



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“IMPLEMENTACIÓN DEL PORTAL WEB DE ADMINISTRACIÓN DE  
EVENTOS PARA GRUPOS DE INVESTIGACIÓN”**

**INFORME DE PROYECTO DE GRADUACIÓN**

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES  
ESPECIALIZACIÓN EN SISTEMAS MULTIMEDIA**

Autores:

**GUILLERMO OMAR PIZARRO VÁSQUEZ**

**RAFAEL EDUARDO RIVADENEIRA CAMPODÓNICO**

**GUAYAQUIL – ECUADOR**  
Año: 2009

## **DEDICATORIA**

Al Señor quien me ha ayudado e inspirado a tomar buenas decisiones a lo largo de mi vida.

A mis padres, quienes me han facilitado en todo, en la medida de sus posibilidades, para que pueda llegar a este momento y cumplir con uno de mis sueños.

**Guillermo Omar Pizarro Vásquez**

Le dedico este proyecto a Dios, por estar siempre a mi lado dándome las fuerzas necesarias para salir adelante todos los días.

A mis padres por todo lo que me han dado en la vida, por su apoyo y confianza a lo largo de la carrera.

A mi abuelita por sus sabios consejos que me han sabido guiar.

A mi hermano Alex por esa confianza, positivismo y orgullo de terminar la carrera.

A mi prometida Karen, por estar siempre a mi lado.

**Rafael Eduardo Rivadeneira Campodónico**

## **AGRADECIMIENTO**

Agradecemos a Dios por guiarnos todos los días y a nuestras familias por apoyarnos a lo largo de nuestra carrera. A la Facultad de Ingeniería en Electricidad y Computación de la ESPOL por prepararnos a ser futuros profesionales.

Un agradecimiento especial a las Ingenieras Verónica Macías y Mónica Villavicencio por habernos guiado con sus conocimientos y consejos para hacer posible la culminación de nuestro Proyecto de Grado.

# TRIBUNAL DE SUSTENTACIÓN

---

Ing. Sergio Flores  
DECANO DE LA FIEC  
PRESIDENTE

---

Ing. Mónica Villavicencio  
DIRECTORA DE PROYECTO  
DE GRADUACIÓN

---

Ing. Ana Tapia  
MIEMBRO DEL TRIBUNAL

---

Ing. Cristina Abad  
MIEMBRO DEL TRIBUNAL

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

---

Guillermo O. Pizarro Vásquez

---

Rafael E. Rivadeneira Campodónico

## RESUMEN

El presente trabajo tiene el propósito de implementar una Aplicación Web utilizando conjuntamente las metodologías **MDA** (Model Driven Architecture) y **MERODE** (Model - driven, Existence - dependency Relation Object - oriented DEvelopment), para facilitar la Administración de un Evento en un Grupo de Investigación.

Es importante mencionar que anteriormente se ha implementado, como Proyecto de Tesis, un producto llamado AppVlir8<sup>1</sup>, el mismo que es también un Portal Web de Administración de eventos. Nuestra idea es retomar el diseño original de este software y modificarlo, corrigiendo defectos y agregando mejoras que se han identificado durante el tiempo en que el software ha estado en producción.

---

1 <http://www.vlir8.espol.edu.ec>

En el primer capítulo se señalan los antecedentes para retomar el Proyecto AppVlir8, la justificación del desarrollo de los módulos siguiendo la elección de la metodología **MERODE** que ayuda a elaborar un diseño independiente del dominio, método que es completamente compatible con **MDA** y los objetivos que se esperan alcanzar.

En el segundo capítulo se redactan los fundamentos teóricos en los que se basa este Proyecto de Grado, justificando el uso de cada elemento incluido en la arquitectura del mismo.

En el tercer capítulo se documentan los requerimientos funcionales y no funcionales levantados para mejorar el diseño del sistema anterior. Además de la especificación del PIM y del PSM, previos a la fase de la transformación al código.

En el cuarto capítulo se mencionan los cambios realizados en la arquitectura del sistema anterior y las novedades en el actual, para poder satisfacer algunos requerimientos claves del usuario del sistema.

En el quinto capítulo se realiza un análisis con respecto a la implementación realizada y las pruebas que se realizaron para la entrega de un producto de calidad.

Finalmente se exponen las conclusiones y recomendaciones del Proyecto de Grado.



# ÍNDICE GENERAL

RESUMEN.....	6
ÍNDICE GENERAL .....	9
ABREVIATURAS .....	11
ÍNDICE DE FIGURAS Y GRÁFICOS .....	12
ÍNDICE DE TABLAS .....	14
CAPÍTULO I .....	15
1. ANTECEDENTES Y JUSTIFICACIÓN .....	15
1.1. Antecedentes .....	15
1.2. Justificación.....	15
1.3. Objetivos.....	16
CAPÍTULO II .....	18
2. FUNDAMENTOS TEÓRICOS.....	18
2.1. Descripción General de MDA y MERODE .....	18
2.2. Tecnologías Open Source.....	22
2.2.1. Uso de la Arquitectura J2EE.....	22
2.2.2. Librerías en el Servidor: JSTL, JSON y Velocity.....	24
2.2.3. Librerías en el Cliente: Dojo y YUI. ....	25
2.2.4. Base de Datos: MySQL y HSQLDB. ....	26
2.2.5. Uso de la Herramienta CASE: StarUML.....	27
CAPÍTULO III.....	28
3. ANÁLISIS DEL SISTEMA .....	28
3.1. Requerimientos Funcionales .....	28
3.2. Requerimientos no Funcionales .....	30
3.3. Especificación del Modelo Independiente de la Plataforma (PIM) usando técnicas de MERODE.....	32
3.3.1. El Gráfico de la Dependencia de Existencia (EDG) .....	33

3.3.2. La Tabla de Eventos de Objetos (OET) .....	34
3.3.3. Máquina de Estados Finitos Relevantes (FSM) .....	35
3.4. Definición del Modelo Específico de la Plataforma (PSM) .....	37
3.5. Definición de Roles en los Módulos .....	37
<b>CAPÍTULO IV</b> .....	<b>39</b>
<b>4. DISEÑO DEL SISTEMA</b> .....	<b>39</b>
4.1. Diseño de la Arquitectura del Sistema .....	39
4.1.1. Diseño Arquitectónico .....	41
4.1.2. Módulo del Sistema .....	42
4.1.3. Arquitectura Basada en Patrones de Diseño .....	43
4.2. Diagramas de Clases del Sistema .....	44
4.3. Modelo Lógico de la Base de Datos .....	46
<b>CAPÍTULO V</b> .....	<b>48</b>
<b>5. IMPLEMENTACIÓN Y PRUEBAS</b> .....	<b>48</b>
5.1. Capas del Sistema y Comunicación entre Capas .....	48
5.1.1. Implementación Basada en MERODE y MDA .....	48
5.2. Plan de Pruebas .....	52
5.3. Resultados de las Pruebas y Métricas tomadas .....	57
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>61</b>
<b>APÉNDICE A: Gráfico de Dependencia Existencia</b> .....	<b>64</b>
<b>APÉNDICE B: Tabla de Objetos-Eventos</b> .....	<b>69</b>
<b>APÉNDICE C: Máquinas de Estado Finitos</b> .....	<b>71</b>
<b>APÉNDICE D: Diagramas de Clases por Módulos</b> .....	<b>73</b>
<b>APÉNDICE E: Diagrama del Modelo Lógico de la Base de Datos por Módulos</b> .....	<b>77</b>
<b>APÉNDICE F: Pruebas Unitarias</b> .....	<b>81</b>
<b>REFERENCIAS</b> .....	<b>86</b>

## ABREVIATURAS

Las abreviaturas presentadas en la siguiente tesis son las siguientes:

<b>MDA:</b>	Model Driven Architecture.
<b>MERODE:</b>	Model - driven, Existence - dependency Relation Object - oriented DEvelopment.
<b>COCOMO:</b>	COst COnstructive MOdel.
<b>OMG:</b>	Object Management Group
<b>AJAX:</b>	Asynchronous JavaScript And XML
<b>PIM:</b>	Moledo independiente de la plataforma
<b>PSM:</b>	Modelos especificos de la plataforma
<b>PSI:</b>	Modelos especificos de implementación
<b>EDG:</b>	Existence Dependency Graph
<b>FSM:</b>	Finite State Machines
<b>JSP:</b>	Java Server Pages
<b>OET:</b>	Object-Event Table
<b>OMT:</b>	Object Modelling Technique
<b>JSON:</b>	JavaScript Object Notation
<b>XML:</b>	Extensible Markup Language

## ÍNDICE DE FIGURAS Y GRÁFICOS

Figura 2.1 Arquitectura Dirigida por Modelos.....	19
Figura 2.2 Arquitectura de MERODE.....	21
Figura 2.3 Arquitectura de J2EE.....	23
Gráfico 4.1. Diseño de la Arquitectura del Sistema de Administración de Eventos y Grupos de Investigación (WebSAEGI). ....	40
Gráfico 4.2. Diseño de la Arquitectura WebSAE. ....	41
Gráfico 4.3 Módulos del Sistema.....	43
Gráfico 4.4 Diagrama de Clases - Módulo de Convocatoria y Evaluación de Artículo (MCE) .....	45
Gráfico 4.5 ER – Módulo de Convocatoria y Evaluación de Artículos (MCE) .....	47
Figura 5.1 Regla de transformación de la capa del dominio.....	50
Figura 5.2 Regla de transformación de la capa de control de eventos.....	51
Gráfico 5.1 Resultados de las pruebas unitarias. ....	58
Gráfico 5.2 Pruebas unitarias fallidas. ....	58
Gráfico 5.3 Estimación de tiempos realizada en COCOMO. ....	59
Gráfico 5.4 Tiempos reales de cada módulo.....	59
Gráfico 5.5 Porcentajes de error con respecto a lo estimado con el tiempo real.....	60
Gráfico A.1 EDG – Sistema Web para la Administración de Eventos Científicos (WebSAE).....	64
Gráfico A.2 EDG - Módulo de Administración Central (MAC).....	65
Gráfico A.3 EDG - Módulo de Suscripción de Usuarios (MSU).....	66
Gráfico A.4 EDG - Módulo de Administración de Eventos (MAE).....	67
Gráfico A.5 EDG - Módulo de Convocatoria y Evaluación de Artículos (MCE) .....	68

Gráfico C.1 FSM del objeto AE_Registro.....	71
Gráfico C.2 FSM del objeto CE_Articulo .....	72
Gráfico D.1 Diagrama de Clases - Módulo de Administración Central (MAC) .....	73
Gráfico D.2 Diagrama de Clases - Módulo de Suscripción de Usuarios (MSU) .....	74
Gráfico D.3 Diagrama de Clases - Módulo de Administración de Eventos (MAE) (Parte I).....	75
Gráfico D.3 Diagrama de Clases - Módulo de Administración de Eventos (MAE) (Parte II).....	76
Gráfico E.1 ER – Módulo de Administración Central (MAC).....	77
Gráfico E.2 ER – Módulo de Suscripción de Usuarios (MSU).....	78
Gráfico E.3 ER – Módulo de Administración de Eventos (MAE) (Parte I) .....	79
Gráfico E.3 ER – Módulo de Administración de Eventos (MAE) (Parte II) .....	80

## ÍNDICE DE TABLAS

Tabla 3.1. Definición de Roles en los Módulos .....	38
Tabla 5.1. Prueba Unitaria N°: 01 .....	52
Tabla 5.2. Prueba Unitaria N°: 02 .....	53
Tabla N.3. Prueba Unitaria N°: 03.....	54
Tabla N.4. Prueba Unitaria N°: 04.....	55
Tabla N.5. Prueba Unitaria N°: 05.....	56
Tabla B.1 OET – Objeto AE_Evento.....	70
Tabla F.1. Prueba Unitaria N°: 01 .....	81
Tabla F.2. Prueba Unitaria N°: 02 .....	81
Tabla F.3. Prueba Unitaria N°: 03 .....	83
Tabla F.4. Prueba Unitaria N°: 04 .....	84
Tabla F.5. Prueba Unitaria N°: 05.....	85

# CAPÍTULO I

## 1. ANTECEDENTES Y JUSTIFICACIÓN

### 1.1. Antecedentes

Debido a la importancia que tiene el desarrollo de software en el ámbito de los negocios, es necesaria la aplicación de métodos, técnicas y herramientas que nos ayuden a obtener productos de calidad y uno de los aspectos importantes a considerar es la flexibilidad, debido a que tanto en los negocios como en las tecnologías se perciben cambios y avances, que en el campo de la Ingeniería de Software plantean el reto de crear aplicaciones que se puedan adaptar de forma simple y eficiente.

Para poder implementar requerimientos emergentes se vuelve necesario utilizar alguna metodología, de tal manera que si existen cambios en el dominio del negocio, esto no afecte de manera crítica a sus actividades; así mismo, hay que considerar las nuevas tecnologías que puedan servir para mejorar algún servicio que se ha implementado.

### 1.2. Justificación

Según investigaciones realizadas [1], una forma de lograr una mayor flexibilidad y un alto nivel de mantenibilidad en las aplicaciones de

software se da cuando se separa el “modelo del dominio” (modelo del negocio o modelo conceptual) de las “diversas tecnologías” que surgen a través de los tiempos.

Dos metodologías que se apoyan en este concepto de separar el modelo del negocio de la tecnología asociada son **MDA** y **MERODE**. MDA es una propuesta del OMG para el desarrollo de software desde el diseño de modelos; proporciona una solución para los cambios de negocio y de tecnología, permitiendo construir aplicaciones independientes de la plataforma. MERODE es una metodología de análisis orientado a objetos basado en el paradigma del modelamiento del dominio.

### **1.3. Objetivos**

Al desarrollar este Proyecto de Grado se quiere cumplir con los siguientes objetivos:

- Dar a conocer las ventajas que conlleva el uso de MDA y MERODE en conjunto
- Analizar, diseñar, desarrollar e implementar un portal Web de administración de eventos para grupos de investigación. Este portal Web permitirá la organización de eventos, publicaciones de



grupos de investigación, suscripción de usuarios y evaluaciones de artículos.

# CAPÍTULO II

## 2. FUNDAMENTOS TEÓRICOS

### 2.1. Descripción General de MDA y MERODE

#### **MDA, una arquitectura dirigida por modelos. -**

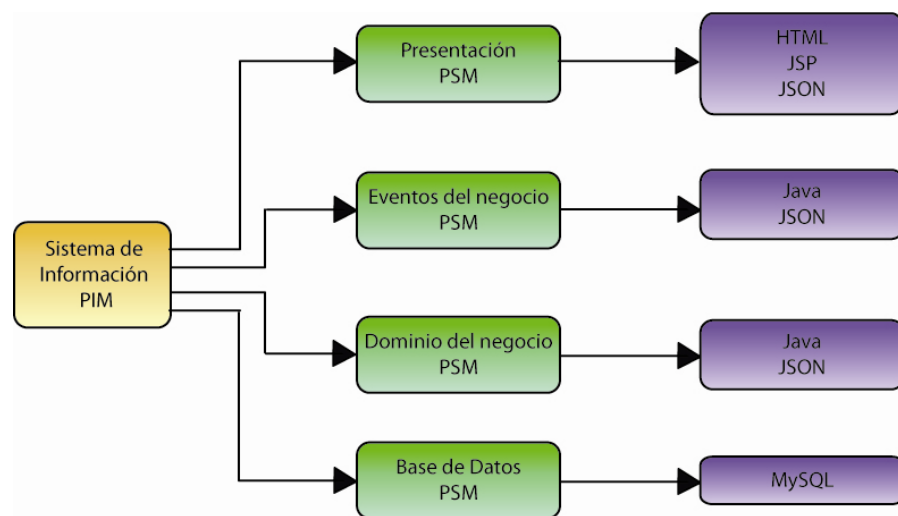
En el año 2000, el OMG publicó un artículo [2] en el que se presentaba una metodología donde todo se centraba en modelos. Su principal objetivo era diseñar software independiente de la plataforma, de tal manera que si el modelo de negocio exigía pasar -el software en producción- a otra tecnología, simplemente se genere el código respectivo a partir del modelo.

