

# Universidad de las Ciencias Informáticas

## Facultad 3



*Título: Análisis y diseño de una aplicación web para la gestión de información de eventos científicos.*

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

**Autora:** Anaibis Alvarez Morales.

**Tutor:** Ing. Odiel Estrada Molina.

**Co-Tutor:** Ing. Yoansy López Reyes.

Ciudad de La Habana, Cuba.

Junio, 2011

### Datos de contacto

Autora: Anaibis Alvarez Morales.

Correo electrónico: aamorales@estudiantes.uci.cu

Tutor: Ing. Odiel Estrada Molina.

Correo Electrónico: oestrada@uci.cu

Co-Tutor: Ing. Yoansy López Reyes.

Correo Electrónico: yoansylr@uci.cu

# Declaración de autoría

---

## Declaración de autoría

Declaro que soy la única autora del trabajo titulado:

**“Análisis y diseño de una aplicación web para la gestión de información de eventos científicos”.**

Y le otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de junio del año 2011.

---

Anaibis Alvarez Morales

---

Ing. Odiel Estrada Molina

---

Ing. Yoansy López Reyes

### Agradecimientos

*A mis padres por haberme dado siempre su apoyo, el amor y la fuerza para seguir adelante y poder llegar hasta aquí.*

*A toda mi familia en especial mis hermanos Osley y Ada Iris, a mi cuñada Madelín, a ellos que de una forma u otra me han apoyado para que lograra ser Ingeniera.*

*Al tribunal por sus señalamientos oportunos.*

*A la Revolución por ofrecerme la posibilidad de formarme como profesional.*

*A todos mis amigos y compañeros de aula por compartir estos 5 años.*

*A todos, muchas gracias.*

### **Dedicatoria**

*Dedico mi tesis a mis padres, las personas que más amo en mi vida. Ellos que han contribuido a que hoy mi sueño de ser ingeniera se haya hecho realidad, ustedes que han dedicado sus vidas a mi educación, con sus sacrificios han sido mi aliento, mi apoyo y mis ganas de seguir superándome.*

## Resumen

Desde el surgimiento de la Universidad de las Ciencias Informáticas(UCI) vienen desarrollándose diferentes eventos científicos en aras de socializar y fomentar la investigación en la comunidad universitaria, la Dirección de Investigaciones (DI) como entidad rectora en la organización de dichos eventos lleva a cabo todo el proceso de gestión de información de forma manual y el envío de la misma mediante correo electrónico, provocando un engorroso procesamiento de los datos así como retraso en el cronograma previsto para el evento.

El presente trabajo tiene como objetivo realizar el análisis y diseño de una aplicación web, partiendo del estudio de tendencias actuales y la necesidad de gestionar eficientemente la información de los eventos científicos que se realizan en la universidad. Durante la investigación se identifican los procesos fundamentales y los responsables de efectuarlos, se generan los artefactos correspondientes a cada flujo de trabajo, obteniéndose finalmente una propuesta de diseño, la cual es validada mediante métricas de diseño.

La propuesta contribuirá a una posterior implementación de la aplicación, permitiendo informatizar los procesos de gestión de información de los eventos científicos desarrollados en la UCI.

**Palabras claves:** eventos científicos, aplicación web, gestión de información.

## Tabla de contenidos

---

Introducción .....	11
Capítulo 1. Fundamentación teórica .....	15
1.1 Introducción.....	15
1.2 Gestión de la información.....	15
1.2.1 Gestión de la información de eventos científicos.....	15
1.3 Aplicaciones web.....	16
1.3.1 Aplicaciones web relacionadas con la gestión de información de eventos científicos .	16
1.4 Metodologías de desarrollo de software.....	18
1.4.1 El Proceso Unificado del Rational (RUP).....	19
1.4.2 Extreme Programing (XP).....	20
1.4.3 Metodología seleccionada para el desarrollo del trabajo .....	20
1.5 Lenguaje de modelado .....	21
1.6 Herramientas CASE .....	22
1.6.1 Visual Paradigm for UML .....	22
1.6.2 Rational Rose .....	23
1.6.3 Herramienta CASE seleccionada para el modelado.....	23
1.7 Arquitectura Cliente-Servidor .....	24
1.8 Tecnologías del lado del servidor.....	25
1.8.1 Servidor Web Apache Tomcat .....	25
1.8.2 Sistemas Gestores de Base de Datos .....	25
1.8.3 Gestor de Bases de Datos seleccionado .....	28
1.8.4 Lenguajes de programación .....	28
1.8.5 Lenguaje de programación seleccionado .....	29
1.9 Tecnología del lado del cliente.....	30
Javascript .....	30
1.10 Entornos de desarrollo integrado.....	30
1.10.1 NetBeans IDE .....	30
1.10.3 Eclipse .....	31
1.10.4 Propuesta del IDE.....	31
1.11 Framework de desarrollo.....	31

## Tabla de contenidos

---

1.11.1 Spring .....	31
1.11.2 Hibernate .....	32
1.12 Conclusiones.....	32
Capítulo 2. Modelado de negocio y requerimientos .....	33
2.1 Introducción.....	33
2.2 Descripción de los procesos de negocio .....	33
2.3 Modelo de negocio .....	34
2.3.1 Actores del negocio .....	34
2.3.2 Trabajadores del negocio .....	35
2.3.3 Diagrama de casos de uso del negocio.....	36
2.3.4 Reglas del negocio .....	37
2.3.5 Descripción de los casos de uso del negocio .....	37
2.3.6 Diagrama de actividades .....	40
2.3.7 Modelo de objetos del negocio .....	41
2.4 Especificación de los requisitos de software .....	41
2.4.1 Requerimientos funcionales.....	42
2.4.2 Requisitos no funcionales.....	44
2.5 Modelo de casos de uso del sistema .....	47
2.5.1 Actores del sistema .....	47
2.5.2 Casos de uso del sistema (CUS).....	47
Capítulo 3. Análisis y diseño del sistema.....	63
3.1 Introducción.....	63
3.2 Modelo de análisis.....	63
3.2.1 Diagramas de clases del análisis.....	63
3.3 Modelo de diseño .....	65
3.3.2 Diagramas de interacción. Diagramas de secuencia.....	67
3.3.3 Diseño de la base de datos .....	69
3.4 Patrones empleados en el diseño.....	71
3.4.1 Patrón de arquitectura .....	71
3.4.2 Patrones de diseño.....	72



## Tabla de contenidos

---

Capítulo 4. Validación de la solución propuesta .....	74
4.1 Introducción.....	74
4.2 Métricas utilizadas para validar el modelo de diseño propuesto .....	74
4.2.1 Tamaño de clase (TC) .....	74
4.2.2 Métrica Relaciones entre Clases (RC).....	77
4.3 Conclusiones.....	80
Conclusiones generales.....	81
Recomendaciones .....	82
Bibliografía.....	83
Glosario de Términos.....	86

### Índice de tablas

TABLA 1. ACTOR DEL NEGOCIO .....	34
TABLA 2. TRABAJADORES DEL NEGOCIO .....	36
TABLA 4. DESCRIPCIÓN DEL CASO DE USO: TRAMITAR EVENTO .....	40
TABLA 5. ACTORES DEL SISTEMA .....	47
TABLA 6. DESCRIPCIÓN DEL CASO DE USO: GESTIONAR CONVOCATORIA.....	53
TABLA 7. DESCRIPCIÓN DEL CASO DE USO: GESTIONAR EVENTO.....	62
TABLA 8. MÉTRICA TAMAÑO DE CLASE (TC).....	75
TABLA 9. RANGO DE VALORES DE PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD RELACIONADOS CON LA MÉTRICA TC.....	75
TABLA 10. MÉTRICA RELACIÓN ENTRE CLASES (RC).....	77
TABLA 11. RANGO DE VALORES PARA LA EVALUACIÓN TÉCNICA DE LOS ATRIBUTOS DE CALIDAD RELACIONADOS CON LA MÉTRICA RC .....	78

### Índice de figuras

FIGURA 1. DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	37
FIGURA 2. DIAGRAMA DE ACTIVIDADES DEL CASO DE USO: TRAMITAR EVENTO .....	40
FIGURA 3. DIAGRAMA DE CLASES DEL MODELO DE OBJETOS DEL CASO DE USO: TRAMITAR EVENTO. ....	41
FIGURA 4. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	48
FIGURA 5. DIAGRAMA DE CLASES DEL ANÁLISIS CU: GESTIONAR CONVOCATORIA.....	64
FIGURA 6. DIAGRAMA DE CLASES DEL ANÁLISIS CU: GESTIONAR EVENTO .....	64
FIGURA 7. DIAGRAMA DE CLASES DEL DISEÑO WEB: GESTIONAR CONVOCATORIA. ....	66
FIGURA 8 DIAGRAMA DE CLASES DEL DISEÑO WEB: GESTIONAR EVENTO.....	67
FIGURA 9. GESTIONAR EVENTO: ESCENARIO ADICIONAR EVENTO. ....	68
FIGURA 10 GESTIONAR EVENTO: ESCENARIO MODIFICAR EVENTO. ....	68

## Tabla de contenidos

---

FIGURA 11 GESTIONAR EVENTO: ESCENARIO ELIMINAR EVENTO.....	69
FIGURA 12. DIAGRAMA DE CLASES PERSISTENTES DE LA BASE DE DATOS.....	70
FIGURA 13. DIAGRAMA ENTIDAD-RELACIÓN DE LA BASE DE DATOS .....	71
FIGURA 14. REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO, AGRUPADOS EN LOS INTERVALOS DEFINIDOS.....	75
FIGURA 15. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO RESPONSABILIDAD.....	76
FIGURA 16. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO COMPLEJIDAD DE IMPLEMENTACIÓN.....	76
FIGURA 17. FIGURA REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TC EN EL ATRIBUTO REUTILIZACIÓN.....	76
FIGURA 18. REPRESENTACIÓN EN % DE LOS RESULTADOS OBTENIDOS EN EL INSTRUMENTO, AGRUPADOS EN LOS INTERVALOS DEFINIDOS.....	78
FIGURA 19. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO ACOPLAMIENTO.....	79
FIGURA 20. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO COMPLEJIDAD DE MANTENIMIENTO.....	79
FIGURA 21. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO CANTIDAD DE PRUEBAS.....	79
FIGURA 22. REPRESENTACIÓN DE LA INCIDENCIA DE LOS RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC EN EL ATRIBUTO REUTILIZACIÓN.....	80

## Introducción

A medida que la sociedad se desarrolla, los adelantos científico-técnicos han ido incrementándose en función de generar y compartir cultura científica, dando como resultado adelantos científicos meritorios que han impulsado a la humanidad a interesarse por diversas ramas del conocimiento humano como la biología, la química, la física, la matemática y las ciencias de la información. Es evidente que las ciencias por separado no logran su expansión y difusión, para ello se organizan diferentes eventos donde se evidencia la calidad y el nivel científico alcanzado por investigadores, profesionales, académicos y estudiantes en general de diversas ramas de la ciencia, con el objetivo de compartir los resultados obtenidos en sus proyectos de investigación-desarrollo(I+D).

El avance científico-técnico en Cuba ha permitido que todos los años se convoquen diferentes encuentros y presentaciones, tanto nacionales como internacionales, que elevan el nivel profesional como centro de referencia en algunas ramas; de ahí que existan actualmente muchas organizaciones cubanas inmersas en estas actividades, manejando grandes cantidades de información relacionada con los eventos que desarrollan.

En la UCI los estudiantes se forman desde la docencia, la producción, la investigación y la extensión universitaria, todo ello favorece que frecuentemente se convoquen diversos eventos científicos tales como Jornada Científica Estudiantil (JCE), Fórum de Ciencia y Técnica (FCT) y Conferencia Científica Uciencia, realizados en aras de socializar los resultados de las investigaciones realizadas por parte de estudiantes, trabajadores y profesionales en general.

Actualmente en la UCI todo el proceso de gestión de los eventos científicos se realiza de forma manual, la información de los mismos es registrada en papel y en el mejor de los casos en documentos digitales, lo que trae consigo un difícil y engorroso procesamiento de los datos, debido a la abundante información que se maneja. Cuando se convoca a un evento la información transita por diferentes niveles mediante correo electrónico, lo que puede provocar pérdida o retraso de la información, afectando de manera directa el cronograma del evento. Al no existir un sistema informático público en la red universitaria, capaz de

## Introducción

---

gestionar estos procesos, surge la necesidad de una aplicación que permita la gestión de información de los eventos científicos desarrollados en la universidad.

Debido a la situación problemática antes mencionada se define como **problema a resolver**: ¿Cómo contribuir al desarrollo de un sistema que permita la gestión de información de los eventos científicos de la UCI?

El **objeto de estudio** de la investigación que se establece es la gestión de información de los eventos científicos, especificándose como **campo de acción** la gestión de información de los eventos científicos de la UCI, teniendo en cuenta el problema anterior se traza como **objetivo general**: realizar el análisis y diseño de una aplicación web que permita la gestión de información de los eventos científicos de la UCI.

La presente investigación está sustentada por la siguiente **idea a defender**: con el análisis y diseño de una aplicación web para la gestión de información de los eventos científicos, se obtendrá como resultado la documentación técnica necesaria para facilitar la futura implementación de la misma.

Las **tareas de investigación** que se proponen son:

1. Elaboración del marco teórico-conceptual de la investigación.
2. Argumentar el uso de las herramientas, tecnologías, metodologías de desarrollo, lenguajes de modelado, así como la propuesta de herramientas que se emplearán para una posterior implementación.
3. Realización del modelo de negocio de la aplicación web.
4. Identificación de los requisitos funcionales y no funcionales de la aplicación web.
5. Realización del análisis y diseño de la aplicación web.
6. Validación del diseño de la aplicación web, mediante el empleo de métricas de diseño orientado a objetos.

Para el desarrollo de la investigación se emplean diferentes métodos científicos que se describen a continuación:

## Métodos Teóricos

Histórico-Lógico: permite consultar la bibliografía necesaria para dar cumplimiento a las tareas de la investigación y resumir los aspectos fundamentales de cada una de ellas.

Analítico-Sintético: permite analizar las soluciones existentes que gestionan la información de los eventos científicos, tanto a nivel nacional como internacional.

Modelación: permite la obtención de todos los artefactos de la Ingeniería del Software necesarios que permitan guiar a la posterior implementación de la aplicación.

## Métodos Empíricos

Observación: permitió visualizar lo que ocurre en el entorno del problema de la investigación.

Entrevistas: se utiliza para obtener a través del personal que labora en la DI la información referente a como se lleva a cabo el proceso de trámite de eventos, garantizando así que la solución propuesta cumpla las necesidades existentes.

Para lograr una correcta comprensión del trabajo, el presente documento consta de 4 capítulos, donde la información está distribuida de la siguiente forma:

**Capítulo 1. Fundamentación teórica:** Relaciona conceptos fundamentales asociados al objeto de estudio del trabajo, se realiza una investigación de diferentes aplicaciones web que manejan la gestión de información de los eventos científicos, además se describen y seleccionan las tecnologías, herramientas, metodologías de desarrollo, lenguajes de modelado más factibles para desarrollar el Análisis y Diseño de la aplicación, se realiza la propuesta del lenguaje de implementación y de diferentes herramientas para una futura implementación.

**Capítulo 2. Modelado de negocio y requerimientos:** En este capítulo se realiza el modelado de negocio, además se especifican los requisitos funcionales y no funcionales que guían el desarrollo del trabajo y se concluye con el modelado del sistema.

# Introducción

---

**Capítulo 3. Análisis y diseño del sistema:** Se expone la construcción de la solución propuesta en términos de la realización de los artefactos que forman parte del flujo de trabajo de Análisis y Diseño a partir de los diagramas de clases del análisis y diseño, diagramas de secuencia del diseño y modelo de datos.

**Capítulo 4. Validación de la solución propuesta:** El capítulo contiene lo referente a la validación de la propuesta de solución, mediante el empleo de métricas de software.

Para finalizar se presentan las conclusiones, las recomendaciones, la bibliografía, un glosario de términos y el conjunto de anexos para un mejor entendimiento de lo expuesto en todo el trabajo.

## Capítulo 1. Fundamentación teórica

### 1.1 Introducción

El capítulo relaciona conceptos fundamentales asociados al objeto de estudio del trabajo. Se realiza una investigación de diferentes aplicaciones web que manejan la gestión de información de los eventos científicos, permitiendo analizar y sintetizar todas las características comunes que estas presentan. Se exponen las principales metodologías de desarrollo de software, herramientas, tecnologías y el lenguaje de modelado, definiéndose las que serán utilizadas para todo el proceso de modelado de la aplicación, además se realiza una propuesta del lenguaje de programación y herramientas de implementación más factibles para una futura implementación de la solución propuesta.

### 1.2 Gestión de la información

Actualmente las organizaciones que tienen bien definidas sus metas y que buscan convertirse en organizaciones de “excelencia”, o lo que es lo mismo, alcanzar el éxito en el cumplimiento de su misión y objetivos, han concebido la necesidad imperiosa de gestionar eficazmente un recurso muy importante: la información.

