



Facultad 3

Desarrollo de las funcionalidades de solicitud, instalación y configuración de eventos en la plataforma electrónica GRECIA de gestión de eventos científicos.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autora

Arianna Galán Quiala

Tutor

Ing. Carlos Miguel Pérez Reyes

Co-tutor

Ing. José Miguel Argilagos Yi

La Habana, Junio de 2013

Año 54 de la Revolución

Declaración De Autoría

Declaro ser autora del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de junio del año 2013.

Arianna Galán Quiala
Autora

Ing. Carlos Miguel Pérez Reyes
Tutor

Ing. José Miguel Argilagos Yi
Co-tutor

Agradecimientos

A mi familia por darme su apoyo y confiar en mí en cada momento de la carrera. A José Luis Sablón Mora por la invaluable ayuda que me ha brindado durante estos cinco años. A mis amigos y amigas que durante todos estos años han compartido junto a mí. En especial Arletys, Polanco, Abel, Jannet y Addiel por haberme acompañado en los momentos buenos y malos de la carrera. A todas las personas que contribuyeron en mi formación, especialmente a mis tutores y profesores por el trabajo que han realizado.

Dedicatoria

A mis padres por confiar en mí y brindarme su apoyo incondicionalmente durante estos 5 años de carrera, por darme ánimos aun cuando están lejos de mí físicamente, pero espiritualmente siempre los llevo conmigo, por sacrificarse por mí, les agradezco todo lo que soy hasta este momento y lo que puedo llegar a ser. A mi mamá por brindarme siempre su amor y cariño y estar ahí siempre que la necesite, por darme la fuerza de seguir adelante. A mi padrastro, por verme criado como su hija y por darme su apoyo incondicional. A mi hermana y hermano por ser tan especial es para mí y por estar a mi lado en cada momento de mi vida. A mi bisabuela por el amor y el cariño que me ha entregado. Se la dedico especialmente a mi abuela que durante el tiempo que estuvo a mi lado supo darme su amor, cariño, y quererme con mis virtudes y mis defectos. Los quiero.

Resumen

La Universidad de las Ciencias Informáticas, como parte de la estrategia de informatización de la gestión de eventos científicos, emplea la plataforma electrónica GRECIA. La presente investigación surge al existir dificultades en la forma en que se realizan los procesos de solicitud, instalación y configuración de nuevos eventos en dicha plataforma, evidenciadas fundamentalmente, en la lentitud e ineficacia de estos procesos y la dependencia de los usuarios finales con el equipo de desarrollo. El objetivo de la investigación es desarrollar las funcionalidades de solicitud, instalación y configuración, que permitan a los usuarios realizar estos procesos sin depender del equipo de desarrollo y de manera amigable. Para llevarlo a cabo, se realizó un estudio del estado del arte sobre sistemas que se enfoquen en la gestión de solicitud, instalación y configuración de un sitio web, que nutrido con entrevistas a especialistas de la Dirección de Investigaciones de la UCI, permitió especificar los requerimientos necesarios. Se determinó realizar el desarrollo con una arquitectura modular, quedando finalmente definido el desarrollo del módulo y el perfil que se integran a la plataforma (Módulo de Gestión de Solicitudes, instalación y configuración del perfil). El desarrollo fue guiado por el proceso de desarrollo de *software* con enfoque ágil y CMMI nivel 2 y utilizando como tecnología base Drupal 7. Finalmente, la solución fue sometida a diferentes pruebas que validan el cumplimiento de los requerimientos definidos arrojando resultados satisfactorios.

Palabras clave: configuración, gestión de eventos científicos, instalación, Plataforma GRECIA, solicitud.

Contenido

Introducción	1
Capítulo 1: Fundamentación teórica	5
1.1 Marco Teórico	5
1.1.1 Sistema de Gestión	5
1.1.2 Solicitud	5
1.1.3 Instalación.....	5
1.1.4 Configuración.....	5
1.2 Soluciones similares.....	6
1.3 Ambiente de desarrollo.....	7
1.3.1 Proceso para el desarrollo de <i>software</i> con un enfoque ágil y CMMI nivel 2.....	7
1.3.2 Sistema de Gestión de Contenidos.....	9
1.3.3 Lenguajes programación.....	12
1.3.4 Lenguajes del lado del cliente.....	12
1.3.5 Servidor Web.....	14
1.3.6 Entornos de Desarrollo Integrado (IDE)	15
1.3.7 Sistema Gestor de Base de Datos	15
1.3.8 Herramienta de Diseño	16
Conclusiones del capítulo.....	17
Capítulo 2. Descripción de la solución propuesta.....	19
2.1 Modelo de Dominio.....	19
2.2 Descripción de la propuesta de solución	20
2.3 Captura de requerimientos del <i>software</i>	22
2.3.1 Mecanismo empleado para la captura de requisitos.....	22
2.4 Especificación de Requisitos	22
2.4.1 Requisitos funcionales.....	23
2.4.2 Requisitos no Funcionales	23
2.4.3 Validación de requisitos	26
2.4.4 Actores del sistema.....	26
2.4.5 Descripción de requisitos	27