MDA define modelos (Figura 2.1) en diversos niveles de abstracción, que van desde el modelo del dominio al modelo de especificaciones tecnológicas, cada uno con alcance definido. A continuación se listan estos niveles:

1. Modelo independiente de la plataforma.
2. Modelos específicos de la plataforma.
3. Modelos específicos de implementación.

El PIM es una vista que representa la especificación de un dominio, es decir, su estructura, sus restricciones (pre-condiciones y pos-

condiciones), en fin, es tal la abstracción a la que se desea llegar, que podrá ser aplicada a diferentes tecnologías. El PSM es una vista específica de una tecnología definida. El PSI es el código del sistema a desarrollar, el cual puede ser generado en gran parte, mediante algún software.



**Figura 2.1 Arquitectura Dirigida por Modelos.**

**MERODE** una metodología de análisis orientada a objetos bajo el paradigma de modelamiento basado en el dominio.-

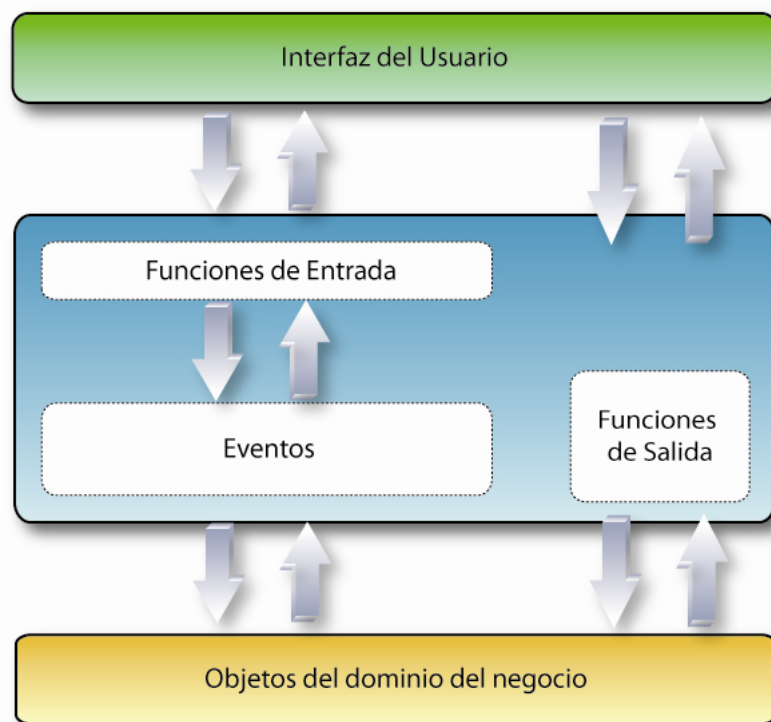
MERODE es una metodología, que al igual a **MDA**, se basa en el principio de que el modelamiento del dominio debe estar separado de las especificaciones tecnológicas.

En el análisis de la aplicación **MERODE** separa los objetos en capas: la capa interna, formada por los objetos propios del dominio, la capa media, que contiene los objetos del sistema de información; y la capa externa, los objetos de interfaz de usuario.

Los objetos de cada capa sólo pueden comunicarse entre los objetos de la misma capa o con los objetos de la capa inferior inmediata. El posible cambio en un objeto, sólo afecta a los objetos de las capas superiores. Por lo tanto, si colocamos los objetos que menos cambian en las capas inferiores, será menor el esfuerzo al realizar un cambio en la aplicación si se colocan los objetos que más cambian en las capas superiores [3].

**MERODE** nos brinda una arquitectura (Figura 2.2) provista de las especificaciones que un negocio necesita; es decir, los requerimientos del negocio en ***una capa para los objetos del dominio***; las especificaciones funcionales que provienen de la realización de procesos que implican la manipulación de datos en otra ***capa denominada funcional*** (que contiene: eventos, que a su vez son accedidos mediante funciones de entrada y que proporciona información consultando los objetos del dominio mediante una función de salida); por último tenemos la ***capa de interfaz del usuario*** (aquella a la que accede el usuario como tal).

Por último, **MERODE** nos ofrece la posibilidad de realizar un apropiado control de calidad, sobre el análisis de sistema. Esto lo proporcione, mediante la representación de un solo tipo de relación entre los objetos, denominada relación de dependencia-existencia; además de las técnicas necesarias para representar los aspectos estáticos y dinámicos del modelo del dominio y, lo más importante, un conjunto de reglas que permiten la verificación automática de la consistencia interna entre las diferentes vistas del modelo [4].



**Figura 2.2 Arquitectura de MERODE.**

## **2.2. Tecnologías Open Source**

### **2.2.1. Uso de la Arquitectura J2EE.**

Java 2 Enterprise Edition, es una plataforma de programación, parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de múltiples capas distribuidas, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones [5].

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, entre otros y define como coordinarlos. También configura algunas especificaciones únicas para componentes Java EE; estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificaciones de Portlets Java), Java Server Pages y varias tecnologías de servicios Web, soportados a través de intercambios de datos integrados basados en estándares abiertos y protocolos XML (eXtensible Markup Lenguaje). Esto permite al desarrollador crear aplicaciones empresariales. La aplicación empresarial o lógica de la empresa está dividida entre componentes dependiendo su función y al integrar una aplicación J2EE, dichos componentes son instalados en diferentes máquinas dependiendo del nivel del ambiente multicapa J2EE al cual pertenece.



**Figura 2.3 Arquitectura de J2EE.**

Como se mencionó en la sección anterior, MERODE distribuye los objetos del negocio en capas (capa interna, capa media y capa externa), lo cual permite que el software final tenga una mejor administración y mantenimiento. Por otro lado, J2EE es una arquitectura de diseño basada en la distribución de aplicaciones en capas (Figura 2.3), lo que permite un mejor manejo de la información que se está procesando, es por esto, que a más de tener MERODE distribuido en capas y J2EE diseñado para dispositivos para aplicativos en capas, esta integración de ambos, nace como un requerimiento del Sistema de utilizar una arquitectura de código abierto, que permita la portabilidad del sitio Web y su implementación, en cualquier sistema operativo y mantener activo el funcionamiento del portal sin mayores inconvenientes. Es por esto que la integración de ambos componentes se mantiene y es utilizada en la nueva versión del sitio Web.

### 2.2.2. Librerías en el Servidor: JSTL, JSON y Velocity.

**JSTL** (JSP Standard Tag Library) es un componente dentro de la especificación del Java 2 Enterprise Edition (J2EE) y es controlada por Sun Microsystems. JSTL no es más que un conjunto de librerías de etiquetas simples y estándares que encapsulan la funcionalidad principal que es usada comúnmente para escribir páginas JSP. Las etiquetas JSTL están organizadas en 4 librerías [6]:

- Core: comprende las funciones scripts básicas como loops, condicionales, y entrada/salida.
- Xml: comprende el procesamiento de XML
- Fmt comprende la internacionalización y formato de valores como de moneda y fechas.
- Sql: comprende el acceso a base de datos

**JSON** (JavaScript Object Notation) es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notificación literal de objetos de JavaScript que no requiere el uso de XML [7].

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un



analizador semántico de JSON. En JavaScript, JSON puede ser analizado trivialmente usando el procedimiento `eval()`, lo cual ha sido fundamental para la aceptación de JSON por parte de la comunidad de desarrolladores Ajax, debido a la ubicuidad de JavaScript en casi cualquier navegador Web.

**Velocity** es un sencillo, pero poderoso, motor de plantilla que se puede utilizar como generador de contenido dinámico como alternativa al estándar impuesto. En algunos casos proporciona mejores rendimientos que las páginas JSP, y ciertamente fuerza a los desarrolladores Web a pensar de forma diferente en la forma de separar el contenido del código [8].

### **2.2.3. Librerías en el Cliente: Dojo y YUI.**

**Dojo** es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la

habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos [9].

**YUI** (Yahoo User Interface) una serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones interactivas (RIA). Liberadas bajo licencia BSD por parte de la compañía Yahoo. Dichas bibliotecas son utilizadas para el desarrollo Web específicamente para ser usadas como la programación de aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX [10].

#### **2.2.4. Base de Datos: MySQL y HSQLDB.**

HSQLDB es una base de datos relacional escrita en Java. La gran utilidad de esta base es la portabilidad, ya que no es necesaria que sea instalada. Es la principal diferencia que tiene con MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 es una subsidiaria de Sun Microsystems y

ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual [11].

#### **2.2.5. Uso de la Herramienta CASE: StarUML.**

StarUML es una excelente aplicación UML en código abierto. Soporta UML 2.0 y MDA (Modelo de Arquitectura Dirigida). StarUML permite trabajar con todos los diagramas relacionados. Puede realizar generación de códigos para Java, C++ y C#, y documentos para todos los programas que componen el paquete de Microsoft Office, tales como: Word, Excel y PowerPoint. StarUML es fácilmente extensible a través de sus módulos, que utilizan una arquitectura de plug-ins basada en comandos, y de sus plantillas, que permiten hacer nuevos generadores de códigos. StarUML incluso importa desde Rational Rose [12].

## **CAPÍTULO III**

### **3. ANÁLISIS DEL SISTEMA**

#### **3.1. Requerimientos Funcionales**

De manera general, los nuevos requerimientos que se van a implementar son los siguientes:

- Añadir el módulo de pago en línea;
- Dar soporte para varios idiomas.

Los requerimientos que se van a modificar para adaptarlos a las necesidades de los diferentes usuarios, son los siguientes:

- Mostrar resúmenes de evaluaciones, con los siguientes datos: recomendación y exportación de los datos ingresados por los evaluadores.
- Manejar copias ocultas para emails de rechazo de artículos, es decir, se debe enviar un correo electrónico a la persona que presentó el artículo con una copia oculta al administrador del evento que rechaza el artículo, el remitente debe ser el email del evento.
- Mayor flexibilidad en el momento de la elaboración y modificación de las preguntas frecuentes.
- Agregar la posibilidad de subir un resumen previo, por parte del usuario participante del evento científico, a la versión final del artículo

científico que se va a evaluar, permitiendo además la evaluación de dicho resumen por parte del comité científico del evento.

- Permitir la modificación de un artículo sin tener que borrar y volverlo a subir, siempre que al artículo no haya finalizado la etapa de evaluación.
- Después de evaluado un artículo, se debe seguir mostrando, a los administradores del evento, los nombres de los evaluadores del artículo.
- El registro del usuario debe ser internacional y no sólo dirigido a nuestro país.
- Generar los siguientes reportes: trabajos por autores, evaluadores asignados por artículo, resúmenes de artículos clasificados con su respectiva recomendación y el estado de cada artículo. Exportar los reportes a los formatos: .doc/.pdf.
- Adjuntar la carta de sesión de derecho de los artículos aceptados.
- Permitir que el administrador pueda eliminar un artículo.
- Ocultar la contraseña de la lista de los usuarios.
- Manejar un correo electrónico de contacto para cada evento diferente.
- En los registros: permitir ingresar abonos cuando un usuario realice un pago en efectivo durante el periodo de pre-registros, El sistema debe notificar además la fecha máxima de pago permitida.

- Poder escoger el autor de un artículo en caso de que una persona alterna suba el artículo por el autor.
- Permitir que un evento particular tenga un comité organizador, quienes podrán evaluar los resúmenes y aprobar o rechazar la presentación preliminar de un trabajo en un evento particular.

Los errores detectados en la versión anterior, que se van a corregir son los siguientes:

- Un autor no debe poder evaluar su propio artículo.
- Los autores secundarios también deben constar en la lista de un artículo.
- No permitir el ingreso al sistema cuando se ha introducido una clave errónea después de haber salido recientemente de una sesión de acceso exitosa.

### **3.2. Requerimientos no Funcionales**

Los requerimientos no funcionales que se desea para el Portal son los siguientes:

- Manejo más apropiado de la usabilidad: estandarización de los íconos de la aplicación y mostrar errores específicos, no generales. (mejor retroalimentación);

- Utilizar MVC 2 en el diseño e implementación; puesto que este ofrece mayor flexibilidad al producto final;
- Reparar defectos encontrados, entre los cuales existen algunos que afectan al rendimiento de la Aplicación Web, tales como: la optimización de las consultas a la Base de Datos y el uso de scriptlets en las páginas JSP;
- Separar la parte de manejo de eventos para que pueda instalarse y ejecutarse independientemente de la parte de administración de grupos de investigación y sus áreas de trabajo.

Los requerimientos mínimos, en cuanto a hardware, son los siguientes:

- Capacidad de memoria en el servidor: para un buen desempeño del sistema, una memoria de 2 GB de RAM ó superior.
- Capacidad de disco duro en el servidor: se recomienda un disco duro de 80 GB como mínimo.
- Tipo de procesador en el servidor: Se recomienda un procesador Core 2 Duo 2.0GHz.
- Ancho de Banda para el servidor: Ancho de Banda recomendado 128 Mbits.
- Navegadores: Para un buen uso del sistema se recomienda Firefox 3.0 ó superior.

- Sistema Operativo en el Servidor: Microsoft Windows XP ó superior, o cualquier distribución de GNU/Linux, se recomienda Ubuntu.

### **3.3. Especificación del Modelo Independiente de la Plataforma (PIM) usando técnicas de MERODE**

**MERODE** nos permite la posibilidad de modelar el dominio de manera independiente de la plataforma que se desee implementar, en nuestro caso deseamos diagramar un **PIM** (Platform Independent Model) mediante un **EDG**, este fue diseñado mediante la herramienta llamada **MERMAID**<sup>2</sup>.

Es necesario mencionar que los diagramas a continuación representan un solo **PIM**; sin embargo se los ha dividido en varios módulos para mejorar su comprensión, la división en varios módulos es necesaria no sólo para una mejor concepción del diseño, si no para identificar las partes independientes que conforman el modelo.

---

<sup>2</sup> <http://merode.econ.kuleuven.ac.be/activate.aspx>



### **3.3.1. El Gráfico de la Dependencia de Existencia (EDG)**

A continuación se definen los módulos con sus respectivos Gráficos de la Dependencia de Existencia o EDG, cada uno de los gráficos se encuentra en los anexos.

#### **MÓDULO DE ADMINISTRACIÓN CENTRAL (MAC)**

**MAC** (Gráfico A.2) no forma parte del dominio en sí, pero su utilización es necesaria para manejar los diferentes roles que puede tener el usuario en la aplicación, y por ende las operaciones que ese usuario pueda realizar dependiendo del rol o roles que posea.

#### **MÓDULO DE SUSCRIPCIÓN DE USUARIOS (MSU)**

Es un módulo para la administración de Usuarios (Gráfico A.3), además maneja la suscripción de los usuarios visitantes del Portal.

