



**Facultad CITEC**

# **Sistema para la gestión de eventos estudiantiles en la Universidad de las Ciencias Informáticas**

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

**Autores:**

Alejandro Miguel Medina Pichs

María Teresa Sardañas Rodríguez

**Tutor:**

Ing. Víctor Gabriel González Cardoso

La Habana, junio de 2017  
Año 59 de la Revolución

Declaración Jurada de Autoría:

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año 2017.

\_\_\_\_\_  
Alejandro Miguel Medina Pichs

\_\_\_\_\_  
María Teresa Sardañas Rodríguez

\_\_\_\_\_  
Ing. Victor Gabriel González Cardoso

## **AGRADECIMIENTOS:**

A Dios, por ayudarnos en todo momento y por darnos la oportunidad de estudiar en esta universidad. Gracias le damos por su fidelidad y su amor, que nunca faltaron en nuestros años de estudio.

A nuestros padres: Raúl, Camilo, Javier, Aimeé y Tatiana.

A nuestros hermanos: Ana Beatriz, Ana Laura, Rosalia, Álvaro y Samir.

A nuestros familiares: Azucena, Juanito, Yosbel, Yordany, Grisel, Ayán, Marito, Liuda, Idania, Idanelis, Joe, Agustina, Yusimí, Camilo, Yolanda, Tony, Yadira, Yuraisi, Dailyn, Arais, Xiomara.

A nuestros padres, hermanos y familiares agradecemos porque todos, de una u otra manera, se han preocupado y también ocupado de nosotros, se han esforzado y sacrificado muchísimo para que hoy este sueño se hiciera realidad.

Agradecemos también a nuestros hermanos en la fe, ha sido una bendición conocerlos y compartir todos estos años juntos; sin ellos nuestra estancia en la universidad no hubiera sido la misma. Hoy podemos decir que gracias a ellos nos mantuvimos creciendo en los caminos de Dios, y el amor que nos demostraron a diario fue una gran motivación para seguir adelante. Gracias por sus oraciones.

A nuestras amistades: Carlos Jesús, Yadrían, Aldito, César, Luis, Rafael, Enmanuel, Lilibeth, Gisell, Diana, Jose, Luis David, Roberto, Daniel, Ronny, Nelson, Zaida, Katy, Yerandy, Marquitos, Eliecer, Ángel, Yaneisi, Flavia, Darayne, Yazmín, Víctor, Yuli, Mariela, Handy, Josué.

A nuestro tutor Víctor, porque nos ha tratado siempre con mucha dedicación y nos brindó su más sincera amistad; también nos ayudó muchísimo en el desarrollo de nuestro trabajo de diploma y se preocupó por nosotros en todas las esferas.

A nuestra oponente, porque su buen trabajo contribuyó en gran manera a que pudiéramos realizar nuestra tesis con excelencia.

Al presidente del tribunal Rafael, porque desde el principio de este semestre no ha dejado de animarnos y alentarnos, haciéndonos ver este proceso de tesis desde un punto de vista muy positivo y optimista.

Al trabajador del nodo central que por humildad no nos dio su nombre, pero nos ayudó en el momento preciso desinteresadamente.

A nuestros profesores y a todos los que contribuyeron en nuestra formación como ingenieros.

“Muchas gracias”.

DEDICATORIA:

*A Dios, a nuestros padres y hermanos.*

**RESUMEN:**

En la Universidad de las Ciencias Informáticas existe una gran dinámica en cuanto a los eventos estudiantiles que en ella se realizan. La participación de los estudiantes en dichos eventos da constancia de su formación integral y tributa a su caracterización y evaluación integral, proceso que se lleva a cabo en la universidad y que constituye la herramienta clave para el seguimiento de la formación como profesional de los estudiantes que en ella cursan. La gestión de los eventos estudiantiles en la universidad presenta dificultades en cuanto a la accesibilidad, estandarización de la información y persistencia de los datos, lo que dificulta el proceso de caracterización y evaluación integral. La presente investigación propone el desarrollo de una herramienta informática para gestionar la información asociada a los eventos estudiantiles en la Universidad de las Ciencias Informáticas. Además, se describen los métodos y técnicas para la realización de las pruebas con el propósito de validar la propuesta de solución.

**Palabras claves:** evaluación integral, evento estudiantil, sistema de gestión.

**ABSTRACT:**

The University of Informatics Science hosts a lot of student events every year. The participation of a student in these events gives evidence of his integral education and contributes to the characterization process. This process constitutes the key tool for the follow-up of the integral education of the students. The management of student events in the university presents difficulties in terms of accessibility, standardization of information and persistence of data, which makes the characterization and evaluation process more difficult. This research proposes the development of a computer tool to manage information related to student events at the University of Informatics Science. Finally, the methods and techniques for testing are described in order to validate the proposed solution.

**Keywords:** Integral evaluation, management system, student event.

# ÍNDICE

Introducción .....	1
Capítulo 1: Fundamentación teórica .....	5
1.1    Introducción .....	5
1.2    Conceptos asociados al dominio del problema .....	5
1.2.1    Sistema de gestión .....	5
1.2.2    Información.....	5
1.2.3    Sistema de gestión de información .....	5
1.2.4    Sistema Informático .....	6
1.2.5    Evento estudiantil .....	6
1.3    Descripción de los procesos asociados a los eventos estudiantiles .....	6
1.4    Estudio del estado del arte.....	7
1.5    Ecosistema de aplicaciones MAYA.....	9
1.6    Herramientas y tecnologías .....	10
1.6.1    Entorno de Desarrollo Integrado .....	10
1.6.2    Herramienta CASE .....	10
1.6.3    Sistema de control de versiones .....	11
1.6.4    Servidor de aplicaciones de prueba .....	11
1.6.5    Herramientas para la realización de las pruebas .....	11
1.6.6    Sistema gestor de bases de datos .....	11
1.6.7    Lenguaje de programación .....	12
1.6.8    Framework de desarrollo .....	12
1.7    Metodología de desarrollo de software .....	13
Conclusiones parciales.....	16
Capítulo 2: Descripción de la solución propuesta.....	17
2.1    Introducción .....	17
2.2    Descripción de los procesos de negocio asociados a la organización de los eventos estudiantiles 17	
2.3    Lista de reserva del producto.....	18
2.3.1    Requisitos funcionales.....	18
2.3.2    Requisitos no funcionales .....	24
2.4    Historias de usuario .....	26
2.5    Plan de entrega .....	28
2.6    Estimación y planificación.....	28
2.7    Tareas de ingeniería .....	29
	VI

2.8	Arquitectura del Sistema .....	31
2.8.1	Patrón Modelo Vista Controlador .....	31
2.8.2	Módulos del Sistema .....	33
2.9	Tarjetas CRC .....	34
2.10	Personas relacionadas con el sistema .....	36
2.11	Patrones de Diseño .....	37
	Conclusiones parciales.....	39
Capítulo 3: Implementación y validación del sistema .....		40
3.1	Introducción .....	40
3.2	Modelo de datos físicos .....	40
3.3	Integración con otros sistemas .....	41
3.4	Pruebas de Software .....	44
3.4.1	Implementación de casos de pruebas.....	46
3.4.2	Casos de Pruebas de Aceptación.....	52
3.5	Implantación del sistema .....	53
3.6	Validación de la solución .....	55
	Conclusiones parciales.....	57
Conclusiones generales.....		58
Recomendaciones .....		59
Referencias bibliográficas .....		60
Anexos.....		64
	Anexo 2: Planes de entrega .....	64
	Anexo 3: Estimación y planificación.....	65
	Anexo 6: Modelo de base de datos .....	66
	Anexo 8: Carta del consejo FEU .....	70

## ÍNDICE DE FIGURAS

Figura 1 Patrón MVC .....	32
Figura 2 Módulos del sistema. ....	33
Figura 3 Modelo de datos físicos: fragmento dentro del módulo eventos científicos. ....	40
Figura 4 Integración con MAYA. ....	41
Figura 5 Página de inicio.....	54
Figura 6 Vista del perfil del estudiante. ....	54
Figura 7 Vista de detalles del evento. ....	55

## ÍNDICE DE GRÁFICAS

Gráfica 1 Volumen de actividad en el 2do semestre del curso 2015-2016. ....	48
Gráfica 2 Volumen de actividad de 11:00 AM a 3:59 PM el 22 de febrero.....	49
Gráfica 3 Tiempo de respuesta medio y máximo. ....	50
Gráfica 4 Tiempo de respuesta medio por grupo de peticiones.....	51
Gráfica 5 Tiempo de respuesta máximo por grupo de peticiones.....	51

## ÍNDICE DE TABLAS

Tabla 1 Requisitos funcionales asociados al Módulo de administración y seguridad. ....	18
Tabla 2 Requisitos funcionales asociados al Módulo Común. ....	19
Tabla 3 Requisitos funcionales asociados al Módulo de eventos científicos. ....	20
Tabla 4 Requisitos funcionales asociados al Módulo de eventos deportivos. ....	22
Tabla 5 Requisitos funcionales asociados al Módulo de eventos culturales. ....	23
Tabla 6 Requisitos funcionales asociados al Módulo de copas académicas. ....	24
Tabla 7 Historia de Usuario Gestionar evento. ....	26
Tabla 8 Historia de Usuario Asignar trabajo a comisión. ....	27
Tabla 9 Historia de Usuario Subir trabajo de nivel. ....	27
Tabla 10 Historia de Usuario Finalizar evento. ....	27
Tabla 11 Plan de entrega para el Módulo de eventos científicos. ....	28
Tabla 12 Estimación y planificación para la primera entrega. ....	28
Tabla 13 Tarea de Ingeniería No. 1.1 para HU "Gestionar evento". ....	29
Tabla 14 Tarea de Ingeniería No. 1.2 para HU "Gestionar evento". ....	30
Tabla 15 Tarea de Ingeniería No. 2.1 para HU "Asignar trabajo a comisión". ....	30
Tabla 16 Tarea de Ingeniería No. 2.2 para HU "Asignar trabajo a comisión". ....	30
Tabla 17 Tarjeta CRC Clase: Evento. ....	35
Tabla 18 Tarjeta CRC Clase: Comisión. ....	35
Tabla 19 Tarjeta CRC Clase: Trabajo. ....	35
Tabla 20 Tarjeta CRC Clase: Tribunal. ....	36
Tabla 21 Personas relacionadas con el sistema. ....	36
Tabla 22 Servicios utilizados por Eventos de ALMA, Sistema de Gestión de datos primarios. ....	42
Tabla 23 Servicios brindados por Eventos. ....	44
Tabla 24 Caso de Prueba de Aceptación HU1_P1 para la HU "Gestionar evento". ....	52
Tabla 25 Caso de Prueba de Aceptación HU1_P2 para la HU "Gestionar evento". ....	52
Tabla 26 Caso de Prueba de Aceptación HU1_P3 para la HU "Gestionar evento". ....	52
Tabla 27 Caso de Prueba de Aceptación HU1_P4 para la HU "Gestionar evento". ....	53
Tabla 28 Caso de Prueba de Aceptación HU1_P5 para la HU "Gestionar evento". ....	53
Tabla 29 Plan de entrega del Módulo de eventos deportivos. ....	64
Tabla 30 Plan de entrega del Módulo eventos culturales. ....	64
Tabla 31 Plan de entrega del Módulo copas académicas. ....	64
Tabla 32 Estimación y planificación del Módulo de eventos deportivos. ....	65
Tabla 33 Estimación y planificación del Módulo de eventos culturales. ....	65
Tabla 34 Estimación y planificación del Módulo de copas académicas. ....	65

## INTRODUCCIÓN

El desarrollo avanzado de las nuevas Tecnologías de la Información y las Comunicaciones (TICs), ha provocado profundos cambios en el contexto social en el que se desenvuelve el hombre moderno. Como resultado directo de este proceso ha emergido el concepto de Sociedad de la Información, donde los conocimientos, su creación y propagación son el elemento definitorio de las relaciones entre los individuos y entre las naciones (Tejo Delarbre, 1996).

La alta disponibilidad y accesibilidad de la información en el mundo actual, ha abierto el espectro a un conjunto de aplicaciones orientadas al manejo de la enorme cantidad de datos que son generados a diario. Cuba no ha estado exenta de este fenómeno y muchos han sido los esfuerzos del gobierno encaminados a dotar al país de la infraestructura necesaria para la implementación de los procesos relacionados con la gestión y automatización de la información; con este fin se ha creado un plan de informatización que incluye a cada uno de los sectores de la población (Guevara, 2016). En la educación se ha incrementado la conectividad de las escuelas, el software educativo empleado en estas ha evolucionado de manera significativa y se dan pasos importantes para su uso progresivo e intensivo en las clases (Oceguera Martínez, y otros, 2009).

Las universidades cubanas constituyen un eje importante en la sociedad ya que son instituciones innovadoras de excelencia científica, académica y productiva que forma profesionales altamente calificados, con valores patrios y un profundo sentido humanista reflejado en la formación integral del profesional (MES, 2015). Dentro de las instituciones del Ministerio de Educación Superior (MES) se encuentra la Universidad de las Ciencias Informáticas (UCI), organización vanguardia en la rama de la informática cuya misión fundamental es formar profesionales comprometidos con su Patria y altamente calificados, además de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, y también servir de soporte a la industria cubana de la informática (UCI, 2012). En esta universidad se lleva a cabo un proceso de caracterización y evaluación de los estudiantes universitarios el cual consiste en la recopilación permanente de información significativa para la conformación de juicios de valor, con el fin de facilitar la comprensión de las situaciones del desarrollo de los educandos (López Rodríguez del Rey, y otros, 2012). Uno de los puntos fundamentales en el proceso de caracterización y evaluación integral es el registro de la participación de los estudiantes en los eventos estudiantiles.

En la UCI actualmente existe una gran dinámica en cuanto a los eventos estudiantiles, siendo lugar de acogida de eventos tales como el Fórum de Historia, el Seminario Juvenil Martiano, los Juegos Mella, la Peña Tecnológica, los Festivales de Artistas Aficionados, la Jornada del Ingeniero de las Ciencias Informáticas, MiWebXCuba, entre otros. La participación de los estudiantes en dichos eventos, da

constancia de su formación general integral y tributa a su caracterización y evaluación integral. Los eventos que se organizan en la universidad se agrupan y clasifican en cuatro grandes tipos, ellos son: científicos, culturales, deportivos y copas académicas. Un evento transita por las fases de: lanzamiento de la convocatoria, organización y realización del evento, evaluación y anuncio de los resultados.

El proceso de gestión de los eventos estudiantiles presenta los siguientes problemas: el gran volumen de información que se genera dificulta el proceso de recuperación de los datos; la existencia de diversas áreas que aportan evidencias a las participaciones de los estudiantes en los eventos, cada una de estas evidencias en formatos distintos con el uso de herramientas como Word, Excel o PDF<sup>1</sup>, provocando que la información no siga ningún tipo de estandarización; el procesamiento de la información se realiza generalmente de forma manual, esto trae como consecuencia que los responsables tengan que emplear mucho tiempo en esta actividad y que el procesamiento de la información esté sujeto a errores humanos; existe poca disponibilidad y accesibilidad de la información relacionada con los eventos estudiantiles, ya que no se cuenta con un archivo histórico, donde se pueda obtener la información de años precedentes para consultar las evidencias de las participaciones de los estudiantes en los eventos, con el fin de contribuir al proceso de caracterización y evaluación integral de los mismos.

Teniendo en cuenta la problemática anterior, se plantea como **problema a resolver**: ¿cómo gestionar la información asociada a los eventos estudiantiles de manera que contribuya al aumento de la accesibilidad, estandarización de la documentación y persistencia de las evidencias de los estudiantes en la Universidad de las Ciencias Informáticas?, definiéndose como **objeto de estudio**: los sistemas informáticos para la gestión de eventos estudiantiles, enmarcado en el **campo de acción**: los sistemas informáticos para la gestión de eventos estudiantiles en la Universidad de las Ciencias Informáticas. Para dar solución a este problema se plantea como **objetivo general**: desarrollar una aplicación informática para gestionar la información asociada a los eventos estudiantiles de manera que contribuya al aumento de la accesibilidad, estandarización de la documentación y persistencia de las evidencias de los estudiantes en la Universidad de las Ciencias Informáticas.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de gestión de la información asociada a los eventos estudiantiles?
- ¿Qué tecnologías y herramientas son las adecuadas para implementar la solución propuesta teniendo en cuenta las políticas de migración al software libre por las que aboga el país?
- ¿Cuáles son las características que debe cumplir la aplicación informática para que contribuya al

---

<sup>1</sup> Del inglés Portable Document Format: formato de documento portátil.

aumento de la accesibilidad, estandarización de la documentación y persistencia de las evidencias de los estudiantes en la UCI?

- ¿Qué tipos de pruebas de software se pueden utilizar en la validación de la propuesta de solución?

Para dar cumplimiento al objetivo general, se definen las siguientes **tareas de investigación**:

1. Descripción de los procesos de negocio asociados a los eventos estudiantiles.
2. Realización de un estudio del estado del arte a nivel internacional, nacional y local, para conocer los referentes teóricos del objeto de estudio.
3. Selección de las tecnologías y herramientas de desarrollo adecuadas para la implementación del software.
4. Análisis y diseño del sistema para la gestión de los eventos estudiantiles.
5. Descripción de la integración de la aplicación con el “Ecosistema de aplicaciones informáticas para la evaluación integral de los estudiantes” de la UCI.
6. Implementación del sistema para la gestión de los eventos estudiantiles.
7. Validación de la solución propuesta a través de pruebas de software.

Para llevar a cabo las tareas antes mencionadas se utilizan los siguientes **métodos científicos** de investigación:

#### ***Métodos teóricos:***

*Analítico-Sintético:* este es necesario en el estudio de la base teórica, de las herramientas y tecnologías a emplear, para definir las características esenciales de las mismas y sus relaciones.

*Modelación:* se utiliza en el desarrollo de la investigación para la generación del modelo de datos físico.

#### ***Métodos empíricos:***

*Entrevista:* Se realiza una entrevista estructurada a la responsable de la esfera de ideología dentro del secretariado de la FEU<sup>2</sup> de la UCI, la estudiante Niurka Socarrás Hernández, con el objetivo de obtener información referente a los pormenores en el proceso de gestión de los eventos estudiantiles, las principales dificultades que presenta este proceso y las características más importantes que debían ser incluidas en la solución.

*Observación:* se utiliza para comprender, mediante la percepción directa y objetiva, cómo se realiza el proceso de gestión de un evento estudiantil en sus diferentes fases.

---

<sup>2</sup> Federación Estudiantil Universitaria.

El presente trabajo se estructura en tres capítulos, en los que se encuentra el contenido distribuido de la siguiente manera:

En el **primer capítulo** se expone la fundamentación teórica de la investigación, la cual consiste en un estudio del estado del arte de los sistemas informáticos para la gestión de la información de los eventos estudiantiles; se describen los procesos asociados a estos, así como la metodología, las herramientas y las tecnologías seleccionadas para el desarrollo del software.