El proceso de gestión de información está encaminado a la utilización eficaz de los datos, la información y el conocimiento en la investigación, la programación de las investigaciones, la política científica y otras políticas en los contextos más diversos. (1)

#### 1.2.1 Gestión de la información de eventos científicos

Un evento es una actividad que requiere ser planeada, ejecutada, controlada, encaminada a lograr objetivos propuestos, bien sea de índole empresarial, asociativa o particular. (2)

La realización de eventos científicos son actividades que realizan diversas organizaciones y que posibilita la participación de investigadores, estudiantes, profesionales y de forma general la comunidad científica internacional; brindando aportes en diferentes temáticas, con el objetivo de promover los adelantos

científicos, las nuevas tecnologías, intercambiar experiencias, compartir los resultados de proyectos de investigación, de adquirir nuevos conocimientos o afianzar los que se poseen. Durante la realización de eventos científicos se gestiona información enfocada a diversos temas de investigación, siendo de vital importancia en las organizaciones que todo el proceso de gestión de la misma sea eficiente, controlado y encaminado a impulsar el avance científico en todos los sectores de la sociedad.

### 1.3 Aplicaciones web

Las aplicaciones web son aplicaciones que los usuarios pueden utilizar y que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP, estas aplicaciones se distinguen según su función en:

- ✓ Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación, esta una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.
- ✓ Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarse en la máquina del usuario. Son las aplicaciones tipo Javascript, el servidor proporciona el código de las aplicaciones al cliente y este mediante el navegador, las ejecuta. (3)

#### 1.3.1 Aplicaciones web relacionadas con la gestión de información de eventos científicos

En la actualidad existen diferentes aplicaciones web que manejan información relacionada con los eventos científicos en el ámbito nacional e internacional. A continuación se realiza un análisis de aplicaciones web similares a la que se quiere desarrollar en el presente trabajo de diploma.

Como resultado de la adopción e implantación de modelos de buenas prácticas como CMMI, PSP, TSP, ITIL y otros estándares ISO en muchas empresas del mundo se desarrolla SEPGLA (Grupo de Ingeniería de Procesos de Software para Latinoamérica), este es un evento que se desarrolla en Medellín, Colombia, dirigido a profesionales implicados en actividades de mejora sistemática de procesos y tecnologías en organizaciones donde el software es un elemento clave para la consecución del éxito empresarial. Constituye uno de los eventos más relevantes que se realizan en el mercado Latinoamericano en el ámbito de la Ingeniería de Software y prácticas de gestión relacionadas con el desarrollo de software y



## Capítulo 1. Fundamentación teórica

---

sistemas, permite a los participantes compartir y aprender de las experiencias y casos de éxito en implementación de mejora de procesos. (4)

El evento posee una aplicación web para gestionar toda la información del mismo, esta ofrece diferentes funcionalidades como: patrocinador, programa y comité científico del evento, propiciando información relevante a los usuarios interesados en participar en el evento.

En el ámbito nacional el Centro Coordinador para la Formación y el Desarrollo del Capital Humano (FORDES), organiza eventos como acción formativa, resultado de la concepción estratégica y su misión de difundir lo más posible el conocimiento acumulado en las esferas que son objeto central del trabajo del Sistema MIC (Ministerio de la Informática y las Comunicaciones). FORDES se da la tarea de crear encuentros que propician espacios de reflexión y debate para el análisis de diversos temas que resultan de interés en temáticas importantes para el sector en general, fomentando el aprendizaje de sus participantes. Surge así el evento “Semana Tecnológica”, el cual se realiza anualmente en La Habana con el objetivo de promover el intercambio de experiencias, el debate, el análisis reflexivo, así como las proyecciones futuras sobre las temáticas tratadas. (5)

El evento Semana Tecnológica cuenta con una aplicación web que lleva a cabo todo el seguimiento del evento, gestionando el desarrollo del mismo mediante funcionalidades como: convocatoria, temáticas, inscripción de participación, comité organizador, comité científico, auspiciador, evento virtual y envío de trabajos.

También en el ámbito nacional el MIC desarrolla la Convención y Feria Internacional Informática Habana, la cual sesiona en el Palacio de Convenciones de La Habana y en el recinto ferial Pabexpo, el evento tiene como objetivo promover los adelantos científicos, las nuevas tecnologías y novedades del sector, permitiendo la adquisición de conocimientos, el intercambio de experiencias y que se afiance la amistad entre los pueblos. (6)

El evento tiene una aplicación web que lleva a cabo todo el seguimiento del mismo, posee funcionalidades como: convocatoria, noticias, memorias del evento, comité organizador, comité científico, programa del evento, salón virtual, patrocinadores, auspiciadores y personalidades presentes durante la realización del evento.

## Capítulo 1. Fundamentación teórica

---

En la UCI se identificaron dos aplicaciones web realizadas por estudiantes en sus tesis de grado, en los cursos académicos 2008-2009 y 2009-2010, de las mismas se pudo constatar que en ambas solo se les da un seguimiento a los eventos científicos mediante funcionalidades básicas como: nombre del evento, objetivo, fecha inicio, fecha fin, alcance, tipo de evento y envío de trabajo. Las aplicaciones desarrolladas no permiten conocer toda la información referente a los eventos científicos que se realizan en la Universidad, basado en las necesidades existentes en la DI. (7) (8)

En el presente trabajo de diploma se propone una aplicación web para la UCI que gestione la información de los eventos científicos desarrollados, basado en las necesidades de la DI, mediante la definición de características como: temáticas, comité científico, políticas de envío, noticias del evento, memorias del evento, próximos eventos y programa general del evento, las cuales son relevantes para realizar dicha gestión de forma completa.

### 1.4 Metodologías de desarrollo de software

Todo desarrollo de software es riesgoso y difícil de controlar, de ahí la importancia de contar con una metodología que indique paso a paso las actividades que deben desarrollarse, las personas involucradas en la realización de las mismas, así como el papel que desempeñan estas en aras de lograr un proceso de software exitoso. Lo más importante antes de elegir la metodología que se usará para la implementación de un software es determinar el alcance que tendrá el mismo y seguidamente analizar cuál sería la más factible para su desarrollo.

**Las metodologías de desarrollo de software se pueden clasificar en dos grandes grupos:**

✓ **Metodologías Ligeras/Ágiles:** *metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software*, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. (9)

**Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:**

✓ XP (Extreme Programming).

- ✓ Scrum.
- ✓ Crystal Clear.
- ✓ DSDM (Dynamic Systems Development Method).
- ✓ FDD (Feature Driven Development).
- ✓ ASD (Adaptive Software Development).
- ✓ XBreed.
- ✓ Extreme Modeling.
- ✓ **Metodologías Pesadas/Tradicionales:** *metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán.*

**Entre las metodologías tradicionales o pesadas podemos citar:**

- ✓ RUP (Rational Unified Procces).
- ✓ MSF (Microsoft Solution Framework).
- ✓ Win-Win Spiral Model.
- ✓ Iconix. (10)

A continuación se describen dos de las metodologías más utilizadas en el desarrollo de software:

### 1.4.1 Rational Unified Process RUP (Proceso Unificado de Rational)

RUP define entre sus principales elementos a los trabajadores, actividades, artefactos y flujo de actividades, los cuales contribuyen a un proceso de desarrollo de software exitoso. El ciclo de vida de RUP se caracteriza por estar dirigido por casos de uso, los mismos reflejan lo que los usuarios futuros necesitan y desean, además está centrado en la arquitectura, la cual muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, otra característica de RUP es que es iterativo e incremental, debido a que propone que cada fase se desarrolle en iteraciones, las cuales hacen referencia a pasos en el flujo de trabajo, mientras que los incrementos contribuyen al crecimiento del producto.

RUP divide el proceso en cuatro fases: fase de concepción o inicio, fase de elaboración, fase de construcción y fase de transición, dentro de las cuales se realizan varias iteraciones en número variable, consta además de nueve flujos de trabajo: Modelo del Negocio, Requerimiento, Análisis y Diseño, Implementación, Prueba, Instalación, Administración del Proyecto, Administración de Configuración y Cambios, Ambiente, estos flujos serán visitados una y otra vez durante todo el proceso de desarrollo de software y de los cuales se obtienen diferentes artefactos (11)

### 1.4.2 Extreme Programming (XP)

XP se clasifica como Metodología Ágil y es una de las más exitosas en la actualidad, utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega es en tiempo récord. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (12)

El ciclo de vida ideal de XP está compuesto por seis fases que permiten todo el proceso de desarrollo en un tiempo record, estas fases son las siguientes:

- ✓ Exploración.
- ✓ Planificación de la entrega.
- ✓ Iteraciones.
- ✓ Producción.
- ✓ Mantenimiento.
- ✓ Muerte del proyecto. (13)

### 1.4.3 Metodología seleccionada para el desarrollo del trabajo

Para el desarrollo de este trabajo se decidió utilizar una de las metodologías pesadas debido a que las mismas están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo y el proceso es mucho más controlado, con diferentes políticas y normas, mientras que las ágiles están basadas en heurísticas que provienen de prácticas de producción de código y procesos menos controlados y con pocos principios.

La metodología seleccionada fue RUP, ya que es una de las más generales que existen en la actualidad, debido a que está pensada para adaptarse a cualquier proyecto, principalmente con grupos de producción grandes, donde el ciclo de vida está dirigido por casos de uso que reflejan lo que los usuarios requieren y desean, capturándose en el modelado del negocio y representándose a través de los requerimientos, los casos de uso guían a partir de aquí, todo el proceso de desarrollo, donde la arquitectura del software es fundamental y muestra una visión común del sistema, en la que el equipo de proyecto y los usuarios deben estar de acuerdo. En RUP el cliente no forma parte del equipo de desarrollo, este solo interactúa con el equipo mediante reuniones y el establecimiento de un contrato prefijado, a diferencia por ejemplo de XP donde el cliente se convierte en un miembro más del equipo.

### 1.5 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Este lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar, incluye además conceptos semánticos, notación, y principios generales. UML no pretende ser un método de desarrollo completo, no incluye un proceso de desarrollo paso a paso, pero si incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

UML se puede usar para modelar sistemas de software, sistemas de hardware y organizaciones del mundo real.

UML ofrece diferentes diagramas, los cuales se clasifican en:

#### **Diagramas de estructura estática:**

- ✓ Diagrama de clases.
- ✓ Diagrama de objetos.
- ✓ Diagrama de casos de uso.

### Diagramas de comportamiento

- ✓ Diagramas de interacción (secuencia y colaboración).
- ✓ Diagrama de actividad.
- ✓ Diagrama de estados.

### Diagramas de implementación

- ✓ Diagrama de componentes.
- ✓ Diagrama de despliegue. (14)

### 1.6 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering), traducido al español Ingeniería de Software Asistida por Ordenador, son un conjunto de métodos, utilidades y técnicas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un software, facilitando el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

A continuación se identifican dos de las herramientas CASE más empleadas a nivel internacional

#### 1.6.1 Visual Paradigm para UML

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (15) Entre sus principales características están:

- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Entorno de creación de diagramas para UML 2.0.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Dispone de múltiples versiones dependiendo de la necesidad.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Posee un modelo y un código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad en múltiples plataformas.

**Visual Paradigm for UML permite la creación de los siguientes diagramas:**

- ✓ Componentes.
- ✓ Despliegue.
- ✓ Secuencia.
- ✓ Casos de uso.
- ✓ Clase.
- ✓ Actividad.
- ✓ Estado. (16)

### 1.6.2 Rational Rose

Rational Rose es una herramienta CASE que da soporte al modelado visual con UML, ofreciendo distintas perspectivas del sistema, da soporte a RUP para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la etapa de Pruebas. Esta herramienta ofrece un diseño dirigido por modelos que redundan en una mayor productividad de los desarrolladores, admitiendo el lenguaje de modelado UML y Técnicas de Modelado de Objetos (OMT), esta favorece:

- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Disponibilidad en múltiples plataformas.
- ✓ Capacidades de ingeniería inversa.
- ✓ Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo. (17)

### 1.6.3 Herramienta CASE seleccionada para el modelado

Después de haber realizado un análisis de las herramientas CASE antes mencionadas se selecciona para efectuar el modelado de este trabajo el Visual Paradigm para UML 6.4, debido a que el mismo facilita todo el proceso de modelado de la aplicación web, permitiendo crear los diagramas correspondientes durante todo el ciclo de vida de desarrollo de software, es una de licencias libre, multiplataforma, orientada a la

creación de diseños usando el paradigma de programación orientada a objetos, además de proveer soporte para la generación de código.

### 1.7 Arquitectura Cliente-Servidor

La arquitectura cliente- servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario (18)

**Las principales ventajas del modelo Cliente/Servidor son las siguientes:**

- ✓ **Interoperabilidad:** los componentes clave (cliente, servidor y red) trabajan juntos.
- ✓ **Flexibilidad:** la nueva tecnología puede incorporarse al sistema.
- ✓ **Escalabilidad:** cualquiera de los elementos del sistema puede reemplazarse cuando es necesario, sin impactar sobre otros elementos. Si la base de datos crece, las computadoras cliente no tienen que equiparse con memoria o discos adicionales. Esos cambios afectan solo a la computadora en la que se ejecuta la base de datos.
- ✓ **Usabilidad:** mayor facilidad de uso para el usuario.
- ✓ **Integridad de los datos:** entidades, dominios, e integridad referencial son mantenidas en el servidor de la base de datos.
- ✓ **Accesibilidad:** los datos pueden ser accedidos desde múltiples clientes.
- ✓ **Rendimiento:** se puede optimizar el rendimiento por hardware y procesos.
- ✓ **Seguridad:** la seguridad de los datos está centralizada en el servidor (19).



## 1.8 Tecnologías del lado del servidor

### 1.8.1 Servidor Web Apache Tomcat

Se propone la utilización de ApacheTomcat 6.0.14 como servidor web de aplicaciones, debido a que el mismo fue escrito en Java y es el lenguaje de programación propuesto para implementar la aplicación, funciona en cualquier sistema operativo que tenga instalado la Máquina Virtual de Java, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad, es un software de código abierto, funciona como un contenedor de de servlets (objetos que corren dentro del contexto de un servidor de aplicaciones y extienden su funcionalidad), además implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystem. (20)

### 1.8.2 Sistemas Gestores de Base de Datos (SGBD)

Los Sistemas de Gestión de Bases de Datos son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y como proporcionar un acceso controlado a la misma. Un SGBD es la aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos.

#### Los objetivos de un SGBD son:

- ✓ Definir la base de datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- ✓ Separar la descripción y manipulación de los datos, permitiendo un mayor entendimiento de los objetos, además de flexibilidad de consulta y actualización de los datos.
- ✓ Permitir la inserción, eliminación, actualización y consulta de los datos mediante el Lenguaje de Manejo de Datos, lo que permite resolver el problema que presentan los sistemas de archivos, donde hay que trabajar con un conjunto fijo de consultas o la necesidad de tener muchos programas de aplicaciones.
- ✓ Proporcionar acceso controlado a la base de datos:
  - Seguridad: los usuarios no autorizados no pueden acceder a la base de datos.
  - Integridad: mantiene la integridad y consistencia de la base de datos.

## Capítulo 1. Fundamentación teórica

---

- Control de Recurrencia: permite el acceso compartido a la base de datos.
  - Control de Recuperación: restablece la base de datos después de producirse un fallo de software o hardware.
  - Diccionario de Datos o Catálogo: contiene la descripción de los datos de la base de datos y es accesible por el usuario.
- ✓ Gestionar la estructura física de los datos y su almacenamiento, proporcionando eficiencia en las operaciones de la base de datos y el acceso al medio de almacenamiento.
  - ✓ Permitir una fácil administración de los datos.
  - ✓ Proporcionar un mecanismo de vistas, que permita a cada usuario tener su propia vista o visión de la base de datos.
  - ✓ Eliminar la redundancia de datos, establecer una mínima duplicidad en los datos y minimizar el espacio en disco utilizado.
  - ✓ Proveer interfaces procedimentales y no procedimentales, permitiendo la manipulación por usuarios interactivos y programadores.
  - ✓ Independizar la estructura de la organización lógica de los datos.
  - ✓ Independizar la descripción lógica de la base de datos y las descripciones particulares de los diferentes puntos de vistas de los usuarios. (21)

**A continuación se identifican algunos de los SGDB más usados a nivel internacional.**

### **PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS), debido a que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. Utiliza un modelo cliente/servidor y el lenguaje SQL para llevar a cabo las búsquedas de información, para garantizar la estabilidad del sistema usa multiprocesos, en vez de multihilos. Este sistema de gestión de bases de datos objeto-relacional es de código abierto. (22)

## Capítulo 1. Fundamentación teórica

---

Se caracteriza por ser un sistema estable, de alto rendimiento y gran flexibilidad, ya que funciona en la mayoría de los sistemas Unix, puede ser integrado al ambiente Windows, permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes, además posibilita desarrollar o migrar aplicaciones desde Access, Visual Basic, Foxpro, Visual Foxpro, C/C++ Visual C/C++, Delphi, entre otros, para que utilicen a PostgreSQL como servidor de base de datos. (23)

### MySQL

MySQL es un sistema de gestión de bases de datos relacional, propietario, su diseño multihilo le permite soportar una gran carga de forma muy eficiente y rápida. Existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además resulta fácil su instalación y configuración.