#### **MÓDULO DE ADMINISTRACIÓN DE EVENTOS (MAE)**

En **MAE** (Gráfico A.4), se lleva el control de las suscripciones al Evento Científico, definición de precios y descuentos en intervalos de fechas para los registros, gestión de los materiales, conferencistas invitados y/o expositores, sub eventos, agenda general, etc.

## **MÓDULO DE CONVOCATORIA Y EVALUACIÓN DE ARTÍCULOS (MCE)**

En este módulo (Gráfico A.5) se genera la convocatoria para la presentación de los artículos, la asignación de evaluadores, aceptación ó rechazo de los artículos.

### **3.3.2. La Tabla de Eventos de Objetos (OET)**

Para presentar la Tabla de Eventos de Objetos<sup>3</sup>, se ha considerado separarla por objetos para que pueda ser apreciada de una mejor manera.

La tabla especifica los eventos (las filas) que afectan al objeto AE\_Evento (columna). En la tabla se respetan las reglas de propagación, regla del tipo de efecto y regla de contrato.

---

<sup>3</sup> Es una matriz que contiene una fila por cada tipo de evento y una columna por cada tipo de objeto.

En la regla de propagación se define que todos los tipos de eventos pertenecientes al alfabeto de un tipo de objeto dependiente son propagados a su tipo de objeto maestro, se detalla a continuación en el OET el tipo de proveniencia: propio (O: Owned), adquirido (A: Acquired), especializado (S: Specialised) o heredado (I: Inherited).

### **3.3.3. Máquina de Estados Finitos Relevantes (FSM)**

Se muestra a continuación los FSM de dos objetos principales: AE\_Registro y CE\_Articulo, los correspondientes gráficos se encuentran en el anexo.

En el objeto AE\_Registro, se muestran (Gráfico C.1) seis estados: ESPERA, APROBADO, PENDIENTE, RECHAZADO, REGISTRADO y CADUCADO. Se mantiene en ESPERA si el usuario que ha aplicado a un descuento en su registro se mantiene a la expectativa de que le aprueben el descuento, pasa a APROBADO si el registro ha sido autorizado por el administrador ó también puede ser RECHAZADO si el administrador demuestra que no aplica al descuento solicitado. Un registro está en PENDIENTE sólo si no ha aplicado a ningún descuento. Para estar en estado REGISTRADO, debe pagar o si se registró a un evento o sub evento gratuito. El registro que se encuentra en estado CADUCADO es porque no ha pagado el registro

o no se le ha aprobado el descuento al que aplicó; sin embargo, se da la posibilidad de que pase ha estado REGISTRADO, sólo si el usuario demuestra haber pagado en efectivo antes de la fecha prevista.

Para el objeto CE\_Articulo (Gráfico C.2) existen los estados: EVALUADO, PENDIENTE, EN CONSULTA, APROBADO, RECHAZADO y EN ESPERA. Cuando en un evento se tiene previsto revisar resúmenes previos (este es un paso opcional) antes de pasar al proceso de evaluación el artículo se mantiene en estado EN CONSULTA, una vez que los administradores del evento (miembros del comité) hayan decidido cuáles resúmenes aplican o no al proceso de revisión el artículo pasa al estado PENDIENTE, en este estado se le asigna evaluadores a un artículo, luego de que todos los evaluadores hayan revisado a un artículo éste pasa a estado EVALUADO, después los administradores decidirán si el artículo es APROBADO o RECHAZADO, según las evaluaciones emitidas por los revisores. Si el artículo es APROBADO, si los evaluadores han comentado que el artículo sea corregido, el administrador lo pasa al estado EN ESPERA hasta que sea corregido por el usuario.

### **3.4. Definición del Modelo Específico de la Plataforma (PSM)**

Luego de elaborar los correspondientes EDGs, OETs y diagramas de estados que conforman nuestro PIM, se procedió a continuar con la elaboración del PSM, el cual, fue diagramado mediante una herramienta Open Source, llamada StarUML<sup>4</sup>.

En este paso se diagraman en UML los módulos que fueron diseñados anteriormente siguiendo la metodología de MERODE (considerando las reglas de transformación en cuanto a las relaciones de existencia-dependencia entre los objetos).

### **3.5. Definición de Roles en los Módulos**

En la siguiente tabla (Tabla 3.1), se muestran las funciones que ejerce cada usuario de un Grupo de Investigación desde el punto de vista de los acontecimientos del evento y desde el punto de vista del Sistema.

---

<sup>4</sup> <http://staruml.sourceforge.net/en/>

Cada usuario registrado en el Sistema, tiene la posibilidad de tener de uno o más perfiles, de tal manera que pueden ejercer varios roles, dependiendo de las políticas especificadas por el Comité Organizador.

**Tabla 3.1. Definición de Roles en los Módulos**

<b>Usuario</b>	<b>Función dentro del negocio</b>	<b>Función dentro del Sistema</b>
<b>Director del Grupo de Investigación</b>	Representante del Grupo de Investigación, encargado de planificar y elaborar las directrices para los demás miembros organizadores, con la finalidad de llevar a cabo un evento de ámbito científico.	Administrador General
<b>Consejo Técnico Organizador</b>	Consejo conformado por los organizadores, una de sus tareas es la de decidir qué resumen de un artículo científico (presentado por un usuario al conocer la convocatoria) es válido para seguir en el proceso de evaluación.	Administrador
<b>Asistente del Grupo de Investigación</b>	Ayudante del Director del centro que sigue las directrices planificadas para la elaboración exitosa del evento. Cabe recalcar que el asistente tiene un perfil igual al del Director,	Administrador General
<b>Suscriptor</b>	Usuario visitante del Portal con la finalidad de estar vinculado con un Grupo de Investigación o por el deseo de registrarse a un evento.	Suscriptor
<b>Evaluador</b>	Participante del evento en calidad de evaluador, con el propósito de verificar la utilidad del conocimiento demostrado en el artículo.	Evaluador
<b>Conferencista</b>	Usuario que presentó un artículo de índole científico con un resultado exitoso, el conferencista también puede ser invitado por el Comité Organizador.	Conferencista

## CAPÍTULO IV

### 4. DISEÑO DEL SISTEMA

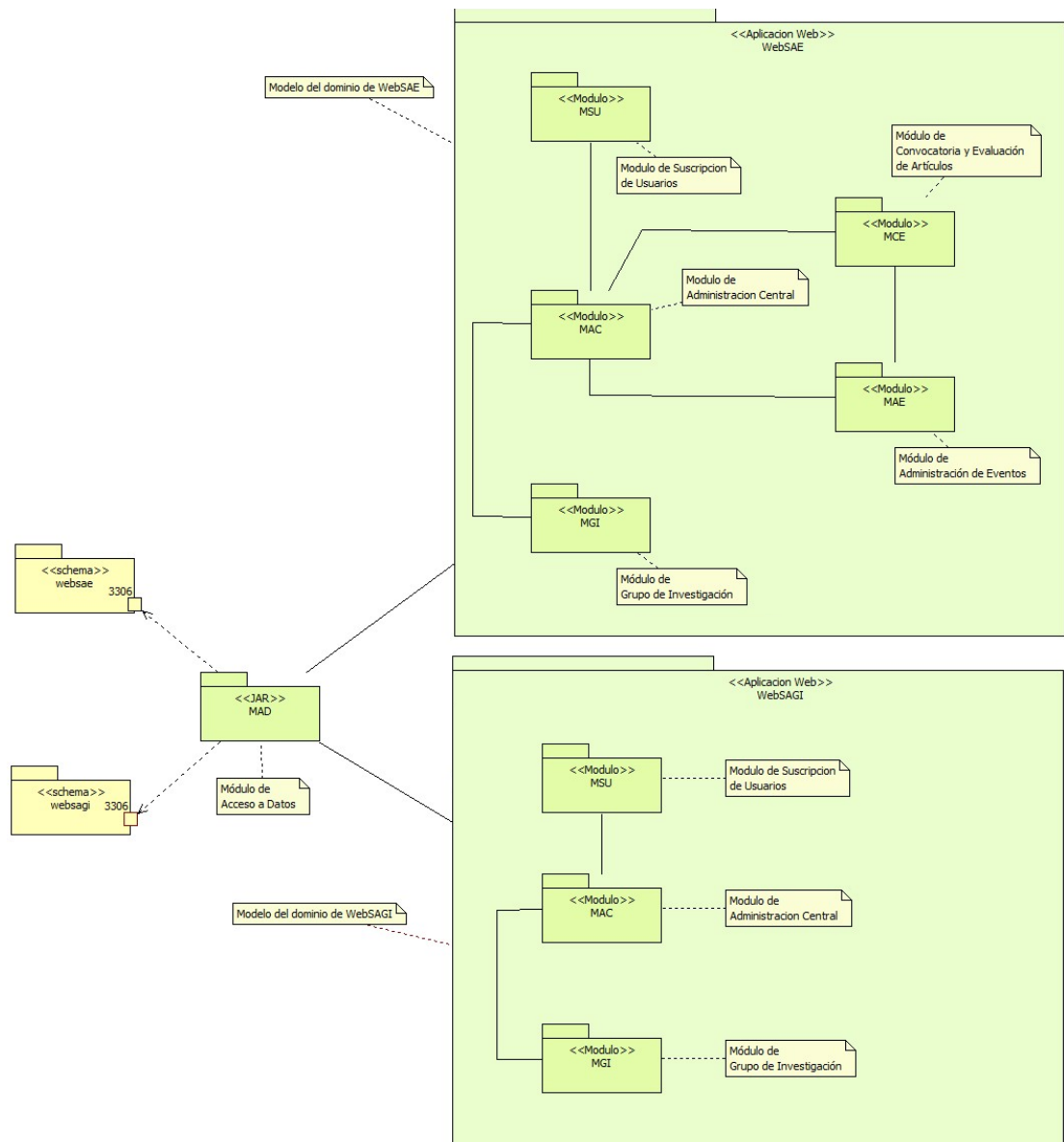
#### 4.1. Diseño de la Arquitectura del Sistema

Debido al requerimiento de separar el Sistema actual AppVlir8 en dos productos, uno en la Administración de Eventos Científicos y el otro en la Administración de Grupos de Investigación (WebSAGI), es concebida como dos sistemas independientes que se unen bajo un módulo que no ha sido mencionado hasta el momento, debido a que es un módulo que administra las conexiones a las Bases de Datos, llamado **Módulo de Acceso a Datos (MAD)**<sup>5</sup> y no pertenece a ninguno de los dominios de los Sistemas mencionados.

---

<sup>5</sup> Este módulo fue diseñado por la Srta. Diana Crespo –coordinadora de Sistemas en Molemotor S.A.-, llamado inicialmente ADA, por el diseño portable de éste, se lo adecuó a MySQL (debido a que fue implementado para SQL Server 2000) para que pueda conectarse con las Bases de Datos WebSAE (para la Administración de Eventos) y WebSAGI (para la Administración de los Grupos de Investigación).

En el gráfico 4.1 se muestra el diseño arquitectónico de ambos sistemas y la unión de los mismos a través del módulo MAD. Es importante recalcar que para este trabajo de Proyecto de Grado sólo se implementó el Sistema de Administración de Eventos (**WebSAE**).

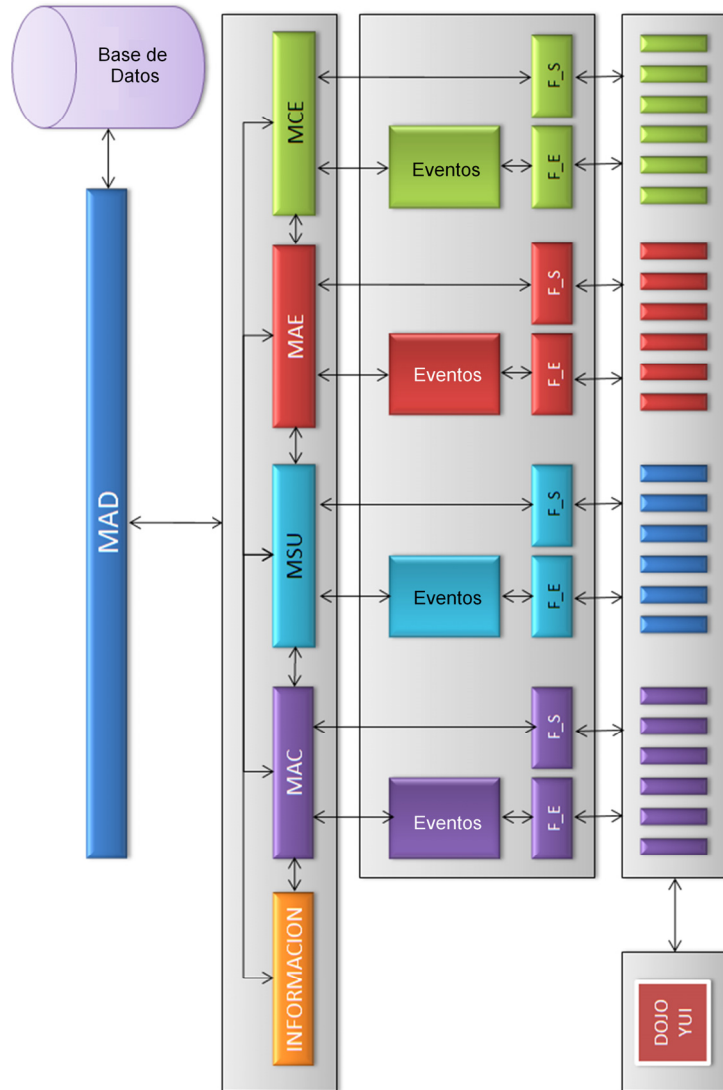


**Gráfico 4.1.** Diseño de la Arquitectura del Sistema de Administración de Eventos y Grupos de Investigación (WebSAEGI).



### 4.1.1. Diseño Arquitectónico

La arquitectura de MERODE fue aplicada a WebSAE; a continuación se detalla en el Gráfico 4.2. la aplicación de la misma.



**Gráfico 4.2.** Diseño de la Arquitectura WebSAE.

Cada módulo se encuentra diferenciado por un color y separado en tres capas: la del dominio, la funcional y la interfaz.

Algunas observaciones:

- En la capa del dominio, se puede observar que los objetos de los diversos módulos, se pueden comunicar.
- Entre los eventos de diversos módulos, no sucede lo mismo; sin embargo, desde un evento en un módulo se puede acceder a objetos de módulos distintos.

