretende entrar en la web y realizar búsquedas. Los portales web están enfocados, fundamentalmente, a resolver necesidades de un grupo de personas o de acceso a la información y servicios de una institución pública o privada.

Un portal web debe conseguir que los usuarios accedan a Internet usando siempre una Web determinada; para conseguir esta fidelidad esta aplicación debe implementar dos aspectos muy importantes:

Servicios:

- Espacio para páginas Web.
- Chat.
- Comunidades Virtuales.
- Motor de Búsqueda.
- Foros.
- Servicio de Correo Electrónico.

Contenidos:

- Noticias variadas.
- Novedades.
- Información actualizada y completa sobre el tema.

Se conoce la existencia de tres tipos de portales web:

Portales Horizontales: Se enfocan a satisfacer a una audiencia extensa, pues pretenden atraer distintos tipo de usuarios con temas variados.

Portales Verticales: son los que enfocan su contenido dentro de un tema específico y buscan público interesado en ese contenido.

Portales Diagonales: constituyen la fusión entre un portal horizontal y uno vertical.

1.2 CMS

¿Qué es un CMS?

Un sistema de gestión de contenidos (Content Management Systems o CMS) es un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una Intranet. Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las webs. (1)

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas web puedan concentrarse en el contenido. Hay otras herramientas como la edición de los documentos en XML, utilización de aplicaciones ofimáticas con las que se integra el CMS, importación de documentos existentes y editores que permiten añadir marcas, habitualmente HTML, para indicar el formato y estructura de un documento.

Gestión de contenido

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de los datos de la web, como los relativos a documentos (versiones hechas, autor, fecha de publicación y caducidad), datos y preferencias de los usuarios, aspectos que definen la estructura de la web. Esta se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos.

Publicación

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas.

Presentación

Un CMS puede gestionar automáticamente la accesibilidad del web y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm) y la capacidad de internacionalización.

1.2.1 Necesidad de un CMS

Muchos usuarios particulares utilizan CMS gratuitos para elaborar y gestionar sus webs personales, obteniendo sitios web dinámicos y llenos de funcionalidades. El resultado que obtienen es superior al de algunas empresas que se limitan a tener páginas estáticas que no aportan ningún valor añadido. A continuación se enumeran los puntos más importantes que hacen útil y necesaria la utilización de un CMS: (2)

Inclusión de nuevas funcionalidades en la web.

Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS la introducción de nuevas funcionalidades puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.

Mantenimiento de gran cantidad de páginas.

En una web con muchas páginas se necesita un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.

Reutilización de objetos o componentes.

Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.

Páginas interactivas.

Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición.

Cambios del aspecto de la web.

Si no existe una organización entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (Cascading Style Sheets) con lo que se consigue la independencia de presentación y contenido.

Control de acceso.

Controlar el acceso en un sitio web no consiste simplemente en permitir la entrada al mismo, sino que es necesario gestionar los diferentes permisos aplicados a grupos o individuos en cada área de la web.

1.3 Aplicaciones web de gestión de eventos

1.3.1 Internacional

Las aplicaciones web existentes en el área internacional proporcionan elementos que ayudan a la confección de un portal que gestione la información y permita realizar reportes en base a los recursos asociados a este, pero no ofrecen una solución estándar para la elaboración de una aplicación web que englobe todo lo antes planteado.

1.3.1.1 ConfWiz

Es una solución especialmente diseñada para pequeñas, medianas empresas, departamentos y divisiones de grandes organizaciones que lleven la gestión on-line de sus eventos presenciales o virtuales. (3)

Es fácil y amigable para todos los usuarios que la visiten. Proporciona además una gestión avanzada para usuarios más experimentados, generación automática de la agenda del evento en formatos HTML, texto plano y PDF, y se caracteriza por ser muy rápida y segura.

1.3.1.2 Gestkal XXI

Empresa dedicada a la gestión de eventos deportivos y culturales en la zona del Levante español. Esta web ha sido configurada para que el personal de la empresa pueda editar a través de la intranet los resultados de los juegos deportivos, las noticias de interés, artículos, subpáginas de clientes, y agenda de eventos. (4)

1.3.1.3 Eventtual

Es una empresa dedicada al diseño, producción y gestión de eventos. Partiendo de la ideas de sus clientes, crean un concepto práctico y tangible. De la mesa de diseño a la clausura del evento, coordinan cada aspecto del proyecto y realizan la gestión integral del mismo, ahorrando tiempo y costo. (5)

1.3.2 Nacional

En Cuba existen varios portales web que proporcionan diferentes informaciones con respecto a los eventos que se realizarán, pero de forma estática. No gestionan la información requerida para un control y generación de varios reportes de acuerdo a las especificaciones de cualquier evento.

1.3.2.1 Eventos en Cuba

Es un portal web que gestiona un gran flujo de información de todos los eventos que se efectuarán en el país de diversa índole, el cual brinda a sus usuarios una breve descripción de estos. No le permite al usuario obtener diferentes reportes relacionados con la gestión de un evento específico.

1.3.2.2 Portal NeuroRehabana2008

Es un portal que gestiona información relacionada con el área organizativa y del evento, no se puede realizar la gestión de los recursos humanos del mismo, no se conocen los participantes, su currículum, los trabajos presentados por ellos y las distintas actividades que se realizarán.

1.3.3 UCI

En la Universidad de Ciencias Informáticas (UCI) existe una aplicación web denominada Cátedra de Programación Avanzada que gestiona prácticamente todo el flujo de información de los principales eventos de programación que se efectúan en la universidad. Proporciona además a los usuarios una serie de facilidades que les permite mandar soluciones de diversos ejercicios mediante un jurado online.

En la UCI se efectúan anualmente una serie de competencias deportivas entre sus distintas facultades, donde todos los resultados y diferentes informaciones se muestran en un sitio web, el cual genera diversos reportes de acuerdo a lo buscado por cualquier usuario y brinda una gran ayuda a la organización y a la información del evento deportivo. Esta aplicación se puede tomar como punto de partida para ver la organización y gestión de los distintos recursos para el desarrollo del portal web de gestión de eventos, específicamente el evento NeuroRehabana 2010.

1.4 Tendencias, tecnologías y metodologías actuales a considerar.

En este epígrafe se abordan los conceptos fundamentales relacionados con la arquitectura, tecnologías, herramientas y metodología que van a ser utilizadas para el proceso de desarrollo del portal gestión del evento científico.

1.4.1 Servidor Web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada).

Servidor Web Apache

El servidor web es un elemento indispensable y fundamental en el desarrollo de cualquier portal, pues tiene la responsabilidad de atender las peticiones de los clientes y enviarles las páginas Web solicitadas. Existen variedad de servidores web, sin embargo entre ellos se destaca el Apache, el cual es un software funciona sobre cualquier plataforma. Es un software de código abierto, altamente flexible, rápido, configurable, robusto y estable, que lo convierten en el servidor preferido por excelencia.

Entre las características del servidor web Apache se destacan:

- Presenta un elaborado índice de directorios.
- Posee un directorio de alias.
- Permite la integración de imágenes del lado del servidor
- Incluye negociación de contenidos.
- Emite informe de errores HTTP configurable.
- Posibilita la reescritura de las URL y la comprobación de la ortografía de las estas.
- Es modular, ya que puede ser adaptado a diferentes entornos y necesidades.
- Puede escuchar peticiones para direcciones IP diferentes.

1.4.2 Lenguajes de programación web y protocolos

1.4.2.1 HTML

HTML (HyperText Markup Language por sus siglas en inglés) es actualmente uno de los lenguajes de programación web más importante debido a que es tolerado por todos los navegadores web existentes. Se usa para describir la estructura y el contenido en forma de texto y para complementar este último con objetos.

Ventajas

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.
- Los documentos HTML son de pequeño tamaño por lo que transferirlos a través de Internet se convierte en una operación breve.
- Los documentos HTML se despliegan en cualquier plataforma, pues son independientes de los dispositivos; simplemente necesitan un visualizador para la plataforma en que se ejecuten las operaciones, que tenga la capacidad de entenderlo.

1.4.2.2 PHP

PHP (*Hypertext Pre-processor*) es un lenguaje interpretado debido a que no necesita ser compilado para ejecutarse y se utiliza para la creación de páginas web dinámicas. Es un lenguaje de programación del lado del servidor, por lo que el cliente sólo recibe el resultado de la ejecución. Contienen en gran parte sintaxis de lenguajes conocidos como Java, Perl, C y puede ser incrustado dentro de código HTML. Para su funcionamiento necesita tener un servidor Web con el intérprete de PHP. Los archivos cuentan con la extensión (php).

Ventajas: (6)

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.

- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

1.4.2.3 JavaScript

- JavaScript es un lenguaje interpretado y cuyo uso se ha generalizado en la construcción de páginas web. Es un lenguaje similar a Java, aunque no es un lenguaje de programación orientada a objetos. Es un lenguaje de gran popularidad pues su código es interpretado por la mayoría de los navegadores en sus versiones más recientes. El código JavaScript puede ser integrado dentro de las páginas web.

Ventajas

- Lenguaje interpretado seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

1.4.2.4 CCS

CCS (Cascading Style Sheets por sus siglas en inglés) es un lenguaje que implementa un mecanismo sencillo, capaz de describir la forma en que puede ser mostrado un documento en la pantalla. Puede definir además la forma en que el documento se puede imprimir, así como cual será la apariencia de este mostrada a través de un dispositivo de lectura. Es por estas razones que los desarrolladores que emplean JavaScript tienen la totalidad del control sobre el estilo y formato de sus documentos. (7)

Ventajas:

- Permite mantener el control centralizado de la presentación de un sitio web en su totalidad, por lo que la actualización de este se realiza de manera sencilla y ágil.

- Hace posible que los usuarios a través de los navegadores, especifiquen su propia hoja de estilo a aplicar a un sitio web.
- Posibilita que el documento HTML sea más legible y fácil de entender, reduciendo tu tamaño.
- Permite definir con mayor precisión y muchas más unidades, atributos como: pixels (px) y porcentaje (%), como antes; pulgadas (in); puntos (pt); y centímetros (cm).

1.4.2.5 XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Ventajas

- Brinda, tanto a autores como proveedores, la posibilidad de diseñar el tipo de documento que deseen, por lo que las manipulaciones se hacen más fáciles.
- Este lenguaje aporta, a la representación visual, mayores ventajas y facilidades.
- Permite realizar búsquedas con más significado, ya que con XML los datos se pueden etiquetar de forma exclusiva.
- Posibilita la integración de datos estructurados precedentes de distintas fuentes, debido a su capacidad de ampliación y flexibilidad.
- Los datos en formato XML pueden ser analizados, editados y manipulados de forma local, o sea, estas operaciones pueden ser realizada por lo usuarios.
- Brinda posibilidad de obtener varias vistas de lo datos, pues al describirlos de forma abierta, simple y extensible, este lenguaje se convierte en complemento de HTML.

1.4.2.6 SOAP (Simple Object Access Protocol)

SOAP es un protocolo estándar que permite la comunicación y la interoperabilidad entre diversas aplicaciones Web desarrolladas bajo tecnologías diferentes. En el núcleo de los servicios Web se encuentra el protocolo simple de acceso a datos SOAP, que proporciona un mecanismo estándar de

empaquetar mensajes. SOAP ha recibido gran atención debido a que facilita una comunicación del estilo RPC entre un cliente y un servidor remoto. (8)

Algunas de las Ventajas de SOAP son:

- **No está asociado con ningún lenguaje:** SOAP no especifica una API (Application Programming Interface por sus siglas en inglés), por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.
- **No se encuentra fuertemente asociado a ningún protocolo de transporte:** Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- **No está atado a ninguna infraestructura de objeto distribuido** La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.
- **Aprovecha los estándares existentes en la industria:** Los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrollo sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas.

1.4.3 Framework

1.4.3.1 Zapatec

Brinda una cantidad de herramientas para el diseño de las interfaces en las aplicaciones web, entre las que se encuentran por ejemplo: calendarios, menús, explorador árbol, formularios, grid, slider, tabs y efectos. Su uso es vital para la elaboración de los diferentes listados y los formularios para almacenar, editar y eliminar la información gestionada en el evento científico NeuroRehabana 2010.

1.4.4 Gestor de base de datos (SGBD)

Un Sistema Gestión de Bases de Datos (Data Base Management System DBMS) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a éstos, que tiene como objetivo primordial proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

1.4.4.1 MySQL

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. (9)

Características:

- Ha sido probado bajo un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- Emplea GNU Automake, Autoconf, y Libtool para portabilidad.
- Tiene APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Brinda sistemas de almacenamiento, tanto transaccionales como no transaccionales.
- Contiene un sistema de reserva de memoria que se caracteriza por su rapidez y por estar basado en hilos.
- Su servidor puede ser usado como un programa separado para emplear en un entorno de red cliente/servidor.
- Es seguro, posee un sistema de privilegios y contraseñas muy flexible y seguro.
- Se caracteriza por soportar a grandes bases de datos (que contienen hasta 50 millones de registros y 60.000 tablas).

Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Es soportado por gran variedad de sistemas operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

- Conectividad y seguridad.
- Integración con PHP.

1.4.5 Arquitectura de Software

Una Arquitectura de Software, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que analistas, diseñadores y programadores, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

1.4.5.1 Modelo Vista Controlador (MVC)

La arquitectura de software Modelo Vista Controlador (MVC), fue diseñada con el objetivo de reducir el esfuerzo de programación necesario empleado en la implementación de sistemas múltiples y sincronizados que manejan los mismos datos.

Descripción del patrón MVC

El Modelo: es la entidad que contiene y maneja los datos del programa controlando sus transformaciones.

La Vista: es la entidad encargada de manipular la manera en que se presentan visualmente los datos que contiene el Modelo.

El Controlador: es la entidad que toma las órdenes de los usuarios y les impregna significado, accionando sobre los datos del Modelo.

Modelo de arquitectura MVC presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa.
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica.

En conclusión, al utilizar el patrón de arquitectura Modelo Vista Controlador en el diseño, se logra construir por separado los componentes de un programa, para luego unirlos en tiempo de ejecución.

1.4.5.2 Arquitectura Orientada a Servicios (SOA)

Nace como una estrategia de integración y constituye una tendencia creciente que intenta reconciliar la visión técnica y de negocios, basándose en estándares abiertos y promoviendo la interoperabilidad entre diversas organizaciones y plataformas de manera eficiente y flexible a los cambios.

SOA es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente, pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

Algunos beneficios de usar la arquitectura SOA son:

- La reducción de costos y tiempo en el desarrollo de aplicaciones: SOA permite reutilizar los módulos de aplicaciones existentes para resolver problemas comunes a otras aplicaciones. Como consecuencia también se reducen los costos de mantenimiento y se logra aumentar la robustez del nuevo sistema, al utilizarse software ya probado.
- Facilita el proceso de integración: se interactúa con elementos que se abstraen de la ubicación de los servicios y tecnología con la que fueron creados los mismos.

1.4.6 Desarrollo basado en RUP

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requerimientos de un usuario en un software. Sin embargo, RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema o software en construcción está formado por componentes interconectados a través de interfaces.

RUP utiliza el UML para representar todos los esquemas del sistema de software a desarrollar, de hecho este es una parte esencial de esta metodología, pero sus verdaderos aspectos definitorios se resumen en tres ideas claves: dirigida por casos de uso, centrada en la arquitectura e iterativas e incremental. Un sistema de software se define para dar servicios a sus usuarios, por lo tanto para construirlo con éxito se debe conocer qué es lo que los futuros usuarios necesitan así como que para

llevar a cabo el proceso de desarrollo es conveniente dividir el trabajo en partes pequeñas o miniproyectos, representando cada uno de ellos lo que se conoce como iteración.

UML (Lenguaje de Modelado Unificado)

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

1.4.7 Tecnologías de desarrollo

1.4.7.1 AJAX

AJAX (Asynchronous JavaScript And XML) es una técnica para el desarrollo de aplicaciones Web interactivas o RIA (Rich Internet Applications) que no constituye una tecnología en sí, sino una combinación de cuatro tecnologías ya existentes: (10)

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- DOM (Document Object Model) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un ícono de reloj de arena esperando a que el servidor haga algo.

1.4.8 Servicios Web

Los servicios web son un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

1.4.9 Herramientas

1.4.9.1 Embarcadero ER/Studio 7.0

Incorpora nuevas capacidades de modelado, que permiten administrar grandes modelos en forma colaborativa, al facilitar y reforzar los estándares en estos modelos, además de la creación de diagramas en su documentación. La solución captura los metadatos clave en un formato de datos lineales. (11)

Las nuevas características de ER/Studio incluyen:

Administración del modelado empresarial – Permite la creación y consolidación de modelos de proyecto en un modelo global, para la administración holística de la información.

Capacidades avanzadas en la administración de metadatos, incluyen la consolidación de los mismos en un repositorio y en un formato visual navegable.

Integración de datos y soporte para el almacenamiento de información mejorados – Trae nuevas capacidades para la documentación lineal de la información, de tal forma que los usuarios pueden rastrear los múltiples niveles de la información lineal.

Modelado Físico Mejorado – Para soportar la transición de lógico a físico, ER/Studio 7.0 provee a los administradores de base de datos dos nuevas características: capacidad de planificación y modelado seguro.

1.4.9.2 Zend Studio 6.1.1

Zend Studio es un editor de texto para páginas en código PHP que proporciona ayudas a medida que vas desarrollando tu proyecto, desde la creación y gestión de proyectos hasta la depuración del código. (12)

Zend Studio tiene soporte superior para el desarrollo de modernas aplicaciones con Ajax. Al proporcionar potentes capacidades de PHP, la mejora de soporte de JavaScript, hacen que el desarrollo de aplicaciones se realice en un tiempo récord.

Características:

- Excelente completamiento de código
- Coloreado en la sintaxis del código
- Administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP.

Ventajas:

- Agiliza el trabajo, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de las aplicaciones.

1.4.9.3 Macromedia Dreamweaver:

Dreamweaver es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Javascript para crear efectos e interactividades
- Inserción de archivos multimedia

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas. (13)

1.4.9.4 Enterprise Architect

Enterprise Architect es una herramienta de uso muy sencillo y multiusuario, que aborda el diseño y análisis UML y cubre el desarrollo de software desde la captura de requerimientos a lo largo de las etapas de análisis, diseño, pruebas y mantenimiento. Así como también es diseñada para ayudar a construir software robusto y fácil de mantener. Además, permite generar documentación e informes flexibles y de alta calidad.

Proporciona trazabilidad completa desde el análisis de requerimientos y los artefactos de diseño, hasta la implementación y el despliegue. En combinación con la asignación de recursos y tareas que incorpora, los equipos de Gestión de Proyectos y Calidad están dotados con toda la información necesaria para ayudarles a controlar los proyectos y sus entregas.

1.4.9.5 EMS MySQL Manager

EMS MySQL Manager es una aplicación que proporciona potentes herramientas para llevar la administración de bases de datos MySQL. Gracias a su interfaz gráfica, EMS MySQL Manager permite crear y editar todos los objetos de bases de datos MySQL de una forma simple y directa, además de poder diseñar las bases de datos visualmente.

Tiene también las siguientes características:

- Administrar usuarios y privilegios de usuario.
- Construir consultas SQL visualmente.
- Crear informes de estructuras de bases de datos en formato HTML.
- Exporta/importar datos y muchos más servicios que hacen que el trabajo con EMS MySQL Manager sea muy fácil.

1.4.9.6 CMS a Utilizar

De los varios tipos de CMS existentes en la actualidad, sólo unos pocos realmente están llevando la web a una nueva era de agilidad en la inclusión de contenido, estímulo de la comunicación colectiva y reconocimiento al usuario como generador de información útil, en lugar de simple manejador pasivo de artículos y noticias. Entre los más usados se destacan Drupal, Plone, Joomla y WordPress producto de

Capítulo 1: Fundamentación Teórica

un exhaustivo estudio se llegó a la conclusión de utilizar Joomla. Desde su aparición al público, **Joomla** ha avanzado a pasos agigantados, siendo uno de los líderes en la lista de los mejores CMS, aún por encima de aplicaciones comerciales.

Su triunfo se atribuye a que es uno de los mejores CMS de Código Abierto, es fácil de instalar, simple de manejar y confiable, una de las razones principales para optar por él, ya que en el lugar donde se va a instalar no necesariamente siempre va a estar administrado por el personal mejor capacitado.

Entre los diferentes usos de Joomla están:

- Webs Corporativas o Portales
- Pequeños Sitios de Negocios
- Aplicaciones Gubernamentales
- Webs de Escuelas o Agrupaciones
- Páginas Personales o Familiares
- Portales de Comunidades
- Revistas y Periódicos

Realiza un gran trabajo gestionando el contenido necesario para que un sitio web funcione. Pero para muchas personas, el verdadero potencial de Joomla recae en la arquitectura de la aplicación, que posibilita que miles de desarrolladores en el mundo puedan crear potentes plugins y extensiones.