En el **segundo capítulo** se realiza una descripción de la solución propuesta, se analizan y describen los procesos de negocio, la lista de reserva del producto, la cual contiene los requisitos funcionales y no funcionales, las historias de usuario, la estimación y planificación de las mismas, así como las tareas de ingeniería asociadas a cada una de ellas, la arquitectura del sistema, las personas relacionadas con el sistema y los patrones de diseño utilizados para la solución del objetivo propuesto.

En el **tercer capítulo** se expone la descripción de los principales elementos desarrollados en la etapa de implementación y los resultados de las diferentes pruebas de software (unitarias, funcionales, del sistema y de aceptación) con el objetivo de garantizar la entrega de un sistema confiable tanto para el programador como para el cliente.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## 1.1 Introducción

En el presente capítulo se describen los procesos que se realizan para organizar los eventos estudiantiles. Luego, se realiza un estudio del estado del arte de los sistemas informáticos relacionados con el objeto de estudio. Finalmente, se presentan los elementos considerados en la selección de las tecnologías y herramientas necesarias para la implementación del software.

## 1.2 Conceptos asociados al dominio del problema

Para un mejor entendimiento del objeto de estudio, es de vital importancia definir los siguientes conceptos.

### 1.2.1 Sistema de gestión

Un sistema de gestión es el medio a través del cual una organización gestiona las partes interrelacionadas de su negocio para lograr sus objetivos. Estos objetivos pueden relacionarse con una serie de temas diferentes, incluyendo la calidad del producto o servicio, la eficiencia operativa, el desempeño ambiental, la salud y la seguridad en el lugar de trabajo y muchos más (ISO, 2017).

### 1.2.2 Información

La información es un conjunto organizado de datos procesados, que constituyen un mensaje sobre un determinado ente o fenómeno; es el conocimiento agrupado de manera que pueda usarse para fines diferentes. También, la información es un fenómeno que brinda significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los patrones del pensamiento humano (Ortega Martínez, 1997).

### 1.2.3 Sistema de gestión de información

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades (Bustelo Ruesta, y otros, 2001). Esto se restringe a la información manejada por una organización separándola de los demás aspectos como los recursos humanos. Además, constituye la vertiente más importante de la gestión del conocimiento, que abarca todos los procesos y actividades vinculadas a la generación, procesamiento, uso y transformación de los datos como fuentes futuras de información y posterior conocimiento. Diversos especialistas consideran que sin una adecuada gestión de la información es imposible llegar a la gestión del conocimiento, y la importancia que se le otorga es de un máximo nivel, esto queda reflejado en la siguiente afirmación: *“Es por lo tanto el paso previo, que cualquier organización debería dar antes de tratar de implantar un sistema de Gestión del Conocimiento.”* (Bustelo Ruesta, 2012). De forma general se puede decir que es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o

colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma (Curto Díaz, 2013).

#### **1.2.4 Sistema Informático**

Un sistema informático es un sistema funcional integrado, implantado en un entorno local de actividad empresarial que permite a los usuarios individuales obtener, compartir y gestionar la información mediante una combinación específica de software y hardware (FAO, 2005).

#### **1.2.5 Evento estudiantil**

Según el diccionario de la Real Academia Española, un evento es un suceso importante y programado, de índole social, académica, artística o deportiva (DRAE, 2017). En el contexto universitario, los protagonistas de estos sucesos son los estudiantes, y el propósito fundamental de cada evento es contribuir a la formación integral como profesional de los mismos.

### **1.3 Descripción de los procesos asociados a los eventos estudiantiles**

En la UCI se organizan eventos científicos, culturales, deportivos y copas académicas, los cuales son creados y organizados por el secretariado de la FEU, de la facultad o de la universidad, en dependencia del nivel del evento.

Un evento científico es aquel en el que los estudiantes participan con una ponencia de la cual son autores. Los eventos que se realizan en la universidad de carácter científico son: la WebXCuba, la Peña tecnológica, la Jornada Científica Estudiantil, el Fórum de Historia y el Seminario Juvenil Martiano, entre otros. Las ponencias o trabajos presentados en estos eventos, son agrupados en comisiones, estas últimas tienen asociado un tribunal que evalúa dichos trabajos. Un tribunal puede estar conformado por estudiantes y profesores, donde cada uno cumple funciones tales como: secretario, vocal, presidente, miembro o alumno ayudante. Este tipo de eventos puede realizarse a nivel de facultad, de universidad o nacional, permitiendo que un trabajo que obtuvo como resultado la condición de relevante, destacado o mención pueda ser presentado en los diferentes niveles del mismo evento.

El deporte constituye una de las esferas que, a partir del trabajo extensionista en la Universidad, permite promover acciones prácticas para consolidar la formación general integral de los estudiantes universitarios (UCI, 2012). Un evento deportivo es aquel en el que los estudiantes compiten de manera individual o por equipos en un deporte. Los eventos deportivos que se realizan en la universidad son: los Juegos Inter-Años, los Juegos Mella, entre otros. Además, la universidad tiene su propio circuito de torneos nacionales de ajedrez: UCI (en octubre), Remberto Fernández (en abril) y Ernesto Guevara (en junio). Los participantes se agrupan en delegaciones (generalmente se corresponden con las facultades), las cuales compiten entre sí mediante los resultados obtenidos (oro, plata o bronce) en las competencias.

La actividad cultural de la Universidad se organiza y coordina desde la Vicerrectoría de Extensión Universitaria, en coordinación con las facultades docentes y las organizaciones juveniles y estudiantiles; está dirigida, fundamentalmente, a la formación de una cultura general e integral de la comunidad en interacción con la sociedad (UCI, 2012). Un evento cultural es aquel en el que los estudiantes demuestran sus talentos y aptitudes artísticas, compitiendo en las distintas manifestaciones del arte, como la música, la danza, el teatro, las artes plásticas, el arte digital, los audiovisuales, entre otros. El evento cultural más importante que se realiza en la universidad es el Festival de Artistas Aficionados el cual se organiza desde las facultades hasta el nivel nacional. Al igual que en los eventos deportivos, en este tipo de eventos los estudiantes también se agrupan en delegaciones y estas compiten según la suma de los puntos obtenidos por los resultados de los participantes.

Las copas académicas son competencias donde se miden las habilidades y conocimientos relacionados a una materia o disciplina en particular, por ejemplo: técnicas de programación, ingeniería y gestión de software, bases de datos, idiomas, matemática, entre otras. Las copas académicas que se realizan en la universidad son: la Olimpiada de Idiomas, la Copa Pascal, la Copa de Ingeniería de Software, la Copa de Bases de Datos, la Olimpiada de Matemáticas, entre otras. Las participaciones pueden ser individuales o por equipos y los resultados son otorgados según el conocimiento, agilidad mental, calidad de la respuesta, o habilidad demostrada por parte de los estudiantes en las competencias.

#### 1.4 Estudio del estado del arte

A continuación, se realiza un análisis de los sistemas homólogos encontrados que gestionan información sobre eventos estudiantiles:

En el ámbito internacional se encontraron varios software de gestión de eventos, entre los que se pueden mencionar *EventBoost: El software de gestión de eventos* (Eventboost SA, 2016), *Bcongresos* (Brigantia, 2016), *Activa Impulso Tecnológico* (Activa Impulso Tecnológico, 2016), *USI* (Ungerboeck Software International, 2016) y *GesinTur* (Gesintur, 2016). Estas aplicaciones realizan la gestión de los eventos de manera online; permiten crear una página oficial para un evento, mediante la cual se inscriben las entidades interesadas en participar, definiendo los recursos que necesitan a través de un formulario de solicitud. Una vez que las entidades están registradas, los organizadores del evento les asignan los recursos solicitados o negocian con los clientes por medio del correo electrónico especificado en el momento de la inscripción. Dichos sistemas establecen opciones de pago, procesan automáticamente las tarjetas de crédito y permiten mantener actualizados a los participantes del evento mediante el envío de mensajes automáticos de confirmación y recordatorio. Es importante destacar que a pesar de que se encontraron diversas soluciones informáticas para la gestión de eventos estudiantiles a nivel mundial, no es factible utilizar las referenciadas anteriormente porque son software privativo, que para poder usarlos la UCI debe pagar licencias, además

no satisfacen necesidades propias del negocio a informatizar como por ejemplo la relación entre múltiples eventos y el soporte a los diferentes tipos de eventos que se celebran en la universidad.

*Sistema de Eventos de la Universidad de Murcia* (Universidad de Murcia, 2015); se puede utilizar para darle promoción a un evento o permitir la participación en el mismo; pero tiene como desventaja que es insuficiente para administrar los trabajos, los resultados y las comisiones; gestionando el evento pero no el proceso, o sea, no cubre las necesidades específicas del proceso de negocio a informatizar. Además, sería costosa la integración o comunicación con sistemas de la UCI, ya que sólo se puede acceder a los servicios que él mismo brinda.

*Open Conference Systems (OCS)* es una herramienta de publicación web gratuita que crea una web completa para una conferencia académica. Es una solución altamente flexible basada en PHP y puede hacer uso de bases de datos MySQL o PostgreSQL. (Simon Fraser University Library, 2014). Este sistema provee apoyo tecnológico a todos los procesos involucrados en la gestión de una conferencia, entre las características más relevantes se destacan las siguientes:

- Creación del sitio web de una o varias conferencias simultáneas.
- Redacción y envío de convocatoria de ponencias, ya sea por medio de anuncios en el sitio o envío de correos masivos mediante la plataforma.
- Recepción y aceptación de propuestas de ponencias y el resumen completamente en línea, además del monitoreo y control del proceso de revisión de los trabajos.
- Publicación de memorias del evento en línea, así como la posibilidad de búsqueda de trabajos por nombre, autor y tema.
- Calendarización de trabajos aprobados y exposiciones especiales del evento.
- Importación y exportación de los datos de la conferencia.
- Generación de informes y estadísticas del evento: número de inscritos, trabajos aprobados y rechazados, listado de participantes, ponentes, etc.

OCS es una herramienta potente, la integración de los módulos de un evento, la inscripción en línea, el envío de propuestas en línea y el monitoreo de los procesos de revisión científica son algunas de sus ventajas, pero tiene como desventaja que solamente se puede utilizar para gestionar conferencias académicas.

En el ámbito nacional se encontró el *Gestor de Eventos de la Universidad de Oriente* el cual utiliza OCS en su versión 2.3.6.0. Gestiona la participación en eventos pero tiene como desventaja que solamente incluye eventos científicos (Universidad de Oriente, 2016).

*Multieventos: Sistema de gestión de eventos*, fue desarrollado por la dirección de informatización de la UCI, permite acceder desde su página principal a diferentes sitios en los cuales se gestiona un evento de carácter científico. Tiene como desventaja que no permite la relación entre múltiples eventos ya que cada evento se gestiona de manera independiente, lo cual dificulta el seguimiento de una ponencia hacia los diferentes niveles de un evento; además no da cobertura a la diversidad de eventos que se realizan en la universidad, siendo estos no solo de carácter científico, sino también deportivos y culturales (Universidad de las Ciencias Informáticas, 2016).

A pesar de que se encontraron sistemas para la gestión de los eventos estudiantiles, el estudio realizado demuestra que estos solucionan de manera parcial el problema de la presente investigación, ya que no incluyen el proceso completo ni los diferentes escenarios a tener en cuenta. Por lo anteriormente expresado, se concluye que no se conoce ninguna herramienta de gestión de la información asociada a los eventos estudiantiles que solvante los problemas planteados y se adapte a los nuevos escenarios en la UCI.

### **1.5 Ecosistema de aplicaciones MAYA**

La presente investigación surge en el año 2014 cuando se crea en la Facultad Introdutoria de Ciencias Informáticas (FICI) el proyecto MAYA: “Ecosistema de aplicaciones informáticas para la evaluación integral de los estudiantes en el contexto universitario” (González Cardoso, 2015), el cual integra en una plataforma las aplicaciones que tributan evidencias sobre el desempeño estudiantil, lo que contribuye a mejorar el proceso de recopilación, procesamiento y persistencia de los grandes volúmenes de datos que se generan para la caracterización de los estudiantes en el contexto universitario.

Un ecosistema según la definición de la Real Academia de la Lengua Española es “comunidad de los seres vivos cuyos procesos vitales se relacionan entre sí y se desarrollan en función de los factores físicos de un mismo ambiente”. Llevando este concepto a la informática se puede definir como ecosistema de aplicaciones informáticas al: “conjunto de aplicaciones informáticas con funcionamiento independiente que se relacionan entre sí y comparten información en función de un contexto determinado” (González Cardoso, 2015).

Los sistemas que componen el ecosistema MAYA son:

- *ALMA*: “Sistema para la gestión de datos primarios”, es la aplicación encargada de controlar la información base de los estudiantes, profesores, especialistas y trabajadores que interactúan con la plataforma.
- *EDUCA*: Es un sistema para la gestión de la información asociada a la estrategia educativa, gestiona las disciplinas, los méritos y los objetivos por diferentes niveles (Cabrera Isasi, 2016).

- *dataFEU v.2.0*: Es un sistema para la gestión de los procesos de la FEU. Se centra principalmente en el proceso de integralidad y caracterización de los estudiantes registrando mediante evidencias la trayectoria de cada uno de ellos en un expediente digital (Rodríguez Angel, y otros, 2016).

El proyecto MAYA define para el desarrollo de estas aplicaciones informáticas el marco de trabajo Symfony 2 y la arquitectura REST<sup>3</sup> para la integración de las aplicaciones a través de servicios, ajustándose a la política de soberanía tecnológica del país.

## 1.6 Herramientas y tecnologías

### 1.6.1 Entorno de Desarrollo Integrado

Durante el desarrollo del sistema se utiliza como herramienta para la implementación del código, NetBeans 8.0, pues es un entorno de desarrollo integrado libre, gratuito y de código abierto. Sirve para un número importante de lenguajes de programación (Netbeans, 2016), entre los que se encuentra PHP<sup>4</sup>. Esta herramienta permite crear aplicaciones web con PHP, posee un potente depurador integrado y además viene con soporte para Symfony, un framework<sup>5</sup> Modelo-Vista-Controlador escrito en PHP y que además está definido por el proyecto MAYA para el desarrollo de sus aplicaciones informáticas.

### 1.6.2 Herramienta CASE<sup>6</sup>

Se selecciona Visual Paradigm para UML en su versión 8.0 como herramienta CASE para el modelado de la aplicación, ya que permite automatizar actividades manuales y mejorar la visión general de la ingeniería. Facilita la realización de las actividades asociadas a los procesos de software y el trabajo de análisis, diseño y codificación (Pressman, 2010).

Se utiliza UML como lenguaje de modelado en su versión 2.0, debido a que se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Viabiliza la realización de diagramas estáticos, dinámicos, de entorno y organizativos. Ofrece un estándar para describir de la manera más sencilla y entendible para cualquier desarrollador los aspectos conceptuales tales como procesos de negocio y funciones del sistema (Booch, y otros, 2000). UML permite comunicar ciertos conceptos claramente. El lenguaje natural es demasiado impreciso y se complica cuando se trata de conceptos más complejos. Por otra parte, el código es preciso, pero demasiado detallado, mientras UML ofrece cierta cantidad de precisión al resaltar solamente los detalles importantes.

---

<sup>3</sup> Del inglés Representational State Transfer: Transferencia de Estado Representacional.

<sup>4</sup> Del inglés Hypertext Preprocessor: Preprocesador de Hipertextos.

<sup>5</sup> Marco de trabajo.

<sup>6</sup> Del inglés *Computer Aided Software Engineering*: Ingeniería de Software Asistida por Computadora.

### 1.6.3 Sistema de control de versiones

Como herramienta para mantener el control de versiones se utiliza el Git en su versión 2.6.0, pues es un sistema de control de versiones distribuido, libre y de código abierto, diseñado para manejar desde pequeños a muy grandes proyectos con rapidez y eficiencia. Supera a otras herramientas, con funciones tales como ramificación local (la cual consiste en que el usuario puede tener múltiples ramas locales independientes entre sí en las cuales alojar su proyecto) y múltiples flujos de trabajo (GitHub, 2016).

### 1.6.4 Servidor de aplicaciones de prueba

Como servidor de aplicaciones web se utiliza Apache en su versión 2.4.9 debido a que es un servidor HTTP<sup>7</sup> de código abierto para sistemas operativos modernos, incluyendo UNIX, Microsoft Windows, Mac OS / X y NetWare. Su meta es proporcionar un seguro, eficiente y extensible servidor que proporcione servicios HTTP en ajuste a los actuales estándares HTTP (The Apache Software Foundation, 2016).

### 1.6.5 Herramientas para la realización de las pruebas

Para el desarrollo de las pruebas de carga sobre el sistema se utiliza la herramienta JMeter en su versión 2.12. JMeter es un software de código abierto, escrito en Java y diseñado para cargar el comportamiento funcional y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga (The Apache Software Foundation, 2016).

PHPUnit en su versión 4.8.35, es un framework de pruebas orientado a programadores para PHP. Es una instancia de la arquitectura xUnit y se utiliza fundamentalmente para la ejecución de pruebas unitarias en el lenguaje de programación PHP (Bergmann, 2016). PHPUnit se creó con la idea de que, cuanto antes se detecten los errores en el código, antes podrán ser corregidos. Además, utiliza *assertions* para verificar que el comportamiento de una unidad de código es el esperado (Bergmann, 2015).

### 1.6.6 Sistema gestor de bases de datos

Como sistema gestor de bases de datos se utiliza PostgreSQL en su versión 9.3, a causa de que está distribuido bajo licencia BSD<sup>8</sup> (la cual permite el uso del código fuente en software no libre), demuestra un buen funcionamiento con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (PostgreSQL, 2016).

---

<sup>7</sup> Del inglés *Hypertext Transfer Protocol*: Protocolo de transferencia de hipertextos.

<sup>8</sup> Del inglés Berkeley Software Distribution: Distribución de software Berkeley.

### 1.6.7 Lenguaje de programación

Como lenguaje de programación del lado del servidor se utiliza PHP en su versión 5.5.12, porque es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML<sup>9</sup>. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Es orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Posee capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destacándose su conectividad con PostgreSQL (PHP, 2016).

Javascript 1.8 como lenguaje de programación del lado del cliente. Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos porque emplea clases y herencia, típicas de la Programación Orientada a Objetos (Flanagan, 2006), es basado en prototipos, compatible con la mayoría de los navegadores, débilmente tipado y dinámico (Dominguez-Dorado, 2015). Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

### 1.6.8 Framework de desarrollo

Como marco de trabajo se utiliza Symfony en su versión 2.7.6, pues cuenta con una comunidad activa: desarrolladores, integradores, usuarios y otros colaboradores participan en el enriquecimiento en curso de esta herramienta. Se distribuye bajo licencia Open Source MIT de software libre, no impone restricciones y permite el desarrollo de código abierto. Symfony no solo es un conjunto de clases PHP sino que además provee toda una filosofía de trabajo y define el modelo de la arquitectura del sistema (Symfony, 2016). Está diseñado para explotar las potencialidades de PHP, lo cual supone mejoras en el rendimiento del framework.