2.6 Diagrama de paquete	28
2.6 Seguridad de la solución propuesta.....	30
2.7 Descripción de estilos arquitectónicos.....	30
2.8 Patrones de diseño.....	32
Conclusiones del capítulo.....	34
Capítulo 3. Implementación y validación	35
3.1 Estándares de codificación.....	35
3.2 Modelo de Implementación.....	38
3.3 Diagrama de componentes.....	40
3.4 Diagrama de despliegue del sistema	42
3.5 Validación.....	43
3.5.1 Pruebas de <i>software</i>	43
3.5.2 Pruebas funcionales.....	44
3.5.3 Descripción de caso de pruebas	44
3.5.4 Pruebas de carga y estrés	47
3.5.5 Validación de las variables.....	49
Conclusiones del capítulo.....	50
Conclusiones	51
Recomendaciones	52
Bibliografías Referenciadas	53
Anexo. Entrevista.....	55
Anexo. Descripción de los requisitos.....	56
Anexo. Descripción de casos de pruebas	63

Introducción

La sociedad científica moderna enfrenta grandes desafíos donde las Tecnologías de la Información y las Comunicaciones (TIC) desempeñan un papel fundamental. El impacto tecnológico sobre el conocimiento científico se evidencia en la utilización de recursos geográficamente distribuidos como son las revistas científicas electrónicas, las redes sociales académicas, eventos virtuales, entre otros.

Dentro de este escenario, los eventos científicos tradicionales (sin soporte tecnológico) son menos frecuentes. En sustitución, la gran mayoría de los eventos científicos de la actualidad son soportados por sistemas informáticos que colaboran en la socialización del conocimiento y en la gestión de parte importante de los procesos que en estos se llevan a cabo.

Según estudios recientes, la industria de las conferencias en línea y los eventos virtuales ha crecido más del doble en 2008-2009 (1), y se estima que su volumen de negocio será de 18,6 billones de dólares en el periodo 2010-2015 (2). Pese a este impactante dato, en Cuba el desarrollo de los eventos científicos se ve afectado en ocasiones por no tener herramientas eficaces que permitan emplear al máximo las ventajas de las TIC.

La Universidad de las Ciencias Informáticas (UCI), fue creada en el año 2002, como institución fundamental dentro del arduo proceso de informatización de la sociedad cubana (3), siendo parte además del comienzo de una nueva era de producción de *software* para el mercado nacional e internacional (4). En la UCI se desarrolla la segunda versión de una plataforma electrónica para la gestión de eventos científicos, denominada GRECIA. Esta, lleva a cabo de forma simultánea la gestión de un conjunto de eventos científicos, prestando servicio a una creciente demanda institucional y nacional a través de solicitudes.

Sin embargo, la plataforma no satisface las necesidades actuales: las funcionalidades de solicitud, instalación y configuración de un evento solo se pueden realizar por personal del equipo de desarrollo debido a su complejidad, provocando demoras en cada uno de estos procesos. En la actualidad las solicitudes se realizan mediante carta a la institución o por correo electrónico, trayendo consigo falta de organización, pérdida de la información, desconocimientos por parte del solicitante del estado de su solicitud y demora en todos los procesos.

Luego de aprobada una solicitud no es posible realizar la instalación y configuración de un nuevo evento por parte del solicitante, ya que estos procesos son llevados a cabo por personal con vastos conocimientos en las tecnologías empleadas y con acceso pleno a los recursos disponibles para su

alojamiento. Debido a que no es posible otorgar privilegios a los solicitantes para acceder a estos recursos y no poder garantizar además, los conocimientos de los clientes sobre las tecnologías empleadas, este proceso se hace exclusivamente por el equipo de desarrollo esto provoca dependencia del equipo de desarrollo y cada vez que el solicitante desee realizar algún cambio tiene que solicitarlo. Incluso proporcionando acceso a los solicitantes para realizar los cambios, los cuales no están necesariamente capacitados para utilizar las tecnologías en las cuales está desarrollada la plataforma.

