

Universidad de las Ciencias Informáticas

Facultad 1



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

Título:

*Módulo para la gestión del plan anual de eventos y
del catálogo de revistas científicas*

Autor:

Adrián Castillo Chávez

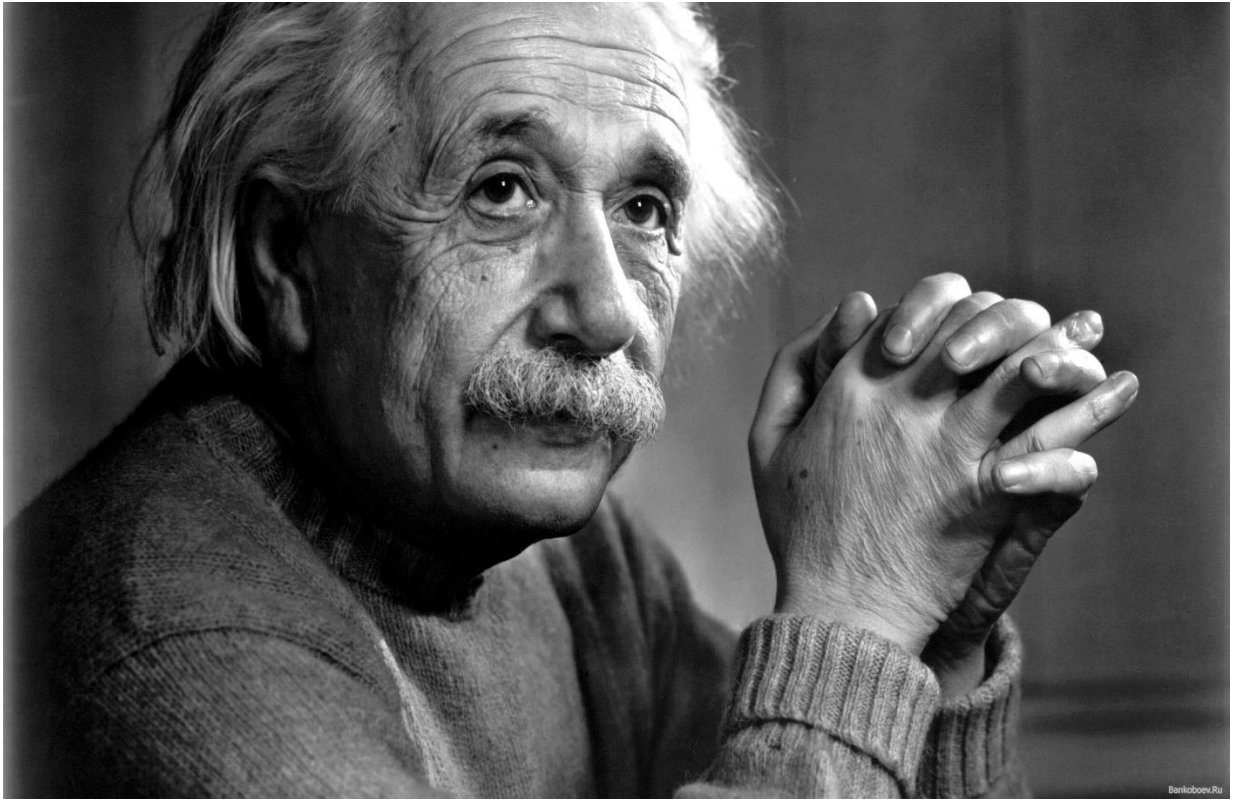
Tutores:

Ing. Geidis Sánchez Michel

MSc. Osiris Pérez Moya

MSc. Delly Lien González Hernández

La Habana, junio 2016.



*Nunca consideres el estudio como una obligación, sino como una
oportunidad para penetrar en el bello y maravilloso mundo del
saber.*

Albert Einstein.

Agradecimientos

A mi mamá por todo el amor, la dedicación y la confianza que me ha brindado en todos los momentos de mi vida. ¡Te quiero mucho!

A mi hermana y a mi sobrina por el cariño que me tienen.

A mi cuñado por cuidar de las mujeres de mi casa el tiempo que he estado estudiando.

A mi familia que siempre me ayudó y me dio su apoyo.

A mi novia por su cariño, comprensión y apoyo.

A mis compañeros de aula por compartir cada momento conmigo durante estos cinco años.

A mis amigos de la universidad por los momentos que compartimos juntos.

A mis profesores de la UCI, por brindarme cada día con paciencia y esmero sus conocimientos que me ayudaron en mi formación.

A mis tutores por su amistad, guía y apoyo.

A mi tribunal por sus constructivas críticas.

Dedicatoria

A mi mamá que ha sacrificado todo por mí y me ha guiado siempre por el mejor camino, que siempre está ahí cuando la necesito, y a la que amo por encima de todo en este mundo.

Muy especial a una persona que hoy no está a mi lado, pero que sé que estaría feliz de ver cómo me convierto en un profesional, a esa persona que me dio un amor infinito y a quién voy a querer siempre, aunque ya no esté.

Para ti papi.

A mi hermana, que después de mi mamá es la mujer que más amo en el mundo, a la que me ha dado su amor y su apoyo incondicional.

A mi sobrina a la que amo con todo mi corazón.

Declaración de autoría

Declaro que soy el único autor de este trabajo de diploma Módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas y le otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Firma del autor:

Adrián Castillo Chávez

Firma del tutor:

Ing. Geidis Sánchez Michel

Firma del tutor:

MSc. Osiris Pérez Moya

Firma del tutor:

MSc. Delly Lien González Hernández

Datos de contacto

Tutor: Ing. Geidis Sánchez Michel.

Graduada de Ingeniera en Ciencias Informáticas en el año 2007, en la Universidad de las Ciencias Informáticas (UCI). Trabaja en la UCI desde ese mismo año. Estudiante de la Maestría en Gestión de Proyectos Informáticos de la UCI. Asistente.

Correo electrónico: gsanchez@uci.cu

Tutor: MSc. Osiris Pérez Moya.

Graduado de Ingeniero en Ciencias Informáticas en el año 2007, en la Universidad de las Ciencias Informáticas (UCI). Trabaja en la UCI desde ese mismo año. Egresado de la Maestría en Gestión de Proyectos Informáticos de la UCI en 2013. Asistente.

Correo electrónico: operez@uci.cu

Tutor: MSc. Delly Lien González Hernández.

Graduada de Licenciada en Cibernética Matemática en el año 1993, en la Universidad Central “Marta Abreu” de Las Villas (UCLV). Trabajó 10 años como investigadora en un centro de Villa Clara y trabaja en la UCI desde el año 2004. Egresada de la Maestría en Estudios Sociales de la Ciencia y la Tecnología de la Universidad de La Habana en 2008. Profesor Auxiliar.

Correo electrónico: delly@uci.cu

Resumen

La presente investigación se centra en el desarrollo de un módulo para la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas, para mejorar la gestión de estos procesos en la Facultad 1 de la Universidad de las Ciencias Informáticas. Se realizó un estudio de fuentes bibliográficas y sistemas sobre la gestión de planes de eventos a nivel internacional, nacional y en el contexto de la UCI. El estudio evidenció la necesidad de desarrollar una nueva solución. Para la implementación, se empleó la metodología SXP, Drupal 7 como sistema gestor de contenido, *Visual Paradigm* 8.0 como herramienta para el modelado y los sistemas gestores de base de datos Postgres SQL 9.3 y MySQL 5. Como resultado se obtuvo un módulo que integra la elaboración y control de los planes anuales de eventos y la promoción de los catálogos de revistas científicas en la Facultad 1. Dicho módulo permite la creación, modificación y eliminación del plan anual de eventos, el plan de misiones y el catálogo de revistas científicas, así como generar reportes sobre el estado del cumplimiento de los planes. Brinda además una funcionalidad que permite generar certificados de participación y de premio en eventos. Finalmente, el módulo se sometió a pruebas funcionales, de carga y estrés, de seguridad y de integración cuyos resultados arrojaron que el módulo funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración. Los errores encontrados durante las pruebas fueron corregidos.

Palabras clave: gestión de eventos, investigación científica, módulo, revistas científicas.

Índice

Introducción	- 1 -
Capítulo I: Fundamentación teórica	- 6 -
1.1. Introducción	- 6 -
1.2. Marco conceptual.....	- 6 -
1.3. Análisis de sistemas homólogos	- 10 -
1.4. Ambiente de desarrollo	- 13 -
1.5. Conclusiones del capítulo.....	- 24 -
Capítulo II: Características y arquitectura del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas	- 25 -
2.1. Introducción	- 25 -
2.2. Descripción de la propuesta de solución	- 25 -
2.3. Modelo conceptual.....	- 26 -
2.4. Requisitos de software.....	- 28 -
2.5. Historias de usuario.....	- 33 -
2.6. Arquitectura.....	- 36 -
2.7. Patrones de diseño.....	- 37 -
2.8. Tarjetas CRC.....	- 38 -
2.9. Modelo de despliegue	- 39 -
2.10. Conclusiones del capítulo	- 41 -
Capítulo III: Construcción y pruebas del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas	- 42 -
3.1. Introducción	- 42 -
3.2. Diagrama de componentes	- 42 -
3.3. Estándares de codificación	- 44 -
3.4. Plan de iteraciones	- 48 -
3.5. Plan de entregas.....	- 50 -
3.6. Tareas de ingeniería	- 50 -
3.7. Validación de la propuesta de solución.....	- 51 -
3.8. Conclusiones del capítulo.....	- 60 -

Conclusiones	- 61 -
Recomendaciones	- 62 -
Referencias bibliográficas	- 63 -
Bibliografía Consultada	- 68 -
Glosario de términos	- 70 -
Acrónimos	- 71 -

Índice de tablas

Tabla 1. Lista de reserva del producto.....	- 28 -
Tabla 2. Resumen de la etapa de captura de los RF.	- 32 -
Tabla 3. HU_Insertar evento.....	- 33 -
Tabla 4. HU_Eliminar revista.	- 35 -
Tabla 5. Tarjeta CRC gesecc.module.	- 39 -
Tabla 6. Tarjeta CRC gesecc.install.	- 39 -
Tabla 7. Plan de iteraciones.	- 49 -
Tabla 8. Plan de entrega.	- 50 -
Tabla 9. Tarea de ingeniería 1.	- 51 -
Tabla 10. Tarea de ingeniería 2.	- 51 -
Tabla 11. Caso de prueba 1.	- 52 -
Tabla 12. Resultado del escaneo de vulnerabilidades para pruebas de seguridad.	- 55 -

Índice de ilustraciones

Figura 1. Fases y flujos de trabajo de SXP	- 17 -
Figura 2. Modelo conceptual.....	- 26 -
Figura 3. Arquitectura de la aplicación.....	- 37 -
Figura 4. Modelo de despliegue.....	- 40 -
Figura 5. Diagrama de componentes.....	- 43 -
Figura 6. Uso de indentación y espacios en blanco.....	- 44 -
Figura 7. Uso de operadores.....	- 45 -
Figura 8. Uso de estructuras de control.....	- 45 -
Figura 9. Uso de llaves en las estructuras de control.....	- 46 -
Figura 10. Longitud de línea con menos de 80 caracteres.....	- 46 -
Figura 11. Longitud de línea con más de 80 caracteres.....	- 46 -
Figura 12. Uso de arreglos.....	- 47 -
Figura 13. Uso de comillas.....	- 47 -
Figura 14. Uso de etiquetas PHP.....	- 47 -
Figura 15. Uso de punto y coma.....	- 48 -
Figura 16. Nombre de funciones.....	- 48 -
Figura 17. Nombre de variables.....	- 48 -
Figura 18. Comportamiento de las no conformidades por iteración.....	- 54 -
Figura 19. Configuración para proteger archivos sensibles.....	- 56 -
Figura 20. Valores obtenidos para una muestra de 25 usuarios.....	- 57 -
Figura 21. Valores obtenidos para una muestra de 50 usuarios.....	- 59 -



Introducción

Desde el surgimiento mismo de la humanidad, el hombre ha sentido siempre la necesidad de conocer todo cuanto le rodea, para lo cual se ha valido en todo momento de su capacidad de pensar, siendo este el principal rasgo que lo diferencia del resto de las especies del reino animal. El perfeccionamiento constante del conocimiento adquirido ha devenido en el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). Las TIC se definen como un conjunto de herramientas y procesos para acceder, recuperar, guardar, organizar, manipular, producir, intercambiar y presentar información por medios electrónicos. Esto incluye hardware, *software* y telecomunicaciones en la forma de computadores y programas tales como aplicaciones multimedia y sistemas de bases de datos (González, y otros, 2015).

La sociedad científica moderna no escapa del espectro de oportunidades que hoy ofrecen las TIC. En la actualidad son grandes los desafíos a los que se enfrenta el conocimiento científico. La gran mayoría de los eventos científicos actualmente son soportados por sistemas informáticos que colaboran en la socialización del conocimiento. La comunidad científica cubana también aprovecha el uso de las TIC y numerosos son los esfuerzos que realizan las universidades cubanas en materia de ciencia, entre ellas, la Universidad de las Ciencias Informáticas (UCI). Desde su creación, la UCI ha tenido como uno de sus objetivos el desarrollo de soluciones informáticas que respondan a las necesidades de informatización de la sociedad y hagan frente a las limitaciones tecnológicas que afectan al país. Los resultados alcanzados en materia de ciencia se corresponden con la labor de investigación científica que hoy se realiza en esta casa de altos estudios.

En la UCI se elabora anualmente un plan de eventos científicos, que se conforma a partir del plan propuesto por cada facultad. Cada facultad elabora su plan atendiendo a sus prioridades y con la identificación previa de cada evento. Estos eventos se divulgan de manera anticipada a través de varios medios de alcance local, municipal, provincial, nacional, regional e internacional. La Facultad 1 de la UCI elabora su plan de eventos científicos iniciando cada año fiscal teniendo como base la propuesta de cada área, dígame los cuatro departamentos docentes: Técnicas de Programación, Ingeniería de *Software*, Ciencias Básicas y Ciencias Sociales y Humanidades y los tres centros de desarrollo de *software*: Centro de Identificación y Seguridad Digital (CISED), Centro de Ideoinformática (CIDI) y Centro de Soluciones Libre (CESOL). A partir de estas propuestas el Vicedecanato de Investigación y Posgrado revisa los eventos científicos previstos, su nivel, calidad y pertinencia, la fecha, la correspondencia con las líneas de investigación priorizadas por



la UCI y las líneas propias de cada área de la Facultad. La información asociada al plan queda recogida en un documento con formato establecido desde 2010. Luego de ser aprobado el plan por la Comisión Científica de la Facultad, se entrega a la Dirección de Investigaciones con la rúbrica del decano. Esto se hace como constancia de su aprobación, para ser incluido en el plan de eventos científicos de la UCI y divulgado por todas las vías posibles en la universidad.

El plan de eventos científicos de la Facultad 1 se corresponde con los indicadores que se miden en los objetivos de la proyección estratégica de la Facultad para cada año. Esto se hace concretamente en el Área de Resultados Clave (ARC) 2 y 3, claustro revolucionario de excelencia e impacto económico y social, respectivamente. El balance de estas ARC y del cumplimiento de los indicadores se hace trimestralmente en marzo, junio, septiembre y diciembre. En la confección del plan de la Facultad intervienen los cuatro jefes de departamentos docentes con sus respectivos gestores para el trabajo de Ciencia e Innovación Tecnológica (CTI), más los tres subdirectores de los centros. Estos últimos a su vez dependen de las coordinaciones con los profesores de sus áreas, para valorar las propuestas de eventos. Esto se hace sobre la base de resultados reales de investigación-desarrollo-innovación que puedan ser expuestos y/o publicados en esos escenarios académicos de diferentes niveles.