Doctrine 2.0 se utiliza como mapeador de objetos-relacional (ORM por sus siglas en inglés), debido a que se sitúa encima de una poderosa capa de abstracción de base de datos (DBA<sup>10</sup>). Una de sus principales características es la opción de escribir consultas de bases de datos en un dialecto SQL<sup>11</sup> orientado a objetos denominado DQL<sup>12</sup>, inspirado por Hibernate HQL<sup>13</sup>, el cual permite mantener la flexibilidad sin requerir duplicado de código innecesario (Doctrine Team, 2016).

Se utiliza como motor de plantilla Twig en su versión 2.0, porque es rápido, seguro y flexible. Además, tiene una sintaxis muy concisa, que hace las plantillas más legibles. Twig tiene accesos directos para los patrones

---

<sup>9</sup> Del inglés HyperText Markup Language: Lenguaje de Marcas de Hipertexto.

<sup>10</sup> Del inglés Database Administrator: Administrador de Bases de Datos.

<sup>11</sup> Del inglés Structured Query Language: Lenguaje de consulta estructurada.

<sup>12</sup> Del inglés Doctrine Query Language: Lenguaje de Consulta de Doctrine.

<sup>13</sup> Nombre del lenguaje de consulta utilizado por Hibernate.

comunes (como tener un texto predeterminado que se muestra cuando se itera sobre un arreglo vacío) y es compatible con todo lo necesario para construir plantillas de gran alcance con facilidad (Twig, 2016).

HTML5 se utiliza como lenguaje de marcado, pues es el elemento de construcción más básico de una página web. Se usa para crear y representar visualmente una página web. Soporta imágenes, videos, juegos y también otro tipo de elementos multimedia. Es el lenguaje que describe la estructura y el contenido semántico de un documento web (Foundation Mozilla Developer, 2016).

Se utiliza CSS<sup>14</sup> para los estilos, ya que ofrece nuevas e interesantes posibilidades para crear un impacto con sus diseños y permite utilizar diversas hojas de estilo (WebFlux, 2012). Es utilizado para definir el aspecto de cada elemento: color del texto, tamaño y tipo de letra, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros. Como librería de estilos CSS se usó Bootstrap 3.3.1, este simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que se pueden crear interfaces que se adapten a los distintos navegadores apoyándose en un marco de trabajo potente con numerosos componentes web, los cuales ahorran esfuerzo y tiempo (Bootstrap, 2016). Esto permite que las páginas cuenten con un diseño adaptativo, con el fin de mostrar la información de manera apropiada en cualquier tipo de dispositivo.

Como librería de JavaScript se utiliza JQUERY, porque es rápida, pequeña y rica en funciones de JavaScript. Permite el recorrido y la manipulación de documentos HTML, además del manejo de eventos, animación y AJAX<sup>15</sup> mucho más simple, con un API<sup>16</sup> fácil de usar que funciona a través de una multitud de navegadores (jQuery Foundation, 2016).

Se utiliza AJAX para peticiones asincrónicas al servidor, debido a que engloba a todo un grupo de tecnologías (XHTML<sup>17</sup>, JavaScript, CSS, API y DOM<sup>18</sup>) y mantiene una comunicación asíncrona con el servidor en segundo plano, lo que permite realizar continuos cambios sin necesidad de recargar las páginas (Babin, 2007).

## 1.7 Metodología de desarrollo de software

Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Según Sommerville, estas metodologías son métodos de desarrollo iterativo que se centran en la especificación, diseño e implementación del sistema de forma incremental e implican directamente a los usuarios en el proceso de desarrollo (Sommerville, 2005).

---

<sup>14</sup> Del inglés Cascading Style Sheets: Hojas de Estilo en Cascada.

<sup>15</sup> Del inglés Asynchronous JavaScript And XML: JavaScript asíncrono y XML.

<sup>16</sup> Del inglés Application Programming Interface: Interfaz de Programación de Aplicaciones.

<sup>17</sup> Del inglés Extensible Hypertext Markup Language: Lenguaje de Marcado de Hipertexto Entendido.

<sup>18</sup> Del inglés Document Object Model: Modelo de Objetos del Documento.

Programación Extrema (XP por sus siglas en inglés), es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y preparación para enfrentar los cambios (Beck, 1999). Esta metodología posee las siguientes ventajas:

- Constante retroalimentación del cliente que ayuda a agregar nuevas características al sistema que se está desarrollando
- La programación en pares proporciona una mejor codificación porque las personas en un equipo tienen diferentes formas de pensar, cada uno ve un aspecto particular de la programación. Así que la codificación sería el resultado de la unión de estos aspectos.
- A medida que se desarrolla un nuevo código, se prueba e integra con el código antiguo. Toda la base de código se reconstruye constantemente y se vuelve a probar de forma automatizada.
- Errores simples como errores de sintaxis o nombres repetidos de variables pueden ser fácilmente capturados y arreglados, esto puede reducir el tiempo de depuración.
- Propone una planificación constante, planteando el hecho de planear y re-planear todo el tiempo pues la realidad siempre interrumpe los planes.
- La refactorización constante del software. Esto significa que se buscan posibles mejoras del software y se implementan inmediatamente. De esta manera el software siempre debe ser fácil de entender y cambiar cuando se implementen nuevas historias (Amlani, 2012).

El ciclo de vida de XP consiste de las siguientes fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. La explicación de cada una de estas fases se muestra a continuación:

**Exploración:** los clientes realizan a grandes rasgos las historias de usuario (técnica utilizada en XP para especificar los requisitos del software) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La etapa de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

**Planificación de la Entrega:** el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de

ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

**Iteraciones:** incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas y tareas no terminadas en la iteración anterior. Al final de la última iteración el sistema estará listo para entrar en producción.

**Producción:** requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta etapa. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la etapa de mantenimiento).

**Mantenimiento:** Consiste en la implementación de nuevas versiones, asegurando a la vez el funcionamiento del sistema. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. Cada nueva entrega, debe comenzar por una etapa de exploración y es ahí cuando se cierra el ciclo. La única manera de interrumpir el ciclo de XP es que ocurra la muerte del proyecto.

**Muerte del proyecto:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo (Beck, 1999).

El proyecto MAYA define la metodología XP como modelo de desarrollo para sus aplicaciones, por brindarle al equipo de trabajo la comodidad y flexibilidad a cambios que surjan en cualquier etapa del ciclo de vida de la aplicación. Además, el grupo de desarrollo es pequeño, con una formación elevada y los clientes tienen una estrecha vinculación con el equipo de desarrollo. Los artefactos que se generan mediante el uso de esta metodología son: las historias de usuarios, las tarjetas CRC<sup>19</sup> y las tareas de ingeniería.

---

<sup>19</sup> Clase-Responsabilidad-Colaboración.

## Conclusiones parciales

Una vez realizadas las tareas 1, 2 y 3 de la investigación, cuyos resultados han sido expuestos en el presente capítulo, los autores concluyen que:

- A partir de la descripción de los procesos asociados a la gestión de los eventos estudiantiles se obtuvo una visión general de cómo funciona actualmente este proceso en la UCI.
- El estudio del estado del arte realizado demostró que no se conoce ninguna herramienta para la gestión de la información asociada a los eventos estudiantiles que solvante los problemas planteados, ni se adapte a los nuevos escenarios de la UCI. No obstante, este estudio permitió identificar características útiles a tener en cuenta en la propuesta de solución.
- Las herramientas y tecnologías elegidas para el desarrollo de la aplicación son libres, lo cual está en correspondencia con la política de migración al software libre ejecutada en la universidad y en el país actualmente.

## CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### 2.1 Introducción

En el presente capítulo se describen los procesos de negocio y la lista de reserva del producto. Se muestran algunas de las historias de usuarios, así como algunas tareas de ingeniería asociadas a ellas. Se exponen los elementos asociados a la estimación y planificación del primer plan de entregas. Además, se define la arquitectura del sistema, se muestran algunas de las tarjetas CRC realizadas y se describen los patrones de diseño utilizados y las personas que tienen relación con el sistema.

### 2.2 Descripción de los procesos de negocio asociados a la organización de los eventos estudiantiles

#### ***Eventos científicos***

El proceso comienza con la elaboración y lanzamiento de la convocatoria por parte de los dirigentes estudiantiles a un nivel específico (universidad o facultad). Los estudiantes que deseen participar envían sus trabajos a la comisión organizadora. Dichos trabajos son agrupados por temáticas en comisiones. A cada comisión se le asigna un tribunal integrado por dos profesores (presidente y vocal) y un alumno ayudante. Una vez concluida la distribución de los trabajos por comisiones se notifica a todos los involucrados. El día de la realización del evento los participantes exponen sus trabajos y los tribunales evalúan los mismos seleccionando aquellos que cumplen con los requisitos para ser relevante, destacado o mención. Una vez finalizado el evento la información asociada al mismo es almacenada y divulgada.

#### ***Eventos deportivos***

El proceso comienza cuando los dirigentes estudiantiles a un nivel específico (universidad o facultad) definen los deportes en los que se va a competir y las delegaciones que participarán (a nivel de universidad se corresponden con las facultades y a nivel de facultad se corresponden con agrupaciones de años). Una vez definido esto, proceden a la elaboración y lanzamiento de la convocatoria. Los responsables de cada delegación definen los capitanes de los deportes notificando a los interesados los datos de estos. Los estudiantes que deseen participar en un deporte se ponen en contacto con el capitán quien tiene la responsabilidad de seleccionar los que participarán. En cada uno de los deportes se puede competir en dos modalidades: individual o por equipos. El día de la realización de la competencia los atletas participan en la pugna por las medallas de oro, plata o bronce. Una vez finalizado el evento la información asociada al mismo es almacenada y divulgada.

#### ***Eventos culturales***

El proceso comienza cuando los dirigentes estudiantiles a un nivel específico (universidad o facultad) definen las manifestaciones artísticas en las que se va a competir y las delegaciones que participarán. Una

vez definido esto, proceden a la elaboración y lanzamiento de la convocatoria. Los responsables de cada delegación definen los encargados de las unidades artísticas notificando a los interesados los datos de estos. Los estudiantes que deseen participar en una manifestación se ponen en contacto con el encargado quien tiene la responsabilidad de seleccionar los que participarán. El día de la realización del evento un jurado especializado evalúa las obras presentadas por los artistas, otorgando los resultados de oro, plata o bronce según corresponda. Una vez finalizado el evento la información asociada al mismo es almacenada y divulgada.

### **Eventos de copas académicas**

El proceso comienza con la elaboración y lanzamiento de la convocatoria por parte de los dirigentes estudiantiles a un nivel específico (universidad o facultad). Los participantes compiten por categorías según lo definido para el evento de manera individual o por equipo. El día de la realización de la copa los estudiantes participan en cada una de las categorías con la posibilidad de obtener primero, segundo o tercer lugar. Una vez finalizado el evento la información asociada al mismo es almacenada y divulgada.

## **2.3 Lista de reserva del producto**

En la lista de reserva del producto se definen y priorizan las funcionalidades que tiene el sistema y se describen los requisitos no funcionales del software (Acebo, 2012). Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes para un sistema que ayude a resolver algún problema, como el control de un dispositivo, hacer un pedido o encontrar información (Sommerville, 2005). A continuación se presenta la lista de reserva del producto de la presente investigación:

### **2.3.1 Requisitos funcionales**

*Tabla 1 Requisitos funcionales asociados al Módulo de administración y seguridad.*

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF1</b>	Autenticar usuario	ALTA
<b>RF2</b>	Registrar usuario	ALTA
<b>RF3</b>	Modificar usuario	ALTA
<b>RF4</b>	Listar usuarios	BAJA
<b>RF5</b>	Buscar usuario por el nombre de usuario	BAJA
<b>RF6</b>	Bloquear usuario	BAJA
<b>RF7</b>	Eliminar usuario	BAJA
<b>RF8</b>	Listar trazas	ALTA

<b>RF9</b>	Mostrar traza	ALTA
<b>RF10</b>	Buscar traza	MEDIA
<b>RF11</b>	Mostrar perfil del usuario	ALTA

*Tabla 2 Requisitos funcionales asociados al Módulo Común.*

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF12</b>	Registrar evento	ALTA
<b>RF13</b>	Listar eventos	ALTA
<b>RF14</b>	Modificar evento	ALTA
<b>RF15</b>	Mostrar evento	ALTA
<b>RF16</b>	Eliminar evento	MEDIA
<b>RF17</b>	Finalizar evento	MEDIA
<b>RF18</b>	Definir evento superior	BAJA
<b>RF19</b>	Buscar evento por su nombre	BAJA
<b>RF20</b>	Registrar tipo de resultado	MEDIA
<b>RF21</b>	Listar tipos de resultado	BAJA
<b>RF22</b>	Modificar tipo de resultado	BAJA
<b>RF23</b>	Eliminar tipo de resultado	MEDIA
<b>RF24</b>	Definir tipo de resultado para un evento	MEDIA
<b>RF25</b>	Eliminar el tipo de resultado para un evento	MEDIA
<b>RF26</b>	Evaluar participación	ALTA
<b>RF27</b>	Modificar si se presentó o no una participación	ALTA
<b>RF28</b>	Registrar tipo de evento	MEDIA
<b>RF29</b>	Modificar tipo de evento	BAJA
<b>RF30</b>	Listar tipos de evento	BAJA
<b>RF31</b>	Eliminar tipo de evento	MEDIA
<b>RF32</b>	Registrar estado de evento	MEDIA
<b>RF33</b>	Listar estados de evento	BAJA
<b>RF34</b>	Modificar estado de evento	BAJA
<b>RF35</b>	Eliminar estado de evento	MEDIA

<b>RF36</b>	Registrar nivel	MEDIA
<b>RF37</b>	Listar niveles	BAJA
<b>RF38</b>	Modificar nivel	BAJA
<b>RF39</b>	Eliminar nivel	MEDIA
<b>RF40</b>	Registrar miembro de la comisión organizadora del evento	ALTA
<b>RF41</b>	Listar miembros de la comisión organizadora del evento	MEDIA
<b>RF42</b>	Eliminar miembro de la comisión organizadora del evento	ALTA
<b>RF43</b>	Registrar delegación	ALTA
<b>RF44</b>	Modificar delegación	BAJA
<b>RF45</b>	Eliminar delegación	MEDIA
<b>RF46</b>	Mostrar delegación	BAJA
<b>RF47</b>	Registrar responsable de delegación	MEDIA
<b>RF48</b>	Listar responsables de delegación	MEDIA
<b>RF49</b>	Eliminar responsable de delegación	MEDIA
<b>RF50</b>	Generar reportes	MEDIA
<b>RF51</b>	Enviar correos	BAJA
<b>RF52</b>	Notificar por correo	MEDIA

*Tabla 3 Requisitos funcionales asociados al Módulo de eventos científicos.*

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF53</b>	Registrar comisión	ALTA
<b>RF54</b>	Listar comisiones	ALTA
<b>RF55</b>	Modificar comisión	MEDIA
<b>RF56</b>	Mostrar comisión	ALTA
<b>RF57</b>	Eliminar comisión	MEDIA
<b>RF58</b>	Listar trabajos sin asignar	MEDIA
<b>RF59</b>	Listar trabajos asignados	MEDIA

<b>RF60</b>	Asignar trabajo a una comisión	ALTA
<b>RF61</b>	Eliminar trabajo de una comisión	ALTA
<b>RF62</b>	Registrar trabajo	ALTA
<b>RF63</b>	Modificar trabajo	MEDIA
<b>RF64</b>	Mostrar trabajo	ALTA
<b>RF65</b>	Eliminar trabajo	ALTA
<b>RF66</b>	Registrar autor de un trabajo	ALTA
<b>RF67</b>	Listar autores de un trabajo	MEDIA
<b>RF68</b>	Eliminar autor de un trabajo	ALTA
<b>RF69</b>	Registrar tutor de un trabajo	MEDIA
<b>RF70</b>	Listar tutores de un trabajo	MEDIA
<b>RF71</b>	Eliminar tutor de un trabajo	MEDIA
<b>RF72</b>	Subir trabajo de nivel	MEDIA
<b>RF73</b>	Registrar miembro del tribunal de una comisión	ALTA
<b>RF74</b>	Listar miembros del tribunal	MEDIA
<b>RF75</b>	Modificar miembro del tribunal	MEDIA
<b>RF76</b>	Eliminar miembro del tribunal	MEDIA
<b>RF77</b>	Registrar función	MEDIA
<b>RF78</b>	Listar funciones	BAJA
<b>RF79</b>	Modificar función	BAJA
<b>RF80</b>	Eliminar función	MEDIA
<b>RF81</b>	Registrar estado de trabajo	MEDIA
<b>RF82</b>	Listar estados de trabajo	BAJA
<b>RF83</b>	Modificar estado de trabajo	BAJA
<b>RF84</b>	Eliminar estado de trabajo	MEDIA
<b>RF85</b>	Registrar tipo de trabajo	MEDIA
<b>RF86</b>	Listar tipos de trabajo	BAJA
<b>RF87</b>	Modificar tipo de trabajo	BAJA
<b>RF88</b>	Eliminar tipo de trabajo	MEDIA

Tabla 4 Requisitos funcionales asociados al Módulo de eventos deportivos.

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF89</b>	Registrar deporte en un evento	ALTA
<b>RF90</b>	Listar deportes de un evento	ALTA
<b>RF91</b>	Modificar deporte de un evento	MEDIA
<b>RF92</b>	Mostrar deporte de un evento	ALTA
<b>RF93</b>	Eliminar deporte de un evento	MEDIA
<b>RF94</b>	Registrar competencia de un deporte	ALTA
<b>RF95</b>	Modificar competencia de un deporte	MEDIA
<b>RF96</b>	Mostrar competencia de un deporte	ALTA
<b>RF97</b>	Eliminar competencia de un deporte	MEDIA
<b>RF98</b>	Registrar equipo deportivo en una competencia	ALTA
<b>RF99</b>	Mostrar equipo deportivo	ALTA
<b>RF100</b>	Eliminar equipo deportivo	ALTA
<b>RF101</b>	Registrar integrante de un equipo deportivo	ALTA
<b>RF102</b>	Listar integrantes de un equipo deportivo	ALTA
<b>RF103</b>	Eliminar integrante de un equipo deportivo	ALTA
<b>RF104</b>	Registrar participante en una competencia	ALTA
<b>RF105</b>	Listar participantes de una competencia	ALTA
<b>RF106</b>	Eliminar participante de una competencia	MEDIA
<b>RF107</b>	Registrar deporte	MEDIA
<b>RF108</b>	Listar deportes	BAJA

<b>RF109</b>	Modificar deporte	BAJA
<b>RF110</b>	Eliminar deporte	MEDIA

*Tabla 5 Requisitos funcionales asociados al Módulo de eventos culturales.*

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF111</b>	Registrar manifestación en un evento	ALTA
<b>RF112</b>	Listar manifestaciones del evento	ALTA
<b>RF113</b>	Modificar manifestación del evento	MEDIA
<b>RF114</b>	Mostrar manifestación del evento	ALTA
<b>RF115</b>	Eliminar manifestación del evento	MEDIA
<b>RF116</b>	Registrar modalidad de una manifestación	ALTA
<b>RF117</b>	Modificar modalidad de una manifestación	MEDIA
<b>RF118</b>	Mostrar modalidad de una manifestación	ALTA
<b>RF119</b>	Eliminar modalidad de una manifestación	MEDIA
<b>RF120</b>	Registrar participante en una modalidad	ALTA
<b>RF121</b>	Listar participantes de una modalidad	ALTA
<b>RF122</b>	Eliminar participante de una modalidad	MEDIA
<b>RF123</b>	Registrar manifestación	MEDIA
<b>RF124</b>	Listar manifestaciones	BAJA
<b>RF125</b>	Modificar manifestación	BAJA
<b>RF126</b>	Eliminar manifestación	MEDIA

Tabla 6 Requisitos funcionales asociados al Módulo de copas académicas.