Conclusiones

En este capítulo, se profundizó en los conceptos y definiciones necesarias para comprender el proceso de desarrollo de un portal web que gestione las distintas especificaciones. Además se realizó, un análisis de las tecnologías, metodologías, lenguajes y herramientas que se tienen en cuenta para llevar a cabo dicho proceso con la eficiencia requerida.

Capítulo 2: Características del sistema

El primer paso en el desarrollo de software es lograr que los desarrolladores identifiquen el problema para ofrecer una solución que esté de acuerdo con las terminologías y especificaciones del cliente. Se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que se propone, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.1 Modelo de Dominio

Debido a la poca complejidad de la estructura y los mecanismos en el proceso de administración y las funcionalidades del portal, teniendo en cuenta que la problemática analizada es poco compleja, no necesita un modelado completo del negocio.

Por las características expuestas se utilizará un modelo de dominio que permitirá contribuir a la comprensión del sistema, y por lo tanto también a la comprensión de los requerimientos del sistema que se desprenden de este contexto.

El Modelo de Dominio es la representación de conceptos de importancia, constituido por reglas de modelado, no por componentes propios de un sistema de software. Es posible realizar un elevado grado de detalles en un Modelo de Dominio, ya que tiene como objetivo fundamental capturar lo necesario para comprender el funcionamiento del sistema. Puede ser tomado como punto de partida para el desarrollo de la aplicación web para la gestión del evento científico NeuroRehabana 2010.

2.1.1 Conceptos fundamentales

A continuación se identifican todos los conceptos necesarios para entender el modelo de dominio:

Portal: es la aplicación Web a realizar, en este caso el Portal de Gestión del Evento Científico NeuroRehabana2010.

Evento: se refiere al evento científico o de otra índole, efectuado en Cuba, en este caso el Evento Científico NeuroRehabana 2010.

Usuario: personas que interactúan directamente con el portal.

Participantes: son las personas que son invitadas al evento y pueden o no presentar trabajos.

Modalidad: se refiere a las distintas particularidades del evento, en cuanto a los distintos tipos de exposición.

Capítulo 2: Características del sistema

Trabajo: presentación de una investigación o accesorios relacionados con el evento.

Comentario: se refiere a la opinión que pueden tener los usuarios de un trabajo en específico.

Información: a toda aquella información publicada por el administrador del portal o un usuario.

Patrocinador: persona o institución que de una forma u otra aportan accesorios o ayuda monetaria para la realización del evento.

Noticia: son las novedades de última hora del evento y distintas informaciones del mismo.

Búsqueda: resultado de la información buscada sobre un usuario o los diferentes trabajos que se presentan.

Administrador: es la persona que tendrá acceso a configurar el diseño del portal así como publicar o quitar informaciones relacionadas al evento.

Editor: es la persona que podrá publicar, eliminar o editar informaciones o noticias referentes al portal web.

Web Services: se encarga de brindar los servicios de conexión y acceso mediante una interfaz WDSL.

2.1.2 Modelo de dominio

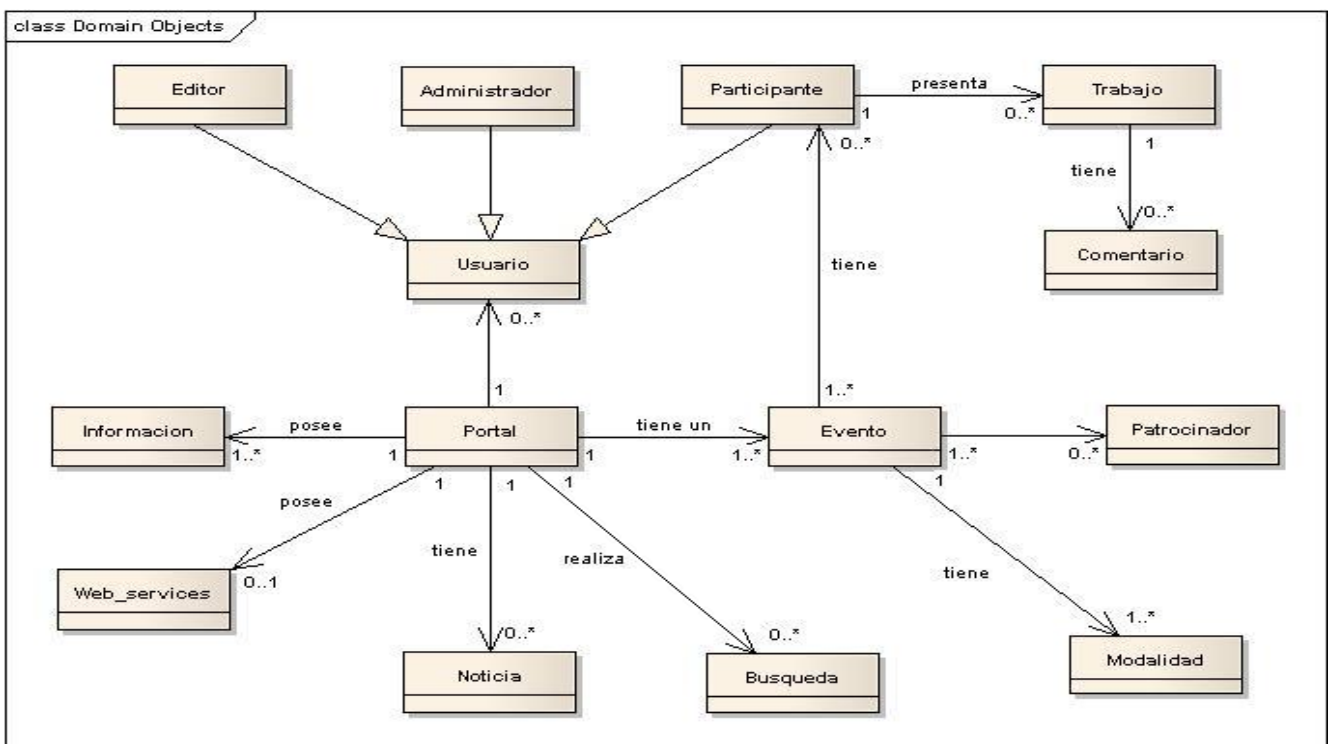


Fig. 1 Modelo del dominio.

2.2 Captura de requisitos

Luego de tener un conocimiento general del sistema y todo el contexto en que se desarrolla se procede al próximo paso, la captura de los requisitos del sistema, que son definidos en RUP como una condición que el sistema debe cumplir o capacidad que este debe tener. Estos se clasifican en dos importantes grupos, **los requerimientos funcionales** y **los no funcionales** estos son de gran importancia para el posterior desarrollo del software.

2.2.1 Requerimientos funcionales

Los requerimientos funcionales representan las acciones que el sistema debe ejecutar. En síntesis constituyen condiciones o capacidades que el sistema debe cumplir, definiendo como se comportará internamente. Los requerimientos funcionales son estables y se mantienen constantes ante cualquier propiedad o funcionalidad con la que se relacionen.

Tras describir brevemente en este epígrafe la esencia del modelo de domino a realizar se llegan a los siguientes requerimientos funcionales:

RF1 Autenticar usuario.

RF2 Gestionar evento.

2.1 Adicionar evento.

2.2 Modificar evento.

2.3 Eliminar evento.

RF3 Mostrar lista de eventos.

RF4 Gestionar sede.

4.1 Adicionar sede.

4.2 Modificar sede.

4.3 Eliminar sede.

RF5 Mostrar lista de sedes.

RF6 Gestionar local.

6.1 Adicionar local.

6.2 Modificar local.

6.3 Eliminar local.

RF7 Mostrar lista de locales.

RF8 Gestionar participante.

8.1 Adicionar participante.

8.2 Eliminar participante.

RF9 Modificar participante.

RF10 Mostrar lista de participantes.

RF11 Gestionar patrocinador.

11.1 Adicionar patrocinador.

11.2 Modificar patrocinador.

11.3 Eliminar patrocinador.

RF12 Gestionar modalidad.

12.1 Adicionar modalidad.

12.2 Modificar modalidad.

12.3 Eliminar modalidad.

RF13 Gestionar trabajo.

13.1 Adicionar trabajo.

13.2 Modificar trabajo.

13.3 Eliminar trabajo.

RF14 Gestionar trabajo complementario

13.1 Adicionar trabajo complementario.

13.2 Modificar trabajo complementario.

13.3 Eliminar trabajo complementario.

RF15 Mostrar lista de trabajos.

RF16 Gestionar comentario.

16.1 Adicionar comentario.

16.2 Modificar comentario.

16.3 Eliminar comentario.

16.4 Listar comentarios.

RF17 Gestionar noticias.

17.1 Adicionar noticias.

17.2 Modificar noticias.

17.3 Eliminar noticias.

RF18 Mostrar noticias.

RF19 Gestionar actividad.

19.1 Adicionar actividad.

19.2 Modificar actividad.

19.3 Eliminar actividad.

RF20 Mostrar cronograma.

RF21 Gestionar información.

21.1 Adicionar información.

21.2 Modificar información.

21.3 Eliminar información.

RF22 Mostrar trabajo complementario

RF23 Mostrar información.

RF24 Buscar información.

RF25 Buscar persona.

RF26 Buscar trabajo.

2.2.2 Requerimientos no funcionales del sistema

Los requerimientos no funcionales representan propiedades o cualidades que el producto debe tener. Son un conjunto de características que definen un producto atractivo, usable, rápido y confiable. Los requerimientos no funcionales del sistema son fundamentales en ocasiones para garantizar el éxito de un producto. A través de estos tipos de requerimientos, los clientes y usuarios finales pueden aceptar o no las características no funcionales del producto, que al cumplirse plenamente, garantizan la aceptación del producto.

Requerimientos no funcionales definidos:

Usabilidad.

RNF 1 El sistema podrá ser accedido y manipulado por personas que posean una preparación básica en el manejo de la computadora, así como interacción mínima en un ambiente Web.

Rendimiento.

RNF 2 La velocidad de procesamiento de la información será rápida.

Soporte.

RNF 3 El servidor de base de datos debe soportar una velocidad de procesamiento mínima, al igual que volúmenes menores de datos. También debe ofrecer rapidez en las respuestas de accesos concurrentes.

RNF 4 En el lado del cliente es necesario un navegador que interprete el lenguaje de programación JavaScript.

Seguridad.

RNF 5 La información solo podrá ser publicada por el administrador del sistema.

RNF 6 La información que maneja el sistema deberá estar protegida de accesos no autorizados

RNF 7 El sistema brindará acceso a las funcionalidades de acuerdo al nivel de usuario activo.

RNF 8 Se garantizará la protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

RNF 9 El sistema garantizará la verificación de acciones irreversibles o acciones de eliminación.

Apariencia o interfaz externa.

RNF 10 Se realizará un diseño que le garantice al usuario una interfaz amigable, fácil y simple de usar para cualquier invitado al evento.

Capítulo 2: Características del sistema

Portabilidad.

RNF 11 El sistema podrá ser ejecutado sobre cualquier plataforma (multiplataforma).

Confiabilidad.

RNF 12 La herramienta usada para la implementación del sistema debe brindar soporte para recuperación ante fallos y errores.

Funcionalidad.

RNF 13 El tiempo mínimo de respuesta o de carga del portal debe ser reducido al mínimo.

RNF 14 El sistema permitirá almacenar en caché páginas de contenido de forma tal que se agilice la navegación por el portal.

Software.

RNF 15 En las PC clientes se requiere de un navegador compatible Mozilla FireFox o Internet Explorer.

RNF 16 En las PC servidoras se requiere el servidor Apache Web Server 2.2.8, el PHP en su versión 5.2 o superior y el servidor de base de datos MySQL Database 5.0.

Hardware.

RNF 17 Las PC clientes deben poseer periféricos, mouse y teclado.

RNF 18 Las PC clientes tendrán tarjeta de red, modem o red arrendada.

RNF 19 Las PC servidoras tendrán 256 MB RAM y 1 GB de disco duro.

2.3 Modelo de sistema

El modelo de casos de uso del sistema permite un mayor entendimiento del problema a resolver.

2.3.1 Definición de los actores del sistema.

A continuación se identifican los diferentes actores que intervienen en el sistema y su descripción:

Actor	Descripción
Administrador	Es el encargado de distribuir los permisos a los usuarios para acceder a las informaciones a las cuales pueden tener acceso y se encarga de gestionar la información referente a los recursos del evento.

Capítulo 2: Características del sistema

2.4 Descripción textual de los casos de uso críticos del sistema.

CU 1	Autenticar Usuario	
Actores:	Administrador (inicia), Editor (inicia), Autor (inicia), Invitado (inicia)	
Propósito:	Confirmar la validez de los usuarios, redireccionándolos a las páginas correspondientes según su rol.	
Resumen:	El caso de uso confirma la validez de los usuarios, mostrándoles las funcionalidades a que tienen acceso según su rol.	
Referencias	RF 1	
Precondiciones		
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1.1 El sistema muestra la página de autenticación.	
2. Introduce usuario y contraseña y pulsa el botón "Aceptar".	2.1. El sistema verifica los permisos y muestra la página principal. <ul style="list-style-type: none"> • Si no tiene permisos ver flujo alterno 1. 	
Flujo alterno 1		
Acción del actor	Respuesta del sistema	
	2.1 El sistema informa que no tiene permisos suficientes.	
Puntos de extensión		
Poscondiciones:	El usuario se autenticó correctamente.	
Prioridad:	Crítica	
Interfaz:		

CU 2	Gestionar evento	
Actores:	Administrador (Inicia)	
Propósito:	Gestionar los eventos científicos y de otra índole que serán manejados por el portal.	
Resumen:	El caso de uso permite al administrador adicionar, modificar y eliminar eventos.	
Referencias	RF 2	
Precondiciones	El administrador debe estar registrado.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción gestionar evento.	1.2 El sistema le muestra la interfaz gestionar evento con un listado de eventos y las opciones para Adicionar, Modificar y Eliminar evento.	

Capítulo 2: Características del sistema

	<ul style="list-style-type: none"> • Si desea adicionar evento ver sección "Adicionar evento". • Si desea modificar evento ver sección "Modificar evento". • Si desea eliminar evento ver sección "Eliminar evento".
Sección "Adicionar evento"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción adicionar evento.	1.1 El sistema muestra un formulario con las opciones para adicionar un evento.
2. El Administrador introduce los datos solicitados y presiona el botón "Aceptar".	2.1 El sistema adiciona el nuevo evento.
	2.2 Muestra el evento adicionado en el listado de eventos.
Sección "Modificar evento"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción modificar evento.	1.1 El sistema muestra un formulario que brinda la posibilidad de modificar los datos de un evento.
2. El Administrador modifica los datos deseados y presiona el botón "Guardar".	2.1 El sistema guarda los datos modificados.
	2.2 Muestra los cambios efectuados en el listado de eventos.
Sección "Eliminar evento"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona el evento a eliminar y presiona el botón eliminar.	1.1 El sistema muestra un mensaje preguntado si está seguro que desea eliminar el evento.
2. El Administrador acepta.	2.1 El sistema elimina el evento seleccionado del listado de eventos.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El evento fue gestionado (adicionado, modificado y eliminado).
Prioridad:	
Interfaz:	

Capítulo 2: Características del sistema

CU 13	Gestionar trabajo	
Actores:	Autor (Inicia)	
Propósito:	Gestionar los trabajos presentados en los eventos.	
Resumen:	El caso de uso permite al autor adicionar, modificar y eliminar los trabajos presentados en un evento.	
Referencias	RF 14	
Precondiciones	El autor debe estar debidamente autenticado y registrado entre los participantes.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El Autor selecciona la opción “mostrar mis trabajos”.	1.2 El sistema muestra un listado de los trabajos correspondientes al Autor autenticado y las opciones para Adicionar, Modificar y Eliminar. <ul style="list-style-type: none"> • Si desea adicionar ver sección “Adicionar trabajo”. • Si desea modificar ver sección “Modificar trabajo”. • Si desea eliminar trabajo ver sección “Eliminar Trabajo”. 	
Sección “Adicionar trabajo”		
Acción del actor	Respuesta del sistema	
1. El Autor selecciona la opción para adicionar un trabajo.	1.1 El sistema muestra un formulario para adicionar trabajo.	
1.2 El Autor introduce los datos solicitados y presiona el botón “Aceptar”	2.1 El sistema adiciona el nuevo trabajo.	
	2.2 El sistema muestra el trabajo adicionado en el listado de trabajos.	
Sección “Modificar trabajo”		
Acción del actor	Respuesta del sistema	
1. El Autor selecciona el trabajo que desea modificar y selecciona la opción modificar.	1.1 El sistema muestra un formulario que brinda la posibilidad de modificar los datos de un trabajo.	
2. El Autor modifica los datos deseados y presiona el botón “Aceptar”.	2.1 El sistema guarda los datos modificados.	
	2.2 El sistema muestra los cambios efectuados	

Capítulo 2: Características del sistema

	en el listado de trabajos.
Sección "Eliminar trabajo"	
Acción del actor	Respuesta del sistema
1. El Autor selecciona el trabajo que desea eliminar y selecciona la opción eliminar.	1.1 El sistema muestra un mensaje de confirmación preguntado si está seguro que desea eliminar el trabajo.
2. El Autor acepta.	2.1 El sistema elimina el trabajo del listado de trabajos.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El trabajador fue gestionado (adicionado, modificado y eliminado).
Prioridad:	
Interfaz:	

CU 8	Gestionar participante
Actores:	Administrador (Inicia)
Propósito:	Gestionar los participantes que se encuentran en los eventos.
Resumen:	El caso de uso permite al administrador adicionar y eliminar los trabajos presentados en un evento.
Referencias	RF 8
Precondiciones	El administrador debe estar debidamente autenticado y registrado.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción "mostrar participantes".	1.2 El sistema muestra un listado de los participantes correspondientes a los eventos y las opciones para Adicionar y Eliminar. <ul style="list-style-type: none"> • Si desea adicionar ver sección "Adicionar participante". • Si desea eliminar trabajo ver sección "Eliminar participante".
Sección "Adicionar participante"	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la opción para adicionar un participante.	1.1 El sistema muestra un formulario para adicionar participante.

Capítulo 2: Características del sistema

1.2 El administrador introduce los datos solicitados y presiona el botón "Aceptar"	2.1 El sistema adiciona el nuevo participante.
	2.2 El sistema muestra el participante adicionado en el listado de participante.
Sección "Eliminar participante"	
Acción del actor	Respuesta del sistema
1. El administrador selecciona el participante que desea eliminar y selecciona la opción eliminar.	1.1 El sistema muestra un mensaje de confirmación preguntado si está seguro que desea eliminar el participante.
2. El administrador acepta.	2.1 El sistema elimina el participante del listado de participantes.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El trabajador fue gestionado (adicionado, modificado y eliminado).
Prioridad:	
Interfaz:	
CU 26	Buscar trabajo
Actores:	Administrador (inicia), Editor (inicia), Autor (inicia), Invitado (inicia).
Propósito:	Buscar cualquier tipo de información deseada por el usuario.
Resumen:	El Caso de Uso permite buscar cualquier tipo de información gestionada por el portal.
Referencias	RF 26
Precondiciones	El usuario debe estar debidamente autenticado.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El usuario introduce el criterio de búsqueda en el campo de búsqueda y presiona el botón Buscar.	1.2 El sistema realiza la búsqueda teniendo en cuenta el criterio introducido por el usuario.
	1.3 El sistema muestra una página con los resultados de la búsqueda efectuada. <ul style="list-style-type: none"> Si no encuentra la información de acuerdo al criterio de búsqueda introducido ver flujo alterno 1.

