

Universidad de las Ciencias Informáticas.



Facultad 1

Análisis y Diseño de la Gestión de los eventos científicos de la
FEU en la UCI.

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

Autor: Jonathan González Pieras.

Tutor: Lic. Yoemny González Almaquer.

Co-tutor: Ing. Osay González Fuentes.

Ciudad Habana, junio del 2007

Año 49 de la Revolución.

Agradecimiento

A mis compañeros de la escuela, especialmente a Ygraine Borges, José Enríquez Saura, Ariel Ramírez, Arian Abel Couso, Jorge Antonio Gómez, Heidy Valenzuela, Isel Batista, Dailiany Soler, Linsadro Díaz y Antonio Marrero.

A mis profesores de mi escuela, especialmente a mi tutor Yoemny González, a Hugo Vargas y Mario Lázaro Mesa, Ernesto González, Henry mi profesor de comercio electrónico.

A Nuestro Comandante en Jefe Fidel Castro Ruz que tuvo la idea de crear este proyecto futuro para nosotros, y por el desarrollo de nuestro país.

Dedicatoria

A mi familia por apoyarme en todo, principalmente mi mamá Gladys Miriam Pieras y mi hermana Zabily González.

El dedico este trabajo a la Federación Estudiantil Universitaria a la cual pertenezco.

A la Universidad de las Ciencias Informáticas, mi escuela.

A nuestra Revolución Cubana que me ha dado la posibilidad de cursar esta carrera

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al local de FEU de las Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2007.

Jonathan González Pieras

Autor

Yoemny González Almaquer

Tutor

Osay Gonzalez Fuente

Tutor

OPINIÓN DEL TUTOR.

Título: Análisis y Diseño de la Gestión de los eventos científicos de la FEU en la UCI.

Autor: Jonathan González Pieras.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

Firma

Firma

Fecha

Resumen

Esta labor nos muestra una síntesis de las tendencias actuales sobre la Gestión de los eventos científicos y las descripciones de las herramientas que son utilizadas para dicho desarrollo. El estudio del funcionamiento de estos eventos científicos estudiantiles de la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas donde se define una nueva propuesta para darle solución al problema dado, además se da cumplimiento al objetivo general, el cual es la realización del análisis y diseño de la gestión de los eventos científicos de la FEU en UCI que será incorporado al portal Web de la FEU nacional realizado por el equipo del proyecto Kainos, se efectúa la estimación de costo del proyecto este trabajo donde podemos ver la duración del y el costo del proyecto.

En estos años, el crecimiento de los acontecimientos científicos realizado por nuestro centro han sido amplio para los protagonistas de la FEU, debido al incremento de la matrícula de los participantes y el surgimiento de nuevos eventos desempeñado en el nuestra universidad, sus organizadores requieren de un sistema digital que permita procesar la información y transferirla entre sus miembros, solicitando un nuevo módulo dispone una solución al problema presentado, aportando como ventaja la calidad del control y el manejo la información de los eventos científicos estudiantiles distribuidos en la UCI.

Índice

Introducción.....	1
Capítulo 1 Fundamentación Teórica.....	3
Introducción	3
1.1 Sistema de la Gestión de e los Eventos Científicos. Tendencias Actuales.....	3
1.2 Descripción de Herramientas y Tecnologías utilizada.....	4
1.2.1 Visual Paradigm.....	4
1.2.2 Subversion (Tortoise).....	4
1.2.3 J2EE.....	4
1.2.4 Diseño de la Arquitectura de Alto nivel. Arquitectura basada en Hibernate+Spring+JSF.....	5
1.25 RUP	7
1.3 Propuestas para el desarrollo del módulo.....	7
1.3.1 Apache Tomcat.....	7
1.3.2 IDE Eclipse.....	8
1.3.3 Postgresql.....	8
1.3.4 Java.....	9
Conclusiones	10
Capítulo 2 Característica del Sistema.....	11
Introducción	11
2.1 Objeto de estudio.....	11
2.1.1 Problema y situación problemática.....	11
2.1.2 Objeto de automatización	11
2.1.3 Información que se maneja.....	12
2.1.4 Propuesta del sistema.....	12
2.2 Modelo del Negocio.....	12
2.2.1 Definición de Actores del Negocio.....	12
2.2.2 Definición de Trabajadores del Negocio.....	13
2.2.3 Diagrama de Casos de Uso del Negocio.....	14
2.2.4 Descripción de los procesos del negocio mediante los Diagramas de Actividades. ...	14
2.2.5 Modelo de Objetos del Negocio.....	19
2.3 Especificación de los requisitos del software.....	20
2.3.1 Dependencias y relaciones.....	20

2.3.2 Requisitos Funcionales.....	20
2.3.3 Requisitos no Funcionales.....	20
2.4 Modelo de Sistema.....	21
2.4.1 Definición de los Actores del Sistema.....	21
2.4.2 Definición de los Casos de Usos del Sistema.....	21
2.4.3 Distribución de los Casos de Usos por ciclo.....	23
2.4.4 Diagrama y descripción expandida de los Casos de Uso del Sistema.....	24
Conclusiones	30
Capítulo 3 Análisis y Diseño del Sistema.....	31
Introducción	31
3.1 Modelo Análisis.....	31
3.2 Modelo Diseño.....	31
3.2.1 Diagrama de Clase del Diseño. Descripción de las clases del diseño.....	32
3.2.2 Diagrama de Interacciones.....	32
3.2.3 Descripciones de las clases.....	32
3.2.3.1 Clases Entidades.....	32
3.2.3.2 Clases Dao	34
3.2.3.3 Clase de Servicios	38
3.2.3.4 Clases ServiceLocator	40
3.2.3.5 Clase Bean	40
3.2.3.6 Clases Interfaz.....	47
3.3 Diseño de la Base de Datos.....	50
3.3.1 Diagrama de Entidad Relación de las Bases de Datos.....	51
3.3.2 Descripción de las tablas.....	51
3.4 Definiciones de Diseño.....	52
3.5 Concepción general de la ayuda.....	53
3.6 Tratamiento de errores.....	53
Conclusiones.....	54
Capítulo 4 Estimación de Costo del Proyecto.....	55
Introducción	55
4.1 Cálculo de Puntos de Casos de Uso sin ajustar.....	55
4.2 Cálculo de los Puntos de Casos de Uso ajustados.....	56
4.3 Estimación de esfuerzo a través de los puntos de casos de uso.....	58
4.4 Calcular esfuerzo de todo el proyecto.....	58

4.5 Costo del Proyecto	59
Conclusiones	59
Conclusiones Generales.....	60
Recomendaciones	61
Referencias Bibliográficas	62
Bibliografía.....	63
Anexos.....	64
Figura 2 Diagrama de Casos de Uso del Negocio	64
Figura 3 Diagrama de Actividades del CUN Gestionar Evento Científico	65
Figura 4 Diagrama de Actividades del CUN: Obtener de la Emulación del Evento.	66
Figura 5 Diagrama de Clases del Modelo de Objetos.	67
Figura 6 Diagrama de Casos de Uso del Sistema.....	68
Diagrama de Clases de Análisis.....	69
Figura 7 Diagrama de Clases de Análisis del CUS Gestionar Orientación del Evento.	69
Figura 8 Diagrama de Clases de Análisis del CUS Recibir Orientación.....	69
Figura 9 Diagrama de Clases de Análisis del CUS Gestionar Evento.....	69
Figura 10 Diagrama de Clases de Análisis del CUS Gestionar Comisión.....	70
Figura 11 Diagrama de Clases de Análisis del CUS Gestionar Jurado.....	70
Figura 12 Diagrama de Clases de Análisis del CUS Gestionar Equipo.....	71
Figura 13 Diagrama de Clases de Análisis del CUS Obtener Resultados del Evento	71
Figura 14 Diagrama de Clases de Análisis del CUS Mostrar Reporte de Resultados del Evento.....	71
Figura 15 Diagrama de Clases de Análisis del CUS Obtener Emulación del Evento.....	72
Figura 16 Diagrama de Clases de Análisis del CUS Mostrar Reporte de Emulación del Evento.....	72
Diagrama de Interacciones.....	73
Diagramas de Interacción del Gestionar Orientación.	73
Figura 17 Diagrama de Secuencia de Insertar Orientación.....	73
Figura 18 Diagrama de Secuencia de Eliminar Orientación.	73
Figura 19 Diagrama de Secuencia de Actualizar Orientación.	74
Figura 20 Diagrama de Interacción del Obtener Orientación.	74
Diagramas de Interacción del Gestionar un Evento.	74
Figura 21 Diagrama de Secuencia de Insertar Evento.	74
Figura 22 Diagrama de Secuencia de Eliminar Evento.	75

Figura 23 Diagrama de Secuencia de Actualizar Evento.	75
Diagramas de Interacción del Gestionar un Comisión.	76
Figura 24 Diagrama de Secuencia de Insertar Comisión.	76
Figura 25 Diagrama de Secuencia de Eliminar Comisión.	76
Figura 26 Diagrama de Secuencia de Actualizar Comisión.....	76
Diagramas de Interacción del Gestionar Jurado.	77
Figura 27 Diagrama de Secuencia de Insertar	77
Figura 28 Diagrama de Secuencia de Eliminar	77
Figura 29 Diagrama de Secuencia de Actualizar.....	78
Diagrama de Interacción del Gestionar Equipo.	78
Figura 30 Diagrama de Secuencia de Insertar Equipo.	78
Figura 31 Diagrama de Secuencia de Eliminar Equipo.	79
Figura 32 Diagrama de Secuencia de Actualizar Equipo.	79
Figura 33 Diagrama de Interacción del Obtener Resultado del Evento Científico.....	80
Figura 34 Diagrama de Interacción del Mostrar Reporte de Resultado del Evento Científico.	80
Figura 35 Diagrama de Interacción del Obtener Emulación del Evento Científico.	81
Figura 36 Diagrama de Interacción del Mostrar Reporte de Emulación del Evento Científico.	81
Figura 37 Diagrama de Clases del Diseño.	82
Figura 38 Diagrama de Entidad Relación.....	83
Glosario de Términos.	84

Índice de tabla

Tabla 1 Actores del Negocio.....	13
Tabla 2 Trabajadores del Negocio.....	13
Tabla 3 Especificación del caso de uso del negocio Gestionar Evento Científico.....	14
Tabla 4 Especificación del caso de uso del negocio Gestionar la Emulación de los Evento Científico.....	17
Tabla 5 Actores del Sistema.....	21
Tabla 6 Descripción del CUS1.....	21
Tabla 7 Descripción del CUS2.....	22
Tabla 8 Descripción del CUS3.....	22
Tabla 9 Descripción del CUS4.....	22
Tabla 10 Descripción del CUS5.....	22
Tabla 11 Descripción del CUS6.....	22
Tabla 12 Descripción del CUS7.....	23
Tabla 13 Descripción del CUS8.....	23
Tabla 14 Descripción del CUS9.....	23
Tabla 15 Descripción del CUS10.....	23
Tabla 16 Distribución de los Casos de Usos por ciclo.....	23
Tabla 17 Descripción de Casos de Uso del Sistema Gestionar Orientación del evento.	24
Tabla 18 Descripción de Casos de Uso del Sistema Revisar las orientaciones del evento.	24
Tabla 19 Descripción de Casos de Uso del Sistema Gestionar un evento científico.	25
Tabla 20 Descripción de Casos de Uso del Sistema Gestionar una comisión.	26
Tabla 21 Descripción de Casos de Uso del Sistema Gestionar un equipo.	27
Tabla 22 Descripción de Casos de Uso del Sistema Gestionar Jurado.	27
Tabla 23 Descripción de Casos de Uso del Sistema Obtener resultados del evento.....	28
Tabla 24 Descripción de Casos de Uso del Sistema Mostrar Reporte de Resultados.....	29
Tabla 25 Descripción de Casos de Uso del Sistema Obtener Emulación del Evento.....	29
Tabla 26 Descripción de Casos de Uso del Sistema Mostrar Reporte de Emulación.	30
Tabla 27 Descripción de la clase entidad Orientación.....	32
Tabla 28 Descripción de la clase entidad Evento.....	33
Tabla 29 Descripción de la clase entidad Comisión.....	33

Tabla 30 Descripción de la clase entidad Jurado.	33
Tabla 31 Descripción de la clase entidad Equipo.	33
Tabla 32 Descripción de la clase entidad Integrante.	34
Tabla 33 Descripción de la clase entidad Resultado.	34
Tabla 34 Descripción de la clase OrientacionDao.	34
Tabla 35 Descripción de la clase OrientaciónDaoImpl.	34
Tabla 36 Descripción de la clase EventoDao.	35
Tabla 37 Descripción de la clase EventoDaoImpl.	35
Tabla 38 Descripción de la clase ComisionDao.	35
Tabla 39 Descripción de la clase ComisionDaoImpl.	36
Tabla 40 Descripción de la clase JuradoDao.	36
Tabla 41 Descripción de la clase JuradoDaoImpl.	36
Tabla 42 Descripción de la clase EquipoDao.	36
Tabla 43 Descripción de la clase EquipoDaoImpl.	37
Tabla 44 Descripción de la clase IntegranteDao.	37
Tabla 45 Descripción de la clase IntegranteDaoImpl.	37
Tabla 46 Descripción de la clase ResultadoDao.	37
Tabla 47 Descripción de la clase ResultadoDaoImpl.	38
Tabla 48 Descripción de la clase ServiciosEventosCientificos.	39
Tabla 49 Descripción de la clase ServiciosEventosCientificosImpl.	40
Tabla 50 Descripción de la clase ServiceLocator.	40
Tabla 51 Descripción de la clase ServiceLocatorImpl.	40
Tabla 52 Descripción de la clase OrientacionBean.	41
Tabla 53 Descripción de la clase ListOrientacionesBean.	41
Tabla 54 Descripción de la clase EventoBean.	41
Tabla 55 Descripción de la clase ListEventosBean.	42
Tabla 56 Descripción de la clase ComisionBean.	42
Tabla 57 Descripción de la clase ListComisionesBean.	42
Tabla 58 Descripción de la clase JuradoBean.	43
Tabla 59 Descripción de la clase ListJuradosBean.	43
Tabla 60 Descripción de la clase EquipoBean.	44
Tabla 61 Descripción de la clase ListEquiposBean.	44
Tabla 62 Descripción de la clase IntegranteBean.	45
Tabla 63 Descripción de la clase ListResultadoBean.	45

Tabla 64 Descripción de la clase ListReporteResultadosBean. 46

Tabla 65 Descripción de la clase ListEmulacionBean. 46

Tabla 66 Descripción de la clase ListReporteEmulacionBean. 47

Tabla 67 Descripción de la clase insertarOrientacion..... 47

Tabla 68 Descripción de la clase listaOrientacion. 47

Tabla 69 Descripción de la clase actualizarOrientacion. 47

Tabla 70 Descripción de la clase obtenerOrientacion. 48

Tabla 71 Descripción de la clase insertarEvento..... 48

Tabla 72 Descripción de la clase listaEvento. 48

Tabla 73 Descripción de la clase actualizarEvento. 48

Tabla 74 Descripción de la clase insertarComision. 48

Tabla 75 Descripción de la clase listaJurado..... 49

Tabla 76 Descripción de la clase actualizarJurado..... 49

Tabla 77 Descripción de la clase insertarEquipo..... 49

Tabla 78 Descripción de la clase inseratrIntegrante..... 49

Tabla 79 Descripción de la clase listaEquipos..... 49

Tabla 80 Descripción de la clase actualizarEquipo. 49

Tabla 81 Descripción de la clase actualizarIntegrante. 50

Tabla 82 Descripción de la clase obtenerResultado..... 50

Tabla 83 Descripción de la clase reporteResultado. 50

Tabla 84 Descripción de la clase obtenerEmulacion. 50

Tabla 85 Descripción de la clase reporteEmulacion..... 50

Tabla 86 Descripción del la Tabla Orientación. 51

Tabla 88 Descripción del la Tabla Evento. 51

Tabla 89 Descripción del la Tabla Comisión..... 51

Tabla 90 Descripción del la Tabla Jurado..... 51

Tabla 91 Descripción del la Tabla Equipo. 52

Tabla 92 Descripción del la Tabla Integrante. 52

Tabla 93 Descripción del la Tabla Resultado 52

Tabla 94 Factor de peso de los actores sin ajustar 55

Tabla 95 Factor de peso de los casos de uso sin ajustar..... 55

Tabla 96 Factor de complejidad técnica. 57

Tabla 97 Factor de ambiente..... 57

Tabla 98 Porcentajes. 59

Introducción.

La batalla de ideas, en materia de educación, posibilitó el rescate creador de muchos de los principios de la educación revolucionaria no alcanzados totalmente, ya fuera por problemas materiales, de la organización escolar, falta de maestros o falta de disciplina; rescató atención diferenciada al alumno; el conocimiento a plenitud de realidad social y familiar; los grupos docentes con el número de estudiantes que es posible atender y educar, acorde con el ideal de joven que queremos formar; una preparación mayor con la aparición de ideas novedosas.

Dentro de los programas de la Revolución nuestra Universidad surgió con el calor de la batalla de ideas, podemos decir que es hija de esta batalla, la cual ha revolucionado a todos los niveles de la educación y ha comenzado también la transformación de otros niveles educacionales; la impulsó la Universalización y la municipalización de la educación superior y hace hoy reanalizar el concepto tradicional de la universidad; a través de este proyecto que es la Universidad de la Ciencias Informáticas.

Decía el Comandante: *"Debe ser una Universidad más nueva, más universal, más creativa."*

En nuestra Universidad después de haber realizados distintos eventos científicos hemos llegado a un cumplimiento donde se nos hace necesario buscar vías que permitan que la gestión de información tenga una mayor organización y accesibilidad. La gestión de la informatización no es más que la perfección de los servicios solicitados por los usuarios haciendo uso de las Tecnologías de la Información y la Comunicación.

El proyecto Kainos organizado en la UCI, tiene el compromiso de digitalizar el funcionamiento de la Federación Estudiantil Universitaria en nuestro país, durante su marcha, la dirección de la organización estudiantil de Universidad de las Ciencias Informática estableció determinados modulo que son solicitados por el grupo de miembros de secretariado, un ejemplo es la gestión de los eventos científicos en la Jornada Científicas Estudiantil, organizada por los miembros de secretariado de Producción-Investigación. La gestión de los eventos científicos estudiantiles dirigido por la FEU, consiste en la mejor comunicación y la eficiencia los servicios solicitados en la Jornada Científica entre los miembros de esta organización en la Universidad.

Estructuración del contenido

Capítulo 1: Fundamentación del tema En este capítulo se hace un estudio del estado del arte de las herramientas para el Modelado de Negocio y Gestión de Requisitos. Se explican las metodologías, herramientas y tecnologías a utilizar para el desarrollo del análisis y diseño.

Capítulo 2: Características del Sistema En este capítulo se realiza el Modelado de Negocio y Levantamiento de Requisitos realizado una la propuesta al problema.

Capítulo 3: Análisis y Diseño del sistema. En este capítulo se realiza el Análisis y Diseño dando cumplimiento a los objetivos planteados.

Capitulo 4 Estimación de Costo de Proyecto. En este capitulo se realiza la Estimación de Costo del Proyecto calculando la durabilidad del proyecto y el costo.

Situación Problémica:

La dirección de la FEU de la UCI ha tenido dificultades en la organización de los eventos científicos realizados durante el curso escolar, entre ella esta la falta de eficiencia en la comunicación entre sus miembros y la información de los resultados para la emulación entre las facultades a nivel de universidad no ha sido precisa en estos últimos años debido al número de estudiantes que participa en la Jornadas Científicas.

El Problema:

¿Cómo la FEU de la Universidad de las Ciencias Informáticas gestiona los eventos científicos estudiantiles?

Objeto de estudio:

Gestión de los eventos científicos de la FEU en la Universidad de las Ciencias Informáticas.

Objetivo general:

Desarrollar el analisis y diseño Web que le permita gestionar a la FEU de la Universidad de las Ciencias Informáticas los eventos científicos estudiantiles que se realizan en el curso.

Objetivos específicos:

Conocer el funcionamiento y la estructura de los eventos científicos y de la emulación de estos dirigidos por FEU de la UCI.

Análisis y Diseño de un sistema para la gestión de los eventos científicos de la FEU en la UCI.

Hipótesis:

El análisis y diseño de una aplicación Web que se alcanzará acceder de manera más eficiente y con mejor calidad en los servicios solicitados por los miembros a la FEU de la Universidad de las Ciencias Informáticas que organizan los eventos científicos estudiantiles durante el curso.

Campo de Acción:

La Federación Estudiantil Universitaria de la Universidad de las Ciencias Informaticas.

Capítulo 1 Fundamentación Teórica.

Introducción

En este capítulo haremos un estudio sobre el estado actual de los sistemas de la Gestión de los Eventos Científicos, también se incluye una breve descripción de las herramientas seleccionadas para el desarrollo del misma.

1.1 Sistema de la Gestión de e los Eventos Científicos. Tendencias Actuales.

De la Federación Estudiantil Universitaria de UCI surgió un equipo de estudiantes que sus principales metas son digitalizar los servicios y funcionamiento de la FEU nacional, este grupo se denomina proyecto Kainos; donde se han realizado grandes aportes para el portal Web de nuestra organización, entre ellos el módulo de funcionamiento y estadísticas, pero durante este desarrollo productivo los integrantes del proyecto ha decido seguir ampliado el portal, entre ellos gestionar los eventos científicos estudiantiles de la UCI, el cual no tenemos en la universidad un sistema que le permita la solución.