<b>Nº</b>	<b>Nombre</b>	<b>Prioridad para el cliente</b>
<b>RF127</b>	Registrar categoría	ALTA
<b>RF128</b>	Listar categorías	ALTA
<b>RF129</b>	Modificar categoría	MEDIA
<b>RF130</b>	Mostrar categoría	ALTA
<b>RF131</b>	Eliminar categoría	MEDIA
<b>RF132</b>	Registrar equipo en una categoría	ALTA
<b>RF133</b>	Mostrar equipo	ALTA
<b>RF134</b>	Eliminar equipo	ALTA
<b>RF135</b>	Registrar integrante de un equipo	ALTA
<b>RF136</b>	Listar integrantes de un equipo	ALTA
<b>RF137</b>	Eliminar integrante de un equipo	ALTA
<b>RF138</b>	Registrar participante en una categoría	ALTA
<b>RF139</b>	Listar participantes de una categoría	ALTA
<b>RF140</b>	Eliminar participante de una categoría	MEDIA
<b>RF141</b>	Registrar local	MEDIA
<b>RF142</b>	Listar locales	BAJA
<b>RF143</b>	Modificar local	BAJA
<b>RF144</b>	Eliminar local	MEDIA
<b>RF145</b>	Registrar estado de equipo	MEDIA
<b>RF146</b>	Listar estados de equipo	BAJA
<b>RF147</b>	Modificar estado de equipo	BAJA
<b>RF148</b>	Eliminar estado de equipo	MEDIA

### **2.3.2** *Requisitos no funcionales*

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido o confiable. Se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar. Se conocen

como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (Sommerville, 2011).

#### *Usabilidad:*

- **RNF1:** La aplicación debe estar dirigida a registrar la información referente al proceso de gestión de los eventos estudiantiles en la Universidad de las Ciencias Informáticas.
- **RNF2:** Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista los íconos y menús correspondientes.
- **RNF3:** Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que se está realizando mediante el mismo.
- **RNF4:** Para el despliegue del software, deberá estar instalado en el servidor el sistema operativo Linux, donde se encuentre PHP v5.5.12 con las librerías php5-ldap, php5-gd, php5-mcrypt, php5-xsl, php5-openssl, Apache 2.4.9 con el módulo rewrite activado.
- **RNF5:** Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 3.6 o superior.
- **RNF6:** La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos PostgreSQL.

#### *Seguridad:*

- **RNF7:** El sistema de autenticación registrará datos de los usuarios (usuario, nombre, apellidos, id<sup>20</sup> de expediente) y tendrá control de la identidad de los mismos a través del servidor LDAP<sup>21</sup> UCI.
- **RNF8:** Debe proveer algún mecanismo seguro de encriptación y transferencia de datos.
- **RNF9:** Se debe mantener una seguridad a nivel de usuarios y contraseñas codificadas para el acceso a la base de datos.
- **RNF10:** El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.

#### *Eficiencia:*

- **RNF11:** El sistema debe responder las solicitudes de los usuarios en menos de dos segundos como promedio, no superando los cuatro segundos salvo en la generación de reportes.

---

<sup>20</sup> Abreviatura para identificador.

<sup>21</sup> Del inglés Lightweight Directory Access Protocol: Protocolo Ligero/Simplificado de Acceso a Directorios.

- **RNF12:** El sistema debe ser capaz de responder satisfactoriamente ante la concurrencia de los usuarios que interactúen con el mismo.

*Interfaz:*

- **RNF13:** El sistema debe ser capaz de obtener de sistemas externos (aplicaciones del ecosistema MAYA) la información que necesita para complementar su funcionamiento.
- **RNF14:** El sistema debe brindar la información relacionada a los eventos estudiantiles a través de servicios web públicos a sistemas externos.

## 2.4 Historias de usuario

La historia de usuario es una técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Jeffries, y otros, 2000). En ellas se emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema, se utilizan para hacer estimaciones de tiempo y para el plan de entregas, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

En la presente investigación se redactaron 51 historias de usuario, de las cuales a continuación se muestran cuatro, el resto se encuentra en el Anexo 1:

*Tabla 7 Historia de Usuario Gestionar evento.*

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Nombre de Historia de Usuario:</b> Gestionar evento
<b>Usuario:</b> Organizador, Administrador	
<b>Programador:</b> Alejandro M. Medina Pichs	<b>Iteración asignada:</b> Primera
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 0,5
<b>Riesgo en desarrollo:</b> Bajo	<b>Puntos reales:</b> 0,5
<b>Descripción:</b> Esta actividad debe permitir registrar, mostrar, modificar, listar y eliminar un evento. De un evento se registra: el nombre del evento, la edición, la fecha de inicio y de fin, la entidad a la que pertenece dicho evento, el lugar donde será desarrollado, una breve descripción del evento, el logo que le identificará, el tipo de evento, el nivel al que pertenece, la convocatoria y el cronograma.	

**Observación:** El usuario deberá proporcionar los archivos del cronograma, la convocatoria y el logo del evento.

*Tabla 8 Historia de Usuario Asignar trabajo a comisión.*

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre de Historia de Usuario:</b> Asignar trabajo a comisión
<b>Usuario:</b> Organizador, Administrador	
<b>Programador:</b> Alejandro M. Medina Pichs	<b>Iteración asignada:</b> Primera
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 0,5
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 0,5
<b>Descripción:</b> Esta actividad debe permitir asignar un trabajo a una comisión. Dado el conjunto de trabajos sin asignar, se seleccionan algunos para ser asignados a una comisión determinada.	
<b>Observación:</b> El estado del trabajo deberá cambiar a “Asignado”.	

*Tabla 9 Historia de Usuario Subir trabajo de nivel.*

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre de Historia de Usuario:</b> Subir trabajo de nivel
<b>Usuario:</b> Organizador, Administrador	
<b>Programador:</b> Alejandro M. Medina Pichs	<b>Iteración asignada:</b> Primera
<b>Prioridad en el negocio:</b> Media	<b>Puntos estimados:</b> 0,5
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 0,5
<b>Descripción:</b> Esta actividad debe permitir registrar un trabajo en el evento del nivel superior. El objetivo de subir un trabajo de nivel es que los trabajos que obtuvieron resultados en niveles inferiores, puedan estar presentes en el evento de nivel superior sin la necesidad de subir el documento del trabajo nuevamente.	
<b>Observación:</b> El trabajo debe quedar registrado en el evento de nivel superior como un trabajo no asignado.	

*Tabla 10 Historia de Usuario Finalizar evento*

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre de Historia de Usuario:</b> Finalizar evento

<b>Usuario:</b> Organizador, Administrador	
<b>Programador:</b> Alejandro M. Medina Pichs	<b>Iteración asignada:</b> Primera
<b>Prioridad en el negocio:</b> Media	<b>Puntos estimados:</b> 0,3
<b>Riesgo en desarrollo:</b> Medio	<b>Puntos reales:</b> 0,3
<b>Descripción:</b> Esta actividad debe permitir que un evento esté en un estado en el cual no puedan ser modificados los resultados, ni ninguna otra información del mismo.	
<b>Observación:</b> El estado del evento debe cambiar a "Finalizado".	

## 2.5 Plan de entrega

Se planificaron cuatro entregas, la primera entrega correspondiente al módulo de eventos científicos; la segunda al módulo de eventos deportivos; la tercera al módulo de eventos culturales y la cuarta al módulo copas académicas. La descripción de cada uno de los módulos se encuentra en el epígrafe 2.8.2. A continuación se presenta el plan de la primera entrega, el resto de los planes de entrega se encuentran en el Anexo 2:

*Tabla 11 Plan de entrega para el Módulo de eventos científicos.*

<b>Aplicación</b>	<b>Fin de la 1ra Iteración (25/10/2015)</b>	<b>Fin de la 2da Iteración (20/12/2015)</b>
Sistema para la gestión de los eventos estudiantiles	Funcionalidades con prioridad alta.	Funcionalidades con prioridad media.

## 2.6 Estimación y planificación

La estimación y planificación de iteraciones tiene como entrada la relación de historias de usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente. La duración de la historia de usuario es el valor real que se le asignó, esta duración se expresa en semanas. Como resultado de las historias de usuarios recogidas con anterioridad, se llegó a la siguiente planificación para la primera entrega, las otras tres planificaciones que se realizaron se encuentran en el Anexo 3:

*Tabla 12 Estimación y planificación para la primera entrega.*

<b>Iteraciones</b>	<b>Descripción</b>	<b>Estimación</b>	<b>Duración total</b>
<b>Prioridad: Alta</b>			
	Autenticar usuario	0,3	
	Gestionar usuario	0,5	
	Mostrar perfil de usuario	0,5	
	Gestionar eventos	0,5	
	Asignar trabajo a comisión	0,5	
	Evaluar participación	0,5	

<b>1ra</b>	Modificar si se presentó o no una participación	0,5	6,4 semanas
	Gestionar miembro de la comisión organizadora del evento	0,5	
	Gestionar delegación	0,3	
	Gestionar comisión	0,5	
	Eliminar trabajo de una comisión	0,3	
	Gestionar trabajo	0,5	
	Gestionar autor de un trabajo	0,5	
	Gestionar miembros del tribunal	0,5	
<b>Prioridad: Media</b>			
<b>2da</b>	Buscar trazas	0,5	6,35 semanas
	Subir trabajo de nivel	0,5	
	Finalizar evento	0,3	
	Gestionar tipo de resultado	0,3	
	Definir tipo de resultado para un evento	0,3	
	Eliminar tipo de resultado para un evento	0,15	
	Gestionar tipo de evento	0,3	
	Gestionar estado del evento	0,3	
	Gestionar nivel	0,3	
	Gestionar responsable de delegación	0,5	
	Generar reportes	0,5	
	Notificar por correo	0,5	
	Listar trabajos sin asignar	0,5	
	Gestionar tutor de un trabajo	0,5	
	Gestionar funciones	0,3	
	Gestionar estado del trabajo	0,3	
Gestionar tipo de trabajo	0,3		
<b>Prioridad: Baja</b>			
<b>3ra</b>	Bloquear usuario	0,5	1,5 semanas
	Definir evento de nivel superior	0,5	
	Enviar correos	0,5	

## 2.7 Tareas de ingeniería

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración (Letelier Torre, y otros, 2006). Estas asignaciones quedan descritas mediante las tareas de ingeniería. A continuación se muestran 5 de las 87 realizadas durante el desarrollo de la presente investigación, el resto se encuentra en el Anexo 4:

Tabla 13 Tarea de Ingeniería No. 1.1 para HU "Gestionar evento".

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 1.1	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre tarea:</b> Diseñar las interfaces para gestionar un evento	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0,2

<b>Fecha inicio:</b> 20/03/2015	<b>Fecha fin:</b> 22/03/2015
<b>Programador responsable:</b> Alejandro M. Medina Pichs	
<b>Descripción:</b> Desarrollar la interfaz de usuario, incluyendo los formularios necesarios para las opciones de registrar, modificar, listar y eliminar un evento.	

*Tabla 14 Tarea de Ingeniería No. 1.2 para HU "Gestionar evento".*

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 1.2	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre tarea:</b> Implementar las funcionalidades para gestionar un evento.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0,3
<b>Fecha inicio:</b> 23/03/2015	<b>Fecha fin:</b> 25/03/2015
<b>Programador responsable:</b> Alejandro M. Medina Pichs	
<b>Descripción:</b> Implementar las funcionalidades de registrar, modificar, listar y eliminar un evento.	

*Tabla 15 Tarea de Ingeniería No. 2.1 para HU "Asignar trabajo a comisión".*

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 2.1	<b>Historia de Usuario:</b> No. 2 Asignar trabajo a comisión
<b>Nombre tarea:</b> Diseñar la interfaz para asignar un trabajo a una comisión.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0,2
<b>Fecha inicio:</b> 26/03/2015	<b>Fecha fin:</b> 27/03/2015
<b>Programador responsable:</b> Alejandro M. Medina Pichs	
<b>Descripción:</b> Desarrollar la interfaz de usuario para asignar un trabajo a una comisión, teniendo en cuenta la posibilidad de asignar varios trabajos a la vez.	

*Tabla 16 Tarea de Ingeniería No. 2.2 para HU "Asignar trabajo a comisión".*

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 2.2	<b>Historia de Usuario:</b> No. 2 Asignar trabajo a comisión
<b>Nombre tarea:</b> Implementar la funcionalidad de asignar trabajo a comisión.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0,3
<b>Fecha inicio:</b> 28/03/2015	<b>Fecha fin:</b> 30/03/2015
<b>Programador responsable:</b> Alejandro M. Medina Pichs	

<b>Descripción:</b> Implementar la funcionalidad de asignar trabajo a comisión, teniendo en cuenta la posibilidad de asignar varios trabajos a la vez.
--

## 2.8 Arquitectura del Sistema

La arquitectura de software de un sistema informático es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de esos elementos y la relación entre ellos (Bass, 1998).

La arquitectura no es el software operacional sino la representación que capacita al ingeniero del software para: (1) analizar la efectividad del diseño para la consecución de los requisitos fijados, (2) considerar las alternativas arquitectónicas en una etapa en la cual hacer cambios en el diseño es relativamente fácil, y (3) reducir los riesgos asociados a la construcción del software (Pressman, 2010).

### 2.8.1 Patrón Modelo Vista Controlador

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que el sistema sea mucho más fácil de implementar y mantener.

Este patrón presenta varias ventajas (Fernández Romero, y otros, 2012):

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- API muy bien definida; cualquiera que use la API podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión dinámica entre el Modelo y sus Vistas; se produce en tiempo de ejecución, no en tiempo de compilación.

En la siguiente figura se muestra el patrón MVC sobre el que trabaja Symfony 2:

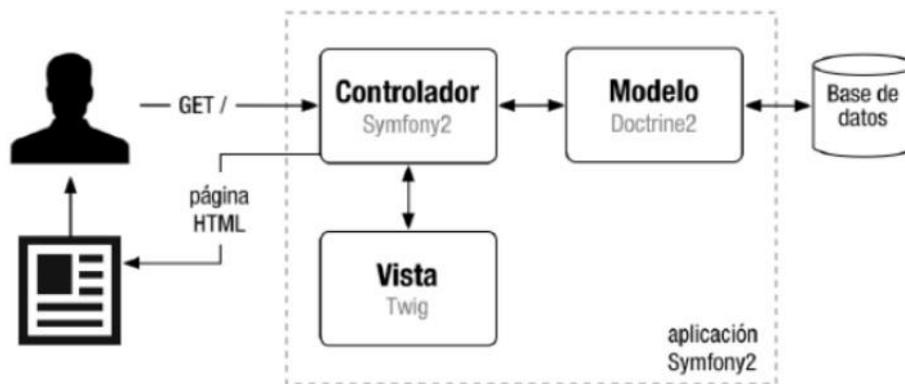


Figura 1 Patrón MVC

*El Modelo* es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones, no tiene conocimiento específico de los Controladores o de las Vistas, ni contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo. En el sistema el modelo está representado por el conjunto de clases PHP ubicadas en el directorio `src/AppBundle/Entity`, cuyos nombres hacen uso del patrón CamelCase en su variante UpperCamelCase, diferenciando las tablas de los nomencladores por los prefijos Tb y Nm, respectivamente. El uso de este patrón sigue la práctica XP de emplear estándares de codificación (Letelier Torre, y otros, 2006).

*La Vista* es el objeto que maneja la presentación visual de los datos representados por el Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo. Todas las vistas en la aplicación se encuentran en el directorio `app/Resources/views` y cada una de estas posee la extensión `.html.twig`.

*El Controlador* es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. El Controlador interactúa con el Modelo a través de una referencia al propio Modelo. El framework Symfony define su directorio genérico en `src/AppBundle/Controller`, encontrándose todas las clases controladoras en esta carpeta, utilizando también para la nomenclatura el patrón CamelCase en su variante UpperCamelCase, terminando con el sufijo Controller definido por Symfony.

El patrón MVC se utiliza para el diseño de aplicaciones con interfaces complejas. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Se trata de realizar un diseño que desacople la Vista del Modelo, con la finalidad de mejorar la reusabilidad de las

partes. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos (Fernández Romero, y otros, 2012).

### 2.8.2 Módulos del Sistema

Teniendo en cuenta los requisitos definidos en el epígrafe 2.3.1, el software se divide en 6 módulos, los cuales se describen a continuación:



Figura 2 Módulos del sistema.

#### Módulo de administración y seguridad

Se encarga del funcionamiento y seguridad del sistema utilizando el bundle<sup>22</sup> ZTEC de Symfony 2, el cual se integra con los directorios activos. Se empleó el patrón de control de acceso basado en roles (RBAC por sus siglas en inglés), el mismo es una función de seguridad, donde se establecen las funcionalidades específicas a las cuales puede acceder cada usuario. Enfoca su interés principalmente en determinar qué usuarios y qué grupos de usuarios pueden ejecutar qué tipo de operación sobre qué tipo de recurso. Los roles definidos en el sistema se describen en el epígrafe 2.10. En este módulo se gestionan los usuarios y las trazas de las acciones que estos realizan en el sistema.

#### Módulo Común

Gestiona las funcionalidades que son comunes a todos los tipos de eventos, por ejemplo: el registro y finalización del evento, la gestión de las comisiones organizadoras, los tipos de resultados, las delegaciones, el envío de notificaciones por correo, entre otras. Cuenta con las siguientes especializaciones: Módulo de

<sup>22</sup> Nombre que reciben los paquetes en Symfony.

eventos científicos, Módulo de copas académicas, Módulo de eventos culturales y Módulo de eventos deportivos.

### **Módulo de eventos científicos**

Gestiona los eventos científicos, lo cual incluye la gestión de los trabajos, las comisiones, los tribunales y los resultados. Haciendo uso del bundle TCPDF genera reportes nominales y estadísticos, por ejemplo: listado de comisiones, listado de participantes, listado de trabajos con resultados, listado de tribunales, entre otros.

### **Módulo de copas académicas**

Gestiona las copas académicas, lo cual incluye la gestión de las categorías, los equipos y las participaciones de los estudiantes. Haciendo uso del bundle TCPDF genera reportes nominales y estadísticos, por ejemplo: listado de participantes por categorías, listado de participantes por delegación, listado de participantes por local, entre otros.

### **Módulo eventos culturales**

Gestiona los eventos culturales, lo cual incluye la gestión de las manifestaciones, las delegaciones, las modalidades y las participaciones. Haciendo uso del bundle TCPDF genera reportes nominales y estadísticos, por ejemplo: listado de participantes por manifestación artística, listado de participantes por delegación, listado de participantes con resultados, entre otros.

