



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PROYECTO DE GRADO

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

**Previa a la Obtención del Título de Ingeniero en Computación
especialización Sistemas Multimedia**

**PRESENTADA POR:
Guillermo Omar Pizarro Vásquez
Rafael Eduardo Rivadeneira Campodonico**

GUAYAQUIL ECUADOR

2009

AGRADECIMIENTO

*A Dios,
a mis padres y
a mis profesores*

DEDICATORIA

*Con cariño,
para mis padres*

TRIBUNAL DE GRADO

PRESIDENTE

Ing. Jorge Aragundi

DIRECTOR DE TESIS

Ing. Mónica Villavicencio

MIEMBROS PRINCIPALES

M.Sc.

M.Sc.

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Guillermo Omar Pizarro Vásquez

Rafael Eduardo Rivadeneira Campodonico

RESUMEN

El presente Proyecto de Grado 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 AppVlir8¹, 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 cuya necesidad se ha identificado durante el tiempo en que el software ha estado en producción (a partir de Septiembre del 2006).

En el primer capítulo se señalan los antecedentes de retomar este Proyecto, el hecho de que MERODE si ayuda en el diseño independiente del dominio, vital para seguir la metodología MDA, aunque se va a usar la misma tecnología J2EE, utilizada en AppVlir8, pero con la especificación JSP 2.0 siguiendo el patrón de diseño MVC 2, los objetivos que se espera alcanzar y la justificación del desarrollo de los módulos.

¹<http://www.vlir8.espol.edu.ec>

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 anterior Sistema. 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

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE GRADO	iv
DECLARACIÓN EXPRESA	v
RESUMEN	vi
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	x
INDICE DE TABLAS	xi
 ANTECEDENTES	 1
<i>Antecedentes</i>	1
<i>Justificación</i>	1
<i>Objetivos</i>	2
FUNDAMENTOS TEÓRICOS	3
<i>Descripción General de MDA y MERODE</i>	3
<i>Tecnologías Open Source</i>	3
Uso de la arquitectura J2EE.	4
Librerías en el Servidor: JSTL, JSON y Velocity.	4
Librerías en el Cliente: Dojo y YUI.	4
Base de Datos: MySQL y HSQLDB.	4
Uso de la herramienta CASE: StarUML.	4
ANÁLISIS DEL SISTEMA	5
<i>Requerimientos Funcionales</i>	5
<i>Requerimientos No Funcionales</i>	7
<i>Especificación del Modelo Independiente de la Plataforma (PIM)</i> <i>usando técnicas de MERODE</i>	9
El gráfico de la dependencia de existencia (EDG)	10
La tabla de eventos de objetos (OET)	22
Máquina de estados finitos relevantes (FSM)	22
<i>Definición del Modelo específico de la plataforma (PSM)</i>	22
<i>Definición de Roles en los Módulos</i>	33
DISEÑO DEL SISTEMA	35
<i>Diseño de la Arquitectura del Sistema</i>	35
Diseño Arquitectónico	37
Módulos del Sistema	39
Arquitectura basada en patrones de diseño	39
<i>Diagrama de Clases del Sistema</i>	39
<i>Modelo lógico de la Base de Datos</i>	39
IMPLEMENTACIÓN Y PRUEBAS	40
<i>Capas del Sistema y comunicación entre capas</i>	40
Implementación basada en el análisis con MERODE y MDA	41
<i>Plan de Pruebas</i>	41
<i>Resultados de las Pruebas y Métricas tomadas</i>	41

CONCLUSIONES	42
<i>Comentarios generales</i>	42
<i>REFERENCIAS</i>	44

ÍNDICE DE FIGURAS

Gráfico 1.EDG - Sistema Web para la Administración de Eventos Científicos (WebSAE)	11
Gráfico 2.EDG - Módulo de Administración Central (MAC)	13
Gráfico 3.EDG - Módulo de Suscripción de Usuarios (MSU)	15
Gráfico 4.EDG - Módulo de Administración de Eventos (MAE)	18
Gráfico 5.EDG - Módulo de Convocatoria y Evaluación de Artículos (MCE)	21
Gráfico 6.OET - Módulo de Administración Central (MAC)	24
Gráfico 7.OET - Módulo de Suscripción de Usuarios (MSU)	25
Gráfico 8.OET - Módulo de Administración de Eventos (MAE)	26
Gráfico 9.OET - Módulo de Convocatoria y Evaluación de Artículos (MCE)	27
Gráfico 10.FSM - Estados del Objeto AE_Evento	28
Gráfico 11.FSM - Estados del Objeto AE_Registro	28
Gráfico 12.UML - Módulo de Administración Central (MAC)	29
Gráfico 13.UML - Módulo de Suscripción de Usuarios (MSU)	30
Gráfico 14.UML - Módulo de Administración de Eventos (MAE)	31
Gráfico 15.UML - Módulo de Convocatoria y Evaluación de Artículos (MCE)	32
Gráfico 1.Diseño de Arquitectura del Sistema de Administración de Eventos y Grupos Científicos.	36
Gráfico 2.Diseño de la Arquitectura del Sistema de Administración de Eventos.	38

INDICE DE TABLAS

Tabla 3. 1. Diferencias entre los objetos de WebSAE y AppVlir8 , en MAC	13
Tabla 3. 2. Diferencias entre los objetos de WebSAE y AppVlir8 , en MSU	15
Tabla 3. 3. Diferencias entre los objetos de WebSAE y AppVlir8 , en MAE	17
Tabla 3. 4. Diferencias entre los objetos de WebSAE y AppVlir8 , en MCE	20
Tabla 3. 5. SEI - Objeto AE Evento (Parte 1)	22
Tabla 3. 6. SEI - Objeto AE Evento (Parte 2)	24
Tabla 3. 7. Definición de Roles en los Módulos	34

CAPÍTULO I.

ANTECEDENTES

Antecedentes

Justificación

Objetivos

CAPÍTULO II.

FUNDAMENTOS TEÓRICOS

Descripción General de MDA y MERODE

Tecnologías Open Source

Uso de la arquitectura J2EE.

Librerías en el Servidor: JSTL, JSON y Velocity.

Librerías en el Cliente: Dojo y YUI.

Base de Datos: MySQL y HSQLDB.

Uso de la herramienta CASE: StarUML.

CAPÍTULO III.