Durante la navegación de nuestra intranet nacional hemos encontrado en el Centro Nacional de Información de Ciencias Médicas (INFOMED) en sistema un servicio de atención a eventos: El Centro Virtual de Convenciones de Salud (CENCOMED), que con el uso de las nuevas tecnologías de la información facilita la divulgación, gestión, promoción y desarrollo de eventos y actividades científico- académicas virtuales y presenciales.

Este gestor de eventos realizado en el lenguaje PHP para las aplicaciones Web permite:

Insertar un evento que tiene 20 campos de los cuales 7 obligatorios:

- *Título, Lugar, Fecha de inicio, Fecha en que finaliza, Descripción del evento, Especialidad y Nombre de contacto.*
- Un sistema de búsqueda de eventos de acuerdo con la fecha de inicio o la especialidad.
- Divulgación por la Lista de Anuncios es enviar un correo a una dirección de correo que será el encargado de transmitir el evento que se realiza a una lista de usuarios que pertenecen al portal.
- Un sistema de ayuda que le permite diferentes modalidades de presentación de trabajos en un evento científico; conferencia, taller, etc.

Ubicación: <http://www.infomed.sld.cu/servicios/eventos/>

A pesar de las condiciones que presenta nuestro país, el Centro Nacional de Información de Ciencias Médicas ha desarrollado un sistema de gestión de los eventos científicos médicos; dado la idea fue diseñado este trabajo, que aporta a nuestra organización en la Universidad de tener su propio sistema de gestión de los eventos científicos estudiantiles, además este sistema puede ser utilizada para otras ciencias que son ofertadas en el nivel superior, favoreciendo una vez más la construcción de portal Web de la FEU Nacional.

1.2 Descripción de Herramientas y Tecnologías utilizada.

Las herramientas utilizadas para la realización de este trabajo de diploma son las siguientes:

1.2.1 Visual Paradigm.

Al escoger las herramientas para analizar y diseñar el software en cuestión nos basamos en las tendencias más actuales a nivel nacional e internacional, seleccionamos para el análisis y diseño la herramienta de modelado Visual Paradigm 5.3 Enterprise Edition pues aunque es Software privativo gratuito para modelado en UML, la universidad adquirió la licencia. Su uso se ha extendido entre los analistas de todo el mundo dados sus facilidades de integración con distintos entornos de desarrollo, incluye UML 2.0, brindando la posibilidad de modelar el sistema orientado a procesos en el futuro.

1.2.2 Subversion (Tortoise).

Para garantizar un correcto control de versiones y gestión de cambio se decidió utilizar la Subversion con el cliente Tortoise, ya que es una de las herramientas que más se ha venido utilizando en nuestra universidad ocupando el 98% de los proyectos que la utilizan actualmente, frente a un 2 % de los proyectos que usan el Visual Source Safe.

1.2.3 J2EE.

Java 2 Enterprise Edition, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Java con arquitectura de varios niveles distribuidos, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE es también considerada informalmente como un estándar. Es por esto que se decidió utilizar J2EE. La utilización de Servidores de Aplicaciones y la ejecución de aplicaciones basadas en tecnología J2EE impactan en mayor o menor medida en distintas disciplinas dentro de una organización. Algunas de las más relevantes:

- Modelado de Datos.
- Análisis de Requerimientos.
- Arquitectura y Diseño.

- Codificación.
- Testing.
- Entorno y Operación.
- Administración de Proyectos.

1.2.4 Diseño de la Arquitectura de Alto nivel. Arquitectura basada en Hibernate+Spring+JSF.

La arquitectura de alto nivel implica subdividir la aplicación en componentes funcionales y particionar estos componentes en capas. El diseño de la arquitectura de alto nivel es neutral a las tecnologías utilizadas.

Arquitectura Multi-capa.

Una arquitectura multicapa particiona todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, acceso a datos, y bases de datos. Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios. La capa del cliente es donde se consumen y presentan los modelos de datos. Para una aplicación Web, la capa cliente normalmente es un navegador Web. Los clientes pequeños basados-en-navegador no contienen lógica de presentación; se trata en la capa de presentación. La capa de presentación expone los servicios de la capa de lógica-de-negocio a los usuarios. Sabe cómo procesar una petición de cliente, cómo interactuar con la capa de lógica-de-negocio, y cómo seleccionar la siguiente vista a mostrar. La capa de la lógica-de-negocio contiene los objetos y servicios de negocio de la aplicación. Recibe peticiones de la capa de presentación, procesa la lógica de negocio basada en las peticiones, y media en los accesos a los recursos de la capa de datos. Los componentes de la capa de lógica-de-negocio se benefician de la mayoría de los servicios a nivel de sistema como el control de seguridad, de transacciones y de recursos. La capa de acceso a datos es el puente entre la capa de lógica-de-negocio y la capa de datos. Encapsula la lógica para interactuar con la capa de datos. Los datos de la aplicación persisten en la capa de datos. Contiene bases de datos relacionales, bases de datos orientadas a objetos, y sistemas antiguos.

- **La Capa de Presentación y JavaServer Faces**

JSF es un marco de trabajo de componentes de interfaces de usuario del lado del servidor para aplicaciones Web basadas en Java. JSF contiene un API para representar componentes UI y manejar sus estados, manejar sus eventos, la validación del lado del servidor, y la conversión de datos, definir la navegación entre páginas, soportar internacionalización y

accesibilidad; y proporcionar extensibilidad para todas estas características. También contiene dos librerías de etiquetas JSP (JavaServer Pages) personalizadas para expresar componentes UI dentro de una página JSP y para conectar componentes a objetos del lado del servidor.

La capa de presentación recoge la entrada del usuario, presenta los datos, controla la navegación por las páginas y delega la entrada del usuario a la capa de la lógica-de-negocio. La capa de presentación también puede validar la entrada del usuario y mantener el estado de sesión de la aplicación.

- **La Capa de Lógica-de-Negocio y el Marco de Trabajo Spring.**

Los objetos y servicios de negocio existen en la capa de lógica-de-negocio. Un objeto de negocio no sólo contiene datos, también la lógica asociada con ese objeto específico.

Los servicios de negocio interactúan con objetos y proporcionan una lógica de negocio de más alto nivel. Se debería definir una capa de interfase de negocio formal, que contenga las interfaces de servicio que el cliente utilizará directamente. Con la ayuda del marco de trabajo Spring, implementará la capa de lógica-de-negocio de la aplicación JCatalog.

- 1) Manejo de Beans con contexto de aplicación.

Spring puede organizar de forma efectiva nuestros objetos de la capa central y manejar las conexiones por nosotros. Spring puede eliminar la proliferación de solitarios y facilita unas buenas prácticas de programación orientada a objetos, por ejemplo utilizando interfaces.

- 2) Integración con Hibernate.

Spring no nos fuerza a utilizar su potente característica de abstracción JDBC. Se integra bien con marcos de trabajo de mapeo O/R, especialmente con Hibernate. Spring ofrece un manejo seguro y eficiente de sesiones Hibernate, maneja la configuración de la SessionFactory de Hibernate y las fuentes de datos JDBC en el contexto de la aplicación, y hace que la aplicación sea más fácil de testear.

- **La Capa de Integración e Hibernate.**

Hibernate es un marco de trabajo de mapeo Open Source que evita la necesidad de utilizar el API JDBC. Hibernate soporta la mayoría de los sistemas de bases de datos SQL. El *Hibernate Query Language*, diseñado como una extensión mínima, orientada a objetos, de SQL, proporciona un puente elegante entre los mundos objeto y relacional. Hibernate ofrece facilidades para recuperación y actualización de datos, control de transacciones, repositorios de conexiones a bases de datos, consultas programáticas y declarativas, y un control de relaciones de entidades declarativas.

1.25 RUP

Utilizamos la metodología RUP por las tendencias actuales que existen, que también hablan mucho de la utilización de XP o Extreme Programming que proponen un ciclo de desarrollo de software de manera ágil y novedosa, contando siempre con un cliente interno que tendrá un contacto directo con el equipo de desarrollo, razón por la cual no es apropiado para nuestro sistema precisamente por las características de nuestro cliente, RUP es la que mejor se adapta a nuestro caso de estudio y a las condiciones del equipo de desarrollo, teniendo en cuenta y analizando que el organigrama de equipo del mismo es descentralizado democrático y la comunicación es abierta. RUP es uno de los procesos más generales y abarcadores de los existentes actualmente, ya que en realidad está pensado englobar todo el desarrollo del mismo, además de ser una metodología probada y segura. RUP se basa en casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal.

1.3 Propuestas para el desarrollo del módulo.

Además de utilizar las herramientas y tecnologías antes mencionadas, en esta investigación proponemos las siguientes para el trabajo en las fases restantes:

1.3.1 *Apache Tomcat.*

Como servidor de aplicaciones la propuesta es Apache Tomcat, debido a que constituye uno de los más completos contenedores de Servlet gratuito, que programadores de Servlet de Java o Páginas de JavaServer (JSP) utilizan con frecuencia para probar su código. Tomcat persigue además la total compatibilidad con las versiones de Servlet y las especificaciones de API JSP que admite. Sin embargo, se trata de algo más que un servidor de pruebas, dado que muchas empresas lo emplean en la actualidad en entornos de producción debido a su contrastada estabilidad.

¿Por qué necesitamos usar Tomcat para ejecutar JAVA en Apache?

El funcionamiento principal de Apache desde su creación fue la de aceptar y responder requests de páginas de internet, y como fue mencionado en servidores de páginas y JAVA Application Server. Estos requests correspondían a documentos estáticos (puro HTML), es por esto que cuando se necesita ejecutar algún tipo de contenido dinámico (programas como JAVA) es necesario coordinar los esfuerzos de Apache con otro ambiente. En el caso de JAVA es precisamente Tomcat. Quién ofrece facilidades para ejecutar los dos componentes más utilizados en ambientes JAVAS: JSP, (Java server Pages) y servlets.

1.3.2 IDE Eclipse.

Como Entorno Integrado de Desarrollo (IDE) proponemos la utilización del Eclipse, el cual no es más que un software creado originalmente por IBM y actualmente, la Fundación Eclipse, una entidad sin ánimo de lucro, fundada por IBM se encarga de llevar a cabo el desarrollo de este proyecto. La definición que da la Fundación Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular". Eclipse es un entorno integrado de desarrollo (IDE, Integrated Development Environment) multiplataforma englobado en el movimiento de software libre para crear aplicaciones clientes de cualquier tipo.

Ventajas:

- Es open-source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta el desarrollo de aplicaciones basadas en GUI y non-GUI.
- Soporta herramientas que manipulan diferentes tipos de archivos como por ejemplo Java, C, C++, EJB, HTML, GIF, etc.
- Corre en una gran cantidad de sistemas operativos incluyendo Windows y Linux.
- Provee a los desarrolladores, herramientas (ej.- PDE) que facilitan la creación de plugins.
- Mediante JDT facilita la creación de aplicaciones programadas en Java.

Desventajas:

- Si bien Eclipse es multiplataforma, los plugins no tienen por qué serlo.
- Existen plugins que sólo corren en una plataforma, o que aún no han sido desarrollados para más de una.
- Al ser una herramienta open-source, se desarrollan plugins que no tienen todas las funcionalidades que tienen en otras herramientas comerciales, como ser IBM Websphere.

1.3.3 Postgresql.

Proponemos el uso de PostgreSQL, por ser un motor de base de datos, servidor de base de datos relacional libre, liberado bajo una licencia de software libre.

Algunas de sus características fundamentales son:

- Alta concurrencia.
- Amplia variedad de tipos nativos.

- Los usuarios pueden crear sus propios tipos de datos.
- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).
- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Funciones.

Inconvenientes:

- Consume recursos y carga el sistema.
- Límite del tamaño de cada fila de las tablas a 8k.
- Es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP.

1.3.4 Java.

Proponemos que la implementación sea en el lenguaje JAVA dadas sus características y potencialidades. Java está totalmente basado en clases y objetos. Todo en Java (aparte de algunos tipos primitivos) es un objeto. Contrariamente a lenguajes híbridos como C++ o el popular Visual Basic, en Java no se permite programar fuera de un objeto o clase. No hay módulos ni funciones globales. Una característica todavía más distintiva de Java es su capacidad multiplataforma. Lenguajes como C o COBOL se han implementado en múltiples plataformas, pero siempre han necesitado recopilaciones o adaptaciones al pasar de una a otra. En cambio, Java desde el principio ha sido pensado para adaptarse a varios entornos. Esto lo consigue no sólo a nivel de código fuente, sino también a nivel de código compilado. El programa que escribimos se puede compilar en Windows o en Linux, y funciona. Pero, además, hasta el programa compilado puede ejecutarse sin más preparación, en distintas máquinas. Eso lo consigue porque Java se compila y ejecuta, no en un procesador o entorno en particular, sino en lo que se llama una "virtual machine", una máquina virtual. Nuestro programa Java podrá ejecutarse en cualquier sistema operativo que tenga una máquina virtual Java compatible. Java no es solamente un lenguaje, es una tecnología. Al estar basado en clases y objetos, viene acompañado de un conjunto de éstos, que nos sirven como base para la programación de aplicaciones, de texto, gráficas o que se ejecuten en una página Web como "applets". Java maneja prácticamente todas las bases de datos relacionales, desde Microsoft SQL Server, Oracle, Sybase, Informix, como Access, MySQL y PostgreSQL.

Conclusiones

En este capítulo se realizó un profundo estudio sobre las tendencias actuales que rodean el problema; y a través de estos conceptos y definiciones planteadas se fundamenta los principios de este trabajo. Además se realizó un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, fundamentándose las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar aunque en esta etapa solo se habla de ideas, es correcto que estén bien fundamentadas, porque estas constituyen la base para el posterior desarrollo de este trabajo.

Capítulo 2 Característica del Sistema.

Introducción

En este capítulo se realiza una propuesta sobre el problema del proceso de Gestión de los eventos científicos del FEU en la UCI, además del modelo del negocio donde asegurarse de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización y del levantamiento de requisitos es una actividad que se llevan a cabo en la Gestión de requisitos del sistema.

2.1 Objeto de estudio.

2.1.1 Problema y situación problemática

La FEU esta asignada para la organización de los eventos científicos estudiantiles realizados en la Universidad y la gestión de la emulación entre las facultades y los equipos que participan en estas actividades durante el curso.

La presidencia de la Universidad solicita la creación un evento científico que es organizado por miembro del secretariado de producción e investigación, esta actividad esta contenida por comisiones donde cada equipo participa con su trabajo de forma competitiva hasta obtener los resultados de premiación que primeramente son efectuado a nivel de facultad, los cuales participarán en la realización los eventos científicos a nivel de Universidad, todo es vinculado en la emulación donde serán seleccionados los grupos y las facultades destacadas.

Los eventos científicos estudiantiles son para el perfeccionamiento de los servicios y funcionamiento de la Universidad, pero no existe un sistema automatizado que gestione los eventos científicos de la FEU en la UCI.

2.1.2 Objeto de automatización

Como solución al problema planteado en el epígrafe anterior se desea realizar el análisis y diseño de un sistema que permita automatizar los procesos de estos eventos científicos de la FEU en la UCI, donde se podrán gestionar los eventos científicos y la emulación de estos realizados durante el curso.

2.1.3 Información que se maneja.

La información que se maneja serán todos los datos del evento científico, como son datos del evento, comisiones, jurado, los equipos que participan y los resultados en cada comisión los cuales son:

- Premio Relevante.
- Premio Destacado.
- Premio Mención.

Además la información también vincula la emulación en los eventos de científicos entre los grupos y facultades que participan, de acuerdo a la cantidad de premios otorgados.

2.1.4 Propuesta del sistema.

Se propone un sistema que podrá gestionar de eventos científicos de la FEU en la UCI, que estará dividido en gestionar los eventos científicos estudiantiles y la gestión de la emulación de estos.

La gestión de los eventos científicos estudiantiles será el manejo de los datos y premios otorgando en el evento. La gestión de la emulación estaría vinculada de acuerdo con los resultados obtenidos una vez termina la actividad.

2.2 Modelo del Negocio.

Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio,

Los objetivos del modelamiento del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

2.2.1 Definición de Actores del Negocio.

Un candidato a actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio. Ejemplo de ello son Clientes o potenciales clientes, Socios, Proveedores, Autoridades, Propietarios, Sistemas de información externos al negocio, Otras partes de la organización, si la organización es grande.

Para cada actor del negocio que se identifica se debe escribir una breve descripción que incluya sus responsabilidades y por qué interactúa con el negocio. Veamos en la siguiente tabla los actores de la Gestión de los Eventos Científicos Estudiantiles:

Actores del Negocio	Justificación
Presidencia de la FEU de la UCI.	Este actor es el que inicia los casos de uso Gestionar Eventos Científicos y Gestionar Emulación, ya que es el que obtiene un beneficio como resultado de todos los procesos del negocio

Tabla 1 Actores del Negocio

2.2.2 Definición de Trabajadores del Negocio.

Los trabajadores del negocio son aquellos que trabajan en el negocio. Muestran las responsabilidades que puede tener una persona, por lo que representa una abstracción de una persona que actúa dentro del negocio. Veamos en la siguiente tabla los trabajadores de la Gestión de los Eventos Científicos Estudiantiles:

Trabajadores del negocio	Justificación
Miembro del Secretariado de Producción e Investigación de la Universidad	Miembro del secretariado de producción e investigación del centro es el encargado de organizar y controlar los eventos realizado por el centro y gestionar estadística de la emulación de las facultades.
Miembro del Secretariado de Producción e Investigación de la Facultad	Miembro del secretariado de producción e investigación de la facultad es el encargado de obtener el control del evento realizado en la facultad y gestionar estadística la emulación de los equipos.
Responsable de Emulación de la Facultad.	Responsable de Emulación de la facultad es el encargado de gestionar la emulación de la facultad
Responsable de Emulación de la Universidad	Responsable de Emulación de la Universidad es el encargado de gestionar la emulación de la Universidad.

Tabla 2 Trabajadores del Negocio

2.2.3 Diagrama de Casos de Uso del Negocio.

Un diagrama de casos de uso del negocio representa gráficamente a los casos de uso del negocio y su interacción con los actores del negocio.

Gestionar de los Eventos Científicos de la FEU en la UCI, esta compuesto básicamente por dos casos de usos del negocio: Gestionar Eventos Científicos y Obtener la Emulación de los Eventos Científicos y como actor que los inicia la Presidencia de la FEU de la UCI. Ver el diagrama de casos de uso del negocio en la figura 2 de los Anexos.

2.2.4 Descripción de los procesos del negocio mediante los Diagramas de Actividades.

Los casos de uso del negocio consisten de secuencias de actividades que, en conjunto, producen algo para el actor del negocio. La estructura del flujo se describe gráficamente con la ayuda de un Diagrama de actividad, el cual describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. En las figuras 3 y 4 de los Anexos podemos ver los Diagramas de Actividades de los casos de uso del negocio Gestionar Eventos Científicos y Obtener la Emulación de los Eventos Científicos.

La descripción textual de un caso de uso del negocio se formaliza en un documento generalmente llamado **Especificación del caso de uso del negocio**.

Tabla 3 Especificación del caso de uso del negocio Gestionar Evento Científico.

Nombre del Caso de Uso del Negocio:	Gestionar Eventos Científicos.
Actor(es):	Presidencia de la FEU nivel UCI
Trabajador(es):	Miembro del secretariado de Producción-Investigación de la UCI. Miembro del secretariado de Producción-Investigación de la facultad.
Breve Descripción:	Se inicia cuando desde la Presidencia de la FEU a nivel de Universidad se desea gestionar los eventos científicos en la UCI. El miembro del secretariado de Producción-Investigación a nivel UCI recibe las orientaciones del evento y les indica a los miembros del secretariado de Producción-Investigación a nivel de facultad acerca de la gestión del evento. Luego dicho miembro organiza los grupos de trabajo en la facultad designando un responsable por cada tarea. Luego al realizarse el evento a nivel de facultad los resultados son informados por el miembro del secretariado de Producción-Investigación de la facultad al Miembro del secretariado de Producción-Investigación de la UCI, el cual con dichos

	<p>resultados procede a realizar el evento a nivel de Universidad, para lo cual distribuye las tareas del mismo entre sus homólogos en las facultades. Luego de realizar el evento a nivel UCI, realiza un resumen del mismo que se le es presentado a la Presidencia de la FEU en la UCI.</p>	
Precondición:	<p>Debe haber un cronograma de los eventos científicos.</p>	
Post-condición:	<p>Debe obtener un registro de los resultados de eventos científicos.</p>	
Flujo Normal de Eventos:	Acción del Actor.	Respuesta del Negocio.
	<p>1 La Presidencia de la FEU nivel UCI solicita la gestión de un evento científico.</p>	<p>1.1 El Miembro del secretariado de Producción-Investigación de la UCI recibe la solicitud.</p> <p>1.2 El Miembro del secretariado de Producción-Investigación de la UCI elabora las orientaciones del evento científico.</p> <p>1.3 El Miembro del secretariado de Producción-Investigación de la UCI da las orientaciones del evento a al Miembro del secretariado de Producción-Investigación de la facultad.</p> <p>1.4 El Miembro del secretariado de Producción-Investigación de la facultad recibe la orientación.</p> <p>1.5 El Miembro del secretariado de Producción-Investigación de la facultad realiza una distribución de las tareas a cada responsable de grupo de trabajo.</p> <p>1.6 El Miembro del secretariado de Producción-Investigación de la</p>

	<p>facultad organiza el evento a nivel de facultad.</p> <p>1.7 Al concluir el evento el Miembro del secretariado de Producción-Investigación de la facultad realiza un resumen con los resultados de mismo.</p> <p>1.8 El Miembro del secretariado de Producción-Investigación de la facultad le entrega resumen del evento en la facultad al miembro del secretariado de Producción-Investigación de la UCI.</p> <p>1.9 El Miembro del secretariado de Producción-Investigación de la UCI recibe el resumen de la facultad.</p> <p>1.10 El Miembro del secretariado de Producción-Investigación de la UCI realiza una distribución de las tareas a cada responsable de grupo de trabajo.</p> <p>1.11 El Miembro del secretariado de Producción-Investigación de la UCI organiza el evento a nivel de facultad.</p> <p>1.12 Al concluir el evento el Miembro del secretariado de Producción-Investigación de la UCI realiza un resumen con los resultados de mismo.</p> <p>1.13 El Miembro del secretariado de Producción-Investigación de la UCI le entrega el resumen del evento en la Universidad a la Presidencia.</p>
--	--

	2 La Presidencia de la FEU de la UCI recibe la información concluyendo así el caso de uso.	
--	--	--

Tabla 4 Especificación del caso de uso del negocio Gestionar la Emulación de los Evento Científico.