### **Módulo eventos deportivos**

Gestiona los eventos deportivos lo cual incluye la gestión de los deportes, las delegaciones, las competencias, los equipos deportivos y las participaciones. Haciendo uso del bundle TCPDF genera reportes nominales y estadísticos, por ejemplo: listado de participantes por deporte, listado de participantes por delegación, listado de participantes con resultados, entre otros.

## **2.9 Tarjetas CRC**

Para el diseño de una aplicación, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar utiliza ciertas técnicas como las tarjetas CRC. Las tarjetas CRC son tarjetas de índices, una por cada clase, sobre las cuales se abrevian las responsabilidades de la clase y se anota una lista de los objetos con los que colaboran para desempeñarlas (Larman, 1999). A continuación se presentan 4 de las 31 tarjetas CRC realizadas, el resto se pueden encontrar en el Anexo 5:

Tabla 17 Tarjeta CRC Clase: Evento.

<b>Tarjeta CRC</b>	
<b>Clase: Evento</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Crear Evento Mostrar Eventos Modificar Evento Eliminar Evento Finalizar Evento Definir Evento superior Subir Trabajo de nivel Asignar Trabajo a comisión	Comisión Organizadora Resultado Trabajo Comisión Tribunal

Tabla 18 Tarjeta CRC Clase: Comisión.

<b>Tarjeta CRC</b>	
<b>Clase: Comisión</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Crear Comisión Mostrar Comisión Modificar Comisión Eliminar Comisión Quitar Trabajo de la comisión Evaluar Trabajo	Trabajo Tribunal Resultado Evento

Tabla 19 Tarjeta CRC Clase: Trabajo.

<b>Tarjeta CRC</b>	
<b>Clase: Trabajo</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>

Crear Trabajo Mostrar Trabajo Modificar Trabajo Eliminar Trabajo Cambiar Estado	
---	--

Tabla 20 Tarjeta CRC Clase: Tribunal

<b>Tarjeta CRC</b>	
<b>Clase: Tribunal</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Crear Tribunal Modificar Tribunal Eliminar Tribunal	Comisión

## 2.10 Personas relacionadas con el sistema

Se define como persona relacionada con el sistema, a toda aquella que de una manera u otra interactúe con este, obteniendo un resultado de uno o varios procesos que se ejecutan en el mismo. También son considerados como personas relacionadas con el sistema, aquellas que se encuentran involucradas en dichos procesos, participan en ellos, pero no obtienen ningún resultado de valor. Según su función en el negocio se agrupan por roles, los cuales se describen en la siguiente tabla.

Tabla 21 Personas relacionadas con el sistema.

<b>Personas relacionadas con el sistema</b>	<b>Descripción</b>
<b>Administrador</b>	Cuenta con todos los permisos en el sistema, tiene la función específica de gestionar la seguridad y funcionamiento del sistema, lo que implica la gestión de los usuarios y las trazas. Además tiene la responsabilidad de administrar los nomencladores en el sistema.
<b>Organizador</b>	Posee los permisos de modificación para eventos determinados, en los que el usuario forme parte de la comisión organizadora. Se encarga de la organización de un evento en el sistema.
<b>Tribunal</b>	Tiene la responsabilidad de evaluar los trabajos de las comisiones en las que forme parte del tribunal.

<b>Usuario</b>	No tiene permisos de modificación, solo accede a la información de los eventos registrados en el sistema.
----------------	---

## 2.11 Patrones de Diseño

Los patrones de diseño brindan una solución probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (Larman, 2005). A continuación se muestra cómo se aplicaron estos patrones en la presente investigación.

### Patrones GRASP

Los patrones GRASP<sup>23</sup> comunican los principios fundamentales de la asignación de responsabilidades en el diseño orientado a objetos (Larman, 1999). Entre ellos se tienen:

*Experto:* Se encarga de asignar una responsabilidad al experto en información (la clase que posee la información necesaria para cumplir con la responsabilidad) (Larman, 1999). El uso del patrón en la solución propuesta se evidencia, por ejemplo, en la asignación de la responsabilidad de asignar un trabajo a una comisión. Un trabajo sólo conoce la comisión a la que está asignado, si es que lo está, y una comisión sólo accede a los trabajos que le han sido asignados. En cambio, un evento conoce todos los trabajos, estén asignados o no, además de todas las comisiones. Por esta razón se asigna la responsabilidad a la clase EventoController, por ser un evento el experto en información para cumplir con esta responsabilidad.

*Creador:* Se encarga de asignar a la clase B la responsabilidad de crear una instancia de clase A, si se cumple una de las siguientes condiciones (Larman, 1999):

1. B contiene A.
2. B agrega A.
3. B tiene los datos de inicialización de A.
4. B registra A.
5. B utiliza A muy de cerca.

Se utiliza en las clases controladoras, pues a estas se le da la responsabilidad de crear instancias de las entidades y formularios que sean necesarios para llevar a cabo una acción. Por ejemplo, para la acción de

---

<sup>23</sup> Del inglés General Responsibility Assignment Software Patterns: Patrones generales de software para asignar responsabilidades.

registrar un trabajo, la clase TbTrabajoController crea instancias de TbTrabajo (una entidad) y de TbTrabajoType (un formulario).

*Controlador:* Se encarga de asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente una de las siguientes opciones (Larman, 1999):

1. El negocio o la organización global (un controlador de fachada).
2. El "sistema" global (un controlador de fachada).
3. Un ser animado del dominio que realice el trabajo (un controlador de papeles).
4. Una clase artificial (Fabricación Pura) que represente el caso de uso (un controlador de casos de uso).

Este patrón se pone de manifiesto en las clases cuyo nombre terminan en Controller, pues la arquitectura de Symfony 2 (MVC) brinda una capa específicamente para los controladores, en la que se evidencia la presencia de este patrón.

*Bajo Acoplamiento:* Se encarga de asignar las responsabilidades de modo que se mantenga bajo acoplamiento (Larman, 1999). Se utilizó este patrón con el objetivo de que las clases estuvieran vinculadas solo de forma imprescindible unas con otras y evitar que los cambios locales sean difíciles de realizar, de analizar de forma independiente y de reutilizar. Este patrón se aplica en todas las clases desarrolladas.

*Alta cohesión:* Se encarga de asignar las responsabilidades de modo que se mantenga una alta cohesión (Larman, 1999). Este patrón se utilizó en todas las clases desarrolladas, especialmente en la asignación de las responsabilidades de los distintos controladores. Esto permitió que cada controlador realizara las acciones más afines al negocio que representa, evitando, por ejemplo, que TbTrabajoController se encargue del registro de una comisión en lugar de TbComisionController.

## Conclusiones parciales

A partir de los elementos abordados en el capítulo se dio cumplimiento a la tarea de investigación número 4, lo que se refleja en que:

- La identificación de 148 requisitos funcionales y 14 no funcionales sirvió de base para la redacción de 51 historias de usuario, definiéndose 87 tareas de ingeniería para su implementación.
- La división del sistema en 6 módulos contribuyó a un diseño reutilizable y sencillo de entender e implementar por parte de los programadores.
- La elaboración de 31 tarjetas CRC facilitó la detección temprana de errores en el diseño del sistema y contribuyó a la aplicación del paradigma orientado a objetos y a la definición de nombres para las clases y métodos.
- El uso de los patrones MVC y GRASP proporcionó un criterio objetivo para asignar responsabilidades a las clases y definir las relaciones que se establecen entre ellas, permitiendo un diseño flexible y fácil de mantener.

## CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

### 3.1 Introducción

En el presente capítulo se analizan los aspectos relacionados con la implementación y las pruebas realizadas a la aplicación. Se exponen los elementos fundamentales del modelo de datos y de la integración del sistema propuesto con el resto del ecosistema MAYA. Se describen los métodos y técnicas para la realización de las pruebas con el propósito de validar la propuesta de solución.

### 3.2 Modelo de datos físicos

La solución propuesta cuenta con 49 tablas de las cuales 13 son nomencladores, diferenciando unas de otras a través del prefijo nm (para las tablas que hacen función de nomencladores) y tb (para el resto). Se tuvo en cuenta el patrón de diseño de base de datos, llave subrogada, el cual propone la generación de una llave primaria única para cada entidad en vez de usar un atributo identificador en el contexto dado. Se utilizó el tipo de dato entero en las llaves primarias, lo que permite que las tablas sean más fáciles de consultar por el identificador, dado que se conoce el mismo en cada una de ellas.

La base de datos se encuentra en tercera forma normal ya que todas sus tablas contienen una llave primaria, los atributos son atómicos y no contiene dependencias funcionales transitivas ni parciales. De esta manera se evita la redundancia de los datos y los problemas que puedan ocurrir con las actualizaciones de estos en las tablas, protegiendo así la integridad de los mismos (Sanchez, 2004).

La imagen que se presenta a continuación, muestra un fragmento del modelo de base de datos correspondiente al módulo de eventos científicos, en el cual se encuentran tres tablas y las relaciones que existen entre ellas, el modelo completo puede ser consultado en el Anexo 6:

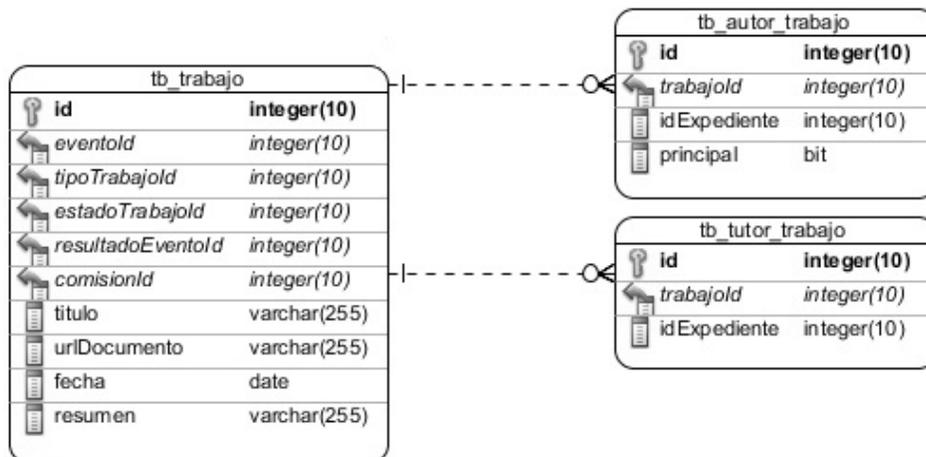


Figura 3 Modelo de datos físicos: fragmento dentro del módulo eventos científicos.

### 3.3 Integración con otros sistemas

Para lograr la integración de la aplicación al ecosistema MAYA se utiliza la arquitectura REST y el protocolo SOAP<sup>24</sup> es empleado en la comunicación con el servidor LDAP de la UCI, a través del cual se logra la autenticación de los usuarios en el sistema.

Roy T. Fielding define *REST* (Fielding, y otros, 2002) como un conjunto coordinado de restricciones arquitectónicas que intenta minimizar la latencia y la comunicación de red mientras que al mismo tiempo maximiza la independencia y escalabilidad de implementaciones de componentes. REST permite el almacenamiento en caché, la reutilización de interacciones y el procesamiento de las acciones por parte de los intermediarios, satisfaciendo así las necesidades de un sistema hipermedia distribuido a escala de Internet.

*LDAP* es un protocolo para el acceso a un servicio de directorio que emplea una estructura de datos similar al que opera sobre TCP/IP<sup>25</sup>. Este es el mecanismo que define la UCI para gestionar su directorio activo.

*Eventos: Sistema para la gestión de eventos estudiantiles*, necesita datos de otros sistemas externos para complementar su funcionamiento. La siguiente figura muestra cómo es su relación con las aplicaciones que integran al ecosistema MAYA, basándose en la arquitectura REST.

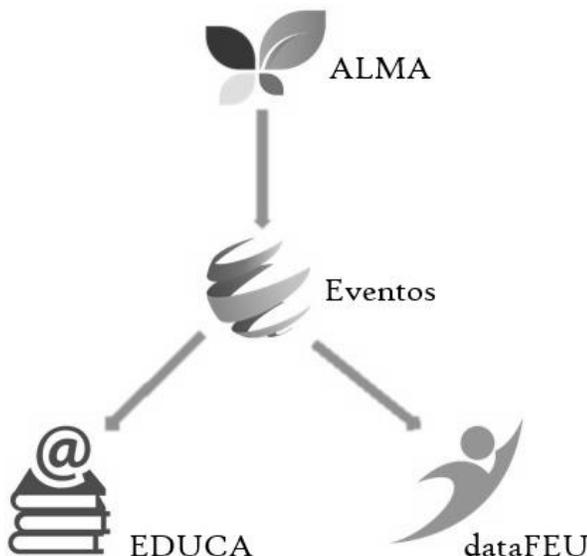


Figura 4 Integración con MAYA.

<sup>24</sup> Del inglés Simple Object Access Protocol: Protocolo Simple de Acceso a Objetos.

<sup>25</sup> Del inglés Transfer Control Protocol / Internet Protocol: Protocolo de Control de Transmisión/Protocolo de Internet.

Las siguientes tablas describen el conjunto de servicios que consume y brinda la solución propuesta:

Tabla 22 Servicios utilizados por Eventos de ALMA, Sistema de Gestión de datos primarios.

Nombre	Uri	Descripción	Salida
<b>Obtenercursoactual</b>	/api/v1/curso/actual.json	Obtiene el curso actual que tiene activo el sistema.	<u>Array con las claves:</u> id fechaInicio fechaFin nombre
<b>Obtenerpersonasdados</b>	/api/v1/personadados/{array}.json	Obtiene las personas dado un conjunto de idExpediente pasado por parámetro. Devuelve los datos generales de la persona, que puede existir o no en el curso actual. Entrada: array de idExpediente	<u>Array de array con las claves:</u> idExpediente usuario pNombre sNombre pApellido sApellido solapin sexo direccion tipoPersona municipio provincia activo En el caso que sea profesor devuelve: cargo <i>En caso de no encontrar datos, retorna false.</i>
<b>Obtenerpersonadadoidexpedienteycurso</b>	/api/v1/personadadoidexpedientes/{idExpediente}/cursos/{curso}.json	Obtiene las personas dado un idExpediente y un curso pasado por parámetro.	<u>Array de array con las claves:</u> idExpediente usuario

		<p>Recomendado para búsquedas donde es necesario consultar los datos que tenía una persona en un curso dado.</p> <p>Entrada: idExpediente y curso</p>	<p>pNombre  sNombre  pApellido  sApellido  solapin  sexo  direccion  tipoPersona  area  municipio  provincia  facultad  foto  activo  curso</p> <p>En caso de que sea un profesor devuelve:</p> <p>cpid  cargo  categoriaDocente  categoriaCientifica</p> <p>En el caso que sea estudiante devuelve:</p> <p>ceid  apto  estadoDocente</p> <p><i>En caso de no encontrar datos, retorna false.</i></p>
--	--	---	---

Tabla 23 Servicios brindados por Eventos.

Nombre	Uri	Descripción	Salida
<b>ObtenerParticipacionEventosdadoldExpedienteyCurso</b>	/api/v1/participacioneventosdadoidexpedientes/{idExpediente}/cursos/{idCurso}.json	Obtiene los trabajos presentados en los diferentes eventos dado el idExpediente y el curso entrado por parámetro. Entrada: idExpediente, idCurso	<u>Array de array con las claves:</u> nombreTrabajo principal tipoTrabajo presentado fechaEvento nombreEvento resultado nivel urlDocumento ( <i>false en caso de que no tenga</i> )  <i>En caso de no tener ninguna participación retorna false.</i>
<b>ObtenerEventosDelSistema</b>	/api/v1/eventos.{_format}	Obtiene todos los eventos resgistrados en el sistema.	<u>Array de array con las claves:</u> Id Nombre Edición  <i>En caso de no tener ninguna evento, retorna false.</i>

### 3.4 Pruebas de Software

Hoy en día, debido al aumento del tamaño y la complejidad del software, el proceso de prueba se ha convertido en una tarea vital dentro del proceso de desarrollo de cualquier sistema (Gutiérrez, y otros, 2004). Los ingenieros de software garantizan la calidad aplicando métodos técnicos sólidos y medidos, llevando a cabo pruebas de software bien planificadas (Pressman, 2010).

El proceso de pruebas de software tiene dos objetivos fundamentales:

1. Demostrar al desarrollador y al cliente que el software satisface sus requisitos.
2. Descubrir defectos en el software en que el comportamiento de este es incorrecto, no deseable o no cumple su especificación (Sommerville, 2005).

Por otra parte, en el diseño de casos de prueba se busca obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se usan dos enfoques diferentes: prueba de caja blanca y prueba de caja negra.

La *prueba de caja blanca* es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de pruebas. Estos últimos aseguran que durante la prueba se han ejecutado, por lo menos una vez, todas las sentencias del programa y que se ejercitan todas las condiciones lógicas (Pressman, 2010).

Las *pruebas de caja negra* son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa. Estas pruebas ignoran intencionalmente la estructura de control y se centra en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba (Pressman, 2010).

Las pruebas no pueden demostrar que el software está libre de defectos o que se comportará en todo momento como está especificado. Siempre es posible que una prueba que se haya pasado por alto pueda descubrir problemas adicionales en el sistema. Generalmente, el objetivo de las pruebas de software es convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su uso operacional. La prueba es un proceso que intenta proporcionar confianza en el software (Sommerville, 2005).

En XP la producción de código está dirigida por las *pruebas unitarias*, las cuales son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial (Letelier Torre, y otros, 2006).

Cualquier característica del programa sin una prueba automatizada simplemente no existe. Los programadores escriben pruebas unitarias para que su confianza en el funcionamiento del programa puede llegar a ser parte del programa en sí. Los clientes escriben *pruebas de aceptación* para que su confianza en el funcionamiento del programa también pueda formar parte del programa. El resultado es un programa que se vuelve más y más confiable con el tiempo: se vuelve más capaz de aceptar el cambio, no menos.

No es necesario escribir una prueba para cada método que se escribe, sólo para los métodos de producción que posiblemente podrían fallar (Beck, 1999).

Los clientes escriben pruebas para cada historia de usuario (pruebas de aceptación), cada escenario que se presenta se convierte en una prueba, en este caso una *prueba funcional* (Beck, 1999). Las pruebas funcionales, en el contexto de Symfony, utilizan un navegador emulado para navegar por las páginas del sitio o aplicación, hacer clic en sus enlaces y rellenar sus formularios para comprobar si los resultados obtenidos son los esperados (Potencier, y otros, 2015). Las pruebas de aceptación son el proceso en el que se prueba el sistema utilizando datos del cliente para verificar que cumple sus necesidades reales, comprobando así que el sistema funciona tal y como se ha especificado (Sommerville, 2005).

Después de la construcción del sistema se realizan las *pruebas del sistema*, estas son usadas para probar el sistema como un todo, comprobando su funcionalidad e integridad, en un entorno lo más parecido posible al entorno final de producción. Entre las pruebas del sistema se encuentran las *pruebas de carga*, las cuales permiten comprobar el rendimiento de un servidor o aplicación cuando el número de clientes o hilos que acceden a él de manera concurrente es elevado. Estas pruebas ayudan a garantizar que el sistema funcionará en las peores circunstancias sin ningún problema crítico. También ayudan a detectar problemas de concurrencia y sincronización no detectados durante el desarrollo del sistema (Gutiérrez, y otros, 2004).