#### **4.1.2. Módulo del Sistema**

Los módulos diseñados e implementados en el Sistema de Administración de Eventos Científicos (WebSAE) son:

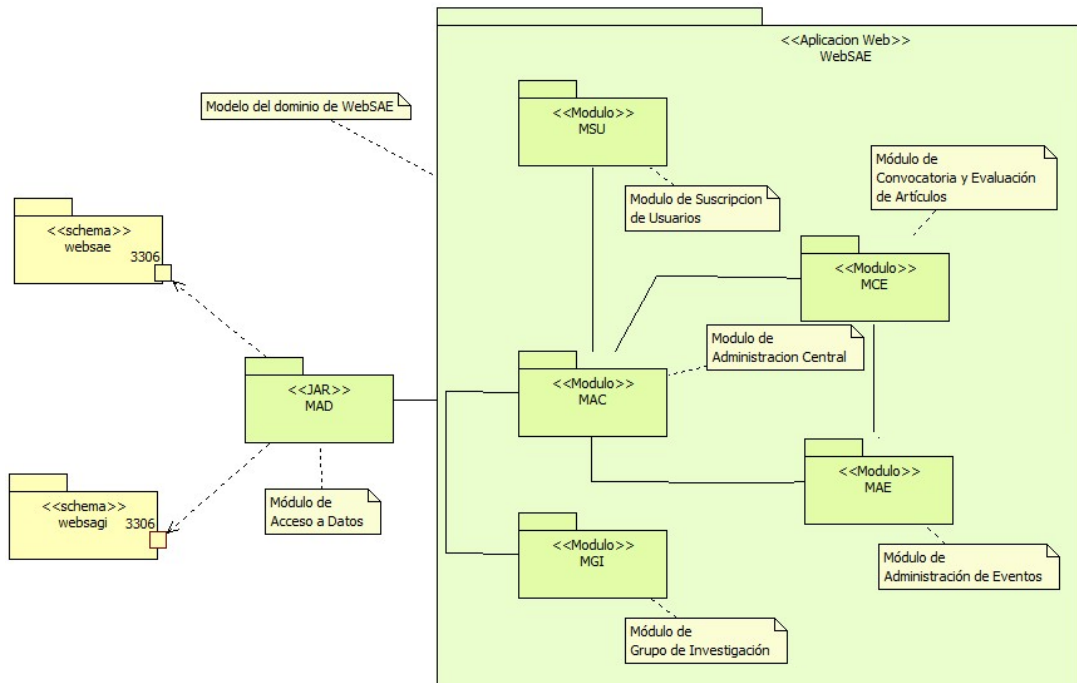
Módulos pertenecientes al dominio:

- Módulo de Administración Central (MAC).
- Módulo de Suscripción de Usuarios (MSU).
- Módulo de Administración de Eventos (MAE).
- Módulo de Convocatoria y Evaluación de Artículos (MCE).

Módulos de información:

- Módulo de Acceso a Datos (MAD).

- Módulo de Grupos de Investigación (MGI).



**Gráfico 4.3 Módulos del Sistema.**

#### 4.1.3. Arquitectura Basada en Patrones de Diseño

La Arquitectura de J2EE propuesta por la SUN, sigue patrones de diseño para hacer uso de ella. El patrón utilizado en el presente proyecto fue MVC, el cual describiremos brevemente a continuación:

##### **Modelo – Vista – Controlador.-**

Este patrón diseño es totalmente compatible con la arquitectura de MERODE. J2EE también lo utiliza, en la especificación JSP 2.0 se puede separar de manera completa la vista del controlador mediante el uso de la librería de etiquetas estándares para páginas

JSP (JavaServer Pages Standard Tag Library, **JSTL**), esto se menciona debido a que anteriormente era necesario el uso de scriptlets en páginas JSP, lo que muchas veces se daba el caso en el que la capa de la vista con el controlador se juntaban.

#### **4.2. Diagramas de Clases del Sistema**

Debido al tamaño del diagrama de clases, hemos considerado conveniente segmentarlos por módulos y presentarlos en el ANEXO 4. En esta sección sólo mostraremos el Diagrama de Clases del Módulo de Convocatoria y Evaluación de Artículos por su relevancia (Gráfico 4.4)



### **4.3. Modelo Lógico de la Base de Datos**

De forma similar a la sección anterior optamos por presentar la totalidad del modelo entidad –relación en el ANEXO 5. En esta sección sólo presentaremos el Modelo Entidad - Relación del Módulo de Convocatoria y Evaluación de Artículos (Gráfico 4.5).

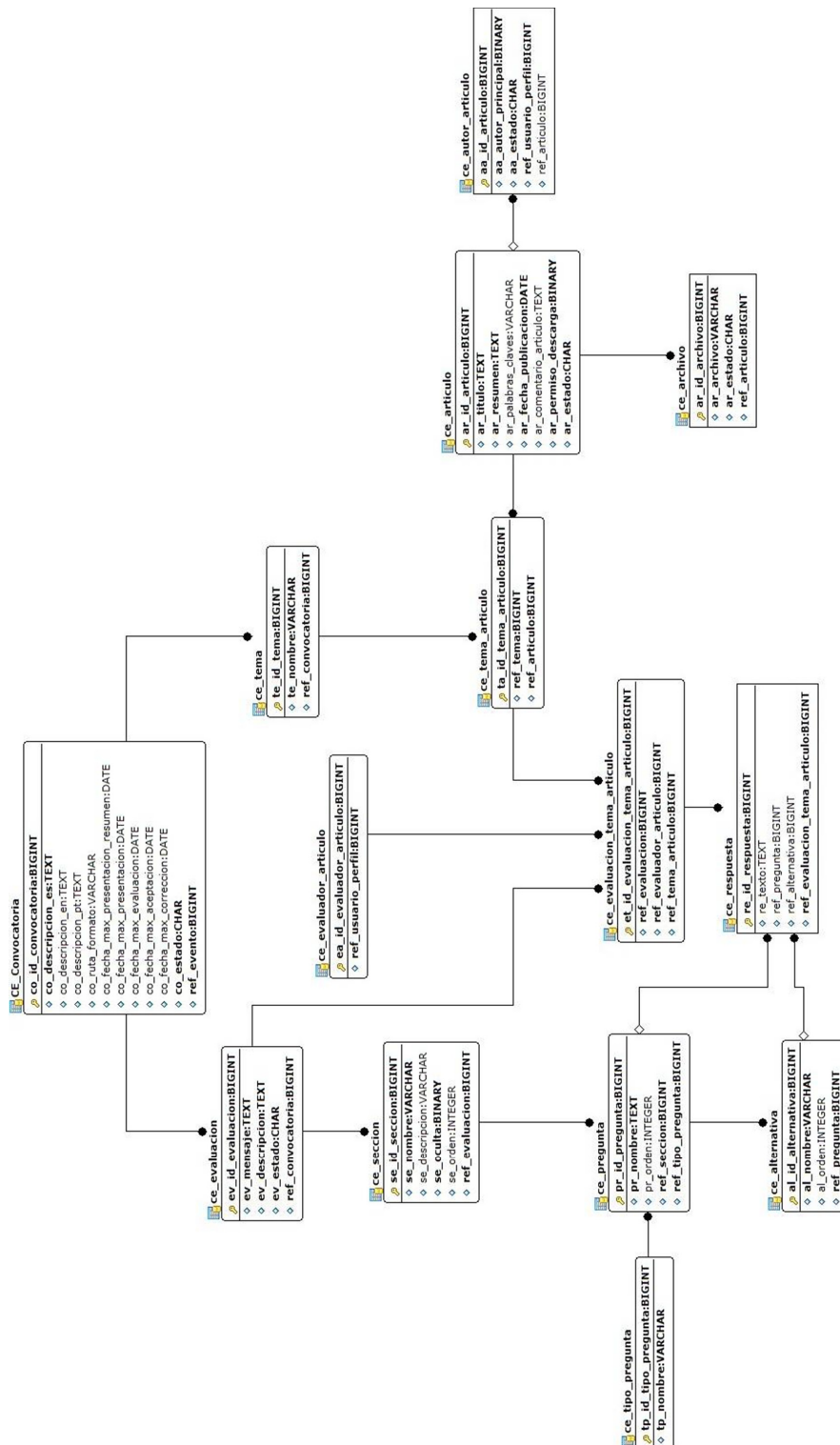


Gráfico 4.5 ER – Módulo de Convocatoria y Evaluación de Artículos (MCE)

# CAPÍTULO V

## 5. IMPLEMENTACIÓN Y PRUEBAS

### 5.1. Capas del Sistema y Comunicación entre Capas

El modelamiento del dominio que se utiliza no usa el paso de mensajes como mecanismo de comunicación entre objetos, usa el concepto de eventos comunes.

A continuación se procede a hacer una correspondencia entre el diseño y la implementación utilizando MERODE y MDA.

#### 5.1.1. Implementación Basada en MERODE y MDA.

Una vez que se han definido los tres modelos utilizados en la aplicación: el PIM, el PSM y el PSI, a continuación se describen las reglas de transformación para el paso del PIM al PSM, del PSM al PSI.

#### **Del PIM al PSM.-**

La primera regla es transformar el EDG en un diagrama de clases. Esta tarea la realizamos de forma manual debido a que no encontramos cómo automatizar todo el proceso de conversión. Cabe



mencionar, que el EDG fue hecho mediante la herramienta MERMAID y los diagramas de clases los elaboramos con StarUML manualmente.

Con respecto a la transformación del EDG, la relación de existencia-dependencia considerada en el análisis de MERODE, al pasarla a UML, ésta se convierte en una **relación de composición**.

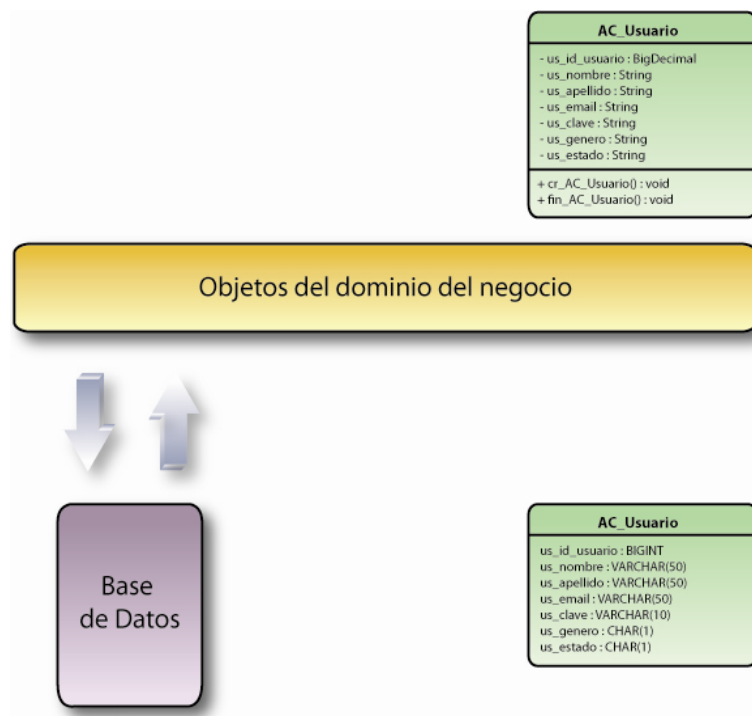
#### **Del PSM al PSI.-**

*En la capa del dominio*, del Diagrama de Clases que se elaboró se generó el código de esta capa (en Java). Así mismo se generó el correspondiente DDL para la creación de la Base de Datos (en nuestro caso MySQL) (Figura 5.1). Este fue el único paso que se realizó de manera automática.

*En la capa de control de eventos*, es importante acotar que en MERODE existen dos reglas para realizar este tipo de transformación: 1) implementar una sola clase por cada evento; y 2) en una sola clase implementar todos los eventos, para salvaguardar la atomicidad de los mismos.

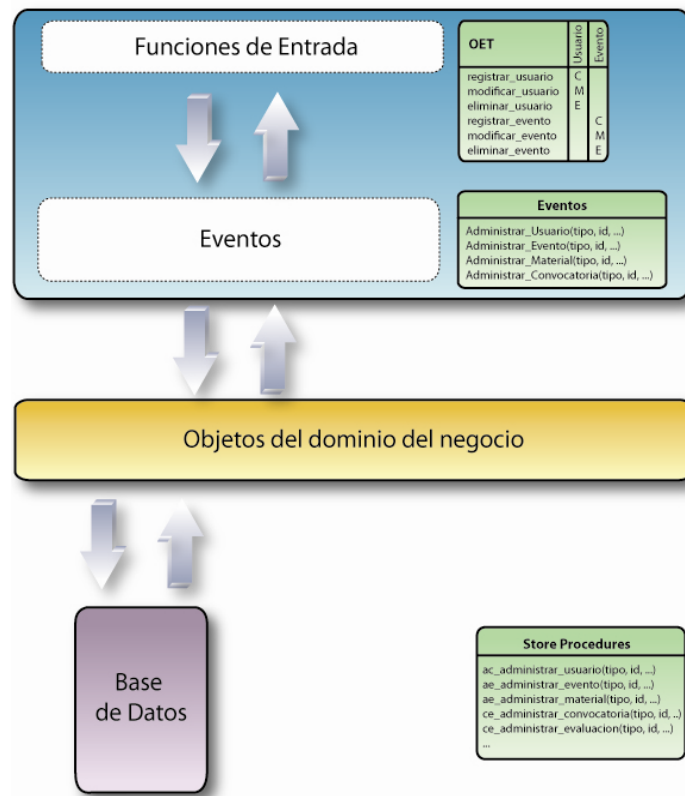
Nosotros utilizamos un híbrido, donde, por cada objeto se creó una clase-evento en la que se realizaban los eventos de crear, modificar y

eliminar. Por ejemplo: para el objeto AC\_Usuario se creó la clase Administrar\_Usuario y dentro de esta clase-evento diferenciamos el tipo de evento a realizar. Con esto nos ahorramos una gran cantidad de clases debido fundamentalmente a la reutilización de código.



**Figura 5.1** Regla de transformación de la capa del dominio.

Así mismo, por cada tipo de evento híbrido, se creó un Procedimiento Almacenado. Siguiendo el ejemplo anterior, ante la clase-evento Administrar\_Usuario se creó el Procedimiento Almacenado administrar\_usuario. (Figura 5.2)



**Figura 5.2** Regla de transformación de la capa de control de eventos.

*En la capa de interfaz, para la mayoría de los objetos del dominio creamos un directorio del lado del cliente y en cada directorio un sub directorio con el evento a implementar. Por ejemplo: con el objeto AC\_Usuario del lado del servidor, se creó el directorio usuario del lado del cliente de la aplicación, y dentro de ese directorio, los sub directorios: crear, modificar y eliminar, cada directorio puede ser accedido dependiendo del perfil asignado a un usuario.*

Para la implementación de seguridades a ciertos directorios de la Aplicación Web, se utilizaron los filtros que nos proporciona la arquitectura J2EE.

## 5.2. Plan de Pruebas

Las pruebas realizadas con el fin de asegurar la calidad del Sistema, en cuanto a funcionalidad, fueron las **Pruebas Unitarias**, se ha preparado una lista de las operaciones más relevantes que se realizan con mayor frecuencia.