Nombre del Caso de Uso del Negocio:	Obtener la Emulación de los Eventos Científicos.	
Actor(es):	Presidencia de la FEU nivel UCI.	
Trabajador(es):	Responsable de Emulación nivel UCI Responsable de Emulación nivel de facultad	
Breve Descripción:	Se inicia cuando desde la Presidencia de la FEU a nivel de Universidad se solicita gestionar la emulación en la UCI. El Responsable de Emulación nivel UCI recibe la información y le indica a los Responsables de Emulación nivel de facultad sobre de la gestión emulación. Después de realizados los eventos a nivel de facultad, los responsables del tema realizan una síntesis de los resultados de emulación y se lo informan al Responsable de Emulación nivel de facultad, se le envía el informe al Responsable de Emulación nivel UCI, este seria el encargado de realizar un resumen después de terminados los eventos a nivel UCI y dicho resumen de la emulación se le presenta a la Presidencia de la FEU en la UCI.	
Precondición:	Deben haberse culminado los eventos científicos y tener resultados tangibles.	
Post-condición:	Debe obtener un registro de los resultados de la emulación los eventos científicos.	
Flujo Normal de Eventos:	Acción del Actor.	Respuesta del Negocio.
	1 La Presidencia de la FEU nivel UCI solicita la gestión de la emulación de los eventos científicos.	1.1 El Responsable de la

		<p>emulación del Universidad recibe la solicitud de la emulación.</p> <p>1.2 El Responsable de la emulación del Universidad elabora la orientación de la emulación de los eventos.</p> <p>1.3 El Responsable de la emulación de la facultad reciben la orientación emulación de los eventos.</p> <p>1.4 El Responsable de emulación de la facultad al concluir el evento de la facultad elabora un resumen con los resultados de la emulación del evento científico.</p> <p>1.5 El Responsable de la emulación de la facultad entregar resumen de la emulación de eventos científico al Responsable de emulación de la Universidad.</p> <p>1.6 El Responsable de emulación de la Universidad recibe el resumen de emulación.</p> <p>1.7 El Responsable de emulación de la UCI al concluir el evento de la Universidad elabora un resumen con los resultados de la emulación del evento</p>
--	--	--

		científico. 1.8 El Responsable de la emulación de la UCI entregar resumen de la emulación de eventos científico a la Presidencia de la FEU de la UCI
	2 La Presidencia de la FEU de la UCI recibe la información concluyendo así el caso de uso.	

2.2.5 Modelo de Objetos del Negocio.

El modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos.

Para Gestionar los Eventos Científicos de FEU en la UCI, tenemos los siguientes trabajadores y entidades:

Trabajadores de Negocio:

- El Miembro del secretariado de Producción-Investigación de la UCI
- El Miembro del secretariado de Producción-Investigación de la facultad.
- El Responsable de la emulación de la UCI
- El Responsable de la emulación de la facultad

Entidades del Negocio:

- Orientación del Eventos Científicos
- Resumen Resultados Evento Científico Universidad
- Resumen Resultados Evento Científico Facultad
- Orientación de la Emulación de Evento
- Resumen de la Emulación del Evento Universidad
- Resumen de la Emulación del Evento Facultad.

El Diagrama de Clases del Modelo de Objetos del Negocio ver en la figura 5 de los Anexos.

2.3 Especificación de los requisitos del software.

2.3.1 Dependencias y relaciones

El sistema debe ser accesible desde el portal del Web de Kainos, donde existen otras funcionalidades relacionadas con el trabajo de la FEU. Se debe autenticar en el sistema de la Universidad donde cada persona tiene un usuario único.

2.3.2 Requisitos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requerimientos funcionales del sistema son los siguientes:

1. Gestionar la orientación del evento.
2. Revisar la orientación del evento.
3. Gestionar un evento científico.
4. Gestionar una comisión.
5. Gestionar un equipo.
6. Gestionar un jurado.
7. Obtener resultados del evento científico.
8. Mostrar reporte resultados del evento científico..
9. Obtener emulación del evento científico.
10. Mostrar reporte de emulación del evento científico.

2.3.3 Requisitos no Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener el sistema. Los requerimientos no funcionales que debe cumplir el sistema son los siguientes:

Usabilidad e Interfaz: El sistema debe tener una interfaz sencilla que le permita al usuario trabajar oportunamente en los eventos realizados.

Rendimiento, Soporte, Portabilidad: La aplicación funcionará a través de un portal Web por lo que es necesario un servidor eficaz y la conectividad interna de las redes de la Universidad, teniendo acceso por navegador Web; de acuerdo con los sistemas operativos utilizados podemos usar Netscape para el sistema operativo Linux y el Internet Explorer por Microsoft Windows, aunque para ambos recurriremos a Mozilla Firefox.

Seguridad, Confiabilidad: La seguridad será basada en un modelo de usuario-contraseña utilizando los diferentes niveles de accesos y esta información se encriptará para su confianza de los miembros que trabajen la aplicación, debido que al ser factible el portal Web desde la red, los servicios de la aplicación estarán disponibles para todos los usuarios.

Ayuda y documentación en línea: Del portal se obtendrá con un manual de usuario para explicar el uso de la aplicación y sus funciones.

2.4 Modelo de Sistema.

2.4.1 Definición de los Actores del Sistema.

Los actores del sistema son los trabajadores del negocio y ciertos actores del negocio que interactúan de alguna forma con el sistema. En la tabla siguiente son mencionados los actores de nuestro sistema:

Actor(es)	Justificación
Miembro del secretariado de producción e investigación de la Universidad	Miembro del secretariado de producción e investigación del centro es el encargado de organizar los eventos realizado por el centro y obtener los resultados en la Universidad.
Miembro del secretariado de producción e investigación de la facultad	Miembro del secretariado de producción e investigación de la facultad es el encargado de organizar los eventos realizado por el centro y obtener los resultados en la facultad.
Responsable de Emulación de la facultad.	Responsable de Emulación de la facultad es el encargado de obtener la emulación entre los equipos.
Responsable de Emulación de la Universidad	Responsable de Emulación de la Universidad es el encargado de obtener la emulación entre las facultades.

Tabla 5 Actores del Sistema.

2.4.2 Definición de los Casos de Usos del Sistema.

Los Casos de Uso del Sistema son procesos que responden a las funcionalidades definidas en los requerimientos funcionales. En las tablas siguientes se realiza una breve descripción de los Casos de Uso del sistema.

CU1	Gestionar la orientación del evento.
Actor	Miembro del Secretariado de la UCI
Descripción	Este caso de uso le permite al actor, eliminar, insertar y modificar la información sobre preparación los eventos.
Referencia	R1

Tabla 6 Descripción del CUS1

CU2	Revisar la orientación del evento.
Actor	Miembro del Secretariado de la Facultad
Descripción	Este caso de uso le permite al actor revisar la orientación del evento.
Referencia	R2

Tabla 7 Descripción del CUS2

CU3	Gestionar un evento científico.
Actor	Miembro del Secretariado de Producción e Investigación.
Descripción	Este caso de uso le permite al actor; eliminar, insertar y modificar un evento.
Referencia	R3

Tabla 8 Descripción del CUS3

CU4	Gestionar una comisión
Actor	Miembro del Secretariado de Producción e Investigación
Descripción	Este caso de uso le permite al actor; eliminar, insertar y modificar una comisión.
Referencia	R4

Tabla 9 Descripción del CUS4

CU5	Gestionar un jurados
Actor	Miembro del Secretariado de Producción e Investigación
Descripción	Este caso de uso le permite al actor; eliminar, insertar y modificar un jurado de la comisión.
Referencia	R5

Tabla 10 Descripción del CUS5

CU6	Gestionar un equipo
Actor	Miembro del Secretariado de Producción e Investigación
Descripción	Este caso de uso le permite al actor; eliminar, insertar y modificar el equipo que participa en el evento.
Referencia	R6

Tabla 11 Descripción del CUS6

CU7	Obtener resultados del evento científico.
Actor	Miembro del Secretariado de Producción e Investigación
Descripción	Este caso de uso le permite al actor obtener los resultados de cada comisión.

Referencia	R7
-------------------	----

Tabla 12 Descripción del CUS7

CU8	Mostrar reporte, resultados del evento científico.
Actor	Miembro del Secretariado de Producción e Investigación
Descripción	Este caso de uso le permite al actor mostrar un reporte con los resultados del evento.
Referencia	R8

Tabla 13 Descripción del CUS8

CU9	Obtener emulación del evento científico.
Actor	Responsable de Emulación
Descripción	Este caso de uso le permite al actor obtener la emulación del evento.
Referencia	R9

Tabla 14 Descripción del CUS9

CU10	Mostrar reporte de emulación del evento científico.
Actor	Responsable de Emulación
Descripción	Este caso de uso le permite al actor mostrar un reporte con la emulación del evento.
Referencia	R10

Tabla 15 Descripción del CUS10

2.4.3 Distribución de los Casos de Usos por ciclo.

Cod	Nombre del Caso de Uso	Paquete	Justificación de la selección.
1	Gestionar la orientación del evento. Revisar las orientaciones del evento. Gestionar un evento científico. Gestionar una comisión. Gestionar un equipo. Gestionar Jurados. Obtener resultados. Generar reporte resultados. Obtener emulación del evento. Generar reporte de emulación.	1	Estos son los casos de usos más generales y de máxima prioridad para la organización gestionar sus procesos. Constituyen parte básica de la arquitectura de la aplicación.

Tabla 16 Distribución de los Casos de Usos por ciclo

2.4.4 Diagrama y descripción expandida de los Casos de Uso del Sistema.

Un diagrama de casos de uso del sistema representa gráficamente a los casos de uso y su interacción con los actores. El Diagrama de Casos de Uso del Sistema del sistema lo podemos ver en el figura 6 de los Anexos.

Tabla 17 Descripción de Casos de Uso del Sistema Gestionar Orientación del evento.

Caso de Uso del Sistema:	
R1	Gestionar la orientación del evento.
Propósito:	Crear, modificar o eliminar la orientación para los preparativos del evento científico estudiantil y la emulación.
Actor(es):	Miembro del secretariado de producción e investigación de Universidad. Responsable de emulación de la Universidad
Resume:	El caso de uso inicia cuando el Miembro del secretariado la Universidad desea crear, modificar o eliminar la orientación del evento.
Referencias:	
Flujo Normal de Eventos	
Sección “Insertar datos de la orientación”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor solicita crear una orientación. 2. El actor escribe los datos de la orientación y presiona el botón insertar. 	<ol style="list-style-type: none"> 1.1. El sistema visualiza la página para insertar los datos de la orientación. 2.1. El sistema verificar si los campos se hallan llenado correctamente y adicionar el nuevo evento y muestra un mensaje indicando la nueva orientación.
Sección “Modificar datos de la orientación.”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor selecciona la opción modificar orientación. 2. El actor realiza presiona en la orientación que desea modificar. 3. El acto realiza los cambios de los datos de la orientación. 	<ol style="list-style-type: none"> 1.1 El sistema visualiza el listado de las orientaciones. 2.1 El sistema muestra los datos de la orientación que desea modificar. 3.1 El sistema verifica que los datos sean correctos, cambia los datos correspondientes y muestra el listado de las orientaciones.
Sección “Eliminar orientación.”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar orientación. 2. El actor realiza presiona en el orientación que desea eliminar. 	<ol style="list-style-type: none"> 1.1 El sistema visualiza el listado de las orientaciones. 2.1 El sistema elimina el evento y muestra el listado de las orientaciones.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 18 Descripción de Casos de Uso del Sistema Revisar las orientaciones del evento.

Caso de Uso del Sistema:

R2	Revisar las orientaciones del evento.	
Propósito:	Revisar las orientaciones sobre el evento y su emulación.	
Actor(es):	Miembro del secretariado de producción e investigación de facultad. Responsable de emulación de la facultad	
Resume:	El caso de uso inicia cuando los actores desean revisar las orientaciones sobre el evento y su emulación.	
Referencias:	R1	
Flujo Normal de Eventos		
Sección “Recibir orientación”		
Acción de actor	Respuesta del sistema	
1. El actor selecciona la opción revisar orientación.	1.1 El sistema visualiza la orientación del evento	
Flujo alternativo		
Acción de actor	Respuesta del sistema	
Punto de extensión		

Tabla 19 Descripción de Casos de Uso del Sistema Gestionar un evento científico.

Caso de Uso del Sistema:		
R3	Gestionar un evento científico.	
Propósito:	Crear, modificar y eliminar los datos de un evento científico estudiantil que puede ser a nivel de facultad o a nivel de Universidad	
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.	
Resume:	El caso de uso inicia cuando el actor desea crear, modificar o eliminar los datos del evento.	
Referencias:	R1,R2	
Flujo Normal de Eventos		
Sección “Insertar evento”		
Acción de actor	Respuesta del sistema	
1. El actor solicita crear un evento científico. 2. El actor escribe los datos del evento y presiona el botón insertar.	1.1. El sistema visualiza la página para insertar los datos del evento. 2.1. El sistema verificar si los campos se hallan llenado correctamente y adicionar el nuevo evento y muestra un mensaje indicando el nuevo evento.	
Sección “Modificar evento”		
Acción de actor	Respuesta del sistema	
1. El actor selecciona la opción modificar evento. 2. El actor realiza presiona en el evento que desea modificar. 3. El acto realiza los cambios de los datos del evento	1.1 El sistema visualiza el listado de los eventos. 2.1 El sistema muestra los datos del evento que desea modificar. 3.1 El sistema verifica que los datos sean correctos, cambia los datos correspondientes y muestra el listado de los eventos.	

Sección “Eliminar evento”	
Acción de actor	Respuesta del sistema
1. El actor selecciona la opción eliminar evento. 2. El actor realiza presiona en el evento que desea eliminar.	1.1 El sistema visualiza el listado de los eventos. 2.1 El sistema elimina el evento y muestra el listado de los eventos.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 20 Descripción de Casos de Uso del Sistema Gestionar una comisión.

Caso de Uso del Sistema:	
R4	Gestionar una comisión.
Propósito:	Crear, modificar y eliminar los datos del una comisión del evento que puede ser a nivel de facultad o a nivel de Universidad
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.
Resume:	El caso de uso inicia cuando el actor desea crear, modificar o eliminar los datos de la comisión del evento.
Referencias:	R3
Flujo Normal de Eventos	
Sección “Insertar comisión”	
Acción de actor	Respuesta del sistema
1. El actor solicita insertar una comisión del evento. 2. El actor escribe los datos de la comisión y presiona el botón insertar.	1.1. El sistema visualiza la página para insertar los datos de la comisión. 2.1. El sistema verificar si los campos se hallan llenado correctamente y adicionar el nuevo comisión en el evento y muestra un mensaje indicando el nuevo evento.
Sección “ Modificar comisión”	
Acción de actor	Respuesta del sistema
1. El actor selecciona la opción modificar comisión. 2. El actor realiza presiona en la comisión que desea modificar. 3. El acto realiza los cambios de los datos de la comisión.	1.1 El sistema visualiza el listado de las comisiones. 2.1 El sistema muestra los datos de la comisión que desea modificar. 3.1 El sistema verifica que los datos sean correctos, cambia los datos correspondientes y muestra el listado de la comisión.
Sección “Eliminar comisión”	
Acción de actor	Respuesta del sistema
1. El actor selecciona la opción eliminar comisión. 2. El actor realiza presiona en la comisión que desea eliminar.	1.1 El sistema visualiza el listado de las comisiones. 2.1 El sistema elimina el evento y muestra el listado de las comisiones.
Flujo alternativo	
Acción de actor	Respuesta del sistema

Punto de extensión

Tabla 21 Descripción de Casos de Uso del Sistema Gestionar un equipo.

Caso de Uso del Sistema:	
R5	Gestionar un equipo.
Propósito:	Crear, modificar y eliminar los datos un equipo que participa en el evento puede ser a nivel de facultad o a nivel de Universidad
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.
Resume:	El caso de uso inicia cuando el actor desea crear, modificar o eliminar los datos del equipo.
Referencias:	R3,R4
Flujo Normal de Eventos	
Sección “Insertar Equipo”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor solicita insertar el equipo 2. El actor escribe los datos del equipo y presiona el botón insertar. 3. El actor escribe los datos del integrante y presiona el botón insertar. 	<ol style="list-style-type: none"> 1.1. El sistema visualiza la página para insertar los datos del jurado. 2.1. El sistema verificar si los campos se hallan llenado correctamente y adicionar el nuevo evento y muestra el formulario de insertar integrante. 3.1 El sistema verificar si los campos se hallan llenado y muestra un mensaje que ha insertado nuevo equipo.
Sección “Modificar equipo”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor solicita actualizar el equipo. 2. El actor seleccionar dato del equipo que desea actualizar 	<ol style="list-style-type: none"> 1.1. El sistema visualiza listado de equipos 2.1. El sistema actualiza los campos modificados y muestra un mensaje que ha actualizado el equipo.
Sección “Eliminar equipo”	
Acción de actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar equipo. 2. El actor realiza presiona en el equipo que desea eliminar. 	<ol style="list-style-type: none"> 1.1 El sistema visualiza el listado de los equipos. 2.1 El sistema elimina el evento y muestra el listado de los equipos.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 22 Descripción de Casos de Uso del Sistema Gestionar Jurado.

Caso de Uso del Sistema:	
R6	Gestionar Jurados.
Propósito:	Crear, modificar y eliminar los datos del jurado de una comisión

	del evento que puede ser a nivel de facultad o a nivel de Universidad	
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.	
Resume:	El caso de uso inicia cuando el actor desea crear, modificar o eliminar los datos del jurado de una comisión.	
Referencias:	R3,R4	
Flujo Normal de Eventos		
Sección “Insertar jurado”		
Acción de actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. El actor solicita insertar el jurado. 2. El actor escribe los datos del evento y presiona el botón insertar. 	<ol style="list-style-type: none"> 1.1. El sistema visualiza la página para insertar los datos del jurado. 2.1. El sistema verificar si los campos se hallan llenado correctamente y adicionar el nuevo evento y muestra un mensaje indicando el nuevo jurado. 	
Sección “Modificar jurado”		
Acción de actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. El actor selecciona la opción modificar jurado. 2. El actor realiza presiona en el jurado que desea modificar. 3. El acto realiza los cambios de los datos del jurado. 	<ol style="list-style-type: none"> 1.1 El sistema visualiza el listado de los jurados. 2.1 El sistema muestra los datos del jurado que desea modificar. 3.1 El sistema verifica que los datos sean correctos, cambia los datos correspondientes y muestra el listado del jurado. 	
Sección “Eliminar jurado”		
Acción de actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar jurado. 2. El actor realiza presiona en el jurado que desea eliminar. 	<ol style="list-style-type: none"> 1.1 El sistema visualiza el listado de los jurados. 2.1 El sistema elimina el evento y muestra el listado de los jurados. 	
Flujo alternativo		
Acción de actor	Respuesta del sistema	
Punto de extensión		

Tabla 23 Descripción de Casos de Uso del Sistema Obtener resultados del evento.

Caso de Uso del Sistema:	
R7	Obtener resultados del evento.
Propósito:	Obtener los resultados de la cada comisión del evento que puede ser a nivel de facultad o a nivel de Universidad.
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.
Resume:	El caso de uso inicia cuando el actor desea obtener los resultados de la cada comisión del evento.

Referencias:	R3,R4,R5
Flujo Normal de Eventos	
Sección “Obtener resultados”	
Acción de actor	Respuesta del sistema
1. El actor solicita obtener resultados del evento	1.1. El sistema visualiza los resultados del evento.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 24 Descripción de Casos de Uso del Sistema Mostrar Reporte de Resultados.