Capítulo 2: Características del sistema

Flujo alterno	
Acción del actor	Respuesta del sistema
	1.3 El sistema muestra un mensaje indicando que no existen resultados para la búsqueda deseada.
Puntos de extensión	
Poscondiciones:	El sistema muestra la información resultante de la búsqueda.
Prioridad:	
Interfaz:	

CU 15	Mostrar listado de trabajos
Actores:	Administrador (inicia), Editor (inicia), Autor (inicia), Invitado (inicia).
Propósito:	Listar los diferentes trabajos realizados por los participantes de un evento determinado.
Resumen:	El Caso de Uso permite listar los distintos trabajos de un evento.
Referencias	RF 15
Precondiciones	El usuario debe estar debidamente autenticado.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El usuario selecciona la opción "Trabajos".	1.1 Se muestra una página que contiene el listado de todos los trabajos en un evento determinado.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El sistema muestra el listado de trabajos resultantes de un evento.
Prioridad:	
Interfaz:	

CU 10	Mostrar listado de participantes
Actores:	Administrador (inicia), Editor (inicia), Autor (inicia), Invitado (inicia).
Propósito:	Listar los diferentes participantes de un evento determinado.
Resumen:	El Caso de Uso permite listar los distintos participantes de un evento.
Referencias	RF 10
Precondiciones	El usuario debe estar debidamente autenticado.
Flujo normal de eventos	

Capítulo 2: Características del sistema

Acción del actor		Respuesta del sistema
1. El usuario selecciona la opción "Participantes".		1.1 Se muestra una página que contiene el listado de todos los participantes en un evento determinado.
Flujo alterno		
Acción del actor		Respuesta del sistema
Puntos de extensión		
Poscondiciones:	El sistema muestra el listado de participantes resultantes de un evento.	
Prioridad:		
Interfaz:		

Conclusiones

En este capítulo, se identificaron los requisitos funcionales y no funcionales del sistema, obtenidos luego de un análisis del proceso del negocio. Se elaboró el modelo de dominio y se describieron detalladamente los casos de uso críticos de la aplicación, por lo que se puede dar paso a la construcción del sistema.

Capítulo 3: Diseño y construcción del sistema

En este capítulo se realiza el diseño para posteriormente pasar a la construcción del sistema. Se modelan los artefactos necesarios para la construcción de una aplicación web en sentido general. En el capítulo se presentan los diagramas de clases de diseño web, el modelo de datos que posibilita confeccionar la base de datos que sirva como soporte al sistema y se describen las tablas de la misma. Se trata además algunos principios del diseño del sistema, estándares de codificación y manejo de errores.

3.1 Modelo de diseño

El Modelo de Diseño es otro de los artefactos que se tienen en cuenta en el ciclo de vida del desarrollo de un software, consumiendo una elevada cantidad de esfuerzo para su realización. Este modelo es de vital importancia para los especialistas y desarrolladores. Ya que pone de relieve una solución lógica: cómo el sistema cumple con los requerimientos; además define e identifica las consecuencias del ambiente de implementación.

Es un modelo físico, no genérico y específico para una implementación, más formal, dinámico. El mismo contiene un sinnúmero de estereotipos físicos sobre las clases en dependencia del lenguaje utilizado. Ha sido creado principalmente como programación visual en procesos iterativos e incrementales.

3.1.1 Patrones de diseño utilizados

Los patrones de diseño describen un problema que ocurre repetidas veces en algún contexto determinado de desarrollo de software, y entregan una buena solución ya probada. Esto ayuda a diseñar correctamente en menos tiempo y a construir problemas reutilizables y extensibles, y facilita la documentación.

3.1.1.1 Bajo acoplamiento

El patrón Bajo Acoplamiento pertenece al conjunto de patrones GRASP, acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Este patrón hace referencia a las relaciones que tienen las clases entre sí dentro de un sistema, mide qué tan fuerte está una clase conectada con otras, es decir, cuántas clases

conoce y necesita. Una clase con bajo o débil acoplamiento no depende de "muchas otras" clases. Una clase con alto o fuerte acoplamiento recurre a muchas otras clases.

3.1.1.2 Alta cohesión

El patrón de Alta Cohesión es otro de los que conforman el grupo de patrones GRASP. Este patrón indica que cada clase del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, además plantea que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.

3.1.2 Estructura del diseño

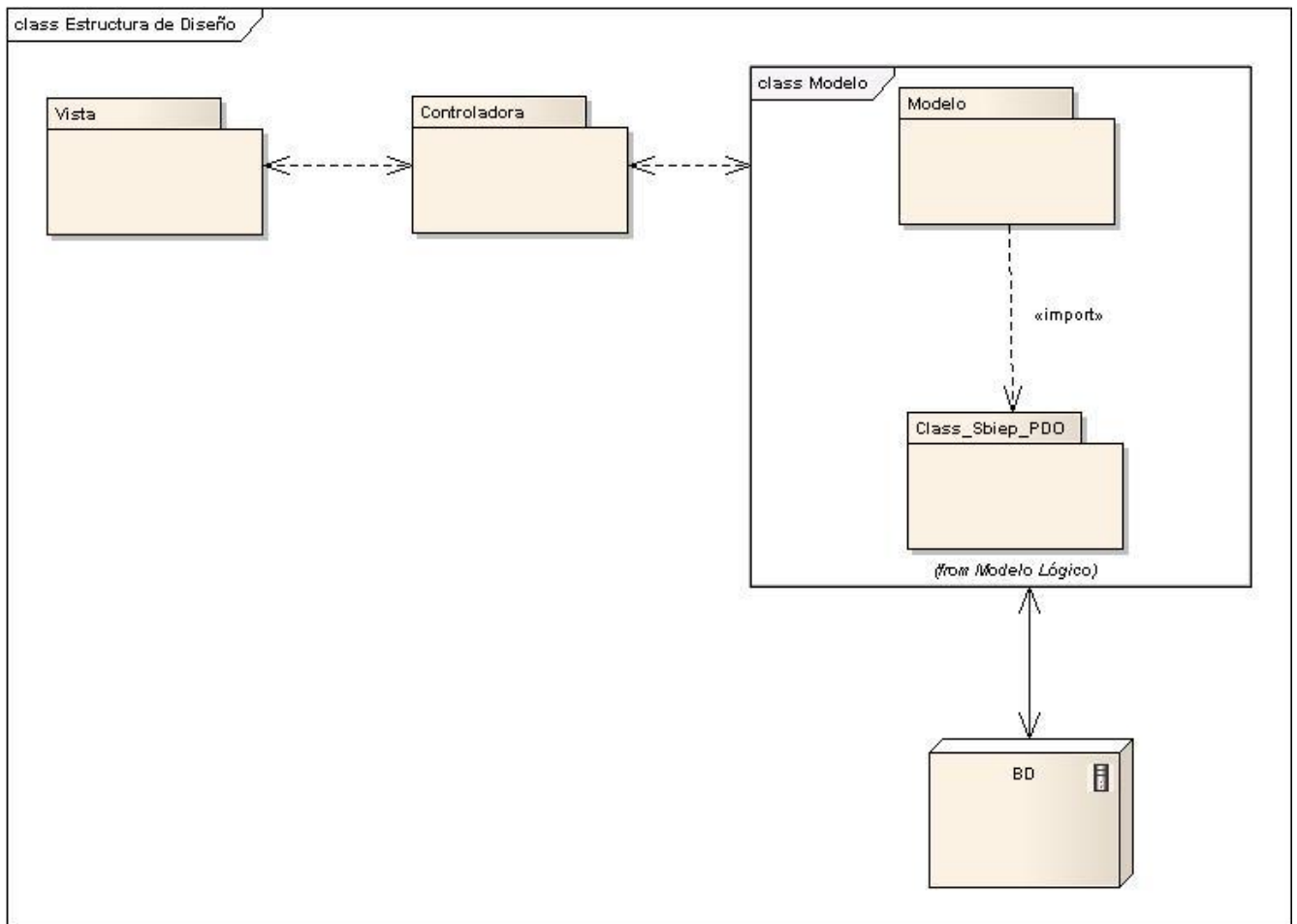


Fig. 3 Estructura del diseño.

Capítulo 3: Diseño y construcción del sistema

Vistas:

En las vistas se encuentran todos los formularios, páginas clientes y servidoras mediante las cuales el usuario selecciona las distintas acciones que desea realizar con la aplicación.

Herramientas utilizadas en las Vistas:

Diseño	DreamWeaver
--------	-------------

Controladora:

Establece la comunicación entre las Vistas y el Modelo, es la encargada de recibir y responder a las distintas peticiones de los usuarios. Las diferentes clases que la conforman reciben las solicitudes de estos, se comunica con el Modelo y envía una respuesta a los mismos de acuerdo a su petición.

Implementación	Zend Studio
----------------	-------------

Modelo:

Aquí se encuentran los distintos ficheros que interactúan con la clase SbieP_PDO, la cual logra la comunicación con la base de datos.

Implementación	Zend Studio
----------------	-------------

Gestor de Base de datos:

El servidor utiliza MySQL 5 para el almacenamiento de datos, recibe todas las solicitudes de intercambio de información con el Modelo.

Gestor de Base de Datos	MySQL 5
-------------------------	---------

3.1.3 Diagrama de clases del diseño con estereotipos web

Un diagrama de clases de diseño web describe gráficamente las especificaciones de las clases del software, en donde se maneja la información relacionada con:

- Clases, asociaciones y atributos.
- Métodos.
- Información sobre los tipos de atributos.
- Navegabilidad.
- Dependencias.

Capítulo 3: Diseño y construcción del sistema

A continuación se muestra una tabla con la terminología que se usará en los diagramas para un mayor entendimiento de los mismos.

Clases	Descripción
fp	Representa un formulario web.
cp	Representa una página cliente.
sp	Representa una página servidora.

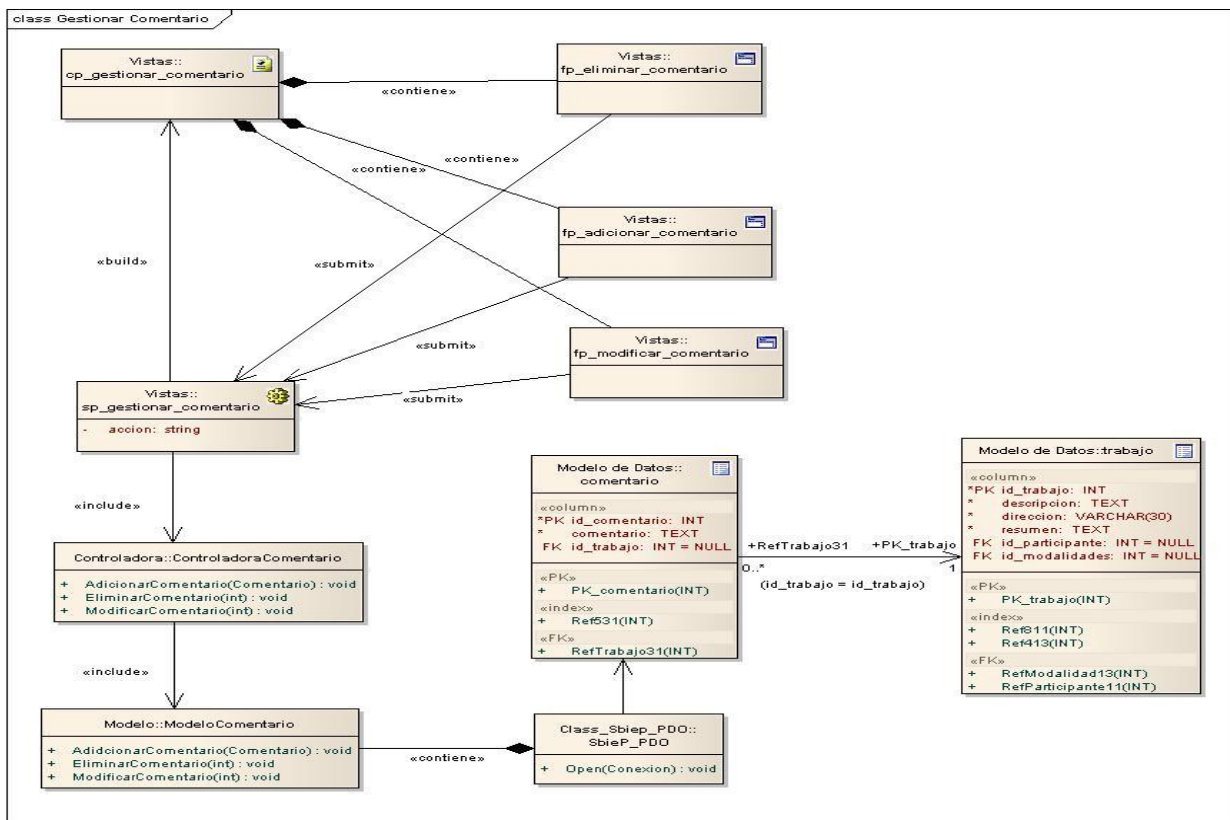


Fig. 4 Diagrama de clases para el caso de uso Gestionar Comentario.

Capítulo 3: Diseño y construcción del sistema

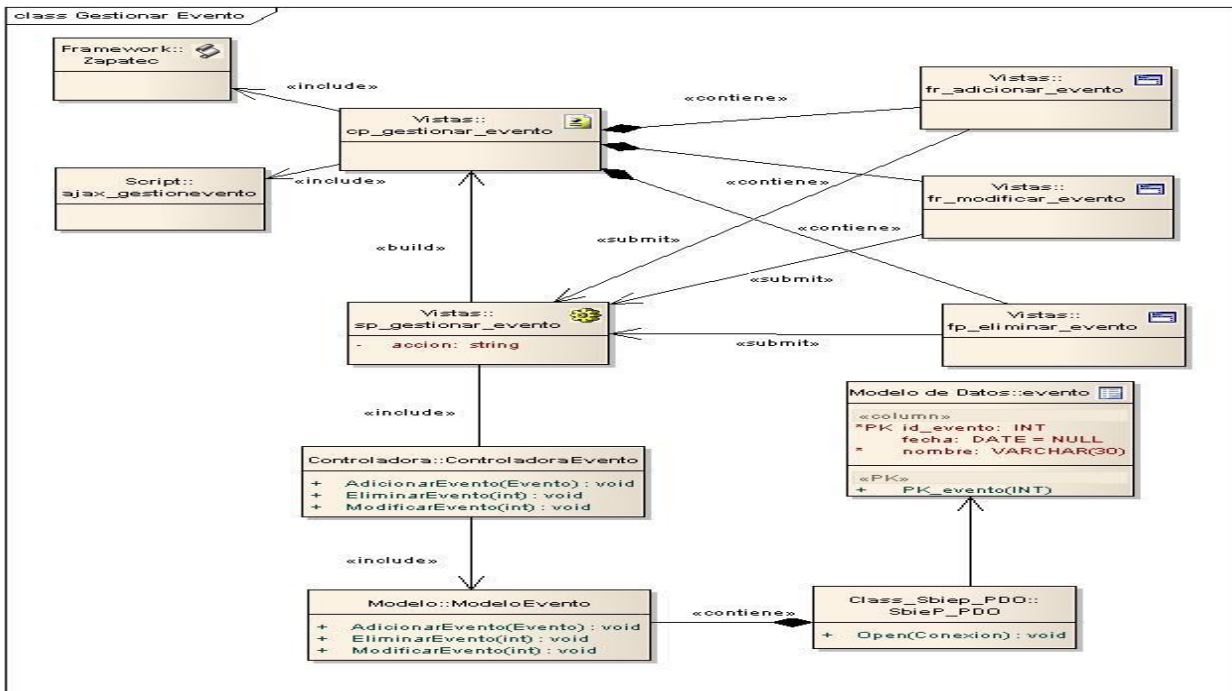


Fig. 5 Diagrama de clases para el caso de uso Gestionar Evento

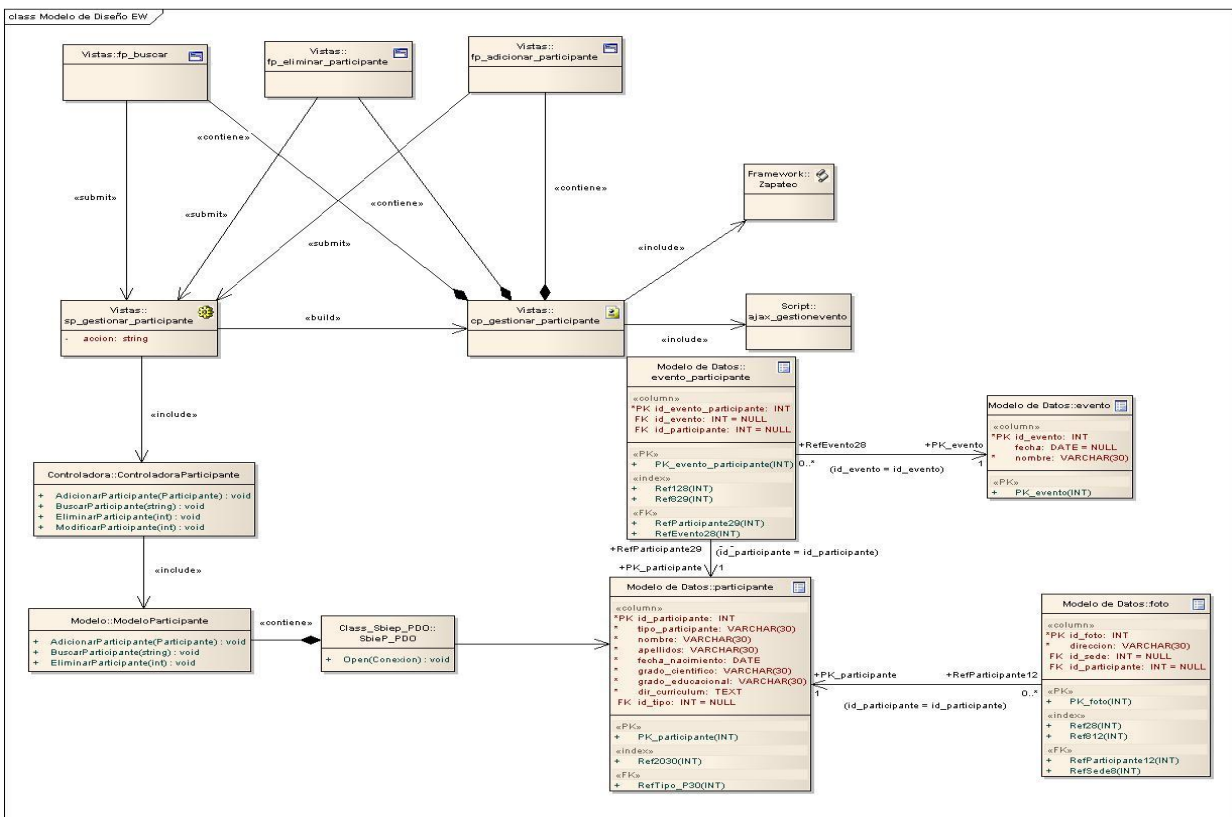


Fig. 6 Diagrama de clases para el caso de uso Gestionar Participante.

Capítulo 3: Diseño y construcción del sistema

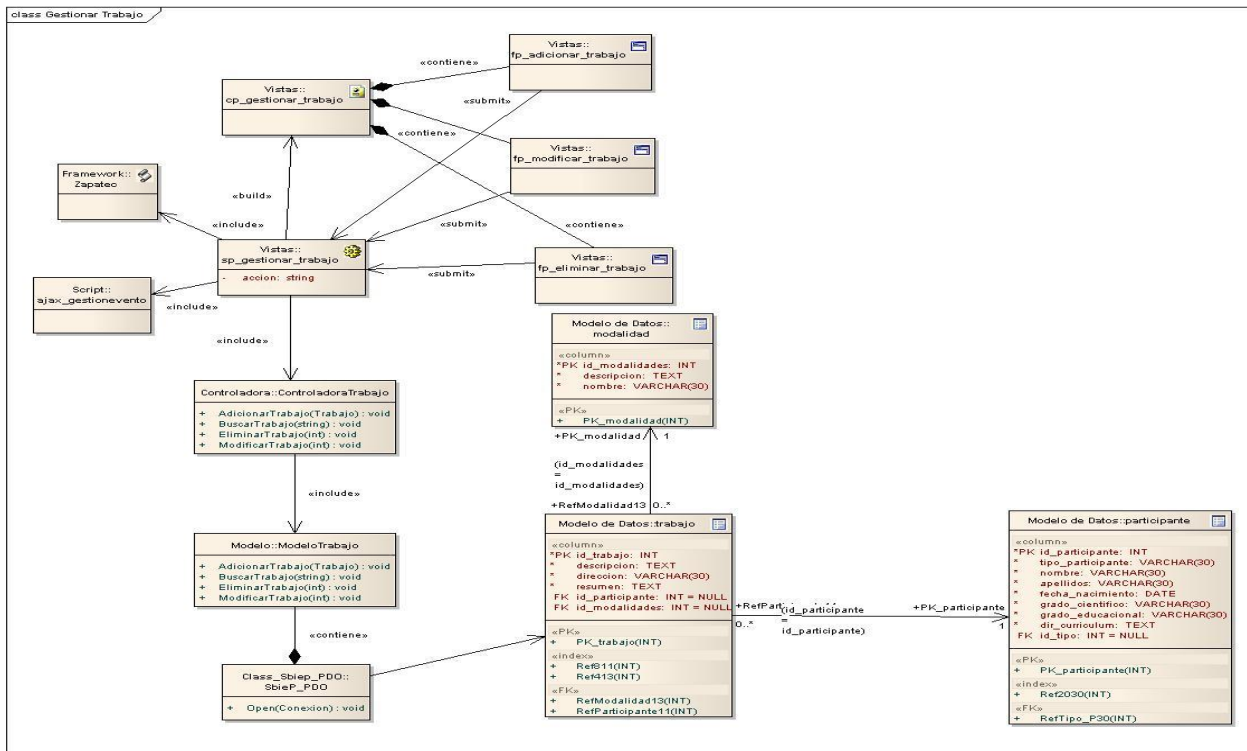


Fig. 7 Diagrama de clases para el caso de uso Gestionar Trabajo.