ANÁLISIS DEL SISTEMA

Requerimientos Funcionales

De manera general, los requerimientos que se han pedido para mejorar el Portal son los siguientes:

- Añadir el módulo de pago en línea;
- Dar soporte para varios idiomas;
- Mostrar resúmenes de evaluaciones, con los siguientes datos: recomendación y exportación de los datos ingresados por los evaluadores;

- Implementar requerimientos adicionales pedidos por el Sub - Componente de Telecomunicaciones, tales como:
 - Manejar copias ocultas para emails de rechazo de artículos, es decir, el correo electrónico debe enviarse con copia al correo electrónico del evento y con copia oculta al administrador que rechaza el artículo;
 - 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- al artículo científico final que se va a evaluar;
 - Se debe poder modificar un artículo sin tener que borrar y volverlo a subir;
 - Después de evaluado un artículo, se debe seguir mostrando -a los administradores del evento- quienes fueron sus evaluadores;
 - El registro del usuario debe ser internacional y no sólo dirigido a nuestro país;
 - Corregir lo siguiente: un autor no debe poder evaluar su propio artículo, los autores secundarios también deben constar en la lista de un artículo, y en el manejo de sesiones (a veces se sale del sistema y se desea ingresar con clave errónea, el sistema lo permite);
 - Generar los siguientes reportes: trabajos por autores, eva-

luadores asignados a cada artículo, resúmenes de artículos clasificados con su respectiva nota y el estado de cada artículo. Exportar los reportes a los formatos: CSV y pdf;

- Adjuntar los artículos de derecho de traspaso de los artículos aceptados;
- Permitir que el administrador pueda eliminar un artículo;
- Ocultar clave de la lista de los usuarios;
- Manejar un email de contacto para cada evento diferente;
- Guardar la fecha de publicación de un artículo, cuando se lo suba al Sistema;
- En los registros: cuando un usuario realice un pago en efectivo, permitir ingresar el valor que pagó; en el pre registro notificar al usuario la fecha límite para realizar el pago correspondiente;
- Poder escoger el autor de un paper en caso de que una persona alterna suba el artículo por el autor.

Requerimientos No Funcionales

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

- Manejo más apropiado de IHM¹: 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;
- 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.

¹Interacción Hombre - Máquina.

- 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.

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 un futuro próximo, en nuestro caso deseamos diagramar un PIM (Platform Independent Model) mediante un EDG, fue diseñado mediante la herramienta llamada MERMAID².

Es necesario mencionar, el hecho de haber diagramado en un solo PIM el dominio de la administración de usuarios, la administración de eventos, la convocatoria y la evaluación de artículos. Al realizar esta actividad de revisión y rediseño, se ha pensado de manera sistemática la diagramación de un solo PIM en varios módulos.

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

El gráfico de la dependencia de existencia (EDG)

A continuación se muestran los Gráficos de la Dependencia de existencia o EDG, como sistema (Gráfico 1) y luego separado por módulos para un mejor estudio de ellos.

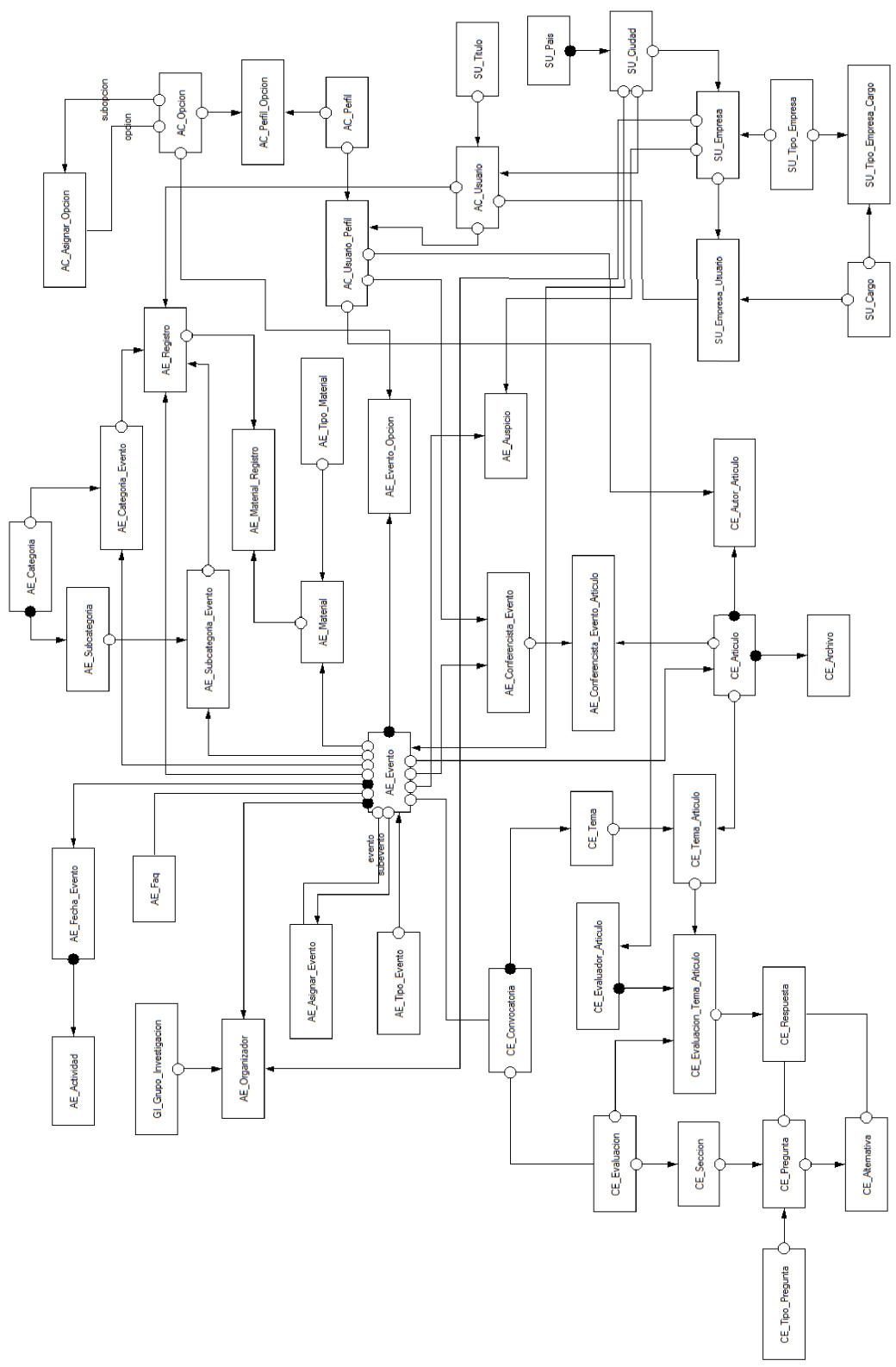


Gráfico 1 : EDG - Sistema Web para la Administración de Eventos Científicos (WebSAE)

Se necesitaron un total de 48 objetos para implementar el Sistema actual, que ha sido denominado como **WebSAE**. En el Sistema original, denominado como **AppVlir8**, se diseñó con 42 objetos en total, es decir, que en el actual se han aumentado 6 objetos.

A continuación se enumerarán los objetos propios de cada módulo con el correspondiente EDG:

MÓDULO DE ADMINISTRACIÓN CENTRAL (MAC)

MAC (Gráfico 2) tiene 6 objetos que se listarán a continuación:

1. AC_Usuario
2. AC_Usuario_Perfil
3. AC_Perfil
4. AC_Perfil_Opcion
5. AC_Opcion
6. AC_Asignar_Opcion

Es necesario enfatizar que **MAC** surge en el momento de pensar en la Arquitectura de **WebSAE**, la mitad de los objetos son nuevos, a continuación la siguiente Tabla 3. 1 nos explica las diferencias entre los objetos, dependiendo si es el Sistema actual o el anterior.