Caso de Uso del Sistema:	
R8	Mostrar reporte resultados
Propósito:	Mostrar un reporte con los resultados del evento finalizado.
Actor(es):	Miembro del secretariado de producción e investigación de la Universidad. Miembro del secretariado de producción e investigación de la facultad.
Resume:	El caso de usos se inicia cuando el actor desee mostrar el reporte los resultados obtenidos del evento.
Referencias:	R7
Flujo Normal de Eventos	
Sección “Reporte de resultado”	
Acción de actor	Respuesta del sistema
1. El actor solicita obtener reporte resultados del evento	1.1. El sistema visualiza el reporte de resultados del evento.
Sección	
Acción de actor	Respuesta del sistema
Sección	
Acción de actor	Respuesta del sistema
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 25 Descripción de Casos de Uso del Sistema Obtener Emulación del Evento

Caso de Uso del Sistema:	
R9	Obtener emulación del evento.
Propósito:	Obtener la emulación del evento que puede ser a nivel de facultad o a nivel de Universidad
Actor(es):	Responsable de Emulación de la Universidad Responsable de Emulación de la Facultad
Resume:	El caso de uso inicia cuando el actor desea obtener la emulación del evento.
Referencias:	R7,R8

Flujo Normal de Eventos	
Sección “Obtener emulación.”	
Acción de actor	Respuesta del sistema
1. El actor solicita obtener emulación del evento	1.1. El sistema visualiza la emulación del evento.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Tabla 26 Descripción de Casos de Uso del Sistema Mostrar Reporte de Emulación.

Caso de Uso del Sistema:	
R10	Mostrar reporte de emulación.
Propósito:	Mostrar Reporte de emulación del evento finalizado
Actor(es):	Responsable de Emulación de la Universidad Responsable de Emulación de la Facultad
Resume:	El caso de usos se inicia cuando el actor desee mostrar el reporte con la emulación obtenida del evento
Referencias:	R9
Flujo Normal de Eventos	
Sección “Reporte de emulación”	
Acción de actor	Respuesta del sistema
1. El actor solicita obtener reporte emulación del evento	1.1. El sistema visualiza el reporte de emulación del evento.
Flujo alternativo	
Acción de actor	Respuesta del sistema
Punto de extensión	

Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, y a través de los procesos de negocio y el levantamiento de requisitos, se pudo obtener un listado con las funciones que debe tener el sistema y se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. A partir de esto se comenzará a desarrollar la etapa posterior que sería el análisis y diseño.

Capítulo 3 Análisis y Diseño del Sistema.

Introducción.

En este capítulo vamos a traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos, además de algunos aspectos de diseño y funcionalidades del sistema.

3.1 Modelo Análisis.

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Estas clases son clasificadas de las siguientes maneras:

- **Las clases Interfaz** son aquellas que permiten al actor interactuar con el sistema, pueden ser una pantalla, reporte, formulario etc.
- **Las clases de Control** se utilizan para representar coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. También se utilizan para representar cálculos complejos, como la lógica del negocio, se pueden identificar en los casos de uso como verbos.
- **Las clases Entidad** representan una información o un dato persistente; suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.

En las figuras de la 7 a la 16 de los Anexos se representan los diagramas de clases del Análisis de los Casos de Uso del Sistema.

3.2 Modelo Diseño.

El diseño es un modelo de objetos que describe la realización física de los casos de uso y se centra en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a considerar.

3.2.1 Diagrama de Clase del Diseño. Descripción de las clases del diseño.

El Diagrama de Clase del Diseño es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema. Representa las clases y sus relaciones. Ver en la figura 37 de los Anexo.

3.2.2 Diagrama de Interacciones.

Los Diagramas de Interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

1. Modelan el comportamiento dinámico del sistema; el flujo de control en una operación.
2. Describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas.
3. Las interacciones provee un “comportamiento” y típicamente implementan un Caso de Uso.
4. Existen dos tipos de diagramas de interacción en UML
 - **Diagramas de Secuencia:** es un dibujo que muestra, para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas
 - **Diagramas de Colaboración:** son los que proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros.

Vea entre el figura 17 hasta el 36 de los Anexos; todos los Diagramas de Secuencia Diseño.

3.2.3 Descripciones de las clases.

3.2.3.1 Clases Entidades

Nombre	Orientacion
Tipo de Clase	Entidad
Atributo	Tipo
idOrientacion	Int
nombreOrientacion	String
tipoOrientacion	String
descripcion	String
Para cada responsabilidad	
Nombre	Descripción

Tabla 27 Descripción de la clase entidad Orientación.

Nombre	Evento
Tipo de Clase	Entidad
Atributo	Tipo

idEvento	Int
nombreEvento	String
orientacion	Int
cronograma	String
Para cada responsabilidad	
Nombre	Descripción

Tabla 28 Descripción de la clase entidad Evento.

Nombre	Comision
Tipo de Clase	Entidad
Atributo	Tipo
idComision	Int
nombreComision	String
evento	Int
Para cada responsabilidad	
Nombre	Descripción

Tabla 29 Descripción de la clase entidad Comisión

Nombre	Jurado
Tipo de Clase	Entidad
Atributo	Tipo
idJurado	Int
nombreJurado	String
apellidoJurado	String
comision	Int
Para cada responsabilidad	
Nombre	Descripción

Tabla 30 Descripción de la clase entidad Jurado.

Nombre	Equipo
Tipo de Clase	Entidad
Atributo	Tipo
idEquipo	Int
nombreEquipo	String
cantIntegrante	Int
comision	Int
resultado	Int
trabajoEquipo	String
Para cada responsabilidad	
Nombre	Descripción

Tabla 31 Descripción de la clase entidad Equipo.

Nombre	Integrante
Tipo de Clase	Entidad
Atributo	Tipo
idIntegrante	Int

nombreIntegrante	String
apellidoIntegrante	String
facultad	String
equipo	Int
grupo	String
Para cada responsabilidad	
Nombre	Descripción

Tabla 32 Descripción de la clase entidad Integrante.

Nombre	Resultado
Tipo de Clase	Entidad
Atributo	Tipo
IdResultado	Int
nombreResultado	String
valor	Int
Para cada responsabilidad	
Nombre	Descripción

Tabla 33 Descripción de la clase entidad Resultado.

3.2.3.2 Clases Dao

Nombre	OrientacionDao
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarOrientacion() eliminarOrientacion() obtenerOrientacion() obtenerTodasOrientaciones() actualizarOrientacion()	Llamar el método de la clase de la Clase OrientacionDaoImpl

Tabla 34 Descripción de la clase OrientacionDao.

Nombre	OrientacionDaoImpl
Tipo de Clase	Control
Atributo	Tipo
orientacion	Orientacion
Para cada responsabilidad	
Nombre	Descripción
guardarOrientacion() eliminarOrientacion() obtenerOrientacion() obtenerTodasOrientaciones() actualizarOrientacion()	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos.

Tabla 35 Descripción de la clase OrientaciónDaoImpl.

Nombre	EventoDao
Tipo de Clase	Control

Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardar Evento() eliminar Evento() obtener Evento() obtenerTodosEventos() actualizar Evento()	Llamar el método de la clase de la Clase EventoDaoImpl

Tabla 36 Descripción de la clase EventoDao.

Nombre	EventoDaoImpl
Tipo de Clase	Control
Atributo	Tipo
evento	Evento
Para cada responsabilidad	
Nombre	Descripción
guardarEvento() eliminarEvento() obtenerEvento() obtenerTodosEvento() actualizarEvento()	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos.

Tabla 37 Descripción de la clase EventoDaoImpl.

Nombre	ComisionDao
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarComision() eliminarComision() obtenerComision() obtenerTodasComisiones() actualizarComision() obtenerTodasComisionE(int E)	Llamar el método de la clase ComisionDaoImpl

Tabla 38 Descripción de la clase ComisionDao.

Nombre	ComisionDaoImpl
Tipo de Clase	Control
Atributo	Tipo
comision	Comision
Para cada responsabilidad	
Nombre	Descripción
guardarComision() eliminarComision() obtenerComision() obtenerTodasComision() actualizarComision() obtenerTodasComisionE(int E)	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos. Obtener todas las comisiones de un evento

Tabla 39 Descripción de la clase ComisionDaolmpl.

Nombre	JuradoDao
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarJurado() eliminarJurado() obtenerJurado() obtenerTodosJurado() actualizarJurado()	Llamar el método de la clase JuradoDaolmpl

Tabla 40 Descripción de la clase JuradoDao.

Nombre	JuradoDaolmpl
Tipo de Clase	Control
Atributo	Tipo
jurado	Jurado
Para cada responsabilidad	
Nombre	Descripción
guardarJurado() eliminarJurado() obtenerJurado() obtenerTodosJurado() actualizarJurado()	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos.

Tabla 41 Descripción de la clase JuradoDaolmpl.

Nombre	EquipoDao
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarEquipo() eliminarEquipo() obtenerEquipo() obtenerTodosEquipo() actualizarEquipo() obtenerTodosEquipoC	Llamar el método de la clase EquipoDaolmpl

Tabla 42 Descripción de la clase EquipoDao.

Nombre	EquipoDaolmpl
Tipo de Clase	Control
Atributo	Tipo
equipo	Equipo
Para cada responsabilidad	
Nombre	Descripción
guardarEquipo() eliminarEquipo()	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos

obtenerEquipo() obtenerTodosEquipo() actualizarEquipo() obtenerTodosEquipoC	Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos. Obtener todos los equipos dados una comisión y un resultado.
--	--

Tabla 43 Descripción de la clase EquipoDaoImpl.

Nombre	IntegranteDao
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarIntegrante() eliminarIntegrante() obtenerIntegrante() obtenerTodosIntegrante() actualizarIntegrante()	Llamar el método de la clase IntegranteDaoImpl

Tabla 44 Descripción de la clase IntegranteDao

Nombre	IntegranteDaoImpl
Tipo de Clase	Control
Atributo	Tipo
integrante	Integrante
Para cada responsabilidad	
Nombre	Descripción
guardarIntegrante() eliminarIntegrante() obtenerIntegrante() obtenerTodosIntegrante actualizarIntegrante obtenerTodosIntegranteQ(int idequipo)	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos. obtener todos los integrante de un equipo

Tabla 45 Descripción de la clase IntegranteDaoImpl.

Nombre	ResultadoDao
Tipo de Clase	
Atributo	Tipo
	Control
Para cada responsabilidad	
Nombre	Descripción
guardarResultado eliminarResultado() obtenerResultado() obtenerTodosResultados() actualizarResultado()	Llamar el método de la clase ResultadoDaoImpl

Tabla 46 Descripción de la clase ResultadoDao.

Nombre	ResultadoDaoImpl
Tipo de Clase	Control
Atributo	Tipo

resultado	Resultado
Para cada responsabilidad	
Nombre	Descripción
guardarResultado() eliminarResultado() obtenerResultado() obtenerTodosResultados() actualizarResultado()	Insertar el objeto en la Base de Datos Eliminar el objeto en la Base de Datos Obtener el objeto en la Base de Datos Obtener todos los objeto en la Base de Datos Actualizar el objeto en la Base de Datos.

Tabla 47 Descripción de la clase ResultadoDaolmpl.

3.3.2.3 Clase de Servicios

Nombre	ServiciosEventosCientificos
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
guardarOrientacion() guardarEvento() guardarComisio() guardarJurado() guardarEquipo() guardarIntegrante() guardarResultado() eliminarOrientacion() eliminarEvento() eliminarComision() eliminarJurado() eliminarEquipo() eliminarIntegrante() eliminarResultado() obtenerOrientacion() obtenerEvento() obtenerComision() obtenerJurado() obtenerEquipo() obtenerIntegrante() obtenerResultado() obtenerTodasOrientaciones() obtenerTodosEvento() obtenerTodasComision() obtenerTodosJurados() obtenerTodosEquipos() obtenerTodosIntegrantes() obtenerTodosResultados() actualizarOrientacion() actualizarEvento() actualizarComision() actualizarJurado() actualizarEquipo()	Estos métodos llaman a los métodos de la clase ServiciosEventosCientificosImpl.

actualizarIntegrante() actualizarResultado() obtenerTodosEquipoC obtenerTodosIntegranteQ(int idequipo) obtenerTodasComisionE(int E)	
---	--

Tabla 48 Descripción de la clase ServiciosEventosCientificos.

Nombre	ServiciosEventosCientificosImpl
Tipo de Clase	Control
Atributo	Tipo
orientaciondao	Orientaciondao
eventodao	Eventodao
comisiondao	Comisionado
juradodao	Juradodao
equipodao	Equipodao
integrantedao	Integrantedao
resultadodao	Resultadodao
Para cada responsabilidad	
Nombre	Descripción
guardarOrientacion() guardarEvento() guardarComisio(n) guardarJurado() guardarEquipo() guardarIntegrante() guardarResultado() eliminarOrientacion() eliminarEvento() eliminarComision() eliminarJurado() eliminarEquipo() eliminarIntegrante() eliminarResultado() obtenerOrientacion() obtenerEvento() obtenerComision() obtenerJurado() obtenerEquipo() obtenerIntegrante() obtenerResultado() obtenerTodasOrientaciones() obtenerTodosEvento() obtenerTodasComision() obtenerTodosJurados() obtenerTodosEquipos() obtenerTodosIntegrantes() obtenerTodosResultados() actualizarOrientacion() actualizarEvento() actualizarComision() actualizarJurado()	Estos métodos llaman a los métodos de las clases Dao.

actualizarEquipo() actualizarIntegrante() actualizarResultado() obtenerTodosEquipoC obtenerTodosIntegranteQ(int idequipo) obtenerTodasComisionE(int E)	
---	--

Tabla 49 Descripción de la clase ServiciosEventosCientificosImpl

3.3.2.4 Clases ServiceLocator

Nombre	ServiceLocator
Tipo de Clase	Control
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción
getServiciosEventosCientificos() setServiciosEventosCientificos()	Estos métodos llaman a los métodos de la clase ServiceLocatorImpl

Tabla 50 Descripción de la clase ServiceLocator.

Nombre	ServiceLocatorImpl
Tipo de Clase	Control
Atributo	Tipo
serviciosEventosCientificos	serviciosEventosCientificos
Para cada responsabilidad	
Nombre	Descripción
getServiciosEventosCientificos() setServiciosEventosCientificos()	obtener y modificar los objetos de clase ServiciosEventosCientificos

Tabla 51 Descripción de la clase ServiceLocatorImpl.

3.3.2.5 Clase Bean

Nombre	OrientacionBean
Tipo de Clase	Control
Atributo	Tipo
idOrientacion	Int
nombreOrientacion	String
tipoOrientacion	String
descripcion	String
Para cada responsabilidad	
Nombre	Descripción
OrientacionBean() init() getIdOrientacion() setIdOrientacion() getNombreOrientacion() setNombreOrientacion() getTipoOrientacion() setTipoOrientacion() getDescripcion() setDescripcion() insertarOrientacion()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Insertar Objeto

actualizarOrientacion() eliminarOrientacion() obtenerOrientacion(int idcomsion)	Actualizar Objeto Eliminar Objeto Obtener una orientación.

Tabla 52 Descripción de la clase OrientacionBean.

Nombre	ListOrientacionesBean
Tipo de Clase	Control
Atributo	Tipo
listaOrientaciones idOrientacion	List int
Para cada responsabilidad	
Nombre	Descripción
ListOrientacionesBean() init() getListaOrientaciones() setListaOrientaciones() getIdOrientacion() setIdOrientacion()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

Tabla 53 Descripción de la clase ListOrientacionesBean.

Nombre	EventoBean
Tipo de Clase	Control
Atributo	Tipo
idEvento nombreEvento orientacion cronograma	Int String Int String
Para cada responsabilidad	
Nombre	Descripción
EventoBean() init() getIdEvento() setIdEvento() getNombreEvento() setNombreEvento() getOrientacion() setOrientacion() getCronograma() setCronograma() insertarEvento() actualizarEvento() eliminarEvento()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Insertar Objeto Actualizar Objeto Eliminar Objeto

Tabla 54 Descripción de la clase EventoBean

Nombre	ListEventosBean
Tipo de Clase	Control
Atributo	Tipo
listaEventos idEvento	List int

Para cada responsabilidad	
Nombre	Descripción
ListEventosBean() init() getListaEventos() setListaEventos() getIdEvento() setIdEvento()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

Tabla 55 Descripción de la clase ListEventosBean.

Nombre	ComisionBean
Tipo de Clase	Control
Atributo	Tipo
idComision nombreComision evento	Int String Int
Para cada responsabilidad	
Nombre	Descripción
ComisionBean() init() getIdComision() setIdComision() getNombreComision() setNombreComision() getEvento() setEvento() insertarComision() actualizarComision() eliminarComision()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Insertar Objeto Actualizar Objeto Eliminar Objeto

Tabla 56 Descripción de la clase ComisionBean.

Nombre	ListComisionesBean
Tipo de Clase	Control
Atributo	Tipo
listaComision idComision	List int
Para cada responsabilidad	
Nombre	Descripción
ListComisionesBean() init() getListaComisiones() setListaComisiones() getIdComision() setIdComision()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

Tabla 57 Descripción de la clase ListComisionesBean.

Nombre	JuradoBean
Tipo de Clase	Control
Atributo	Tipo
idJurado	Int

nombreJurado	String
apellidoJurado	String
comision	Int
Para cada responsabilidad	
Nombre	Descripción
JuradoBean() init() getIdJurado() setIdJurado() getNombreJurado() setNombreJurado() getApellidoJurado() setApellidoJurado() getComision() setComision() insertarJurado() actualizarJurado() eliminarJurado()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Insertar Objeto Actualizar Objeto Eliminar Objeto

Tabla 58 Descripción de la clase JuradoBean.

Nombre	ListJuradosBean
Tipo de Clase	Control
Atributo	Tipo
listaJurado	List
idJurado	int
Para cada responsabilidad	
Nombre	Descripción
ListJuradosBean() init() getListJurado() setListaJurado() getIdJurado() setIdJurado()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

Tabla 59 Descripción de la clase ListJuradosBean.

Nombre	EquipoBean
Tipo de Clase	Control
Atributo	Tipo
idEquipo	Int
nombreEquipo	String
cantIntegrante	Int
comision	Int
resultado	Int
trabajoEquipo	String
Para cada responsabilidad	
Nombre	Descripción
EquipoBean() init() getIdEquipo() setIdEquipo()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

getNombreEquipo() setNombreEquipo() getCantIntegrante() setCantIntegrante() getComision() setComision() getResultado() setResultado() getTrabajoEquipo() setTrabajoEquipo() insertarEquipo() actualizarEquipo() eliminarEquipo()	Insertar Objeto Actualizar Objeto Eliminar Objeto
--	---

Tabla 60 Descripción de la clase EquipoBean.

Nombre	ListEquiposBean
Tipo de Clase	Control
Atributo	Tipo
listaEquipo	List
idEquipo	int
Para cada responsabilidad	
Nombre	Descripción
ListEquiposBean() init() getListEquipo() setListaEquipo() getIdEquipo() setIdEquipo()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

Tabla 61 Descripción de la clase ListEquiposBean.

Nombre	IntegranteBean
Tipo de Clase	Control
Atributo	Tipo
idIntegrante	Int
nombreIntegrante	String
apellidoIntegrante	String
facultad	String
equipo	Int
grupo	String
Para cada responsabilidad	
Nombre	Descripción
IntegranteBean() init() getIdIntegrante() setIdIntegrante() getNombreIntegrante() setNombreIntegrante() getApellidoIntegrante() setApellidoIntegrante() getFacultad()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

setFacultad() getGrupo() setGrupo() getEquipo() setEquipo() insertarIntegrante() actualizarIntegrante() eliminarIntegrante()	Insertar Objeto Actualizar Objeto Eliminar Objeto
---	---

Tabla 62 Descripción de la clase IntegranteBean.

Nombre	ListResultadosBean
Tipo de Clase	
Atributo	Tipo
listaComisiones	List
listaResultados	List
listaEquipos	List
listaResultados	List
idEvento	int
Para cada responsabilidad	
Nombre	Descripción
ListResultadosBean() init() getListaComisiones() setListaComisiones() getListaResultados() setListaResultados() getListaEquipos() setListaEquipos() getListaResultados() setListaResultados() obtenerTodasComisiones(int E) obtenerTodosResultados() obtenerTodosEquipos(int C, int R) obtenerResultados()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Obtener los listados especificos

Tabla 63 Descripción de la clase ListResultadoBean.

Nombre	ListReporteResultadosBean
Tipo de Clase	
Atributo	Tipo
listaComisiones	List
listaResultados	List
listaEquipos	List
listaIntegrante	List
listaReporteResultados	List
idEvento	int
Para cada responsabilidad	
Nombre	Descripción
ListReporteResultadosBean() init() getListaComisiones()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase.

setListaComisiones() getListaResultados() setListaResultados() getListaEquipos() setListaEquipos() getListaIntegrantes() setListaIntegrantes() getListaReporteResultados() setListaReporteResultados() obtenerTodasComisiones(int E) obtenerTodosResultados() obtenerTodosEquipos(int C, int R) obtenerTodosIntegrantes(int Q) obtenerResultados()	Obtener los listados especificos
---	----------------------------------

Tabla 64 Descripción de la clase ListReporteResultadosBean.