**Las principales características de este gestor de bases de datos son las siguientes:**

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API' s en gran cantidad de lenguajes (C, C++, Java, PHP).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos. (24)

### Oracle

Oracle es una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales.

El sistema gestor de base de datos Oracle puede ejecutarse en todas las plataformas, soporta todas las funciones que se esperan de un servidor "serio", utiliza un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente, permite además el uso de particiones para la mejora de la eficiencia de replicación e incluso ciertas versiones admiten la

administración de bases de datos distribuidas. El software del servidor puede ejecutarse en multitud de sistemas operativos y su soporte es bastante aceptable. Este sistema ha comenzado a evolucionar con respecto a sus objetos, añadiendo tipos de clases, referencias, tablas anidadas, matrices y otras estructuras de datos complejas. (25)

### 1.8.3 Gestor de Bases de Datos seleccionado

Al realizar un análisis de los SGBD antes identificados se propone utilizar PostgreSQL 8.4, el mismo es confiable, estable y libre, siendo manejado por la licencia de Distribución de Software Berkeley, además es un gestor multiplataforma, diseñado para ambientes de alto volumen, por tanto sería conveniente utilizarlo como SGBD.

### 1.8.4 Lenguajes de programación

Los lenguajes de programación son herramientas que posibilitan la creación de programas y software, están constituidos por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos con sentido único y una regla principal.

#### PHP

PHP es un lenguaje de programación interpretado de propósito general. Es ampliamente utilizado en el desarrollo web y puede ser embebido en páginas HTML.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- ✓ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MSSQL Server, SybasemSQL e Informix.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF y analizar código XML.
- ✓ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.

- ✓ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- ✓ Con PHP se puede hacer cualquier cosa que se puede realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. (26)

### Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (27)

Hasta la fecha, la plataforma Java ha atraído a más de 6.5 millones de desarrolladores de todo el mundo y está presente en un gran número de dispositivos, redes y equipos. Se caracteriza por su portabilidad y seguridad lo cual la ha convertido en la tecnología ideal para su aplicación a redes. (28)

El lenguaje para la programación en Java es un lenguaje orientado a objeto y de una plataforma independiente. Los programas en Java generalmente son compilados y luego interpretados por una máquina virtual. Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda "escribir el programa una vez, y correrlo en cualquier lado". También existen compiladores nativos de Java, tanto comerciales como libres.

#### 1.8.5 Lenguaje de programación seleccionado

Partiendo de los lenguajes de programación antes abordados se propone que para una posterior implementación se utilice el lenguaje Java, debido a que el mismo es un lenguaje de programación multiplataforma, implementa en diferentes niveles un sistema seguro de validación, logrando la seguridad de acceso a las aplicaciones web, además posee un amplio conjunto de bibliotecas estándar para trabajar

con colecciones; estructuras de datos; ficheros; acceso a base de datos e interfaces graficas, simplifica varios aspectos a la hora de programar porque realiza la gestión de memoria de forma automática, provee una sintaxis simplificada, es factible para el desarrollo de páginas web dinámicas y además posibilita la creación de programas modulares y códigos reutilizables.

### 1.9 Tecnología del lado del cliente

#### Javascript

Se propone la utilización de Javascript como aplicación del lado del cliente, debido a que es un lenguaje basado en objetos, utilizado para acceder a objetos en aplicaciones. Este lenguaje se utiliza principalmente integrado en un navegador Web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos del Documento (DOM). (29)

### 1.10 Entornos de desarrollo integrado

#### 1.10.1 NetBeans IDE

El Entorno de Desarrollo Integrado Netbeans es una herramienta libre y gratuita sin restricciones de uso, licenciada bajo la CDDL(Common Development and Distribution License), que en español sería Desarrollo Común y Licencia de Distribución, la herramienta es utilizada por los programadores para escribir, compilar, depurar y ejecutar programas, permite el desarrollo de aplicaciones web. Está escrito en Java pero puede soportar cualquier otro lenguaje de programación, esta herramienta es modular debido a que todas las funciones que posee están provistas de módulos como el de soporte y edición. (30)

**Este entorno de desarrollo integrado se caracteriza por:**

- ✓ Editor de código sensible al contenido, con soporte para autocompletar el código, coloreado de etiquetas, autotabulación y uso de abreviaturas para varios lenguajes de programación.
- ✓ Posibilidad de utilizar otras versiones de compiladores depuradores.

- ✓ Creación visual de componentes gráficos.
- ✓ Soporte para JSP, XML, RMI, CORBA, JINI, JDBC y tecnologías Servlet.
- ✓ Soporte para Java, C, C++, XML y lenguajes HTML.
- ✓ Herramientas con asistentes para facilitar la escritura de código. (31)

### 1.10.3 Eclipse

Eclipse es un Entorno de Desarrollo Integrado (IDE) de código abierto y multiplataforma, que permite la programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java, el mismo dispone de un editor de texto con resaltado de sintaxis donde se puede ver el contenido del fichero en el que se está trabajando, la compilación se realiza en tiempo real, además incluye las librerías JUnit lo que le permite integrarse para la automatización de las pruebas unitarias. (32)

### 1.10.4 Propuesta del IDE

A partir del análisis de los IDE que se identificaron, se propone que se utilice en una futura implementación de la aplicación el NetBeans IDE 6.9; debido a que sus características de flexibilidad entre plataformas y la capacidad de administrar la complejidad, ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos de la aplicación. NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web y control de versiones.

## 1.11 Framework de desarrollo

Un framework es una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación, permite acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. (33)

A continuación se realiza una propuesta de dos framework de desarrollo que pueden ser empleados:

### 1.11.1 Spring

Se propone la utilización de Spring Framework debido a que es un framework de código abierto para el desarrollo de aplicaciones en la plataforma Java. Fue desarrollado por Rod Johnson, quien lo describió por primera vez en su libro "ExpertOne-on-One Java EE Design and Development". Es el único framework

que interviene en todas las capas arquitectónicas de una aplicación JEE, además está diseñado para facilitar una flexibilidad arquitectónica. (34)

Los principales valores de Spring, según Rod Johnson, se pueden resumir en: No es agresivo, provee un modelo consistente de programación, ayuda a promover la reusabilidad de código, facilita el diseño Orientado a Objetos en aplicaciones JEE, permite la extracción de valores de configuración desde el código java a archivos XML o archivos de propiedades, está diseñado a fin de que las aplicaciones lo usen para que las pruebas sean lo más fácil posible, Spring hace de soluciones existentes un uso más fácil, dentro de una arquitectura consistente. Además de esto, Spring provee en su paquete Spring Web MVC una implementación del patrón MVC(Modelo Vista Controlador), el cual brinda una limpia y clara separación entre las 3 capas arquitectónicas: Acceso a Datos, Negocio y Lógica de Presentación. Esta implementación facilita una serie de clases controladoras, que manejan la lógica de la navegación e interactúan con la capa de Negocio de la aplicación web. (35)

### 1.11.2 Hibernate

Se propone la utilización de Hibernate debido a que es una solución ORM (Object-Relational Mapping) para Java. Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Permite generar las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución, ofrece también un lenguaje de consulta de datos llamado HQL, al mismo tiempo que una API para construir las consultas programáticamente, conocida como "Críterias". (36)

### 1.12 Conclusiones

Una vez finalizado el presente capítulo se han dejado evidenciadas las bases teóricas que sustentarán el proceso de desarrollo de la solución del problema planteado. El estudio de sistemas enfocados a la gestión de información de los eventos científicos, permitió valorar la definición de funcionalidades para la aplicación a desarrollar así como establecer pautas que permitan crear un sistema completo y que gestione de manera eficiente la información de los eventos científicos. Se caracterizaron diferentes



herramientas, tecnologías, lenguajes, tanto para el modelado de la aplicación como para la futura implementación de la misma.

### Capítulo 2. Modelado de negocio y requerimientos

#### 2.1 Introducción

En el presente capítulo se realiza toda la descripción y modelado del negocio luego de haber efectuado un estudio de cómo se llevan a cabo los procesos de tramitación de eventos en la DI de la universidad. Además se realiza la especificación de los requisitos funcionales y no funcionales que guían el desarrollo del sistema, se determinan y describen los actores y casos de uso del sistema; los cuales quedan representados en el Diagrama de Casos de Uso del Sistema.

#### 2.2 Descripción de los procesos de negocio

La propuesta de solución de la presente investigación comprende el desempeño de procesos relacionados con la gestión de información de los eventos científicos en la UCI. La investigación en la universidad es controlada en los diferentes niveles administrativos, los Vicedecanos de Investigación y Postgrado (VDIP) son los encargados del control y organización de toda actividad científica y posgraduada en las facultades, apoyando y manteniendo actualizada a la DI de todo el proceso, siendo estos últimos los que socializan el trabajo científico que lleva a cabo la UCI.

El negocio comprende básicamente un proceso: Tramitar evento, del cual se describe a continuación la secuencia de actividades que se realizan en el mismo.

##### Tramitar evento

El proceso de trámite de los eventos científicos desarrollados en la UCI es realizado por la DI, el Comité Organizador de la Facultad, el Comité Científico de la Facultad y el Comité Científico UCI. Los mismos de acuerdo a su función dentro del proceso, llevan a cabo la recepción y la revisión de los trabajos enviados por los participantes. Los trabajos inicialmente son revisados a nivel de facultad seleccionando los que poseen las condiciones para ser presentados a nivel de universidad; finalmente el Comité Científico UCI

## Capítulo 2. Modelado de negocio y requerimientos

---

realiza una última revisión de los trabajos para valorar su calidad y decidir si los mismos están aptos para su presentación a nivel de universidad.

### 2.3 Modelo de negocio

El modelado de negocio permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la misma en los modelos de casos de uso del negocio y de objetos.

**Los objetivos del modelado de negocio son:**

- ✓ 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.
- ✓ Derivar los requerimientos del sistema que va a soportar la organización. (14)

Para lograr un mejor entendimiento de cómo funciona el negocio se modela el mismo mediante diagramas de actividades; donde se evidencia la secuencia de pasos que se llevan a cabo por los actores, trabajadores, siendo los últimos los que en ocasiones crean, modifican o consultan a contenedores de información llamados entidades.

#### 2.3.1 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (14)

Actor	Descripción
Participante	Puede ser cualquier persona de la comunidad universitario personas de organismos y/o instituciones.

**Tabla 1. Actor del negocio**

### 2.3.2 Trabajadores del negocio

Un trabajador del negocio representa un rol que juega una persona o grupo de personas, una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades. (14)

## Capítulo 2. Modelado de negocio y requerimientos

---

Trabajadores	Descripción
Dirección de Investigaciones(DI)	Es un especialista dentro de la DI facultado para lanzar a nivel de universidad la convocatoria de un evento.
Comité Organizador Facultad	Es un grupo de personas encargadas de lanzar la convocatoria de un evento a nivel de facultad y que recepciona los trabajos enviados.
Comité Científico Facultad	Es un grupo de personas encargadas en la facultad de evaluar los trabajos que se envían.
Comité Científico UCI	Es un grupo de personas encargadas de evaluar los trabajos propuestos por las facultades para participar en un evento a nivel de Universidad.

Tabla 2. Trabajadores del negocio

### 2.3.3 Diagrama de casos de uso del negocio

El diagrama de casos de uso del negocio es un modelo que describe los procesos de un negocio (casos de uso) y su interacción con los elementos externos (actores). Su objetivo fundamental es describir cómo el negocio es utilizado por sus clientes y socios. (14)

A continuación se define el modelo de negocio correspondiente al proceso de trámite de eventos científicos, desarrollado por la DI, luego de un estudio de cómo se lleva a cabo el proceso así como los actores y trabajadores que interactúan en el negocio:

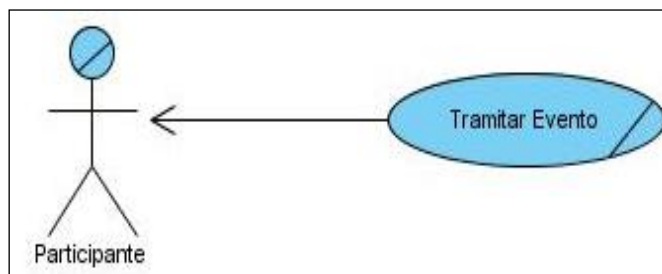


Figura 1. Diagrama de casos de uso del negocio

### 2.3.4 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El proceso de especificación implica que hay que “identificarlas” dentro del negocio, “evaluar” si son relevantes dentro del campo de acción que se está modelando e “implementarlas” en la propuesta de solución (14)

A continuación se especifican las reglas del negocio:

- ✓ Si el evento organizado por la universidad tiene niveles, primero debe lanzarse la convocatoria del nivel superior y posteriormente los niveles inferiores lanzarán las suyas.
- ✓ El nivel superior realizará su evento, solo cuando todas sus áreas hayan realizado el suyo.
- ✓ En el nivel base se podrá lanzar una convocatoria por cada área, mientras que en los demás niveles se lanzará solo una.

### 2.3.5 Descripción de los casos de uso del negocio

La descripción de un Caso de Uso del Negocio (CUN) muestra cómo colaboran los trabajadores y las entidades del negocio para ejecutar determinados procesos; a continuación se realiza la descripción del CUN identificado:

<b>Caso de Uso:</b>	Tramitar evento
<b>Actores:</b>	Participante

## Capítulo 2. Modelado de negocio y requerimientos

<b>Trabajadores:</b>	DI, Comité Organizador Facultad, Comité Científico Facultad, Comité Científico UCI.
<b>Resumen:</b>	El CUN comienza cuando la DI inicia la creación de un evento, primeramente crea la convocatoria, quedando iniciado de esta forma el evento, luego se identifican los demás datos por los que estará compuesto quedando listo para aprobación, luego que el evento sea aceptado se divulga a nivel de universidad. Se publica la convocatoria a nivel de facultad, los participantes envían sus trabajos, son seleccionados por el Consejo Científico de la Facultad los mejores trabajos y por último la DI los distribuye al Consejo Científico UCI para una revisión de los que se presentarán a nivel de universidad, terminando así el CUN.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	1. El caso de uso se inicia cuando la DI crea la convocatoria de un evento, quedando iniciado de esta forma el mismo.
	2. Un miembro de la DI identifica los datos adicionales para el evento, procede a la aceptación del mismo con los demás miembros.
	3. La DI acepta el evento y lo publica a nivel de universidad.
	4. El Comité Organizador de la Facultad lanza la convocatoria en ese nivel.
5. El participante consulta la	6. El Comité Organizador de la Facultad recepciona los

## Capítulo 2. Modelado de negocio y requerimientos

convocatoria y envía el trabajo a nivel de facultad.	trabajos enviados y los distribuye en el Comité Científico de la Facultad.
	7. El Comité Científico de la Facultad revisa los trabajos y envía a la DI los aceptados.
	8. La DI recepciona todos los trabajos aceptados a nivel de Facultad y los distribuye en el Comité Científico UCI.
	9. El Comité Científico UCI realiza el arbitraje de los trabajos.
	10. El Comité Científico UCI emite sugerencias de mejora y envía al participante las mismas.
11. El participante recibe las sugerencias del trabajo, las arregla y procede a enviar nuevamente el trabajo al Comité Científico UCI.	
<b>Flujo alternativo de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	3.1 La DI rechaza el evento e informa al desarrollador que debe arreglarlo.
	10.1 El Comité Científico UCI no emite sugerencias de mejora y envía los resultados finales a la DI.
	10.2 La DI notifica los resultados finales al Comité Organizador de cada facultad.
	10.3 El Comité Organizador de cada facultad recibe la notificación de la DI e informa los resultados al participante.
10.4 El participante recibe la notificación de resultados, finalizando	

## Capítulo 2. Modelado de negocio y requerimientos

así el CUN.