GRECIA está soportada por Drupal 7 y hace uso de los módulos existentes en la comunidad de desarrollo, por lo que resultó necesario realizar estudio preliminar para determinar si existían módulos que solucionaran la problemática descrita de forma parcial o total. Como resultado de este estudio se pudo llegar a la conclusión que no existen módulos que satisfagan las necesidades de la plataforma, ya sea por emplear taxonomías¹, tecnologías y procesos diferentes a los necesarios o por complicar la usabilidad de estos procesos.

Por la problemática antes descrita se define como **Problema a resolver**:

¿Cómo contribuir a la mejora de la gestión de solicitudes, instalación y configuración de los eventos científicos en la Plataforma Electrónica de Gestión de Eventos Científicos GRECIA y mitigar así las dificultades existentes?

El **objeto de estudio** se centra en: Plataformas electrónicas para la gestión de eventos científicos y se especifica como **campo de acción**: Procesos de solicitud, instalación y configuración de eventos en plataformas electrónicas.

Para guiar la investigación se define como **objetivo general**: Desarrollar las funcionalidades de solicitud, instalación y configuración de eventos de la Plataforma Electrónica de Gestión de Eventos Científicos GRECIA, de manera que permita mitigar las deficiencias existentes en la misma.

Para dar solución al objetivo general, el mismo se desglosa en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación a partir del estado del arte sobre los procesos de gestión de solicitudes, instalación y configuración de eventos en plataformas electrónicas de gestión de eventos científicos.

¹ Permite la categorización del contenido a través términos y vocabularios.

- Recopilar los requerimientos del módulo y el perfil a partir de entrevistas a especialistas de la Dirección de Investigaciones y del resultado del estudio de soluciones similares.
- Realizar el análisis, diseño del módulo, perfil a desarrollar y documentar dicho proceso.
- Implementar el módulo y perfil definido.
- Validar el correcto funcionamiento del módulo y el perfil, para que la solución propuesta de cumplimiento al objetivo general de la investigación.

La forma de investigación es desglosada en las siguientes **Tareas de investigación**:

1. Identificación y documentar de las deficiencias existentes en la forma que se llevan a cabo los procesos de solicitud, instalación y configuración de nuevos eventos.
2. Elaboración del marco teórico de la investigación a partir del estado del arte sobre sistemas de gestión de solicitud, instalación y configuración en plataformas de gestión de eventos científicos.
3. Fundamentación de la estrategia a seguir para guiar el desarrollo.
4. Descripción del lenguaje y herramienta de modelado a utilizar.
5. Descripción y caracterización de las tecnologías necesarias para el desarrollo del módulo y el perfil.
6. Recopilación de los requerimientos del *software* a partir de: el estudio de sistemas similares y entrevistas a especialistas de la Dirección de Investigaciones para saber cuáles son sus expectativas del sistema.
7. Documentación de los requisitos para obtener una descripción completa del módulo y perfil a desarrollar.
8. Diseño de los prototipos de interfaz de usuario (IU).
9. Validación de los requisitos a través de los prototipos de interfaz de usuario y revisión del cliente.
10. Descripción de los elementos del diseño arquitectónico.
11. Modelación del sistema a partir de la especificación de requisitos generando los artefactos de entrada basándose en la guía definida para el desarrollo.
12. Implementación del módulo Gestión de solicitud de eventos en la plataforma de gestión de eventos.
13. Implementación del perfil de instalación y configuración de eventos.
14. Realización de pruebas funcionales de caja negra.

Se plantea como **Idea a defender**:

Desarrollando las funcionalidades para la gestión de solicitudes, instalación y configuración de los eventos científicos en la Plataforma Electrónica GRECIA, se permite realizar solicitudes de forma más eficiente e instalar y configurar un nuevo evento sin ser parte del equipo de desarrollo.