Nombre	ListEmulacionBean
Tipo de Clase	
Atributo	Tipo
listaComisiones	List
listaResultados	List
listaEquipos	List
listaEmulacion	List
Para cada responsabilidad	
Nombre	Descripción
ListEmulacionBean() init() getListaComisiones() setListaComisiones() getListaResultados() setListaResultados() getListaEquipos() setListaEquipos() getListaEmulacion() setListaEmulacion() obtenerTodasComisiones(int E) obtenerTodosResultados() obtenerTodosEquipos(int C, int R) obtenerEmulacion()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Obtener los listados especificos

Tabla 65 Descripción de la clase ListEmulacionBean.

Nombre	ListReporteEmulacionBean
Tipo de Clase	
Atributo	Tipo
listaComisiones	List
listaResultados	List
listaEquipos	List
listaIntegrantes	List
listaReporteEmulacion	List
idEvento	int

Para cada responsabilidad	
Nombre	Descripción
ListReporteEmulacionBean() init() getListaComisiones() setListaComisiones() getListaResultados() setListaResultados() getListaEquipos() setListaEquipos() getListaIntegrantes() setListaIntegrantes() getListaReporteEmulacion() setListaReporteEmulacion() obtenerTodasComisiones(int E) obtenerTodosResultados() obtenerTodosEquipos(int C, int R) obtenerTodosIntegrantes(int Q) obtenerEmulacion()	Constructor Ejecuta el Bean. Obtener y modificar atributo de la clase. Obtener los listados especificos

Tabla 66 Descripción de la clase ListReporteEmulacionBean.

3.3.2.6 Clases Interfaz

Nombre	insertarOrientacion
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 67 Descripción de la clase insertarOrientacion.

Nombre	listaOrientacion
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 68 Descripción de la clase listaOrientacion.

Nombre	actualizarOrientacion
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 69 Descripción de la clase actualizarOrientacion.

Nombre	obtenerOrientacion
Tipo de Clase	Interfaz

Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 70 Descripción de la clase obtenerOrientacion.

Nombre	insertarEvento
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 71 Descripción de la clase insertarEvento.

Nombre	listaEvento
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 72 Descripción de la clase listaEvento.

Nombre	actualizarEvento
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 73 Descripción de la clase actualizarEvento.

Nombre	insertarComision
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 74 Descripción de la clase insertarComision.

Nombre	listaJurado
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

--	--

Tabla 75 Descripción de la clase listaJurado.

Nombre	actualizarJurado
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 76 Descripción de la clase actualizarJurado.

Nombre	insertarEquipo
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 77 Descripción de la clase insertarEquipo.

Nombre	insertarIntegrante
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 78 Descripción de la clase inseratrIntegrante.

Nombre	listaEquipos
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 79 Descripción de la clase listaEquipos.

Nombre	actualizarEquipo
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 80 Descripción de la clase actualizarEquipo.

Nombre	actualizarIntegrante
---------------	----------------------

Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 81 Descripción de la clase actualizarIntegrante.

Nombre	obtenerResultados
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 82 Descripción de la clase obtenerResultado.

Nombre	reporteResultados
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 83 Descripción de la clase reporteResultado.

Nombre	obtenerEmulacion
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 84 Descripción de la clase obtenerEmulacion.

Nombre	reporteEmulacion
Tipo de Clase	Interfaz
Atributo	Tipo
Para cada responsabilidad	
Nombre	Descripción

Tabla 85 Descripción de la clase reporteEmulacion.

3.3 Diseño de la Base de Datos.

La Base de Datos es un conjunto de datos relacionados entre sí, se diseña, construye y desarrolla con datos para propósito específico, posibilita obtener y actualizar los datos siempre que sea necesario.

3.3.1 Diagrama de Entidad Relación de las Bases de Datos.

Las relaciones entre entidades tienen como características, la de poder asociar más de dos tipos de entidades, además establecerse entre un mismo tipo de entidad y por ultimo las mismas entidades pueden ser asociadas en cualquier número de relaciones. El modelo de entidad relación es un modelo de datos semánticos y usa diagramas para representar la estructura natural de los datos. Para ver el Diagrama de Entidad Relación de las Bases de Datos del sistema se recomienda ver la figura 38 de los Anexos.

3.3.2 Descripción de las tablas.

Nombre:	Orientacion	
Descripción:		
Atributos	Tipo	Descripción
idOrientacion	Numerico	Llave primaria
nombreOrientacion	String	Aspecto de la orientación
descripcion	String	Descripción del aspecto
tipoOrientacion	String	Tipo orientación (evento o emulación).

Tabla 86 Descripción del la Tabla Orientación.

Nombre:	Evento	
Descripción:		
Atributos	Tipo	Descripción
idEvento	Numerico	Llave primaria
nombreEvento	String	Nombre del evento.
orientacion	Integer	Llave primaria de T – Orientación
cronograma	String	Dirección del cronograma.

Tabla 87 Descripción del la Tabla Evento.

Nombre:	Comision	
Descripción:		
Atributos	Tipo	Descripción
idComision	Numerico	Llave primaria
nombreComision	String	Nombre de la comisión.
evento	Integer	Identificador al evento que pertenece.

Tabla 88 Descripción del la Tabla Comisión.

Nombre:	Jurado	
Descripción:		
Atributos	Tipo	Descripción
idJurado	Numerico	Llave primaria
nombreJurado	String	Nombre del jurado
apellidoJurado	String	Apellido del jurado.
comisionj	Integer	Identificador al comisión que pertenece.

Tabla 89 Descripción del la Tabla Jurado.

Nombre:	Equipo
----------------	--------

Descripción:		
Atributos	Tipo	Descripción
idEquipo	Numerico	Llave primaria
nombreEquipo	String	Titulo del trabajo.
comisione	Integer	Identificador al comisión que pertenece.
resultado	Integer	Identificador al resultado que tiene.
cantIntegrante	Integer	Cantidad de Participante

Tabla 90 Descripción del la Tabla Equipo.

Nombre:	Integrante	
Descripción:		
Atributos	Tipo	Descripción
idintegrante	Numerico	Llave primaria
nombreintegrante	String	Nombre del integrante.
apellidointegrante	String	Apellidos del integrante.
facultad	Integer	Facultad que pertenece.
equipo	Integer	Identificador al equipo que pertenece
grupo	String	Grupo a que pertenece

Tabla 91 Descripción del la Tabla Integrante.

Nombre:	Resultados	
Descripción:		
Atributos	Tipo	Descripción
idResultado	Numerico	Llave primaria
nombreresultado	String	Nombre del resultado (Relevante, destacado, mención, null)
valor	Integer	Valor de cada resultado

Tabla 92 Descripción del la Tabla Resultado

3.4 Definiciones de Diseño.

La navegación entre las páginas Web debe ser simple, clara y organizada por secciones, donde cada usuario tendrá un acceso a los privilegios de la aplicación. El contenido debe ser simple y claro destinado a usuarios de la FEU de la escuela. Además el fondo de la Interfaz permite una fácil lectura al usuario.



Figura 1 Ejemplo de Interfaz.

3.5 Concepción general de la ayuda.

La ayuda estará accesible como parte del menú en todas las páginas de la aplicación, y con el fin de que el usuario vea solo la información que necesita en ese momento, cada página mostrará como realizar solo aquellas operaciones que se estén realizando en el momento, además se aportarán los conceptos que se manejen en la aplicación, para que el usuario se familiarice con algunas entradas.

3.6 Tratamiento de errores.

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones. Por último, se utilizan errores en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Por el lado del servidor verifica los datos y en caso de que tengan errores se envía en mensaje de error y en con los campos que debe corregir.

Conclusiones.

En este capítulo se realizó el análisis y diseño donde se relaciona los requisitos funcionales y los no funcionales, llevándolo a una expresión de la programación entre las diferentes herramientas mencionadas en capítulos anteriores y algunos aspectos importantes para la necesidad del usuario, a su vez la preparación para la etapa de implementación del sistema.

Capítulo 4 Estimación de Costo del Proyecto.

Introducción

En este capítulo se realiza la estimación de proyectos que permite la realización de evaluaciones más precisas que las obtenidas a través de métodos tradicionales (orientadas a calcular individualmente el esfuerzo correspondiente a cada una de las actividades del mismo).

4.1 Cálculo de Puntos de Casos de Uso sin ajustar.

Para comenzar con la estimación de costo tenemos que partir del Cálculo de los Puntos de Casos de Uso sin ajustar que se deriva de la ecuación siguiente:

$$UUCP = UAW + UUCW$$

- **UUCP:** Puntos de Casos de Uso sin ajustar
- **UAW:** Factor de Peso de los Actores sin ajustar
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

En el cálculo del Factor de Peso de los Actores sin ajustar y del Factor de Peso de los Casos de Uso sin ajustar, se realiza un análisis del estudio de cantidad de actores en el sistema y su complejidad como nos muestra las siguientes tablas:

Tipo de Actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	4	0

Tabla 93 Factor de peso de los actores sin ajustar

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	10	50
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Tabla 94 Factor de peso de los casos de uso sin ajustar.

A partir de los datos obtenidos se obtiene estos valores de la manera siguiente:

- Factor de Peso de los Actores sin ajustar.

El sistema tiene 4 actores complejo. El valor del UAW es:

$$UAW = \sum (\text{Factor} * \text{CantActores})$$

$$UAW = 12$$

- Factor de Peso de los Casos de Uso sin ajustar

El sistema tiene 10 Casos de Uso simple. El valor de la UUCW es

$$UUCW = \sum (\text{Factor} * \text{CantCU})$$

$$UUCW = 50$$

Una vez calculado estos valores procedemos al cálculo de los Puntos de Casos de Uso sin ajustar:

$$UUCP = UAW + UUCW$$

$$UUCP = 62$$

4.2 Cálculo de los Puntos de Casos de Uso ajustados.

El segundo paso para el cálculo de la estimación del costo es el Cálculo de los Puntos de caso de Uso ajustado, para obtener este valor utilizamos la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

- **UCP:** Puntos de casos de uso ajustados.
- **UUCP:** Puntos de casos de uso sin ajustar.
- **TCF:** Factor de complejidad técnica.
- **EF:** Factor de ambiente.

Para calcular el Factor de complejidad técnica y Factor de ambiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En las tablas siguientes se representa estos factores del sistema:

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido.	2	0	0
T2	Tiempo de respuesta.	1	2	2
T3	Eficiencia del usuario final.	1	2	2
T4	Funcionamiento Interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	4	4
T6	Facilidad de instalación.	0.5	0	0

T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	0	0
T9	Facilidad de cambio.	1	3	3
T10	Concurrencia.	1	2	2
T11	Incluye objetivos especiales de seguridad.	1	1	1
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios.	1	1	1

Tabla 95 Factor de complejidad técnica.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en la orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	3	6
E7	Personal Part–Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	1	-1

Tabla 96 Factor de ambiente.

A partir de los datos obtenidos se obtiene estos valores de la manera siguiente:

- Factor de Complejidad Técnica.

$$TCF = 0.6 + 0.01 * \sum (\text{Peso} * \text{Valor})$$

$$TCF = 0.79$$

- Factor de Ambiente.

$$EF = 1.4 - 0.03 * \sum (\text{Peso} * \text{Valor})$$

$$EF = 0.65$$

- Puntos de casos de uso sin ajustar.

$$UUCP = 62$$

Una vez calculado estos valores procedemos al cálculo de los Puntos de Casos de Uso ajustar:

$$UCP = UUCP * TCF * EF$$

$$UCP = 32.03$$

4.3 Estimación de esfuerzo a través de los puntos de casos de uso.

Al obtener el valor del Caso de Uso ajustado el siguiente paso es el calcular la estimación de esfuerzo aplicado la siguiente ecuación:

$$E = UCP * CF$$

- **UCP:** Puntos de casos de uso ajustados.
- **CF:** Factor de Conversión.

Este último, el Factor de Conversión se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

- Factor de Conversión.

$$CF = 20 \text{ Horas-Hombre} / \text{Punto de Casos de uso.}$$

- Puntos de casos de uso ajustados.

$$UCP = 32.03$$

Una vez calculado estos valores procedemos al cálculo de la estimación de esfuerzo.

$$E = UCP * CF$$

$$E = 640.77$$

4.4 Calcular esfuerzo de todo el proyecto.

En el cálculo anterior proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Finalmente, para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable:

Actividad	Porcentaje (%)	Hora-Hombre
Análisis	10	160,19
Diseño	20	320,38
Implementación	40	640,77

Prueba	20	240,28
Sobre(otras actividades)	10	240,28
Estimación Total	100	1601,92

Tabla 97 Porcentajes.

Estimación total de Proyecto es 1601.92 horas-hombres.

4.5 Costo del Proyecto.

Con el cálculo de la estimación total de horas- hombre y asumiendo que cada mes tiene como promedio 176 horas obtendríamos un estimación total de mes- hombre de 9.1, pero si se asume también que el salario es de \$100.

Costo Total = Estimación Total de mes hombre + Salario Promedio.

El Costo Total de proyecto es de \$910.00.

Si en el equipo hay 2 personas y todas realizan el mismo esfuerzo, entonces el problema analizado puede terminarse en aproximadamente 5 meses.

Conclusiones

En este capítulo se realizó la estimación del costo del proyecto software, donde pudimos ver la duración y costo del proyecto, la importancia cada vez mayor que toma la información como factor estratégico ha determinado que sea la duración de un proyecto uno de los aspectos mas prioritarios en su realización. Como síntesis de lo expresado anteriormente puede decirse que el aspecto crítico de los métodos tradicionales de estimación costo del proyectos software radica en la imposibilidad de establecer una unidad de medida para la estimación de la duración total del proyecto como un valor dependiente de las características del mismo y no solo en función de los recursos humanos a emplear.

Conclusiones Generales

Este trabajo demostró la necesidad del análisis y diseño de un sistema de Web que automatice la Gestión de los Eventos Científicos estudiantiles de la FEU en la UCI. Entre los aspectos que se tuvieron en cuenta se encuentra:

- El estudio sobre las tendencias actuales que rodean el problema; y a través de estos conceptos y definiciones planteadas se fundamenta los principios de este trabajo y análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto.
- Se desarrolló la propuesta de solución y se pudo obtener un listado con las funciones que debe tener el sistema y se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan.
- Se relaciona los requisitos funcionales y los no funcionales, llevándolo a una expresión de la programación.
- Se realizó la estimación del costo del proyecto software, donde pudimos ver la duración y costo del proyecto.

Como resultado final se obtiene un modelo de análisis y diseño de un módulo del Proyecto Kainos, que es destinado para la construcción del portal Web de la FEU nacional.

Recomendaciones

Se recomienda la implementación de este estudio realizado, debido a la necesidad de darle solución al problema planteado, además basado en este trabajo se puede confeccionar un sistema Web destinado para los Eventos Científicos Nacionales desarrollado durante el año y a lo largo del país.

Referencias Bibliográficas

1. BOOCH, G. "The Unified Modeling Language User Guide". Addison-Wesley 1999. 482 p.
2. SANCHEZ, M. A. M. "Metodologías De Desarrollo De Software". 2004, Disponible en: www.informatizate.net/articulos/pdfs/.
3. SHEN, D. Y. "Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real". 2006, Disponible en: http://programacion.com/tutorial/jap_jsfwork/.
4. STALLMAN, R. M. "El movimiento del Software Libre y el sistema operativo GNU/Linux" En 2007.

Bibliografía

1. GARCIA, R. M. M. Diseño de Bases de Datos. 1999.
2. GONZÁLEZ, H. S. Manual Hibernate. 2003, Disponible en:
<http://www.javahispano.org/articles.article.action?id=82>.
3. GRADY BOOCH, I. J., JAMES RUMBAUGH. El proceso unificado del desarrollo de software 1999, Disponible en: <http://www.monografias.com/trabajos22/desarrollo-software/desarrollo-software.shtml>.
4. LARMAN, C. La Captura de Requerimientos. Venezuela: Universidad Simón Bolívar, Departamento de Computación y Tecnología de la Información, [Consultado el: 05/04 de 2007]. Disponible en: <http://www ldc.usb.ve/~teruel/ci4712/clases/clase2.html>.
5. MOLPECERES, A. Procesos de desarrollo: RUP, XP y FDD. 2002, Disponible en:
<http://www.javahispano.org/articles.article.action?id=76>.
6. PULIDO, A. S. Introducción al UML. 2001, Disponible en:
<http://www.yoprogramo.com/articulo4.php>.
7. SMALDONE, J. Software Libre versus Software Propietario [Consultado el: 08/12 de 2006]. Disponible en: <http://www.smaldone.com.ar/opinion/slvssp.shtml>.
8. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Segunda ed. 2002. 499 p.
9. VILAS, A. F. Diagramas de Casos de Uso [Consultado el: 24/02 de 2007]. Disponible en:
<http://www.javahispano.org/articles.article.action?id=82>.
10. YOEMNY GONZALEZ, A. Y. P. Sistema para la Informatización de los Servicios Médicos de la UCI. Computación. Universidad de la Habana, 2004.