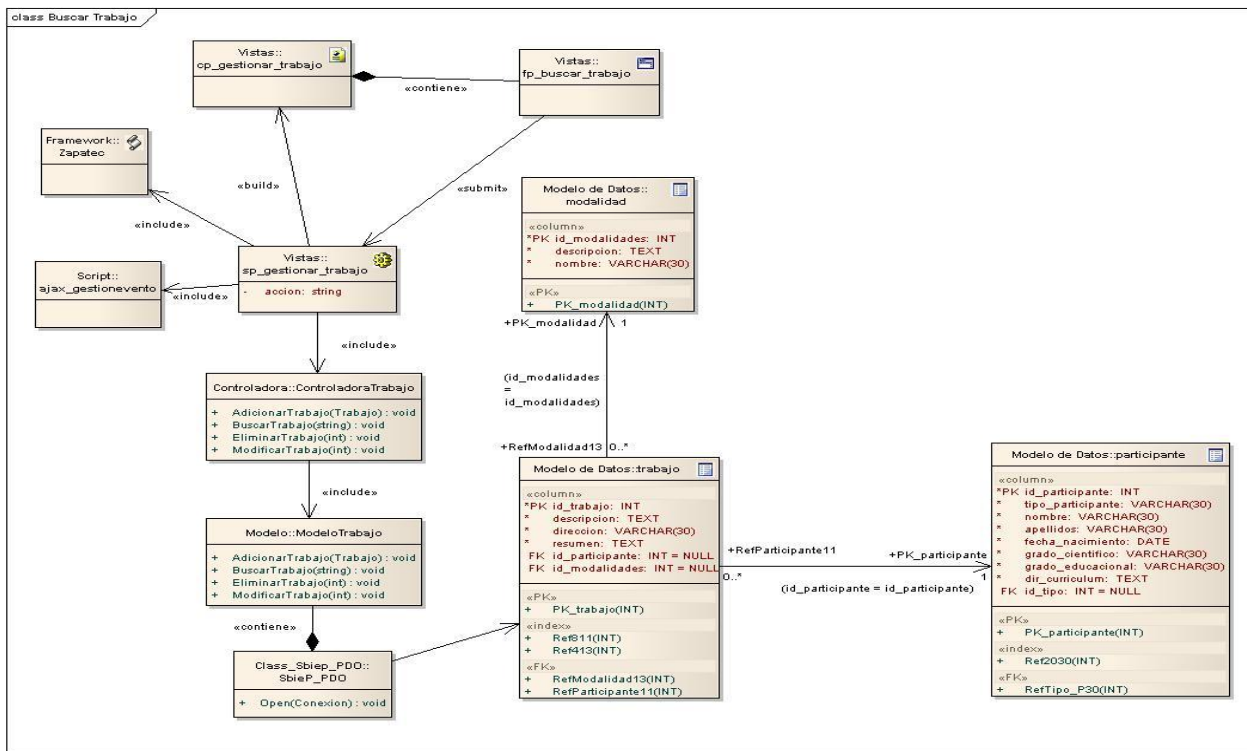


Fig. 8 Diagrama de clases para el caso de uso Buscar Trabajo.

Capítulo 3: Diseño y construcción del sistema

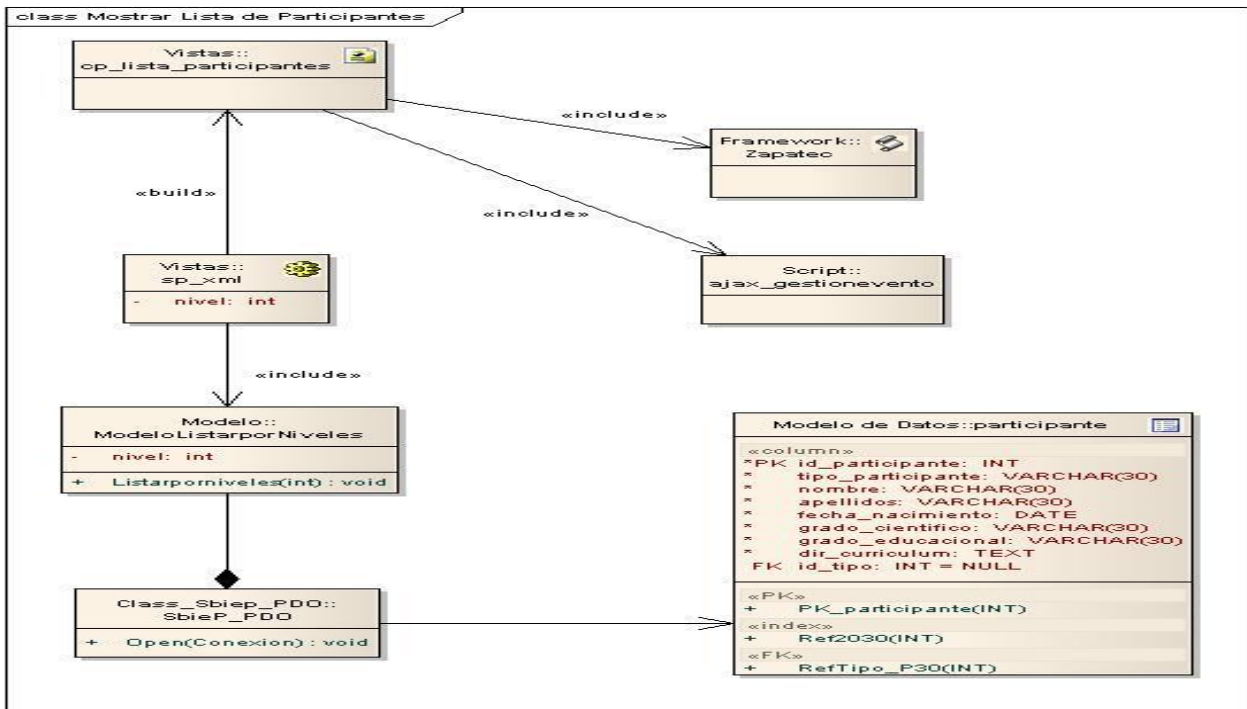


Fig. 9 Diagrama de clases para el caso de uso Mostrar lista de participantes.

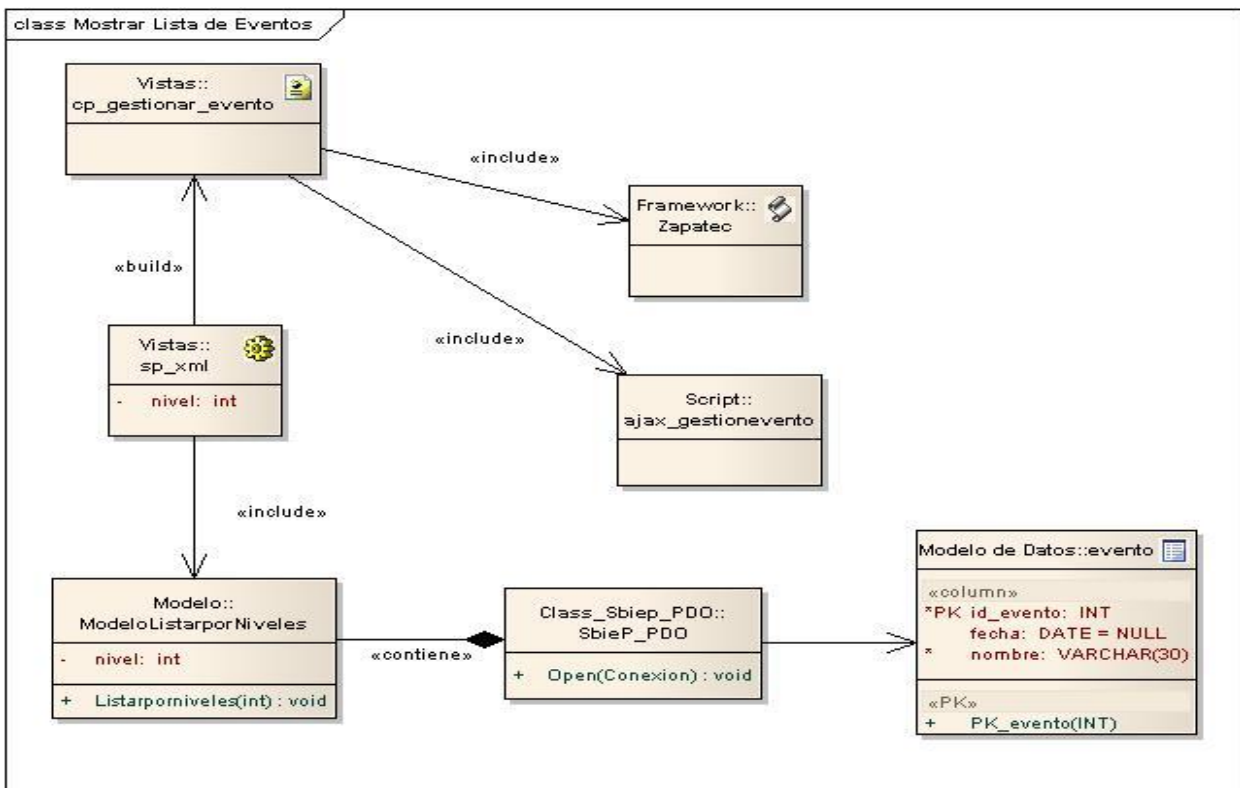


Fig. 10 Diagrama de clases para el caso de uso Mostrar lista de eventos.

3.1.4 Descripción de las clases.

A continuación se describen las clases del diseño representadas en los diagramas mostrados anteriormente.

Nombre: cp_gestionar_comentario
Vista
Tipo: Página cliente
Descripción: La clase cp_gestionar_comentario es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar, modificar y eliminar los datos que serán insertados en la base de datos, tiene un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. La misma se utiliza en el caso de uso: Gestionar comentario.

Nombre: sp_gestionar_comentario
Vista
Tipo: Página servidora
Descripción: Recibe las diferentes acciones de los formularios de la página cliente cp_gestionar_comentario y las envía hacia su correspondiente clase controladora y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a esta misma página, respondiendo el pedido del usuario.

Nombre: cp_gestionar_evento
Vista
Tipo: Página cliente
Descripción: La clase cp_gestionar_evento es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar, modificar y eliminar los datos que serán insertados en la base de datos Evento, tiene un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. La misma se utiliza en el caso de uso: Gestionar evento.

Capítulo 3: Diseño y construcción del sistema

Nombre: sp_gestionar_evento
Vista
Tipo: Página servidora
Descripción: Recibe las diferentes acciones de los formularios de la página cliente cp_gestionar_evento y las envía hacia su correspondiente clase controladora y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a esta misma página, respondiendo el pedido del usuario.

Nombre: cp_gestionar_participante
Vista
Tipo: Página cliente
Descripción: La clase cp_gestionar_participante es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar y eliminar los datos que serán insertados en la base de datos, tiene un conjunto de validaciones en JavaScript que permite lograr una mayor efectividad en el intercambio de información. La misma se utiliza en el caso de uso: Gestionar participante.

Nombre: sp_gestionar_participante
Vista
Tipo: Página servidora
Descripción: Recibe las diferentes acciones de los formularios de la página cliente cp_gestionar_participante y las envía hacia su correspondiente clase controladora y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a esta misma página, respondiendo el pedido del usuario.

Nombre: cp_gestionar_trabajo
Vista
Tipo: Página cliente
Descripción: La clase cp_gestionar_trabajo es una página Web que se ejecuta del lado del cliente sobre un navegador Web. Permite capturar, modificar y eliminar los datos que

Capítulo 3: Diseño y construcción del sistema

serán insertados en la base de datos, tiene un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. La misma se utiliza en el caso de uso: Gestionar trabajo.

Nombre: sp_gestionar_trabajo

Vista

Tipo: Página servidora

Descripción: Recibe las diferentes acciones de los formularios de la página cliente cp_gestionar_participante y las envía hacia su correspondiente clase controladora y una vez concluida la ejecución de sus responsabilidades devuelve una respuesta a esta misma página, respondiendo el pedido del usuario.

Nombre: cp_listar_trabajo

Vista

Tipo: Página cliente

Descripción: La clase cp_listar_trabajo es una página Web en la cual se encuentra una lista de todos los trabajos presentados por los distintos participantes a un evento determinado y permite generación de documentos en Portable Document Format (.pdf) del listado antes mencionado.

Nombre: cp_listar_participante

Vista

Tipo: Página cliente

Descripción: La clase cp_listar_participante es una página Web en la cual se encuentra una lista de todos los participantes a un evento determinado y permite generación de documentos en Portable Document Format (.pdf) del listado antes mencionado.

Capítulo 3: Diseño y construcción del sistema

Nombre: sp_xml

Vista

Tipo: Página servidora

Descripción: Es la encargada de obtener los distintos datos de las tablas de la base de datos, construirlos en un XML y enviarlos a las distintas páginas cliente que poseen algún listado.

Nombre: ControladoraComentario

Controladora

Descripción: Establece la comunicación entre la página de la Vista (GestionarComentario) y su clase Modelo (ModeloComentario). Recibiendo y brindando a los usuarios una respuesta de acuerdo a su petición.

Nombre: ControladoraEvento

Controladora

Descripción: Establece la comunicación entre la página de la Vista (GestionarEvento) y su clase Modelo (ModeloEvento). Recibiendo y brindando a los usuarios una respuesta de acuerdo a su petición.

Nombre: ControladoraParticipante

Controladora

Descripción: Establece la comunicación entre la página de la Vista (GestionarParticipante) y su clase Modelo (ModeloParticipante). Recibiendo y brindando a los usuarios una respuesta de acuerdo a su petición.

Capítulo 3: Diseño y construcción del sistema

Nombre: ControladoraTrabajo

Controladora

Descripción: Establece la comunicación entre la página de la Vista (GestionarTrabajo) y su clase Modelo (ModeloTrabajo). Recibiendo y brindando a los usuarios una respuesta de acuerdo a su petición.

Nombre: ModeloComentario

Modelo

Descripción: Es la encargada de realizar las peticiones a los distintos procedimientos almacenados relacionados a la tabla **comentario** de la base de datos a través de la clase SbieP_PDO y enviar una respuesta a su respectiva clase controladora de acuerdo a la solicitud realizada.

Nombre: ModeloEvento

Modelo

Descripción: Es la encargada de realizar las peticiones a los distintos procedimientos almacenados relacionados a la tabla **evento** de la base de datos a través de la clase SbieP_PDO y enviar una respuesta a su respectiva clase controladora de acuerdo a la solicitud realizada.

Nombre: ModeloParticipante

Modelo

Descripción: Es la encargada de realizar las peticiones a los distintos procedimientos almacenados relacionados a la tabla **participante** de la base de datos a través de la clase SbieP_PDO y enviar una respuesta a su respectiva clase controladora de acuerdo a la solicitud realizada.

Nombre: ModeloTrabajo

Modelo

Descripción: Es la encargada de realizar las peticiones a los distintos procedimientos almacenados relacionados a la tabla **trabajo** de la base de datos a través de la clase SbieP_PDO y enviar una respuesta a su respectiva clase controladora de acuerdo a la solicitud realizada.

Nombre: SbieP_PDO

Modelo

Descripción: Es la encargada de establecer la conexión con la base de datos y le permite a las distintas clases modelo, realizar sus peticiones.

3.1.5 Diagramas de interacción

Los diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento.

Habitualmente, un diagrama de interacción capta el comportamiento de un solo caso de uso. El diagrama muestra cierto número de ejemplos de objetos y el flujo de mensajes que ocurren entre estos objetos dentro del caso de uso.

3.1.6 Diseño de la base de datos

Este modelo proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Se obtiene a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML, siendo los elementos esenciales del modelo las entidades, los atributos y las relaciones entre las entidades. A continuación se muestra el modelo de datos del sistema.

Capítulo 3: Diseño y construcción del sistema

Nombre: evento		
Descripción: Contiene los datos principales de cada uno de los eventos que se crean.		
Atributos	Tipo	Descripción
id_evento	integer	Número de identificación que va a tener cada uno de los eventos.
nombre_e	varchar	Nombre del evento.
edicion	varchar	Edición en la que se encuentra un evento.
dedicado	text	Sobre qué es el evento.
fecha_i	date	Fecha de inicio del evento.
fecha_c	date	Fecha de clausura del evento.

Nombre: participante		
Descripción: Contiene los datos principales de cada uno de los participantes que se encuentran en un evento.		
Atributos	Tipo	Descripción
id_participante	integer	Número de identificación que va a tener cada uno de los participantes.
nombre_p	varchar	Nombre del participante.
apellidos	varchar	Apellidos del participante.
fecha_nacimiento	date	Fecha de nacimiento del participante.
grado_cientifico	varchar	Grado científico del participante.
grado_educacional	varchar	Grado educacional del participante.
dir_curriculum	text	Aquí se guarda la dirección donde se guardará el curriculum adicionado por el participante.
id_tipo	integer	Se guarda el id del tipo de participante que sea (invitado o autor), por lo que constituye una llave foránea de la tabla tipo_participante.

Capítulo 3: Diseño y construcción del sistema

Nombre: tipo_participante		
Descripción: Contiene el tipo de participante que puede existir: invitado o autor.		
Atributos	Tipo	Descripción
id_tipo	integer	Número de identificación que va a tener cada tipo de participante.
tipo	varchar	Contiene el tipo de participante: invitado o autor.

Nombre: evento_participante		
Descripción: Contiene los identificadores de los distintos eventos con sus participantes.		
Atributos	Tipo	Descripción
id_evento_participante	integer	Número de identificación que va a tener esta tabla.
id_evento	integer	Se guarda el id de los distintos eventos adicionados, por lo que constituye una llave foránea de la tabla evento.
id_participante	integer	Se guarda el id de los distintos participantes a un evento determinado, por lo que constituye una llave foránea de la tabla participante.

Nombre: modalidad		
Descripción: Contiene las diferentes modalidades de un evento.		
Atributos	Tipo	Descripción
id_modalidad	integer	Número de identificación que va a tener cada una de las modalidades.
nombre_m	varchar	Nombre de la modalidad.

Capítulo 3: Diseño y construcción del sistema

descripcion	text	Ofrece una breve descripción de la modalidad.
-------------	-------------	---

Nombre: evento_modalidad		
Descripción: Contiene los diferentes identificadores de los eventos con sus respectivas modalidades.		
Atributos	Tipo	Descripción
id_evento_modalidad	integer	Número de identificación que va a tener la tabla.
id_evento	integer	Se guarda el id de los distintos eventos adicionados, por lo que constituye una llave foránea de la tabla evento.
id_modalidad	integer	Se guarda el id de las distintas modalidades de un evento determinado, por lo que constituye una llave foránea de la tabla modalidad.

Nombre: sede		
Descripción: Contiene las diferentes sedes donde se efectuará un evento.		
Atributos	Tipo	Descripción
id_sede	integer	Número de identificación que va a tener cada una de las sedes.
nombre	text	Nombre de la sede.
direccion	text	Dirección donde se localiza la sede.
descripcion	text	En este campo se guarda una breve descripción de la sede.

Capítulo 3: Diseño y construcción del sistema

Nombre: evento_sede		
Descripción: Contiene los diferentes identificadores de los eventos con sus respectivas sedes.		
Atributos	Tipo	Descripción
id_evento_sede	integer	Número de identificación que va a tener la tabla.
id_evento	integer	Se guarda el id de los distintos eventos adicionados, por lo que constituye una llave foránea de la tabla evento.
id_sede	integer	Se guarda el id de las distintas sedes de un evento determinado, por lo que constituye una llave foránea de la tabla sede.

Nombre: patrocinador		
Descripción: Contiene los diferentes patrocinadores de un evento.		
Atributos	Tipo	Descripción
id_patrocinador	integer	Número de identificación que va a tener cada uno de los patrocinadores.
correo	varchar	Se guarda el correo (si lo posee) del patrocinador.
vínculo_web	varchar	Se guarda los enlaces web que posee el patrocinador.
aporte_monetario	float	Aporte en moneda o material a un evento.
id_categoria	integer	Se guarda el id de la categoría que puede obtener un patrocinador en un evento, por lo que constituye una llave foránea de la tabla categoria.