WebSAE	AppVlir8	Observaciones
AC_Usuario_Perfil	Rol_Usuario	
AC_Perfil	Rol	
AC_Perfil_Opcion	Menu	Objeto que asigna las opciones, dependiendo del perfil que se haya deseado crear.
AC_Opcion	Opcion_Menu	Son las diversas opciones que se le presentan al usuario, dependiendo del perfil que tenga asignado.
AC_Asignar_Opcion	Submenu	Este objeto nos sirve para manejar la recursión entre las opciones, debido a que normalmente existen subopciones entre las opciones ³ .

Tabla 3. 1: Diferencias entre los objetos de **WebSAE** y **AppVlir8**, en **MAC**.

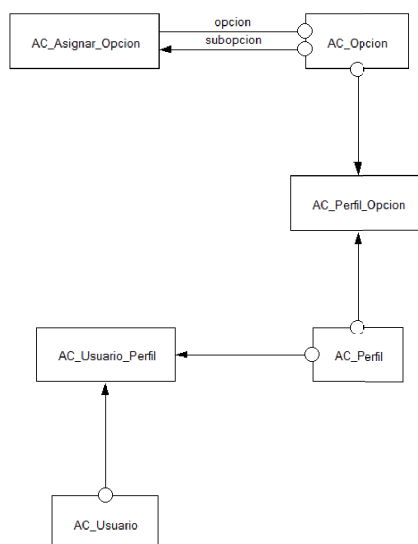


Gráfico 2: EDG - Módulo de Administración Central (MAC)

MÓDULO DE SUSCRIPCIÓN DE USUARIOS (MSU)

MSU (Gráfico 3) tiene 8 objetos que se listarán a continuación:

1. SU_Titulo
2. SU_Pais
3. SU_Ciudad
4. SU_Empresa
5. SU_Tipo_Empresa
6. SU_Empresa_Usuario
7. SU_Tipo_Empresa_Cargo
8. SU_Cargo

En la siguiente Tabla 3. 2 nos explica las diferencias entre los objetos, dependiendo si es el Sistema actual o el anterior.

MÓDULO DE ADMINISTRACIÓN DE EVENTOS (MAE)

MAE (Gráfico 4) tiene 19 objetos que se listarán a continuación:

1. AE_Actividad
2. AE_Asignar_Evento
3. AE_Auspicio

WebSAE	AppVlir8	Observaciones
SU_Tipo_Empresa_Cargo		Este objeto surgió debido a que anteriormente Cargo dependía de manera directa de Tipo_Empresa, en la revisión del diseño se cambió esa relación agregando un objeto, de tal manera que ahora Cargo y Tipo de Empresa son independientes.

Tabla 3. 2: Diferencias entre los objetos de **WebSAE** y **AppVlir8**, en **MSU**.

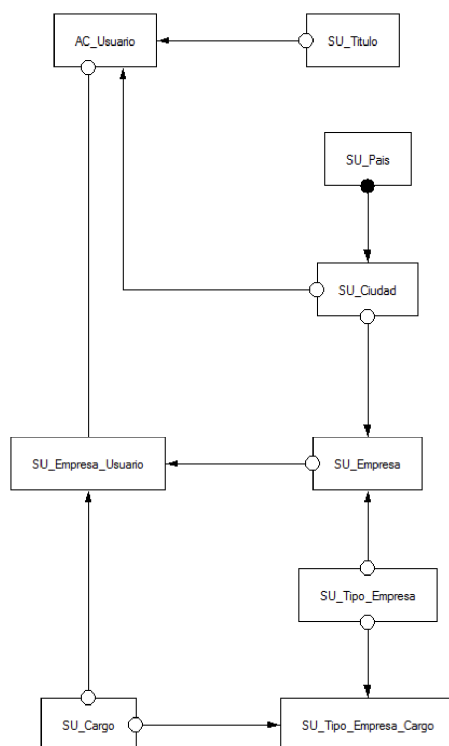


Gráfico 3: EDG - Módulo de Suscripción de Usuarios (MSU)

4. AE_Categoria
5. AE_Categoria_Evento
6. AE_Conferencista_Evento
7. AE_Conferencista_Evento_Articulo
8. AE_Evento
9. AE_Evento_Opcion
10. AE_Faq
11. AE_Fecha_Evento
12. AE_Material
13. AE_Material_Registro
14. AE_Organizador
15. AE_Registro
16. AE_Subcategoria
17. AE_Subcategoria_Evento
18. AE_Tipo_Evento
19. AE_Tipo_Material

En la siguiente Tabla 3. 3 nos explica las diferencias entre los objetos, dependiendo si es el Sistema actual o el anterior.

WebSAE	AppVlir8	Observaciones
AE_Asignar_Evento	Subevento_Evento	
AE_Auspicio	Auspicio	Además de la relación existente con una Empresa, también se le agregó la relación con un Grupo de Investigación, debido a que también pueden ser auspiciantes sin ser los Organizadores.
AE_Categoria_Evento	Categoria_Evento	Es el precio dependiendo de la Categoría.
AE_Subcategoria_Evento	Subcategoria_Evento	Es el porcentaje dependiendo de la Subcategoría.
AE_Conferencista_Evento	Conferencista_Evento	Este objeto en el anterior Sistema estaba en MCE , pero debido a que muchas veces existen conferencistas que son invitados y no necesariamente entran en el proceso de evaluación de artículos, de ahí en la necesidad de cambiar de módulo a este objeto.
AE_Conferencista_Evento_Articulo	AE_Conferencista_Articulo	
AE_Evento_Opcion	Menu_Evento	
	Pregunta_Faq	Eliminado de WebSAE, debido a que en el objeto Faq, se guardan las preguntas en HTML.
	Respuesta_Faq	Eliminado de WebSAE, por la razón anterior.
	Correo	Eliminado de WebSAE.

Tabla 3. 3: Diferencias entre los objetos de **WebSAE** y **AppVlir8**, en **MAE**.