Anexos

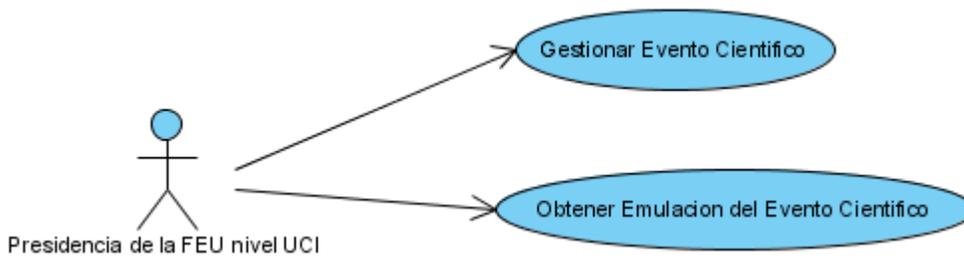


Figura 2 Diagrama de Casos de Uso del Negocio

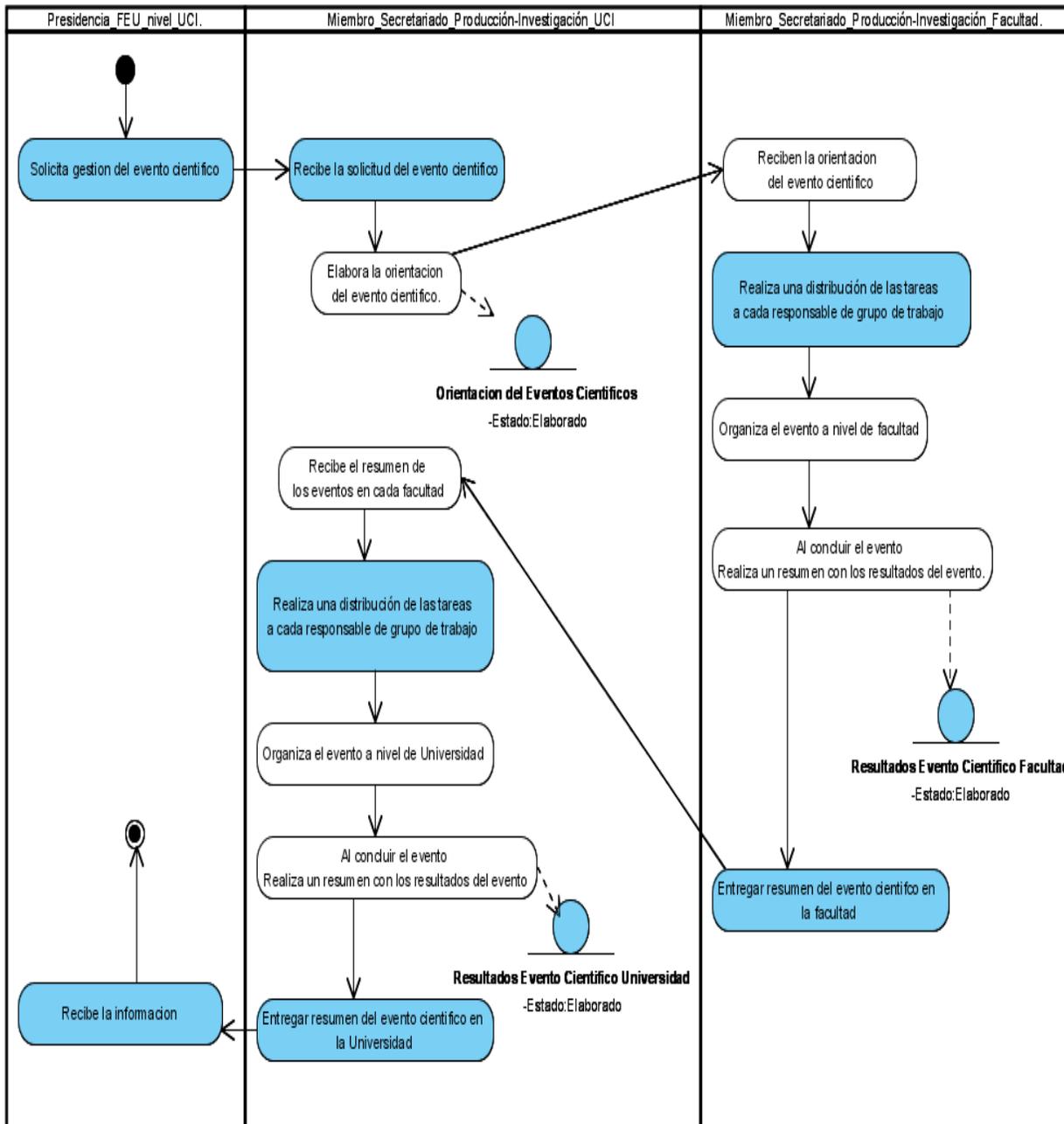


Figura 3 Diagrama de Actividades del CUN Gestionar Evento Científico

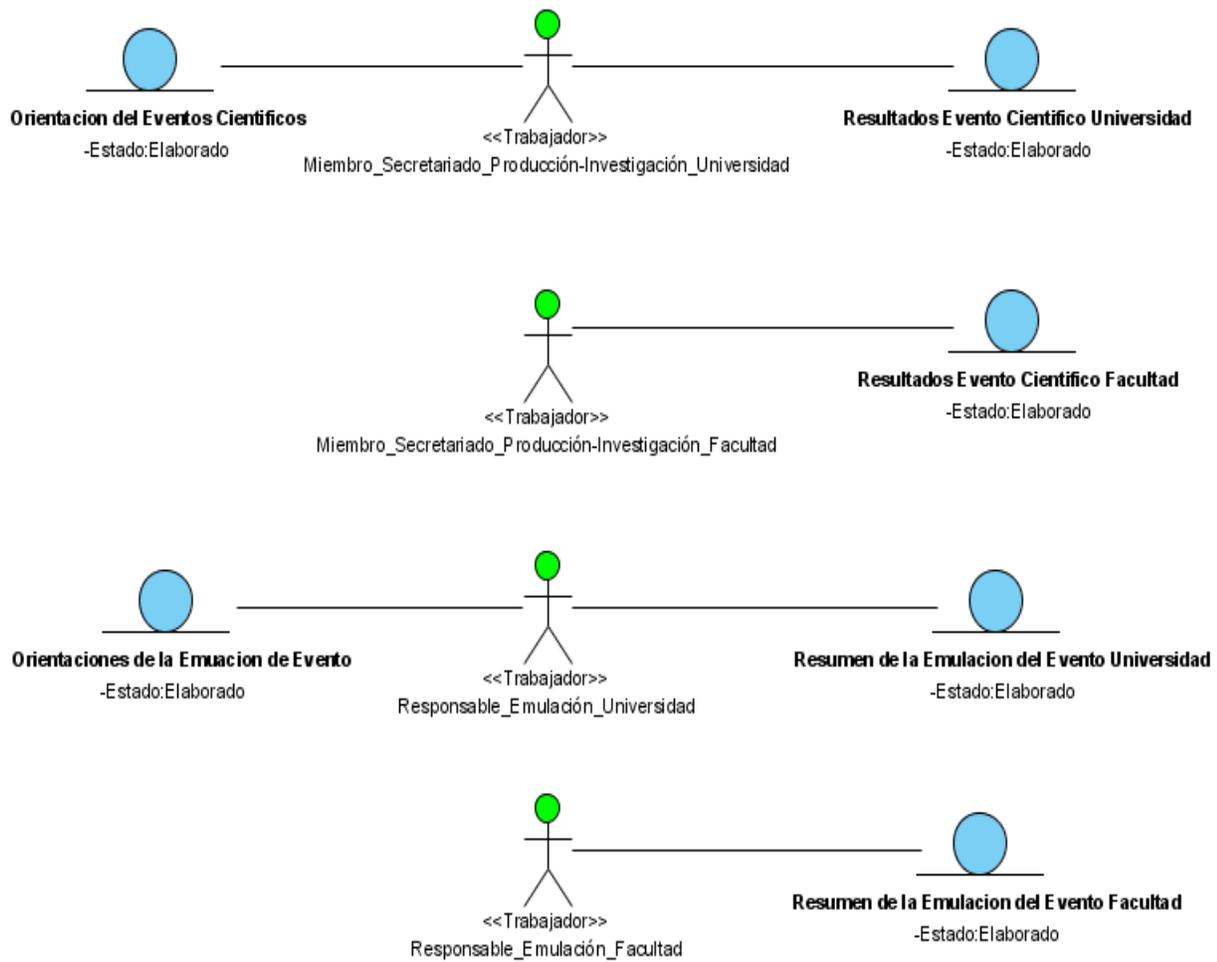


Figura 5 Diagrama de Clases del Modelo de Objetos.

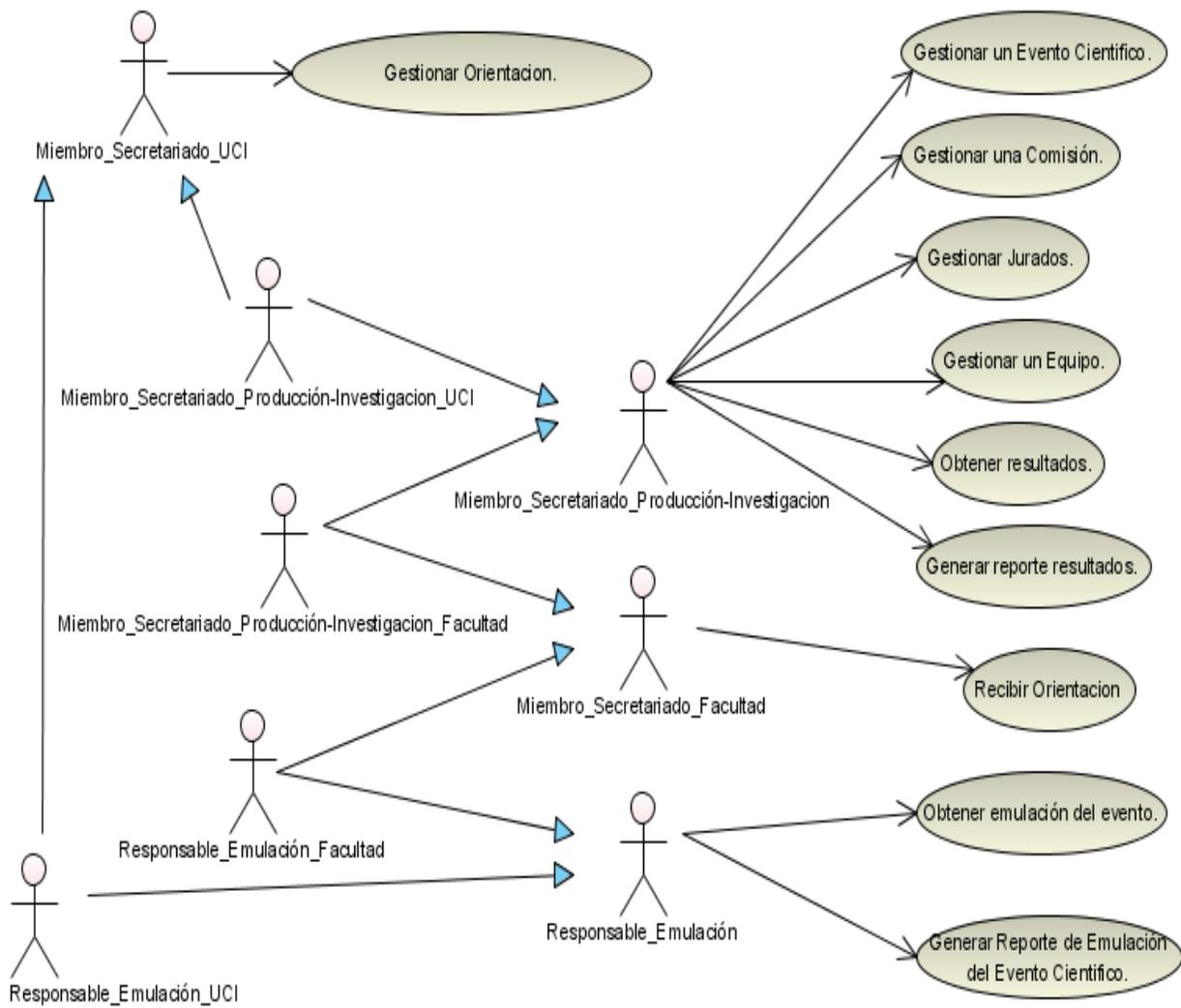


Figura 6 Diagrama de Casos de Uso del Sistema.

Diagrama de Clases de Análisis.

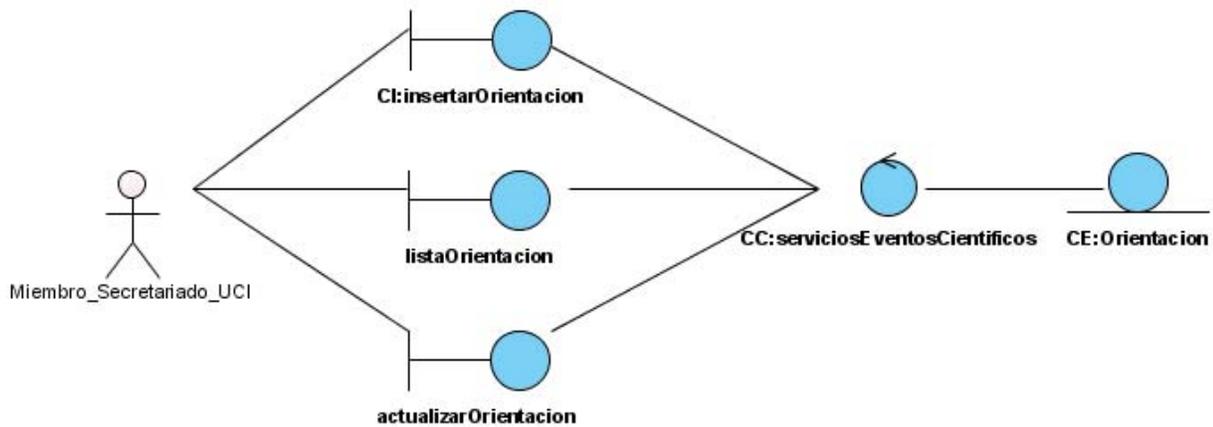


Figura 7 Diagrama de Clases de Análisis del CUS Gestionar Orientación del Evento.

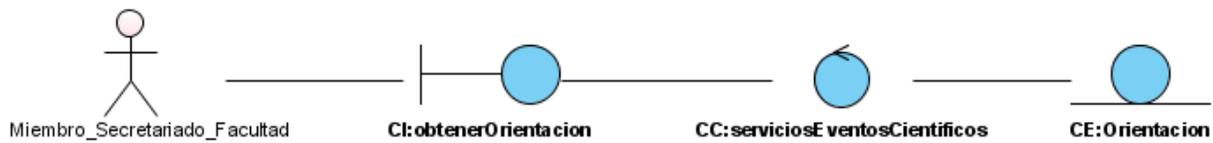


Figura 8 Diagrama de Clases de Análisis del CUS Recibir Orientación.

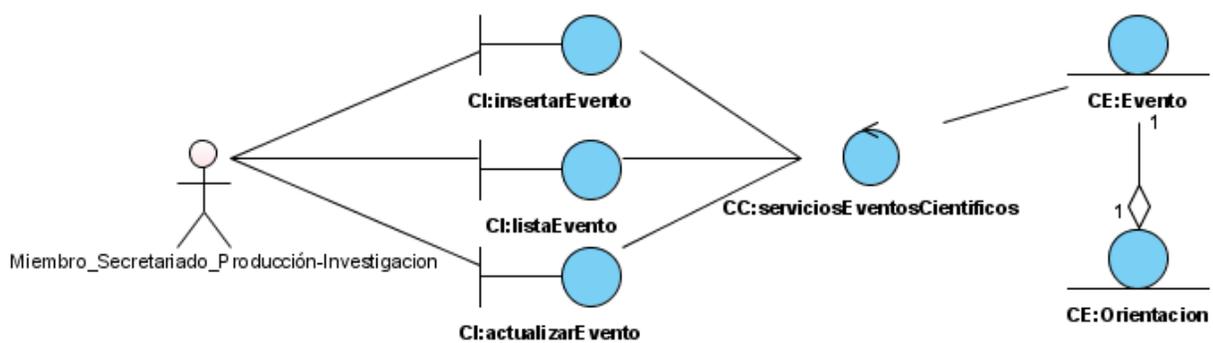


Figura 9 Diagrama de Clases de Análisis del CUS Gestionar Evento.

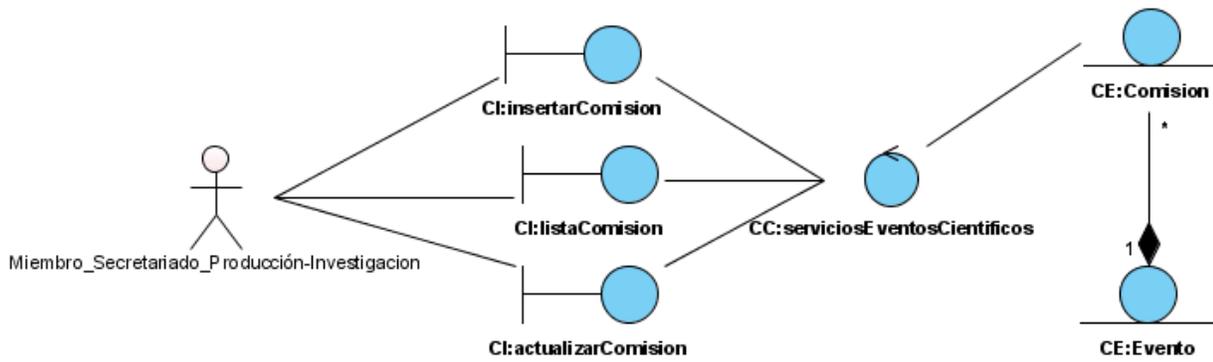


Figura 10 Diagrama de Clases de Análisis del CUS Gestionar Comisión.

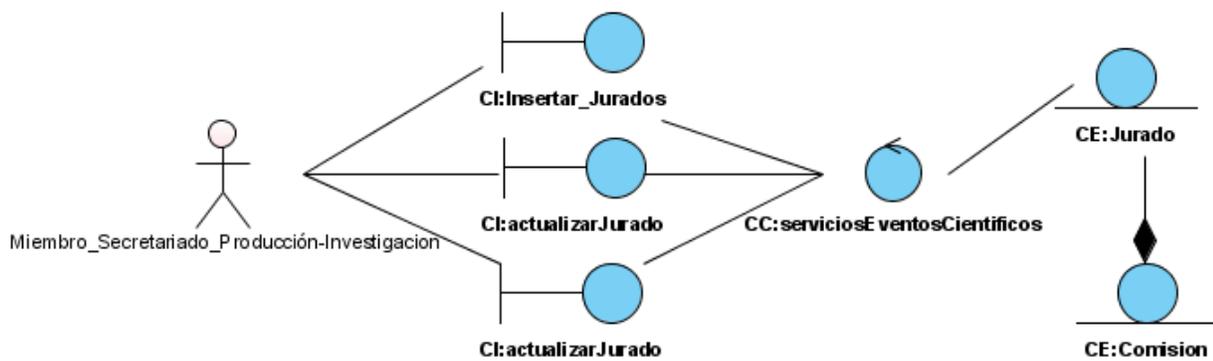


Figura 11 Diagrama de Clases de Análisis del CUS Gestionar Jurado

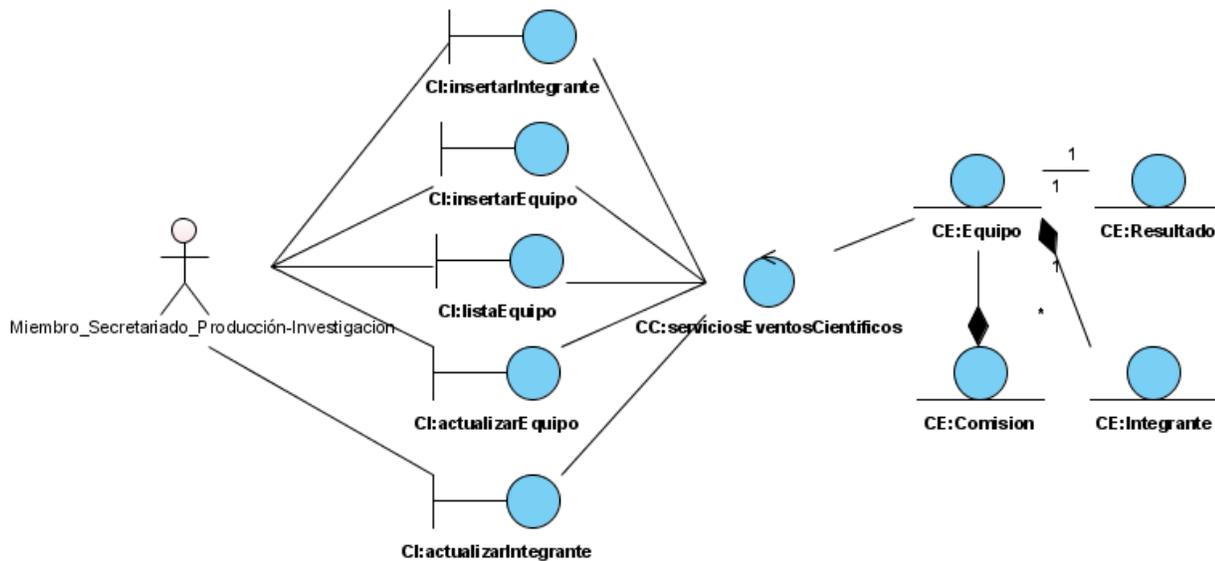


Figura 12 Diagrama de Clases de Análisis del CUS Gestionar Equipo.

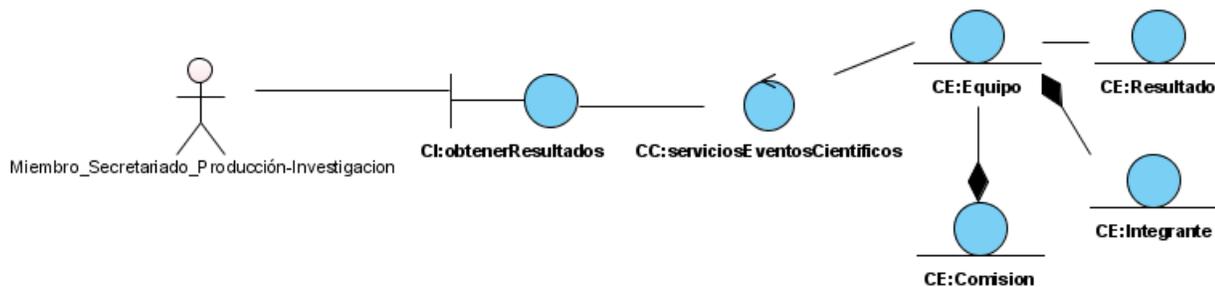


Figura 13 Diagrama de Clases de Análisis del CUS Obtener Resultados del Evento

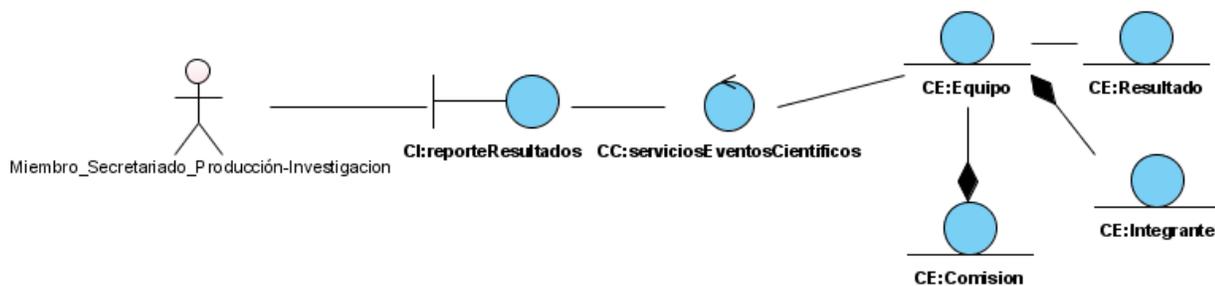


Figura 14 Diagrama de Clases de Análisis del CUS Mostrar Reporte de Resultados del Evento

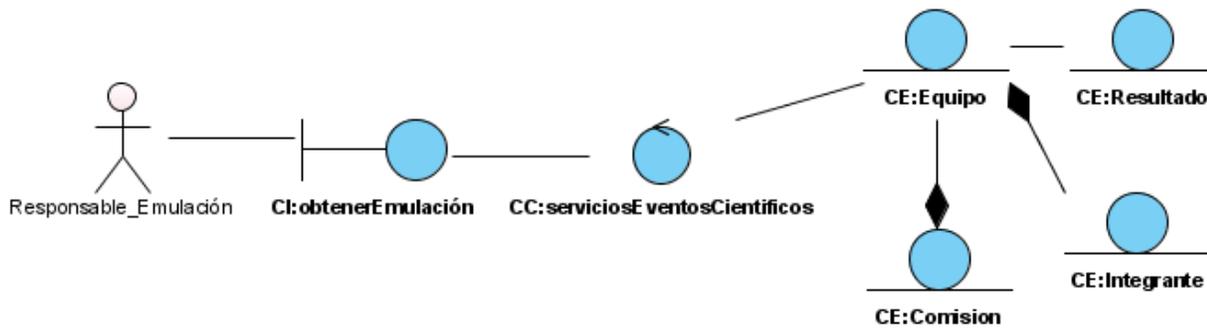


Figura 15 Diagrama de Clases de Análisis del CUS Obtener Emulación del Evento.