Tabla 3. Descripción del caso de uso: Tramitar evento

### 2.3.6 Diagrama de actividades

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. (14)

Los diagramas de actividades correspondientes a cada CUN tienen representados con un color más fuerte las actividades a automatizar. A continuación se muestra el diagrama de actividades general de todo el negocio:

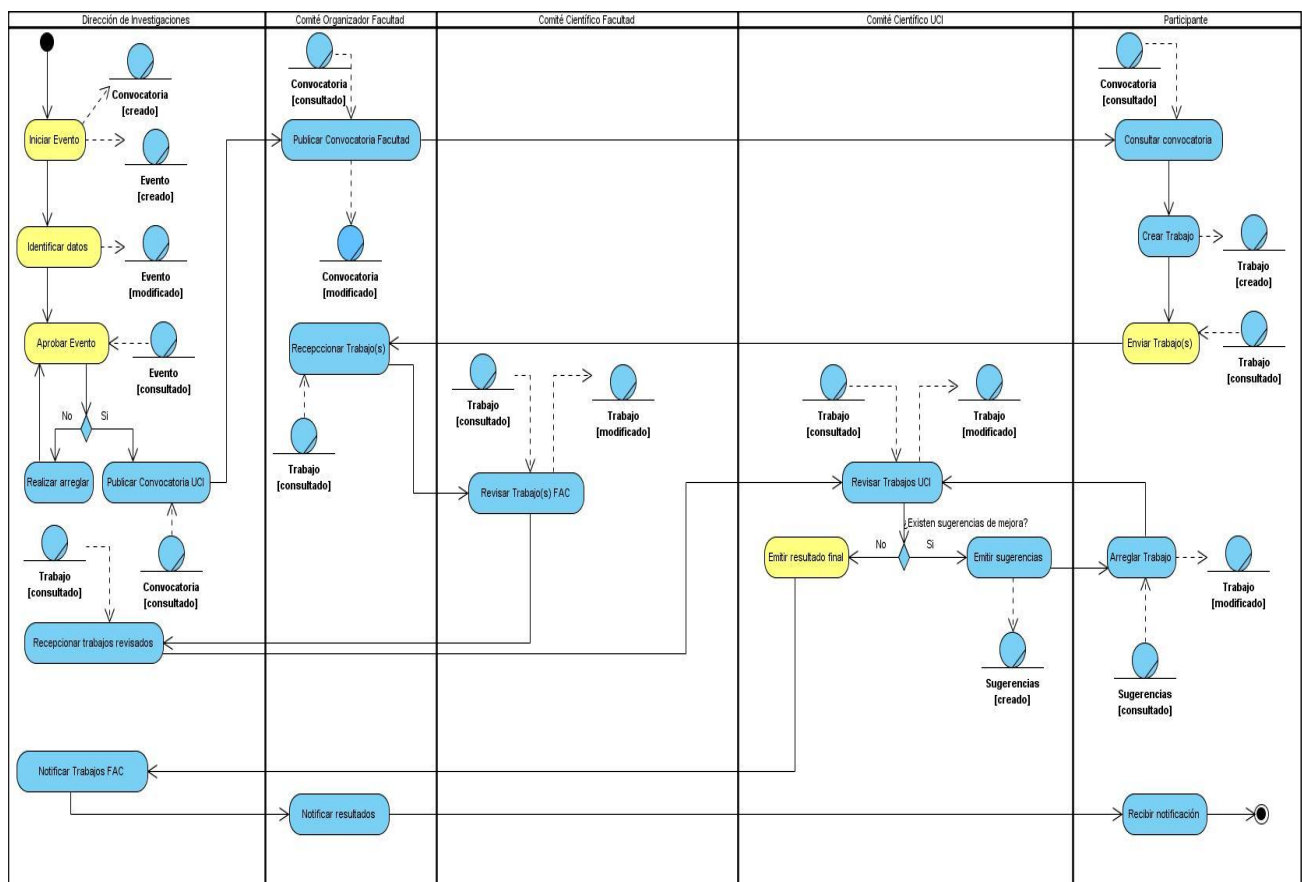


Figura 2. Diagrama de actividades del caso de uso: Tramitar evento



### 2.3.7 Modelo de objetos del negocio

Es un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo modelado de negocio. (14). A continuación se muestra la figura correspondiente al diagrama de clases del modelo de objetos del caso de uso Tramitar evento:

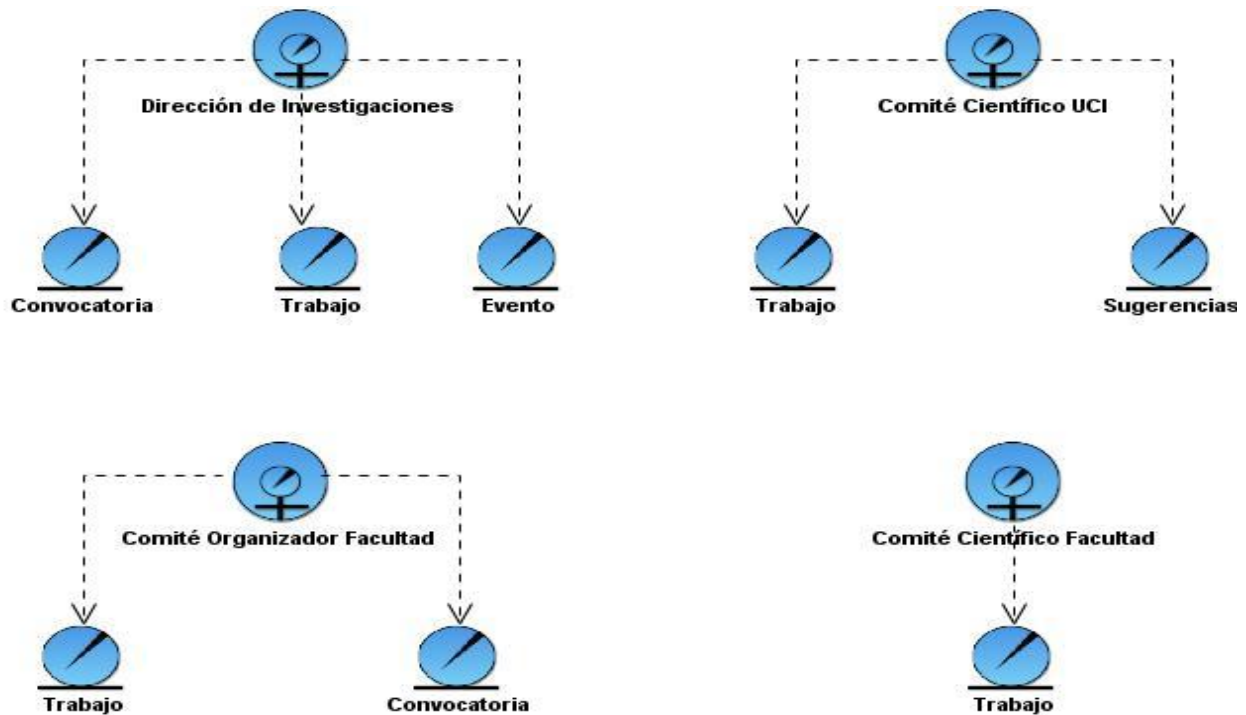


Figura 3. Diagrama de clases del modelo de objetos del caso de uso: Tramitar evento

### 2.4 Especificación de los requisitos de software

Una etapa inicial y muy importante dentro del proceso de la Ingeniería de Software, es la Ingeniería de Requisitos, donde se realiza el proceso de descubrir, analizar, escribir y verificar los servicios del sistema de software que se desea producir; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requisitos del software. Los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar. (14)

### 2.4.1 Requerimientos funcionales

Los requisitos funcionales son condiciones o capacidades que el sistema debe realizar, es decir define qué es lo que el sistema debe hacer. A continuación se definen los identificados durante la entrevista con el cliente y basados en las necesidades existentes en la DI:

#### Módulo Evento

En el módulo se realiza la gestión de toda la información correspondiente al evento.

##### RF1 Gestionar convocatoria

RF1.1 Adicionar convocatoria.

RF1.2 Modificar convocatoria.

RF1.3 Eliminar convocatoria.

RF1.4 Listar convocatoria

##### RF2 Gestionar evento

RF2.1 Adicionar evento

RF2.1.1 Especificar temáticas.

RF2.1.2 Especificar noticias.

RF2.1.3 Especificar políticas de envío.

RF2.1.4 Especificar comité científico.

RF2.1.5 Especificar programa general del evento.

RF2.1.6 Especificar memorias del evento.

RF2.1.7 Especificar eventos próximos.

RF2.2 Modificar evento.

RF2.3 Eliminar evento.

RF2.4 Listar evento.

##### RF3 Gestionar temática.

RF3.1 Adicionar temática.

RF3.2 Modificar temática.

RF3.3 Eliminar temática.

RF3.4 Listar temática.

RF4 Gestionar Comité científico.

RF4.1 Adicionar Comité científico.

RF4.2 Modificar Comité científico.

RF4.3 Eliminar Comité científico.

RF4.4 Listar Comité científico.

RF5 Aprobar evento.

### **Módulo Recepción de Información**

En el módulo se realiza la recepción de los trabajos enviados por los participantes y se registran los resultados de la revisión realizada.

RF6 Registrar revisión.

RF6.1 Listar trabajo.

RF7 Enviar trabajo.

RF8 Recepcionar trabajo.

### **Módulo Seguridad**

En el módulo se realiza la configuración de los permisos, roles y la seguridad de acceso al sistema.

RF9 Autenticar usuario.

RF10 Gestionar usuario.

RF10.1 Adicionar usuario.

RF10.2 Modificar usuario

RF10.3 Eliminar usuario.

RF10.4 Listar usuario.

RF11 Gestionar rol.

RF11.1 Adicionar rol.

RF11.2 Modificar rol.

RF11.3 Eliminar rol.

RF11.4 Listar rol.

RF12 Asignar rol.

RF13 Gestionar permiso

RF13.1 Adicionar permiso

RF13.2 Modificar permiso.

RF13.3 Eliminar permiso

RF13.4 Listar permiso.

RF14 Asignar permiso.

Los requisitos identificados a través de la entrevista con el cliente fueron validados mediante prototipos de interfaz de usuario, donde la DI estuvo de acuerdo y certificó su aceptación mediante un aval (*Ver anexo 1*).

### **2.4.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son todas las características que hacen al producto atractivo, usable, rápido y confiable. (14)

A continuación se describen los requisitos no funcionales en aras de establecer restricciones y cualidades que debe poseer la aplicación, contribuyendo a que la misma pueda ser utilizada fácilmente y provea una interfaz sencilla y profesional.

### 1. Apariencia o interfaz externa

- 1.1. La aplicación web debe contar con una interfaz profesional, siguiendo una arquitectura de información, permitiendo que los usuarios finales de la misma sean capaces de interactuar con esta, aun cuando solo posean conocimientos básicos en el manejo de las computadoras.
- 1.2. Empleo de imágenes y colores claros, identificados con el evento que gestionará la aplicación.

### 2. Usabilidad

- 2.1. La aplicación podrá ser utilizada por personas que tengan un conocimiento básico en el manejo de las computadoras.
- 2.2. La aplicación tendrá siempre visible la opción de Ayuda, lo que posibilitará una mejor explotación de las funcionalidades por parte del administrador.

### 3. Software

La aplicación se ejecutará siguiendo las siguientes especificaciones:

- 3.1. Cualquiera de los sistemas operativos: Linux o Windows.
- 3.2. Navegadores web: Internet Explorer 6 o superior, Mozilla/Firefox (2.0 o superior).

### 4. Hardware

4.1 Para las PCs clientes:

- ✓ Tarjeta de red.
- ✓ Mínimo 256 MB de RAM.
- ✓ Disco duro de 10 GB como mínimo.

4.2 Para el servidor:

- ✓ Tarjeta de red.
- ✓ El Servidor de BD debe tener 1GB de RAM y 60 GB de disco duro como mínimo.

- ✓ El Servidor web debe tener 1GB de RAM.

### 5 Restricciones en el diseño y la implementación

Emplear las siguientes herramientas:

- ✓ Servidor de bases de datos PostgreSQL 8.0 o superior.
- ✓ Máquina virtual de Java: JDK 6.0.
- ✓ Servidor web Apache Tomcat 6.0.14.
- ✓ Visual Paradigm for UML 6.4.
- ✓ Netbeans IDE 6.9.
- ✓ Spring 2.5.6.
- ✓ Hibernate3.5.

### 6 Soporte

6.1 Se necesita un servidor de bases de datos que soporte una capacidad media de datos.

### 7 Seguridad

7.1 Para lograr la confidencialidad, integridad y disponibilidad de la aplicación se necesita:

- ✓ Crear diferentes cuentas de usuario y asignarle a cada una solo los permisos que le corresponden.
- ✓ Mostrar a cada usuario sólo las funcionalidades de la aplicación sobre las cuales tiene permiso de acceso.
- ✓ Ofrecer mensajes de verificación antes de ejecutar acciones irreversibles (eliminación de datos).
- ✓ El servidor donde se encuentre instalada la aplicación debe estar situado en un local protegido, donde no esté expuesto a desastres naturales o robo.
- ✓ La aplicación está diseñada para su funcionamiento constante, permitiendo el acceso a los servicios que brinda durante los siete días de la semana y 24 horas del día.

### 2.5 Modelo de casos de uso del sistema

#### 2.5.1 Actores del sistema

Los actores del sistema son entidades que guardan relación con este, pueden ser personas, otros sistemas que interactúen con el mismo o entidades abstractas como puede ser el tiempo.

Actor	Descripción
Administrador	Usuario con privilegios para gestionar toda la información referente al evento, la gestión de usuarios y la asignación de permisos.
Ponente	Usuario que puede enviar trabajos en la aplicación.
Dirección de Investigaciones	Usuario que gestiona convocatoria, aprueba eventos y recepciona trabajos.
Comité Científico UCI	Usuario que realiza el arbitraje de los trabajos para seleccionar los que se presentarán a nivel UCI.

Tabla 4. Actores del sistema

#### 2.5.2 Casos de uso del sistema (CUS)

Los casos de uso son el componente fundamental del modelado, se centran en ilustrar como un sistema permite a un actor alcanzar una meta determinada, mostrando los caminos posibles y más apropiados que pueden seguir para el logro de la misma, así como las situaciones que podrían hacerlo fallar. A continuación se muestran los casos de uso del sistema identificados:

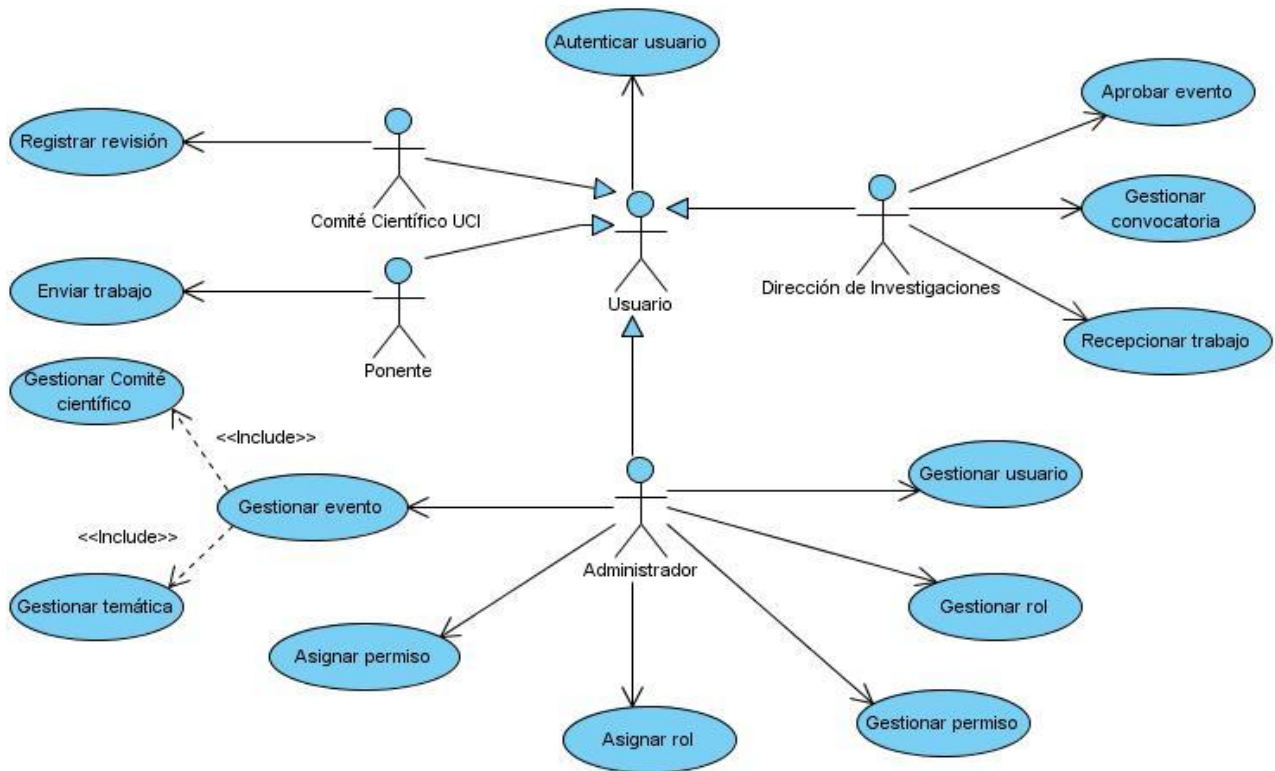


Figura 4. Diagrama de Casos de Uso del Sistema.

### 2.4.1 Descripción de los casos de uso del sistema

Para entender la funcionalidad correspondiente a cada caso de uso no es suficiente con la representación gráfica del Diagrama de Casos de Uso del Sistema (DCUS), se hace necesario realizar la descripción de cada caso de uso identificado. A continuación se describen los casos de uso Gestionar convocatoria y Gestionar evento, los demás quedan especificados en los anexos (*Ver Anexo 2*).