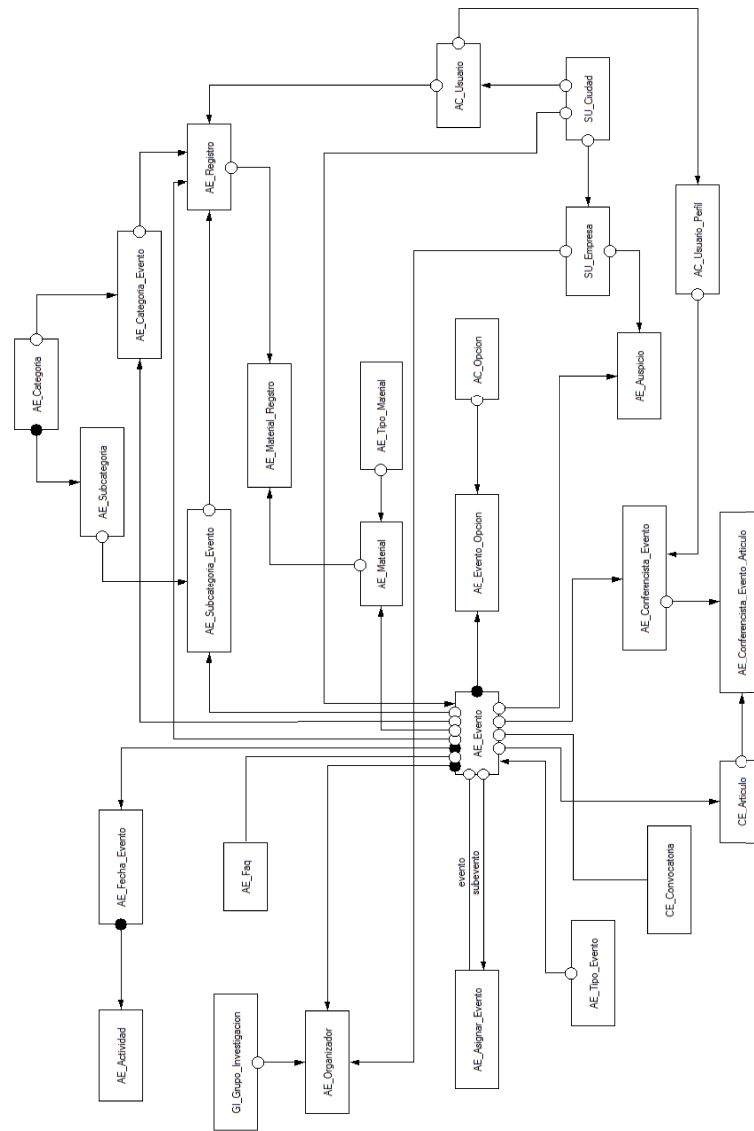


Gráfico 4: EDG - Módulo de Administración de Eventos (MAE)

MÓDULO DE CONVOCATORIA Y EVALUACIÓN DE ARTÍCULOS (MCE)

MCE (Gráfico 5) tiene 14 objetos que se listarán a continuación:

1. CE_Alternativa
2. CE_Archivo
3. CE_Articulo
4. CE_Autor_Articulo
5. CE_Convocatoria
6. CE_Evaluacion
7. CE_Evaluacion_Tema_Articulo
8. CE_Evaluador_Articulo
9. CE_Pregunta
10. CE_Respuesta
11. CE_Seccion
12. CE_Tema
13. CE_Tema_Articulo
14. CE_Tipo_Pregunta

En la siguiente Tabla 3. 4 nos explica las diferencias entre los objetos, dependiendo si es el Sistema actual o el anterior.

WebSAE	AppVlir8	Observaciones
CE_Archivo		Agregado debido a que se desea tener todos los archivos que se generen en el proceso de la evaluación por cada usuario que haya subido su artículo para participar de la convocatoria.
CE_Respuesta	Respuestas	
	Articulo_Evento	Eliminado de WebSAE, se incluyó una relación directa entre CE_Evento y CE_Articulo, en el que CE_Articulo depende de CE_Evento.

Tabla 3. 4: Diferencias entre los objetos de **WebSAE** y **AppVlir8**, en **MCE**.

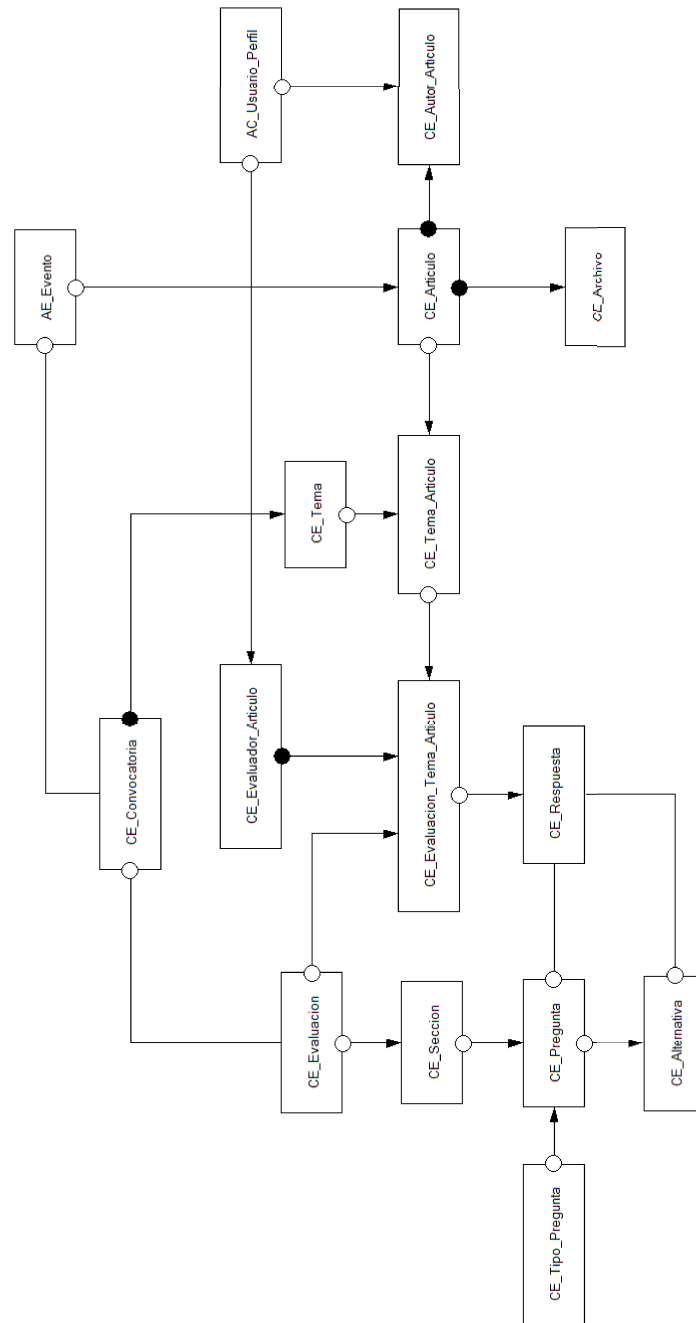


Gráfico 5: EDG - Módulo de Convocatoria y Evaluación de Artículos (MCE)

La tabla de eventos de objetos (OET)

Para presentar la Tabla de Objetos de Eventos⁴, se ha considerado separarla por objetos para que pueda ser apreciada de una mejor manera.

Máquina de estados finitos relevantes (FSM)

Definición del Modelo específico de la plataforma (PSM)

Luego de elaborar el correspondiente EDG, se pasó a la etapa del PSM, el cual fue diagramado mediante una herramienta Open Source, llamada StarUML⁵.

Los módulos del PIM anterior tan sólo fueron pasados al simil en UML, para su correspondiente transformación al código, en el momento debido.

⁴Es una matriz que contiene una fila por cada tipo de evento y una columna por cada tipo de objeto, como se ha detallado en el capítulo 2.