Este es un proceso lento, que lleva precisiones y revisiones en varias iteraciones. El mismo se lleva a cabo para estar seguros de que la propuesta es la que mejor integra las diversas líneas y temáticas de investigación, intereses de las áreas de exponer sus resultados en correspondencia con los profesores involucrados en defensas de maestrías, doctorados o procesos de categoría docente. Además, presupone que la participación de todos los involucrados sea rápida, eficiente, eficaz y concreta y no siempre se logra. Por otra parte, como la propuesta se hace en un documento, muchas veces las áreas modifican los datos solicitados y no los entregan en la forma solicitada, o se omiten informaciones importantes. Esto último limita la calidad final de la propuesta del plan anual de eventos científicos. De forma análoga sucede con el control posterior de cada evento planificado que se convierte en un proceso demorado, con insuficiencias en su gestión. Esto ocurre porque se hace totalmente de forma reactiva y no proactiva para garantizar llamadas o alertas previas a la realización de los eventos.

Otra de las actividades importantes del vicedecanato es la promoción de revistas científicas cuyas temáticas son afines a las líneas de investigación de las áreas de la Facultad 1. Esta promoción se realiza para facilitar



la identificación de fuentes para publicar los trabajos de investigación en forma de artículos, en revistas referenciadas. Es así que desde la Facultad 1 y sus siete áreas, se conforma un catálogo de revistas que cumplan los requisitos previstos por el Ministerio de Educación Superior (MES) para la evaluación de las publicaciones en los cuatro niveles reconocidos. Este catálogo además se corresponde con las líneas de investigación de los departamentos docentes y centros. El mismo se divulga o promueve por el Vicedecanato de Investigación y Posgrado con la participación de gestores, jefes de departamentos, subdirectores y metodóloga. El catálogo se escribe en un documento, no siempre con formato estándar y por tanto en ocasiones el área que hace nuevas propuestas las hace de formas diferentes a las acordadas. El control de la promoción del catálogo adolece de las mismas insuficiencias planteadas del plan de eventos científicos. A esto se le adiciona que el catálogo se genera sin un sistema de alertas tempranas que recuerden las fechas límite de envío de artículos o contribuciones a las revistas. Esto impide garantizar que los autores estén avisados previamente y no se venzan los plazos establecidos.

Por tanto, el **problema de investigación** queda definido de la siguiente forma: ¿Cómo integrar la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas para mejorar la gestión de estos procesos en la Facultad 1 de la UCI?

El presente trabajo tiene como **objeto de estudio**: el proceso de gestión de la información relacionado con el plan anual de eventos y el catálogo de revistas científicas.

Se incidirá específicamente en el proceso de gestión de la información del plan anual de eventos y el catálogo de revistas científicas en la Facultad 1 de la UCI, como **campo de acción**.

El **objetivo general** del presente trabajo es desarrollar un módulo que integre la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas utilizando Drupal 7 para mejorar la gestión de estos procesos en la Facultad 1 de la UCI.

Los **objetivos específicos** quedan desglosados de la siguiente manera:

1. Elaborar el marco teórico-conceptual que sustente la investigación relacionado con el desarrollo de herramientas para la gestión de planes anuales de eventos y la promoción de revistas científicas.



2. Analizar el estado de la gestión de la elaboración y control del plan anual de eventos y de la promoción del catálogo de revistas científicas (Facultad 1, UCI).
3. Diseñar las funcionalidades del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas (Facultad 1, UCI).
4. Implementar las funcionalidades del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas (Facultad 1, UCI).
5. Realizar pruebas a las funcionalidades del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas (Facultad 1, UCI).

La presente investigación está sustentada por la siguiente **idea a defender**: El desarrollo de un módulo que integre la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas utilizando Drupal 7, contribuirá a la mejora de estos procesos en la Facultad 1 de la UCI.

Se proponen las siguientes tareas para dar solución a los objetivos propuestos anteriormente:

1. Realización de un estudio sobre las tendencias en la gestión de planes de eventos y catálogos de revistas científicas.
2. Selección del ambiente de desarrollo que se necesita para implementar la propuesta de solución.
3. Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
4. Implementación de la propuesta de solución.
5. Validación de la propuesta de solución a partir de la ejecución de pruebas.

Se emplean para el desarrollo de la presente investigación métodos científicos que se describen a continuación:

Métodos teóricos

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

Analítico-Sintético: permitió analizar las soluciones existentes que gestionan la información de los planes anuales de eventos y los catálogos de revistas científicas, tanto a nivel nacional como internacional.



Modelación: se utilizó como herramienta para comprender el problema y crear un modelo abstracto del mismo para el diseño de la solución.

Métodos empíricos

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

Entrevistas: se utilizó para obtener a través del personal que labora en el Vicedecanato de Investigación y Posgrado la información referente a como se llevan a cabo los procesos de elaboración y control del plan anual de eventos y la promoción de revistas científicas, garantizando así que la solución propuesta cumpla las necesidades existentes.

El presente trabajo está estructurado en tres capítulos distribuidos de la siguiente forma:

Capítulo 1. Fundamentación teórica: En este capítulo se abarca el estado de los sistemas que guardan relación con la presente investigación a nivel nacional e internacional, para realizar una selección del ambiente de desarrollo a utilizar.

Capítulo 2. Características y arquitectura del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas: En este capítulo se expone la propuesta de solución de manera detallada, describiéndose los requisitos funcionales y no funcionales, así como todo lo referente a la arquitectura del módulo y a los patrones de diseño utilizados.

Capítulo 3. Construcción y pruebas del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas: En este capítulo se detalla la construcción y validación de la propuesta de solución al problema planteado. Se realizan además y describen las pruebas de *software* a la aplicación.

El presente trabajo contiene además **Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía consultada, Glosario de términos, Acrónimos y Anexos.**



Capítulo I: Fundamentación teórica

1.1. Introducción

En el siguiente capítulo se hace un estudio del estado del arte relacionado con soluciones informáticas existentes para la gestión de eventos científicos, se abordan conceptos que están relacionados con los procesos que integran el campo de acción y se describe el ambiente de desarrollo a utilizar para el desarrollo de la propuesta de solución.

1.2. Marco conceptual

En esta sección se explican algunos aspectos fundamentales para la comprensión del contexto de la investigación.

Plan: Según la Real Academia Española es el modelo sistemático de una actuación pública o privada, que se elabora anticipadamente para dirigirla y encauzarla (Real Academia Española, 2016).

Según García (2005), en su forma más simple el concepto de plan se define como la intención y proyecto de hacer algo, ó como proyecto que, a partir del conocimiento de las magnitudes de una economía, pretende establecer determinados objetivos. Asimismo, se ha definido como un documento en que se constan las cosas que se pretenden hacer y forma en que se piensa llevarlas a cabo.

Por otra parte, Blake (2007), define el plan como el documento rector, producto del proceso de planeación. Consiste en el conjunto coordinado de objetivos, metas y acciones que relacionadas con las estrategias y programas jerarquizan una serie de políticas e instrumentos en el tiempo y el espacio, para alcanzar una imagen objetiva propuesta.

Luego de haber estudiado los conceptos emitidos por otros autores, el autor de la presente investigación define un nuevo concepto de plan, definiéndolo como el documento donde queda recogido un conjunto coordinado de acciones para alcanzar un objetivo propuesto.



Catálogo: Según la Real Academia Española es la relación ordenada en la que se incluyen o describen de forma individual libros, documentos, personas, objetos, entre otros, que están relacionados entre sí (Real Academia Española, 2016).

Martínez de Sousa (2002), define el catálogo como el “conjunto de descripciones, de acuerdo con unas normas, de los asientos bibliográficos de los libros y documentos de una biblioteca, con la indicación del lugar que ocupa en ella”.

Según la Asociación de Bibliotecas de los Estados Unidos (*American Library Association, ALA*) es un fichero de registros bibliográficos, creado conforme a principios específicos y uniformes de realización y siguiendo las directrices de una lista oficial de encabezamientos que describe el material de la colección, biblioteca o grupo de éstas (Asociación de Bibliotecas de los Estados Unidos, 2016).

A partir de haber estudiado las definiciones dadas por otros autores, el autor de la presente investigación asume el concepto de catálogo emitido por la Asociación de Bibliotecas de los Estados Unidos.

Gestión: Según la Real Academia Española es la administración, organización y funcionamiento de una empresa, actividad económica u organismo (Real Academia Española, 2016).

Según Ponjuán (2003), es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización.

Salgueiro (2001), llama gestión a aquellas acciones relacionadas con actividades que están dirigidas a la realización y desarrollo de objetivos que han debido establecerse con anterioridad.

Luego de haber estudiado los conceptos de gestión dados por otros autores, en la presente investigación, el autor decide adoptar el concepto de gestión dado por Ponjuán.

Gestión de información: En el *Harrod's Librarian's Glossary*, gestión de la información se define como un "término impreciso" que sirve para designar un conjunto de actividades orientadas a la generación, coordinación, almacenamiento o conservación, búsqueda y recuperación de la información tanto interna como externa contenida en cualquier soporte (Prytherch, 2000).



Según Ponjuán (2003), es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico, la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización. En particular, también se desarrolla en unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información.

A partir de haber estudiado los conceptos dados por otros autores, el autor de la presente investigación decide elaborar un nuevo concepto de gestión de información, definiéndolo como el proceso de organizar, evaluar y presentar información en un contexto determinado, controlando la veracidad, oportunidad y utilidad de la información, así como su disponibilidad en el momento en que dicha información sea requerida.

Evento científico: Según el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela se consideran eventos científicos las reuniones nacionales e internacionales tales como: congresos, seminarios, conferencias, foros, jornadas y simposios destinados a la divulgación e intercambio de resultados originales de investigación (Nicolás, 2015).

Un evento científico es una actividad para establecer un mecanismo que ayude a la generación, fortalecimiento y consolidación del conocimiento científico en los diferentes sectores de la entidad. Es un encuentro donde se estimula el conocimiento adquirido y se presenta información acerca de un tema específico. De esta manera los investigadores enriquecen su formación profesional y aumentan su capacidad investigativa a través del intercambio entre ellos y con los desarrolladores (Cardoso Said, y otros, 2013).

Los eventos científicos, son un medio para consolidar espacios reflexivos, críticos y comprometidos en la discusión de los más variados temas relacionados con la agenda social, política y económica nacional e internacional, resultado de procesos académicos, de sistematización de experiencias y de investigación (Ortigoza, y otros, 2012).

Luego de haber estudiado los conceptos dados por otros autores, el autor de la presente investigación decide elaborar una definición de evento científico. Un evento científico es toda actividad académica efectuada por una o más instituciones dedicadas al fomento de la investigación científica, con el objetivo de



consolidar espacios de discusión de los más variados temas como resultado de sistematización de experiencias, procesos académicos y de investigación.

Revista científica: Según la Asociación de Bibliotecas de los Estados Unidos (*American Library Association*, ALA) es una publicación periódica que publica artículos científicos y/o información de actualidad sobre investigación y desarrollo acerca de un campo científico determinado (Asociación de Bibliotecas de los Estados Unidos, 2016).

Según la Organización Internacional de Estandarización (*International Organization for Standardization*, ISO) es una publicación en serie que trata generalmente de una o más materias específicas y contiene información general o información científica y técnica (Organización Internacional de Estandarización , 2016).

En la presente investigación a partir de haber estudiado los conceptos dados por otros autores se define revista científica como una publicación seriada sobre determinadas materias científicas y/o de actualidad sobre investigación y desarrollo.

Plan de eventos: En la presente investigación, el autor define el plan de eventos como un modelo que se elabora desde la dirección de la Facultad 1 de la UCI, en correspondencia con los indicadores que se miden en los objetivos de la proyección estratégica para cada año. Este plan contiene una serie de eventos sobre la base de resultados reales de investigación-desarrollo-innovación que puedan ser expuestos y/o publicados en escenarios científicos-académicos de diferentes niveles.

Catálogo de revistas científicas: En la presente investigación, el autor define el catálogo de revistas científicas como una colección seleccionada de revistas científicas cuyas temáticas son afines a las líneas de investigación de las áreas de la Facultad 1 de la UCI. Este catálogo tiene como objetivo facilitar la identificación de fuentes para publicar los trabajos de investigación en forma de artículos, en revistas referenciadas.



1.3. Análisis de sistemas homólogos

Con el objetivo de darle visibilidad a los eventos, se han desarrollado disímiles sistemas que promocionen y gestionen actividades científicas. En este sentido, no han sido pocas las instituciones a nivel nacional e internacional, que han realizado grandes esfuerzos con el fin de informatizar estas actividades. Por consiguiente, para desarrollar un nuevo sistema se hace necesario estudiar los antecedentes al mismo. A continuación, se muestra un estudio de una serie de sistemas que guardan relación con la presente investigación.

1.3.1. Open Conference Systems

Open Conference Systems (OCS) es un producto de la universidad de Columbia Británica, que está desarrollado por el proyecto *Public Knowledge Project*¹ (por sus siglas en inglés, PKP) en su versión dos. Constituye una herramienta de publicación *web* gratuita que provee apoyo tecnológico a la organización de conferencias o eventos en general.

El OCS permite la creación de sitios *web* de una o varias conferencias simultáneas y en varios idiomas, así como envíos de trabajos. Además, permite la redacción y envío de convocatorias de ponencias, ya sea por anuncios en el sitio o envío de correos masivos mediante la plataforma. También posibilita monitorear y controlar el proceso de revisión de los trabajos; recibir y aceptar, en línea, las propuestas de ponencias y su resumen. Asimismo, brinda la posibilidad de publicar los trabajos aprobados e incluso manejar comentarios o discusiones pasado el evento. Realizar búsquedas de trabajos por diferentes criterios, dígame: nombre, autor y tema, efectuar registros y pago en línea. Permite generar informes y estadísticas del evento: número de inscritos, trabajos aprobados y rechazados, listado de participantes, ponentes e integrar debates en línea después de una conferencia.

OCS es un producto de código abierto² que está licenciado por la GNU *General Public License*³ (GNU GPL). Está desarrollado completamente en lenguaje de programación PHP y puede utilizar base de datos MySQL

¹ Se dedica a mejorar la calidad académica y pública de la investigación. Proporciona el *software* de código abierto para publicación y gestión de conferencias y publicaciones.

² Código abierto: en inglés, *Open Source*. Es el término con el que se conoce al *software* distribuido y desarrollado libremente.

³ GNU GPL, en español Licencia Pública General de GNU, creada originalmente por Richard Stallman, la cual garantiza la libertad de compartir y modificar todas las versiones de un programa.



o PostgreSQL (Fraser, 2014). Además de que el sistema estudiado presenta una interfaz de usuario poco clara y compleja, puesto que está diseñado para la organización de eventos de los cuales se maneja numerosa información (Brengei, 2012).

Este sistema posee gran número de funcionalidades, algunas de las cuales sirven de base a la presente investigación como son: Realizar búsquedas por diferentes criterios, permite la redacción y envío de convocatorias de ponencias, ya sea por anuncios en el sitio o envío de correos masivos mediante la plataforma. La limitante principal de este sistema es que se centra más en la organización de los eventos científicos y no en la confección de un documento para la planificación de una serie de eventos de los cuales quede recogida información referente al nivel y la calidad de los eventos, fecha de los eventos, fecha límite de envío de ponencias y fecha de notificación de aceptación de las ponencias.

1.3.2. Sistema de gestión para el control de actividades científicas de los docentes

El Sistema de Gestión para el Control de Actividades Científicas de los Docentes (SGCACD), desarrollado en la Facultad de Ciencias Técnicas (FCT) perteneciente a la Universidad de Las Tunas (ULT) Vladimir Ilich Lenin, permite el control y seguimiento de las actividades científicas que se desarrollan en los departamentos docentes de la FCT. El SGCACD fue diseñado utilizando el marco de trabajo *CodeIgniter* y el lenguaje de programación PHP, como gestor de base de datos utilizan PostgreSQL y como metodologías de desarrollo *Rational Unified Process*⁴ (RUP) con modelado ágil.

El sistema desarrollado permite gestionar las actividades científicas, actualizar estados de publicación y elaboración de trabajos, actualizar tipos de trabajo y tipos de documentos, registrar premios obtenidos, asignar premios y actividades. Además, permite realizar búsquedas de trabajos por diferentes criterios y adjuntar el resumen de los trabajos a presentar. Actualmente, este sistema se encuentra en desarrollo de nuevas versiones (Rosabal, 2011).