**Tabla 5.1. Prueba Unitaria N°: 01**

<b>Caso de Uso:</b> Ingresar al Sistema		<b>N°:</b> 01	
<b>Escenario:</b> El ingreso se realiza utilizando correo y contraseña válidos			
<b>Responsable:</b> Vanessa Izquierdo			<b>Fecha:</b> 26-Nov-2009
<b>Precondiciones</b>	Tiene que existir un usuario creado en el sistema para poder ingresar.		
<b>Datos de Entrada</b>	Correo: "vanejoha@gmail.com" Contraseña: "vanessa"		
<b>Descripción de Pasos</b>	Se ingresa el correo "vanejoha@gmail.com" en el campo Correo. Se ingresa la palabra "vanessa" en el campo Contraseña. Se presiona el botón "Ingresar" o presionar la tecla Enter.		
<b>Resultado Esperado</b>	Se ingresa al sistema como el usuario dado, se	<b>Cumplimiento</b>	<b>SÍ</b>
			<b>NO</b>

	presenta la pantalla siguiente con el menú.			
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla 5.2. Prueba Unitaria N°: 02**

<b>Caso de Uso:</b> Crear Cuenta		<b>N°:</b> 02
<b>Escenario:</b> La creación de la cuenta se la realizará utilizando datos correctos		
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 26-Nov-2009
<b>Precondiciones</b>	Se debe ingresar todos los campos obligatorios en formato correcto.	
<b>Datos de Entrada</b>	Correo Electrónico: "correo_electronico@hotmail.com" Contraseña: "contraseña" Re-Contraseña: "contraseña" Nombres: "Nombre1 Nombre2" Apellidos: "Apellido1 Apellido2" Género: "Masculino" Dirección: "" Teléfono: "" Celular: "" Día de Nacimiento: "" País: "Ecuador" Ciudad: "Guayaquil" Carrera o Título: "Ing. en Computación" Tipo Empresa: "Pública" Cargo: "Estudiante" Empresa/Universidad: "ESPOL" Teléfono (Oficina): ""	
<b>Descripción de Pasos</b>	Dar clic en Crear Cuenta debajo de Ingresar al Sistema o en el Menú de Opciones	

	Ingresar los valores en los campos como en Datos de Entrada Dar clic en "Crear Cuenta"			
<b>Resultado Esperado</b>	Se crea correctamente la cuenta, muestra el mensaje correspondiente y es re direccionado a la página de inicio.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla N.3. Prueba Unitaria N°: 03**

<b>Caso de Uso:</b> Crear una convocatoria		<b>N°:</b> 03
<b>Escenario:</b> Crear una convocatoria con intervalos de fechas correctamente ingresadas		
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009
<b>Precondiciones</b>	Estar logeado en el sistema como administrador.	
<b>Datos de Entrada</b>	Fecha de Presentación: "2009-12-13" Fecha de Evaluación: "2009-12-29" Fecha de Notificación: "2010-01-02" Fecha de Corrección: "2010-01-09" Descripción (ES): "Llamada de artículos para este evento." Descripción (EN): "Call for papers for this event." Descripción (PT): "Chamada de trabalhos para este evento."	
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el nombre del evento Dar clic en la opción del menú "Convocatoria"	

	<p>Ingresar los datos descritos en Datos de Entrada          Dar clic en el botón "Examinar" para subir un formato de archivo.          Dar clic en el botón "Subir"          Dar clic en el botón "Actualizar"</p>			
<b>Resultado Esperado</b>	Se actualiza exitosamente la convocatoria, se muestra el mensaje correspondiente.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla N.4. Prueba Unitaria N°: 04**

<b>Caso de Uso:</b> Crear un evento		<b>N°:</b> 04
<b>Escenario:</b> Creación de un evento ingresando solo los campos obligatorias con datos correctos		
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009
<b>Precondiciones</b>	Estar logeado en el sistema como administrador. El evento a crear no debe ya estar creado.	
<b>Datos de Entrada</b>	Tipo de Evento: "Jornadas" Evento (ES): "Nueva Evento de Prueba" Evento (EN): "New Test Event" Evento (PT): "Novo teste de evento" Fecha Inicio: "2009-12-18" Fecha Fin: "2009-12-24" Estado de Registro: "Activo"	
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el icono de Nuevo. Se abre la ventana de popup para ingresar los datos del evento.	

	Ingresar los datos detallados en Datos de Entrada. Dar clic en el botón "Guardar".			
<b>Resultado Esperado</b>	Se crea exitosamente el evento, se muestra el mensaje correspondiente, se cierra el popup y se actualiza la tabla de los eventos.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla N.5. Prueba Unitaria N°: 05**

<b>Caso de Uso:</b> Crear una evaluación		<b>N°:</b> 05
<b>Escenario:</b> Crear/Actualizar la información de evaluación para los artículos		
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009
<b>Precondiciones</b>	Haber iniciado sesión en el sistema como administrador. El tema no debe existir en la convocatoria de ese evento.	
<b>Datos de Entrada</b>	Tema: "Tema nuevo"	
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el nombre del evento Dar clic en la opción del menú "Convocatoria" Dar clic en el botón "Evaluación" <i>Se re direccionará a la página de evaluación</i> En el campo Mensaje al Evaluador ingresar "Gracias por ayudarnos a la evaluación". En el campo Instrucciones ingresar "Estas son las instrucciones a seguir".	



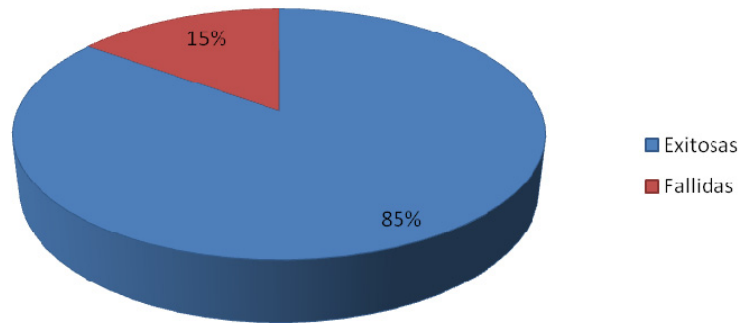
	En la campo Estado seleccionar “Vigente” Dar clic en el botón “Actualizar”			
<b>Resultado Esperado</b>	Se crea/actualiza la evaluación, se muestra en mensaje correspondiente.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

Una muestra de las evidencias de las pruebas unitarias realizadas se encuentra en el **APÉNDICE F**.

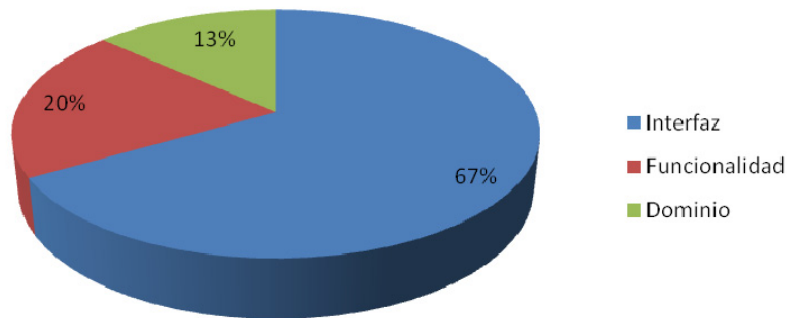
### **5.3. Resultados de las Pruebas y Métricas tomadas**

Con respecto a las pruebas unitarias, se realizaron un total de 100 pruebas al Sistema, de las cuales 85 fueron exitosas y 15 fallidas como se muestra en el Gráfico 5.1.

En las pruebas con resultados fallidos se ha encontrado una mayor incidencia en errores a nivel de la capa de interfaz del Sistema, ver Gráfico 5.2.



**Gráfico 5.1** Resultados de las pruebas unitarias.

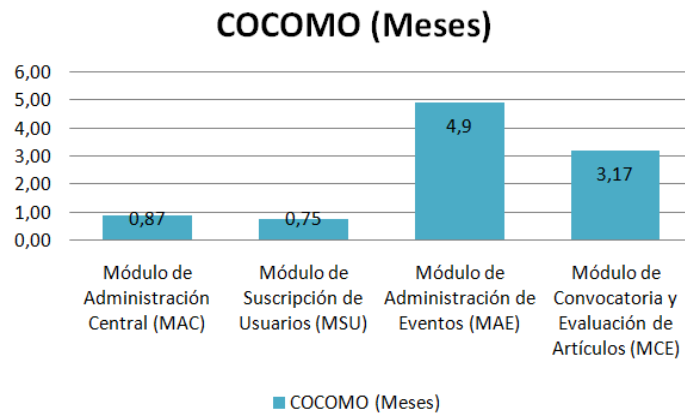


**Gráfico 5.2** Pruebas unitarias fallidas.

Con respecto a las métricas, por cada módulo que fue diseñado e implementado, realizamos una estimación inicial de tiempo en meses, la cual se muestra en el Gráfico 5.3. Para el cálculo se consideraron los siguientes factores:

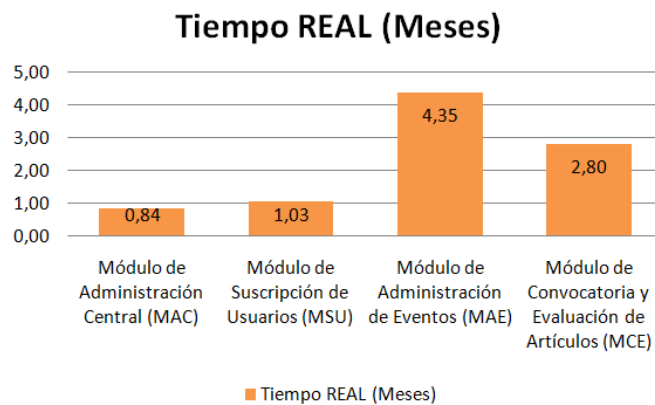
- con respecto al producto: RELY (Bajo), DATA (Nominal), CPLX (Bajo).
- con respecto al personal: ACAP (Muy Alto), AEXP (Muy Alto), PCAP (Alto), VEXP (Alto), LEXP (Muy Alto).

- Con respecto al proyecto: MODP (Muy Alto), TOOL (Muy Alto), SCED (Muy Alto).



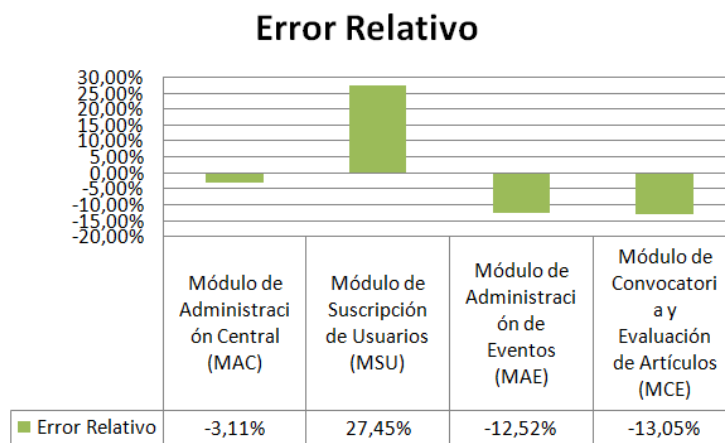
**Gráfico 5.3** Estimación de tiempos realizada en COCOMO.

Durante el desarrollo de cada módulo se levantaron métricas, las mismas que fueron procesadas y cuyos resultados se muestran en el Gráfico 5.4.



**Gráfico 5.4** Tiempos reales de cada módulo.

En el gráfico 5.5 se muestra el porcentaje de error entre el tiempo estimado y el tiempo real. En MAC el porcentaje es -3,11% y no hubieron problemas debido a que era solo código JAVA y se conocían los requerimientos, además ya se encontraba desarrollada la Base de Datos (MySQL) y el código java de la capa del dominio que fueron generados de manera automática. Estos factores contribuyeron al rápido desarrollo de este módulo.



**Gráfico 5.5** Porcentajes de error con respecto a lo estimado con el tiempo real.

Para MSU hubo un retraso, esto fue en parte al tiempo de aprendizaje de algunas librerías. Sin embargo, el tiempo de aprendizaje invertido en este módulo nos sirvió para que en los demás módulos (MAE y MCE), disminuyéramos los tiempos.

## CONCLUSIONES Y RECOMENDACIONES

Una vez concluido el Proyecto de Grado, tenemos como resultado un Sistema de Administración de Eventos Científicos para Grupos de Investigación y podemos concluir lo siguiente:

- 1 Mediante las pruebas unitarias realizadas, se observó que usando la metodología MERODE el porcentaje de cambios a nivel de la capa del dominio (Gráfico 5.2) fue significativamente menor que los realizados en la capa de la interfaz, debido a la distribución de las capas de los objetos que propone MERODE, especificado en la sección 2.1 del cap. 2.
- 2 Podemos verificar la vigencia y el uso efectivo del modelo matemático COCOMO, para una buena estimación de tiempo, usando los parámetros adecuados, detallados en la sección 5.3 del cap. 5.
- 3 Los tiempos estimados con respecto a los reales, no se ven afectados de mayor forma por el uso de MERODE y MDA en el análisis y desarrollo del Sistema. Dadas las ventajas de flexibilidad que ha demostrado el uso de éstas metodologías en el producto final, se comprueba que es factible y recomendable el uso de las mismas.

- 4 Mediante la realización del Portal se han optimizado los procesos de Convocatoria y Evaluación de Artículos.

Las recomendaciones que podemos valorar como necesarias en este Proyecto de Grado son:

- 1 Automatizar los procesos que se realizaron de forma manual, es decir, generar el UML (dependiendo de la herramienta CASE: StarUML, MagicDraw, etc.) a partir del archivo XML que nos proporciona MERMAID, para la automatización de esta transformación.
- 2 Automatizar los estándares que se siguieron a nivel de la capa del control de eventos que pueden ser generados a partir de las clases del dominio, como se muestra en la Figura 5.2.

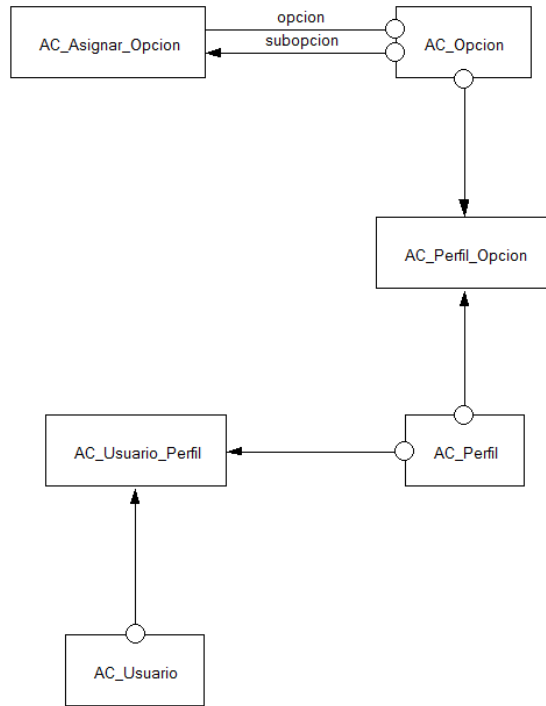
- 3 Se recomienda mantener el uso de la arquitectura J2EE adaptada a la arquitectura de MERODE, si se desea añadir alguna funcionalidad más.
- 4 Para futuros mantenimientos de la aplicación recomendamos el uso del repositorio<sup>6</sup> de los Proyectos de la materia de Ingeniería de Software, sobre el que se ha estado trabajando de manera continua, este repositorio se encuentra versionado bajo la herramienta SubVersion.