<b>CU-1:</b>	Gestionar convocatoria
<b>Actor:</b>	Dirección de Investigaciones
<b>Resumen:</b>	El CUS se inicia cuando la DI interactúa con la aplicación para adicionar, modificar, eliminar o listar la convocatoria de un evento. El CUS finaliza cuando se ejecuta cualquiera de estas acciones.
<b>Referencia:</b>	RF1.1, RF1.2, RF1.3, RF1.4



## Capítulo 2. Modelado de negocio y requerimientos

<b>Precondición:</b>	
<b>Poscondición:</b>	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. La DI accede a la aplicación y selecciona el módulo Evento.	2. El sistema muestra la funcionalidad Gestionar convocatoria.
3. La DI escoge la opción mostrada.	4. El sistema muestra las siguientes funcionalidades: <ul style="list-style-type: none"> <li>- Adicionar convocatoria.</li> <li>- Modificar convocatoria.</li> <li>- Eliminar convocatoria.</li> <li>- Listar convocatoria.</li> </ul>
5. La DI selecciona una de las funcionalidades.	6. Si la Dirección de Investigaciones selecciona: <ul style="list-style-type: none"> <li>- Adicionar convocatoria ver sección: “Adicionar convocatoria”.</li> <li>- Modificar convocatoria ver sección: “Modificar convocatoria”.</li> <li>- Eliminar convocatoria ver sección: “Eliminar convocatoria”.</li> <li>- Listar convocatoria ver sección: “Listar convocatoria”.</li> </ul>
Prototipo de Interfaz	

**Gestionar convocatoria**

Adicionar    Modificar    Eliminar

Nro	Titulo	Descripción

### Sección: "Adicionar convocatoria"

#### Flujo normal de eventos

Acción del Actor	Respuesta del Sistema
<p>7. La DI introduce los siguientes datos para la convocatoria:</p> <ul style="list-style-type: none"> <li>- Título de la convocatoria.</li> <li>- Descripción de la convocatoria.</li> </ul> <p>8. La DI escoge la opción Aceptar.</p>	<p>9. El sistema verifica que al adicionar los datos de la convocatoria no existan campos vacíos, datos incorrectos y que no se encuentre creada. En este caso se guarda la convocatoria, finalizando así el CUS.</p>

#### Prototipo de Interfaz


**Adicionar convocatoria**

Título de la convocatoria


Descripción

Aceptar    Cancelar

## Capítulo 2. Modelado de negocio y requerimientos

Flujo alternativo de eventos	
Acción del Actor	Respuesta del Sistema
8.1 La DI escoge la opción Cancelar. <b>Ir al flujo básico 4.</b>	9.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 7.</b>
Sección: “Modificar convocatoria”	
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
8. La DI selecciona la convocatoria a la que desea modificar datos y escoge la opción Modificar.	7. El sistema lista las convocatorias que han sido creadas.  9. El sistema muestra los siguientes datos para modificar la convocatoria: - Título de la convocatoria. - Descripción de la convocatoria.
10. La DI modifica los datos que desea y escoge la opción Aceptar.	11. El sistema verifica que al modificar los datos de la convocatoria no existan campos vacíos o datos incorrectos. En este caso el sistema actualiza el listado de convocatorias y guarda los cambios realizados, finalizando así el CUS.
Prototipo de Interfaz	
	

## Capítulo 2. Modelado de negocio y requerimientos

Flujo alternativo de eventos	
Acción del Actor	Respuesta del Sistema
10.1 La DI escoge la opción Cancelar. <b>Ir al flujo básico 7.</b>	11.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 9.</b>
Sección: “Eliminar convocatoria”	
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
8. La DI selecciona la convocatoria que desea eliminar.	7. El sistema lista todas las convocatorias que han sido creadas.  9. El sistema muestra el siguiente mensaje de confirmación: “Desea eliminar la convocatoria seleccionada”.
10. La DI escoge la opción Aceptar.	11. El sistema elimina la convocatoria y actualiza el listado, finalizando así el CUS.
Prototipo de Interfaz	
	
Flujo alternativo de eventos	
Acción del Actor	Respuesta del Sistema
10.1 La DI escoge la opción Cancelar. <b>Ir al flujo básico 7.</b>	
Sección: “Listar convocatoria”	
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
	7. El sistema lista las convocatorias que han sido


## Capítulo 2. Modelado de negocio y requerimientos

	creadas, finalizando así el CUS.
<b>Prioridad</b>	Crítico

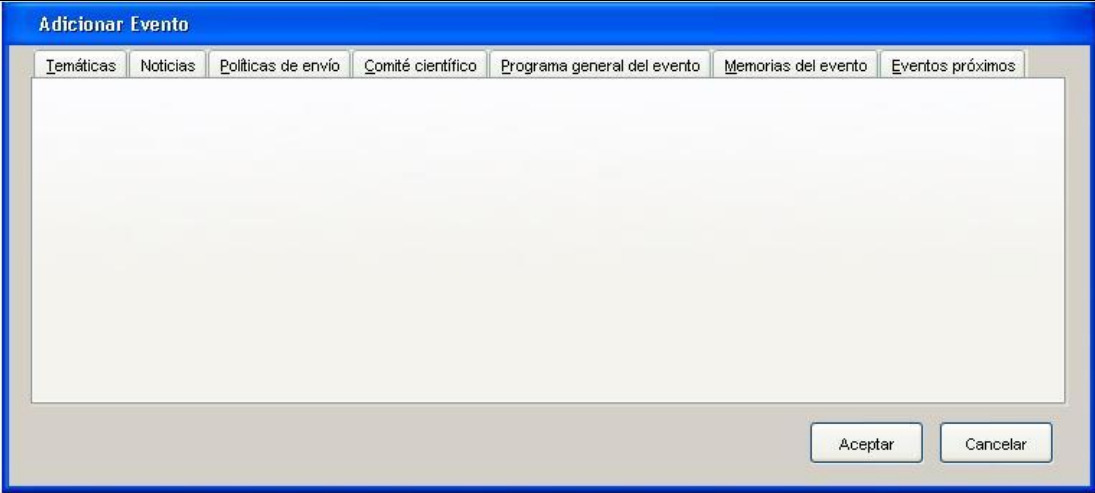
Tabla 5. Descripción del Caso de uso: Gestionar convocatoria

<b>CU-2:</b>	Gestionar evento	
<b>Actor:</b>	Administrador	
<b>Resumen:</b>	El CUS se inicia cuando el Administrador interactúa con la aplicación para adicionar, modificar, eliminar o listar un evento. El CUS finaliza cuando se ejecuta cualquiera de estas acciones.	
<b>Referencia</b>	RF2.1, RF2.2, RF2.3, RF2.4	
<b>Precondición:</b>		
<b>Poscondición:</b>		
<b>Flujo normal de eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El Administrador accede a la aplicación y selecciona el módulo Evento.	2. El sistema muestra la funcionalidad:  Gestionar evento.
	3. El Administrador selecciona la opción mostrada.	4. El sistema muestra las siguientes funcionalidades:  - Adicionar evento. - Modificar evento. - Eliminar evento. - Listar evento.
	5. El Administrador selecciona una de las funcionalidades.	6. Si el Administrador selecciona:  - Adicionar evento ver sección: "Adicionar evento".

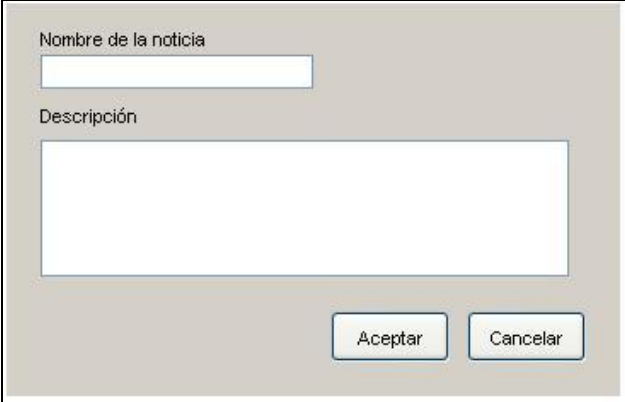
## Capítulo 2. Modelado de negocio y requerimientos

	<ul style="list-style-type: none"> <li>- Modificar evento ver sección: “Modificar evento”.</li> <li>- Eliminar evento ver sección: “Eliminar evento”.</li> <li>- Listar evento ver sección: “Listar evento”.</li> </ul>
<b>Prototipo de Interfaz</b>	
	
<b>Sección: “Adicionar evento”</b>	
<b>Flujo normal de eventos</b>	
Acción del Actor	Respuesta del Sistema
	<p>7. El sistema muestra las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>- Temáticas.</li> <li>- Noticias.</li> <li>- Políticas de envío.</li> <li>- Comité científico.</li> <li>- Programa general del evento.</li> <li>- Memorias del evento.</li> <li>- Eventos próximos.</li> </ul>
8. El Administrador selecciona una de las funcionalidades.	<p>9. Si el administrador selecciona:</p> <ul style="list-style-type: none"> <li>- Temáticas se inicializa el CU: “Gestionar temática”.</li> </ul>

## Capítulo 2. Modelado de negocio y requerimientos

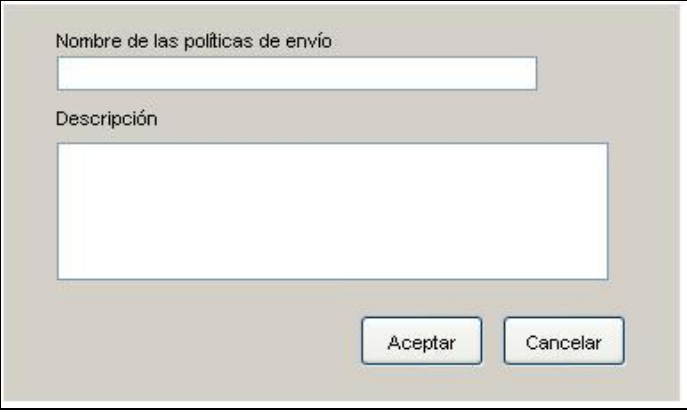
	<ul style="list-style-type: none"> <li>- Noticia ver sección: “Especificar noticia”.</li> <li>- Políticas de envío ver sección: “Especificar políticas de envío”.</li> <li>- Comité científico se inicializa el CU: “Gestionar Comité Científico”.</li> <li>- Programa general del evento ver sección: “Especificar programa general del evento”.</li> <li>- Memorias del evento ver sección: “Especificar memorias del evento”.</li> <li>- Eventos próximos ver sección: “Especificar eventos próximos”.</li> </ul>
<b>Prototipo de Interfaz</b>	
	
<b>Sección: “Especificar noticia”</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>10. El Administrador introduce los siguientes datos de la noticia:</p> <ul style="list-style-type: none"> <li>- Nombre de la noticia.</li> <li>- Descripción de la noticia.</li> </ul>	<p>12. El sistema verifica que al especificar la noticia no existan campos vacíos, datos incorrectos y que no se encuentre creada. En este caso guarda la noticia, finalizando así el CUS.</p>

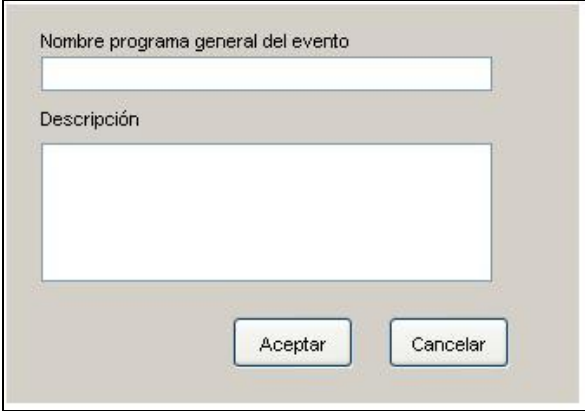
## Capítulo 2. Modelado de negocio y requerimientos

11.El Administrador escoge la opción Aceptar.	
<b>Prototipo de Interfaz</b>	
	
<b>Flujo alternativo de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
11.1 El Administrador escoge la opción Cancelar. <b>Ir al flujo básico 7.</b>	12.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 10.</b>
<b>Sección: “Especificar políticas de envío”</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
10.El Administrador introduce los siguientes datos de las políticas de envío: <ul style="list-style-type: none"> <li>- Nombre de las políticas de envío.</li> <li>- Descripción de las políticas de envío.</li> </ul> 11. El Administrador escoge la opción Aceptar.	12. El sistema verifica que al especificar las políticas de envío no existan campos vacíos, datos incorrectos y que no se encuentren creadas. En este caso guarda las políticas de envío, finalizando así el CUS.
<b>Prototipo de Interfaz</b>	



## Capítulo 2. Modelado de negocio y requerimientos

	
<b>Flujo alternativo de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
11.1 El Administrador escoge la opción Cancelar. <b>Ir al flujo básico 4.</b>	12.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 10.</b>
<b>Sección: “Programa general del evento”</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>10. El Administrador introduce los siguientes datos para el programa general del evento:</p> <ul style="list-style-type: none"> <li>- Nombre del programa general del evento.</li> <li>- Descripción del programa general del evento.</li> </ul> <p>11. El Administrador escoge la opción Aceptar.</p>	12. El sistema verifica que al especificar el programa general del evento no existan campos vacíos, datos incorrectos y que no se encuentre creado. En este caso guarda el programa general del evento, finalizando así el CUS.
<b>Prototipo de Interfaz</b>	

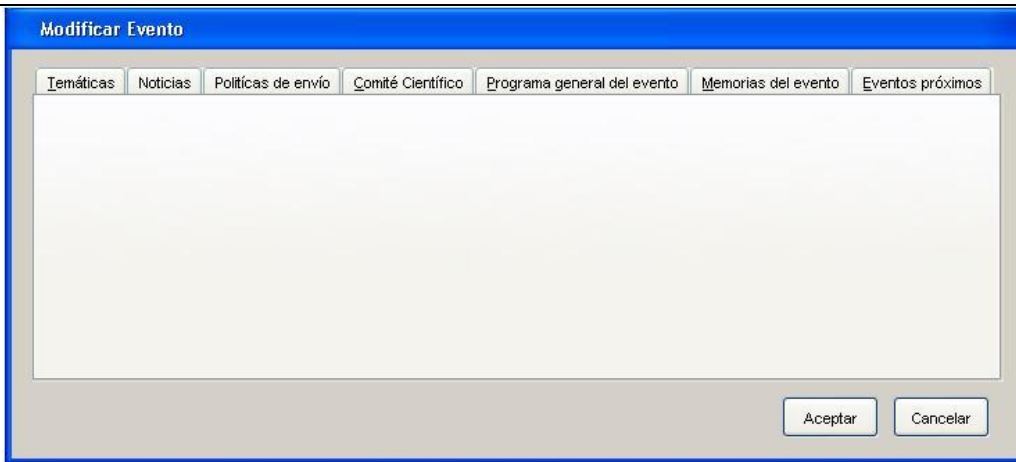
	
<b>Flujo alterno de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
11.1 El Administrador escoge la opción Cancelar. <b>Ir al flujo básico 4.</b>	12.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 10.</b>
<b>Sección: “Memorias del evento”</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
10. El Administrador introduce los siguientes datos para las memorias del evento: <ul style="list-style-type: none"> <li>- Nombre de las memorias del evento.</li> <li>- Descripción de las memorias del evento.</li> </ul> 11. El Administrador escoge la opción Aceptar.	12. El sistema verifica que al especificar las memorias del evento no existan campos vacíos, datos incorrectos y que no se encuentren creadas. En este caso guarda las memorias del evento, finalizando así el CUS.
<b>Prototipo de Interfaz</b>	

## Capítulo 2. Modelado de negocio y requerimientos

<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p>Nombre de las memorias del evento</p> <input type="text"/> <p>Descripción</p> <div style="border: 1px solid black; height: 60px; width: 100%;"></div> <p style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>	
<b>Flujo alternativo de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
11.1 El Administrador escoge la opción Cancelar. <b>Ir al flujo básico 4.</b>	12.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 10.</b>
<b>Sección: "Eventos próximos"</b>	
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
10. El Administrador introduce los siguientes datos para los eventos próximos: <ul style="list-style-type: none"> <li>- Nombre de los eventos próximos.</li> <li>- Descripción de los eventos próximos.</li> </ul> 11. El Administrador escoge la opción Aceptar.	12. El sistema verifica que al especificar los eventos próximos no existan campos vacíos, datos incorrectos y que no se encuentren creados. En este caso guarda las memorias del evento, finalizando así el CUS.
<b>Prototipo de Interfaz</b>	
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p>Nombre eventos próximos <input type="text"/></p> <p>Descripción</p> <div style="border: 1px solid black; height: 40px; width: 100%;"></div> <p style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>	

## Capítulo 2. Modelado de negocio y requerimientos

Flujo alternativo de eventos	
Acción del Actor	Respuesta del Sistema
11.1 El Administrador escoge la opción Cancelar. <b>Ir al flujo básico 4.</b>	12.1 El sistema muestra un mensaje de error. <b>Ir al flujo básico 10.</b>
Sección: “Modificar evento”	
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
8. El Administrador selecciona el evento al cual desea modificar datos.	7. El sistema lista todos los eventos que han sido creados.  9. El sistema muestra los siguientes datos para modificar el evento: <ul style="list-style-type: none"> <li>- Temáticas.</li> <li>- Noticias.</li> <li>- Políticas de envío.</li> <li>- Comité científico.</li> <li>- Programa general del evento.</li> <li>- Memorias del evento.</li> <li>- Eventos próximos.</li> </ul>
10. El Administrador modifica los datos que desea y escoge la opción Aceptar.	11. El sistema verifica que al modificar los datos del evento no existan campos vacíos o datos incorrectos. En este caso el sistema actualiza el listado de eventos y guarda los cambios realizados, finalizando así el CUS.
Prototipo de Interfaz	



### Flujo alternativo de eventos

Acción del Actor	Respuesta del Sistema
10.1 El Administrador escoge la opción Cancelar. Ir al flujo básico 7.	11.1 El sistema muestra un mensaje de error. Ir al flujo básico 9.