Capítulo 3: Diseño y construcción del sistema

Nombre: persona		
Descripción: Contiene las diferentes personas que pueden ser patrocinadores. Esta tabla es hija de patrocinador, por lo que su llave primaria va a ser la misma.		
Atributos	Tipo	Descripción
id_patrocinador	integer	Número de identificación que va a tener cada uno de las personas.
nombre	varchar	Se guarda el nombre de la persona.
apellidos	varchar	Se guarda los apellidos de una persona.
fecha_nacimiento	date	Fecha de nacimiento de la persona.

Nombre: institución		
Descripción: Contiene las diferentes instituciones que pueden ser patrocinadoras. Esta tabla es hija de patrocinador, por lo que su llave primaria va a ser la misma.		
Atributos	Tipo	Descripción
id_patrocinador	integer	Número de identificación que va a tener cada uno de las personas.
nombre_i	text	Se guarda el nombre de la institución.
descripcion	varchar	Se guarda una breve descripción de la institución.
direccion	text	Dirección donde se encuentra localizada la institución.

Nombre: evento_patrocinador		
Descripción: Contiene los diferentes identificadores de los eventos con sus respectivos patrocinadores.		
Atributos	Tipo	Descripción
id_evento_patrocinador	integer	Número de identificación que va a tener la tabla.
id_evento	integer	Se guarda el id de los distintos

Capítulo 3: Diseño y construcción del sistema

		eventos adicionados, por lo que constituye una llave foránea de la tabla evento.
id_patrocinador	integer	Se guarda el id de los distintos patrocinadores de un evento determinado, por lo que constituye una llave foránea de la tabla patrocinador.

Nombre: actividad		
Descripción: Contiene las distintas actividades que se realizarán en un evento determinado.		
Atributos	Tipo	Descripción
id_actividad	integer	Número de identificación que va a tener cada una de las actividades.
nombre	integer	Nombre de la actividad.
fecha	date	Fecha en la cual se realizará la actividad.
id_trabajo	integer	Se guarda el id de los distintos trabajos que se expondrán, por lo que constituye una llave foránea de la tabla trabajo.
id_sede	integer	Se guarda el id de las distintas sedes donde se efectuará una actividad, por lo que constituye una llave foránea de la tabla sede.
id_local	integer	Se guarda el id de los distintos locales donde se efectuará una actividad, por lo que constituye una llave foránea de la tabla local.

Capítulo 3: Diseño y construcción del sistema

Nombre: trabajo		
Descripción: Contiene los distintos trabajos presentados en un evento.		
Atributos	Tipo	Descripción
id_trabajo	integer	Número de identificación que va a tener cada uno de los trabajos.
nombre	integer	Nombre del trabajo.
resumen	text	Breve resumen del trabajo.
direccion_t	integer	Se guarda la dirección donde se encontrará localizado el trabajo.
id_participante	integer	Se guarda el id del participante que entregó un trabajo, por lo que constituye una llave foránea de la tabla participante.
id_modalidad	integer	Se guarda el id de la modalidad en la que se expondrá un trabajo, por lo que constituye una llave foránea de la tabla modalidad.

Nombre: contacto		
Descripción: Contiene los distintos contactos que se realizan a un trabajo.		
Atributos	Tipo	Descripción
id_contacto	integer	Número de identificación que va a tener cada uno de los contactos.
nombre_c	text	Nombre del contacto
apellidos_c	text	Apellidos del contacto.
direccion_c	text	Dirección de correo para contactar.
id_trabajo	integer	Se guarda el id del trabajo que tenga algún contacto por lo que constituye una llave foránea de la tabla trabajo.

Capítulo 3: Diseño y construcción del sistema

Nombre: local		
Descripción: Contiene los distintos locales que posee una sede.		
Atributos	Tipo	Descripción
id_local	integer	Número de identificación que va a tener cada uno de los locales.
nombre	integer	Nombre del local.
capacidad	integer	Capacidad que posee el local.
descripcion	text	Breve descripción de el local.
dimension	text	Dimensión que posee el local.
id_sede	integer	Se guarda el id de la sede en la que se encuentra el local, por lo que constituye una llave foránea de la tabla sede.

Nombre: foto		
Descripción: Contiene las fotos de los distintos locales y participantes		
Atributos	Tipo	Descripción
id_foto	integer	Número de identificación que va a tener cada una de las fotos.
direccion	text	Dirección en donde se encontrará localizada la foto.
id_sede	integer	Se guarda el id de la sede que posee fotos, por lo que constituye una llave foránea de la tabla sede.
id_participante	integer	Se guarda el id de los participantes que poseen fotos, por lo que constituye una llave foránea de la tabla participante.

Capítulo 3: Diseño y construcción del sistema

Nombre: categoria		
Descripción: Contiene las categorías que posee un patrocinador.		
Atributos	Tipo	Descripción
id_categoria	integer	Número de identificación que va a tener cada una de las categorías.
categoria	varchar	Categoría que puede tener un patrocinador.
ventajas	text	Ventajas que puede tener una categoría.

Nombre: comentario		
Descripción: Contiene los distintos comentarios realizados a un trabajo		
Atributos	Tipo	Descripción
id_comentario	integer	Número de identificación que va a tener cada una de los comentarios.
comentario	text	Se guardará el comentario que se le realizará a un trabajo.
id_trabajo	integer	Se guarda el id de los trabajos que poseen comentarios, por lo que constituye una llave foránea de la tabla trabajo.

3.2 Principios de diseño gráfico y estándares de codificación

3.2.1 Principios de diseño gráfico

El desarrollo de la aplicación estuvo encaminado hacia un diseño sencillo, atractivo, y de fácil navegación; lográndose gracias a las miles de facilidades y ventajas que trae incorporadas el CMS Joomla en cuanto a diseño, por su gran variedad de plantillas. Por otra parte, el módulo de administración para la edición y publicación de contenido que trae el propio CMS logra que los editores

y administradores del portal realicen a voluntad y de forma eficiente un uso estructurado de los diferentes contenidos que se manejarán en el sitio.

3.2.2 Estándares

Durante la programación de la aplicación es fundamental regirse por un estándar de codificación para garantizar entre otros aspectos uniformidad. En cuanto al estándar de codificación se puede plantear que comprende todos los aspectos de la generación de código. La legibilidad del código, fuente que repercute directamente en la comprensión de un software por parte de un programador. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento.

3.2.2.1 Identación

Se dejaron dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.

```
public function EliminarEvento($id_evento)
{
    $pQuery='CALL pa_eliminar_evento(?)';
    $sth = $this->conn->prepare($pQuery);
    $this->conn->bind(1, (int) $id_evento, PDO::PARAM_INT);
    $data = $this->conn->query();
    if (is_array($data))
        return true;
    return false;
}
```

Fig. 12 Identación de una función.

3.2.2.2 Comentarios

Se realizaron comentarios al principio de cada función para un mayor entendimiento de la misma, así como los parámetros que usa.

```
/**
 * Funcion Registrar_Evento
 *
 * @param string $nombre
 * @param string $edicion
 * @param string $dedicado
 * @param string $fecha_inicio
 * @param string $fecha_clausura
 * @return bool;
 */
```

Fig. 13 Comentario de la función EliminarEvento.

3.2.2.3 Líneas en blanco

Se emplearon antes y después de una función realizada, así como también entre los diferentes comentarios antes mencionados.

```
 * Funcion Registrar_Evento[]

public function RegistrarEvento($nombre,$edicion,$dedicado,$fecha_inicio,$fecha_clausura)

 * Funcion Eliminar_Evento[]

public function EliminarEvento($id_evento)
```

Fig. 14 Líneas en blanco entre comentarios y funciones.

3.2.2.4 Clases y Objetos

Los nombres de las clases comenzarán con una letra T en mayúscula, la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing. Ejemplo: TMiClase. Para nombrar las funciones se utilizarán verbos que denoten la acción que hace la función, utilizando la misma notación.

```
class TModeloSede public function RegistrarSede
```

Fig. 15 Nombre de clases y funciones.

3.3 Modelo de implementación

Un modelo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes, muestran la estructura del código (diagrama de componentes). Describe además como se organizan y se relacionan unos con otros.

3.3.1 Diagrama de componentes

El diagrama de componentes ayuda a un mejor entendimiento del modelo de implementación. Con él se representan los componentes lógicos de la aplicación así como las relaciones de dependencia que existen entre ellos. A continuación se presenta el diagrama de componentes general del portal web.

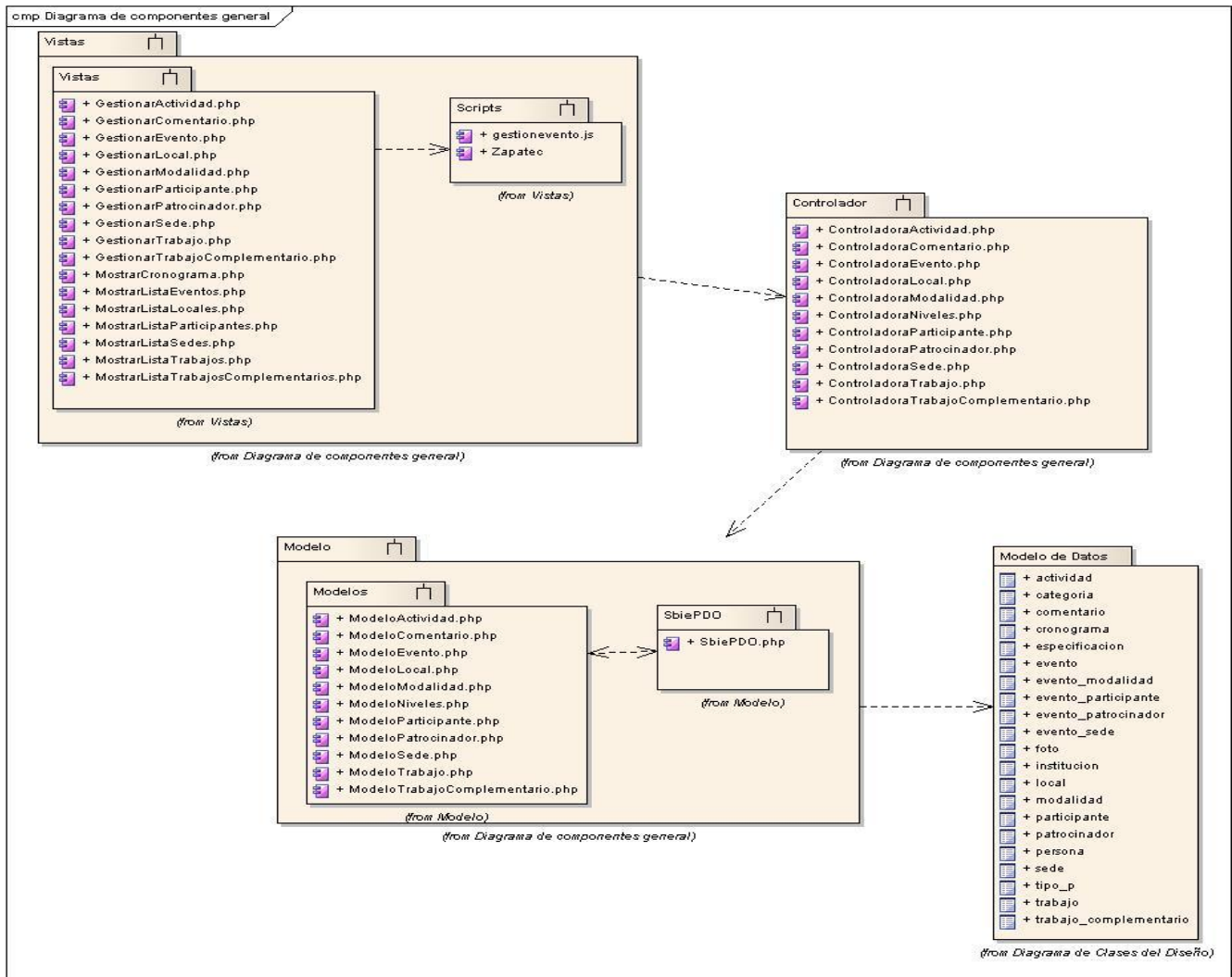


Fig. 16 Diagrama de componentes general.

Capítulo 3: Diseño y construcción del sistema

A continuación se presentan algunos diagramas de componentes para los principales casos de uso.

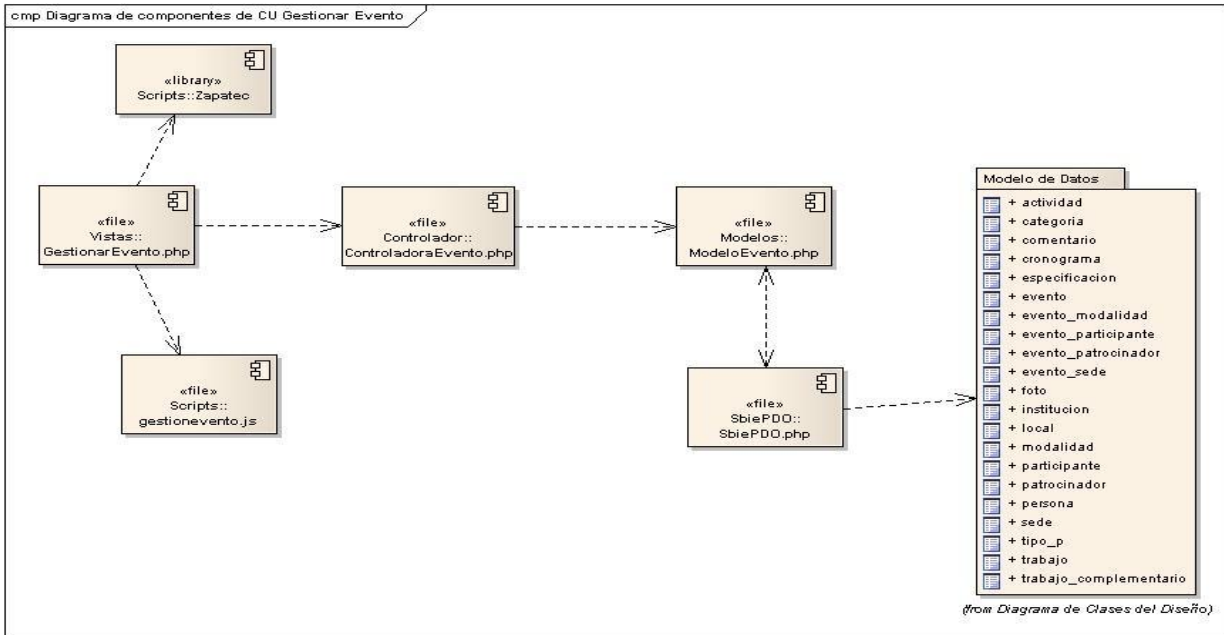


Fig. 17 Diagrama de componentes para el caso de uso Gestionar Evento.

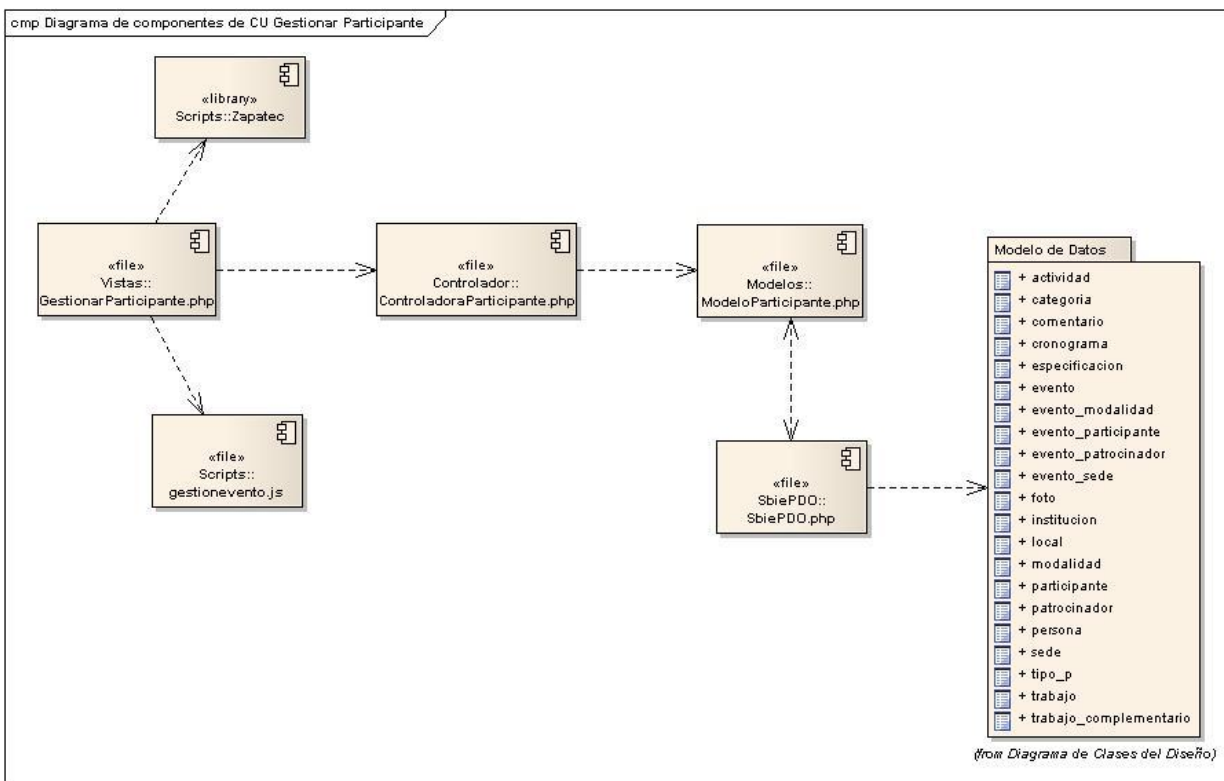


Fig. 18 Diagrama de componentes para el caso de uso Gestionar Participante.

Capítulo 3: Diseño y construcción del sistema

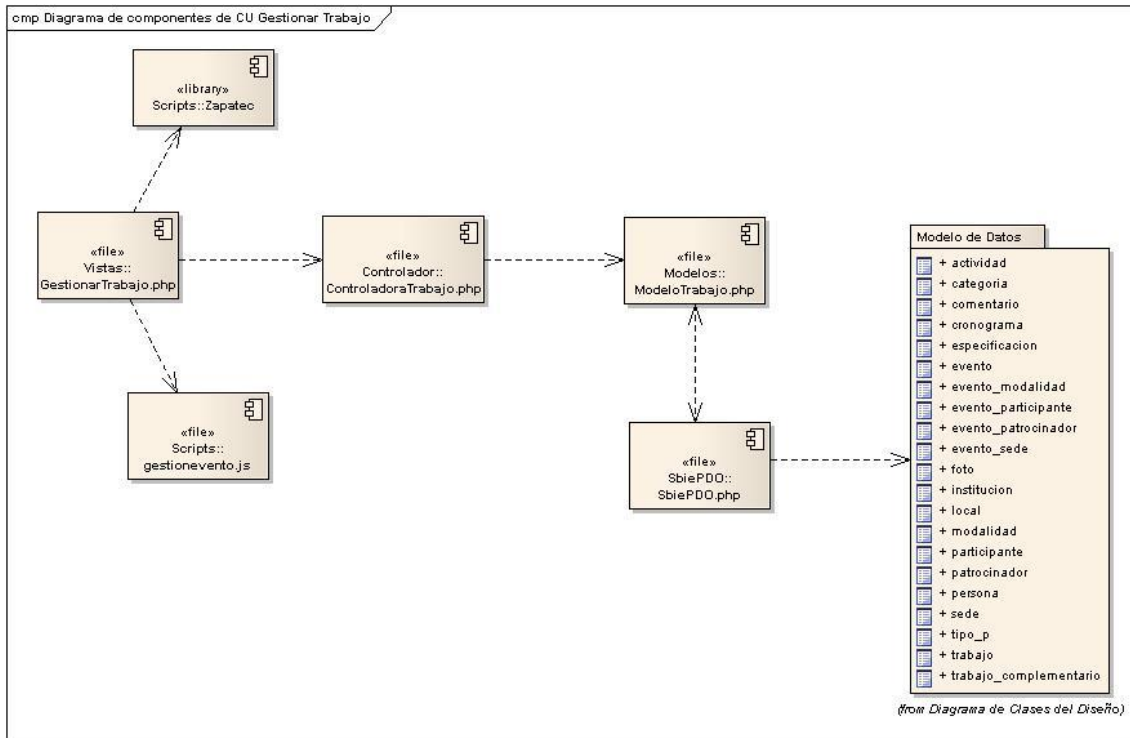


Fig. 19 Diagrama de componentes para el caso de uso Gestionar Trabajo.