Este *software* posee características reutilizables para el desarrollo de la solución que se pretende obtener, sirviendo de apoyo para la propuesta de solución y utilizando como base algunas de las funcionalidades anteriormente mencionadas, como son: actualizar estados y realizar búsquedas por diferentes criterios. La

⁴ Traducción del español: Proceso Unificado de *Rational*. Es un proceso de desarrollo de *software* que define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo.



limitante que presenta y por lo que se descarta dicha solución es que este sistema ha sido solamente diseñado para gestionar las actividades científicas de la FCT, enmarcándose solamente en el proceso de gestionar los trabajos de los profesores, los premios obtenidos por los trabajos, así como las actividades científicas a realizarse durante el curso, por lo que no satisface todas las necesidades funcionales del presente trabajo de diploma.

1.3.3. Plataforma electrónica para la gestión de eventos científicos en la UCI

Esta plataforma electrónica es un sistema multi-site⁵ encargada de gestionar los eventos científicos, específicamente los desarrollados en la UCI. Posee una interfaz visualmente atractiva y es de fácil configuración. Diseñado específicamente para optimizar el tiempo y el esfuerzo en el proceso de gestión de eventos científicos.

El proceso de desarrollo fue guiado por la metodología ágil XP. Para el desarrollo del sistema se utilizaron herramientas como CMS Drupal en su versión 7.12 y MySQL en su versión 5.2.17, siendo este el gestor de base de datos que recomienda Drupal (Pérez Reyes, y otros, 2012). Este sistema permite: gestionar usuario, encuestas, ponencias, convocatorias, talleres, roles y permisos. Además, el envío de notificaciones automáticas vía correo electrónico y hacer búsquedas generales.

El estudio de este sistema permitió identificar algunas funcionalidades que sirven de base a la presente investigación puesto que permite el envío de notificaciones automáticas y la realización de búsquedas generales. Como limitante fundamental se encontró que este sistema fue diseñado específicamente para la gestión de eventos científicos de la UCI, en el período de tiempo en que estos eventos se desarrollan, no cumpliendo con la necesidad de la presente investigación de generar planes de eventos con información sobre fecha del evento, fecha límite de envío de ponencias y fecha de notificación de aceptación de las ponencias, así como el nivel y calidad de los eventos.

1.3.4. Conclusiones del análisis de sistemas homólogos

⁵ Permite tener varios sitios, bajo el mismo árbol, funcionando con el mismo motor.



El estudio realizado evidencia que algunos de los sistemas anteriormente mencionados se han desarrollado para dar solución a determinadas instituciones, por lo cual no se encuentran disponibles de forma libre para ser adaptados y utilizados por otros centros. En el contexto de la UCI se han desarrollado varias soluciones en este sentido, pero han sido de alcance local o solo informatizan de forma aislada el proceso gestión de eventos científicos. Dado que es imposible aprovechar alguna solución existente se propone la implementación de una nueva solución que integre la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas para mejorar la gestión de estos procesos en la Facultad 1 de la UCI.

1.4. Ambiente de desarrollo

Analizar las características particulares de las herramientas, así como las ventajas que reportaría el uso de cada una, son elementos importantes a tener en cuenta en la selección. Para el desarrollo de la solución se utilizan un conjunto de tecnologías y herramientas establecidas en el Vicedecanato de Investigación y Posgrado de la Facultad 1 de la UCI, que deben ser utilizadas en el desarrollo de sistemas de información basados en tecnologías *web*. Estas cumplen además con las políticas de *software* libre del país para lograr la soberanía tecnológica.

1.4.1. Sistemas de gestión de contenido

En los últimos años se ha desarrollado considerablemente el concepto de los sistemas de gestión de contenidos (CMS), *software* que se utiliza principalmente para la creación y gestión de sitios *web* ya sea en internet como en una intranet. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores proporcionando un entorno que posibilita la actualización, mantenimiento y ampliación del portal *web* con la colaboración de múltiples usuarios (Paulsen, 2011).

Existen CMS de código abierto y de código privativo. Los CMS de código abierto brindan más libertad a los desarrolladores de portales *web* en cuanto a modificación y adaptabilidad con el fin de lograr la funcionalidad deseada. Entre estos CMS de código abierto los más utilizados a nivel mundial son Joomla, *WordPress* y



Drupal (Paulsen, 2011). Para la realización de la propuesta de solución se utilizará como CMS de código abierto Drupal en su versión 7.

1.4.1.1. Drupal 7

Para llevar a cabo la propuesta de solución se propone utilizar Drupal 7 como CMS. Drupal 7 cuenta con un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores. Todo su contenido es totalmente indexado en tiempo real y se puede consultar en cualquier momento. El código fuente de Drupal 7 está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de blogs o de gestión de contenido propietarios, es posible extender o adaptar Drupal 7 según las necesidades. La comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante *jabber*, mensajes privados, *bookmarks*, entre otros. Además, cuenta con un robusto entorno de personalización que está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario (Drupal Hispano, 2011).

La gestión de usuarios permite que los usuarios se puedan registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como *jabber*, *blogger*, *live journal* u otro sitio Drupal. Para su uso en una intranet, se puede integrar con un servidor LDAP. Por otra parte, los administradores de Drupal 7 no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles (Drupal Hispano, 2011).

Para gestionar el contenido Drupal 7 cuenta con un sistema de control de versiones que permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior. Todo el contenido creado en Drupal 7 tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro. El contenido creado es funcionalmente un objeto. Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios o no sobre cada objeto. El sistema de temas de Drupal 7 separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio *web*. Se pueden crear plantillas con HTML y/o con PHP (Drupal Hispano, 2011).



Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos. Drupal 7 ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor *web* y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP es totalmente portable (Drupal Hispano, 2011).

Drupal 7 incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable. Además, el mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor (Drupal Hispano, 2011).

1.4.2. Metodología para el desarrollo de software

El desarrollo de *software* puede ser un reto para los desarrolladores, pues crear un *software* en el menor tiempo posible y con calidad puede ser casi imposible, si no se cuenta con algún proceso que ayude a agilizar el desarrollo de *software*. Para lograr esto se emplean las metodologías de desarrollo de *software*. “Una metodología de desarrollo de *software* es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo *software*” (Kruchten, 2004).

Las metodologías se pueden agrupar en dos grandes grupos, las tradicionales y las ágiles. Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. En estas el cliente llega a formar parte del equipo de trabajo. Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, además de las herramientas y notaciones que se usarán.

En la presente investigación se decide utilizar una metodología ágil porque son dinámicas y muy flexibles permitiendo cambios durante el proyecto, mientras que las metodologías tradicionales son estrictas y con cierta resistencia a cambios. Las metodologías ágiles generan pocos artefactos mientras que las tradicionales generan muchos artefactos. Además, los grupos de desarrollo dentro de las metodologías



ágiles son pequeños y trabajan en el mismo sitio. Por otra parte, las metodologías tradicionales tienen grupos de desarrollo más grandes y se encuentran distribuidos.

1.4.2.1. Metodología para el desarrollo de *software* SXP

La metodología SXP (Peñalver Romero, y otros, 2010) está compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que se sepa cuan avanzado se encuentra el proceso de desarrollo. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

Consta de 4 fases principales:

- Planificación: Es donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo: Es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega: Es donde se pone en marcha.
- Mantenimiento: Es donde se realiza el soporte para el cliente.

De cada una de ellas se despliegan 7 flujos de trabajo:

- Concepción inicial.
- Captura de requisitos.
- Diseño con metáforas.
- Implementación.



- Prueba.
- Entrega de la documentación.
- Soporte e investigación, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto.

De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la lista de reserva del producto, definición de las historias de usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a realizar para lograr el producto, pruebas, además de las tareas necesarias para realizar las investigaciones para documentar todo el proceso. En la **figura 1** se muestran las fases y flujos de trabajo de la metodología SXP.

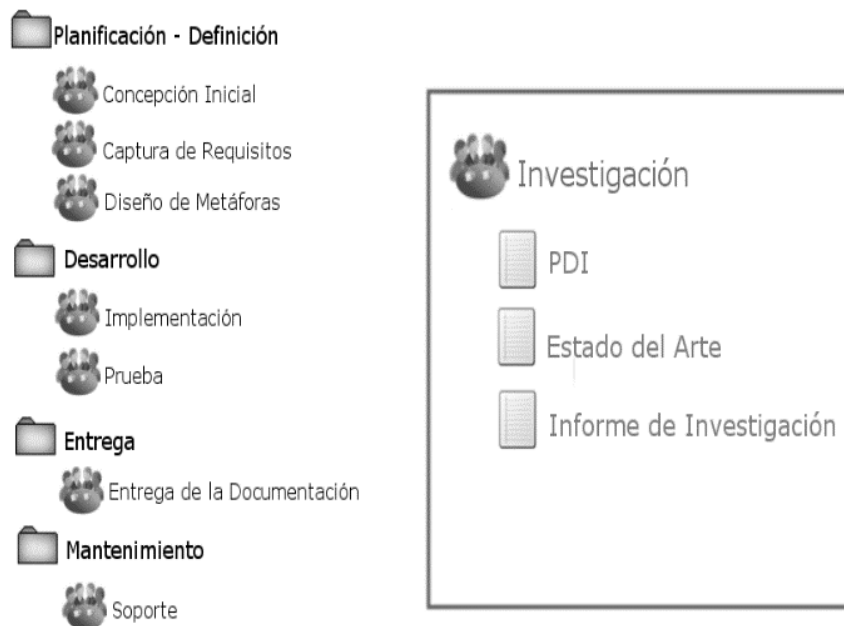


Figura 1. Fases y flujos de trabajo de SXP (Peñalver Romero, y otros, 2010).

Se decide utilizar la metodología SXP porque aumenta el interés y la motivación para el desarrollo. Al generar menos documentación y enfocarse más en la programación extrema permite ganar en tiempo, teniendo en cuenta que el equipo de desarrollo es pequeño. Además, se logra una mayor interacción con el cliente al ser parte del equipo, proporcionando una mejor calidad en el producto a entregar.



1.4.3. Lenguaje de desarrollo

La propuesta de solución a la situación problemática de esta investigación propone el desarrollo de un módulo que integre la elaboración y control del plan anual de eventos y la promoción del catálogo de revistas científicas en la Facultad 1 de la UCI, por lo que es necesaria la selección de lenguajes de programación y de un lenguaje de modelado. En el presente acápite se caracterizan los lenguajes seleccionados para el desarrollo del módulo *web*.

Para del desarrollo de aplicaciones *web* existen gran variedad de lenguajes de programación (ASP, *ActionScript*, Perl, Ajax entre otros). El uso del CMS Drupal restringe el conjunto de lenguajes de programación por lo que se utilizará PHP. Para la creación de los diagramas de diseño e implementación del portal, se utilizará el lenguaje de modelado unificado (UML), por ser un lenguaje estándar en el desarrollo de *software* profesional. Se hace necesario además utilizar las hojas de estilo en cascada (CSS) para complementar la apariencia, funcionalidad y desarrollo del portal, así como el lenguaje de marca de hipertexto (HTML), para la correcta funcionalidad del módulo.

1.4.3.1. UML 2.5

El lenguaje de modelado unificado (UML) es un lenguaje gráfico para visualizar, documentar y especificar cada una de las partes que comprende el desarrollo de *software*. Es un lenguaje de modelado formado por símbolos que es utilizado por muchas metodologías. Mediante el lenguaje UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de *software* previo al proceso intensivo de escribir código (Rodríguez, 2010).

1.4.3.2. PHP 5

El pre-procesamiento de hipertexto (PHP) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo *web* y que puede ser incrustado en HTML. El código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el *script*, desconociendo así el código subyacente que era antes de la ejecución del mismo. El servidor *web* puede ser incluso configurado para que procese todos los ficheros HTML con PHP. Aunque el desarrollo de PHP está centrado en la programación de *scripts* del lado del servidor, se puede utilizar para el desarrollo de otras aplicaciones. PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP,



IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) entre otros, tiene soporte para el intercambio de datos complejos entre los lenguajes de programación *web* (Enamorado, 2014).

1.4.3.3. CSS 3

Hojas de estilo en cascada (*Cascading Style Sheets* por sus siglas en inglés CSS), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y el formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores *web* controlar el estilo y el formato de múltiples páginas *web* al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (Cascading Style Sheets, 2013).

1.4.3.4. HTML 5

HTML 5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente, pero, incluso cuando algunas API (interface de programación de aplicaciones) y la especificación de CSS3 por completo no son parte del mismo. HTML 5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML 5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a internet y la diversidad de interfaces disponibles para interactuar con la *web*, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio (Gauchat, 2012).



1.4.4. Herramienta de modelado

Las herramientas de modelado permiten enfocarse en las características más importantes que posee un sistema. Con el uso de estas herramientas se puede discutir todos los cambios y correcciones en los requisitos, teniendo en cuenta un costo y riesgo mínimo. Además, se puede verificar que el analista comprenda correctamente el ambiente del cliente y que haya sido respaldado con información para que los diseñadores y programadores puedan construir el sistema sin ninguna dificultad.

1.4.4.1. Visual Paradigm 8.0

Es una herramienta CASE (Ingeniería de *Software* Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. *Visual Paradigm* ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Constituye una herramienta de *software* de probada utilidad para el analista. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos. En el presente trabajo se utiliza para representar el diagrama de clases, el diagrama de despliegue y el diagrama de componentes. Esta herramienta está especializada en la ingeniería del *software* de bases de datos. Incluye herramientas para la ingeniería inversa de bases de datos (Burrows, 2010).

1.4.5. Entorno integrado de desarrollo

Un entorno de desarrollo integrado, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDE ofrecen un marco de trabajo para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Anderson, y otros, 2014).

1.4.5.1. NetBeans 8.0



Es una herramienta utilizada por los programadores, pensada para escribir, compilar, depurar y ejecutar programas. Esta herramienta está codificada en el lenguaje de programación Java y permite desarrollar aplicaciones *web*, de escritorio y móviles usando las plataformas de Java. Es libre y gratuito para uso comercial como no comercial, sin restricciones para su uso. Está soportado por una activa comunidad de desarrolladores que ofrece una amplia documentación y recursos de entrenamiento. Es multiplataforma, disponible para diversos sistemas operativos como Windows, Mac OS y GNU Linux. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades (Netbeans, 2015)

1.4.6. Servidor web

La principal tarea de un servidor *web* es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor mediante el protocolo HTTP. El servidor espera peticiones del cliente y le da respuesta con el contenido solicitado. Disponen de intérpretes de lenguajes de programación que ejecutan códigos embebidos dentro del HTML de las páginas que contiene el sitio antes de enviar el resultado al cliente (Arredondo Morales, 2009).

1.4.6.1. Apache 2.4

Apache es un servidor *web* utilizado para el protocolo HTTP, posee la capacidad de funcionar como un proceso independiente, sin necesitar el apoyo de otras aplicaciones o directamente del usuario. Es una herramienta de *software* libre y código abierto desarrollado bajo una variante de la licencia BSD⁶, modular, multiplataforma, extensible, popular (cuenta con un gran soporte y ayuda técnica en la *web*) y gratuito. El servidor *web* Apache es altamente configurable y posee gestión de *logs*. Permite además la personalización de respuestas ante errores que puedan ocurrir (Martínez Guaita, 2012).

Dada su fácil integración con el CMS Drupal, además es una herramienta de *software* libre y código abierto, modular, multiplataforma, extensible, con amplio soporte y ayuda técnica, gratuito y por todas las características antes expuestas que lo destacan como el idóneo se decide escoger el servidor *web* Apache en su versión 2.4.

⁶ (*Berkeley Software Distribution* o en su concepto en español distribución de *software* Berkeley).



1.4.7. Sistema gestor de bases de datos

Un sistema gestor de base de datos (SGBD), es una agrupación de programas que sirven para definir, construir y manipular una base de datos. En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Controlar la concurrencia implica que, si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas. Un SGBD también debe encargarse de cumplir las reglas de integridad y redundancias. Otra función importante en un SGBD es su capacidad de realizar copias de seguridad y de recuperación de datos. Además, permiten la restricción de accesos no autorizados al suministrar múltiples interfaces de usuario (Alegsa, 2010).