### Sección "Eliminar evento"

#### Flujo normal de eventos

Acción del Actor	Respuesta del Sistema
	7. El sistema lista todos los eventos que han sido creados.
8. El Administrador selecciona el evento que desea eliminar.	9. El sistema muestra el siguiente mensaje de verificación "Desea eliminar el evento seleccionado".
10. El Administrador escoge la opción Aceptar.	11. El sistema elimina el evento y actualiza el listado, finalizando así el CUS.

### Prototipo de Interfaz



## Capítulo 2. Modelado de negocio y requerimientos

---

Flujo alternativo de eventos	
Acción del Actor	Respuesta del Sistema
10.1 El Administrador escoge la opción Cancelar. Ir al flujo básico 7.	
Sección: "Listar evento"	
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
	7. El sistema lista todos los eventos que se encuentren creados, finalizando así el CUS.
<b>Prioridad</b>	Crítico

Tabla 6. Descripción del caso de uso: Gestionar evento.

### 2.5 Conclusiones

La adecuada descripción del estado actual del negocio posibilitó una correcta identificación de los principales procesos del mismo, describir detalladamente los correspondientes casos de uso y modelar los respectivos diagramas de actividades y el modelo de objetos. Los requisitos determinados proporcionan la información necesaria para dar comienzo al análisis y diseño del sistema.

### Capítulo 3. Análisis y diseño del sistema

#### 3.1 Introducción

En el presente capítulo se modelan los artefactos correspondientes al análisis y diseño de la solución que se expone. Se realiza el diagrama de clases del análisis, se desarrollan las clases del diseño, conformando el modelo del diseño y se muestra el modelo de datos del sistema.

#### 3.2 Modelo de análisis

Las actividades del análisis son desarrolladas con el objetivo de facilitar la entrada al diseño, por tanto constituyen un paso inicial y una primera aproximación conceptual para aumentar el nivel de especificidad en aras de garantizar el cumplimiento de los requisitos funcionales y obtener una visión de qué hace el sistema (14)

El modelo de análisis está compuesto por artefactos como: clases del análisis y realizaciones de casos de uso del análisis. Las clases del análisis van a representar abstracciones de conceptos, en las cuales deben incluirse atributos y operaciones a un nivel alto. La realización de casos de uso del análisis describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos en interacción. Para ello RUP propone que para cada caso de uso:

- ✓ Se construya un diagrama de clases que muestre las clases de análisis participantes.
- ✓ Se construya alguno de los tipos de diagrama de interacción para representar el flujo en términos de objetos del análisis. (14)

##### 3.2.1 Diagramas de clases del análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. A continuación se muestran los diagramas de clases del análisis correspondiente a los casos de uso Gestionar convocatoria y Gestionar evento, los demás quedan reflejados en los anexos (*Ver Anexo 3*).

## Capítulo 3. Análisis y diseño del sistema

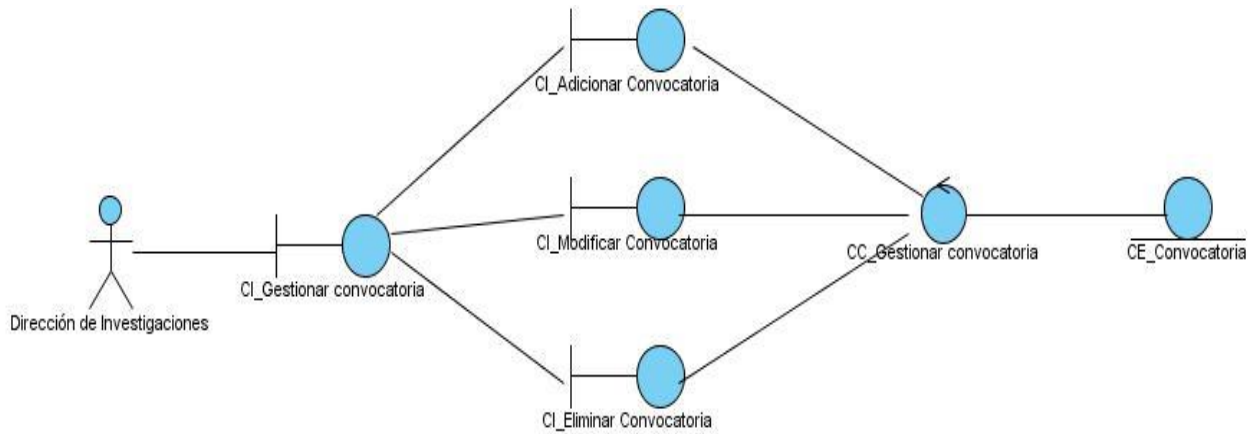


Figura 5. Diagrama de clases del análisis CU: Gestionar convocatoria

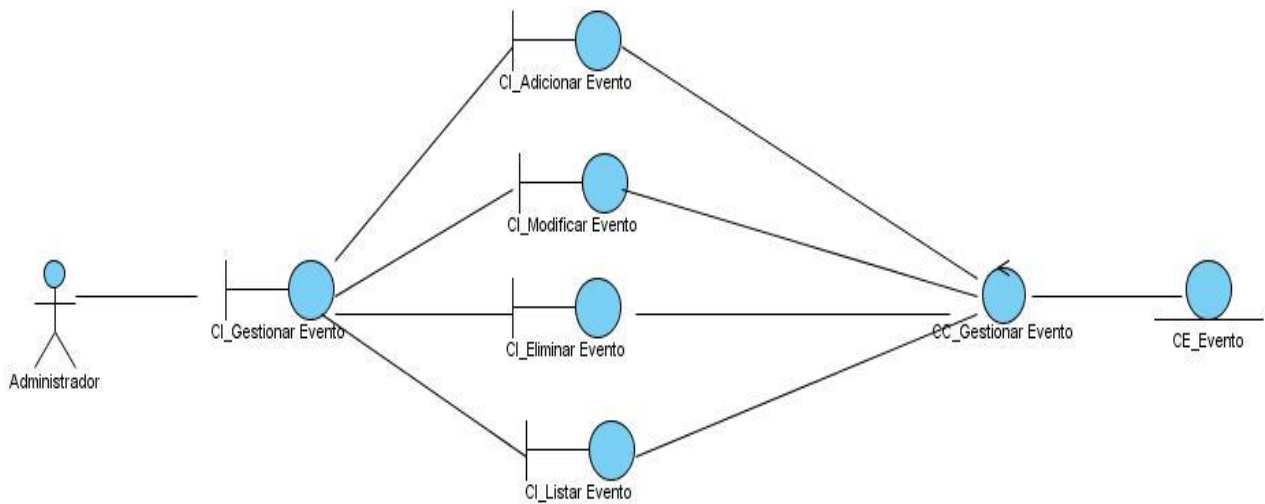


Figura 6. Diagrama de clases del análisis CU: Gestionar evento



### 3.3 Modelo de diseño

RUMBAUGH, JACOBSON y BOOCH, en el libro “El lenguaje unificado de modelado”, señalan que el diseño es: “esa etapa de un sistema que describe cómo se implementará el sistema, en un nivel lógico sobre código real. En el diseño, las decisiones estratégicas y tácticas se toman para resolver los requisitos funcionales y de calidad requeridos de un sistema. Los resultados de esta etapa son representados por los modelos a nivel de diseño, especialmente la vista estática, vista de la máquina de estados, y vista de interacción. Contraste con: análisis, diseño, implementación, y despliegue (14).

De forma general en el diseño se modela el sistema, contribuyendo a una arquitectura estable y sólida, para que soporte todos los requisitos, tanto funcionales como no funcionales y para llevar a cabo la creación del modelo de implementación. El diseño posibilita una entrada apropiada y un punto de partida para las actividades de la implementación, descompone los trabajos de implementación en partes más manejables que pueden ser llevados a cabo por diferentes equipos de desarrollo. El modelo de diseño se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica.

#### 3.3.1 Diagrama de clases del diseño

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones del sistema con sus relaciones estructurales, dando así una vista del diseño estático. En el caso de las aplicaciones web el diagrama de clases de diseño representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. (14)

Para la realización de los diagramas del diseño se tuvo en cuenta la estructura del marco de trabajo Spring, modelando en los mismos solo la lógica de presentación, mediante los controladores que provee, los componentes del acceso a datos no se modelaron al ser transparentes al programador y en aras de administrar la complejidad del diseño.

Seguidamente se representan los diagramas de clases de diseño de los casos de uso Gestionar convocatoria y Gestionar evento, los demás quedan reflejados en los anexos (*Ver Anexo 4*).

# Capítulo 3. Análisis y diseño del sistema

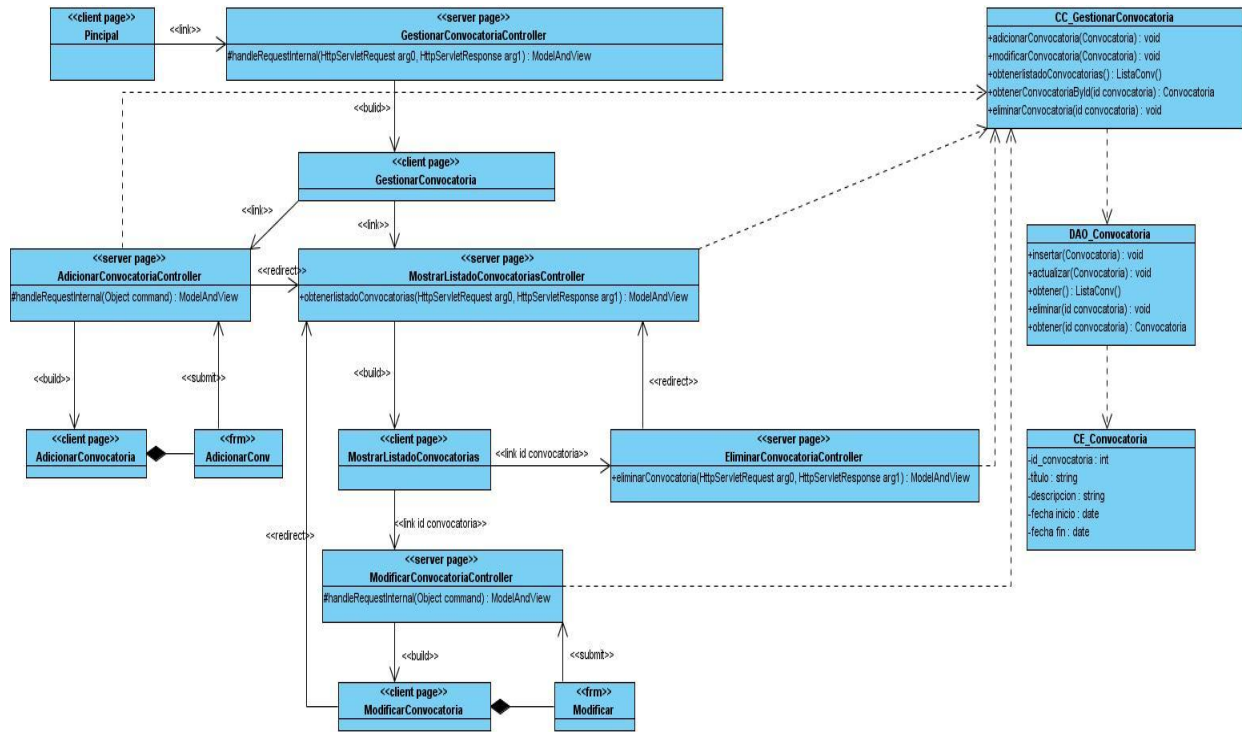


Figura 7. Diagrama de clases del diseño web: Gestionar convocatoria

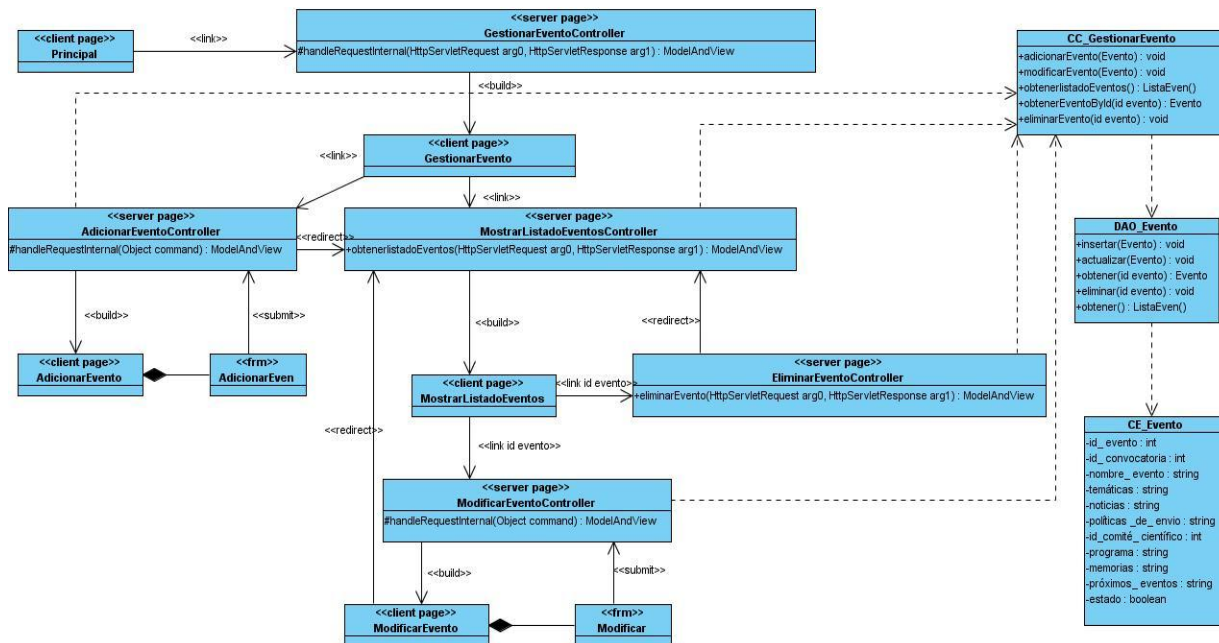


Figura 8. Diagrama de clases del diseño web: Gestionar evento

### 3.3.2 Diagramas de interacción. Diagramas de secuencia

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el escenario que ilustra un determinado comportamiento.

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos. (37)

A continuación se muestra el diagrama de secuencia perteneciente al caso de uso Gestionar evento, los demás se muestran en los anexos (Ver Anexo 5).