Los **métodos científicos** que se utilizan para darle solución a los objetivos propuestos son:

Métodos Teóricos:

- Analítico-Sintético: Permite analizar los sistemas similares relacionados con la gestión de solicitud, instalación y configuración así como identificar características generales de los mismos aplicados al contexto de la investigación.
- Inductivo-Deductivo: Permite asumir posturas sobre las metodologías de desarrollo de *software* a partir del razonamiento lógico y tomando como base los estudios encontrados en la bibliografía consultada.
- La validación: Permitió comprobar el correcto funcionamiento del sistema mediante la realización de las siguientes pruebas: caja negra y carga de estrés.

Métodos Empíricos:

- Entrevista: se utilizó para entrevistar al personal de la Dirección de Investigaciones, con el objetivo de profundizar en el problema y recopilar los requisitos que debe cumplir el sistema para obtener la solución óptima.

El presente documento se estructura por 3 capítulos, a continuación se expone una breve descripción de los mismos.

En el **Capítulo 1** se aborda todo lo relacionado con la fundamentación teórica que sustenta la presente investigación. También el estudio del arte así como de las herramientas, lenguaje de modelado y metodología a emplear.

Durante el desarrollo del **Capítulo 2** se analizan las características de la solución propuesta y se describen los flujos fundamentales. En el mismo se definen los requisitos del sistema y se elaboran una descripción completa del comportamiento del mismo así como la arquitectura y los patrones a utilizar.

Durante el desarrollo del **Capítulo 3** se analizan todos los procesos referentes al desarrollo de la solución y las pruebas realizadas para validar el correcto funcionamiento de la aplicación.

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se describen, los principales conceptos y se realiza un estudio del estado del arte relacionado con las soluciones informáticas existentes que tienen vínculo con el objeto de estudio, así como las tecnologías que se van a emplear y las herramientas a utilizar para el desarrollo del trabajo.

1.1 Marco Teórico

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

1.1.1 Sistema de Gestión

Herramienta de gestión sistemática y transparente que permite dirigir, controlar y evaluar el desempeño institucional en términos de calidad y satisfacción social en la prestación de los servicios. Para dicha investigación hay que tener presente el sistema de gestión debido a que la plataforma electrónica se encarga de gestionar eventos científicos (5).

1.1.2 Solicitud

Es un documento escrito que va dirigido a un organismo público o a una autoridad a los que se pide algo o ante los que se plantea una reclamación con la exposición de los motivos en los que se basan. Mediante la solicitud es que el solicitante va a realizar la petición de crear un nuevo evento científico en la plataforma (6).

1.1.3 Instalación

Es utilizada para la instalación de los módulos del sistema que se encargan de darle funcionalidades al mismo siguiendo un orden lógico, se utiliza la instalación para instalar los perfiles de los usuarios (7).

1.1.4 Configuración

Es realizar cambio a la visualización del sitio, se puede realizar mediante el cambio de la imagen del sitio, el estilo de la letra, los permisos asociados a los módulos, tipo de contenidos, los perfiles, etc. Esta acción la personalización del sitio según las necesidades del usuario (7).

1.1.5 Perfil

Los perfiles son plantillas de configuración que se reutilizan tanto en la realización manual de copias de seguridad como en las tareas programadas, donde el usuario realiza cualquier configuración (8).

1.2 Soluciones similares

Se realizó un estudio sobre sistemas similares para identificar la forma en que estos llevan a cabo los procesos de solicitud, instalación y configuración de nuevos eventos. El estudio se centró en las plataformas para la gestión de eventos científicos que más se ajustaban a las características actuales de la plataforma GRECIA. Se tomaron como muestras las plataformas más empleadas según el artículo "Estudio de la Producción Científica y Tecnológica en colaboración Universidad-Empresa" y las sugeridas por la Dirección de Investigaciones de la UCI (9).

Nombre	Solicitud	Instalación	Configuración	Aspectos de interés
Sistema de Gestión de Eventos Universitarios (10)	<p>-Permite la elaboración de formularios de inscripción personalizados que admiten una gestión eficaz de las inscripciones a cualquier tipo de evento.</p> <p>-El sistema permite elegir entre 5 tipos de inscripción: .Registro libre .Registro con validación .Registro con pago (próximamente) .Registro en otra web .Sin registro</p>	<p>-Crear un evento con sus datos básicos es algo que no le llevará al usuario más de 5 minutos. Una vez introducidos los datos básicos se puede continuar editando el evento, publicarlo o dejarlo en modo borrador para seguir más adelante. Título, fechas, descripción básica y lugar de celebración son los datos básicos obligatorios que pueden completarse con organizador, URL y tipo de evento.</p>	<p>-El usuario puede seleccionar una URL corta para su evento que, además tendrá asociada una URL semántica descriptiva que ayudará a mejorar su posicionamiento en buscadores.</p> <p>-El menú de edición y gestión de un evento se divide en 6 bloques que permiten gestionar el contenido de las páginas del evento; el diseño de la página; herramientas de diversa índole que van desde la creación de lugares en el mapa, hasta la</p>	<p>-El organizador, que actúa como administrador con todos los permisos del evento, podrá asignar nuevos roles a otros usuarios, en particular podrá crear otros administradores o crear grupos usuarios con roles más restringidos, tales como el rol miembro del Comité Científico o Comité Organizador.</p>