### 3.4.1 Implementación de casos de pruebas

Las pruebas unitarias se utilizan para probar la lógica de negocio que se encuentra en clases que son independientes de Symfony (Potencier, y otros, 2015). Por esta razón, no se implementaron pruebas unitarias para las acciones en los controladores ni para las entidades. Sin embargo, se escribieron pruebas unitarias para clases auxiliares, como por ejemplo para la clase Correo. A continuación se muestra un fragmento del código de la prueba unitaria realizada para esta clase:

```
<?php
namespace AppBundle\Tests\Util;
use PHPUnit\Framework\TestCase;
use AppBundle\Util\Correo;
class CorreoTest extends TestCase
{
    public function testObtenerDireccionCorreo()
    {
        $personal = array(
            'tipoPersona' => 'Estudiante',
            'usuario' => 'ammedina',
            'pNombre' => 'Alejandro Miguel',
            'sNombre' => 'Medina Pichs',
        );
        $resultado_esperado1 = array(
            'to' => 'ammedina@estudiantes.uci.cu',
            'nombre' => 'Alejandro Miguel Medina Pichs'
        );
        $persona2 = array(
```

```

        'tipoPersona' => 'Trabajador',
        'usuario' => 'victor',
        'pNombre' => 'Victor Gabriel',
        'sNombre' => 'González Cardoso',
    );
    $resultado_esperado2 = array(
        'to' => 'victor@uci.cu',
        'nombre' => 'Victor Gabriel González Cardoso'
    );
    $this->assertEquals($resultado_esperado1, Correo::obtenerDireccionCorreo($persona1));
    $this->assertEquals($resultado_esperado2, Correo::obtenerDireccionCorreo($persona2));
}
}

```

Para probar el correcto funcionamiento de las acciones en los controladores y de las entidades se implementaron pruebas funcionales. Las pruebas unitarias y funcionales fueron ejecutadas automáticamente gracias al empleo de la herramienta PHPUnit, lo cual permitió que estas fueran ejecutadas cada vez que fuese necesario, contribuyendo al aumento de la confianza del programador en el código.

Siguiendo las buenas prácticas de Symfony, se implementó una prueba funcional simple que asegurara que las páginas de la aplicación fueran cargadas exitosamente (Potencier, y otros, 2015). Además de esta, se escribieron otras 13 pruebas funcionales para comprobar que las historias de usuario fueron implementadas correctamente.

## Pruebas de Carga

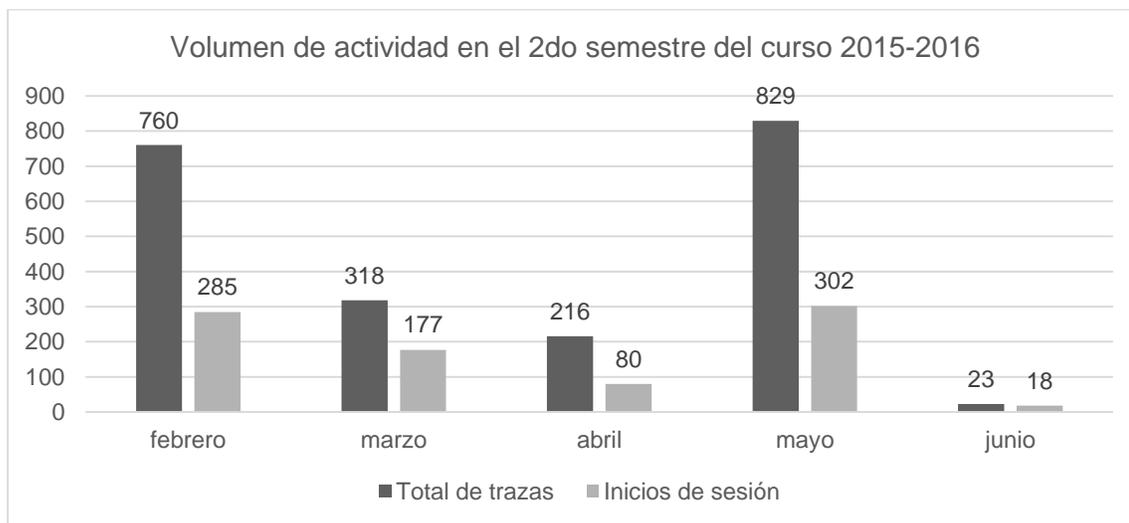
Para la realización de las pruebas de carga sobre el sistema se utilizó la herramienta JMeter. Puesto que el sistema funciona desde un servidor web, una prueba de carga típica será crear un conjunto de hilos que accederán simultáneamente al servidor, simulando de este modo clientes que acceden concurrentemente.

En sistemas web, JMeter simula las principales funcionalidades de un navegador, siendo capaz de manipular resultados en determinada petición y reutilizarlos para ser empleados en una nueva secuencia (Gutiérrez, y otros, 2004). El componente principal de una prueba en JMeter es denominado plan de prueba. Un plan de prueba se compone de una secuencia de componentes que determinarán cómo se simulará la prueba de carga. En estos se definen todos los aspectos relacionados con la prueba, como el número de usuarios a simular, tipo de reportes a generarse con los resultados obtenidos, el uso o no de la memoria caché y cookies, entre otros aspectos.

Dentro de los parámetros a configurar, la cantidad de usuarios a simular es clave. Un número pequeño de usuarios podría hacer que los resultados de la prueba sean irrelevantes, o incluso engañosos, por dar la falsa seguridad de que el sistema soportará la carga a la que será sometido. Por otro lado, un número excesivo de usuarios convertiría la prueba de carga en una prueba de estrés, por lo que no se cumpliría el objetivo propuesto. Dado que este parámetro no debe ser definido arbitrariamente, se procede a estimar la cantidad esperada de usuarios que interactuarán con el sistema de manera simultánea. Para esto, se realiza

un análisis del volumen de actividad en el sistema durante el segundo semestre del curso 2015-2016, a través de las trazas que fueron registradas.

Como se puede apreciar en la Gráfica 1, los meses de mayor actividad en el sistema por parte de los usuarios fueron febrero y mayo. A pesar de ser mayo el mes de mayor actividad, es necesario considerar que los usuarios comenzaron a interactuar con el sistema a partir del 18 de febrero. Esto representa un volumen de actividad en un período de 12 días comparado con el volumen en un mes de 31 días. Es evidente que estos 12 días han sido los de mayor actividad en el sistema, por lo que se procede a analizar cada uno de estos.



Gráfica 1 Volumen de actividad en el 2do semestre del curso 2015-2016.

El día de mayor actividad en el sistema fue el lunes 22 de febrero de 2016, con un total de 176 trazas registradas, de ellas 74 inicios de sesión. La hora pico fue a las 2:00 pm, y el período de 5 horas consecutivas más activo fue de 11:00 am a 3:59 pm (Ver Gráfica 2). Se estima que el tiempo medio de permanencia de los usuarios en el sistema durante ese período fue de una hora y media. La tasa media de arribo al sistema por parte de los usuarios fue de 7,2 usuarios por hora, lo que significa que como promedio un nuevo usuario inició sesión en el sistema cada 8 minutos. Por tanto, el número esperado (o promedio) de usuarios en el sistema, de manera concurrente, se calcula por la Ley de Little (Little, 1961) como sigue:

$$L = \lambda \cdot W$$

Donde:

$L$  = Número esperado de unidades (o usuarios) en el sistema

$W$  = Tiempo promedio de una unidad en el sistema

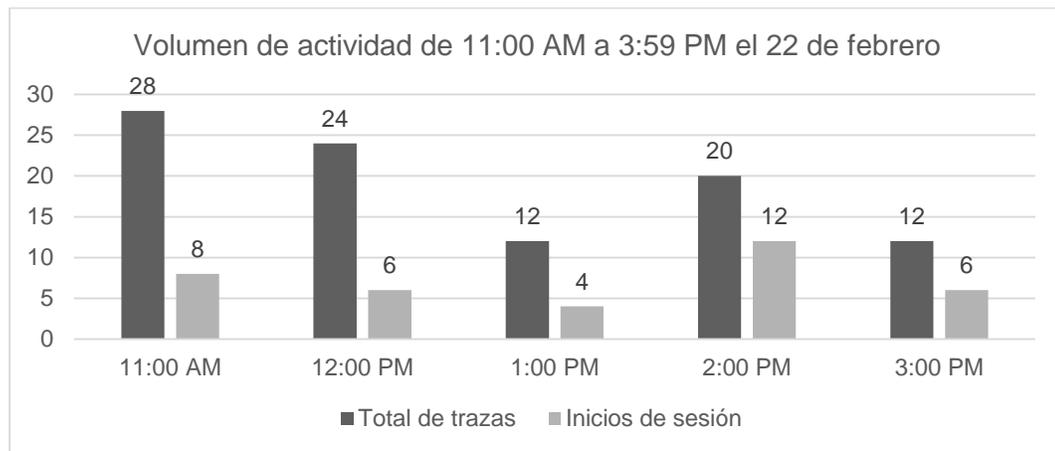
$\lambda$  = Tasa media de arribo al sistema, o como lo explica Little,

$\frac{1}{\lambda}$  = Tiempo esperado entre dos arribos consecutivos al sistema.

Sustituyendo:

$$L = \frac{7,2 u}{h} \cdot 1,5 h$$

$$L = 10,8 u$$



Gráfica 2 Volumen de actividad de 11:00 AM a 3:59 PM el 22 de febrero.

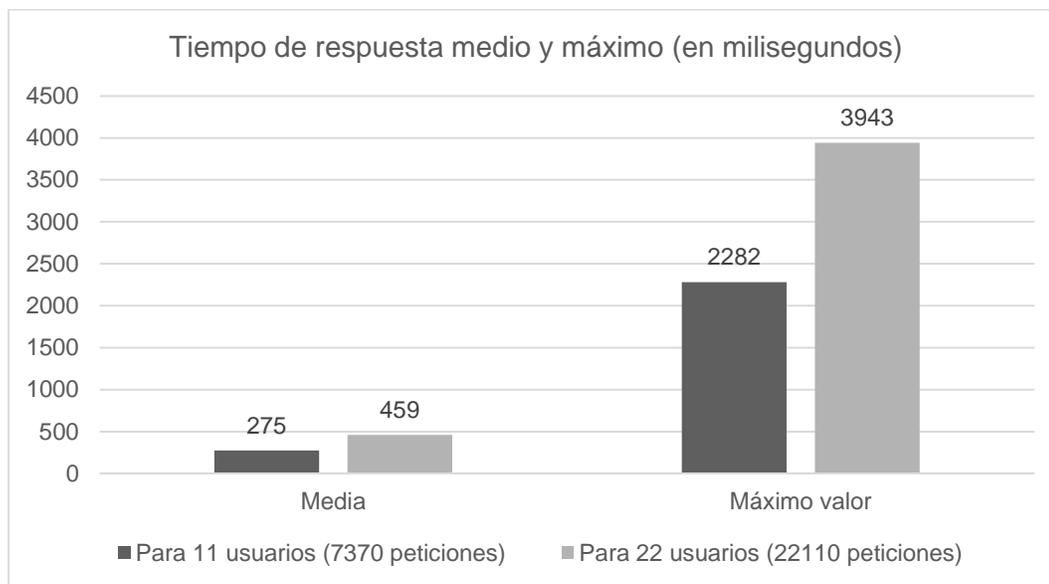
De este modo, se espera alrededor de 11 usuarios en el sistema de manera concurrente en los períodos de mayor volumen de actividad. Este es el caso promedio dentro de los intervalos críticos, pero el sistema debe estar preparado para soportar una mayor cantidad de conexiones concurrentes. Desde la 1:00 PM hasta las 3:59 PM se registraron 22 inicios de sesión, por lo que fue posible la concurrencia de esta cantidad de usuarios ese día.

Por tanto, se decide realizar 2 pruebas de carga cuyos resultados serán comparados. En la primera se establecerá el número de hilos en 11, mientras que en la segunda se elevará a 22. En ambas pruebas se utilizará un período de subida de 1 segundo, lo que define el tiempo en que JMeter distribuirá los hilos. Se fija además el número de iteraciones en 10, lo que contribuirá a la fiabilidad de los cálculos de los promedios y desviaciones.

Una vez aplicada ambas pruebas, se procede a analizar los resultados:

En total se realizaron 22110 peticiones al servidor, de ellas 7370 como parte de la primera prueba, lo cual constituye el tamaño de la muestra para el cálculo de las estadísticas proporcionadas por la herramienta.

Como se puede apreciar en la Gráfica 3, las peticiones tardaron en ser respondidas menos de medio segundo como promedio, y la que más demoró no superó los 4 segundos.

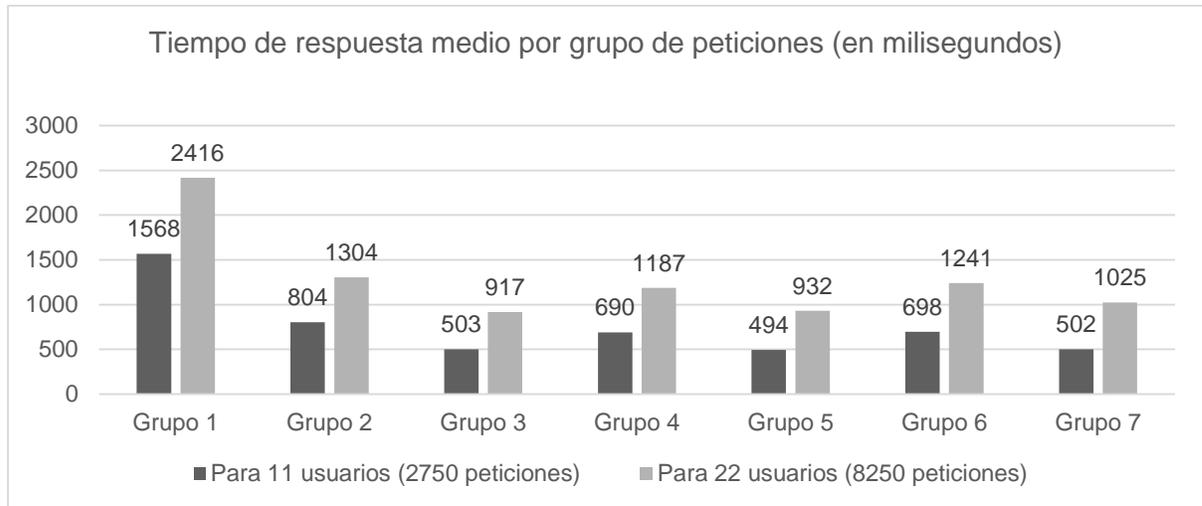


Gráfica 3 Tiempo de respuesta medio y máximo.

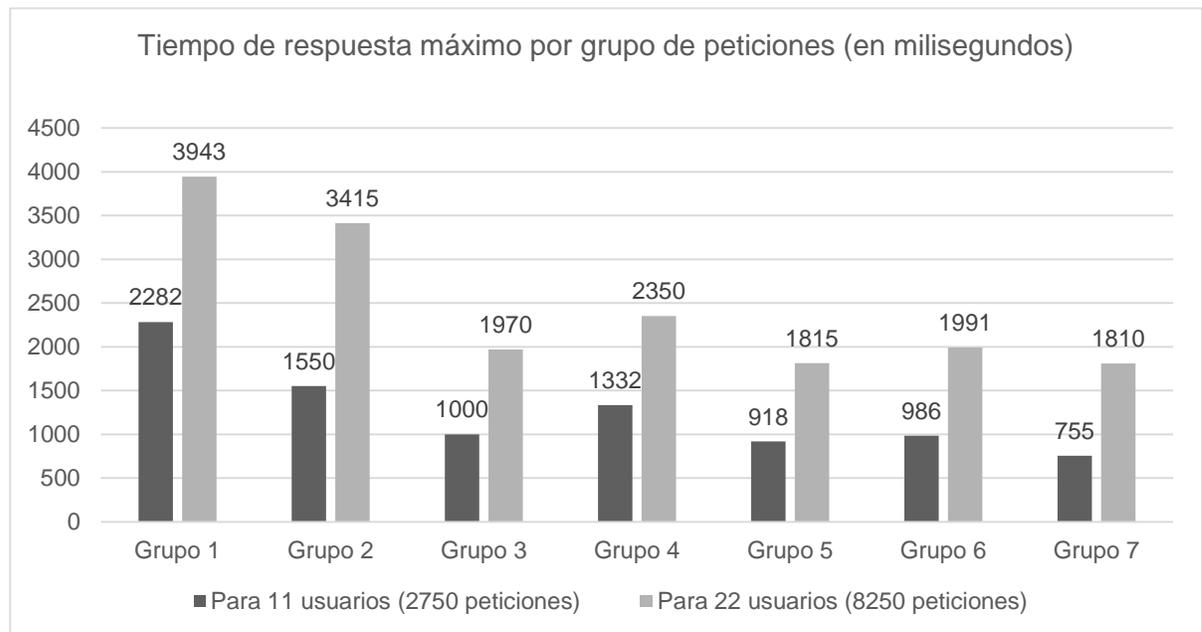
Las peticiones realizadas se pueden agrupar como:

- Grupo 1 - Autenticación en el sistema (incluye la carga de la página principal).
- Grupo 2 - Carga de la foto del usuario autenticado en el sistema.
- Grupo 3 - Carga de una página sin formularios
- Grupo 4 - Carga de una página con formularios
- Grupo 5 - Envío de un formulario, realizando operaciones para insertar o eliminar datos en la Base de Datos.
- Grupo 6 - Realización de operaciones de búsqueda en la Base de Datos.
- Grupo 7 - Cierre de sesión en el sistema (incluye la carga de la página de control de acceso).

Las peticiones de carga de scripts y CSS no se agruparon por tener tiempos de respuesta muy cercanos a cero. A partir de los datos comparados en las gráficas 4 y 5, se puede apreciar que como promedio todas las peticiones fueron atendidas en menos de un segundo y medio, exceptuando las de autenticación en el sistema que se acercaron a los 2,5 segundos. Estas, en conjunto con las asociadas a la carga de la foto del usuario autenticado, fueron las únicas que en sus registros máximos superaron la barrera de los 2,5 segundos para 22 usuarios concurrentes, lo cual es aceptable teniendo en cuenta que un usuario se autentica una sola vez en cada sesión, y la foto es cargada de manera independiente a la carga de la página, lo que permite que el usuario pueda interactuar con el sistema aun cuando la foto no haya sido cargada.



Gráfica 4 Tiempo de respuesta medio por grupo de peticiones.



Gráfica 5 Tiempo de respuesta máximo por grupo de peticiones.

De este modo, se garantiza que el sistema es capaz de responder satisfactoriamente ante la concurrencia de 11 usuarios (cantidad esperada para los períodos de mayor actividad) y de 22 usuarios (número máximo probable para estos períodos).