# Capítulo 3. Análisis y diseño del sistema

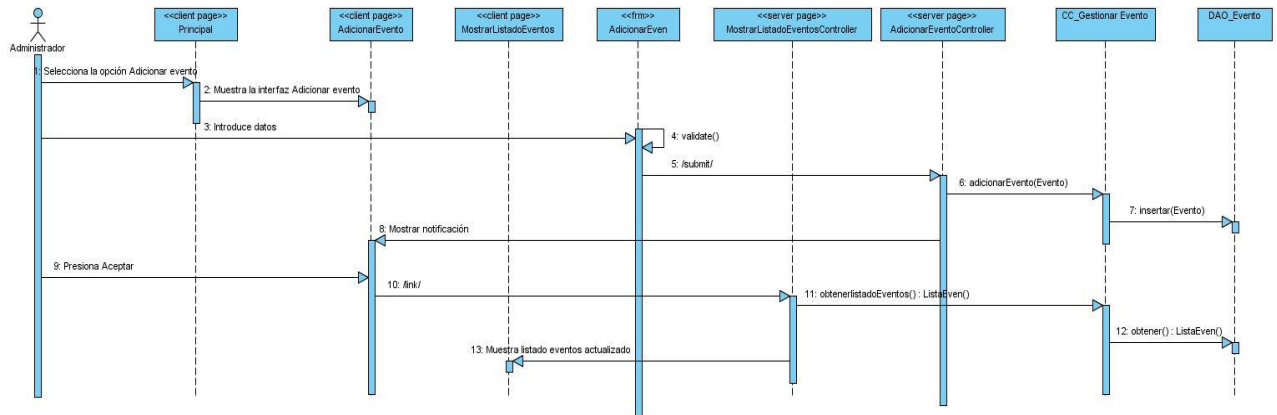


Figura 9. Gestionar evento: Escenario Adicionar evento

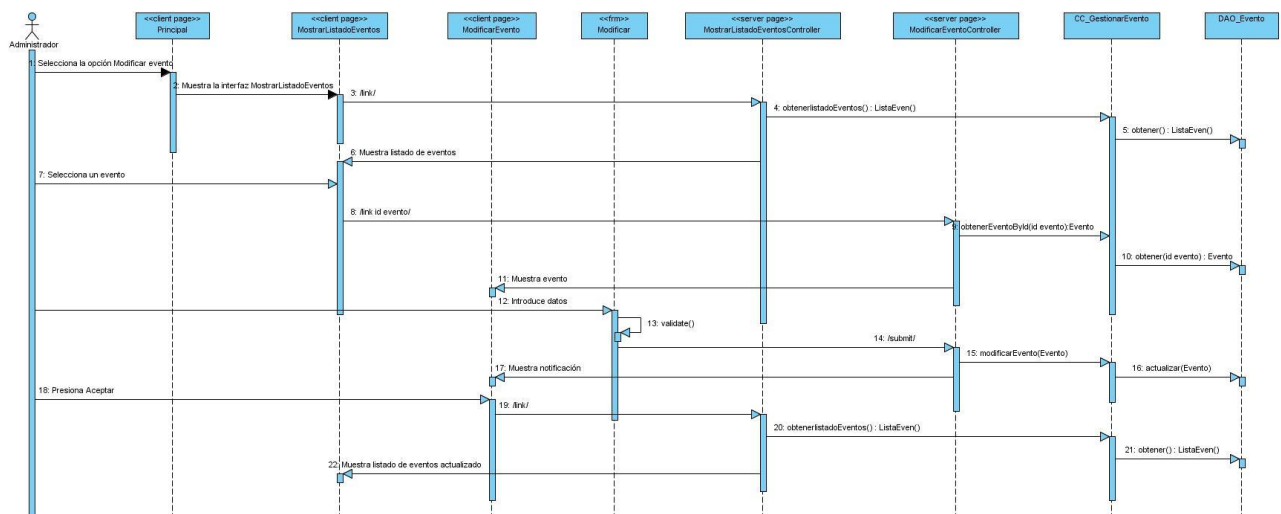
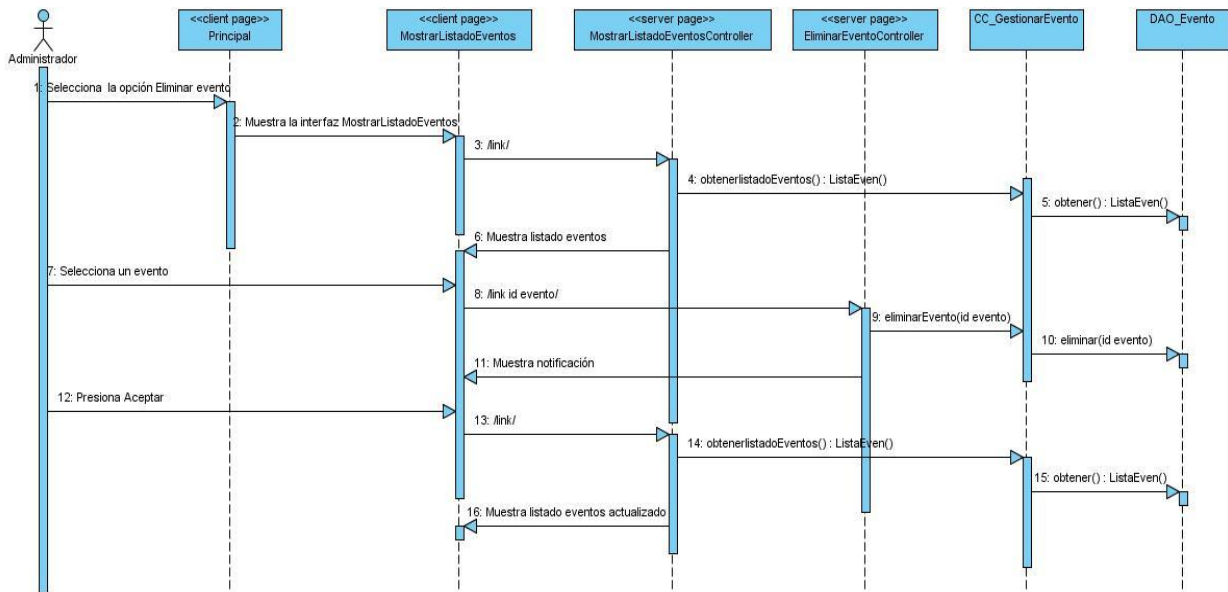


Figura 10 Gestionar evento: Escenario Modificar evento



**Figura 11 Gestionar evento: Escenario Eliminar evento**

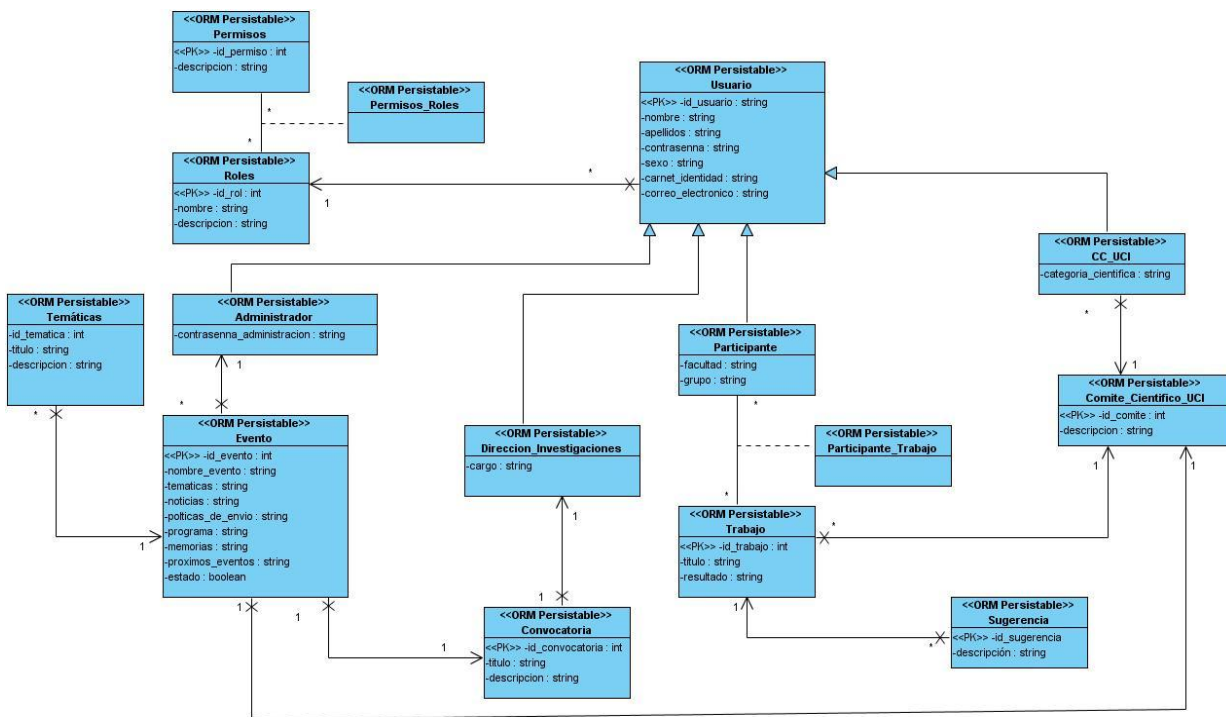
### 3.3.3 Diseño de la base de datos

Generar las tablas que modelan los registros en los que se guarda la información del sistema es el principal objetivo del diseño de la base de datos. Para recuperar de forma rápida y eficiente los datos, es necesario que la información que se almacene no sea redundante.

### Diagrama de clases persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Por lo general las clases persistentes tienen como origen las clases clasificadas como entidad porque ellas modelan la información del sistema y el comportamiento asociado de algún fenómeno o concepto. El diagrama de clases persistentes describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. (38)

A continuación se muestra el diagrama de clases persistentes de la base de datos:



**Figura 12. Diagrama de clases persistentes de la Base de datos**

### Diagrama Entidad Relación de la Base de Datos

El diagrama entidad-relación representa la estructura que debe tener la base de datos para almacenar los objetos persistentes del sistema. El modelo de datos propuesto en la solución cuenta con un total de 15 tablas, para su construcción se tuvo en cuenta la reducción a la mínima expresión de los campos nulos.

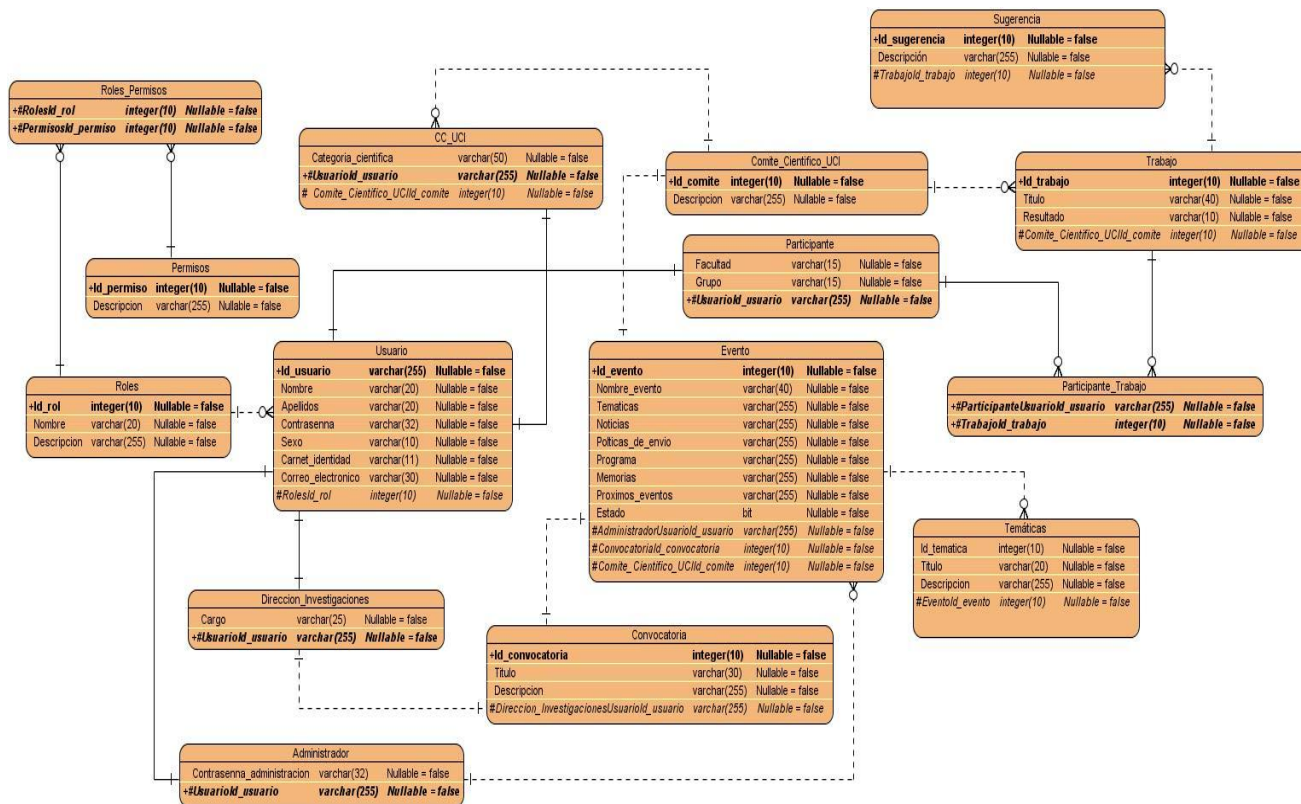


Figura 13. Diagrama Entidad-Relación de la Base de datos

### 3.4 Patrones empleados en el diseño

Durante el diseño de la solución propuesta se utilizan diferentes patrones con el objetivo de desarrollar una aplicación de alta calidad, promover la reutilización y agilizar el proceso de desarrollo de software.

#### 3.4.1 Patrón de arquitectura

Spring está basado en un patrón de diseño web conocido como arquitectura Modelo Vista Controlador (MVC), que está formado por tres niveles:

- ✓ **El Modelo:** Es el objeto que representa los datos del programa, maneja los datos y controla todas sus transformaciones.
- ✓ **La Vista:** Es el objeto que maneja la presentación visual de los datos representados por el Modelo, mostrando los datos al usuario.

✓ **El Controlador:** Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

La separación de vistas, controladores y modelos de este estilo arquitectónico permite la corrección en un solo lugar y no en varios como sucedería si tuviésemos una mezcla de presentación e implementación de la lógica del negocio, agregar nuevas formas de recolectar las ordenes del usuario, modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología, simplificar y reducir el tiempo necesario para labores de mantenimiento, posibilita además que las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación. (39)

### 3.4.2 Patrones de diseño

“Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientados a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan”. (40)

#### Patrones de Asignación de Responsabilidades (GRASP)

- ✓ Experto: se encarga de asignar la responsabilidad al experto en la información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, lo que provee un bajo nivel de acoplamiento. Promueve clases sencillas y cohesivas que son más fáciles de mantener y comprender.
- ✓ Creador: consiste en asignar a un objeto la responsabilidad de crear otro Objeto. Un objeto es responsable de crear una nueva instancia de alguna clase si: agrega o contiene objetos de ella, registra las instancias de sus objetos o tiene los datos de inicialización que serán enviados a ella cuando el objeto sea creado.
- ✓ Bajo acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Este patrón soluciona el inconveniente de dar soporte a una escasa dependencia y a un aumento de la reutilización.
- ✓ Alta cohesión: la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme. Fomenta la reutilización, mejorando la claridad y facilidad del diseño.



- ✓ Controlador: el patrón ofrece una guía para tomar decisiones sobre los eventos de entrada, asignando la responsabilidad del manejo de mensajes de los eventos del sistema a una clase controladora, ya que los elementos de interfaz y sus controladores de eventos, no deben ser responsables de controlar los eventos del sistema (41)

### 3.5 Conclusiones

Con el análisis de los casos de uso identificados preliminarmente se evidenció el flujo de las acciones entre las clases del análisis para cada funcionalidad descrita, lo que facilitó la construcción de un diseño que permitió modelar toda la aplicación web, incluyendo los objetos, que seguidamente se identificaron como clases persistentes en la actividad Diseño de la base de datos, con la realización de los artefactos generados durante el capítulo se pudo conformar la propuesta de solución de la presente investigación.

### Capítulo 4. Validación de la solución propuesta

#### 4.1 Introducción

En el presente capítulo se realiza la validación de la solución propuesta, mediante el empleo de un conjunto de métricas de diseño orientado a objetos que proporcionan una medida de la complejidad y calidad del diseño realizado.

#### 4.2 Métricas utilizadas para validar el modelo de diseño propuesto

La medición es fundamental para cualquier disciplina de ingeniería, y la ingeniería del software no es una excepción. La medición permite tener una visión más profunda, proporcionando un mecanismo para la evaluación objetiva. (42)

Seguidamente se identifican las métricas de diseño orientado a objetos aplicadas para evaluar la solución propuesta:

##### 4.2.1 Tamaño de clase (TOC)

El tamaño general de una clase puede medirse determinando las siguientes medidas:

- ✓ El total de operaciones (operaciones tanto heredadas como privadas de la instancia), que se encapsulan dentro de la clase.
- ✓ El número de atributos (atributos tanto heredados como privados de la instancia), encapsulados por la clase.

Tamaño de clase	
<b>Descripción:</b>	Está dada por el número de métodos asignados a una clase.
<b>Atributos que afecta:</b>	<b>Modo en que lo afecta:</b>
Responsabilidad	El aumento del TC provoca un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación.	El aumento del TC provoca un aumento de la complejidad de implementación.