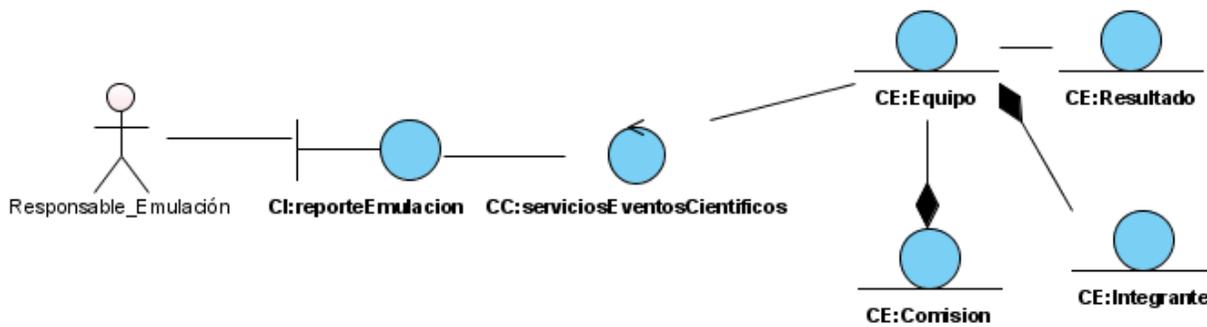


Figura 16 Diagrama de Clases de Análisis del CUS Mostrar Reporte de Emulación del Evento.

Diagrama de Interacciones

Diagramas de Interacción del Gestionar Orientación.

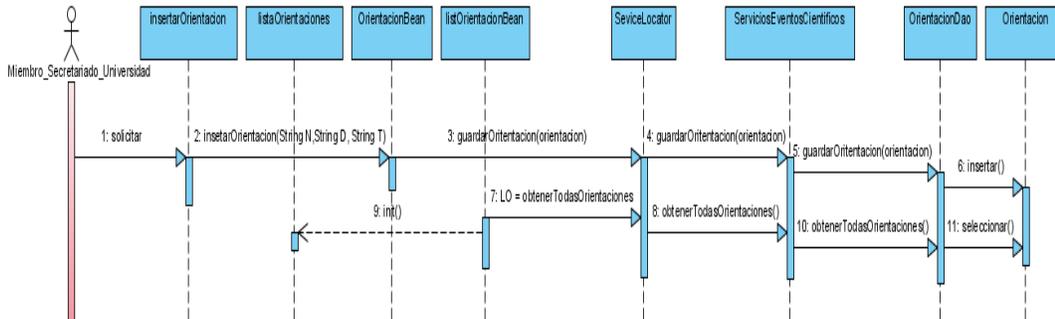


Figura 17 Diagrama de Secuencia de Insertar Orientación.

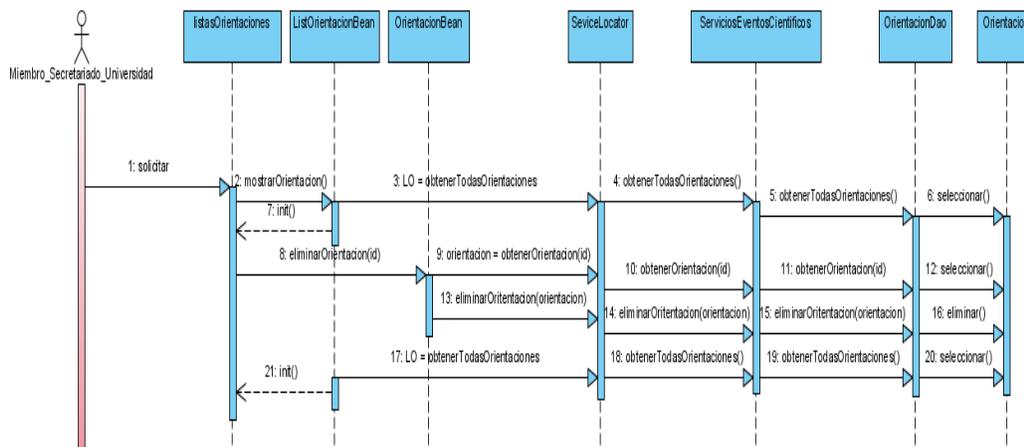


Figura 18 Diagrama de Secuencia de Eliminar Orientación.

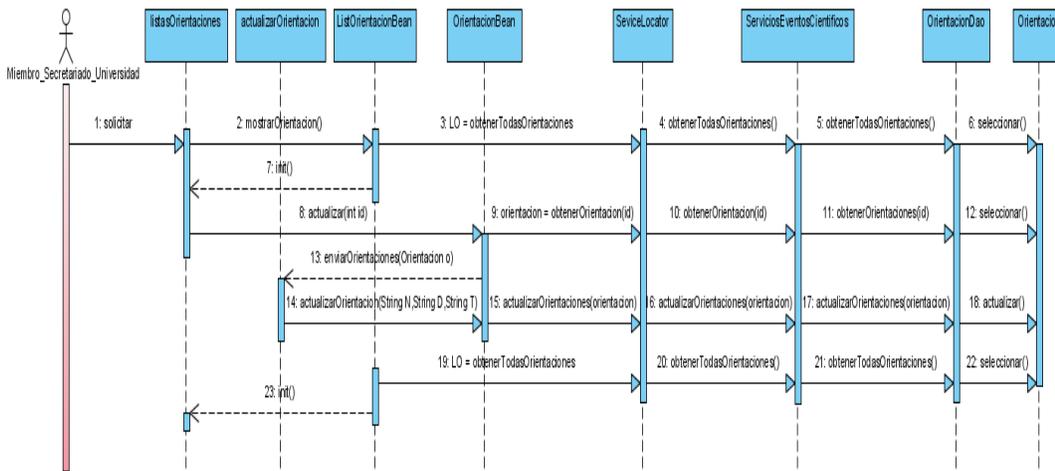


Figura 19 Diagrama de Secuencia de Actualizar Orientación.

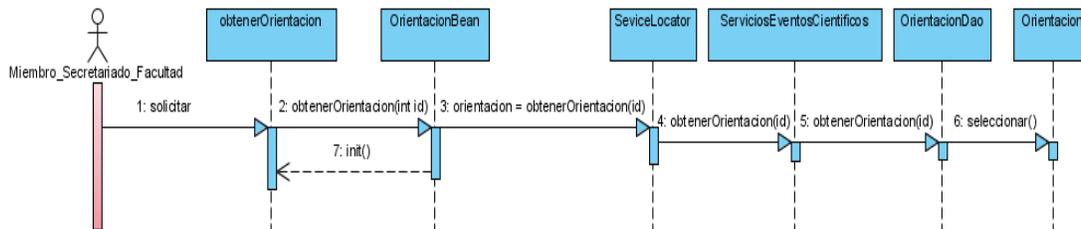


Figura 20 Diagrama de Interacción del Obtener Orientación.

Diagramas de Interacción del Gestionar un Evento.

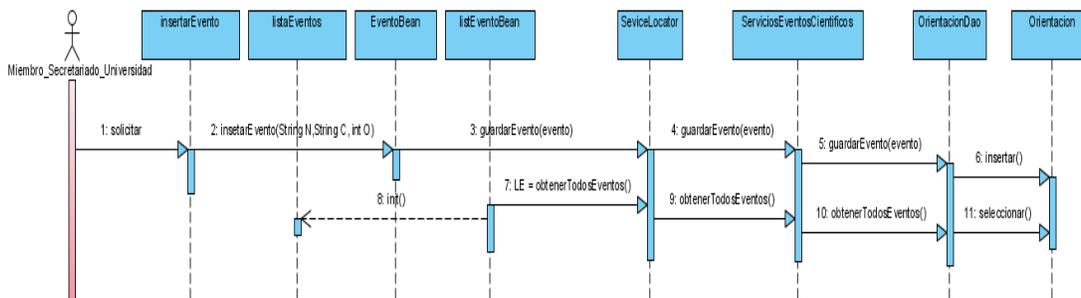


Figura 21 Diagrama de Secuencia de Insertar Evento.

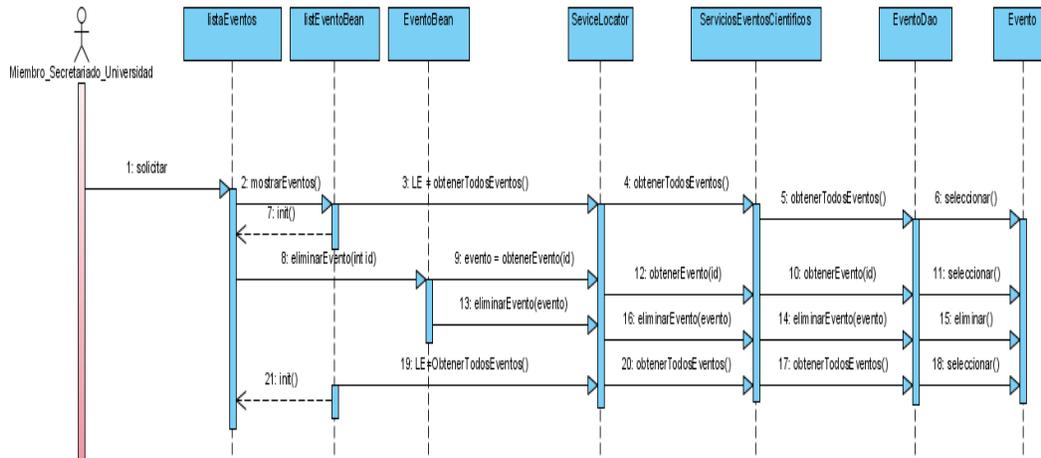


Figura 22 Diagrama de Secuencia de Eliminar Evento.

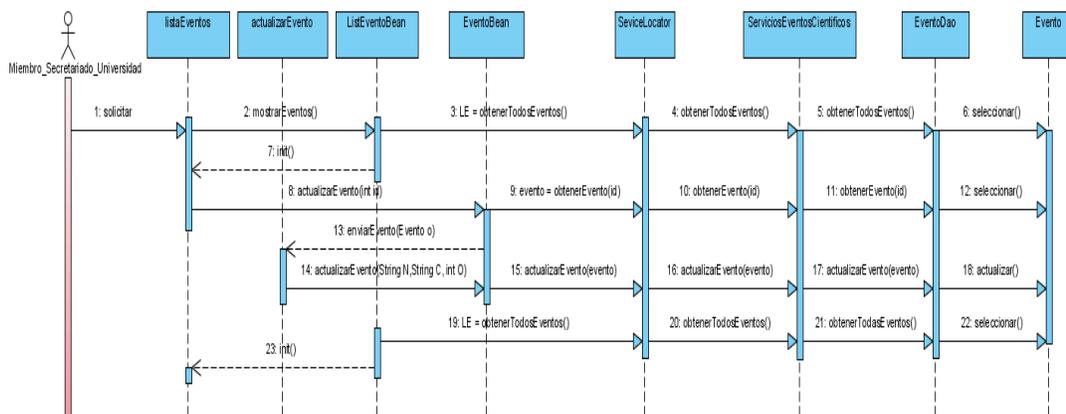


Figura 23 Diagrama de Secuencia de Actualizar Evento.

Diagramas de Interacción del Gestionar un Comisión.

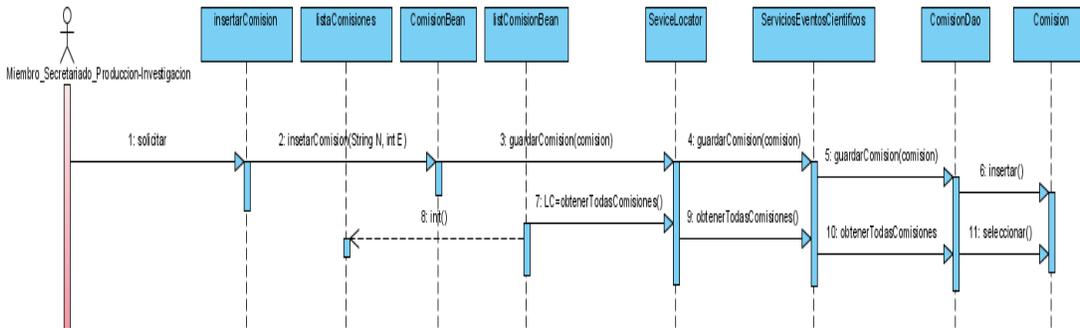


Figura 24 Diagrama de Secuencia de Insertar Comisión.

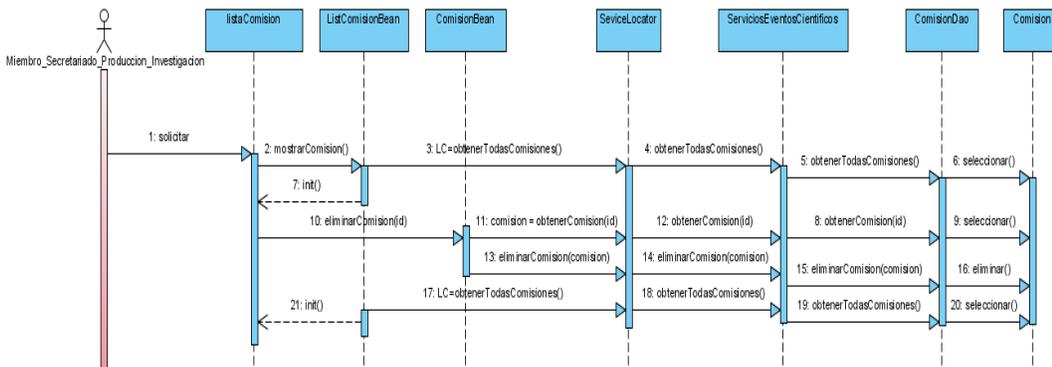


Figura 25 Diagrama de Secuencia de Eliminar Comisión.

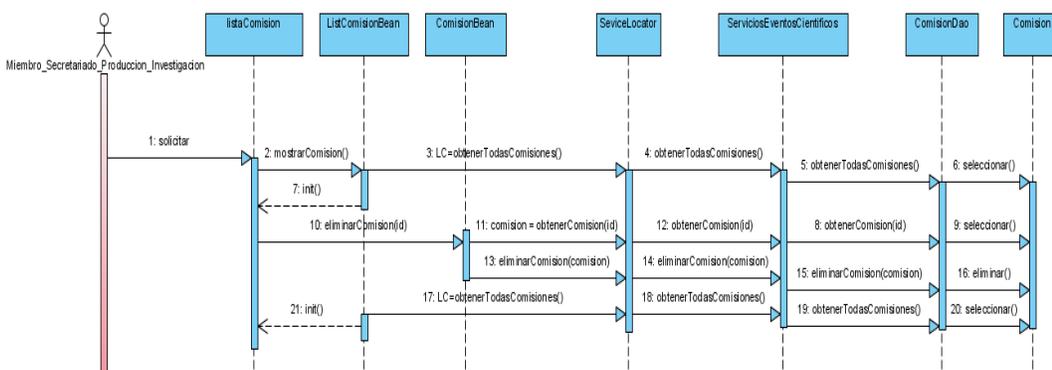


Figura 26 Diagrama de Secuencia de Actualizar Comisión.

Diagramas de Interacción del Gestionar Jurado.

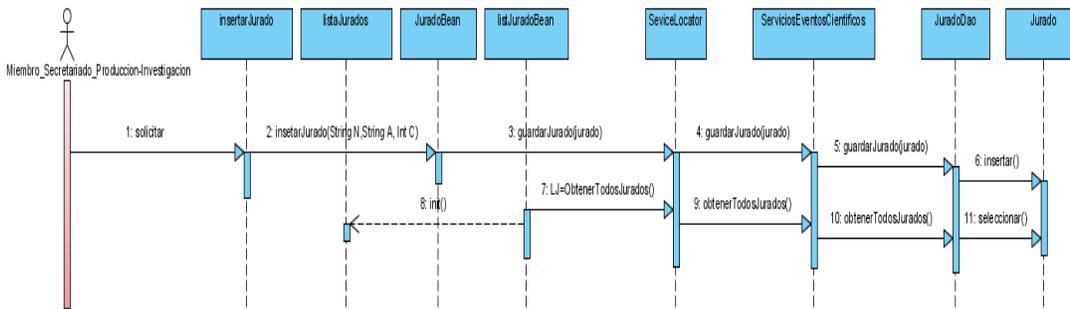


Figura 27 Diagrama de Secuencia de Insertar

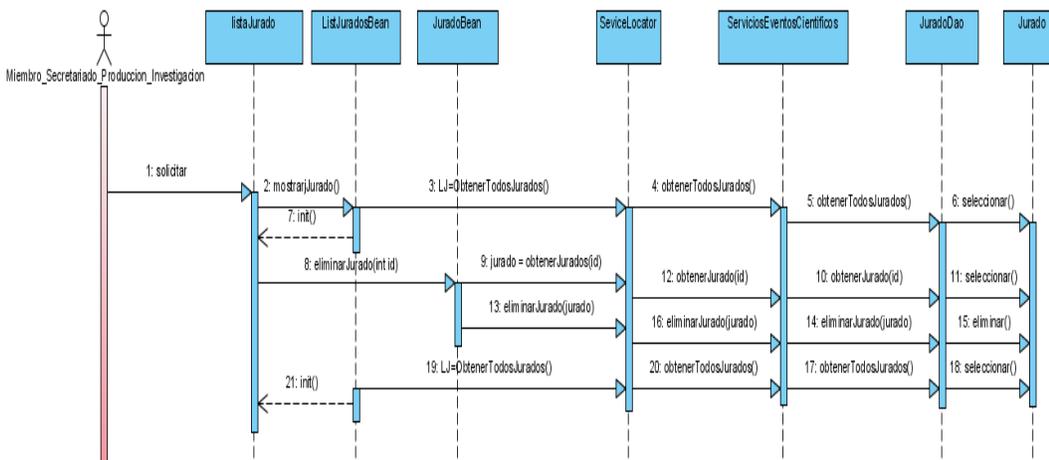


Figura 28 Diagrama de Secuencia de Eliminar

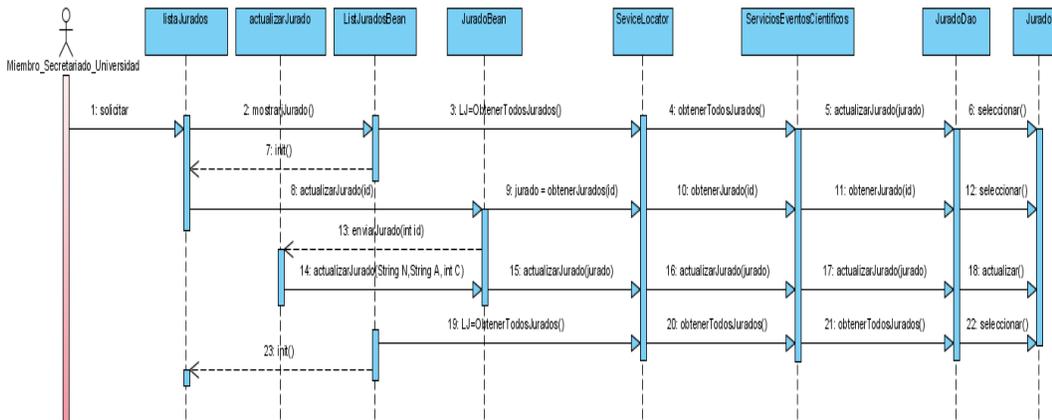


Figura 29 Diagrama de Secuencia de Actualizar

Diagrama de Interacción del Gestionar Equipo.

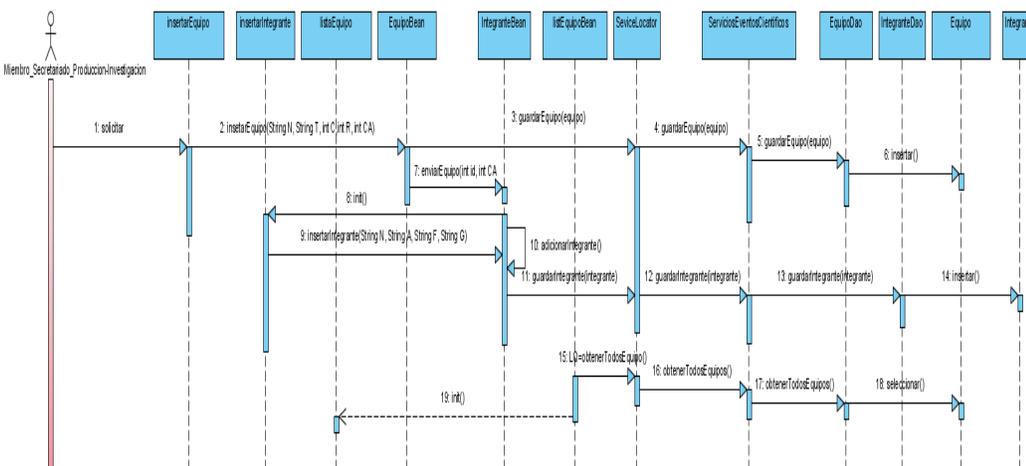


Figura 30 Diagrama de Secuencia de Insertar Equipo.