1.4.7.1. Postgres SQL 9.3

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto potente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (PostgreSQL, 2016).

Es compatible con el almacenamiento de objetos binarios de gran tamaño, como imágenes, sonidos o video. Cuenta con interfaces de programación nativas para C / C++, Java, .NET, Python, Ruby, Tcl, Perl, ODBC, entre otros. Cuenta con características sofisticadas como la replicación asincrónica y sincrónica, transacciones anidadas (puntos de retorno), "tablespaces", copias de seguridad en caliente, múltiples métodos de autenticación, MVCC (Control de Concurrencia multi-Versión) el cual permite que se ejecuten sobre una tabla varias transacciones a la vez y un sofisticado planificador consulta/optimizados (PostgreSQL, 2016). Se hace notable entre sus características su buen funcionamiento con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Es un sistema multiplataforma disponible para Linux y Windows en todas sus variantes, presentando una extensa y gran variedad de documentación.

1.4.7.2. MySQL 5



Utiliza el lenguaje de consulta estructurado (por sus siglas en inglés SQL). Es un producto de código abierto desarrollado bajo la licencia GPL (Licencia Pública General GNU), lo que hace posible que pueda ser modificado por los usuarios que lo utilizan. Es un SGBD usado y estandarizado para acceder a base de datos relacionales. Facilita la integración con programas desarrollados en C y C++ pues fue desarrollado en este lenguaje. Es un sistema multiplataforma, totalmente funcional en sistemas operativos como Linux, Mac X, UNIX y Microsoft Windows. Permite la creación de base de datos con acceso desde páginas *web* dinámicas, dando la posibilidad de realizar múltiples y rápidas consultas (Thibaud, 2006).

1.4.8. Herramientas de prueba

Una excelente forma de validar las aplicaciones y sitios *web* consiste en someterlos a pruebas para comprobar cómo se comportará su rendimiento general y el cumplimiento respecto a las especificaciones iniciales del sistema. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

1.4.8.1. Escáner de vulnerabilidades *web* Acunetix 8.0.

El escáner de vulnerabilidades *web* Acunetix es una herramienta que será capaz de escanear sitios *web* en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor. Detecta técnicas de hacking como pueden ser: ataques de ejecución de código, inyección SQL, autenticación, entre otros. Además destaca por su completo conjunto de herramientas: editor HTTP, *Sniffer* HTTP, Flooder HTTP, encriptador de palabras, control de vulnerabilidades, SQL, entre otros (Acunetix, 2016).

1.4.8.2. Apache JMeter 2.12

Apache JMeter es una de las herramientas *Open Source* para realizar pruebas de carga. Además de HTTP soporta protocolos como POP3, IMAP, FTP, entre otros y permite programar lo que llama planes de prueba que consisten en guiones de ejecución para las pruebas de *stress*. Es un software de código abierto, una aplicación diseñada para cargar la conducta funcional de prueba y medida de rendimiento. Fue diseñado originalmente para aplicaciones *web* de prueba, pero desde entonces se ha expandido a otras funciones de prueba.



Jmeter puede ser utilizado para probar el rendimiento tanto en los recursos estáticos y dinámicos. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, la red o el objeto a probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Se puede utilizar para realizar un análisis gráfico de rendimiento o para probar su comportamiento (Apache Software Foundation, 2016).

1.5. Conclusiones del capítulo

- El estudio de los sistemas homólogos existentes relacionados con la presente investigación evidenció que ninguna de las soluciones satisfacía completamente las necesidades del Vicedecanato de Investigación y Posgrado de la Facultad 1 de la UCI. Por otra parte, el estudio arrojó que estos sistemas comparten funcionalidades como actualizar estados, asignar premios y realizar búsquedas que se incluirán en la propuesta de solución.
- Se seleccionó Drupal 7 como sistema gestor de contenido, NetBeans 8.0 como entorno de desarrollo integrado, como servidor *web* Apache 2.4 y como sistemas gestores de bases de datos PostgreSQL 9.3 y MySQL 5. El proceso de desarrollo estuvo guiado por la metodología SXP.



Capítulo II: Características y arquitectura del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas

2.1. Introducción

En este capítulo se realiza una descripción general de la propuesta de solución, a partir de su funcionamiento. Se representan a través de un modelo conceptual los principales conceptos que se manejan actualmente durante los procesos de gestión de planes de eventos y catálogos de revistas en la Facultad 1. Se define la arquitectura y los patrones arquitectónicos aplicables al sistema de gestión de contenidos seleccionado. Se definen los requisitos funcionales y no funcionales para el módulo a desarrollar. Las funcionalidades se describen mediante las historias de usuario (HU).

2.2. Descripción de la propuesta de solución

El módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas, permitirá la gestión del plan anual de eventos científicos, el catálogo de revistas científicas y el plan de misiones de la facultad 1 de la UCI. Utilizando este módulo se podrá crear, modificar o eliminar la información asociada a los planes de eventos y misiones, así como del catálogo de revistas científicas. El módulo permitirá generar reportes sobre el cumplimiento de los planes. Además, brindará la opción de exportar a excel los planes de eventos y misiones, así como el catálogo de revistas científicas y exportar a pdf el plan de eventos científicos. También generará tres tipos de certificados: de participación, de tribunales y de premios.

Para la gestión del plan de eventos científicos, el plan de misiones y el catálogo de revistas se definen los siguientes roles:

Rol **Administrador**: Puede acceder a todas las funcionalidades del sistema (Vicedecano de Investigación y Posgrado).

Rol **Gestor**: Solo puede acceder a las funcionalidades del sistema para gestionar información sobre el área a la que pertenece (Jefes de Departamentos y Subdirectores de Centro).



2.3. Modelo conceptual

Un modelo conceptual tiene como propósito fundamental organizar y representar, de manera semi-formal y unívoca, el conocimiento de un área o campo específico asociado a un sistema de gestión o de información. Está orientado a describir semántica y aseveraciones sobre la información del dominio particular que representa. La elaboración de dicho modelo es abstracta e independiente de consideraciones de diseño o de tecnología, es decir se identifican las cosas fundamentales sin dar demasiada importancia a ejemplos o instancias particulares y sin dejarse influenciar por la eventual participación de estos conceptos en elementos o componentes de software o de una solución de informática (Ferrer, 2013).

A continuación, en la **figura 2** se muestra el modelo conceptual para la descripción del problema de la presente investigación.

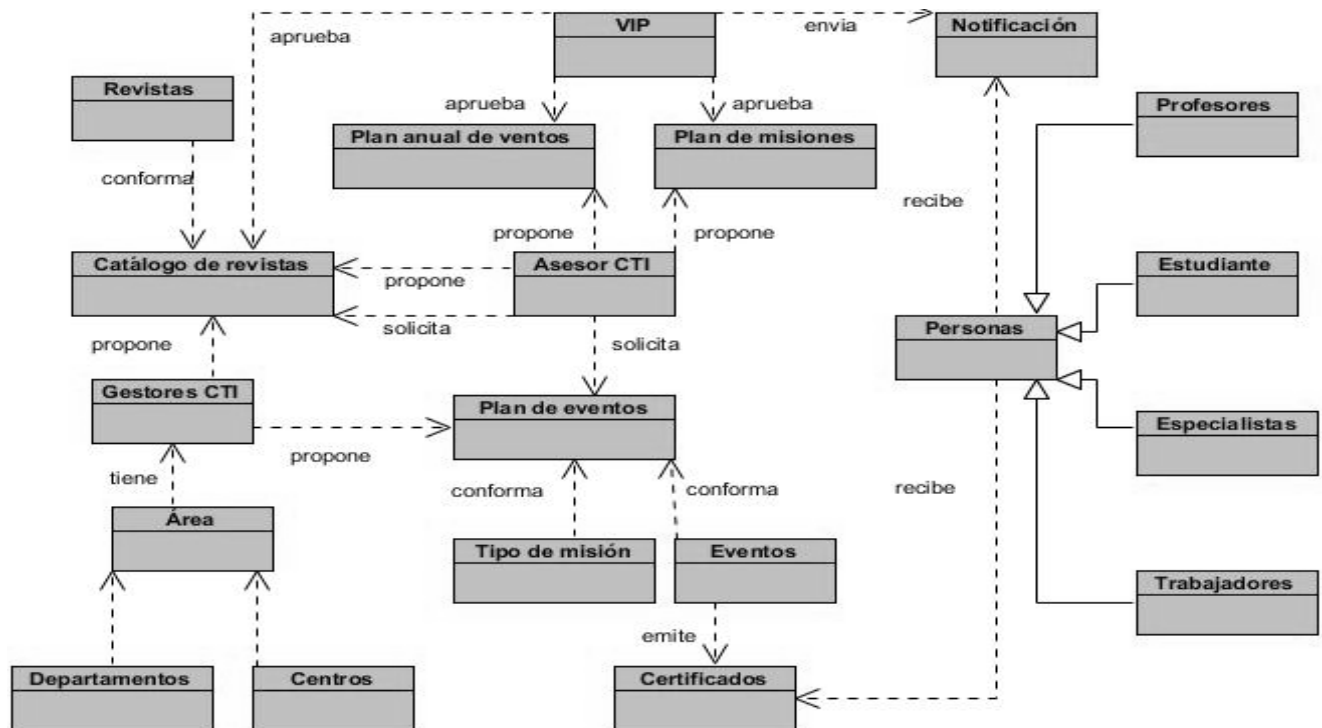


Figura 2. Modelo conceptual.

2.3.1. Definición de conceptos



- **VIP:** Persona que ocupa el puesto de Vicedecano de Investigación y Posgrado en la Facultad 1, encargada de dirigir todo el proceso, tiene la función de gestionar los planes anuales de eventos, planes de misiones y las alertas de publicación en la Facultad.
- **Revistas:** Son aquellas revistas que están contenidas dentro del catálogo de revistas propuesto por cada área para la publicación de artículos científicos.
- **Eventos:** Son aquellos eventos científicos que están contenidos dentro del plan de eventos propuestos por las diferentes áreas y para los cuales se emite un certificado de participación.
- **Tipo de misión:** El tipo de misión puede recibir varias categorías como pueden ser: capacitación, estancia de doctorado, estancia de investigación, evento internacional, o evento.
- **Gestores CTI:** Persona que dirige cada área, es el encargado de elaborar las propuestas de planes de eventos y catálogos de revistas.
- **Área:** Se refiere a las siete áreas con que cuenta la Facultad 1, dígame: los cuatro departamentos docentes y los tres centros de desarrollo de software.
- **Catálogo de revistas:** Contiene las revistas que proponen las áreas para la publicación de los artículos científicos.
- **Asesor CTI:** Persona que ocupa el cargo de Asesor de Ciencia Tecnología e Innovación del Vicedecano de Investigación y Posgrado en la Facultad 1. Esta persona es la encargada de solicitar a las áreas las correspondientes propuestas de catálogos de revistas y planes de eventos, para elaborar las propuestas de planes anuales de eventos, planes de misiones y catálogos de revistas de la Facultad y someterlos a la aprobación del **VIP**.
- **Plan de eventos:** Documento que contiene la propuesta de plan de eventos que realizan las áreas para conformar el plan de eventos de la Facultad.
- **Certificados:** Documento que avala la participación en eventos.
- **Personas:** Esta entidad representa una generalización que engloba a las entidades: Profesores, Especialistas, Trabajadores y Estudiantes.
- **Profesores, Especialistas y Trabajadores:** Estas entidades se corresponden con cargos ocupacionales dentro de la Facultad 1. Además, son entidades que heredan de la entidad **Personas**, y reciben las alertas por correo electrónico, así como los certificados de participación en eventos.
- **Estudiantes:** Esta entidad hereda de la entidad **Personas** y recibe alertas por correo electrónico, así como los certificados de participación en eventos.



- **Plan anual de eventos:** Documento que contiene los eventos que se propusieron en cada plan de eventos correspondiente a cada área de la Facultad, y que se somete a la aprobación del VIP.
- **Plan de misiones:** Documento que contiene las propuestas realizadas por áreas y por la Facultad de eventos científicos, estancias de investigación, becas y otras acciones.
- **Notificación:** Notificación que se envía por correo electrónico a las personas para recordarles las fechas de los eventos.

2.4. Requisitos de software

Los requisitos de software establecen con detalle las funciones o servicios y restricciones operativas del sistema. Los requisitos de software se clasifican en funcionales y no funcionales.

Los requisitos funcionales: Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y como se debe comportar en situaciones particulares. En algunos casos los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

Los requisitos no funcionales: Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2005).

2.4.1. Lista de reserva del producto

A continuación, se describe la lista de reserva del producto, compuesta por los requisitos funcionales (RF en lo adelante) y no funcionales (RNF en lo adelante) de la aplicación *web*.

Tabla 1. Lista de reserva del producto.

Asignado a	Ítem *	Descripción	Estimación	Estimado por
Prioridad		Alta		
Adrián	1	Insertar evento	1/5	prog

*Módulo para la gestión del plan anual de eventos
y del catálogo de revistas científicas.*



Adrián	2	Modificar evento	1/5	prog
Adrián	3	Eliminar evento	1/5	prog
Adrián	4	Generar plan anual de eventos	1/5	prog
Adrián	5	Controlar plan de eventos	1/5	prog
Adrián	6	Generar reportes del plan anual de eventos	1/5	prog
Adrián	7	Insertar misión	1/5	prog
Adrián	8	Modificar misión	1/5	prog
Adrián	9	Eliminar misión	1/5	prog
Adrián	10	Generar plan de misiones	1/5	prog
Adrián	11	Controlar plan de misiones	1/5	prog
Adrián	12	Generar reporte del plan de misiones	1/5	prog
Adrián	13	Mostrar % de cumplimiento del plan de eventos	1/5	prog
Adrián	14	Insertar revista	1/5	prog
Adrián	15	Modificar revista	1/5	prog
Adrián	16	Eliminar revista	1/5	prog
Adrián	17	Generar catálogo de revistas	1/5	prog
Prioridad		Media		
Adrián	18	Eliminar plan anual de eventos	1/5	prog
Adrián	19	Exportar el plan anual de eventos en formato pdf y excel	1/5	prog
Adrián	20	Exportar el plan de misiones en formato excel	1/5	prog
Adrián	21	Eliminar el catálogo de revistas	1/5	prog
Adrián	22	Exportar el catálogo de revistas en formato excel	1/5	prog
Adrián	23	Crear certificado de participación	1/5	prog
Adrián	24	Modificar certificado de participación	1/5	prog
Adrián	25	Eliminar certificado de participación	1/5	prog
Adrián	26	Generar certificado de participación	1/5	prog
Adrián	27	Exportar certificado de participación en formato	1/5	prog



		pdf		
Prioridad		Baja		
Adrián	28	Listar plan de eventos por temáticas y área	1/5	prog
Adrián	29	Listar catálogo de revistas por temáticas y los cuatro niveles del MES	1/5	prog
RNF (Requisitos No Funcionales)				
Usabilidad				
	1	El módulo será fácil de usar al ser visualizado en dispositivos para las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.		prog
	2	El módulo permitirá a los usuarios aprender rápidamente a utilizarlo.		prog
	3	Se utilizarán tamaños de letra apropiados para personas con alguna discapacidad visual. Se hará uso de la tecla Control del teclado y el scroll del <i>mouse</i> para la lectura de pantallas por personas con discapacidad visual.		prog
Seguridad				
	4	El sistema será utilizado solo por los usuarios con permisos de usuario administrador y usuario registrado con permisos.		prog
	5	Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.		prog
Soporte				
	6	Las operaciones de mantenimiento del sistema no deberán exceder de 2 días.		prog
	7	El sistema funcionará en varios sistemas operativos como Windows XP, 7, 8, 8.1, 10, Linux en todas las distribuciones basadas en Debian,		prog



		Ubuntu, OpenSuse, Fedora.		
Interfaz de Hardware				
	8	PC cliente con una capacidad de memoria RAM: 2 GB.		prog
	9	PC Servidor <i>web</i> y Servidor de BD CPU: 2 núcleos de procesamiento Al menos 4 GB de memoria RAM Espacio: Mínimo 20 GB		prog
Interfaz de Software				
	10	El menú de configuración del módulo presentará una interfaz agradable, utilizando colores claros, en tonalidad azul.		prog
	11	Los textos serán presentados en tipografías compatibles con varios sistemas operativos y se resaltarán los encabezados en negrita.		prog
Requisitos Legales				
	12	Las tecnologías y herramientas utilizadas están bajo licencia GNU/GPL en su versión 2 para el CMS Drupal.		prog
	13	Uso de licencia PHP <i>License</i> .		prog
	14	Uso de licencia de <i>Visual Paradigm 8</i> .		prog
Operacionales				
	15	El sistema necesitará para su correcto funcionamiento cargar las librerías fpdf, tcpdf.		prog
	16	Para la distribución del producto y su instalación se necesitará un entorno de trabajo compuesto por: <ul style="list-style-type: none"> • Servidor <i>web</i>: Apache, Nginx, o Microsoft IIS. 		prog



		<ul style="list-style-type: none"> • Base de Datos: MySQL 5, PostgreSQL 9.3, MariaDB. • Sistema Gestor de contenido Drupal 7.42. • Lenguaje de programación PHP 5.2.5 o versiones posteriores recomendadas. • Navegadores <i>web</i> Chrome 48, Firefox 44, Opera 34 e Internet Explorer a partir de su versión 10. 		
Rendimiento				
	17	El sistema debe responder en un máximo de 10 segundos las solicitudes de los usuarios.		prog
	18	El sistema debe soportar un mínimo de 300 peticiones concurrentes.		prog
	19	El sistema debe ser escalable permitiendo que se le pueden agregar nuevas funcionalidades sin afectar las funcionalidades implementadas.		prog

A continuación, se muestra en la **tabla 2** el resumen de la etapa de captura de los requisitos funcionales.