⁵<http://staruml.sourceforge.net/en/>

	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_subcategoria_subevento	A/M
modificar_subcategoria_subevento	A/M
eliminar_subcategoria_subevento	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

Tabla 3. 5: OET - Objeto AE_Evento (Parte 1)

	AE_Evento
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
eliminar_usuario_evento	A/M

Tabla 3. 6: OET - Objeto AE_Evento (Parte 2)

Gráfico 6: OET - Módulo de Administración Central (MAC)

Gráfico 7: OET - Módulo de Suscripción de Usuarios (MSU)

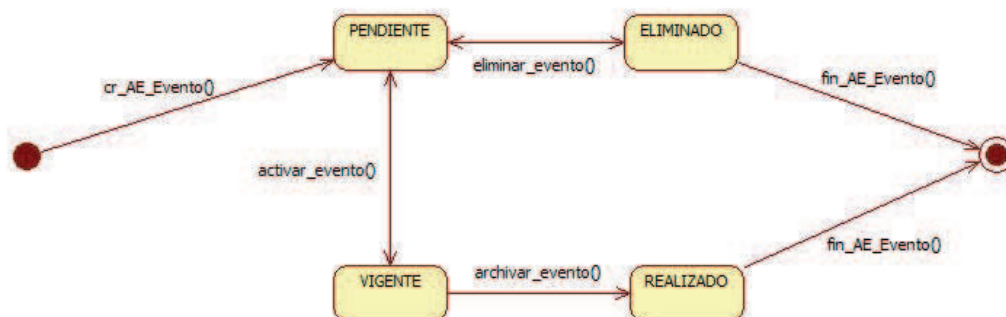


Gráfico 10: FSM - Estados del Objeto AE_Evento

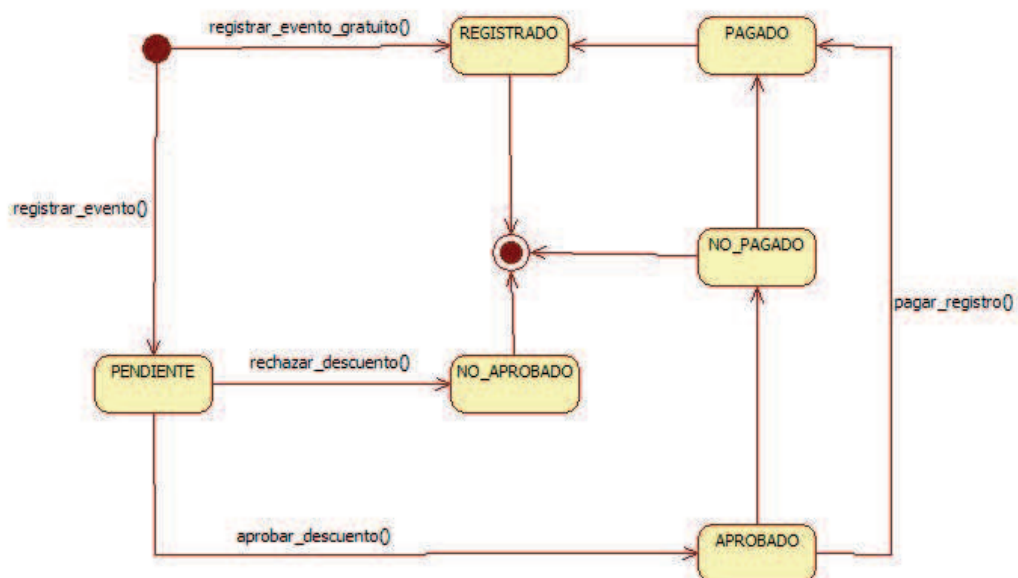


Gráfico 11: FSM - Estados del Objeto AE_Registro

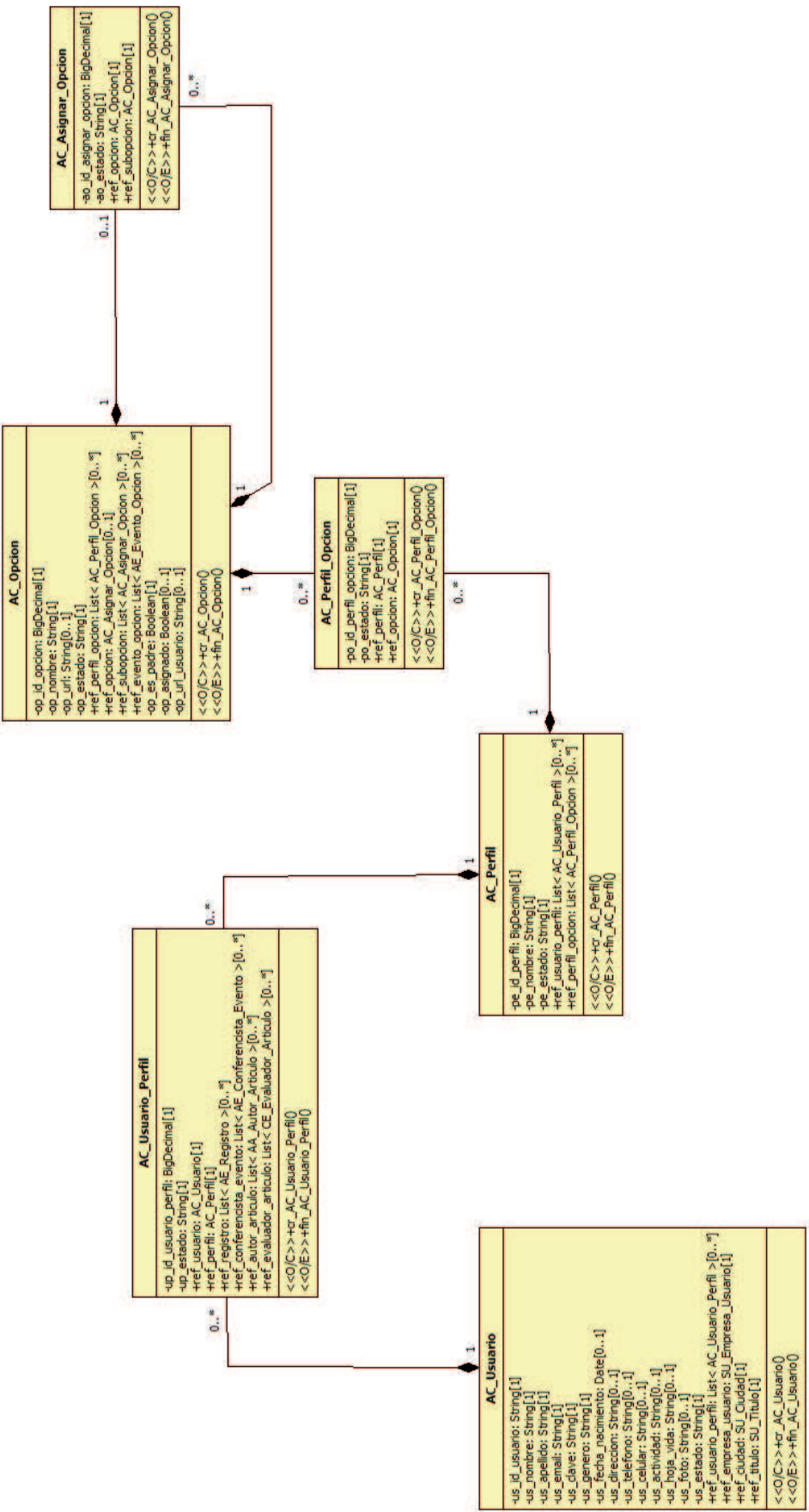


Gráfico 12: UML - Módulo de Administración Central (MAC)