### 3.4.2 Casos de Pruebas de Aceptación

A continuación se presentan 5 de los 142 casos de pruebas de aceptación, los cuales fueron realizados para la historia de usuario “Gestionar evento”, el resto de los casos de pruebas de aceptación se pueden encontrar en el Anexo 7:

Tabla 24 Caso de Prueba de Aceptación HU1\_P1 para la HU "Gestionar evento".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre:</b> Registrar evento	
<b>Descripción:</b> Prueba la funcionalidad de registrar un evento.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. El usuario debe tener el rol de organizador o administrador.	
<b>Entrada/Pasos de ejecución:</b> El usuario introduce los datos solicitados en el formulario de registro del evento. El usuario selecciona la opción “Registrar”.	
<b>Resultado esperado:</b> El sistema registra un nuevo evento si los datos fueron proporcionados correctamente. En otro caso muestra el mensaje de error correspondiente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 25 Caso de Prueba de Aceptación HU1\_P2 para la HU "Gestionar evento".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre:</b> Mostrar evento	
<b>Descripción:</b> Prueba la funcionalidad de mostrar un evento.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “Detalles del evento”.	
<b>Resultado esperado:</b> El sistema muestra la interfaz correspondiente a los detalles de un evento.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 26 Caso de Prueba de Aceptación HU1\_P3 para la HU "Gestionar evento".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P3	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre:</b> Editar evento	
<b>Descripción:</b> Prueba la funcionalidad de editar un evento.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. El usuario debe tener el rol de organizador o administrador.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “Editar Evento”.	

El usuario modifica los datos proporcionados en el formulario. El usuario selecciona la opción "Modificar".
<b>Resultado esperado:</b> El sistema modifica los datos del evento si los nuevos datos fueron proporcionados correctamente. En otro caso muestra el mensaje de error correspondiente.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla 27 Caso de Prueba de Aceptación HU1\_P4 para la HU "Gestionar evento".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P4	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre:</b> Listar evento	
<b>Descripción:</b> Prueba la funcionalidad de listar un evento.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario debe navegar hacia la interfaz principal.	
<b>Resultado esperado:</b> El sistema muestra un listado de los eventos ordenados cronológicamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 28 Caso de Prueba de Aceptación HU1\_P5 para la HU "Gestionar evento".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P5	<b>Historia de Usuario:</b> No. 1 Gestionar evento
<b>Nombre:</b> Eliminar evento	
<b>Descripción:</b> Prueba la funcionalidad de eliminar un evento.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. El usuario debe tener el rol de organizador o administrador.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción "Eliminar Evento". El usuario selecciona la opción "Sí".	
<b>Resultado esperado:</b> El sistema elimina el evento de la base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

### 3.5 Implantación del sistema

Concluido el desarrollo del software, el mismo fue implantado en la UCI en el curso 2015-2016. El ambiente de la aplicación cumple con estándares de diseño avanzados, con el fin de hacerlo familiar e intuitivo para los usuarios que interactúan con el mismo (teniendo en cuenta los requisitos no funcionales), ya que un porcentaje importante de los usuarios finales no es de la rama de la informática.

La aplicación cuenta con un listado de eventos ordenados cronológicamente en la página principal (Ver Figura 5), con el objetivo de avisar al usuario de los eventos próximos, así como los detalles de los mismos.

Sistema de Gestión de Eventos Estudiantiles bienvenido...

Inicio

Facultad  
2016-10-19 **Fórum de Historia (Facultad 3)**  
La Federación Estudiantil Universitaria y la Unión de Jóvenes Comunistas de la Facultad 3 te convocamos no importa si eres estudiante, profesor o trabajador de la Facultad a participar en la X edición del Fórum de Historia a realizarse el 20 de octubre del 2016, con el objetivo de rescatar la memoria histórica como herramienta y fortaleza del proceso revolucionario cubano.  
Detalles del evento

Universidad  
2016-06-20 **Asamblea de Balance (UCI)**  
Este es el máximo evento del funcionamiento de la FEU de la Universidad  
Detalles del evento

Universidad  
2016-05-25 **Jornada Científica Estudiantil (UCI)**  
La Federación Estudiantil Universitaria en coordinación con la Dirección de Investigaciones convoca a todos los estudiantes de la

Figura 5 Página de inicio.

Toda la información generada sobre la participación de un estudiante en los eventos es concentrada en un expediente digital (Ver Figura 6), recopilando evidencias que le servirán para el proceso final de caracterización en el 5to año de la carrera. La información que se muestra se encuentra en un único formato, lo que permite un procesamiento eficiente.

Mi Perfil Detalles de su participación en eventos estudiantiles...

Inicio

Alejandro Miguel Medina Pichs  
**Usuario:** ammedina  
**Grupo:** 6401  
**Solapín:** E120883  
**Expediente:** EH13043  
**Categoría:** Estudiante

Participaciones en Eventos:

PARTICIPANTE	2
TRIBUNAL	0
ORGANIZADOR	4

Participante Tribunal Organizador

No.	Evento	Descripción	Resultado	Acciones
1	VII Fórum de Historia (Facultad 6)	Ponencia: Martí en Fidel	Relevante	Descargar
2	VII Fórum de Historia (UCI)	Ponencia: Martí en Fidel	-	Descargar

Figura 6 Vista del perfil del estudiante.

Las opciones de la aplicación son seleccionables en el menú superior (Ver Figura 7), donde las funcionalidades están distribuidas de forma lógica.



Figura 7 Vista de detalles del evento.

### 3.6 Validación de la solución

Para validar la aplicación desarrollada se toman en cuenta las variables definidas en el problema de la investigación, estas son: accesibilidad, estandarización de la documentación y persistencia de los datos.

Antes de la implantación del sistema, la accesibilidad de la información sobre la participación en los eventos estudiantiles tenía la limitante de que dicha información estaba distribuida en diversas áreas, dándose el caso de que, para que la participación del estudiante fuera válida en el proceso de caracterización, el mismo debía presentar un certificado de participación, ya que de ninguna otra manera se podía acceder a una fuente de información lo suficientemente confiable que lo confirmara. Con el uso del sistema se facilita el acceso a dichas evidencias, agilizando el proceso de caracterización y evaluación integral.

La estandarización de la documentación era inexistente, pues las áreas que tributaban información al proceso de gestión de los eventos estudiantiles la enviaban en distintos formatos, esto se ha solucionado al establecerse un estándar a partir del uso de formularios definidos para el ingreso de la información.

La persistencia de las evidencias de los estudiantes generalmente se hacía en formato duro, lo que conducía a su posible extravío o deterioro, con el uso de la aplicación estas evidencias se encuentran almacenadas en una base de datos alojada en los servidores UCI evitando con ello los problemas anteriores, y a su vez

permite la creación de un archivo histórico de los eventos estudiantiles, con el objetivo de perfeccionar el proceso de caracterización y evaluación de los estudiantes en el contexto universitario.

Para dar constancia a lo anteriormente expresado, se muestra en el Anexo 8, una carta brindada por parte del consejo FEU de la UCI que demuestra que la aplicación satisfizo las necesidades de la organización en cuanto a la gestión de los eventos estudiantiles, siendo de gran valor y ayuda para el desarrollo de los eventos realizados en la universidad.

## Conclusiones parciales

En el capítulo se expuso los principales resultados obtenidos una vez realizadas las tareas 5, 6 y 7 de la investigación, a partir de lo cual los autores concluyen que:

- El modelo de datos proporcionó una guía para la generación de las entidades y para la especificación de sus relaciones, facilitando la creación de un esquema de base de datos de 49 tablas en tercera forma normal.
- La descripción de la integración del sistema con el resto del ecosistema MAYA permitió definir los servicios web que se necesitaban consumir y los que se debían brindar a través de la arquitectura REST.
- La realización de pruebas unitarias, funcionales y de aceptación permitió comprobar que las funcionalidades fueron implementadas correctamente y que el sistema satisface las necesidades del cliente.
- La realización de dos pruebas de carga permitió comprobar que el sistema soporta satisfactoriamente la concurrencia de 11 y 22 usuarios.

## CONCLUSIONES GENERALES

A través del desarrollo del presente trabajo de diploma se les dio cumplimiento a las tareas de investigación definidas al inicio del mismo, teniendo como principal resultado el Sistema para la gestión de eventos estudiantiles en la Universidad de las Ciencias Informáticas. Luego del análisis de los resultados obtenidos los autores concluyen que:

- El estudio del estado del arte realizado demostró que no se conoce ningún sistema de gestión de información asociada a los eventos estudiantiles a nivel local, nacional e internacional que se adapte a las particularidades de la universidad.
- Las herramientas y tecnologías seleccionadas resultaron ser las adecuadas para la implementación de la solución propuesta, pues se aprovecharon las principales ventajas y potencialidades de cada una de ellas gracias a la abundante y precisa documentación con que cuentan.
- El empleo del patrón arquitectónico MVC permitió tener separados los datos, la interfaz de usuario y la lógica de negocio en tres componentes distintos, facilitando la implementación y mantenimiento del sistema.
- Las pruebas de software realizadas al sistema desarrollado permitieron comprobar que el mismo fue implementado correctamente y que satisface las necesidades del cliente.
- La aplicación desarrollada contribuye al aumento de la accesibilidad de la información, a la estandarización de la documentación y a la persistencia de los datos de los estudiantes asociados a los eventos estudiantiles.

## RECOMENDACIONES

- Generalizar el uso de la aplicación por parte de los usuarios de la UCI, mediante la publicación de las bondades y ventajas que esta posee.
- Desplegar el sistema en el resto de las universidades del país como piloto para contribuir a mejorar los procesos de gestión de los eventos estudiantiles.

## REFERENCIAS BIBLIOGRÁFICAS

- Acebo, Yadira Bagarotti. 2012.** *Gestión de la Calidad en el Ciclo de Desarrollo del Software de proyectos*. Manzanillo. Granma. Cuba. : Revista Granma Ciencia., 2012. Vol. 16. ISSN 1027-975X.
- Activa Impulso Tecnológico. 2016.** Activa Impulso Tecnológico. [En línea] 2016. [Citado el: 3 de noviembre de 2016.] <http://www.activaicon.com/gestion-de-eventos>.
- Amlani, Radhika D . 2012.** Advantages and Limitations of Different SDLC Models. 2012, Vol. 1. Saurashtra University,.
- Babin, Lee. 2007.** *Introducción a AJAX con PHP*. [ed.] ANAYA MULTIMEDIA. 2007. 978-84-415-2200-8 84-415-2200-6.
- Bass, L., P. Clements y R. Kazman,. 1998.** *Software Architecture in Practice*. s.l. : Addison-Wesley, 1998.
- Beck, Kent. 1999.** *Extreme Programming Explained.Embrace Change*. s.l. : Addison-Wesley, 1999. ISBN: 0201616416.
- Bergmann, Sebastian. 2016.** PHPUnit. [En línea] 2016. [Citado el: 18 de mayo de 2017.] [www.phpunit.ed](http://www.phpunit.ed).
- . 2015. *PHPUnit Manual*. 2015.
- Booch, Grady, Rumbaugh, James y Jacobson, Ivar. 2000.** *EL Lenguaje Unificado de Modelado*. 2da edición. 2000.
- Bootstrap. 2016.** Bootstrap. [En línea] 2016. [Citado el: 27 de octubre de 2016.] <http://www.oneskyapp.com/es/docs/bootstrap/>.
- Brigantia. 2016.** Bcongresos: Brigantia congresos. *Software para congresos y eventos*. [En línea] Brigantia Servicios Tecnológicos, 2016. [Citado el: 3 de noviembre de 2016.] <http://www.bcongresos.com/>.
- Bustelo Ruesta, Carlota. 2012.** *Gestión y uso de la información en la empresa* . 2012. ISSN: 1886-6344.
- Bustelo Ruesta, Carlota y Amarilla Iglesias, Raquel. 2001.** *Gestión del conocimiento y gestión de la información*. Inforarea, S.L., España : s.n., 2001. ISSN: 1136-1867.
- Cabrera Isasi, Raydel. 2016.** *EDUCA: Sistema para la gestión de la información de las Estrategias Educativas*. Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2016.
- Curto Díaz, Josep. 2013.** Information Management. [En línea] 2013. [Citado el: 19 de diciembre de 2014.] <http://informationmanagement.wordpress.com/page/2/>.
- Doctrine Team. 2016.** Doctrine Object Relational Mapper. [En línea] 2016. [Citado el: 24 de octubre de 2016.] <http://www.doctrine-project.org/projects/orm.html>.
- Dominguez-Dorado, M. 2015.** *Todo Programación*. Madrid : Iberprensa, 2015. 13679-2004.
- DRAE. 2017.** DLE: Diccionario de la lengua española. [En línea] 2017. [Citado el: 8 de junio de 2017.] <http://dle.rae.es/?id=H9JpZQS>.
- Eventboost SA. 2016.** EventBoost. *EL software de gestión de eventos*. [En línea] Eventboost SA, 2016. [Citado el: 3 de noviembre de 2016.] <http://www.eventboost.com/es-ES/>. CH-501.3.018.972-3, CIF CHE-177.500.574 IVA.

- FAO. 2005.** *Informatización de cooperativas agrícolas: Guía práctica.* Roma : s.n., 2005.
- Fernández Romero, Yenisleidy y Díaz González, Yanette. 2012.** Patrón Modelo-Vista-Controlador. La Habana : s.n., 2012, Vol. 11, págs. 47-57.
- Fielding, Roy T. y Taylor, Richard N. . 2002.** *Principled design of the modern web.* 2002. ACM Trans. Inter. Tech. 2(2):115–150.
- Flanagan, David. 2006.** *The Definitive Guide.* s.l. : s.l. : O'Reilly., 2006.
- Foundation Mozilla Developer. 2016.** HTML. [En línea] 2016. [Citado el: 24 de octubre de 2016.] <https://developer.mozilla.org/es/docs/Web/HTML>.
- Gesintur. 2016.** Gesintur: innovación y gestión. [En línea] 2016. [Citado el: 3 de noviembre de 2016.] <https://www.gesintur.com/software-gestion-de-eventos-y-congresos/>.
- GitHub. 2016.** -Git --fast-version-control. [En línea] 2016. [Citado el: 13 de diciembre de 2016.] <https://git-scm.com/>.
- González Cardoso, Ing. Víctor Gabriel. 2015.** *Fundamentos del proyecto MAYA: Ecosistema de aplicaciones para la evaluación estudiantil en el contexto universitario.* 2015.
- Guevara, Yurisander. 2016.** Cubadebate. *Informatización en Cuba: la ruta estratégica.* [En línea] 11 de septiembre de 2016. [Citado el: 26 de octubre de 2016.] Entrevista a Magda Brito D'Toste, directora de Informatización en el Ministerio de Comunicaciones (Mincom). <http://www.cubadebate.cu/especiales/2016/09/11/informatizacion-en-cuba-la-ruta-estrategica/>.
- Gutiérrez, Javier Jesús, y otros. 2004.** *Pruebas Funcionales y Pruebas de Carga.* s.l. : Departamento de Lenguajes y Sistemas Informáticos, 2004.
- ISO. 2017.** ISO: International Organization for Standardization. [En línea] 2017. [Citado el: 7 de junio de 2017.] <https://www.iso.org/management-system-standards.html>.
- Jeffries, Ron, Anderson, Ann y Hendrickson, Chet. 2000.** *Extreme Programming Installed.* 1ra edición. s.l. : Addison Wesley, 2000. ISBN: 0-201-70842-6.
- JQuery Foundation. 2016.** JQuery. [En línea] 2016. [Citado el: 12 de marzo de 2016.] <https://jquery.com/>.
- Larman, Craig. 1999.** *UML y Patrones: introducción al análisis y diseño orientado a objetos.* México : s.n., 1999. ISBN:970-17-0261-1.
- Larman, Craig. 2005.** *UML y patrones Tomo II.* s.l. : PEARSON, 2005.
- Letelier Torre, Patricio Orlando y Penadés, M<sup>a</sup> Carmen. 2006.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006, Vols. Vol. 5,.
- Little, John D. C. 1961.** *A Proof for the Queuing Formula: L=λW.* s.l. : Operations Research, 1961. Vol. 9.
- López Rodríguez del Rey, María Magdalena, Vázquez, Nancy Hermina y Universidad de las Ciencias Médicas de Cienfuegos, Cuba. 2012.** *El seguimiento al diagnóstico de los estudiantes en los procesos de formación: concepciones para la práctica de la universidad.* Cuba : Medisur, 2012. Vol. 10. ISSN:1727-897X.

- MES. 2015.** Ministerio de Educación Superior de la República de Cuba. [En línea] 2015. [Citado el: 22 de febrero de 2015.] <http://www.mes.edu.cu>.
- Netbeans. 2016.** Netbeans.org. *Página Principal de Netbeans*. [En línea] 2016. [Citado el: 13 de diciembre de 2016.] <https://netbeans.org>.
- Oceguera Martínez, Lic. Susana, y otros. 2009.** *Metodología de la enseñanza de la informática*. Curso 58. La Habana, Cuba : Educación Cubana, 2009. ISBN 978-959-18-0466-2.
- Ortega Martínez, José Antonio. 1997.** *Sistemas de información y ventajas competitivas*. Bilbao : Desclee, 1997.
- PHP. 2016.** PHP. [En línea] 2016. [Citado el: 12 de marzo de 2016.] <http://php.net/manual/es/intro-what-is.php>.
- PostgreSQL. 2016.** PostgreSQL. [En línea] 2016. [Citado el: 14 de diciembre de 2016.] <http://www.postgresql.org/es/>.
- Potencier, Fabien, Weaver, Ryan y Eguiluz, Javier. 2015.** *Official Symfony Best Practices*. 2015.
- Pressman, Roger S. 2010.** *Ingeniería de Software, un enfoque práctico*. 7ma edición. Nueva York : McGraw-Hill Interamericana, 2010. 978-607-15-0314-5.
- Rodríguez Angel, Sandra y Arias Lemos, Reynier. 2016.** *Sistema DataFEU v.2.0*. Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2016.
- Sanchez, Jorge. 2004.** *Principios sobre bases de datos relacionales*. California : s.n., 2004.
- Simon Fraser University Library. 2014.** PKP: Public Knowledge Project. [En línea] 2014. [Citado el: 3 de noviembre de 2016.] <https://pkp.sfu.ca/ocs/>.
- Sommerville, Ian. 2005.** *Ingeniería del Software*. 7ma edición. Madrid, España : Pearson Educación, 2005. ISSN 84-7829-074-5.
- . **2011.** *Software Engineering*. 9na edición. s.l. : Addison-Wesley, 2011. ISBN: 0-13-703515-2.
- Symfony. 2016.** Symfony. [En línea] SensioLabs, 2016. [Citado el: 12 de marzo de 2016.] <http://symfony.com/six-good-reasons>.
- Tejo Delarbre, Raúl . 1996.** *La nueva alfombra mágica*. s.l. : Fundesco, 1996.
- The Apache Software Foundation. 2016.** The Apache Software Foundation. [En línea] 2016. [Citado el: 14 de diciembre de 2016.] <http://www.apache.org/>.
- Twig. 2016.** Twig: The flexible, fast, and secure template engine for PHP. [En línea] SensioLabs, 2016. [Citado el: 14 de diciembre de 2016.] <http://twig.sensiolabs.org/contributors>.
- UCI. 2012.** Universidad de las Ciencias Informáticas. [En línea] UCI, 2012. [Citado el: 14 de marzo de 2017.] <http://www.uci.cu>.
- Ungerboeck Software International. 2016.** USI. [En línea] 2016. [Citado el: 3 de noviembre de 2016.] <https://ungerboeck.com/es/productos-de-eventos/ungerboeck-for-conferences/software-de-gestion-de-eventos>.