Tabla 2. Resumen de la etapa de captura de los RF.

Prioridad para el cliente.	Cantidad de requisitos funcionales.
Alta	17
Media	10
Baja	2
Total	29



2.5. Historias de usuario

Se utilizan para especificar los requisitos de las aplicaciones software en las metodologías ágiles (SCRUM, XP, FDD, ASD, AUP, LD, entre otros). Las historias de usuario son tarjetas en dónde el interesado describe brevemente (con el fin de que sean dinámicas y flexibles) las características que el sistema debe poseer, sean requisitos funcionales o no funcionales (Amaro, y otros, 2007).

Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que se pueda implementar en unas tres semanas aproximadamente (para no superar el tamaño de una iteración, que es el tiempo estimado para una entrega de un componente de desarrollo de manera incremental (Villamizar Suaza, 2013).

A continuación, en las **tablas 3 y 4** se muestran las historias de usuario para insertar evento y eliminar revista. El resto de las historias de usuario se encuentran descritas en los anexos (ver Anexo 2).

Tabla 3. HU_Insertar evento.

Historia de usuario	
Número: 1	Nombre Historia de Usuario: Insertar evento
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Adrián Castillo Chávez	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1/5
Riesgo en Desarrollo: Alto	Puntos Reales: 1/5
Descripción: El módulo deberá permitir al usuario con rol administrador insertar los eventos. El módulo deberá permitir a los usuarios con rol gestor insertar los eventos, pero solo aquellos eventos que pertenezcan a su área.	



Observaciones:

Para insertar un evento, el módulo permitirá a los usuarios con roles administrador y gestor, crear nuevos eventos en el módulo. Para ello se deben introducir los siguientes datos:

- Nombre del evento (obligatorio, campo de texto) Ejemplo: Seminario Juvenil de Estudios Martiano.
- Nivel (obligatorio, campo de selección). Ejemplo: UCI.
- Fecha inicio del evento (obligatorio, campo de tipo fecha).
- Fecha fin del evento (obligatorio, campo de tipo fecha).
- Fecha de envío de resumen (obligatorio, campo de tipo fecha).
- Fecha de envío de ponencia (obligatorio, campo de tipo fecha).
- Fecha de aceptación (obligatorio, campo de tipo fecha).
- Coordinador de evento (obligatorio, campo de texto). Ejemplo: Adrián Castillo Chávez.
- URL del evento (obligatorio, campo de texto).
- Temática (obligatorio, campo de selección).
- Estado (obligatorio, campo de selección). Ejemplo: Realizado.
- Área (obligatorio, campo de selección).

Luego de que se haya insertado el evento el módulo mostrará el siguiente mensaje: “La acción se ha realizado satisfactoriamente”.

Prototipo de interface:

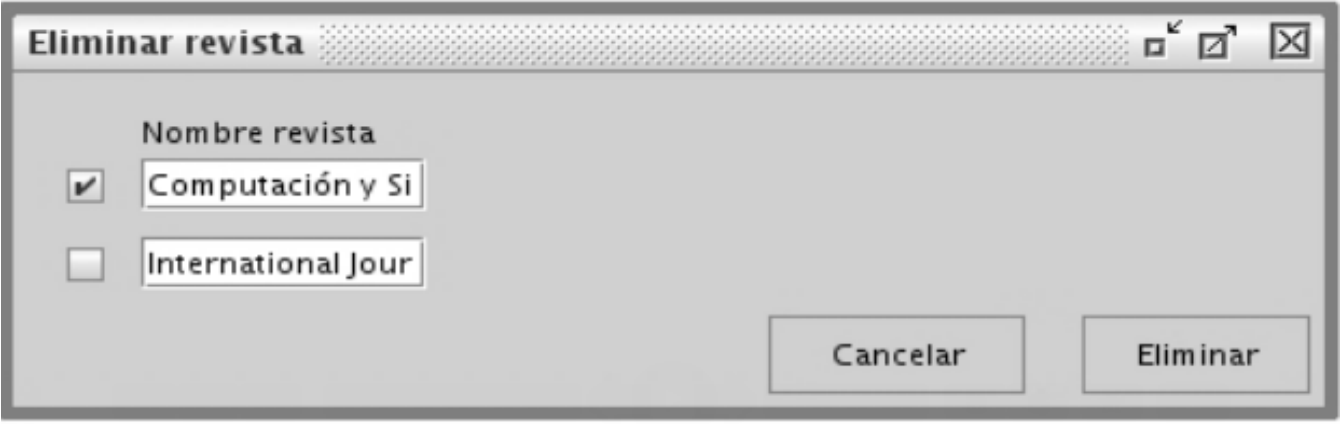
The screenshot shows a web form titled "Insertar evento". The form contains the following fields and controls:

*Nombre	*Nivel	*Fecha inicio	*Fecha fin	*Fecha envío resumen	*Fecha envío ponencia	*Fecha aceptación	*Coordinador	*URL	*Temática	*Estado	*Área
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

At the bottom right of the form, there are two buttons: "Cancelar" and "Insertar".



Tabla 4. HU_Eliminar revista.

Historia de usuario	
Número: 16	Nombre Historia de Usuario: Eliminar revista
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Adrián Castillo Chávez	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1/5
Riesgo en Desarrollo: Alto	Puntos Reales: 1/5
<p>Descripción:</p> <p>El módulo deberá permitir al usuario con rol administrador eliminar las revistas.</p> <p>El módulo deberá permitir a los usuarios con rol gestor eliminar las revistas, pero solo aquellas que pertenezcan a su área.</p>	
<p>Observaciones:</p> <p>Para eliminar una revista, el módulo permitirá a los usuarios con roles administrador y gestor, eliminar las revistas en el módulo. El módulo mostrará una ventana con el siguiente mensaje: “Seguro que desea realizar esta acción”, luego de que se haya eliminado la revista, el sistema mostrará el siguiente mensaje: “La acción se ha realizado satisfactoriamente”.</p>	
<p>Prototipo de interface:</p> 	



2.6. Arquitectura

El secreto de Drupal para conseguir su reconocida flexibilidad y facilidad en la creación de sitios *web* es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio *web* como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuario y plantillas (*templates*) (Borrueal, 2014).

Al asumirse Drupal 7 como CMS para el desarrollo de la presente investigación se decide heredar la arquitectura n-capas de Drupal 7. Esta elección se hace por las potencialidades que brinda esta arquitectura donde el objetivo principal es separar los diferentes aspectos del desarrollo mediante capas y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completamente. Además, permite ampliar funcionalidades e integración de nuevos módulos.

Para la propuesta de solución se definieron cinco capas. Una primera capa para los nodos, donde se ubican los tipos de contenidos, en esta capa se utiliza un conjunto de funciones llamadas *hook* o ganchos como también se les conoce, estos son el medio de comunicación entre los módulos y el sistema. En la segunda capa es donde se ubican los módulos, en este caso el módulo que conforma la propuesta de solución el cual lleva por nombre *gesec* y otros módulos que se puedan necesitar. En la tercera capa se implementa el *hook* menú que permite gestionar los menús del sistema. Una cuarta capa para los permisos y la quinta capa dedicada a darle formato a la interfaz de usuario con los estilos CSS.

A continuación, en la figura 3 se muestra la arquitectura para el desarrollo de la presente investigación.



Figura 3. Arquitectura de la aplicación (Elaboración propia).

2.7. Patrones de diseño

Los patrones de diseño son los principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de software. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Pressman, 2007).

2.7.1. Patrones GoF

Los patrones GoF⁷, describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para ayudar a realizar tareas complejas (Márquez, 2013).

⁷ Banda de los cuatro, compuesta por los siguientes autores: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.



Observer (Observador): Los módulos que implementan un *hook* de inserción o actualización de una entidad, son declarados como observadores de las entidades con las que interactúan (Silvente Lao, 2015). En la aplicación el uso del patrón Observador se evidencia cuando se modifica el contenido revista en el módulo, todas las dependencias del sistema son informadas de la modificación realizada.

Bridge (Puente): La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. El módulo implementado es programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar consultas siguiendo la API definida (Silvente Lao, 2015). En la aplicación el uso del patrón puente se evidencia al eliminar un contenido, pues impide que se afecten los módulos relacionados con el contenido eliminado, permitiéndoles a estos módulos seguir funcionando y que puedan seguir siendo utilizados por otros contenidos.

Chain of responsibility (Cadena de responsabilidades): El sistema de menú de Drupal es la evidencia del patrón cadena de responsabilidades. En cada petición de una página, el sistema de menú de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y que función se debe llamar para procesar la petición. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos (Silvente Lao, 2015).

Singleton (Instancia única): El objetivo de este patrón es asegurarse de que de una clase solo existe una instancia y que esta es accesible, o mejor dicho, ofrecer un punto de acceso a ella (Márquez, 2013). En la aplicación el uso del patrón instancia única se evidencia cuando al intentar crear un tipo de contenido se verifica que ese contenido no haya sido creado con anterioridad.

2.8. Tarjetas CRC

Las tarjetas CRC⁸ proponen una forma de trabajo, preferentemente grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y como colaboran con otros para realizar tareas. Esta técnica utiliza las llamadas tarjetas CRC, las cuales registran el nombre de las clases, sus responsabilidades y las otras clases con la que colaboran (Cortes, y otros, 2013). A continuación, se muestra en las **tablas 5**

⁸ Clase-Responsabilidad-Colaboración.



y 6 las tarjetas CRC de las clases `gesecc.module` y `gesecc.install` respectivamente. El resto de las tarjetas CRC se encuentran descritas en los anexos (ver Anexo 3).

Tabla 5. Tarjeta CRC `gesecc.module`.

Tarjeta CRC	
Clase: <code>gesecc.module</code>	
Responsabilidad	Colaboración
Implementa los <i>hook</i> necesario para crear los tipos de contenidos: evento, misión, revista, certificado entre otros.	<code>gesecc.views_default</code>

Tabla 6. Tarjeta CRC `gesecc.install`.

Tarjeta CRC	
Clase: <code>gesecc.install</code>	
Responsabilidad	Colaboración
Crea los campos necesarios para los diferentes tipos de contenidos.	

2.9. Modelo de despliegue

Los modelos de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (Marca Huallpara , y otros, 2013). A continuación, en la **figura 4** se muestra el modelo de despliegue que resulta de la presente investigación.

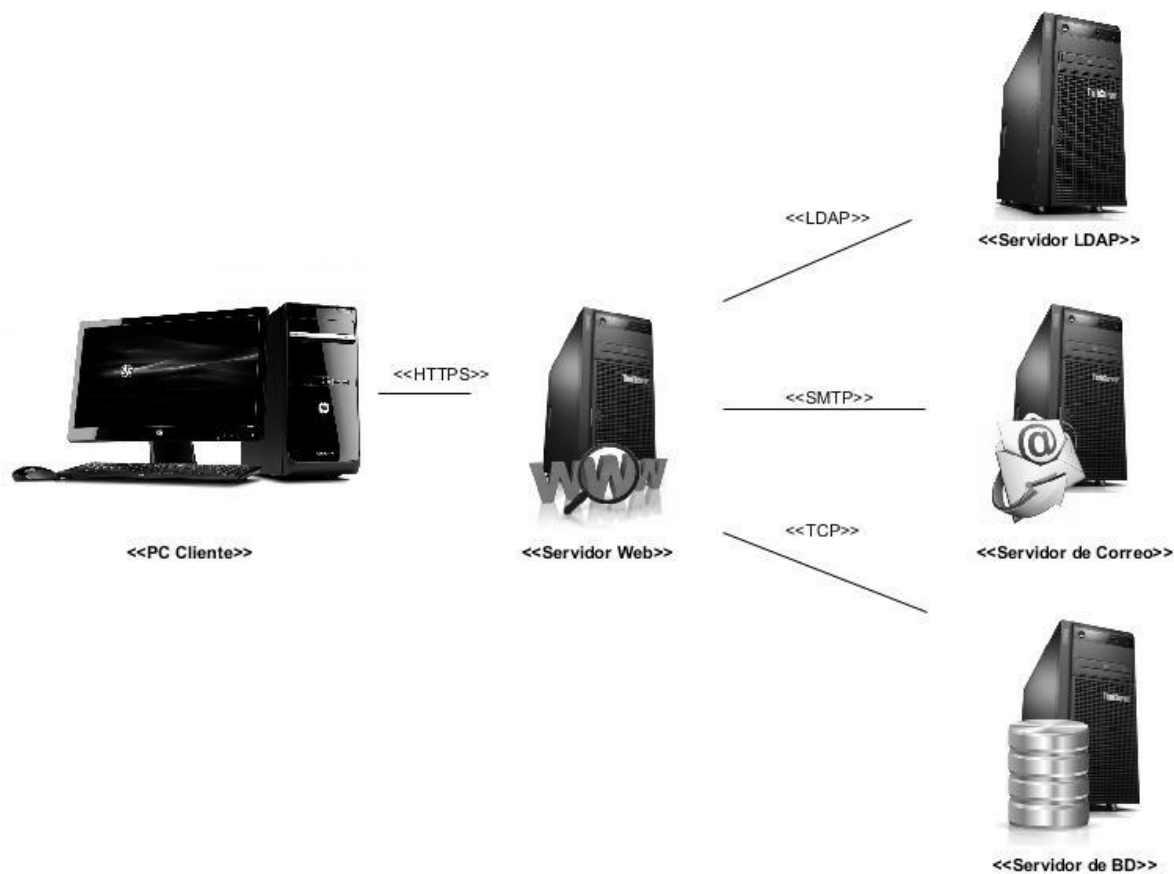


Figura 4. Modelo de despliegue (Elaboración propia).

2.9.1. Descripción de los elementos de interfaz y comunicación

- **<TCP>** Mediante este protocolo se establece la conexión entre el servidor *web* y el servidor de base de datos. Este permite realizar las peticiones y obtener respuestas a las mismas de forma rápida.
- **<SMTP>** Mediante este protocolo se establece la conexión entre el servidor *web* y el servidor de correo. Este permite autenticarse en el sistema mediante cuentas de usuario, así como el envío de notificaciones.
- **<HTTPS>** Mediante este protocolo se logra establecer una conexión segura entre la PC cliente y el servidor *web*.
- **<LDAP>** Mediante este protocolo se establece la conexión entre el servidor de aplicaciones *web* el servidor LDAP.