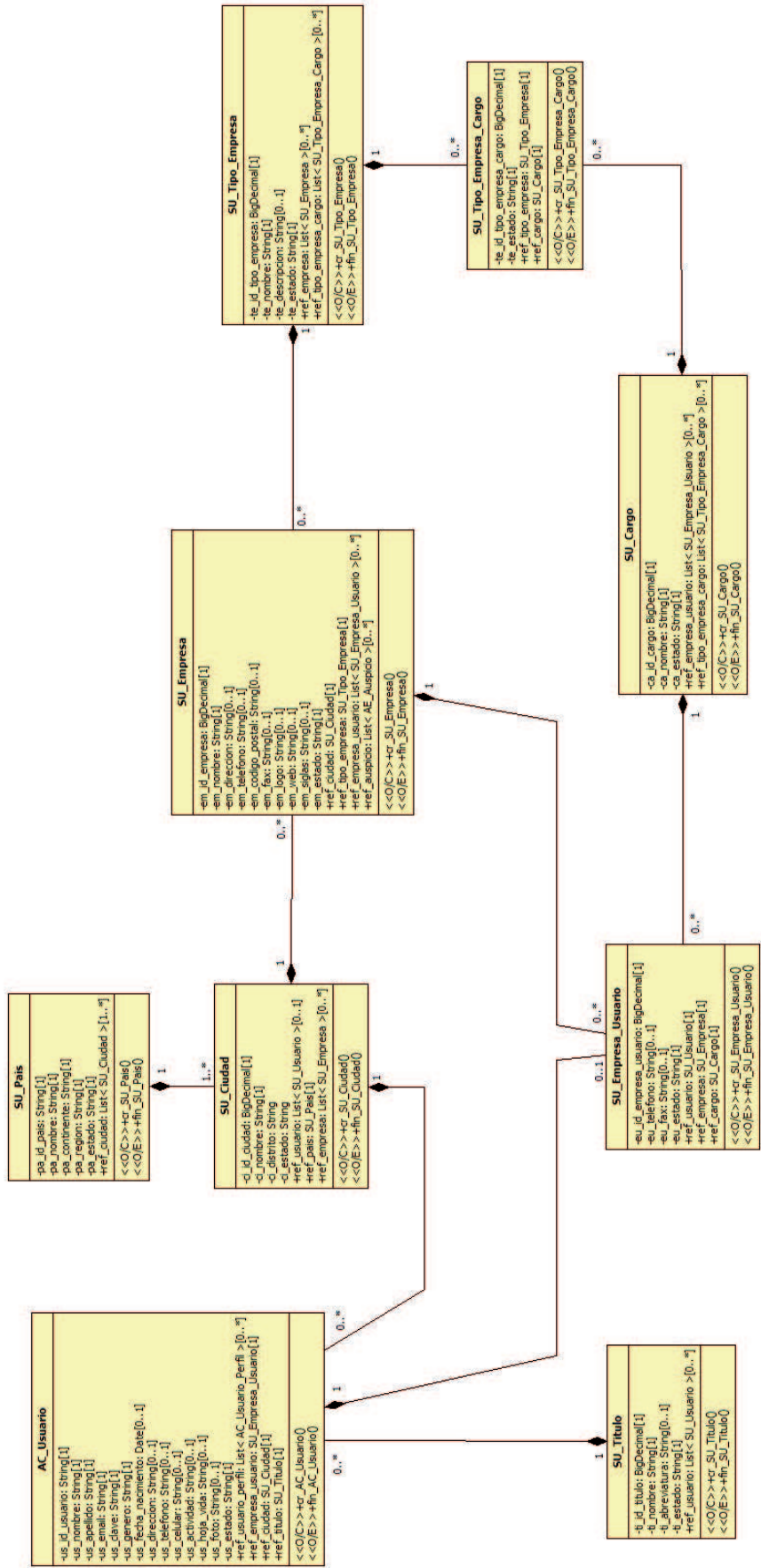


Gráfico 13: UML - Módulo de Suscripción de Usuarios (MSU)

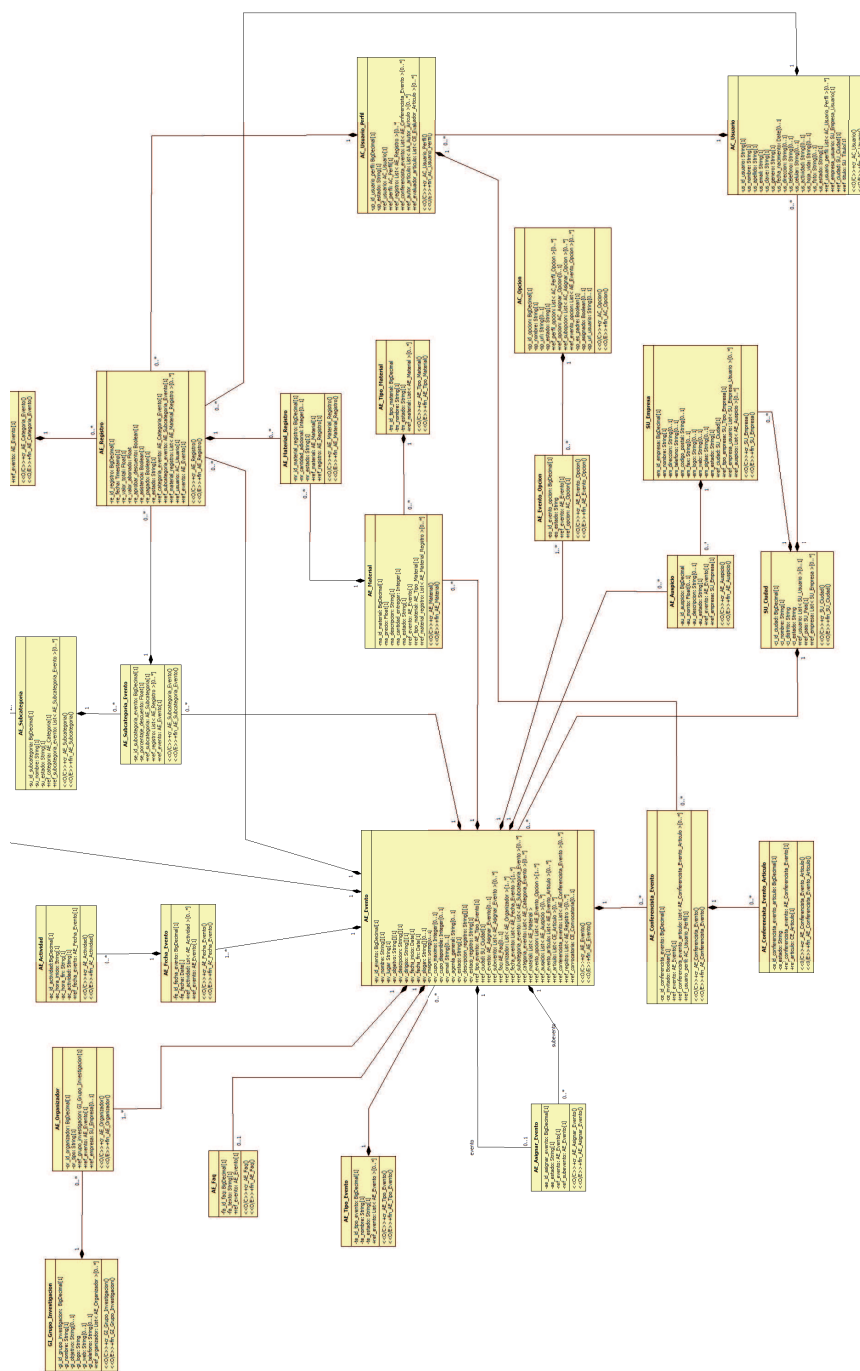


Gráfico 14: UML - Módulo de Administración de Eventos (MAE)