---

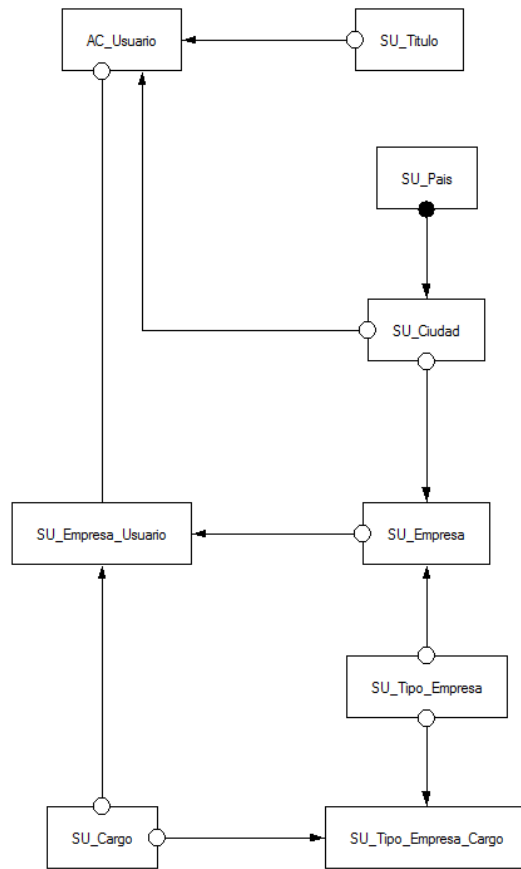
6 <https://proyectossw.espol.edu.ec/svn/vlir8>



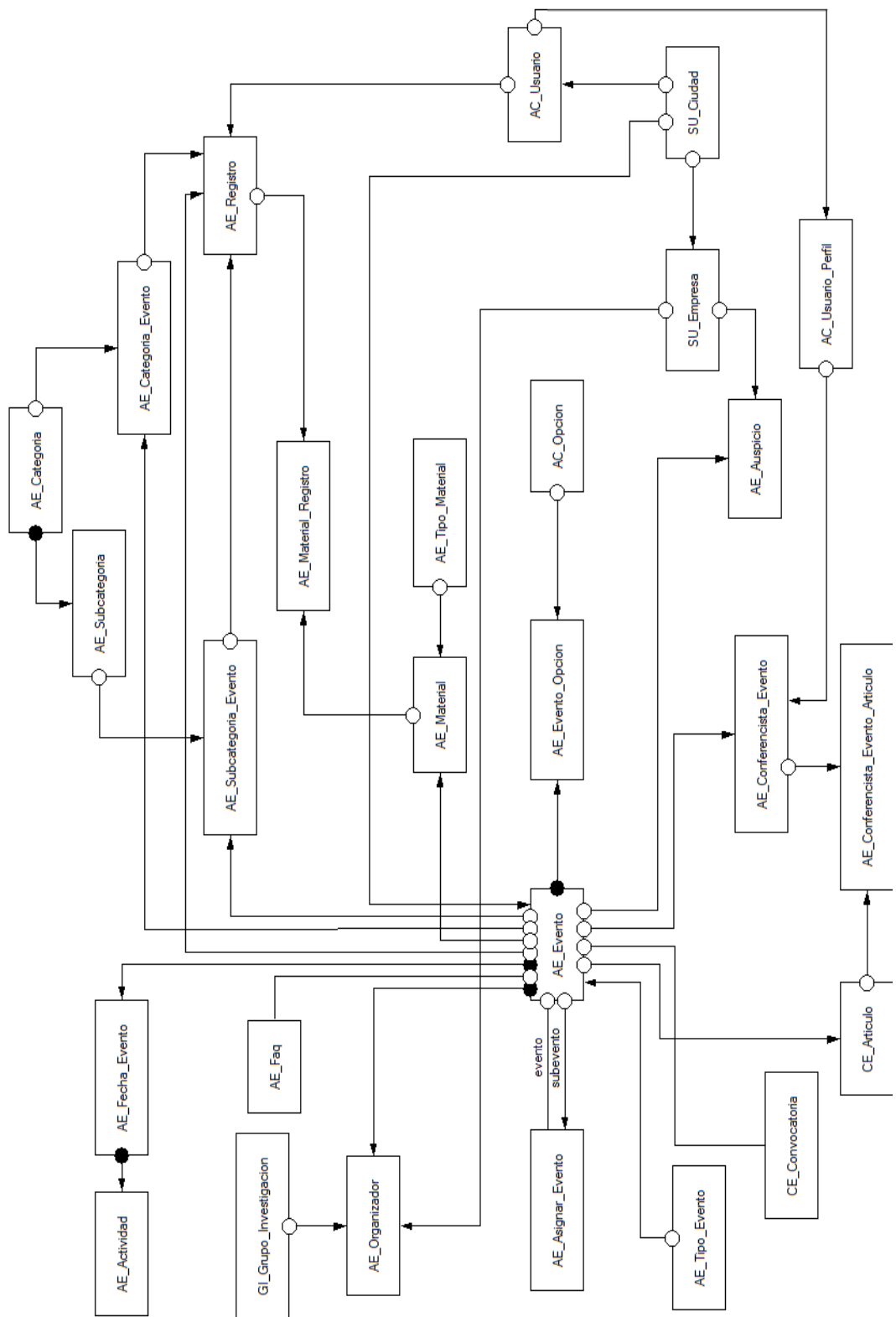




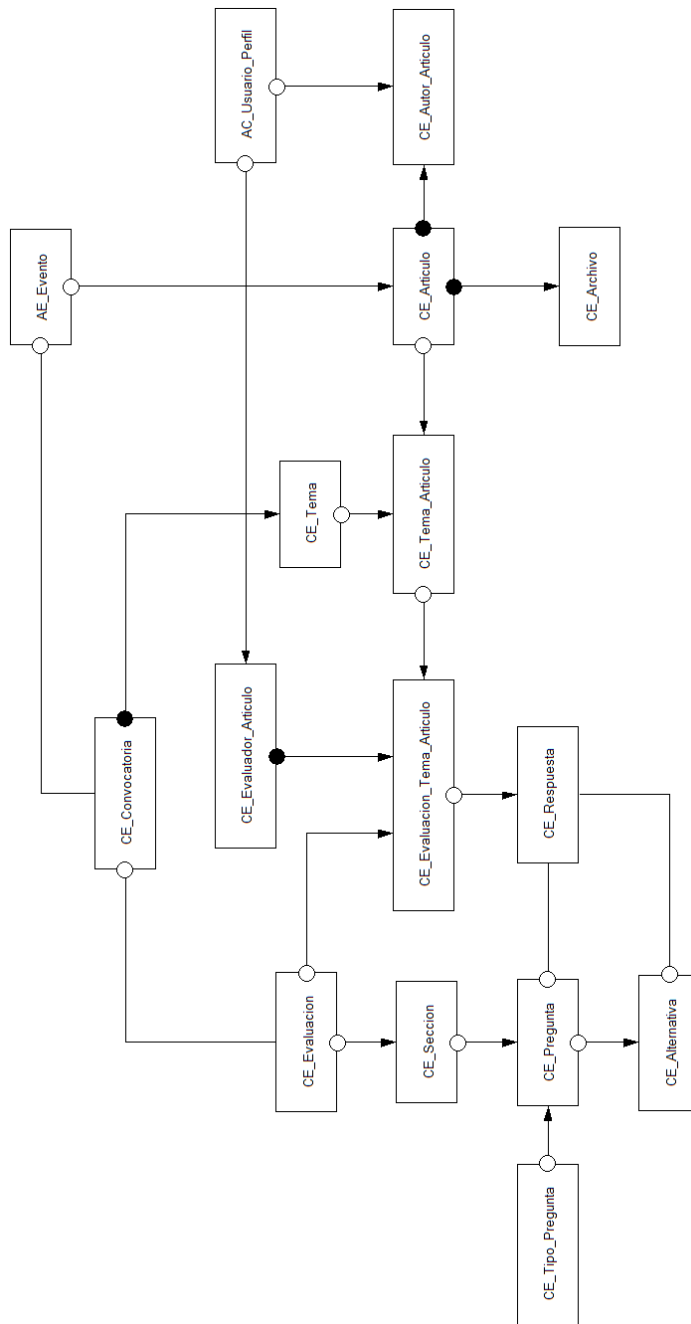
**Gráfico A.2 EDG - Módulo de Administración Central (MAC)**



**Gráfico A.3 EDG - Módulo de Suscripción de Usuarios (MSU)**



**Gráfico A.4 EDG - Módulo de Administración de Eventos (MAE)**



**Gráfico A.5 EDG - Módulo de Convocatoria y Evaluación de Artículos (MCE)**

## APÉNDICE B: Tabla de Objetos-Eventos

	AE_Evento
registrar_evento	O/C
modificar_evento	O/M
eliminar_evento	O/E
registrar_material	A/M
modificar_material	A/M
eliminar_material	A/M
registrar_material_subevento	A/M
modificar_material_subevento	A/M
eliminar_material_subevento	A/M
Asignar_evento_opcion	A/M
actualizar_faq	A/M
registrar_agenda	A/M
modificar_agenda	A/M
eliminar_agenda	A/M
registrar_auspiciante	A/M
modificar_auspiciante	A/M
eliminar_auspiciante	A/M
registrar_horario	A/M
modificar_horario	A/M
eliminar_horario	A/M
registrar_subcategoria	A/M
modificar_subcategoria	A/M
eliminar_subcategoria	A/M
registrar_subcategoria_evento	A/M
modificar_subcategoria_evento	A/M
eliminar_subcategoria_evento	A/M
registrar_categoria_evento	A/M
modificar_categoria_evento	A/M
eliminar_categoria_evento	A/M
registrar_categoria_subevento	A/M
modificar_categoria_subevento	A/M
eliminar_categoria_subevento	A/M
registrar_conferencista_evento	A/M
modificar_conferencista_evento	A/M
eliminar_conferencista_evento	A/M
registrar_conferencista_subevento	A/M
modificar_conferencista_subevento	A/M
eliminar_conferencista_subevento	A/M
registrar_organizador	A/M

eliminar_organizador	A/M
registrar_subevento	O/C
modificar_subevento	O/M
eliminar_subevento	O/E
registrar_usuario_evento	A/M
modificar_usuario_evento	A/M
eliminar_usuario_evento	A/M