2.10. Conclusiones del capítulo

- La obtención de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, garantizó que la solución respondiera a las necesidades del cliente y de los usuarios finales, sirviendo de guía para el desarrollo de las distintas funcionalidades del módulo.
- La selección de la arquitectura y los patrones de diseño utilizados por Drupal permitieron ampliar funcionalidades e integración de nuevos módulos.



Capítulo III: Construcción y pruebas del módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas

3.1. Introducción

En el presente capítulo se modela el diagrama de componentes, se describen los estándares de codificación utilizados durante la implementación. Además, se elaboran los casos de prueba para validar los requerimientos funcionales y el correcto funcionamiento del sistema.

3.2. Diagrama de componentes

El diagrama de componentes es usado para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de este. Proporciona las interfaces necesarias para la utilización de sus funcionalidades (Ulacia Díaz, 2009). La metodología SXP no utiliza este diagrama, pero para una mejor representación de la solución el autor se apoya en este artefacto. A continuación, se describen los componentes del diagrama.

- **index.php:** Constituye el punto de inicio de la aplicación, es a partir de esta entrada que son solicitados los distintos módulos presentes en el CMS Drupal 7.
- **sites:** En este directorio se incorporan los temas, módulos y librerías que se puedan añadir.
- **default:** Contiene las clases de configuración settings.php.
- **scripts:** Contiene archivos .sh que extienden las funcionalidades de Drupal 7.
- **misc:** Contiene imágenes requeridas por el sistema y archivos JavaScript.
- **themes:** Contiene los temas correspondientes a la distribución de Drupal.
- **modules:** Contiene los módulos del núcleo.
- **includes:** Las funciones y *scripts* posibilitan el funcionamiento de los módulos y del CMS Drupal, contiene además funciones que permite la conexión a la base datos.
- **db:** Es la base datos de Drupal. Gestiona el acceso a la información almacenada referente al funcionamiento del sistema.
- **module:** Contiene la Propuesta de solución.



- **gesec.ifo**: Contiene información sobre el módulo (nombre, descripción, versión de Drupal, entre otros).
- **gesec.install**: Archivo que implementa el hook_schema para la creación de los tipos de contenidos: revista, evento y misión en la base de datos del sistema.
- **gesec.module**: Archivo que contiene las llamadas a las funciones *hook* de Drupal y algunas funciones básicas del sistema.

A continuación, en la **figura 5** se muestra el diagrama de componentes.

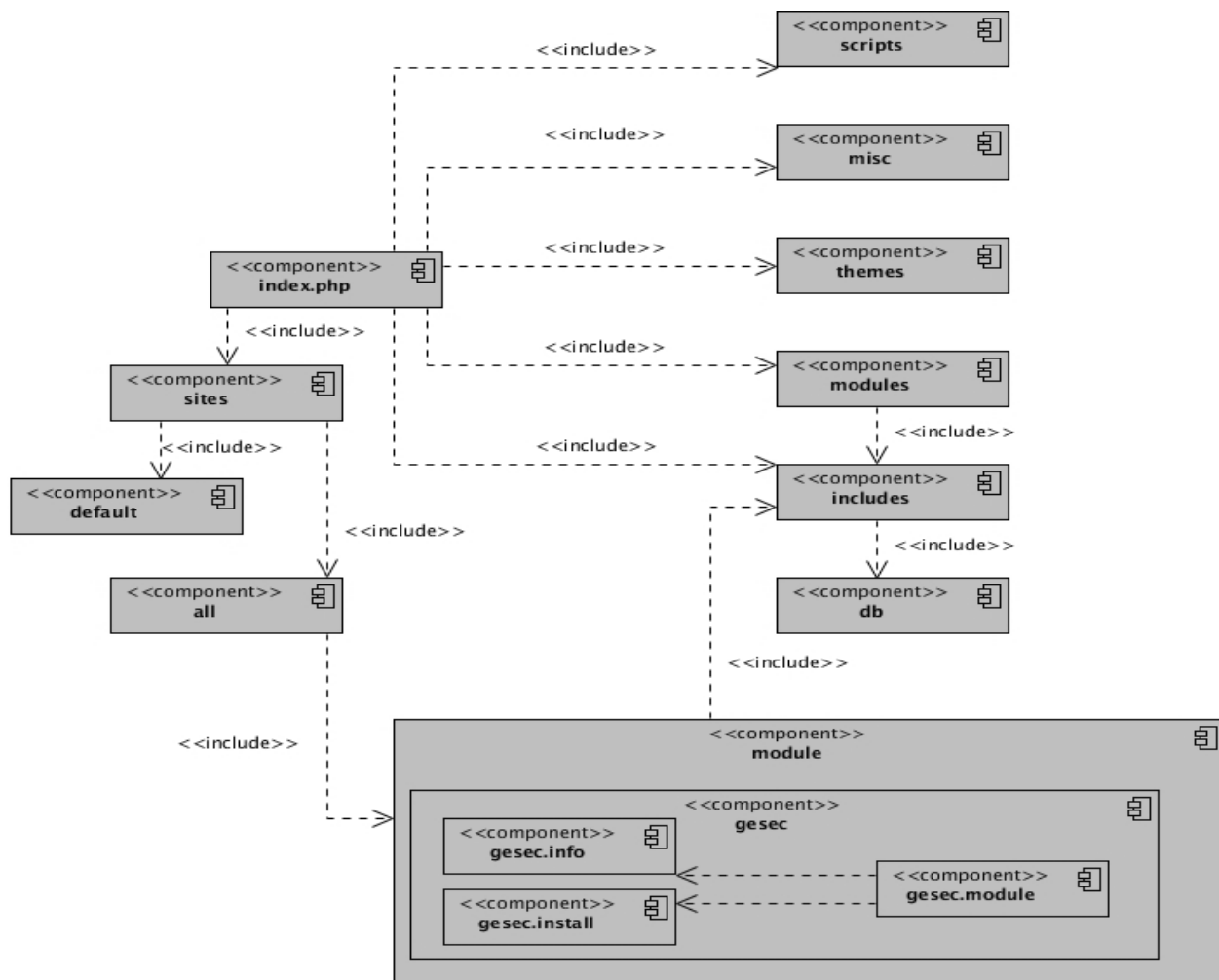


Figura 5. Diagrama de componentes (Elaboración propia).



3.3. Estándares de codificación

Los estándares de codificación o estilos de programación, como también se les llama, definen la estructura y apariencia física del código, lo que facilita la lectura, comprensión y mantenimiento del mismo (Drupal, 2016). Para el desarrollo del módulo, se utilizaron los estándares de codificación que se describen a continuación.

3.3.1. Indentación y espacios en blanco

Para facilitar la comprensión del código se utilizan tabuladores o espacios en blanco en determinadas líneas del mismo. Las líneas no deben tener espacios en blanco al final. Se utiliza una indentación de dos espacios como se muestra en la **figura 6**.

```
11 function planes() {
12     $query = db_select('node', 'n')->condition('n.type', 'plan_de_eventos', '=');
13     $query->join('field_data_field_fecha', 'f', 'f.entity_id=n.nid');
14     $result = $query
15         ->fields('n', array('title', 'nid'))
16         ->fields('f', array('field_fecha_value'))
17         ->execute();
18     return $result;
19 }
```

Figura 6. Uso de indentación y espacios en blanco (Elaboración propia).

3.3.2. Operadores

Todos los operadores binarios como: +, -, =, ==, >, entre otros, deben tener un espacio antes y después del operador, para facilitar la lectura. Los operadores unarios como: ++, no deberían tener un espacio entre el operador y la variable o el número que están operando, como se muestra en la **figura 7**.



```
200 function gesecc_views_api() {
201     return array(
202         'api' => '3.0',
203         'path' => drupal_get_path('module', 'gesecc') . '/includes',
204     );
205 }
```

Figura 7. Uso de operadores (Elaboración propia).

3.3.3. Estructuras de control

Las estructuras de control incluyen *if*, *for*, *while*, *switch*, entre otros. Las mismas deben tener un espacio entre la palabra clave de control y el paréntesis de apertura, para distinguirlas de las llamadas a funciones como se muestra en la **figura 8**.

```
153 for ($color_count = 0; $color_count < 10; $color_count++) {
154     $form['display']['colors'][$color_count] = array(
155         '#type' => 'textfield',
156         '#attributes' => array('TYPE' => 'color'),
157         '#size' => 10,
158         '#maxlength' => 7,
159         '#theme_wrappers' => array(),
160         '#suffix' => ' ',
161         '#default_value' => $options['colors'][$color_count],
162         '#parents' => array_merge($parents, array('colors', $color_count)),
163     );
164 }
```

Figura 8. Uso de estructuras de control (Elaboración propia).

Se recomienda usar siempre las llaves `{}` aún en aquellos casos que no resulte obligatorio su uso y las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior como se muestra en la **figura 9**.



```
1102     foreach ($array_paises as $pais) {  
1103         $list[$pais] = $pais;  
1104     }
```

Figura 9. Uso de llaves en las estructuras de control (Elaboración propia).

3.3.4. Longitud de líneas

Las líneas de código no deben contener más de 80 caracteres como se muestra en la **figura 10**.

```
82     function gesec_node_info() {  
83         return array(  
84             'evento' => array(  
85                 'name' => t('Evento'),  
86                 'base' => 'evento',  
87                 'description' => t('Tipo de contenido evento.'),  
88                 'has_title' => TRUE,  
89                 'title_label' => t('Nombre del evento'),  
90                 'locked' => FALSE,  
91             ),  
92         );
```

Figura 10. Longitud de línea con menos de 80 caracteres (Elaboración propia).

Las líneas que contienen definiciones de clases o funciones, nombres de funciones extensos, declaraciones de variable, pueden exceder los 80 caracteres como se muestra en la **figura 11**.

```
83     'default_value' => isset($options['label_field']) ? $options['label_field'] : $first_field,
```

Figura 11. Longitud de línea con más de 80 caracteres (Elaboración propia).

3.3.5. Arreglos

Los valores dentro de un arreglo deben separarse por un espacio (después de la coma que los separa). El operador => se debe separar por un espacio a ambos lados. En caso de que la línea de declaración del arreglo supere los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se deberá escribir, aunque no existan más elementos como se muestra en la **figura 12**.



```
102     'revista' => array(  
103         'name' => t('Revista'),  
104         'base' => 'revista',  
105         'description' => t('Tipo de contenido revista.'),  
106         'has_title' => TRUE,  
107         'title_label' => t('Título de la revista'),  
108         'locked' => TRUE,  
109     ),
```

Figura 12. Uso de arreglos (Elaboración propia).

3.3.6. Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto o cuando el texto puede incluir alguna comilla simple, como se muestra en la **figura 13**.

```
11     'title' => t('Reportes de planes de eventos'),
```

Figura 13. Uso de comillas (Elaboración propia).

3.3.7. Etiquetas de apertura y cierre de PHP

Se omite la etiqueta de cierre de PHP: (?>) al final de los archivos: *module*, *inc*, *install*, como se muestra en la **figura 14**.

```
1 <?php  
2 function gesecc_menu() {  
3     $items['reportes/eventos'] = array(  
4         'title' => t('Reportes de planes de eventos'),  
5         'base' => 'reportes/eventos',  
6         'description' => t('Tipo de contenido revista.'),  
7         'has_title' => TRUE,  
8         'title_label' => t('Título de la revista'),  
9         'locked' => TRUE,  
10    ),  
11    );  
12 }
```

Figura 14. Uso de etiquetas PHP (Elaboración propia).

3.3.8. Uso de punto y coma (;) en código PHP

En Drupal se hace obligatorio utilizar el terminador de línea (;) como se muestra en la **figura 15**.



```
51 function gesecc_permission() {  
52     return array(  
53         'ver_reportes' => array(  
54             'title' => t('Ver Reportes'),  
55             'restrict access' => true,  
56         ),  
57     );  
58 }
```

Figura 15. Uso de punto y coma (Elaboración propia).

3.3.9. Nombre de funciones

Los nombres de las funciones deben escribirse en minúsculas, así como las palabras separadas por un guión bajo. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. Debe incluirse siempre como prefijo el nombre del módulo, como se muestra en la **figura 16**.

```
60 function gesecc_block_info() {
```

Figura 16. Nombre de funciones (Elaboración propia).

3.3.10. Nombre de variables

Las variables se deben nombrar en minúsculas y las palabras deben estar separadas por un guión bajo, como muestra la **figura 17**.

```
1102 $array_paises = array("Elige tu pais",
```

Figura 17. Nombre de variables (Elaboración propia).

3.4. Plan de iteraciones

El plan de iteraciones de la presente investigación establece las iteraciones necesarias para desarrollar el producto y define qué requisitos se deben implementar en cada una de estas. El objetivo de cada iteración



es obtener una versión funcional del sistema que, aunque no cuente con todos los requisitos definidos, constituya un resultado de valor para el cliente (Joskowicz , 2008).

A continuación, en la **tabla 7** se muestra como queda conformado el plan de iteraciones.

Tabla 7. Plan de iteraciones.

Iteración	Código	Historias de usuario	Estimación (Días)
Iteración 1	HU1	Insertar evento	6/4/2016-11/4/2016
	HU2	Modificar evento	6/4/2016-11/4/2016
	HU3	Eliminar evento	6/4/2016-11/4/2016
	HU4	Generar plan anual de eventos	6/4/2016-11/4/2016
	HU5	Controlar plan de eventos	12/4/2016-14/4/2016
	HU6	Generar reportes del plan anual de eventos	12/4/2016-14/4/2016
	HU7	Insertar misión	12/4/2016-14/4/2016
	HU8	Modificar misión	15/4/2016-19/4/2016
	HU9	Eliminar misión	15/4/2016-19/4/2016
	HU10	Generar plan de misiones	15/4/2016-19/4/2016
	HU11	Controlar plan de misiones	20/4/2016-22/4/2016
	HU12	Generar reporte del plan de misiones	20/4/2016-22/4/2016
	HU13	Mostrar % de cumplimiento del plan de eventos	20/4/2016-22/4/2016
	HU14	Insertar revista	25/5/2016-27/5/2016
	HU15	Modificar revista	25/5/2016-27/5/2016
	HU16	Eliminar revista	25/5/2016-27/5/2016
	HU17	Generar catálogo de revistas	28/4/2016-2/5/2016
Iteración 2	HU18	Eliminar plan anual de eventos	28/4/2016-2/5/2016
	HU19	Exportar el plan anual de eventos en formato pdf y excel	28/4/2016-2/5/2016
	HU20	Exportar el plan de misiones en formato pdf y excel	28/4/2016-2/5/2016
	HU21	Eliminar el catálogo de revistas	3/5/2016-5/5/2016



HU22	Exportar el catálogo de revistas en formato pdf y excel	3/5/2016-5/5/2016
HU23	Crear certificado de participación	3/5/2016-5/5/2016
HU24	Modificar certificado de participación	6/5/2016-10/5/2016
HU25	Eliminar certificado de participación	6/5/2016-10/5/2016
HU26	Generar certificado de participación	6/5/2016-10/5/2016
HU27	Exportar certificado de participación en formato pdf	11/5/2016-13/5/2016
HU28	Listar plan de eventos por temáticas y área	11/5/2016-13/5/2016
HU29	Listar catálogo de revistas por temáticas y los cuatro niveles del MES	11/5/2016-13/5/2016

- La iteración 1 tendrá un total de 4 semanas de duración.
- La iteración 2 tendrá un total de 3 semanas de duración.

3.5. Plan de entregas

Una vez que se concluye la tarea por parte del cliente de elaborar las distintas HU, se comienza con la creación del plan de entrega, para estimar el tiempo de desarrollo de las mismas. Este artefacto se elabora con la intención de fijar qué período de tiempo puede tardar la implementación de cada una de las historias en cada iteración, definiéndose las fechas en que serán liberadas las versiones funcionales del producto. A continuación, en la **tabla 8** se muestra el plan de entrega.

Tabla 8. Plan de entrega.