Definición de Roles en los Módulos

En la siguiente tabla Tabla 3. 7, 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.

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.

Usuario	Función dentro del negocio	Función dentro del Sistema
Director del Grupo de Investigación	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
Consejo Técnico Organizador	Consejo conformado por los organizadores, una de sus tareas es la de decidir qué resumen de un artículo científico -presentado por los posibles conferencistas- es válido para darle paso al siguiente proceso de evaluación.	Administrador
Asistente del Grupo de Investigación	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, sólo por razones prácticas.	Administrador General
Suscriptor	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
Evaluador	Participante del evento en calidad de evaluador, con el propósito de verificar la utilidad del conocimiento demostrado en el artículo.	Evaluador
Conferencista	Usuario que presentó un artículo de índole científico con un resultado exitoso, el conferencista también puede ser invitado por el Comité Organizador.	Conferencista

Tabla 3. 7: Definición de Roles en los Módulos

CAPÍTULO IV.

DISEÑO DEL SISTEMA

Diseño de la Arquitectura del Sistema

Debido al requerimiento no funcional, 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, el diseño de la Arquitectura fue concebido de la siguiente manera.

La definición de la anterior imagen Gráfico 1, es concebida como dos

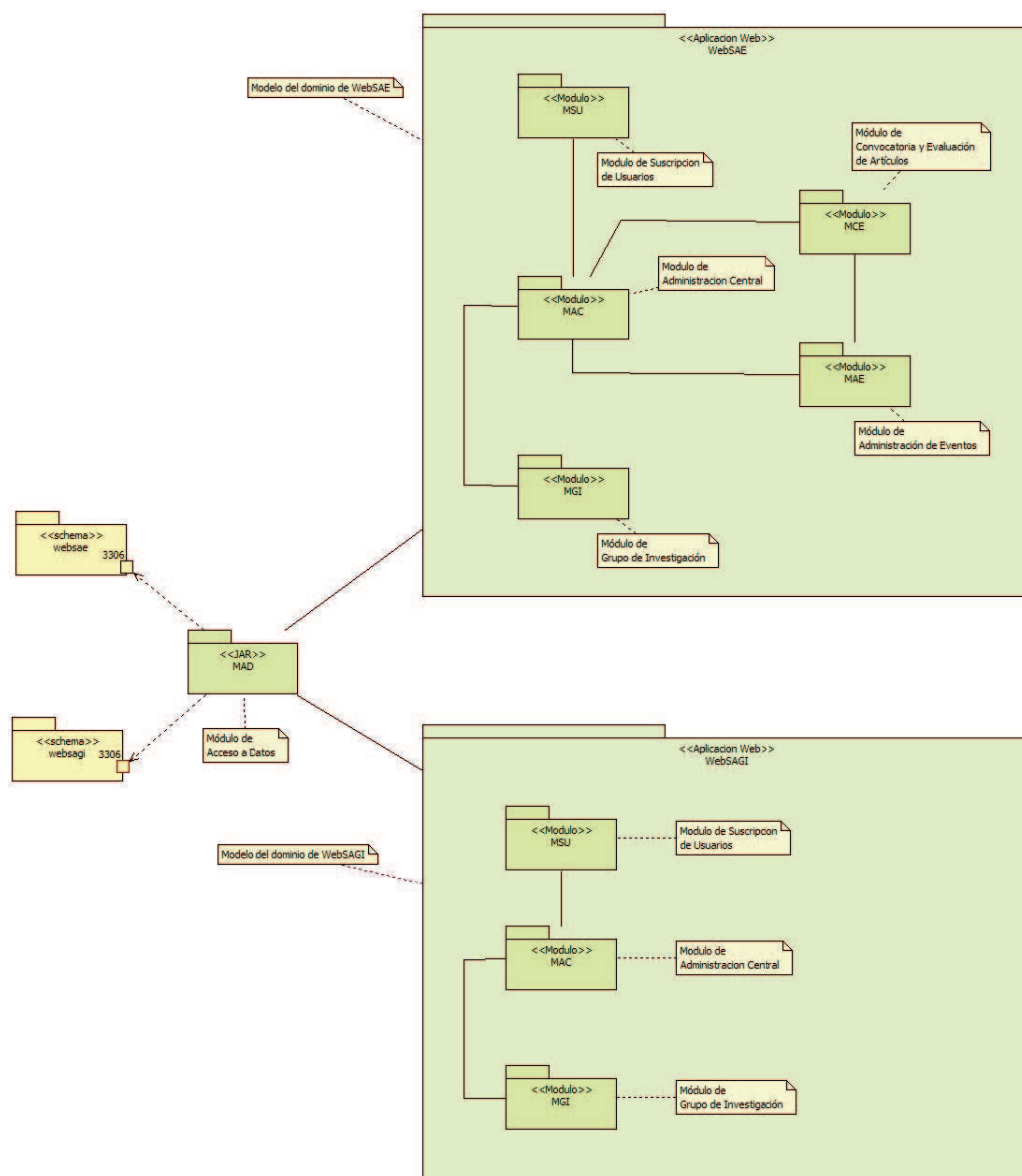


Gráfico 1: Diseño de Arquitectura del Sistema de Administración de Eventos y Grupos Científicos.

sistemas independientes que utilizan otro módulo no mencionado al inicio, debido a que este módulo no pertenece al dominio, si no que surge en el diseño de la Arquitectura, este módulo fue llamado MAD (Módulo de Acceso a Datos)¹ este framework, permite a los Eventos la conexión a la Base de Datos de los Grupos de Investigación y vice-verza.

Diseño Arquitectónico

En nuestro caso, sólo se implementó el Sistema Web de Administración de Eventos Científicos, también denominado **WebSAE**, el mismo que se muestra a continuación Gráfico 2, que sería tan sólo una de las partes antes mostrada.

¹Es importante recalcar que este módulo fue ideado por la Srta. Diana Crespo -jefa de Sistemas en Molemotor-, llamándolo inicialmente ADA, por el diseño portable de éste, se lo adecuó a MySQL -ya que fue implementado para que trabajase con la Base de Datos SQL Server 2000- y para que pueda trabajar con WebSAE y WebSAGI, las bases de datos de la Administración de Eventos y de los Grupos de Investigación, respectivamente.