**Tabla B.1 OET – Objeto AE\_Evento**

## APÉNDICE C: Máquinas de Estado Finitos

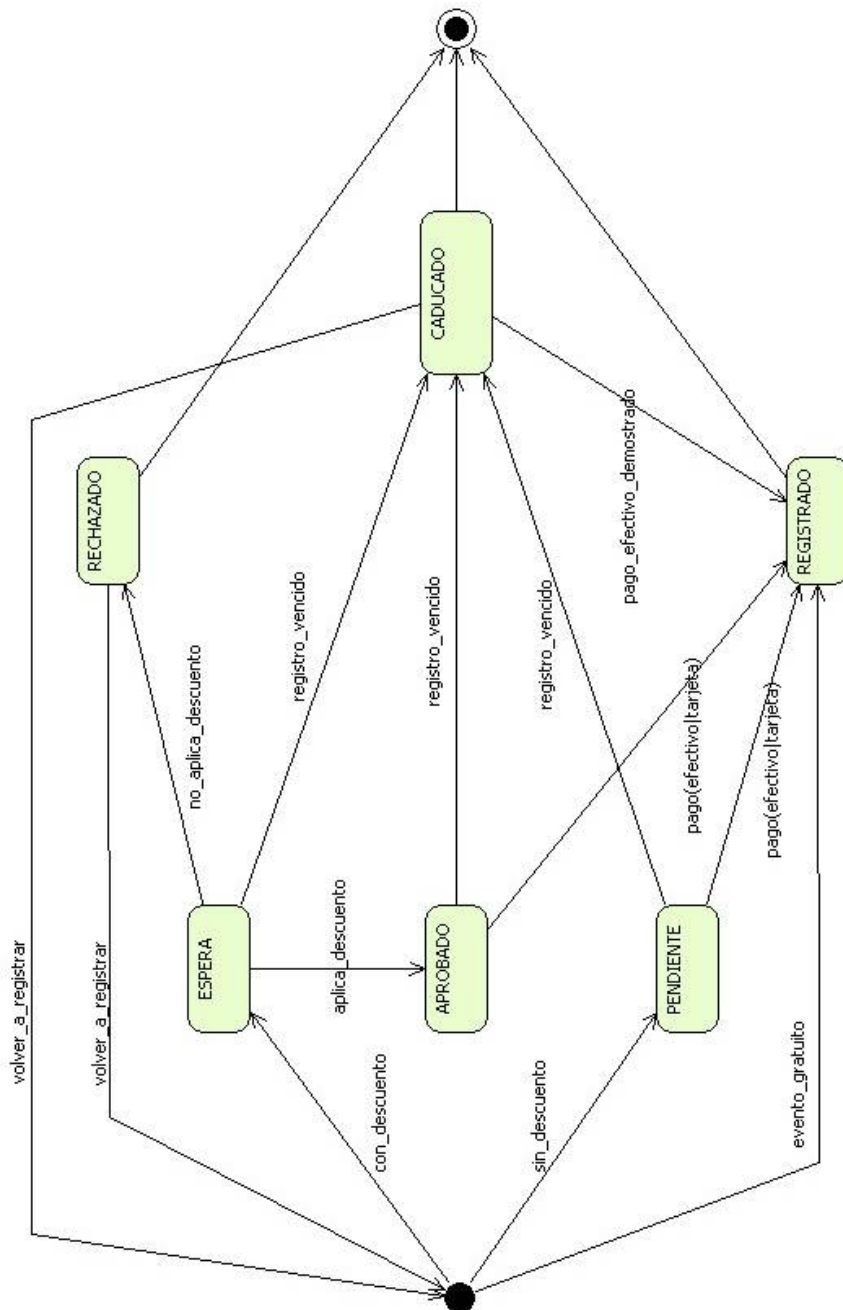


Gráfico C.1 FSM del objeto AE\_Registro

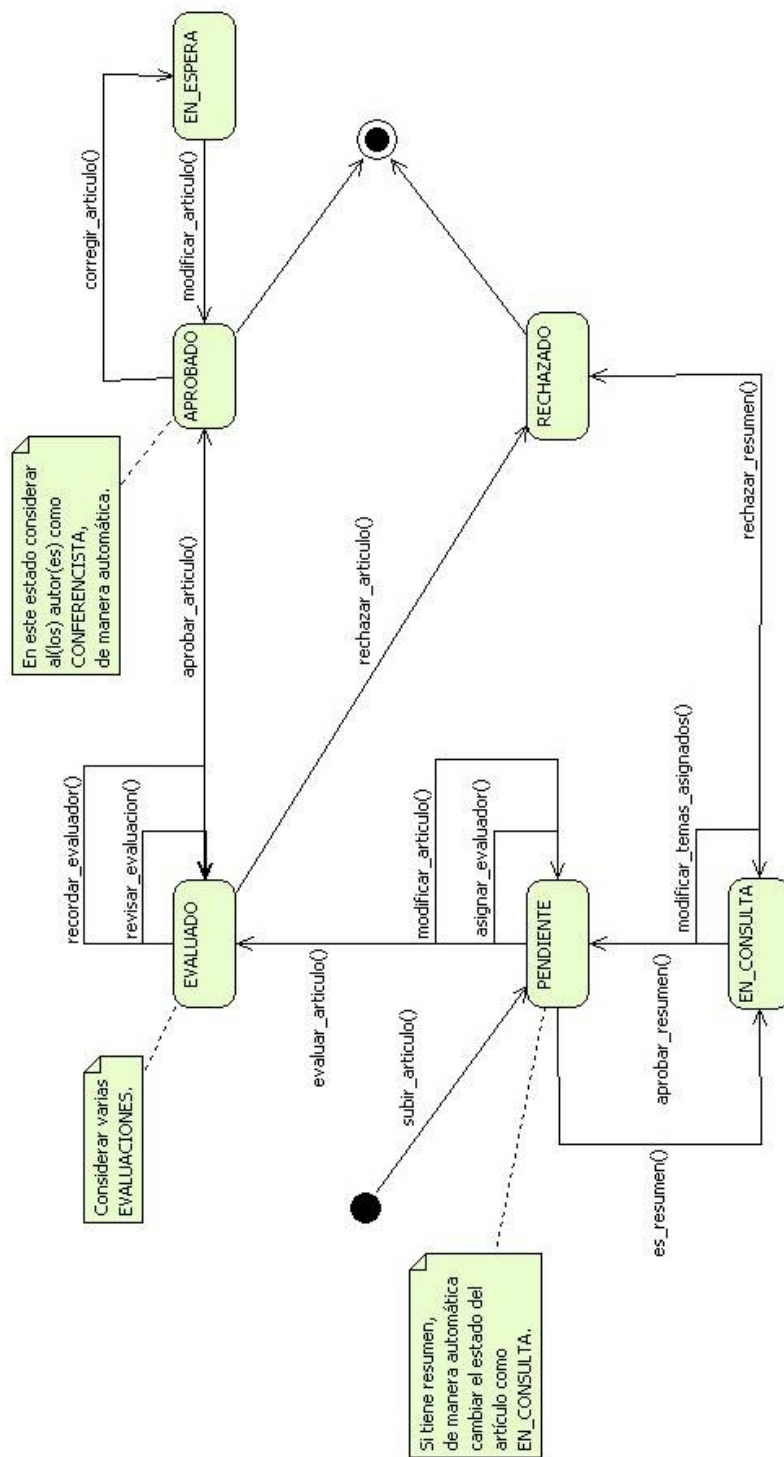


Gráfico C.2 FSM del objeto CE\_Articulo



# APÉNDICE D: Diagramas de Clases por Módulos

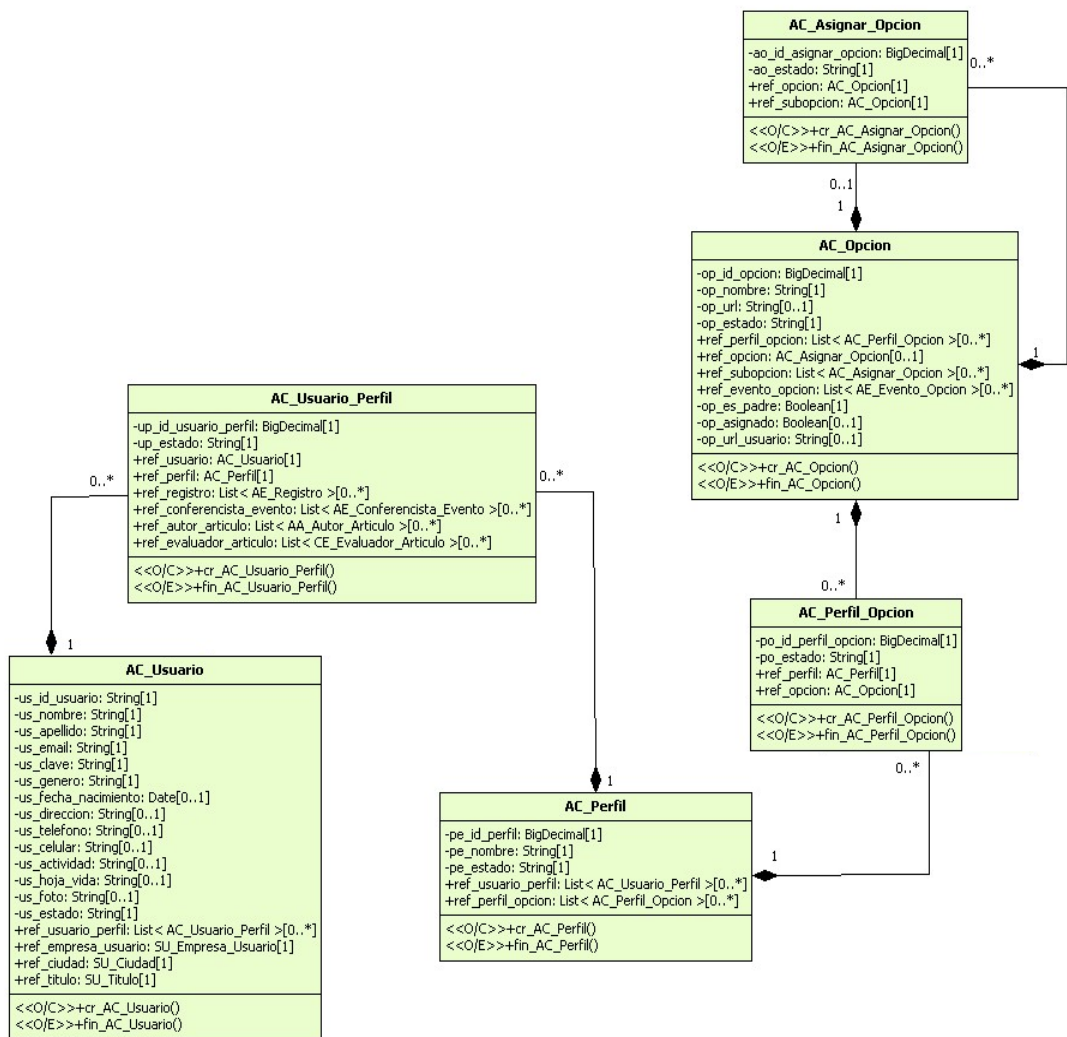


Gráfico D.1 Diagrama de Clases - Módulo de Administración Central (MAC)

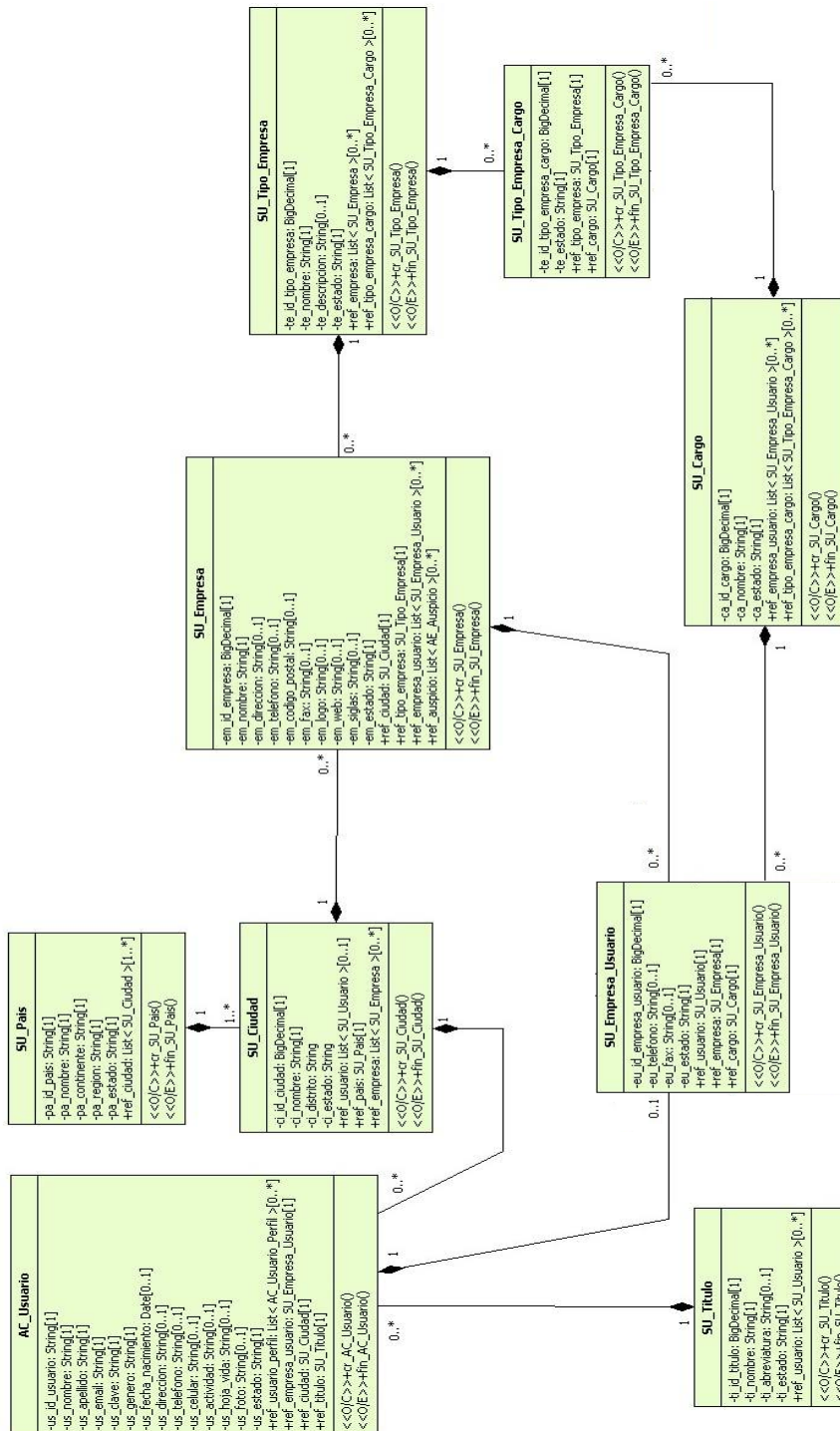


Gráfico D.2 Diagrama de Clases - Módulo de Suscripción de Usuarios (MSU)

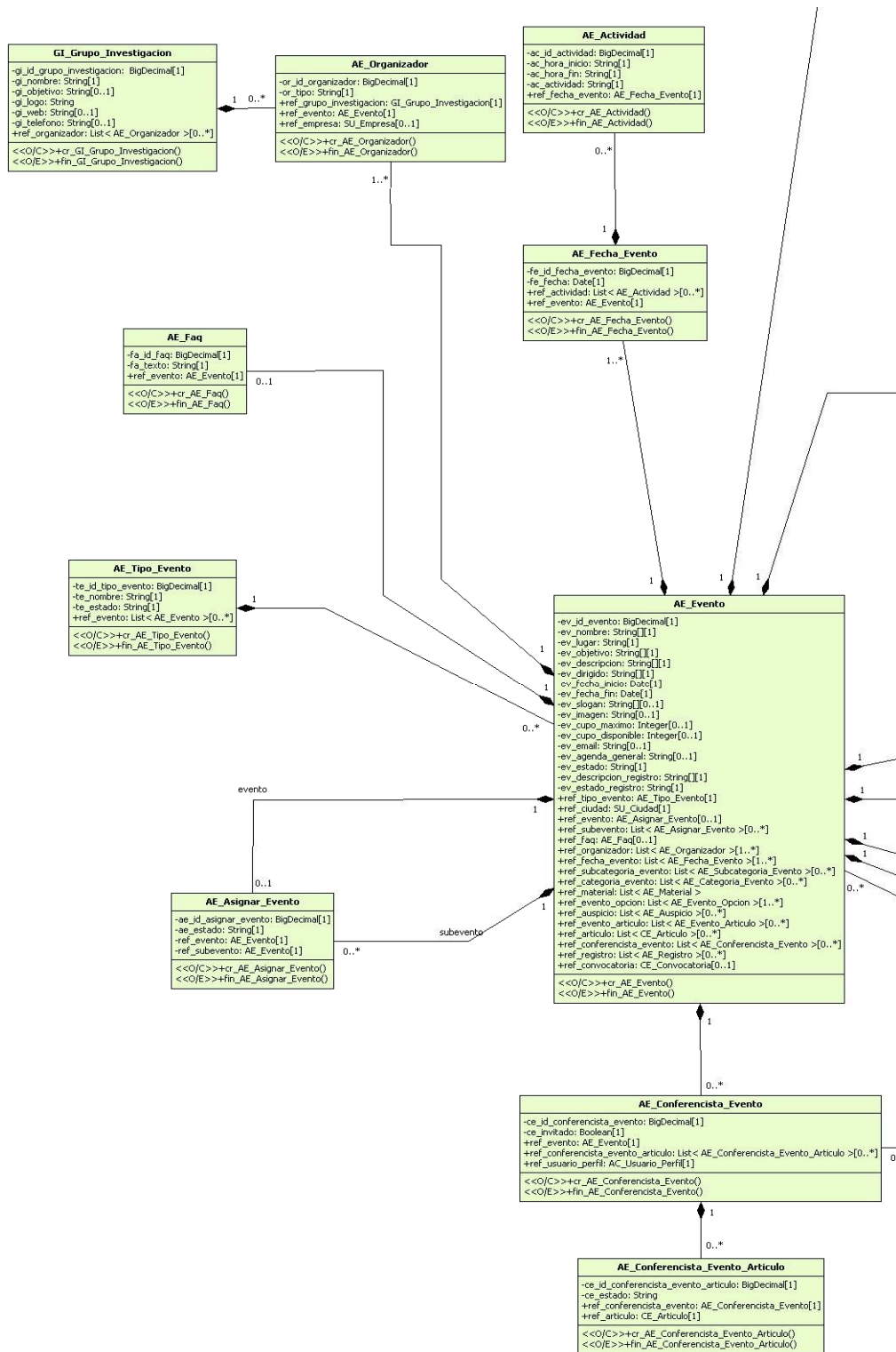


Gráfico D.3 Diagrama de Clases - Módulo de Administración de Eventos (MAE) (Parte I)

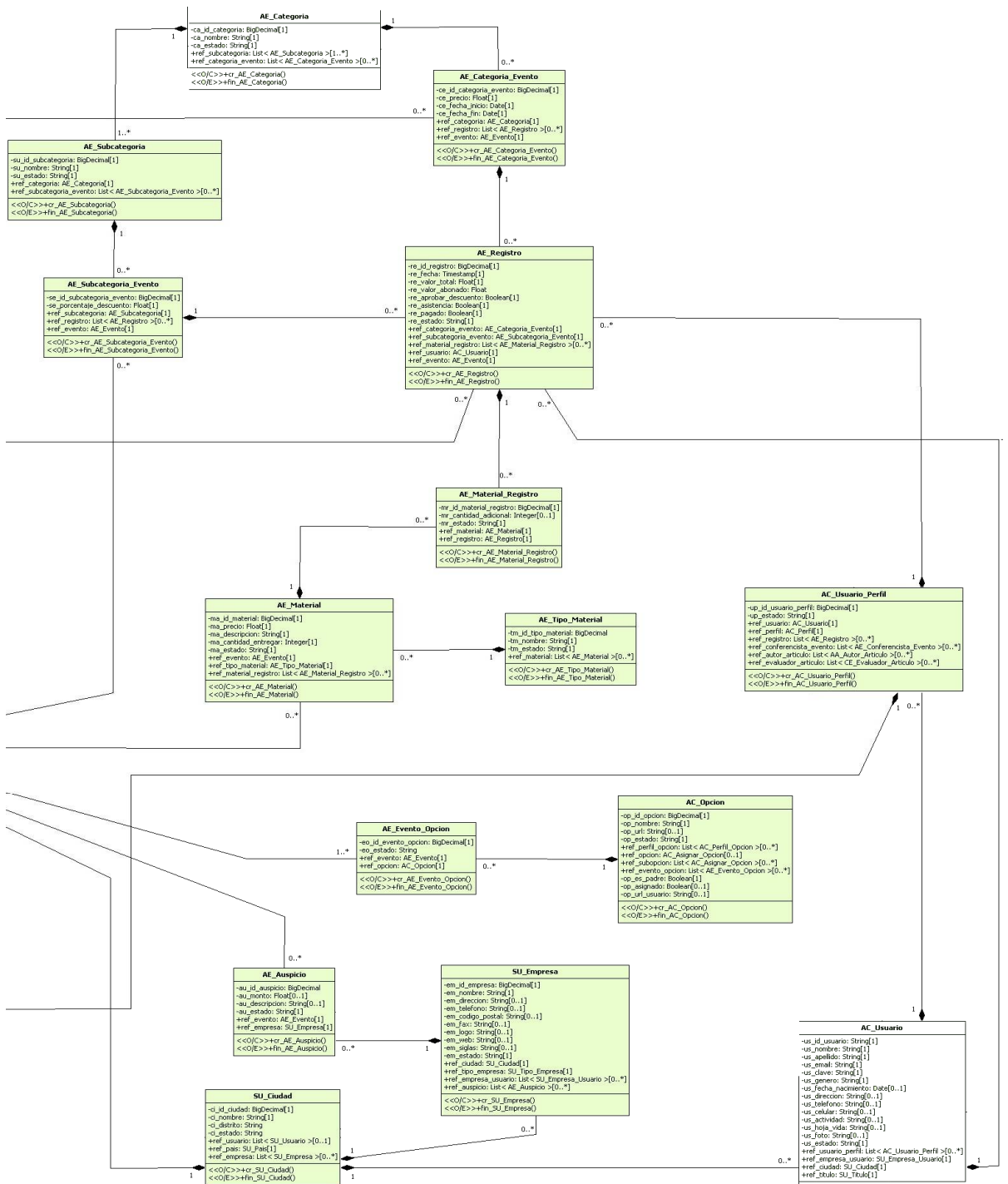


Gráfico D.3 Diagrama de Clases - Módulo de Administración de Eventos (MAE) (Parte II)