1ra Entrega (Iteración 1)	2da Entrega (Iteración 2)
3 de mayo de 2016	16 de mayo de 2016

3.6. Tareas de ingeniería

La metodología de desarrollo de software SXP define que la implementación de cualquier sistema debe hacerse a través de iteraciones, donde se obtiene al finalizar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el curso de estas iteraciones se realiza la implementación de las HU definidas por el cliente y descritas en la etapa de planificación por el equipo de desarrollo. Estas HU se



descomponen en tareas de ingeniería que luego son asignadas a los programadores para que sean implementadas en la iteración correspondiente (Quiles Velázquez, y otros, 2013). A continuación, en las **tablas 9 y 10** se detallan algunas de las tareas de ingeniería realizadas para desarrollar requisitos funcionales del sistema de prioridad media, alta y muy alta:

Tabla 9. Tarea de ingeniería 1.

Tarea de ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 6, 14, 23, 29.
Nombre de la Tarea: Instalación y configuración de las librerías fpdf y tcpdf.	
Tipo de tarea : Desarrollo	Puntos estimados: 3 días
Fecha inicio: 8/1/2016	Fecha fin: 12/1/2016
Programador responsable: Adrián Castillo Chávez	
Descripción: Se utilizan las librerías para exportar a pdf.	

Tabla 10. Tarea de ingeniería 2.

Tarea de ingeniería	
Número de Tarea: 2	Número de Historia de Usuario: 5, 7, 13, 17, 22, 24, 28.
Nombre de la Tarea: Instalación y configuración del módulo <i>views</i> .	
Tipo de tarea : Desarrollo	Puntos estimados: 3 días
Fecha inicio: 8/1/2016	Fecha fin: 12/1/2016
Programador responsable: Adrián Castillo Chávez	
Descripción: Se utiliza el módulo para crear las vistas de la aplicación.	

3.7. Validación de la propuesta de solución

Las pruebas son el conjunto de técnicas que permiten determinar la calidad de un producto de software. Se integran dentro de las diferentes fases del ciclo de vida del software dentro de la ingeniería de software. De esta manera, se ejecuta un programa con el objetivo de descubrir los errores que presenta. La calidad de un sistema de *software* es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir.



Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema. Una prueba tiene éxito si descubre un error no detectado hasta entonces (García Sánchez, 2010).

3.7.1. Pruebas funcionales

Se denominan pruebas funcionales o *Functional Testing*, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software (Baydes Gonzáles, 2012).

Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida (Aleman Jimenez, y otros, 2014).

A continuación, en la **tabla 11** se describe el caso de prueba insertar evento. El resto de los casos de prueba se encuentran descritos en los anexos ([ver Anexo 4](#)).

Tabla 11. Caso de prueba 1.

Caso de prueba 1	
Código de caso de prueba:	Nombre de Historia de Usuario: Insertar evento
Nombre de la persona que realiza la prueba: Adrián Castillo Chávez	
Descripción de la prueba: Prueba a la funcionalidad insertar evento	
Entrada / Pasos de ejecución: La entrada consta de la introducción de los datos en los siguientes campos: <ul style="list-style-type: none">• Nombre del evento: Seminario Juvenil de Estudios Martianos• Nivel: UCI• Fecha de envío de resumen: 19/01/2016• Fecha de envío de ponencia: 21/01/2016	



<ul style="list-style-type: none">• Fecha de aceptación: 22/01/2016• Fecha inicio del evento: 23/01/2016• Fecha fin del evento: 24/01/2016• Coordinador de evento: Adrián Castillo Chávez• URL del evento: http://www.sjm.cu• Temática: Martí y la informática• Estado: Realizado• Área: Departamento de Humanidades
Resultado esperado: Se inserta el evento
Evaluación de la prueba: Satisfactoria

3.7.1.1. Resultados de las pruebas funcionales

Las pruebas funcionales aplicadas a la propuesta de solución se basaron en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional. En la primera iteración se detectaron 10 no conformidades, de ellas 5 de validación, 1 de redacción y 3 de error de idioma, las mismas fueron analizadas y corregidas satisfactoriamente. En la segunda iteración se encontraron un total de 4 no conformidades, de ellas 3 de validación y 1 de error de idioma, las cuales fueron mitigadas con éxito. Se realizó una tercera iteración donde no se detectaron no conformidades, siendo este el resultado final, lo que demuestra que el software funciona correctamente. A continuación, en la **figura 18** se muestran los resultados obtenidos en cada iteración de prueba al módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas.

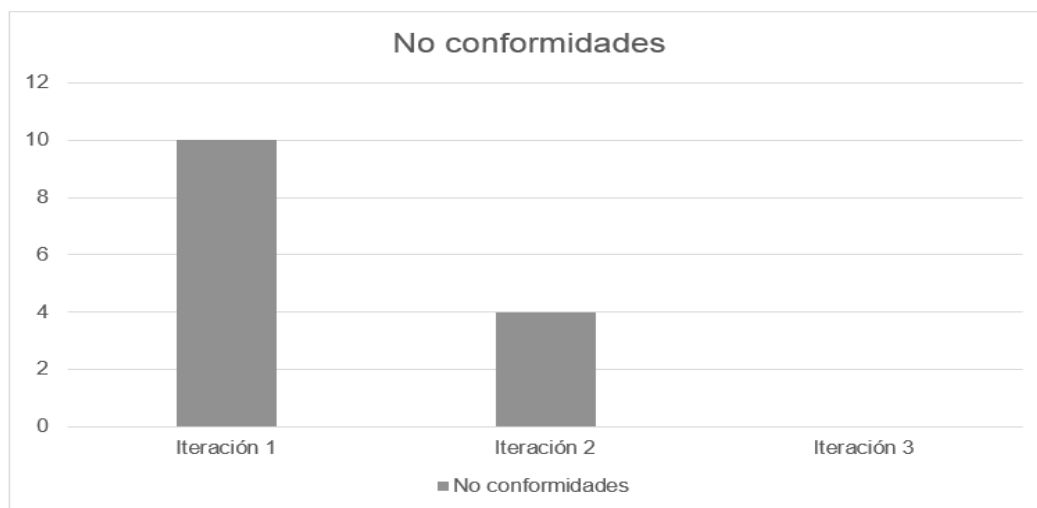


Figura 18. Comportamiento de las no conformidades por iteración (Elaboración propia).

3.7.2. Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y, además, transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2005).

A partir de que se le realizaron las pruebas funcionales a la propuesta de solución, se pudo verificar que las funcionalidades implementadas están en correspondencia con los requisitos funcionales y no funcionales establecidos con anterioridad. A partir de este punto de la investigación se hace necesario verificar la compatibilidad y el funcionamiento de las interfaces que comunican los componentes de la propuesta de solución, para ello se realizan las pruebas de integración.

3.7.2.1. Resultados de las pruebas de integración

Partiendo de que la propuesta de solución es un módulo que debe integrarse a un sistema para la gestión del plan de CTI, se probó la integración del módulo que constituye la propuesta de solución de la presente investigación, al sistema mencionado anteriormente. Se pudo entonces constatar que los tipos de contenidos evento, revista, misión, así como las alertas y certificados se crean correctamente. Las pruebas



realizadas reflejan que el módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración.

3.7.3. Pruebas de seguridad

Las pruebas de seguridad garantizan que los usuarios estén restringidos a funciones específicas o que su acceso esté limitado únicamente a los datos que están autorizados a acceder. Sólo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funcionalidades disponibles. El objetivo fundamental de este tipo de pruebas es comprobar los niveles de seguridad lógica del sistema.

Para evaluar la seguridad de la propuesta de solución, en la presente investigación se desarrollan pruebas utilizando la herramienta *Acunetix Web Vulnerability Scanner 8.0*. A continuación, en la **tabla 12** se muestra el escaneo de vulnerabilidades para pruebas de seguridad aplicadas a la propuesta de solución.

Tabla 12. Resultado del escaneo de vulnerabilidades para pruebas de seguridad.

Categoría de vulnerabilidades	Cantidad
Listado de directorios	27
Posibles directorios sensibles	1
Posibles archivos sensibles	4
Método TRACE ⁹ activado	1

3.7.3.1. Resultados de las pruebas de seguridad

Los resultados obtenidos con las pruebas aplicadas demuestran que las vulnerabilidades encontradas dependen del servidor *web* donde está alojado el módulo y no corresponden a problemas críticos de seguridad del propio módulo. Para solucionar estos problemas se realizaron las siguientes acciones.

- Para desactivar el método TRACE se añadió al final del fichero de configuración de apache la siguiente línea: *TraceEnable off*.

⁹ TRACE: El uso de este método puede conllevar al acceso de la información en las cabeceras http, como *cookies* y datos de autenticación.



- Para prevenir los listados de directorios se removió la palabra clave *Indexes* de cada directiva *Options* en el fichero de configuración de apache, utilizando la siguiente línea: *Options -Indexes*.
- Para impedir que se pueda acceder a archivos con contenido sensible se añadió al fichero de configuración de apache la configuración que se muestra a continuación en la **figura 19**:

```
<FilesMatch "(\\.bak|config|dist|fla|inc|ini|log|psd|sh|sql|swp)|~)$">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</FilesMatch>
```

Figura 19. Configuración para proteger archivos sensibles (Elaboración propia).

3.7.4. Pruebas de carga y estrés

Pruebas de Carga: Estas pruebas se utilizan para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante; se simula la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales (Berges Pedrianes, y otros, 2013).

Pruebas de estrés: Estas pruebas están enfocadas a evaluar cómo el sistema responde bajo condiciones anormales. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible) (Berges Pedrianes, y otros, 2013).

Para la realización de las pruebas de carga y estrés se utilizó la herramienta Apache JMeter 2.12 y un ambiente de prueba con las siguientes características:

- Sistema Operativo: Ubuntu 12.04
- Microprocesador: Intel(R) Core(TM) i5 CPU M 480 @2.67 GHz (4 CPUs)
- Memoria RAM: 4GB
- Disco Duro: 500 GB



La validación realizada consistió en definir dos pruebas de 25 y 50 usuarios. Se definió una lista de enlaces a los que se simuló el acceso aleatorio y a partir de ahí, se recolectaron los datos necesarios para su interpretación.

El análisis

- Se analizaron los resultados a través de un intervalo de confianza¹⁰ con un nivel de confianza al 95%.
- Para el análisis, dado que la muestra se considera grande, no se requiere hacer la suposición de que la muestra tiene una distribución normal ya que por el Teorema Central del Límite (TCL), para n grande implica que X tiene una distribución aproximadamente Normal sin importar la naturaleza de la distribución poblacional (Devore, 2010).

3.7.4.1. Resultados de las pruebas de carga y estrés

Prueba No.1

La primera prueba fue realizada el día 9/05/2016 a las 3:00 pm, se configuró el entorno de prueba para 25 usuarios, cada “1 segundo”. Los valores obtenidos por la herramienta de prueba Apache JMeter se muestran a continuación en la **figura 20**.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
606 /predefe...	25	6092	6091	9798	2215	9846	0,00%	2,3/sec	10,8
641 /predefe...	25	5080	4866	6274	4150	6492	0,00%	1,7/sec	7,3
676 /predefe...	25	4008	4106	4614	2694	5423	0,00%	1,8/sec	7,4
779 /predefe...	25	1455	1543	1923	770	2035	100,00%	2,2/sec	21,3
813 /predefe...	25	1331	1163	1957	676	2062	100,00%	2,4/sec	23,7
914 /predefe...	25	1295	1407	1988	363	2214	100,00%	2,9/sec	28,9
Total	150	3210	2214	6274	363	9846	50,00%	6,6/sec	47,1

Figura 20. Valores obtenidos para una muestra de 25 usuarios.

¹⁰ Se llama *intervalo de confianza* a un intervalo de valores alrededor de un parámetro muestral en los que, con una probabilidad o nivel de confianza determinado, se situar á el parámetro poblacional a estimar.



Como puede verse, el tiempo promedio para acceder a una página es 3,210 segundos, realizándose un total de 150 peticiones al servidor. El tiempo total utilizado para los 25 usuarios se puede calcular con la siguiente fórmula (Díaz, y otros, 2010):

$$\text{Tiempo total} = \# \text{Muestras} * \text{Media} = 150 * 3210 = 481500 \text{ milisegundos}$$

El tiempo total requerido por cada usuario, se puede calcular de la siguiente manera (Díaz, y otros, 2010):

$$((\text{Tiempo total}/1000)/60)/ \text{cantidad de usuarios} = ((481500/1000))/60/25 = 0,321 \text{ minutos}$$

Análisis realizado

En primer lugar, se evaluaron los resultados obtenidos a través de un intervalo de confianza del 95% para muestras grandes. La fórmula del mismo es la siguiente (Díaz, y otros, 2010):

$$[TP - Z_{0,95} * S/\sqrt{n}; TP + Z_{0,95} * S/\sqrt{n},]$$

Donde:

- Tiempo promedio (TP) de respuesta es: 10254
- Estimador de Desvío (S) es: $\sqrt{(\sum_{i=1}^n (X_i - X)^2)/n - 1} = 3042$
- Tamaño de la muestra (n) es de:150
- $Z_{0,95} = 1,96$

El intervalo resultante es el siguiente:

$$[3210 - 1,96 * 3042/\sqrt{150}; 3210 + 1,96 * 3042/\sqrt{150}]$$

$$[3210 - 487; 3210 + 487]$$

$$[2723; 3697]$$



Esto significa que, puede esperarse un tiempo de respuesta promedio entre 8,723 y 3,697 segundos para una cantidad de 25 usuarios simultáneos realizando 150 solicitudes.

Prueba No.2

La segunda prueba fue realizada el día 9/05/2016 a las 3:30 pm, se configuró el entorno de prueba para 50 usuarios, cada "1 segundo". Los valores obtenidos por la herramienta de prueba Apache JMeter se muestran a continuación en la **figura 21**.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
606 /predefe...	50	13785	17324	19171	1551	19494	0,00%	2,5/sec	11,8
641 /predefe...	50	9630	10282	11296	1694	11618	0,00%	1,7/sec	7,4
676 /predefe...	50	7920	7873	8596	4020	9909	0,00%	1,4/sec	5,9
779 /predefe...	50	3289	3342	3855	2500	4122	100,00%	1,4/sec	13,9
813 /predefe...	50	3059	3064	3466	2234	4156	100,00%	1,5/sec	14,4
914 /predefe...	50	2805	2694	3650	1175	4016	100,00%	1,6/sec	16,1
Total	300	6748	3824	15320	1175	19494	50,00%	6,4/sec	46,1

Figura 21. Valores obtenidos para una muestra de 50 usuarios.

Como puede verse, el tiempo promedio para acceder a una página es 6,748 segundos, realizándose un total de 300 peticiones al servidor. El tiempo total utilizado para los 50 usuarios y el requerido para cada uno se puede calcular de la misma manera que en la prueba 1:

$$\text{Tiempo total} = \#Muestras * Media = 300 * 6748 = 2024400 \text{ milisegundos}$$

$$((\text{Tiempo total}/1000)/60)/ \text{cantidad de usuarios} = ((2024400 /1000))/60/50 = 0,6748 \text{ minutos}$$

Análisis realizado

El análisis para la segunda prueba, fue realizado de manera similar que, en la primera, usando las mismas fórmulas.

Con un intervalo de confianza al 95% para muestras grandes, el intervalo resultante es el siguiente:

$$[6748 - 1,96 * 6377/\sqrt{300}; 6748 + 1,96 * 6377/\sqrt{300}]$$



[6748 – 722; 6748 + 722]

[6026; 7470]

Esto significa que, puede esperarse un tiempo de respuesta promedio entre 6,026 y 7,470 segundos para una cantidad de 50 usuarios simultáneos realizando 300 solicitudes.

3.8. Conclusiones del capítulo

- La confección del diagrama de componentes permitió establecer una mayor comprensión de la arquitectura y funcionamiento de los elementos del módulo.
- La ejecución de pruebas al módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer una aplicación con mayor calidad y seguridad. Como resultado de las pruebas se calculó el tiempo promedio de respuesta siendo este de 6,748 segundos, para un total de 300 peticiones al servidor.