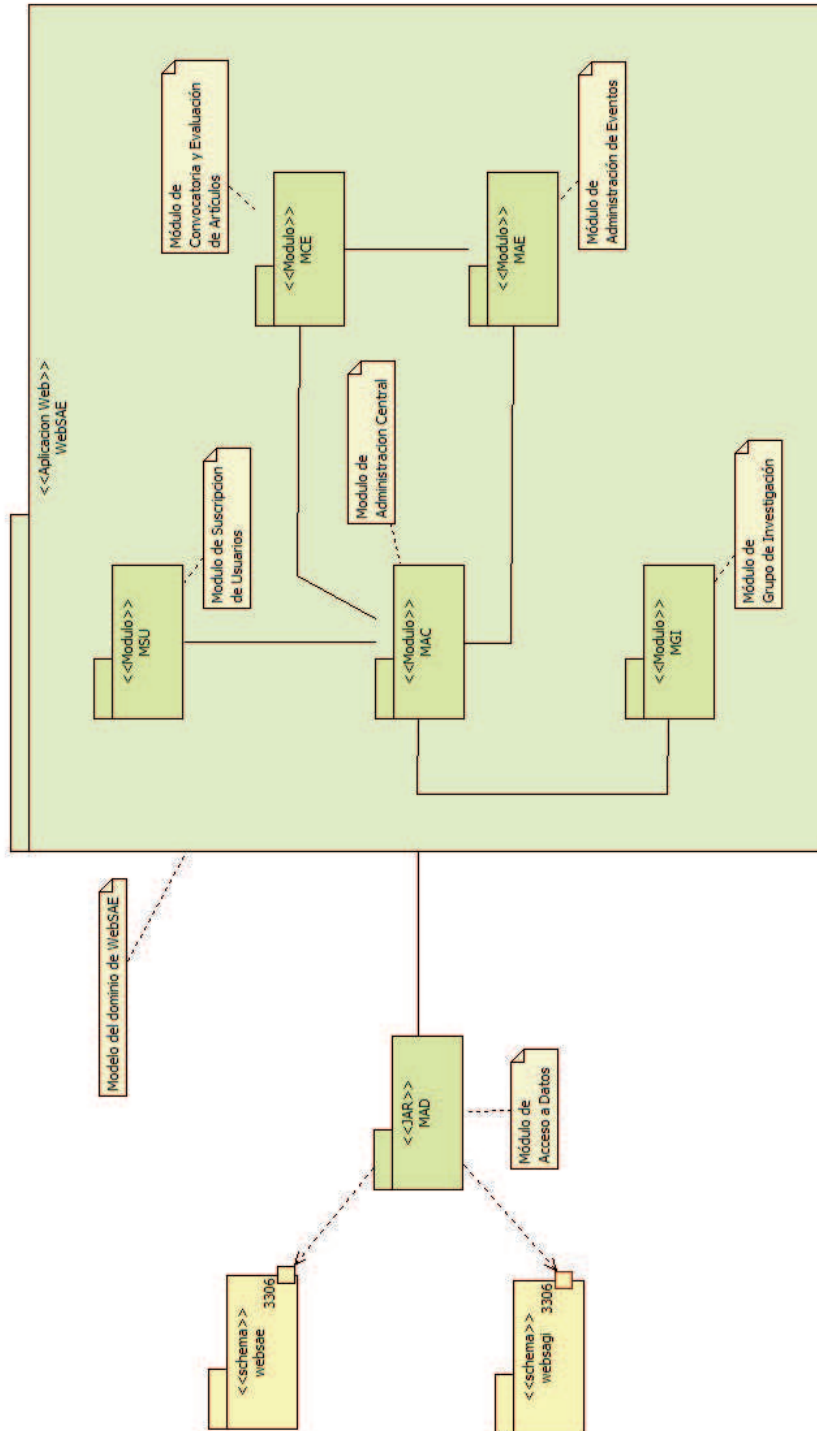


Gráfico 2: Diseño de la Arquitectura del Sistema de Administración de Eventos.

Módulos del Sistema

Arquitectura basada en patrones de diseño

Diagrama de Clases del Sistema

Modelo lógico de la Base de Datos

CAPÍTULO V.

IMPLEMENTACIÓN Y PRUEBAS

Capas del Sistema y comunicación entre capas

Implementación basada en el análisis con MERODE y MDA

Plan de Pruebas

Resultados de las Pruebas y Métricas tomadas

CONCLUSIONES Y RECOMENDACIONES

Comentarios generales

Luego de la elaboración del PSM de cada módulo con su correspondiente PIM, se llevó a cabo la transformación deseada del código, sólo del diseño del dominio.

En este punto es necesario comentar acerca de las herramientas que usamos para la generación deseada del PSM al código, en primera instancia deseamos utilizar una herramienta llamada Magic Draw¹, una herramienta que no sólo nos generó las clases propias del dominio, si no también el correspondiente DDL² de la Base de Datos que nosotros habíamos seleccionado; sin embargo, la curiosidad intelectual nos llevó a utilizar otra herramienta CASE Open Source para utilizar las metodologías MDA y MERODE en conjunto, y la que se seleccionó fue StarUML debido a sus singulares características de las

¹<http://www.magicdraw.com/>

²Data Definition Language

que en su momento deseamos aprovecharnos, pero existió la gran desventaja que esta herramienta solo nos proporcionó la transformación del PSM al código java.

La única ventaja que se pudo sacar de esta herramienta fue la de conocer el hecho que se pueden implementar cartuchos para la transformación del PSM al código, no sólo de la parte del dominio del sistema, si no también de la capa controlador, que en el caso nuestro, fue implementado según la arquitectura J2EE.

Las "transformaciones manuales" que se realizaron en la mayoría del proyecto, nos ayudó a la estandarización del código de la misma, conservando una gran similitud en la parte del código en el lado del servidor y del cliente, conservando el patrón de diseño MVC 2 mencionado como requerimiento no funcional.

REFERENCIAS

- [1] Monique Snoeck, Guido Dedene, Maurice Verhelst, Anne-Marie Depuydt, *Object-Oriented Enterprise Modelling with MERODE*, Leuven University Press, Leuven/Louvain, Belgium, 1999.

- [2] Karina Chong, *Elaboración de un Portal Web que maneje y administre las suscripciones al portal, la organización de eventos y las publicaciones para grupos de investigación, utilizando MERODE (Object Oriented Business Modelling) como metodología de análisis*, Guayaquil, Ecuador, Septiembre 2007.

- [3] Salomón Herrera, *Implementación de los Módulos de Convocatoria de Artículos y Evaluación de Artículos para el Portal Web del Componente 8 del Proyecto VLIR-ESPOL utilizando MERODE como Metodología de Análisis y J2EE como Arquitectura de Diseño*, Guayaquil, Ecuador, Enero 2008.

- [4] María Victoria Di Libero, *Arquitectura Dirigida por Modelos*, Buenos Aires, Argentina.