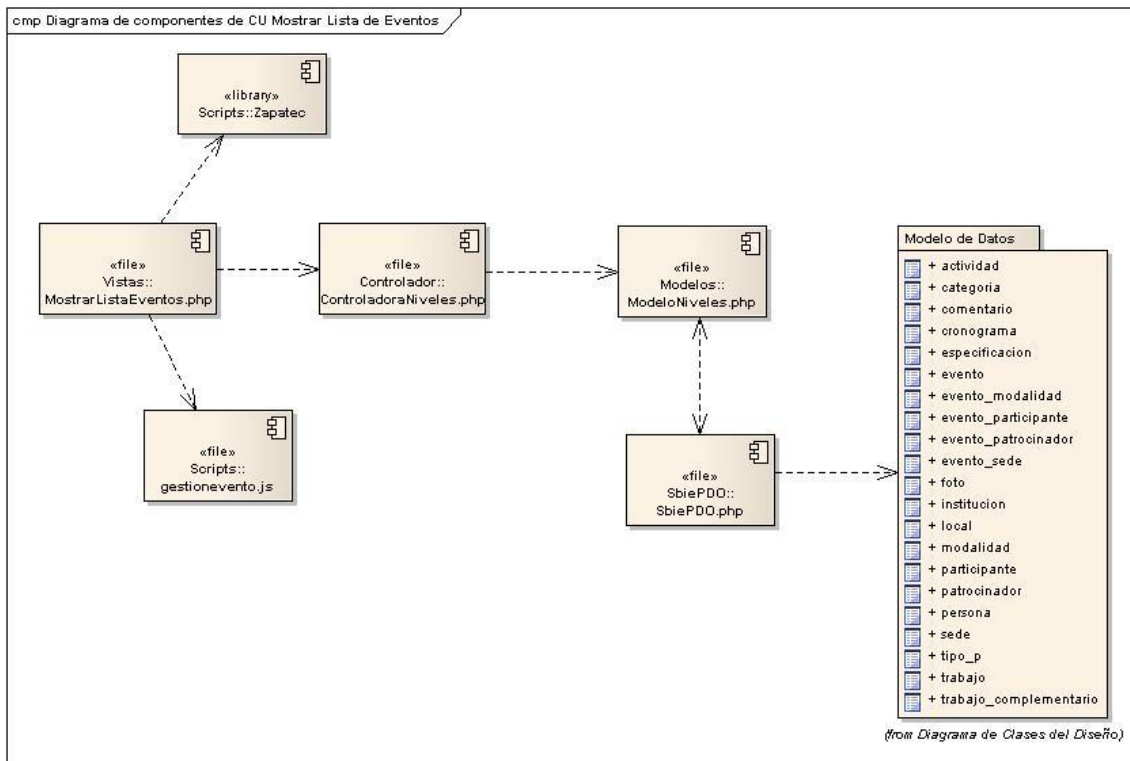


Fig. 20 Diagrama de componentes para el caso de uso Mostrar Lista de Eventos.

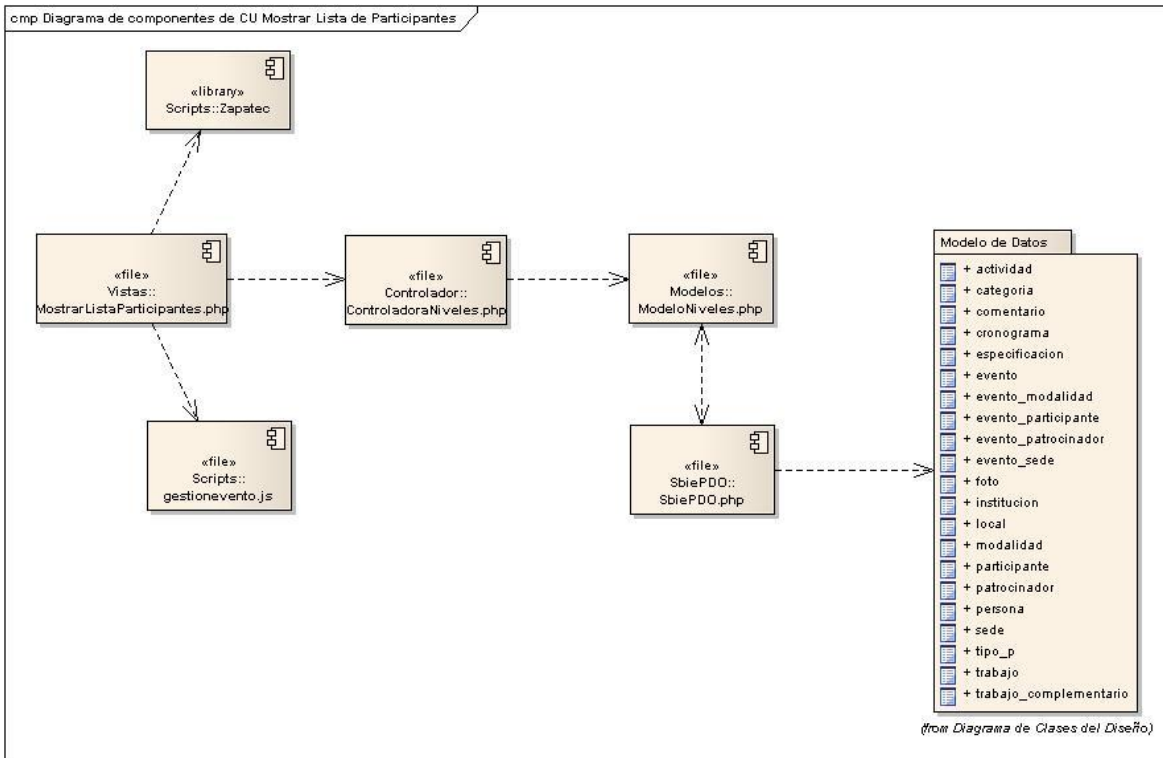


Fig. 21 Diagrama de componentes para el caso de uso Mostrar Lista de Participantes.

3.4 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. A continuación se muestra el diagrama del portal.

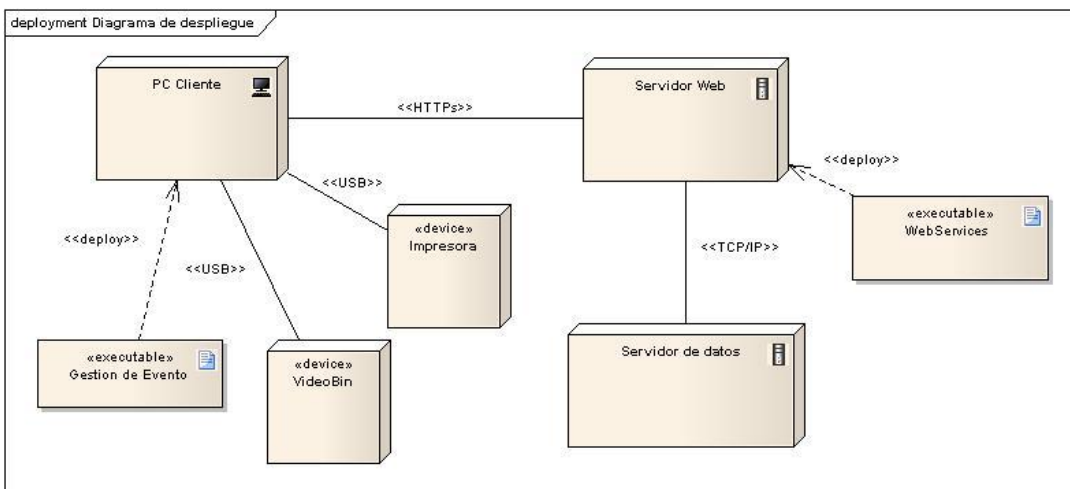


Fig. 22 Diagrama de despliegue.

Capítulo 3: Diseño y construcción del sistema

Conclusiones

En este capítulo, se llevaron a cabo los pasos para el diseño y realización del sistema. Se definieron los distintos patrones de diseño a utilizar. Se representaron mediante diagramas de clases, la lógica del negocio del sistema. Se diseñaron y representaron mediante un diagrama el modelo de datos, para el almacenamiento de la información del sistema. Así cómo también se describió como se relacionan los componentes que forman la arquitectura y como es su distribución física y lógica.

Conclusiones

Tras dar cumplimiento a los objetivos y las tareas de investigación propuestas, los autores arribaron a las siguientes conclusiones:

- A partir del estudio de aplicaciones informáticas relacionadas a la gestión de eventos existentes en Cuba y el mundo se arribó a la conclusión de que éstas no ofrecen una solución estándar para la realización del portal web, pero constituyeron un punto de partida para acumular experiencias de cómo se gestiona un evento.
- Después de analizar las posibles herramientas y tecnologías para el desarrollo de la aplicación se implementó una aplicación web basada en la arquitectura Modelo Vista Controlador, utilizando PHP como lenguaje de programación, apoyándose en Zend Studio para Eclipse como IDE (Entorno de Desarrollo Integrado). Como gestor de base de datos MySQL y para la transmisión de datos entre los modelos y las vistas el lenguaje XML.
- Se generaron los artefactos relacionados con el desarrollo del sistema, establecidos por la metodología RUP.
- Se desarrolló una aplicación web que gestiona toda la información relacionada con los recursos humanos y científicos del evento NeuroRehabana 2010.

Beneficios

Con la utilización de este portal y del buen aprovechamiento del mismo se espera:

- Brindar a los usuarios una aplicación que permita gestionar todo los recursos humanos y científicos del evento NeuroRehabana 2010.
- Publicar servicios web que permitan ofrecer información relacionada con el evento a otros portales.
- Facilitar el trabajo del comité organizador, al brindarle un módulo de administración que les permita tener acceso a toda la información del evento, así como también manipularla en consideración de sus intereses.

Recomendaciones

Los autores del presente trabajo de diploma recomiendan:

- Implementar todo el proceso de gestión del evento científico NeuroRehabana, desde la convocatoria, inscripción de los participantes incluyendo un módulo para la gestión económica, hasta la conclusión del mismo.
- Ampliar el módulo de reportes de PDF que permita a los organizadores del evento obtener listados, resúmenes y datos necesarios para un posterior análisis, incluyendo además la exportación a otros formatos como Excel y Word.
- Continuar la actualización sistemática del CMS Joomla, para lograr una mayor utilización de la gestión de contenido que brinda.
- Implementar un módulo de seguridad que se encargará de gestionar el acceso a las sedes del evento por el personal autorizado.
- Desarrollar un visor de información que se utilice como canal informativo durante los días del evento.
- Mejorar el diseño del portal a través de la opinión de los usuarios que interactúen con el portal, para lograr interfaces más amigables.

Referencias Bibliográficas

1. **Creative Commons.** *UOC.* [En línea] 16 de enero de 2008.
<http://mosaic.uoc.edu/articulos/cms1204.html.todo>.
2. Ver Referencia No. 1.
3. **Neurowork 2009.** *eLiveware.* [En línea] <http://www.eliveware.com>.
4. **Salvador Sandoval.** *Gestkal XXI.* [En línea] <http://www.gestkal.com/>.
5. **QWeb 2009.** *QWeb.es.* [En línea] <http://www.qweb.es>.
6. **Christian Van Der Henst.** *Maestros del web.* [En línea]
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
7. **Miguel Angel Alvarez.** *Desarrollweb.com.* [En línea]
<http://www.desarrollweb.com/articulos/182.php>.
8. —. *Desarrollweb.com.* [En línea] <http://www.desarrollweb.com>.
9. **2SPACIOS, S.C.** *Scepe Studio.* [En línea] <http://www.esepestudio.com>.
10. **Garret, Jesse James.** *Ajax. Un nuevo acercamiento a las aplicaciones Web.* 2005.
11. **Prensa Internet, S.L.** *FinancialTech.* [En línea] <http://www.financialtech-mag.com>.
12. **Manuel Gutiérrez Heredia.** *Tufuncion.* [En línea] <http://www.tufuncion.com>.
13. **Miguel Angel Alvarez.** *Desarrollweb.com.* [En línea]
<http://www.desarrollweb.com/articulos/332.php>.

Bibliografía

2SPACIOS, S.C. *Scepe Studio*. [En línea] <http://www.esepestudio.com>.

Booch, G., Rumbaugh, J. y Jacobson, I. 1999. *El Lenguaje Unificado de Modelado*. 1999.

Christian Van Der Henst. *Maestros del web*. [En línea]

<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

Creative Commons. 2008. *UOC*. [En línea] 16 de enero de 2008.

<http://mosaic.uoc.edu/articulos/cms1204.html.todo>.

—. **2008.** *UOC*. [En línea] 16 de enero de 2008. <http://mosaic.uoc.edu/articulos/cms1204.html.todo>.

—. *Capítulo 2. IAGP 2005/06. Metodologías de desarrollo de software*. [En línea]

<http://www.um.es/docencia/barzana/IAGP/lagp2.html>.

Garret, Jesse James. 2005. *Ajax. Un nuevo acercamiento a las aplicaciones Web*. 2005.

Hispavista 2009. *Hispavista*. [En línea]

http://foros.hispavista.com/ingenieria_del_software/404/59211/m/manual-tutorial-guia-rup/.

J.J. Merelo. *Introducción a los sistemas de gestión de contenidos*. [En línea]

Manuel Gutiérrez Heredia. *Tufuncion*. [En línea] <http://www.tufuncion.com>.

Miguel Angel Alvarez. *Desarrollweb.com*. [En línea] <http://www.desarrolloweb.com/articulos/182.php>.

—. *Desarrolloweb.com*. [En línea] <http://www.desarrolloweb.com/articulos/332.php>.

—. *Desarrolloweb.com*. [En línea] <http://www.desarrolloweb.com>.

MySQL AB. *MySQL*. [En línea] <http://dev.mysql.com/doc/refman/5.0/es/index.html>.

Neurowork 2009. *eLiveware*. [En línea] <http://www.eliveware.com>.

Nielsen, J. 2002. *Usabilidad. Diseño de sitios web*. Madrid : Prentice Hall, 2002.

Prensa Internet, S.L. *Financiatech*. [En línea] <http://www.financiatech-mag.com>.

Pressman, R. 1999. *Software Engineering. A Practitioner's Approach*. Fourth Edition. 1999.

2008. Principios básicos de diseño para web. [En línea] 2008.

<http://www.un.org/spanish/Depts/dpi/seminario/pdf/>.

QWeb 2009. *QWeb.es*. [En línea] <http://www.qweb.es>.

Robertson, James. *Futuro de los CMS*. [En línea]

http://www.steptwo.com.au/papers/cmb_future/index.html.

—. **2009.** *¿Qué es un Sistema Gestor de Contenido (CMS)?* [En línea] 01 de 2009.

http://www.steptwo.com.au/papers/kmc_what/index.html..

—. *¿Cómo evaluar un Sistema Gestor de Contenido (CMS)?*. [En línea]

http://www.steptwo.com.au/papers/kmc_evaluate/index.html..

S.H, Desarrollo de Web. *Joomlaos.net*. [En línea] <http://www.joomlaos.net/caracteristicas-de-joomla.php>.

Salvador Sandoval. *Gestkal XXI*. [En línea] <http://www.gestkal.com/>.

Anexos

Anexos 1. Diagrama de secuencia de los casos de uso significativos.

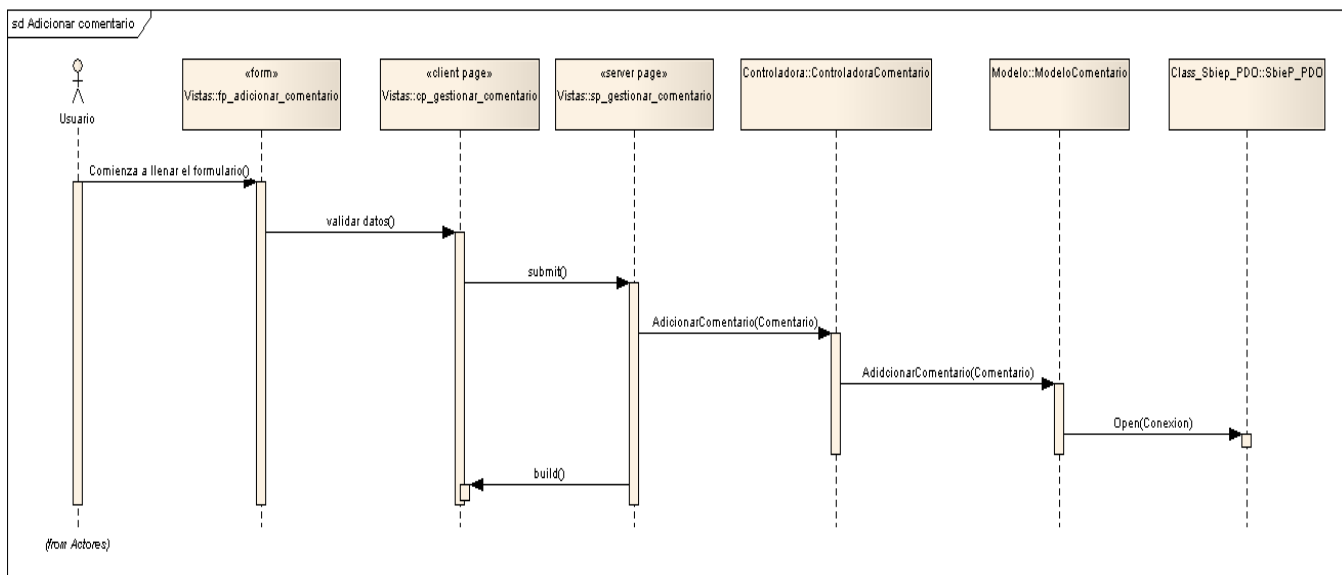


Fig. 23 Diagrama de secuencia de la sección Adicionar comentario del caso de uso Gestionar comentario.

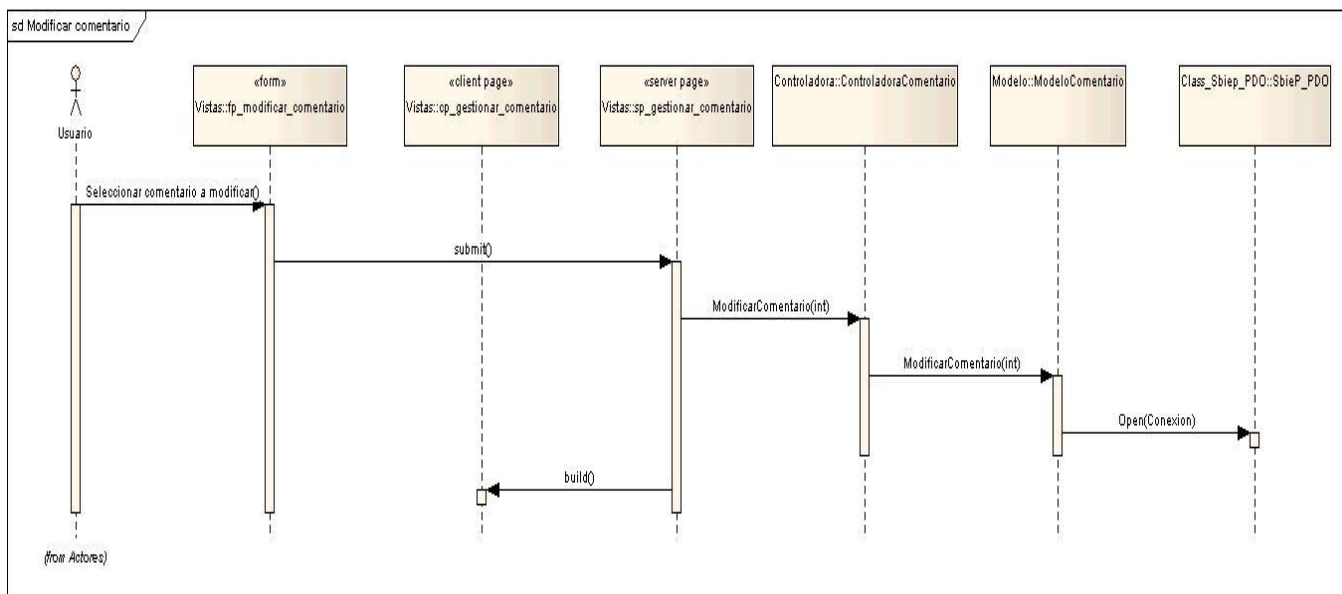


Fig. 24 Diagrama de secuencia de la sección Modificar comentario del caso de uso Gestionar comentario.