Conclusiones

En la presente investigación se realizó un proceso de desarrollo de software, con el objetivo de lograr un producto de calidad en el tiempo establecido. Una vez completada la investigación, se puede concluir que:

- El estudio de soluciones informáticas existentes, relacionadas con la presente investigación, permitió identificar funcionalidades sirvieron de base para el desarrollo de la solución entre las que se encuentran actualizar estados, asignar premios y realizar búsquedas.
- Como resultado del proceso de diseño se logró definir el ambiente de desarrollo, así como la metodología a utilizar, lo que permitió la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución.
- La obtención de los requisitos funcionales y no funcionales, garantizó que la solución respondiera a las necesidades del cliente y de los usuarios finales, sirviendo de guía para el desarrollo de las distintas funcionalidades del módulo.
- El módulo implementado mejora la gestión de la elaboración y control del plan anual de eventos y del catálogo de revistas científicas para el Vicedecanato de Investigación y Posgrado de la Facultad 1.
- La realización de las pruebas de seguridad permitió detectar y corregir insuficiencias encontradas que comprometían la seguridad del mismo. Las pruebas de integración arrojaron que el módulo funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración. Los errores encontrados durante las pruebas funcionales fueron corregidos satisfactoriamente. Con la aplicación de las pruebas de carga y estrés se pudo calcular el tiempo promedio de respuesta resultando de 6,748 segundos, para un total de 300 peticiones al servidor.



Recomendaciones

Para mejorar el módulo para la gestión del plan anual de eventos y de catálogo de revistas científicas de la Facultad 1 de la UCI, se recomienda:

- Implementar en el módulo la funcionalidad que permita modificar la edición de los eventos, para que los eventos almacenados puedan ser reutilizados a la hora de crear nuevos planes de eventos, de modo que solo sea necesario modificar la fecha y la edición del evento.



Referencias bibliográficas

1. **Acunetix. 2016.** Acunetix Web Vulnerability Scanner. [En línea] 2016. [Citado el: 6 de mayo de 2016.] <http://www.acunetix.com/vulnerability-scanner>.
2. **Alegsa, Leandro. 2010.** Alegsa.com.ar. [En línea] 12 de mayo de 2010. [Citado el: 1 de diciembre de 2015.] <http://www.alegsa.com.ar/Dic/sgbd.php>.
3. **Aleman Jimenez, Yulio y Thomas Sosa, Yoniel Jorge . 2014.** *MÓDULO DE CONFIGURACIÓN PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN*. La Habana : s.n., 2014.
4. **Amaro, S. y Valverde, J. 2007.** *Metodologías Ágiles*. Universidad Nacional de Trujillo. Perú : s.n., 2007.
5. **Anderson, Michael, y otros. 2014.** *An integrated development environment for faster feature engineering*. 2014.
6. **Apache Software Foundation. 2016.** Apache Software Foundation. [En línea] 2016. [Citado el: 6 de mayo de 2016.] <http://jmeter.apache.org/>.
7. **Arredondo Morales, Perla Azucena. 2009.** Monografías. *Servidores web*. [En línea] 2009. [Citado el: 1 de diciembre de 2015.] <http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>.
8. **Asociación de Bibliotecas de los Estados Unidos. 2016.** American Library Association. [En línea] 2016. [Citado el: 2 de mayo de 2016.] <http://www.ala.org>.
9. **Baydes Gonzáles, Eylin . 2012.** *Librería cliente para desarrollar aplicaciones con el marco de trabajo jWebSocket utilizando el lenguaje de programación Python*. La Habana : s.n., 2012.
10. **Berges Pedrianes, Zaidee y González Castro, Aliané Amelia . 2013.** *Desarrollo de un Portal Web para la Casa Editora Abril*. La Habana : s.n., 2013.
11. **Blake, Arturo. 2007.** *Los procesos de planificación y sus relaciones con los centros de decisión*. 2007.
12. **Borruel, Julià. 2014.** Tutorial Drupal. [En línea] 2014. [Citado el: 12 de febrero de 2016.] <http://www.cursosdrupal.com/content/arquitectura>.



13. **Brenji, Diego Javier. 2012.** *Trabajo Final de Máster en Software Libre.* [En línea] 2012. [Citado el: 30 de Noviembre de 2015.]
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/14890/4/brenjiTFM0612presentacion.pdf>.
14. **Burrows, William. 2010.** Visual Paradigm 8.0. [En línea] 16 de agosto de 2010. [Citado el: 1 de diciembre de 2015.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.
15. **Cardoso Said, Yadira y Barrera Garrido, Gladis De La Caridad. 2013.** *Solución informática para gestionar la solicitud de participación de eventos científicos externos a la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2013.
16. **Cascading Style Sheets. 2013.** Guía Breve de CSS. [En línea] 2013. [Citado el: 1 de diciembre de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
17. **Cortes, Carlos, y otros. 2013.** Metodologías ágiles: Metodología XP. [En línea] 6 de marzo de 2013. [Citado el: 7 de mayo de 2016.] <http://es.slideshare.net/LisPater1/metodologias-agiles-xp>.
18. **Devore, Jay L. 2010.** *Probabilidades y Estadísticas para Ingeniería y Ciencias.* Sexta. 2010.
19. **Diaz, F. Javier, y otros. 2010.** *Usando JMeter para pruebas de rendimiento.* La Plata : s.n., 2010.
20. **Drupal. 2016.** Coding standards | Drupal.org. [En línea] 2016. [Citado el: 16 de marzo de 2016.] <https://www.drupal.org/coding-standards>.
21. **Drupal Hispano. 2011.** Drupal Hispano. [En línea] 2011. [Citado el: 5 de diciembre de 2015.] <http://drupal.org.es/caracteristicas>.
22. **Enamorado, Abel de la Rúa. 2014.** *Comunidad virtual de Drupal.* 2014.
23. **Ferrer, Carlos Alberto. 2013.** *GUÍA DE ELABORACIÓN DE MODELOS CONCEPTUALES.* 2013.
24. **Fraser, Simon. 2014.** Public Knowledge Project. [En línea] 2014. [Citado el: 30 de Noviembre de 2015.] <http://pkp.sfu.ca/?q=ocs>.
25. **García Sánchez, Alexander . 2010.** *Implementación del módulo de gestión de inventario.* La Habana : s.n., 2010.



26. **García, Saldaña. 2005.** *Análisis y crítica de la metodología para la realización de planes regionales en el estado de Guanajuato.* Guanajuato : s.n., 2005.
27. **Gauchat, Juan Diego. 2012.** *El gran libro de HTML5, CSS3 y Javascript.* Barcelona : s.n., 2012.
28. **González, Liliana, y otros. 2015.** *Campo de investigación en tecnologías de información y comunicación: estrategia de gobernanza en la Universidad de Medellín.* Medellín : s.n., 2015.
29. **Joskowicz , José. 2008.** *Reglas y prácticas en extreme programming.* 2008.
30. **Kruchten, Philippe. 2004.** *El proceso unificado racional: Una introducción.* 2004. ISBN 0-321-19770-4.
31. **Marca Huallpara , Michael Hugo y Quisbert Limachi , Susana Nancy. 2013.** *Diagrama de Despliegue.* 2013.
32. **Márquez, Javier. 2013.** Patrones GoF. [En línea] Enero de 2013. [Citado el: 7 de mayo de 2016.] <https://infow.wordpress.com/category/patrones-de-disenogof/page/2/>.
33. **Martínez Guaita, Alvaro. 2012.** Desarrolloweb.com. [En línea] 21 de febrero de 2012. [Citado el: 1 de diciembre de 2015.] <http://www.desarrolloweb.com/actualidad/nuevo-apache-http-server-2-4-6573.html>.
34. **Netbeans. 2015.** Netbeans. [En línea] 2015. [Citado el: 1 de diciembre de 2015.] <https://netbeans.org/community/releases/80/>.
35. **Nicolás, Bianco. 2015.** Consejo de Desarrollo Científico y Humanístico. [En línea] 2015. [Citado el: 2 de mayo de 2016.] <http://cdch-ucv.net/programa-de-financiamiento-para-la-formacion/subprogramas/eventos-cientificos/>.
36. **Ortigoza, Tamar y Calles, Isabel. 2012.** Propuesta de Reglamento para Eventos Científicos presentado ante la Dirección de Investigación y Posgrado de la UNEFA (DIP). [En línea] 2012. [Citado el: 2 de mayo de 2016.] <https://unefazuliasecretaria.files.wordpress.com/2011/04/lineamientos-eventos-cientificos-dip.pdf>.
37. **Paulsen, Karl. 2011.** *Moving Media Storage Technologies: Applications & Workflows for Video and Media Server Platforms.* 2011. 978-0240814483.



38. **Peñalver Romero, Gladys Marsi, García de la Puente, Sergio Jesus y Menesses Abad, Abel. 2010.** *SXP, Metodología ágil para el desarrollo de software.* La Habana : s.n., 2010.
39. **Pérez Reyes, Carlos Miguel y Argilagos Yi, José Miguel. 2012.** Plataforma electrónica para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas. [En línea] <http://publicaciones.uci.cu/index.php/SC/article/view/913..>
40. **Ponjuán, Gloria. 2003.** *Gestión de la información y la documentación en las organizaciones.* La Habana : s.n., 2003.
41. **PostgreSQL. 2016.** [En línea] 2016. [Citado el: 6 de mayo de 2016.] <http://www.postgresql.org/about/>.
42. **Pressman, Roger. 2007.** *Ingeniería del Software un enfoque práctico. Estrategia de pruebas. Quinta Edición.* 2007. ISBN: 97-0105-473-3.
43. **Prytherch, Ray. 2000.** Harrod's Librarian's Glossary and Reference Book. 2000.
44. **Quiles Velázquez, Rafael Alberto y Aguilera Reyes, Amalia. 2013.** *Sistema de identificación de personas basado en rasgos faciales.* La Habana : s.n., 2013.
45. **Real Academia Española. 2016.** Real Academia Española. [En línea] 2016. [Citado el: 2 de mayo de 2016.] <http://dle.rae.es/?id=TIvEXgq>.
46. **Rodríguez Muñoz, Wendy . 2015.** *Portal Web de la Facultad 1 de la Universidad de las Ciencias.* 2015.
47. **Rodríguez, Milena. 2010.** *Que es UML y para que sirve.* Ecuador : Universidad Autónoma de Ecuador, 2010.
48. **Rosabal, Antonio. 2011.** *Sistema para el control de actividades científicas.* Las Tunas : s.n., 2011.
49. **Salgueiro, Amado. 2001.** *Indicadores de Gestión y Cuadro de Mando.* 2001.
50. **Sevilla, Universidad de. 2014.** *Patrones de Asignación de Responsabilidades (GRASP).* Sevilla : s.n., 2014.



51. **Silvente Lao, Yoel Joaquin . 2015.** *Módulo para la detección de contenido duplicado en portales.* 2015.
52. **Sommerville, Ian. 2005.** *Ingeniería del software 7ma edición.* 2005.
53. **Sousa, Martínez de. 2002.** *Diccionario de Bibliología.* 2002.
54. **Thibaud, Cyril. 2006.** *MySQL 5.* Barcelona : Amadeu Brugués, 2006. ISBN:2-7460-3069-1.
55. **Ulacia Díaz, Yoandy . 2009.** *Plataforma de Gestión de Contenidos para Dispositivos.* 2009.
56. **Villamizar Suaza, Katerine . 2013.** *Definición de equivalencias entre.* Universidad Nacional de Colombia. Colombia : s.n., 2013.
57. **Xiaogang, Zhang . 2016.** International Organization for Standardization. [En línea] 2016. [Citado el: 2 de mayo de 2016.] <http://www.iso.org>.



Bibliografía Consultada

- 1 Diaz, Javier, Tzancoff Banchoff, Claudia, Rodríguez, Anahí y Soria, Valeria. Usando Jmeter para pruebas de rendimiento, 2013.
- 2 Poloni Izeta, Claudio. Estadística Descriptiva: Análisis de Datos en Secciones Transversales, 2012.
- 3 Álvarez , Miguel Angel. Desarrollo *web*. Qué es un CMS, 2008.
- 4 Butcher, M. Y Dunlap, G. Drupal 7 Module Development, 2010.
- 5 Rodríguez, F. G. Experto en Drupal. Nivel Avanzado, 2012.
- 6 Rodríguez, F. G. Experto en Drupal. Nivel Intermedio, 2012.
- 7 Rodríguez, F. G. Experto en Drupal. Nivel Inicial, 2012.
- 8 Tomlinson, T. Beginning Drupal 7, 2010.
- 9 Tomlinson, T. y Vandyk, J. K. Pro Drupal 7 Development, 2010.
- 10 Anegón Moya, Félix. Principales indicadores cuantitativos de la actividad científica chilena, 2014.
- 11 Carvajal, José Carlos. Metodologías Ágiles: Herramientas y Modelo de desarrollo para aplicaciones, 2008.
- 12 Casas, Sandra, Reinaga, Héctor, Un enfoque basado en las tarjetas CRC, 2009.
- 13 Tomlinson, John, Vandyk, Todd. Pro Drupal 7 Development, 2010.
- 14 Rumbaugh, James, Jacobson, Ivar y Booch, Grady. Metodologías para el desarrollo de software, 2014.
- 15 Fernández, M. Ejemplos de uso de Drupal para el desarrollo de aplicaciones *web*, 2008.
- 16 Alonso, J. Curso de metodología de la investigación, 1998.
- 17 Pérez, J. Introducción a CSS, 2008.
- 18 Cambra, Pedro. El libro de Drupal en español, 2009.



- 19 Gil, García. El gran libro de Drupal 7, 2011.
- 20 Sampieri, Hernández. Metodología de la investigación, 2011.



Glosario de términos

CSRF: Hace alusión a la falsificación de peticiones en sitios cruzados. Consiste en un tipo de *exploit* malicioso que transmite comandos no autorizados por un usuario en el cual un sitio *web* confía.

Estandarización: Uniformidad de la documentación generada.

Extensiones: Son librerías externas al núcleo de PHP, y que pueden ser utilizadas para múltiples propósitos.

GPL: Acrónimo de *General Public Licence*, licencia de software libre.

Internet: Conjunto descentralizado de redes de comunicación que se conectan entre si y utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Servidor *web*: Programa informático que espera permanentemente las solicitudes realizadas por clientes *web*, mediante el protocolo HTTP, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente; generando respuestas en cualquier lenguaje o aplicación del lado del cliente.



Acrónimos

A	ALA: Asociación de Bibliotecas de los Estados Unidos. ARC: Área de Resultados Clave.
C	CALISOFT: Centro Nacional de Calidad de Software. CASE: Ingeniería de Software Asistida por Ordenador. CESOL: Centro de Software Libre. CIDI: Centro de Ideoinformática. CMS: Sistema de gestión de contenidos. CISED: Centro de Identificación y Seguridad Digital. CSS: Hojas de estilo en cascada. CRC: Tarjeta Clase-Responsabilidad-Colaboración. CTI: Ciencia e Innovación Tecnológica.
D	DEPSW: Departamento de software.
F	FCT: Facultad de Ciencias Técnicas.
G	GRASP: Patrones de asignación de responsabilidades. GOF: Banda de los Cuatro.
H	HTML: Lenguaje de marca de hipertexto. HU: Historias de usuario.
I	IDE: Entorno de Desarrollo Integrado, también conocido como entorno de diseño integrado o entorno de depuración integrada. ISO: Organización Internacional de Estandarización.
M	MES: Ministerio de Educación Superior. MVCC: Control de Concurrencia multi-Versión.
O	OCS: <i>Open Conference Systems.</i>
P	PHP: Pre-procesamiento de hipertexto. PKP: <i>Public Knowledge Project.</i>
R	RF: Requisitos Funcionales. RNF: Requisitos no Funcionales. RUP: Proceso Unificado de <i>Rational.</i>



S	SGBD: Sistema de Gestor de Base de Datos. SGCACD: Sistema de Gestión para el Control de Actividades Científicas de los Docentes. SQL: Lenguaje de Consulta Estructurado. SXP: Scrum y XP.
T	TIC: Tecnologías de la Información y las Comunicaciones.
U	UCI: Universidad de las Ciencias Informáticas. ULT: Universidad de Las Tunas. UML: Lenguaje Unificado de Modelado.
X	XP: Programación Extrema.