# APÉNDICE E: Diagrama del Modelo Lógico de la Base de Datos por Módulos

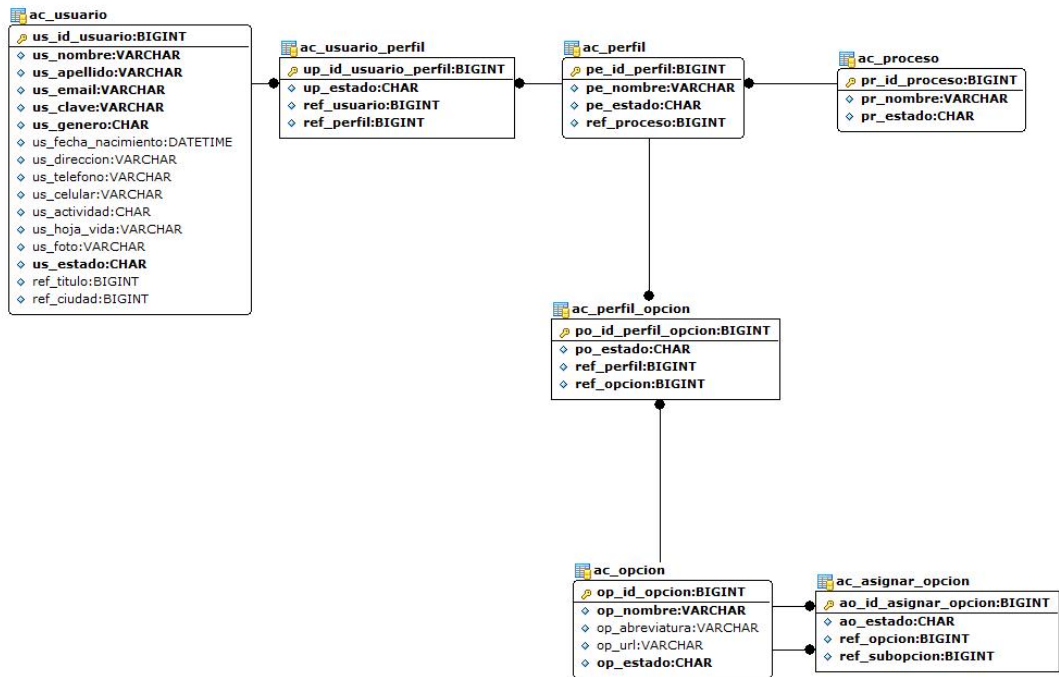
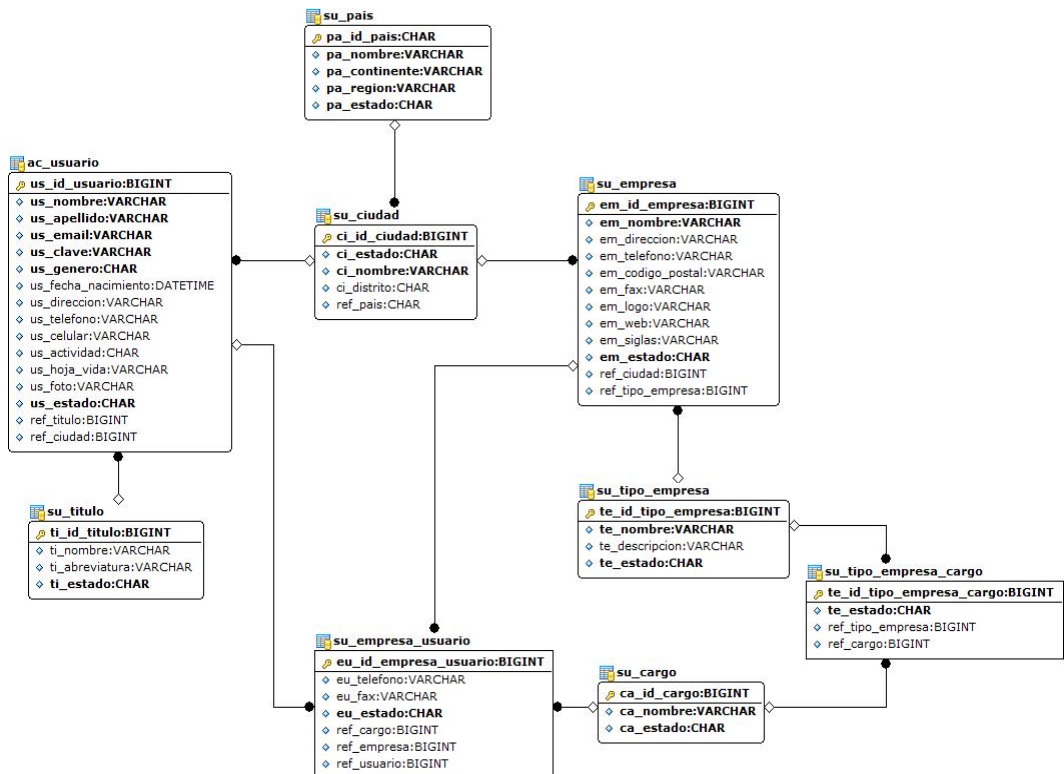


Gráfico E.1 ER – Módulo de Administración Central (MAC)



**Gráfico E.2 ER – Módulo de Suscripción de Usuarios (MSU)**







## APÉNDICE F: Pruebas Unitarias

**Tabla F.1. Prueba Unitaria N°: 01**

<b>Caso de Uso:</b> Ingresar al Sistema		<b>N°:</b> 01	
<b>Escenario:</b> El ingreso se realiza utilizando correo y contraseña válidos			
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 26-Nov-2009	
<b>Precondiciones</b>	Tiene que existir un usuario creado en el sistema para poder ingresar.		
<b>Datos de Entrada</b>	Correo: "vanejoha@gmail.com" Contraseña: "vanessa"		
<b>Descripción de Pasos</b>	Se ingresa el correo "vanejoha@gmail.com" en el campo Correo. Se ingresa la palabra "vanessa" en el campo Contraseña. Se presiona el botón "Ingresar" o apretar la tecla Enter.		
<b>Resultado Esperado</b>	Se ingresa al sistema como el usuario dado, se presenta la pantalla siguiente con el menú.	<b>Cumplimiento</b>	<b>SÍ</b>
			<b>NO</b>
		X	
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>	
<b>Recomendación u Observación</b>	El formato de correo es validado mediante la librería de DOJO, deberá ingresarse un correo válido. La contraseña es sensible a mayúsculas/minúsculas.		

**Tabla F.2. Prueba Unitaria N°: 02**

<b>Caso de Uso:</b> Crear Cuenta	<b>N°:</b> 02
<b>Escenario:</b> La creación de la cuenta se la realizará utilizando datos correctos	

<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 26-Nov-2009		
<b>Precondiciones</b>	Se debe ingresar todos los campos obligatorios en formato correcto.			
<b>Datos de Entrada</b>	Correo Electrónico: "correo_electronico@hotmail.com" Contraseña: "contraseña" Re-Contraseña: "contraseña" Nombres: "Nombre1 Nombre2" Apellidos: "Apellido1 Apellido2" Género: "Masculino" Dirección: "" Teléfono: "" Celular: "" Día de Nacimiento: "" País: "Ecuador" Ciudad: "Guayaquil" Carrera o Título: "Ing. en Computación" Tipo Empresa: "Pública" Cargo: "Estudiante" Empresa/Universidad: "ESPOL" Teléfono (Oficina): ""			
<b>Descripción de Pasos</b>	Dar clic en Crear Cuenta debajo de Ingresar al Sistema o en el Menú de Opciones Ingresar los valores en los campos como en Datos de Entrada Dar clic en "Crear Cuenta"			
<b>Resultado Esperado</b>	Se crea correctamente la cuenta, muestra el mensaje correspondiente y es re direccionado a la página de inicio.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
			X	
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>	Los campos que son numéricos son validados en tiempo de ejecución mediante la librería de DOJO			

**Tabla F.3. Prueba Unitaria N°: 03**

<b>Caso de Uso:</b> Crear una convocatoria		<b>N°:</b> 03		
<b>Escenario:</b> Crear una convocatoria con intervalos de fechas correctamente ingresadas				
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009		
<b>Precondiciones</b>	Haber ingresado en el sistema como administrador.			
<b>Datos de Entrada</b>	Fecha de Presentación: "2009-12-13" Fecha de Evaluación: "2009-12-29" Fecha de Notificación: "2010-01-02" Fecha de Corrección: "2010-01-09" Descripción (ES): "Llamada de artículos para este evento." Descripción (EN): "Call for papers for this event." Descripción (PT): "Chamada de trabalhos para este evento."			
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el nombre del evento Dar clic en la opción del menú "Convocatoria" Ingresar los datos descritos en Datos de Entrada Dar clic en el botón "Examinar" para subir un formato de archivo. Dar clic en el botón "Subir" Dar clic en el botón "Actualizar"			
<b>Resultado Esperado</b>	Se actualiza exitosamente la convocatoria, se muestra el mensaje correspondiente.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
			X	
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla F.4. Prueba Unitaria N°: 04**

<b>Caso de Uso:</b> Crear un evento		<b>N°:</b> 04		
<b>Escenario:</b> Creación de un evento ingresando solo los campos obligatorias con datos correctos				
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009		
<b>Precondiciones</b>	Haber ingresado en el sistema como administrador. El evento a crear no debe ya estar creado.			
<b>Datos de Entrada</b>	Tipo de Evento: "Jornadas" Evento (ES): "Nueva Evento de Prueba" Evento (EN): "New Test Event" Evento (PT): "Novo teste de evento" Fecha Inicio: "2009-12-18" Fecha Fin: "2009-12-24" Estado de Registro: "Activo"			
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el icono de Nuevo. Se abre la ventana de popup para ingresar los datos del evento. Ingresar los datos detallados en Datos de Entrada. Dar clic en el botón "Guardar".			
<b>Resultado Esperado</b>	Se crea exitosamente el evento, se muestra el mensaje correspondiente, se cierra el popup y se actualiza la tabla de los eventos.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
			X	
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

**Tabla F.5. Prueba Unitaria N°: 05**

<b>Caso de Uso:</b> Crear una evaluación		<b>N°:</b> 05		
<b>Escenario:</b> Crear/Actualizar la información de evaluación para los artículos				
<b>Responsable:</b> Vanessa Izquierdo		<b>Fecha:</b> 28-Nov-2009		
<b>Precondiciones</b>	Haber ingresado en el sistema como administrador. El tema no debe existir en la convocatoria de ese evento.			
<b>Datos de Entrada</b>	Tema: "Tema nuevo"			
<b>Descripción de Pasos</b>	Ir a la opción Administración → Eventos → Administrar Eventos Dar clic en el nombre del evento Dar clic en la opción del menú "Convocatoria" Dar clic en el botón "Evaluación" <i>Se re direccionará a la página de evaluación</i> En el campo Mensaje al Evaluador ingresar "Gracias por ayudarnos a la evaluación". En el campo Instrucciones ingresar "Estas son las instrucciones a seguir". En la campo Estado seleccionar "Vigente" Dar clic en el botón "Actualizar"			
<b>Resultado Esperado</b>	Se crea/actualiza la evaluación, se muestra en mensaje correspondiente.	<b>Cumplimiento</b>	<b>SÍ</b>	<b>NO</b>
			X	
<b>Resultado Obtenido</b>	<b>Errores:</b>	<b>Fallas Provocadas:</b>		
<b>Recomendación u Observación</b>				

## REFERENCIAS

[1] Verónica Macías. “Modelamiento basado en el dominio: Estado del Arte”, en memorias de las I Jornadas de Ingeniería de Software, Guayaquil-Ecuador, Noviembre 2004.

[2] Richard Soley and the OMG Staff Strategy Group. “Model Driven Architecture”, Object Management Group. November-2000.

[3] M. Snoeck, G. Dedene, M. Verhelst, A. Depuydt, “Object-Oriented Enterprise Modelling with MERODE”, Leuven University Press, 1999 pp. 12-16.

[4] Snoeck M., Michels C., and Dedene G., “Consistency by construction: the case of MERODE”, Conceptual Modeling for Novel Application Domains, ER 2003 Workshops ECOMO, ICWMQ, AOIS and XSDM Proceedings, October 2003, p. 105-117.

[5] Sun Developer Network, “Java 2 Platform, Enterprise Edition (J2EE) Overview”. [En línea]. Disponible: <http://java.sun.com/j2ee/overview.html>. [Accedido: Nov. 9, 2009]

**[6]** Ciberaula, “Introducción a JSTL”. [En línea]. Disponible: [http://java.ciberaula.com/articulo/introduccion\\_jstl/](http://java.ciberaula.com/articulo/introduccion_jstl/) [Accedido: Nov. 10, 2009]

**[7]** Wikipedia, “JSON”. [En línea]. Disponible: <http://es.wikipedia.org/wiki/JSON> [Accedido: Nov. 10, 2009]

**[8]** Java en Castellano, “Migrar de JSP a Velocity”. [En línea]. Disponible: [http://www.programacion.com/java/articulo/jap\\_velocity/](http://www.programacion.com/java/articulo/jap_velocity/) [Accedido: Nov. 10, 2009]

**[9]** Wikipedia, “Dojo toolkit”. [En línea]. Disponible: [http://es.wikipedia.org/wiki/Dojo\\_toolkit](http://es.wikipedia.org/wiki/Dojo_toolkit) [Accedido: Nov. 11, 2009]

**[10]** Wikipedia, “Yahoo! UI Library”. [En línea]. Disponible: [http://es.wikipedia.org/wiki/Yahoo\\_User\\_Interface](http://es.wikipedia.org/wiki/Yahoo_User_Interface) [Accedido: Nov. 15, 2009]

**[11]** Wikipedia, “MySQL”. [En línea]. Disponible: <http://es.wikipedia.org/wiki/MySQL> [Accedido: Nov. 16, 2009]

**[12]** Alternativa de código abierto, “StarUML 5.0”. [En línea]. Disponible:  
<http://www.osalt.com/es/staruml> [Accedido: Nov. 20, 2009]