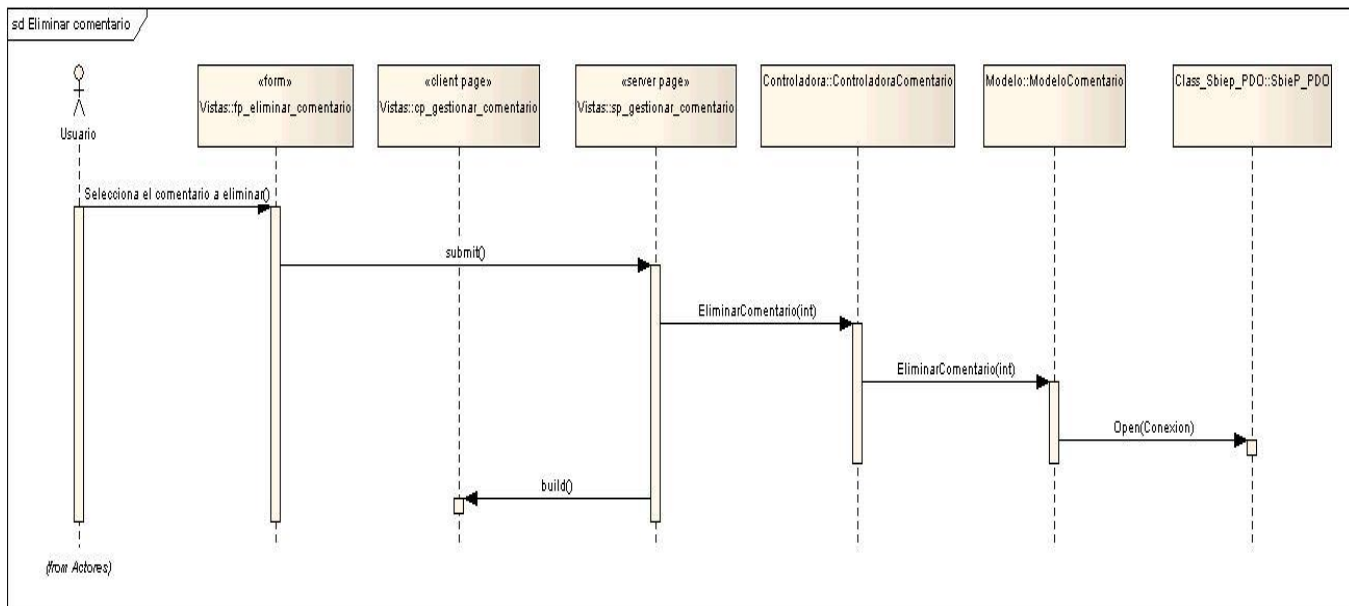


Fig. 25 Diagrama de secuencia de la sección Eliminar comentario del caso de uso Gestionar comentario.

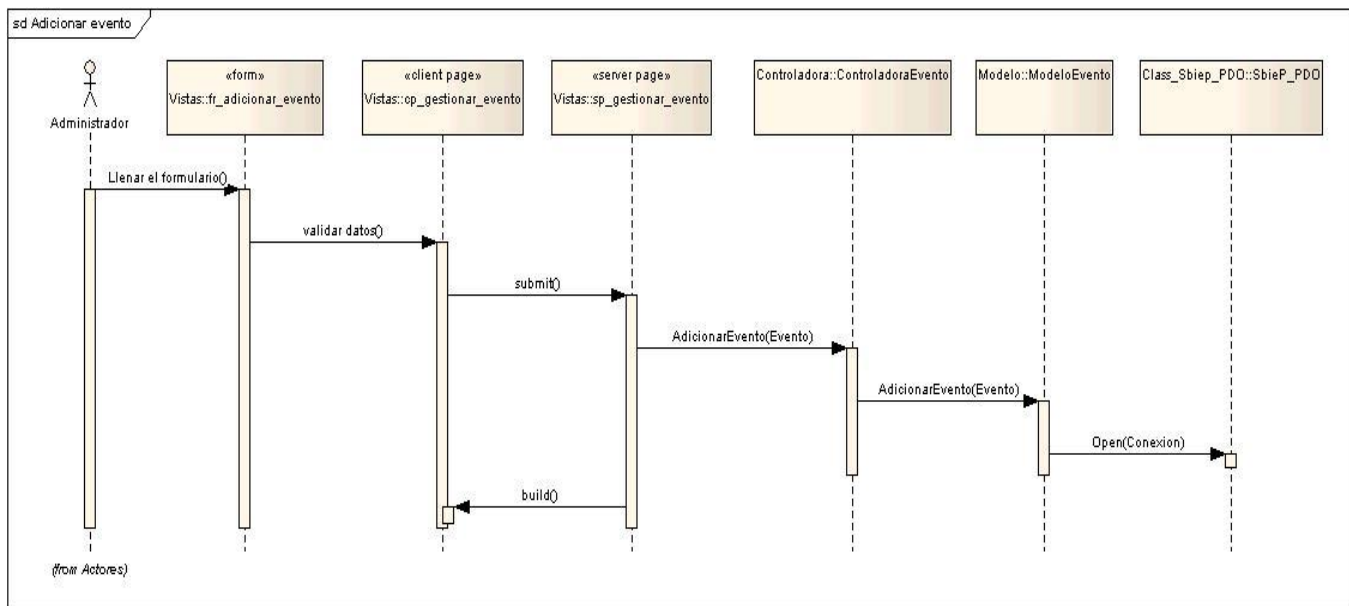


Fig. 26 Diagrama de secuencia de la sección Adicionar evento del caso de uso Gestionar evento.

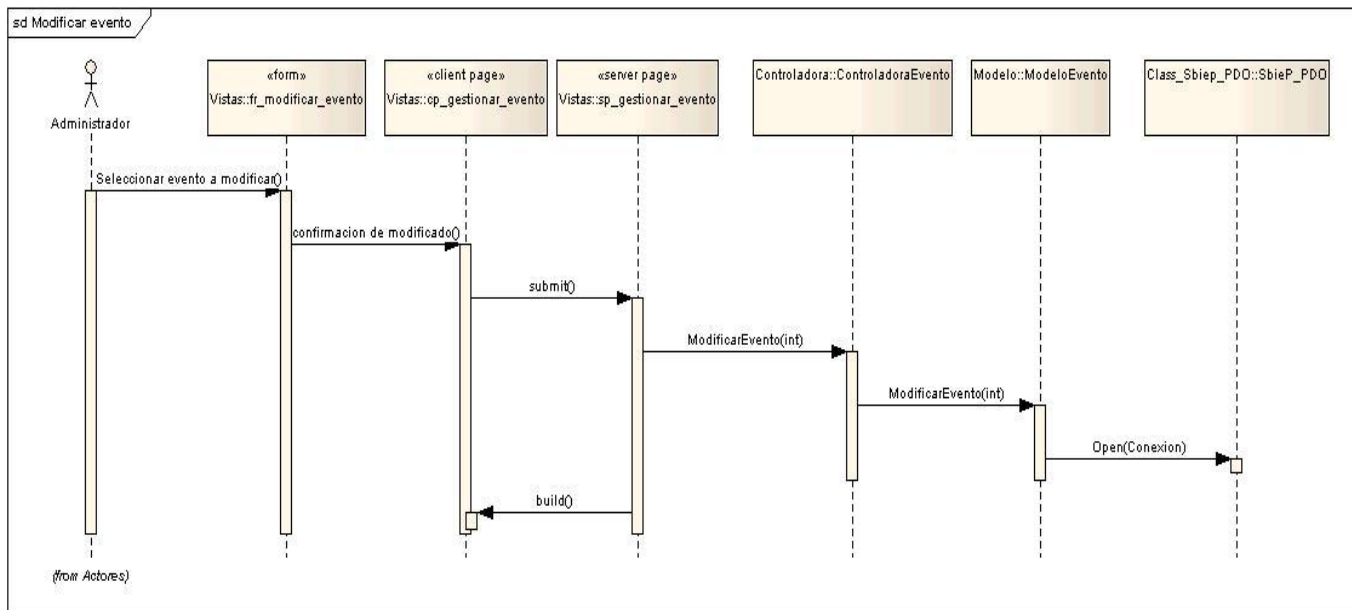


Fig. 27 Diagrama de secuencia de la sección Modificar evento del caso de uso Gestionar evento.

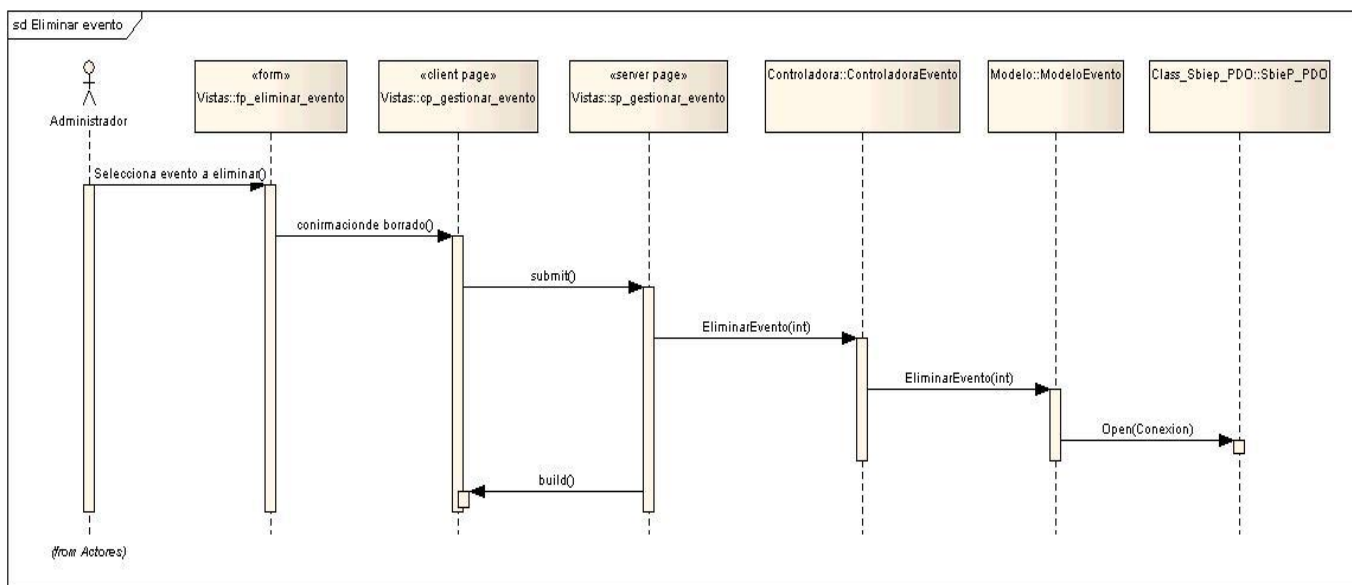


Fig. 28 Diagrama de secuencia de la sección Eliminar evento del caso de uso Gestionar evento.

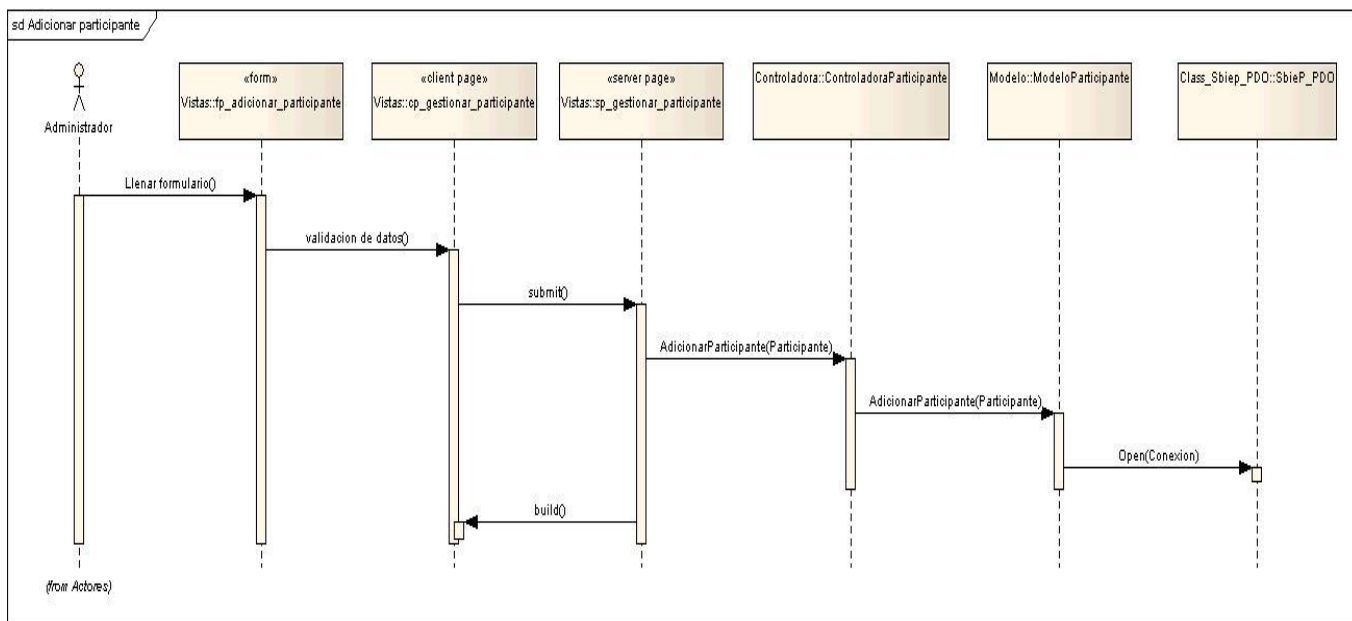


Fig. 29 Diagrama de secuencia de la sección Adicionar participante del caso de uso Gestionar participante.

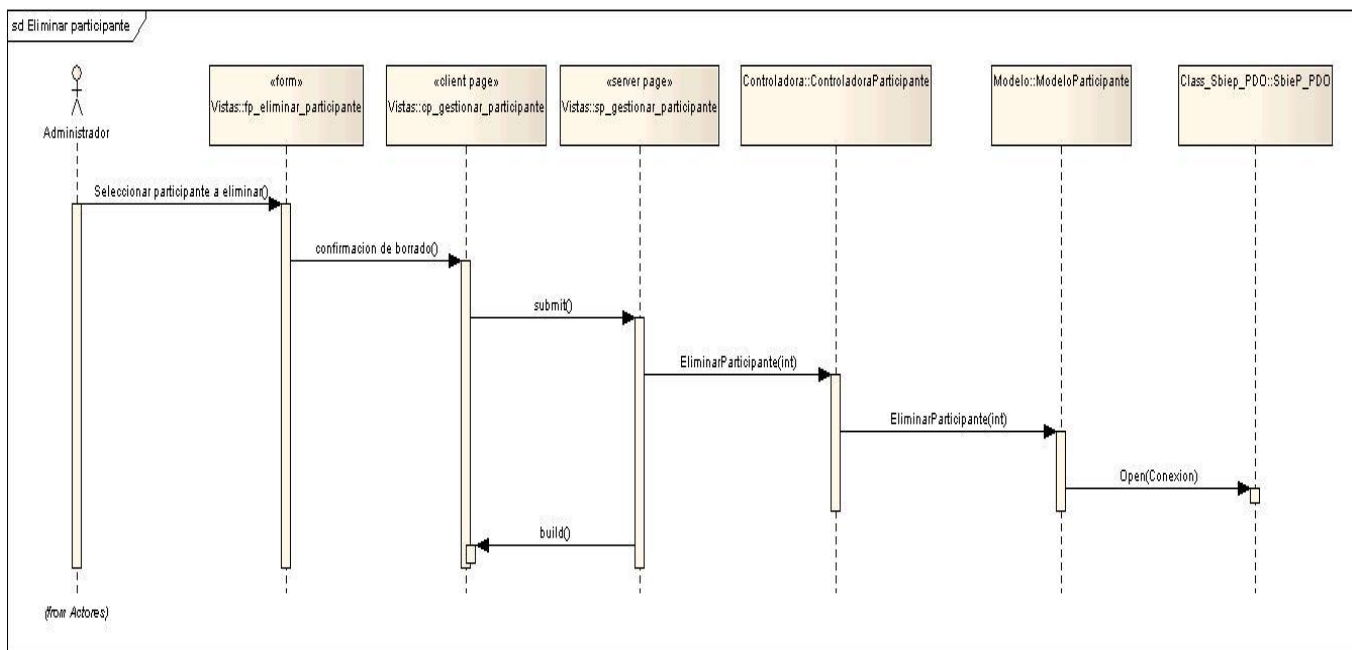


Fig. 30 Diagrama de secuencia de la sección Eliminar participante del caso de uso Gestionar participante.

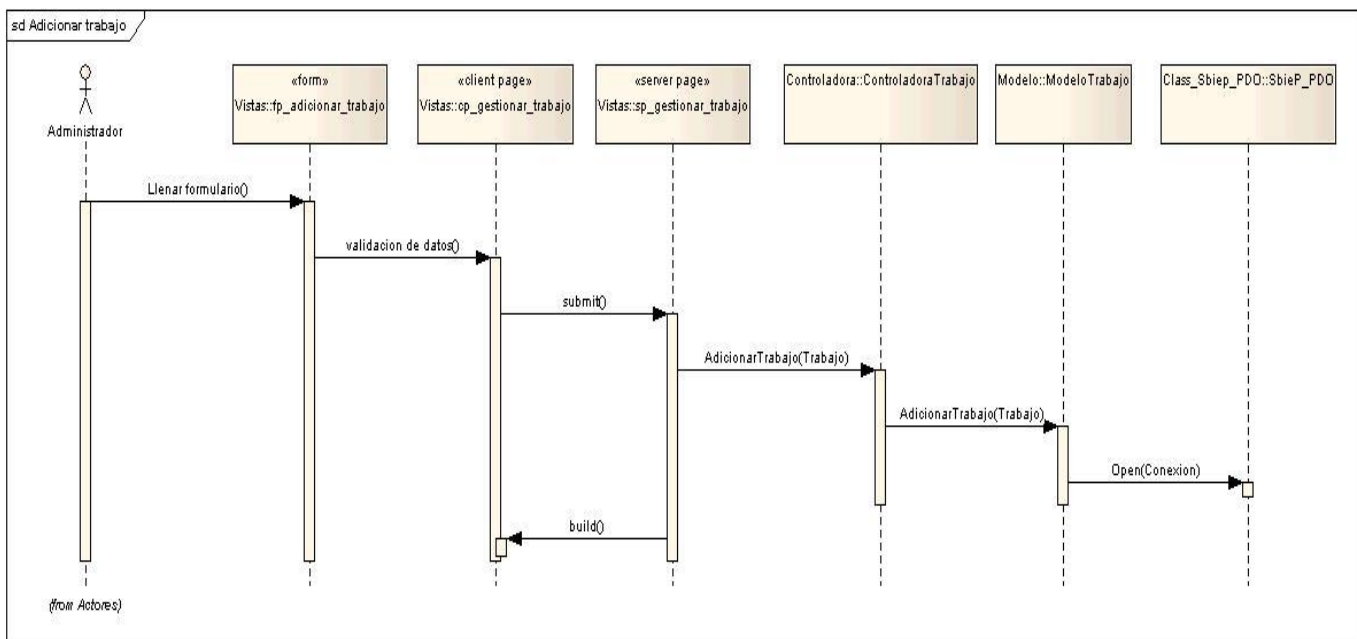


Fig. 30 Diagrama de secuencia de la sección Eliminar participante del caso de uso Gestionar participante.