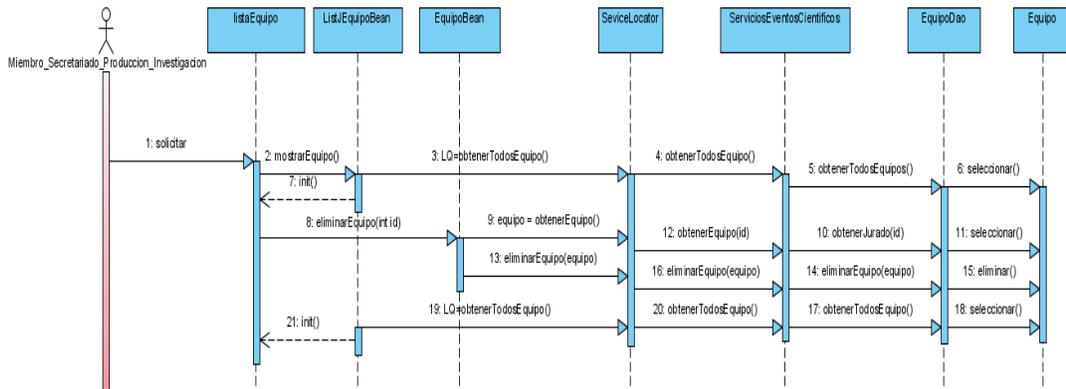


Figura 31 Diagrama de Secuencia de Eliminar Equipo.

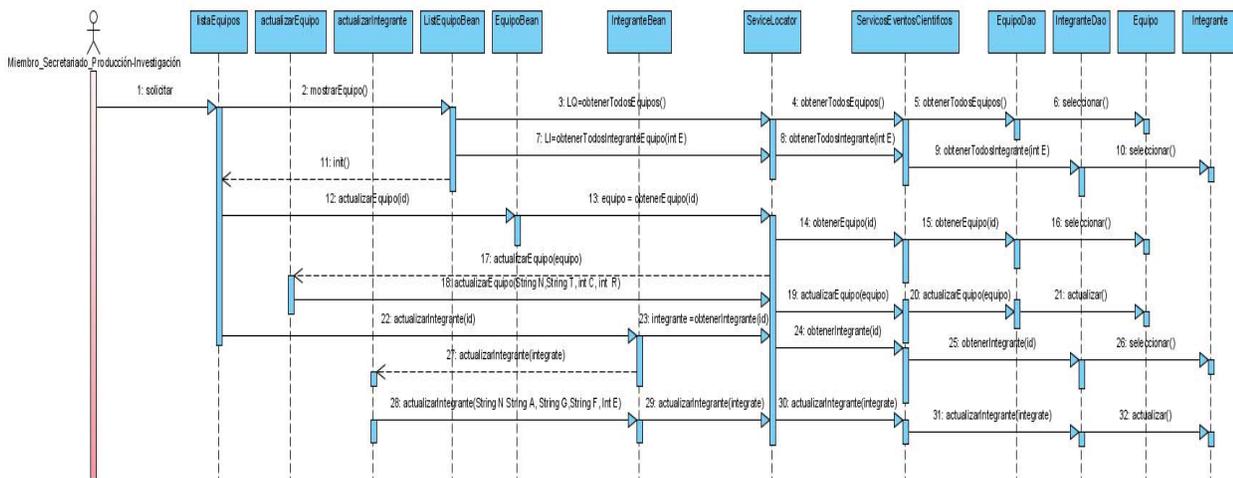


Figura 32 Diagrama de Secuencia de Actualizar Equipo.

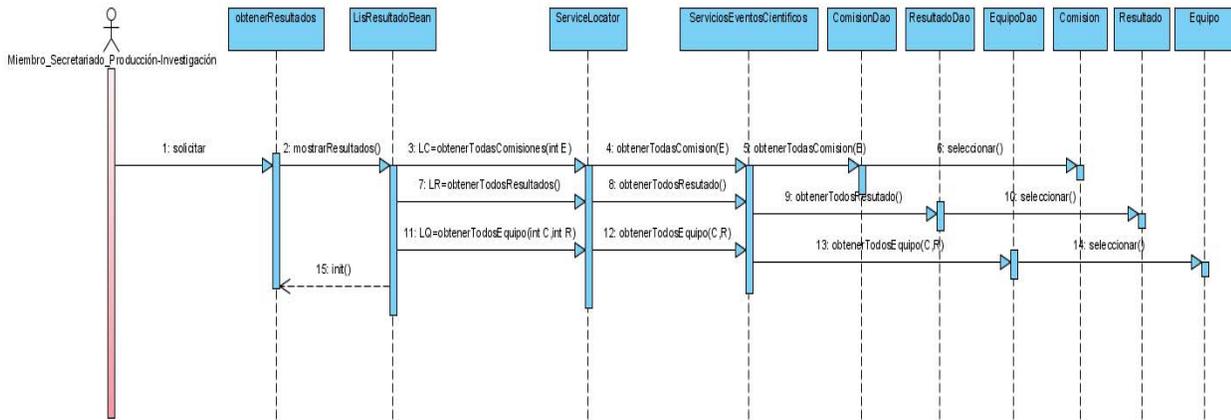


Figura 33 Diagrama de Interacción del Obtener Resultado del Evento Científico.

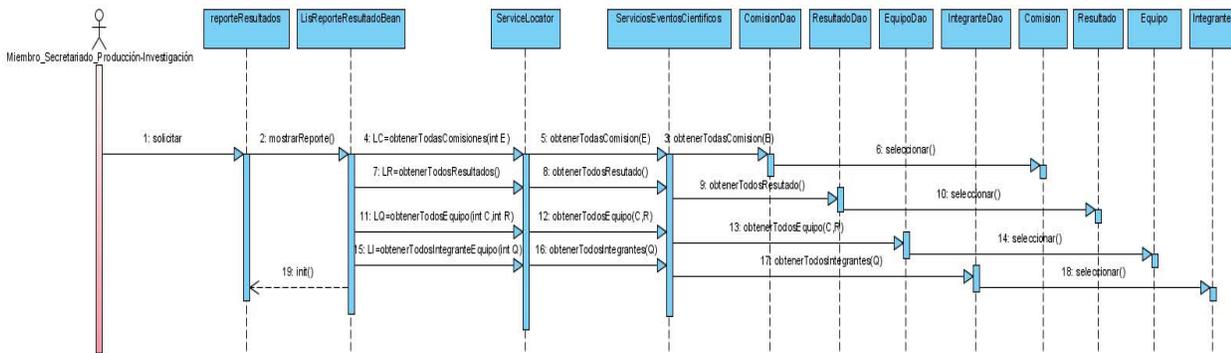


Figura 34 Diagrama de Interacción del Mostrar Reporte de Resultado del Evento Científico.

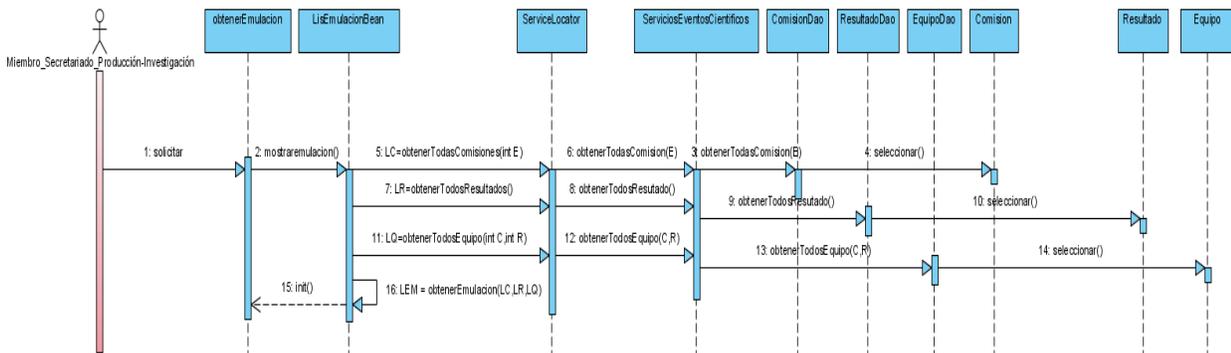


Figura 35 Diagrama de Interacción del Obtener Emulación del Evento Científico.

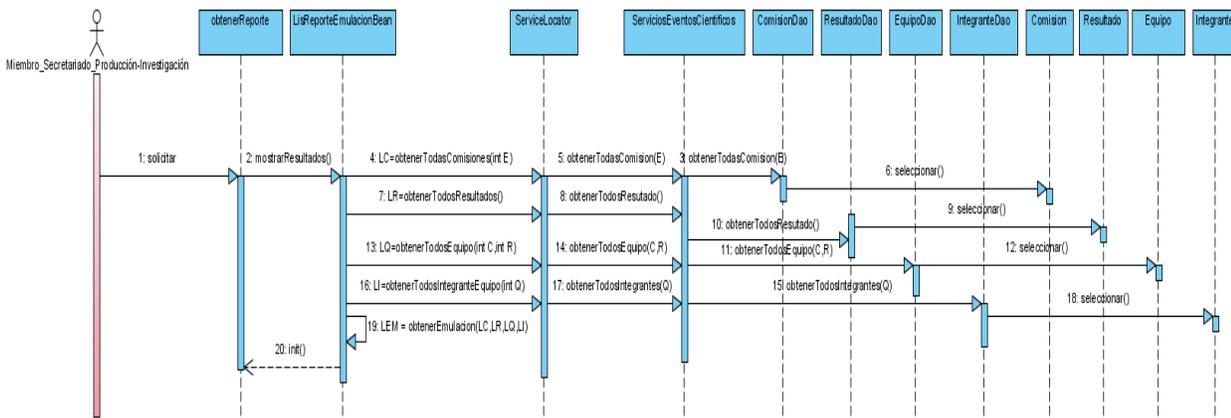


Figura 36 Diagrama de Interacción del Mostrar Reporte de Emulación del Evento Científico.

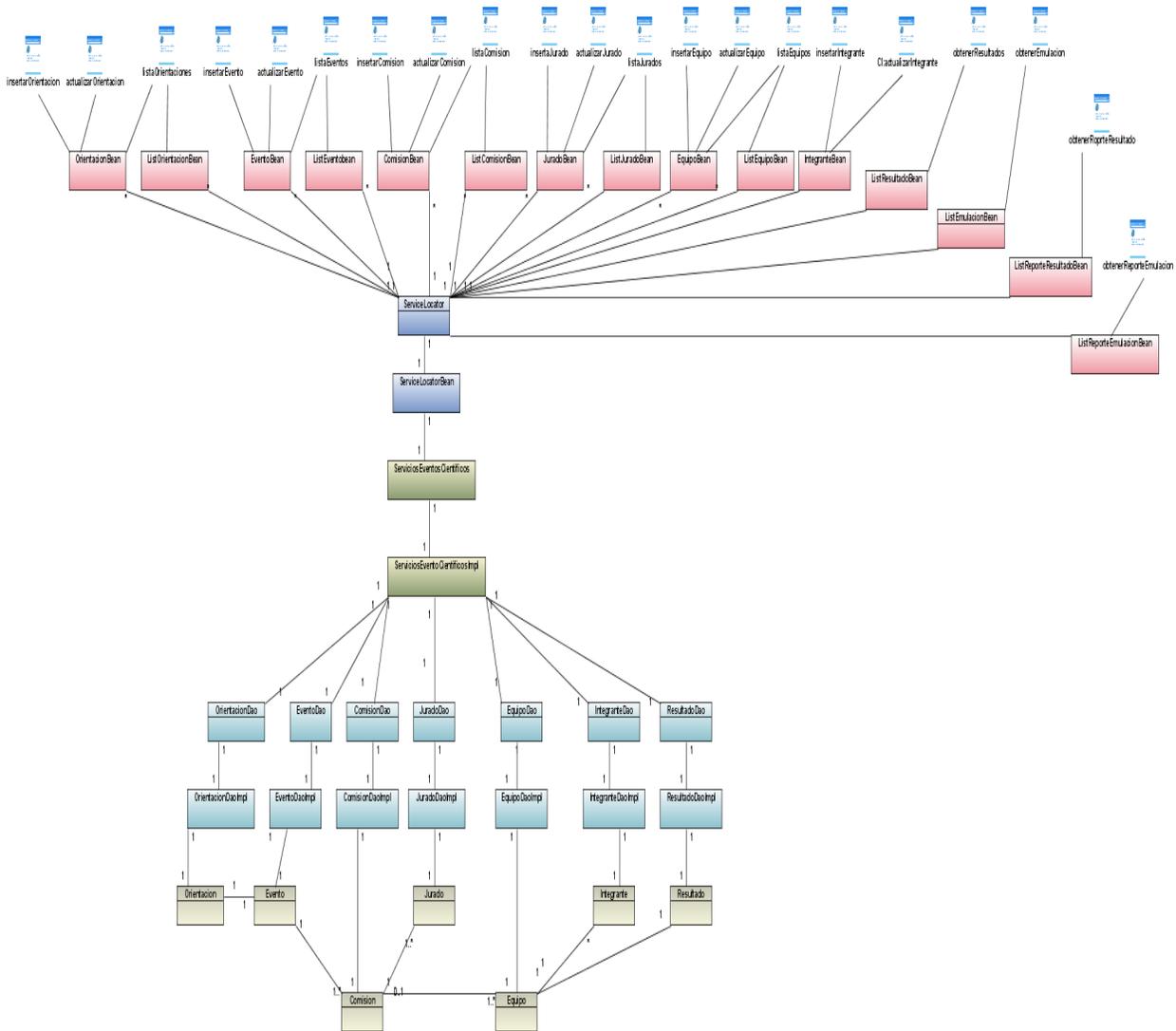


Figura 37 Diagrama de Clases del Diseño.

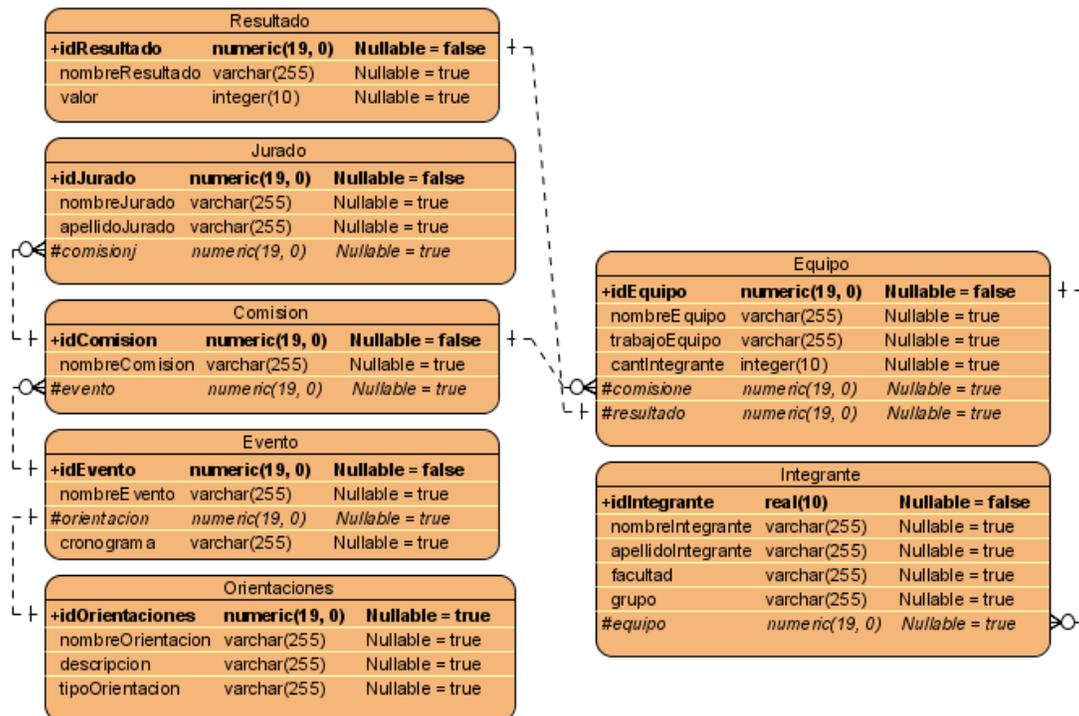


Figura 38 Diagrama de Entidad Relación.

Glosario de Términos.

API: Una API (Del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones, interfaz de programación de la aplicación) es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación

BD: Data Base o Base de datos.

Cliente-Servidor: Cuando se menciona este término no se piensa en otra cosa más que en bases de datos, dado que generalmente (Y de manera incorrecta) este término se usa como sinónimo de esto. Este término, en su más amplia definición, se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como Proceso Cooperativo, dado que dos procesos separados cooperan para completar la tarea en particular.

Componentes UI: Componentes de interfaz de usuario.

CU: Caso de uso.

DAO: Clases de acceso a datos (DAO Data Access Object)

El e-Government: , e-gobierno o gobierno electrónico consiste en el uso de las tecnologías de la información y el conocimiento en los procesos internos de gobierno y en la entrega de los productos y servicios del Estado tanto a los ciudadanos como a la industria. Muchas de las tecnologías involucradas y sus implementaciones son las mismas o similares a aquellas correspondientes al sector privado del comercio electrónico (o e-business), mientras que otras son específicas o únicas en relación a las necesidades del gobierno.

FEU: Federación estudiantil universitaria, organización de masas que representa a todos los universitarios cubanos.

Hibernate: es el puente entre nuestra aplicación y la BD, sus funciones van desde la ejecución de sentencias SQL hasta la creación, modificación y eliminación de objetos persistentes.

IDE: Entorno integrado de desarrollo (IDE siglas en inglés).

Información: Agregación de datos que tiene un significado específico más allá de cada uno de éstos. Un ejemplo: 1, 9, 8 y 7 son datos; 1987 es una información. La información ha sido

siempre un recurso muy valioso, revalorizado hoy más aun por el desarrollo y la expansión de las Tecnologías de la Información y de las Comunicaciones.

Interface: Conexión entre dos dispositivos de hardware, entre dos aplicaciones o entre un usuario y una aplicación que facilita el intercambio de datos.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

J2EE: Java 2 Enterprise Edition, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Java.

Java: es un lenguaje orientado a objetos y desarrollado por Sun Microsystems. Comparte similitudes con C, C++ y Objective C. Basándose en otros lenguajes orientados al objeto, Java recoge lo mejor de todos ellos y elimina sus puntos más conflictivos. El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet (aunque el código fuera escrito de forma maliciosa). Esta característica requiere la eliminación de muchas construcciones y usos de C y C++. El más importante, es que no existen punteros. Java no puede acceder arbitrariamente a direcciones de memoria. Java es un lenguaje compilado en un código llamado "codigo-byte" (byte-code). Este código es interpretado por el intérprete Java, el cual Java fue diseñado también para escribir código libre de bugs, esto se consigue en gran parte, eliminando las operaciones de localización y deslocalización de memoria del lenguaje C. Java no es un lenguaje para ser usado solo en el WWW, pero su despegue y utilización se debe al World Wide Web. Hoy día casi todos los exploradores interpretan código Java.

JSF (Java Server Faces): es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador).

JSP o Java Server Pages: es una tecnología Java que permite a los programadores generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento web. En las JSP se escribe el texto que va a ser devuelto en la salida (normalmente, código HTML) incluyendo código java dentro de él, para poder modificar o generar contenido dinámicamente.

MER: Modelo Entidad relación.

PostgreSQL: Motor de base de datos, servidor de base de datos relacional libre, liberado bajo una licencia de software libre con grandes ventajas y funcionalidades de mucho uso en el mundo actual de las ciencias informáticas.

Proceso: Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

Red: Una red es dos o más computadoras que están físicamente conectadas con las otras y capaces de compartir información.

R-N: Referencia al caso de uso N.

Scripts: En inglés significa "guión". De hecho, el uso es exactamente éste: el navegador lee una línea, la interpreta y la ejecuta, después pasa a la sucesiva y hace lo mismo, y así hasta el cierre del script.

SGBD: Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista a la misma vez.

Sistema: Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto.

Sitio Web: Sistema de computación que corre un servidor Web y que se ha establecido para editar documentos en Web.

Software:(Componentes lógicos, programas, software). -- Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

Spring: es un marco de trabajo para aplicaciones J2EE que pretende reducir el esfuerzo de desarrollo para la construcción de aplicaciones.

Subversión: es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se lo conoce también como svn por ser ese el nombre de la herramienta de línea de comandos.

Tortoise: Es el cliente más recomendable para trabajar con el Subversion Tortoise svn. Este cliente convierte el mismo Explorer en un cliente Subversion, es realmente rápido y cómodo de usar.

UML y UML 2.0: El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas

notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. La diferencia mas notable entre UML y UML 2.0 es que el primero es orientado a objetos y el segundo es orientada a procesos, lo cual muestra su principal ventaja en que en el mundo real la mayoría de las cosas son procesos.

URL: (Uniform Resource Locator) -- (Localizador Uniforme de Recursos) Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos WWW, Gopher, FTP, News,... Ejemplos de URL son: <http://cubasi.cu> o <ftp://ftp.ati.es>.

WEB (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.