**Universidad de las Ciencias Informáticas. 2016.** Gestor de eventos estudiantiles. [En línea] 2016. [Citado el: 11 de marzo de 2016.] <http://eventos.uci.cu/>.

**Universidad de Murcia. 2015.** Eventos Universidad de Murcia. [En línea] 2015. [Citado el: 30 de septiembre de 2015.] <http://eventos.um.es/#1-next>.

**Universidad de Oriente. 2016.** Gestor de Eventos de la Universidad de Oriente. [En línea] 2016. [Citado el: 3 de noviembre de 2016.] <http://www.convenciones.uo.edu.cu/index.php/index/index/index/index>.

**WebFlux. 2012.** CSS3.info. [En línea] 2012. [Citado el: 12 de marzo de 2016.] <http://www.css3.info/>.

## ANEXOS

### Anexo 2: Planes de entrega

*Tabla 29 Plan de entrega del Módulo de eventos deportivos.*

<b>Aplicación</b>	<b>Fin de la 1ra Iteración (05/02/2016)</b>	<b>Fin de la 2da Iteración (25/02/2016)</b>
Sistema para la gestión de los eventos estudiantiles	Funcionalidades con prioridad alta.	Funcionalidades con prioridad media.

*Tabla 30 Plan de entrega del Módulo eventos culturales.*

<b>Aplicación</b>	<b>Fin de la 1ra Iteración (05/06/2016)</b>	<b>Fin de la 2da Iteración (15/09/2016)</b>
Sistema para la gestión de los eventos estudiantiles	Funcionalidades con prioridad alta.	Funcionalidades con prioridad media.

*Tabla 31 Plan de entrega del Módulo copas académicas.*

<b>Aplicación</b>	<b>Fin de la 1ra Iteración (05/11/2016)</b>	<b>Fin de la 2da Iteración (10/12/2016)</b>
Sistema para la gestión de los eventos estudiantiles	Funcionalidades con prioridad alta.	Funcionalidades con prioridad media.

### Anexo 3: Estimación y planificación

Tabla 32 Estimación y planificación del Módulo de eventos deportivos.

Iteraciones	Descripción	Estimación	Duración total
<b>Prioridad: Alta</b>			
<b>1ra</b>	Gestionar competencia de un deporte	0,5	2,5 semanas
	Gestionar equipo deportivo	0,5	
	Gestionar integrante de equipo deportivo	0,5	
	Gestionar deporte en un evento	0,5	
	Gestionar participante de una competencia	0,5	
<b>Prioridad: Media</b>			
<b>2da</b>	Gestionar deporte	0,3	0,3 semanas

Tabla 33 Estimación y planificación del Módulo de eventos culturales.

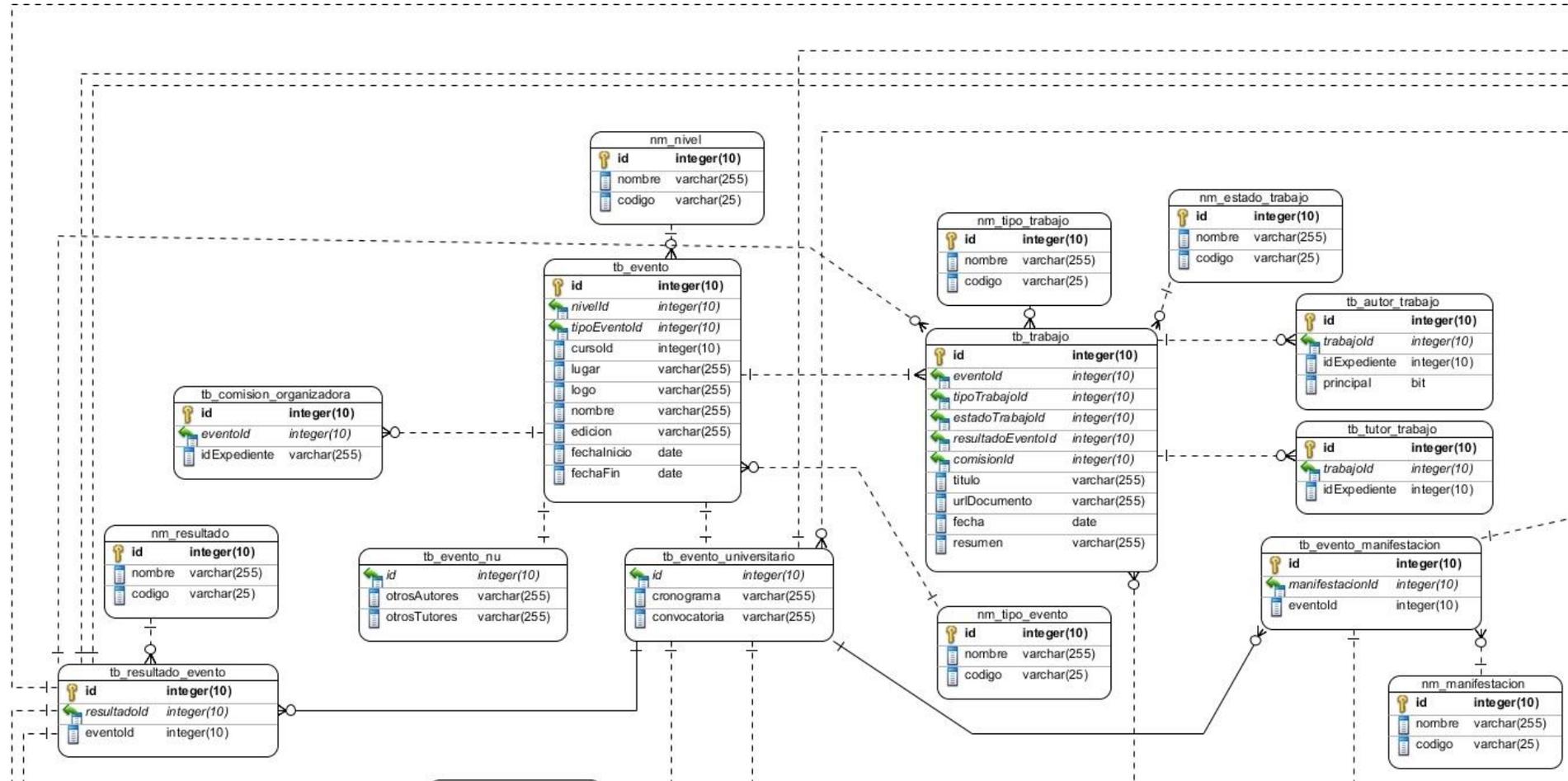
Iteraciones	Descripción	Estimación	Duración total
<b>Prioridad: Alta</b>			
<b>1ra</b>	Gestionar manifestación en un evento	0,5	1,5 semanas
	Gestionar modalidad de una manifestación	0,5	
	Gestionar participante en una modalidad	0,5	
<b>Prioridad: Media</b>			
<b>2da</b>	Gestionar manifestación	0,3	0,3 semanas

Tabla 34 Estimación y planificación del Módulo de copas académicas.

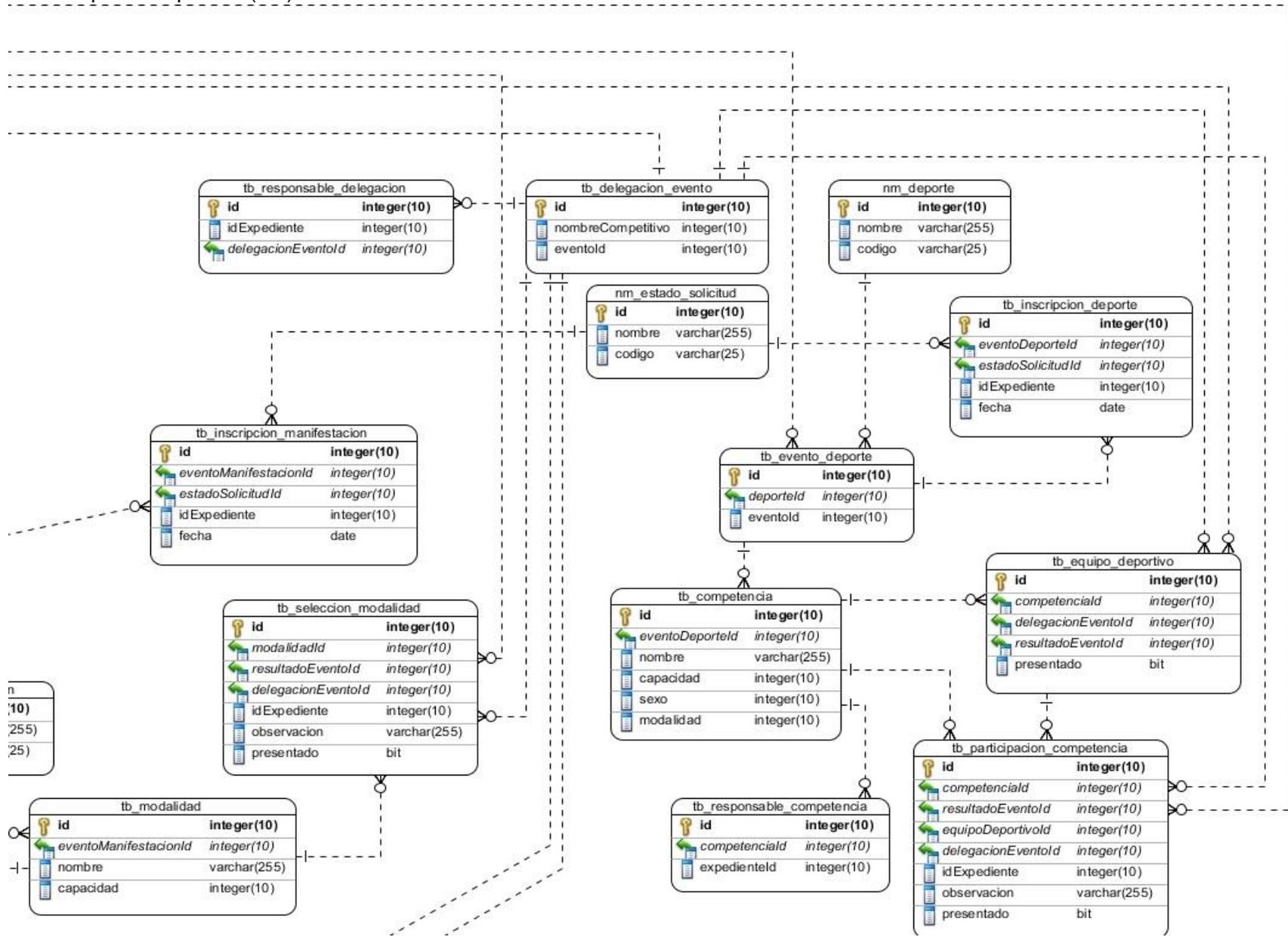
Iteraciones	Descripción	Estimación	Duración total
<b>Prioridad: Alta</b>			
<b>1ra</b>	Gestionar categoría	0,5	2 semanas
	Gestionar equipo en una categoría	0,5	
	Gestionar integrante de un equipo	0,5	
	Gestionar participante de una categoría	0,5	
<b>Prioridad: Media</b>			
<b>2da</b>	Gestionar local	0,3	0,6 semanas
	Gestionar estado del equipo	0,3	

## Anexo 6: Modelo de base de datos

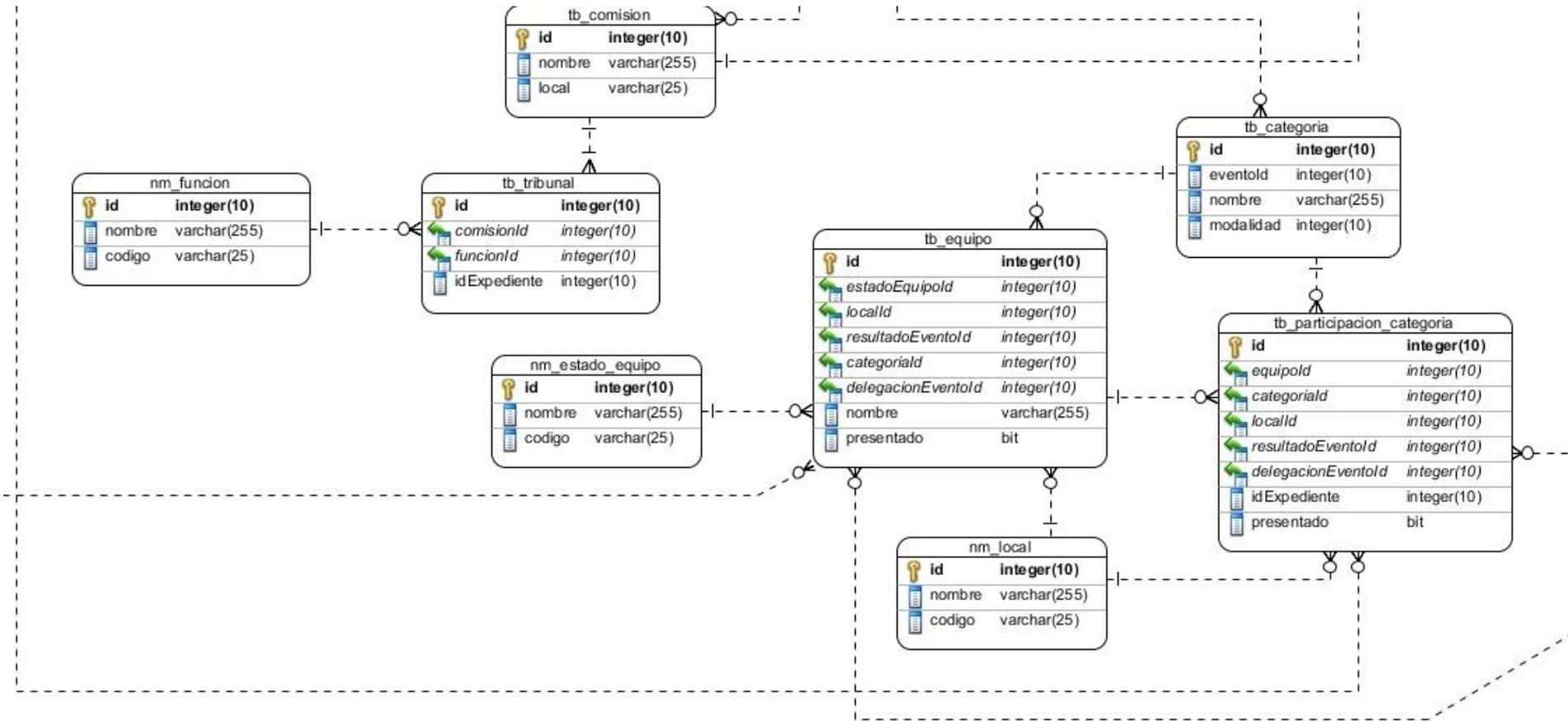
Parte superior izquierda (1/4):



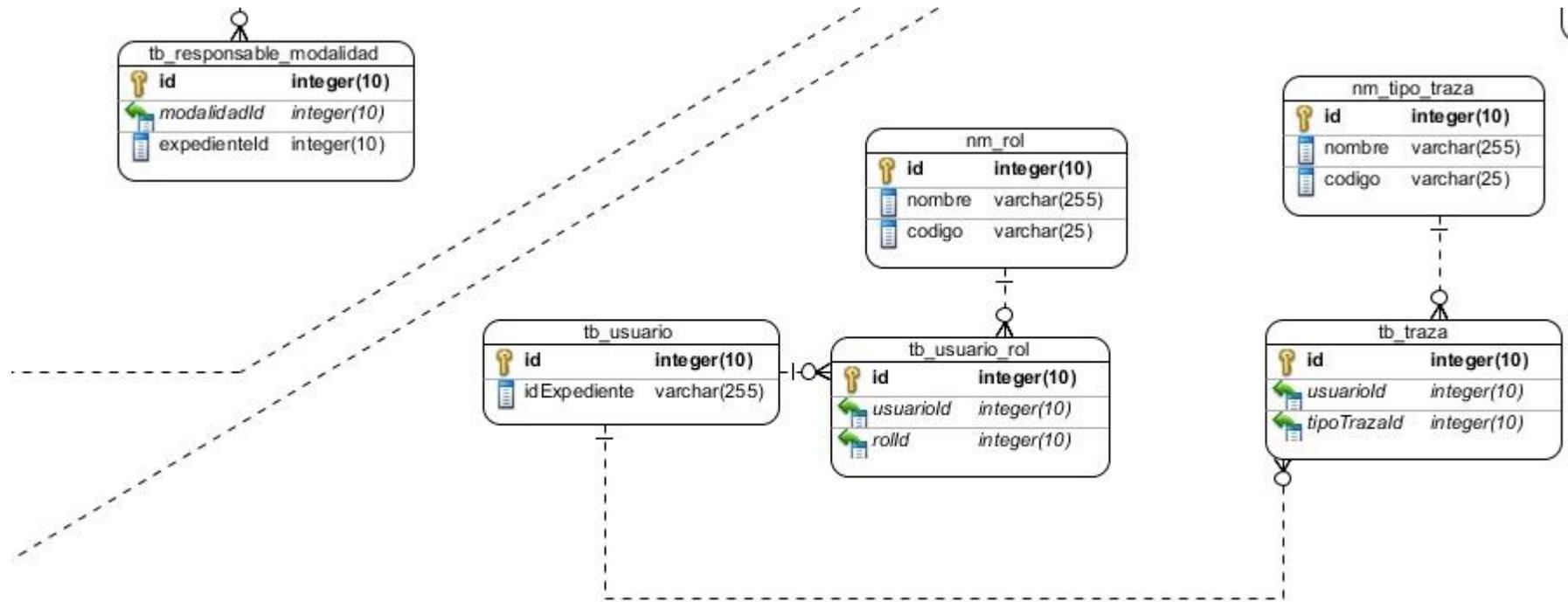
Parte superior izquierda (2/4):



Parte inferior izquierda (3/4):



Parte inferior derecha (4/4):



## Anexo 8: Carta del consejo FEU

La Habana 1ro de mayo del 2017  
"Año 59 de la Revolución"

De: Consejo de la FEU UCI

A: A quien le pueda interesar

Por medio de la presente se hace constar que el Sistema para la Gestión de Eventos Estudiantiles de los estudiantes Alejandro Miguel Medina Pichs y María Teresa Sardañas Rodríguez fue utilizado por la Federación Estudiantil Universitaria de la Universidad de las Ciencias Informáticas para el desarrollo de varios de los eventos que se efectuaron en el curso 2015-2016. El empleo de este sistema sirvió de gran valor y ayuda para el eficiente y eficaz desarrollo de los eventos de la organización.



Consejo de la FEU UCI