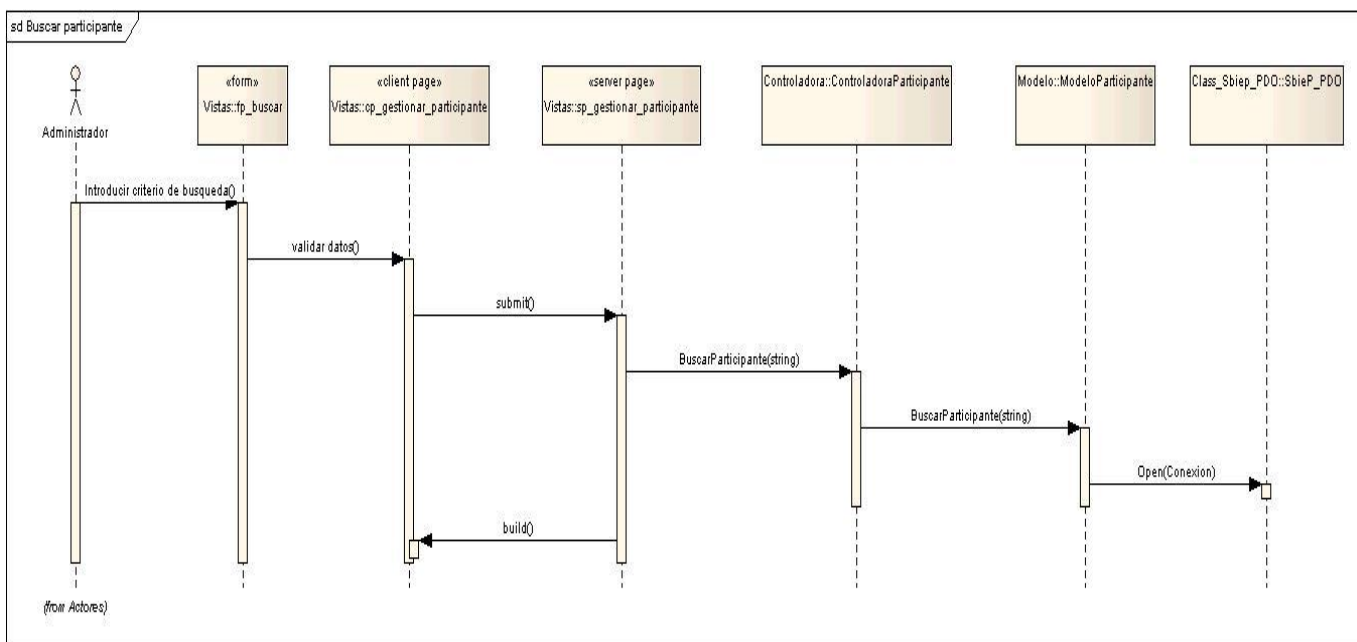


Fig. 31 Diagrama de secuencia del caso de uso Buscar participante.

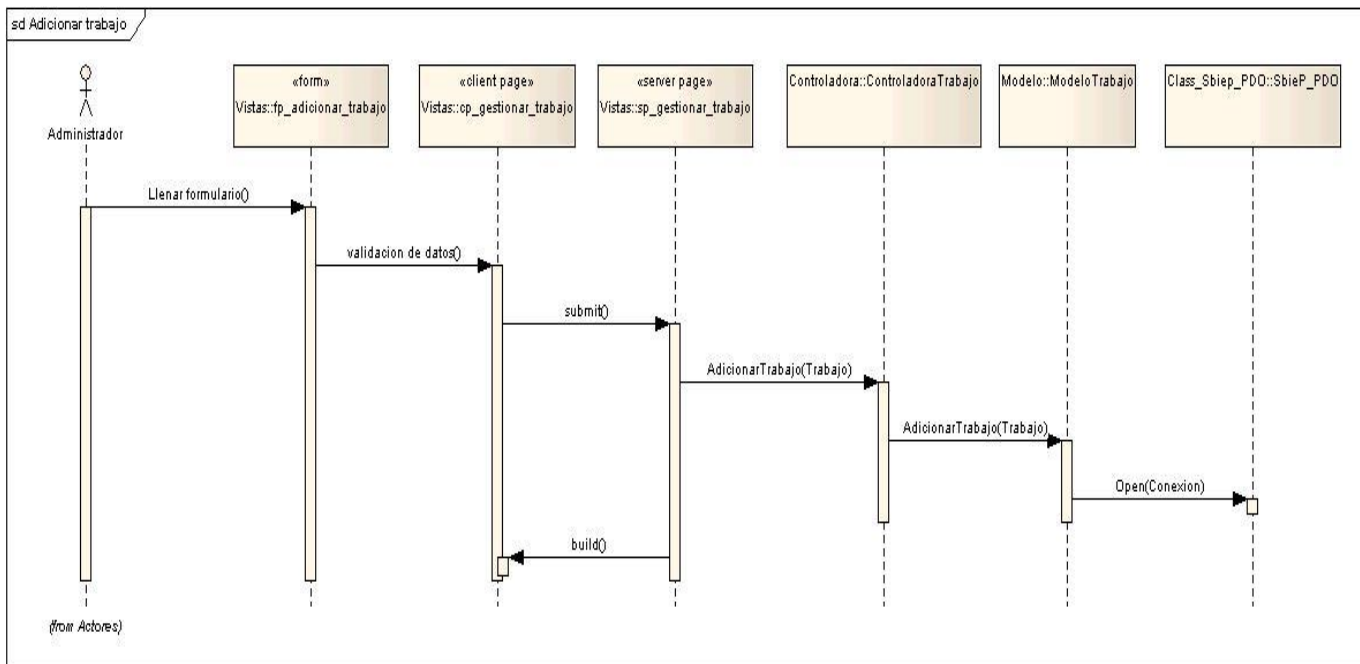


Fig. 32 Diagrama de secuencia de la sección Adicionar trabajo del caso de uso Gestionar trabajo.

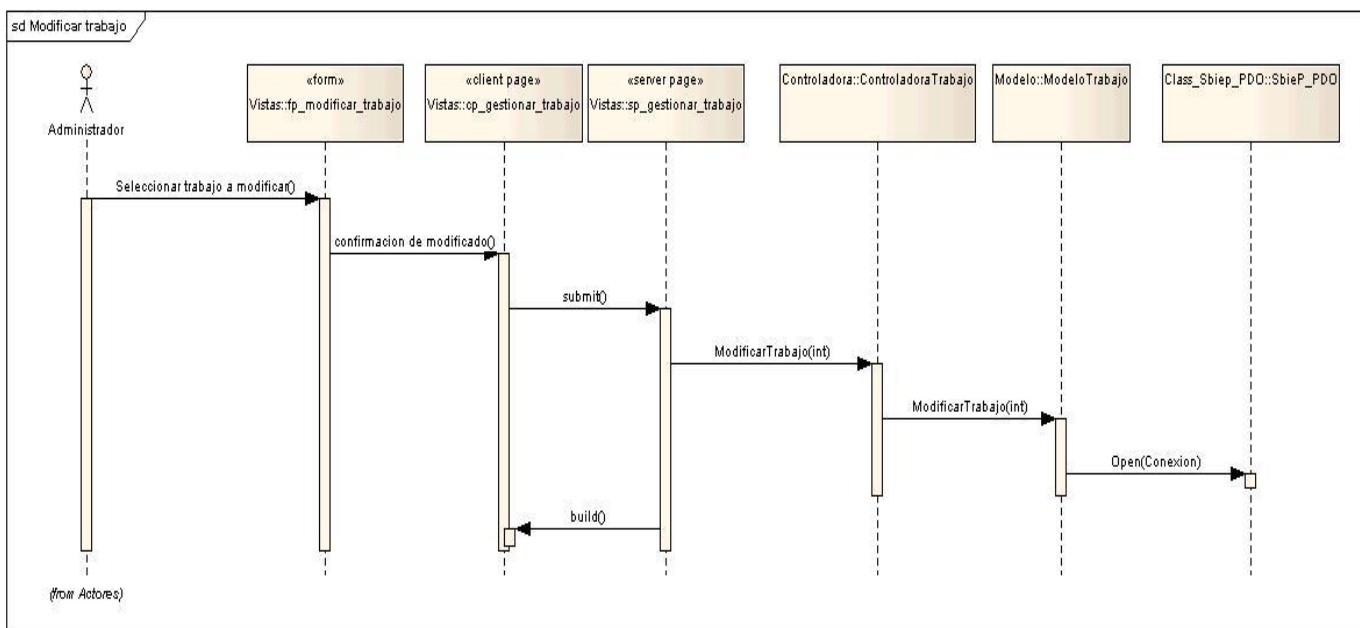


Fig. 33 Diagrama de secuencia de la sección Modificar trabajo del caso de uso Gestionar trabajo.

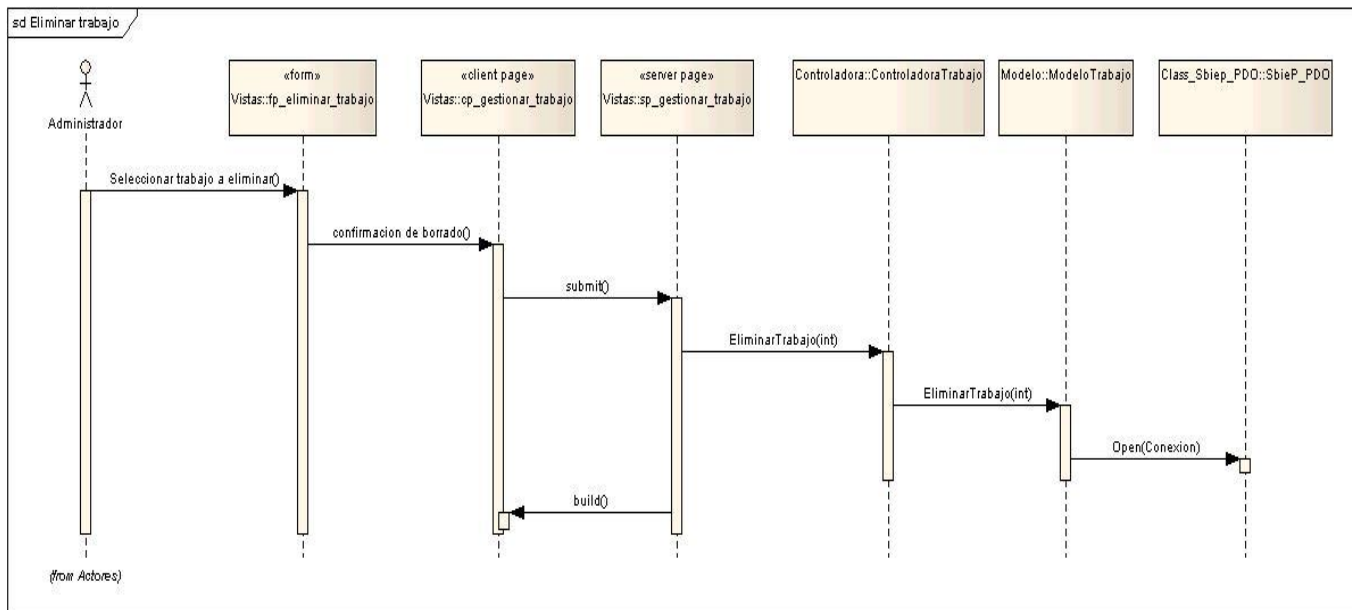


Fig. 34 Diagrama de secuencia de la sección Eliminar trabajo del caso de uso Gestionar trabajo.

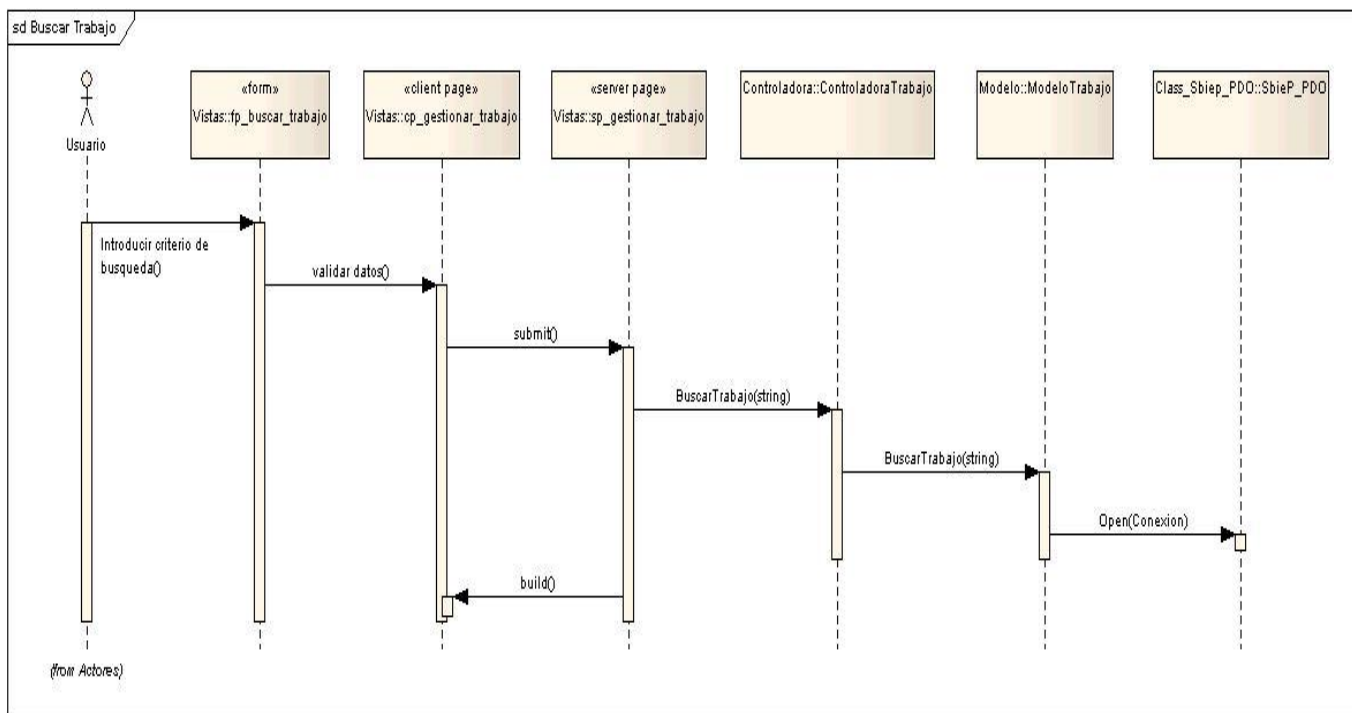


Fig. 35 Diagrama de secuencia del caso de uso Buscar trabajo.

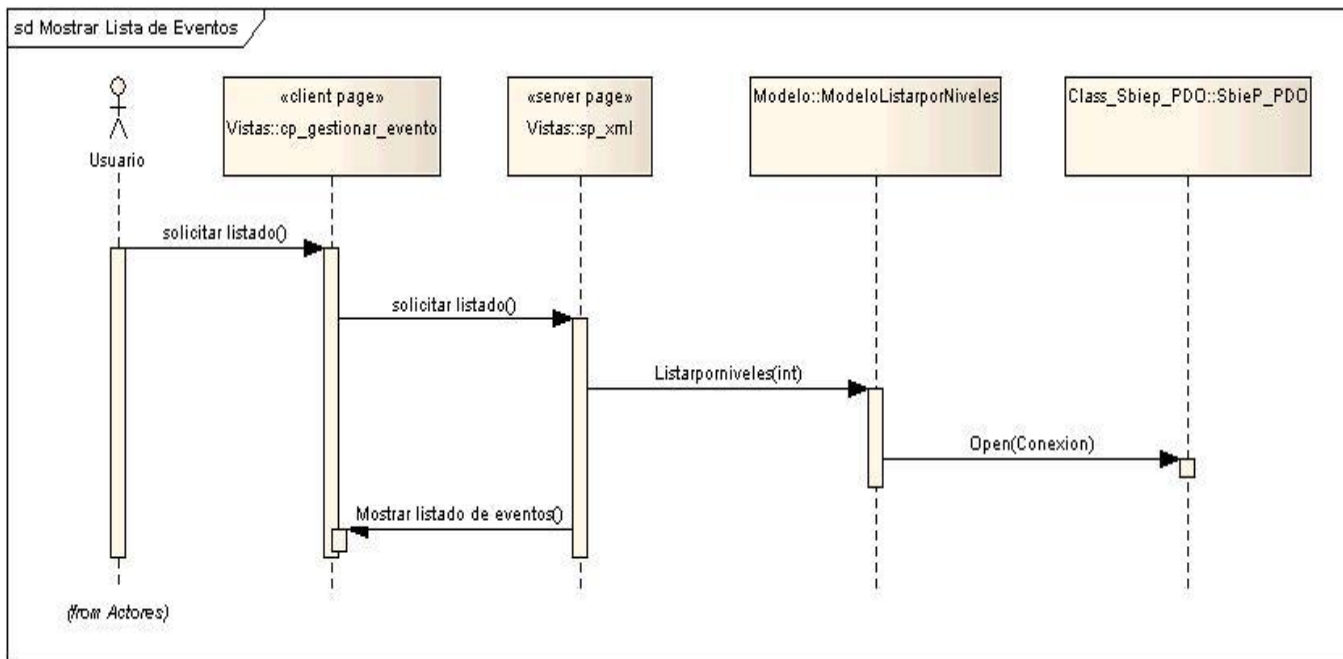


Fig. 36 Diagrama de secuencia del caso de uso Mostrar lista de eventos.

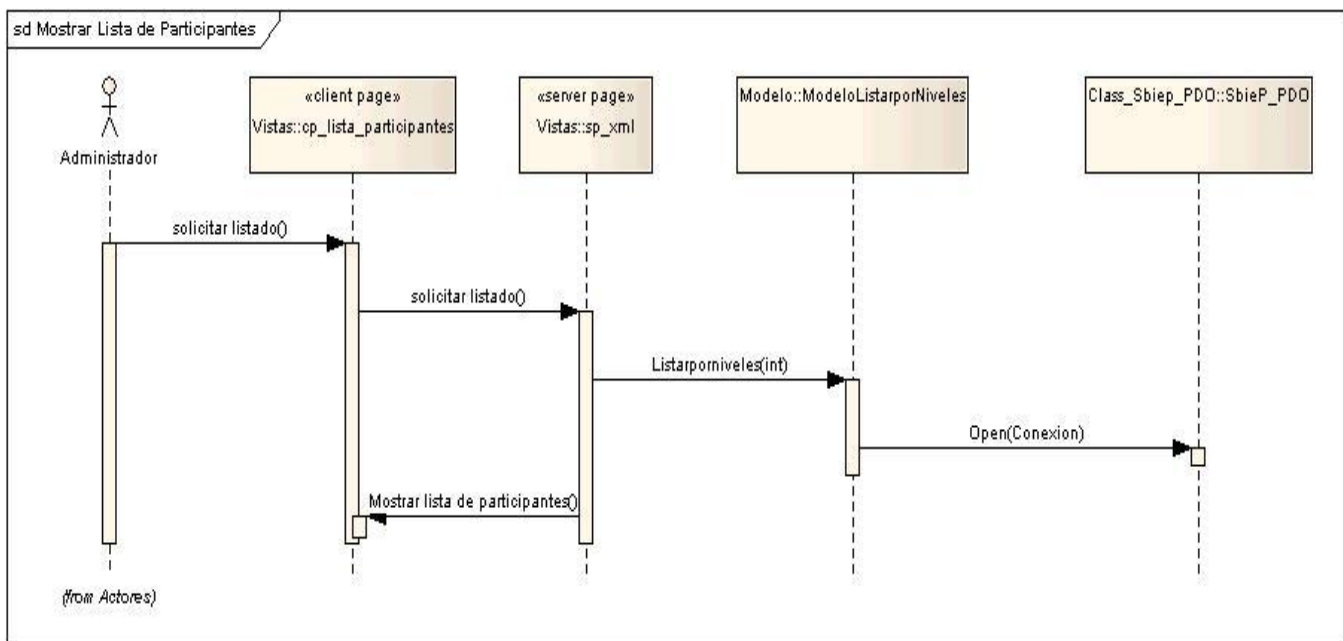


Fig. 37 Diagrama de secuencia del caso de uso Mostrar lista de participantes.

Glosario de Términos

Apache: servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.).

Base de datos: conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Caso de uso: Descripción de un conjunto de secuencia de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

CCS: hojas de estilo en cascada (**Cascading Style Sheets**,) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

CMS (Content Management System): sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de Webs.

Diagrama: Presentación gráfica de un conjunto de elementos y sus relaciones.

Hardware: Conjuntos de componentes que integran la parte material de una computadora. Componentes físicos de una tecnología.

HTML: Siglas de **HyperText Markup Language** (*Lenguaje de Marcas de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas web.

HTTP (HyperText Transfer Protocol): protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML).

Interfaz: Término informático que permite la circulación correcta y sencilla de información entre varias aplicaciones y entre el propio programa y el mono usuario.

Metodología: es la parte del proceso de investigación que sigue a la propedéutica y permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Es pues, una etapa, una parte del proceso.

Modelado del negocio: No es más que comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema; comprender los problemas actuales de la organización e identificar las mejoras potenciales; asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y derivar los requisitos del sistema que va a soportar la organización.

MySQL: base de datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

Navegador: es un uso del software que permite a usuarios tener acceso al World Wide Web (WWW).

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

Requisito: Condición o cualidad que debe cumplir alguien o algo.

RUP (Rational Unified Process): 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.

Servicio Web: Colección de protocolos y estándares que se utilizan para intercambiar datos entre aplicaciones.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

UML (Unified Modeling Language): lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

URL (Uniform Resource Locator): dirección de un recurso en la Web. Tiene el formato `protocol://máquina.domini:port/ruta/recurs`. Por ejemplo: `http://www.uoc.edu/dt/20396/index.html` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

XML: Sigla en inglés de *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas, una manera de definir lenguajes para diferentes necesidades.

WYSIWYG (*What You See Is What You Get*): lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.