## Capítulo 4. Validación de la solución propuesta

Reutilización	El aumento del TC provoca una disminución en el grado de reutilización de la clase.
---------------	---

Tabla 7. Métrica tamaño de clase (TC)

Atributo	Categoría	Criterio
Responsabilidad	Baja	$\leq$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$> 2^*$ Prom

Complejidad de implementación	Baja	$\leq$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$> 2^*$ Prom

Reutilización	Baja	$> 2^*$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$\leq$ Prom

Tabla 8. Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica TC

### Resultados del instrumento de evaluación de la métrica (TC)

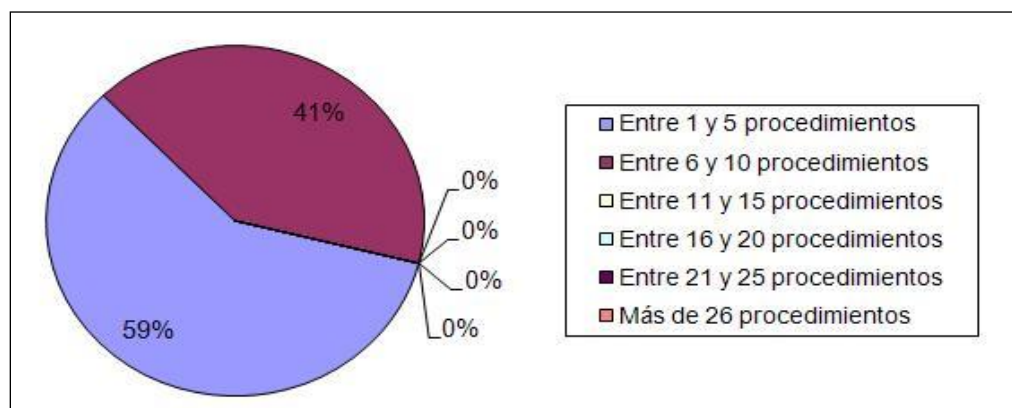


Figura 14. Representación en % de los resultados obtenidos de la métrica TC, agrupados en los intervalos definidos



Figura 15. Representación del resultado de la métrica TC en el atributo Responsabilidad

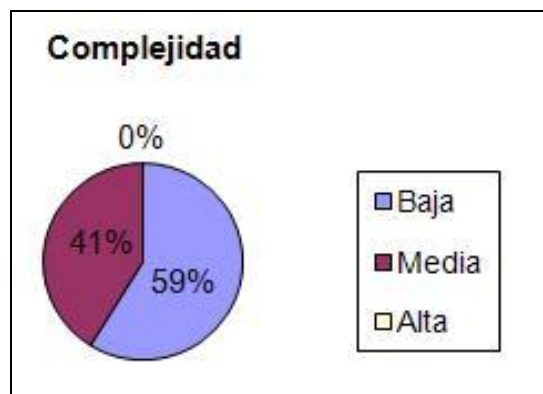


Figura 16. Representación del resultado de la métrica TC en el atributo Complejidad de Implementación



Figura 17. Representación del resultado de la métrica TC en el atributo Reutilización

## Capítulo 4. Validación de la solución propuesta

El análisis de los resultados obtenidos luego de la aplicación de la métrica TOC arroja que para un total de 17 clases evaluadas se realizaron 92 operaciones, obteniéndose un promedio de operaciones por clases de 5,41. La mayoría de las clases analizadas son de tamaño pequeño, este resultado es positivo e implica una alta reutilización de las clases, baja responsabilidad de las mismas, así como baja complejidad de las pruebas, demostrando que la propuesta de diseño posee una calidad aceptable.

### 4.2.2 Métrica Relaciones entre Clases (RC)

Tamaño de clase	
<b>Descripción:</b>	Está dada por el número de relaciones de uso de una clase con otras.
<b>Atributos que afecta:</b>	<b>Modo en que lo afecta:</b>
Acoplamiento	El aumento del RC provoca un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	El aumento del RC provoca un aumento de la complejidad del mantenimiento de la clase.
Reutilización	El aumento del RC provoca una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	El aumento del RC provoca un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 9. Métrica Relación entre clases (RC)

### Resultados del instrumento de evaluación de la métrica Relaciones entre clases (RC)

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	> 2

## Capítulo 4. Validación de la solución propuesta

Complejidad Mantenimiento	Baja	$\leq$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$> 2^*$ Prom

Reutilización	Baja	$> 2^*$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$\leq$ Prom

Cantidad de Pruebas	Baja	$\leq$ Prom
	Media	Entre Prom y $2^*$ Prom
	Alta	$> 2^*$ Prom

Tabla 10. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC

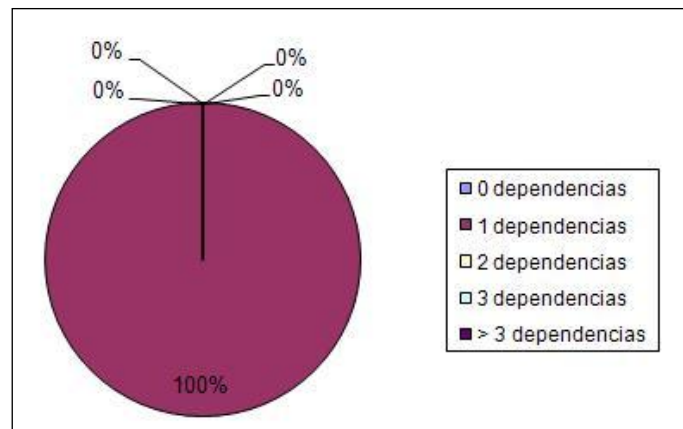


Figura 18. Representación en % de los resultados obtenidos de la métrica RC, agrupados en los intervalos definidos



Figura 19. Representación del resultado de la métrica RC en el atributo Acoplamiento

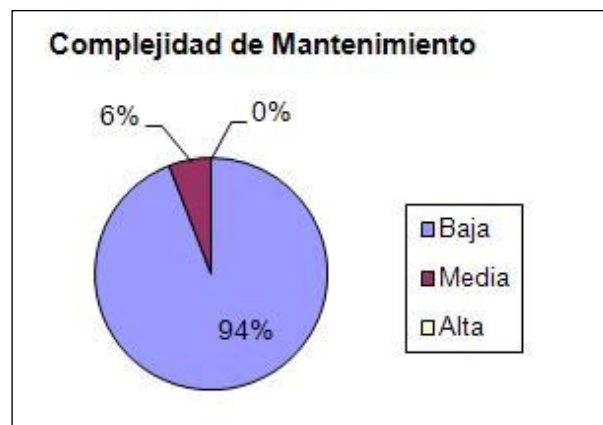


Figura 20. Representación del resultado de la métrica RC en el atributo Complejidad de Mantenimiento

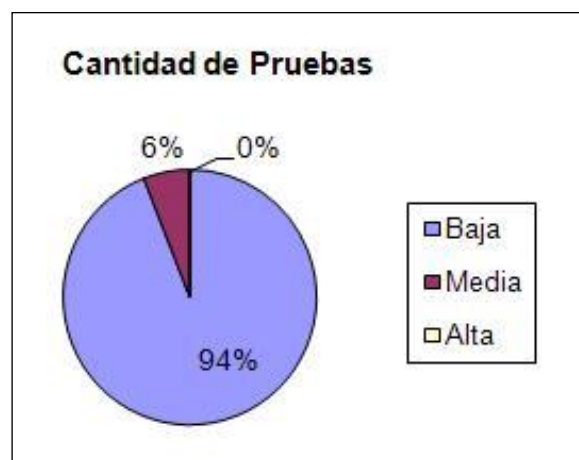


Figura 21. Representación del resultado de la métrica RC en el atributo Cantidad de Pruebas



Figura 22. Representación del resultado de la métrica RC en el atributo en el atributo Reutilización

Durante la evaluación de la métrica RC los resultados obtenidos demostraron que el 100% de las clases poseen una dependencia entre estas, para un promedio de una asociación de uso, lo que indica que los atributos se comportan satisfactoriamente, ya que en el 100% de las clases existe un acoplamiento mínimo y una alta reutilización, la complejidad de mantenimiento y la cantidad de pruebas se comportan favorablemente para un 94% de afectación en las clases.

### 4.3 Conclusiones

Durante el capítulo se aplicaron métricas para evaluar la calidad del diseño realizado y finalmente dar cumplimiento al objetivo planteado en la investigación. Se pudo constatar que el diseño propuesto no presenta una alta complejidad, permitiendo que las pruebas no sean complejas. Se puede concluir que el diseño obtenido posee una calidad aceptable, permitiendo una correcta implementación en etapas posteriores.



## Conclusiones generales

---

### Conclusiones generales

A partir del análisis de los procesos llevados a cabo por la DI y las aplicaciones vinculadas a la gestión de información de eventos científicos, se demostró la necesidad e importancia de realizar el análisis y diseño de una aplicación web que permita realizar la gestión de información de los eventos que se realizan en la UCI, de forma eficiente y controlada.

De acuerdo con el objetivo general trazado en la investigación, se obtuvo el diseño de la aplicación web para la gestión de información de eventos científicos, adaptándose a las necesidades actuales de la DI.

Se validó la solución propuesta mediante el empleo de métricas de diseño para sistemas orientado a objetos, lo cual demostró la solidez del Modelo diseño presentado, el cual servirá de guía para la posterior implementación del sistema.

# Recomendaciones

---

## Recomendaciones

- ✓ Realizar la implementación de la solución propuesta.
- ✓ Realizar un estudio de posibles funcionalidades que le aporten a la aplicación valor agregado.

### Bibliografía

1. Pérez Labrada., Lic. Magdiel. Sistema automatizado de Gestión de Información para el trabajo por Proyectos en el IDICT. *PONENCIA CONGRESO INTERNACIONAL INFO 2002*.
2. Scribd. [En línea] [Citado el: 12 de diciembre de 2010.] <http://www.scribd.com/doc/14686867/Organizacion-de-eventos>.
3. mailxmail.com. *mailxmail.com*. [En línea] [Citado el: 5 de enero de 2011.] <http://www.mailxmail.com/curso-php-mysql-aplicaciones-web-1/servidor-web-que-es..>
4. SEPG Latinoamérica 2010 . *SEPG Latinoamérica 2010* . [En línea] [Citado el: 10 de enero de 2011.] <http://www.esi.es/SEPGLA>.
5. Semana Tecnológica. *Semana Tecnológica*. [En línea] [Citado el: 12 de febrero de 2011.] <http://semanatecnologica.fordes.co.cu>.
6. Convención y Feria Internacional Informática 2011. [En línea] [Citado el: 11 de febrero de 2011.] <http://www.informaticahabana.cu>.
7. Guará Faure, Danay y Verdecia Rondón, Adriana. Aplicación web para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas. 2010.
8. Rojas Santiesteban, Nidia y Yordanka, Hechavarría Melo. Aplicación Web para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas. 2009.
9. Carrillo Pérez, Isaías , Pérez González, Rodrigo y Rodríguez Martín, Aureliano David. *Metodología de Desarrollo de Software*.
10. eumed.net. *eumed.net*. [En línea] [Citado el: 24 de enero de 2011.] <http://www.eumed.net/libros/2009c/584/Descripcion%20de%20las%20herramientas%20a%20utilizar%20para%20aplicar%20RUP%20y%20Scrum.htm>.
11. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software.
12. Mendoza Sanchez, María A. Metodologías De Desarrollo De Software.
13. LetelierTorres, Patricio y Penadés, Maria Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
14. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El lenguaje Unificado de Modelado.Manual de Referencia.

## Bibliografía

---

15. Free Download Manager. *Free Download Manager*. [En línea] [Citado el: 17 de enero de 2011.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
16. Visual Paradigm. [En línea] [Citado el: 25 de enero de 2011.] <http://www.visual-paradigm.com/>.
17. eumed.net. *eumed.net*. [En línea] [Citado el: 18 de 1 de 2011.] <http://www.eumed.net/libros/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>.
18. EcuRed. [En línea] [Citado el: 2 de febrero de 2011.] <http://www.ecured.cu/index.php/Ciente-Servidor>.
19. González Martín, Óscar y Ruiz González, Francisco. Arquitecturas de Sistemas de Bases de Datos.
20. Cabrera Rosique, Juan Pablo y Valera Fdez., Jose Fernando. Defenición de: HTML, CSS, PHP, SERVIDOR APACHE, SERVIDOR TOMCAT.
21. Gil, Fidel, Albrigo, Javier y Do Rosario, Javier. SISTEMAS DE GESTIÓN DE BASE DE DATOS SGBD / DBMS. Valencia : s.n., 2005.
22. PostgreSQL-es. [En línea] [Citado el: 25 de enero de 2011.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
23. Denzer, Patricio. PostgreSQL.
24. MySQL. [En línea] [Citado el: 26 de enero de 2011.] <http://www.mysql.com/documentation/index.html>.
25. slideshare. *slideshare*. [En línea] [Citado el: 5 de 2 de 2011.] <http://www.slideshare.net/triana25/motor-de-bases-de-datos-oracle-1503795>.
26. Rodas Hinostrroza, Raul. Características de PHP.
27. EcuRed. *EcuRed*. [En línea] [Citado el: 6 de febrero de 2011.] [http://www.ecured.cu/index.php/Lenguaje\\_de\\_programaci%C3%B3n\\_Java](http://www.ecured.cu/index.php/Lenguaje_de_programaci%C3%B3n_Java).
28. Java Team, Conozca más sobre la tecnología Java. [En línea] 2009. [Citado el: 9 de febrero de 2011.] <http://www.java.com/es/about/>.
29. joomla-chile.cl. *joomla-chile.cl*. [En línea] [Citado el: 10 de marzo de 2011.] <http://www.joomla-chile.cl/unidad-i/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.html>.
30. EcuRed. *EcuRed*. [En línea] [Citado el: 11 de marzo de 2011.] <http://www.ecured.cu/index.php/NetBeans>.

## Bibliografía

---

31. Scribd. *Scribd*. [En línea] [Citado el: 5 de marzo de 2011.] <http://www.scribd.com/doc/6615294/NetBeans>.
32. EcuRed. [En línea] [Citado el: 20 de febrero de 2011.] [http://www.ecured.cu/index.php/Eclipse,\\_entorno\\_de\\_desarrollo\\_integrado](http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado).
33. Gutiérrez, Javier J. ¿Qué es un framework web?
34. EcuRed. [En línea] [Citado el: 1 de marzo de 2011.] <http://www.ecured.cu/index.php/Spring>.
35. Johnson, Rod y Holler, Juergen. *Expert One-on-One J2EE Design and Development*.
36. EcuRed. *EcuRed*. [En línea] [Citado el: 3 de marzo de 2011.] <http://www.ecured.cu/index.php/Hibernate>.
37. Ingeniería de software 2. Conferencia # 1. *Entorno virtual de aprendizaje*. [En línea] [Citado el: 4 de marzo de 2011.] <http://eva.uci.cu>.
38. Un Método para el Diseño de la Base de Datos a partir del Modelo Orientado a Objetos. *E-journal*. [En línea] [Citado el: 5 de marzo de 2011.] <http://www.ejournal.unam.mx/cys/vol07-04/CYS07402.pdf>.
39. Ecured. *Ecured*. [En línea] [Citado el: 21 de marzo de 2011.] [http://www.ecured.cu/index.php/Patr%C3%B3n\\_M\\_Ventajas\\_de\\_utilizar\\_MVC](http://www.ecured.cu/index.php/Patr%C3%B3n_M_Ventajas_de_utilizar_MVC).
40. Gracia, J. Patrones de diseño. *IngenieroSoftware*. [En línea] [Citado el: 12 de enero de 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
41. Larman, Craig. UML y PATRONES. Introducción al análisis y diseño orientado a objetos.
42. Pressman, Roger S. 2005. Ingeniería del Software: Un enfoque práctico. [En línea] [Citado el: 9 de marzo de 2011.] <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.

### Glosario de Términos

**API** (*Applications Programming Interface*): es una serie de funciones que están disponibles para realizar programas para un cierto entorno.

**Aplicación**: programa informático que proporciona servicios de alto nivel al usuario, generalmente utilizando otros programas más básicos que se sitúan por debajo.

**BSD** (*Berkeley Software Distribution*): es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público.

**CASE** (*ComputerAided Software Engineering*): son aplicaciones informáticas para aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

**CC**: son las clases controladoras que se encargan de dirigir y controlar el funcionamiento de una petición, decidiendo quien procesa y quien muestra.

**CDDL** (*Common Development and Distribution License*): es una licencia de código abierto y libre.

**CE**: son las clases entidades contienen los atributos, según el tópico.

**CI**: es la clase interfaz, estereotipo para identificar las clases vistas.

**CMMI** (*Capability Maturity Model Integration*): es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

**DOM** (*Document Object Model*): es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

**Hardware**: corresponde a todas las partes físicas y tangibles de una computadora.

## Glosario de términos

---

**HQL** (*Hibernate Query Language*): es un lenguaje de consulta de datos.

**HTML** (*Hypertext Markup Language*): lenguaje de marcado para hipertextos, utilizado para la confección de páginas web.

**Http** (*Hypertext Transfer Protocol*): protocolo de transferencia de hipertextos, para el intercambio de información confeccionada con el lenguaje HTML.

**IDE** (*Integrated Development Environment*): es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

**ISO** (*International Organization for Standardization*): es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

**ITIL** (*Information Technology Infrastructure Library*): es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general.

**JavaScript**: es un lenguaje basado en objetos, utilizado para acceder a objetos en aplicaciones, se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

**JSP** (*Java Server Pages*): tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo.

**JUnit**: es un conjunto de clases que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.

## Glosario de términos

---

**Multiplataforma:** es un término usado para referirse a los programas, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

**ORM** (*Object-Relational Mapping*): es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

**PSP** (*Personal Software Process*): es un conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software, en tareas de desarrollo y mantenimiento de sistemas.

**Servlets:** son componentes de la parte del servidor de Java EE, encargados de generar respuestas a las peticiones recibidas de los clientes.

**SQL** (*Structured Query Language*): es un lenguaje declarativo de acceso a bases de datos relacionales.

**Software:** se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas.

**TSP** (*Team Software Process*): es un modelo para dirigir el trabajo de mejora y desarrollo de software además de establecer un entorno donde el trabajo efectivo de equipo sea normal y natural.

**Unix:** es un sistema operativo portable, multitarea y multiusuario.

**Web:** sistema para presentar información en internet basado en hipertexto.

**XML** (*Extensible Markup Language*): es un metalenguaje que nos permite definir lenguajes de marcado adecuados a usos determinados. Es un estándar internacionalmente reconocido y no pertenece a ninguna compañía por lo que su utilización es libre.