				inserción de fechas de interés o noticias; la conexión con redes sociales con Facebook o Twitter; la gestión de documentos y la gestión de asistentes.	
Sistema de Conferencia EasyChair (11)	-Mediante un formulario de solicitud para gestionar una conferencia o un taller.	un de para una o un taller.	-La instalación de la conferencia o taller es realizada por los desarrolladores del sistema, el usuario no tiene que instalar.	-La configuración de la conferencia o taller es realizada por los desarrolladores del sistema, el usuario no tiene que configurar.	-Algunas características que presenta es su fácil uso, flexible.

Tabla 1.1 Característica de los sistemas similares.

Se concluye que las plataformas analizadas realizan los procesos de solicitud, instalación y configuración de formas diferentes, pero tienen en común, satisfacer la necesidad de clientes específicos. Estas aportan a la investigación ideas a la hora de realizar la propuesta de solución y definir las funcionalidades con lo que debe contra la plataforma de gestión de eventos científicos, que serán definidas en capítulos posteriores.

1.3 Ambiente de desarrollo.

A continuación se describe el proceso de desarrollo por cual se guía la investigación, utilizando el sistema definido en el marco de trabajo de la plataforma GRECIA.

1.3.1 Proceso para el desarrollo de *software* con un enfoque ágil y CMMI nivel 2.

Un proceso de desarrollo de *software* tiene como propósito la producción eficaz y eficiente de un producto *software* que reúna los requisitos del cliente (12).

Pressman establece un marco común del proceso definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del *software*, con independencia de su tamaño o complejidad (12).

Enfoque ágil:

El desarrollo ágil de *software* es un marco de trabajo conceptual de la ingeniería de *software* que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando *software* en cortos plazos de tiempo. El *software* desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

CMMI (*Capability Maturity Model Integration*)

El SEI (*Software Engineering Institute*) es el instituto que creó y mantiene el modelo de calidad CMM – CMMI, el CMM - CMMI es un modelo de calidad del *software* que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir *software* (13).

Integración de modelos de madurez de capacidades es un enfoque de mejora de procesos que proporciona a las organizaciones los elementos esenciales de procesos efectivos, lo que mejora su rendimiento. El proceso basado en CMMI, incluye la identificación de las fortalezas y debilidades de los procesos de la organización y realiza cambios en el proceso para convertir debilidades en fortalezas.

Este modelo cuenta con cinco niveles:

- Inicial o Nivel 1 CMM - CMMI. Este es el nivel en donde están todas las empresas que no tienen procesos. No hay control sobre el estado del proyecto, el desarrollo del mismo es completamente opaco, no sabes lo que pasa en él.
- Repetible o Nivel 2 CMM – CMMI: la principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.
- Definido o Nivel 3 CMM – CMMI: alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida, quiere decir que está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos.

- Cuantitativamente Gestionado o Nivel 4 CMM – CMMI: los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.
- Optimizado o Nivel 5 CMM – CMMI: los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

Lo que se pretende con el nivel 2 de CMMI es conseguir que en los proyectos de la organización haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados. Está compuesto por 7 áreas de procesos:

- Administración de requisitos.
- Medición y Análisis.
- Monitoreo y control de acceso.
- Planeación del proyecto, tareas de investigación asignadas, las reuniones frecuentes con el cliente.
- Aseguramiento de la calidad del proceso y el producto.
- Administración de acuerdos con proveedores.
- Administración de la configuración.

La utilización del proceso para el desarrollo de *software* con un enfoque ágil y CMMI nivel 2, es factible emplearlo en la investigación debido que el equipo de desarrollo es pequeño, se genera poca documentación y artefactos, además el cliente forma parte del equipo de desarrollo permitiendo una mayor retroalimentación de los avances del proyecto y se obtiene una solución a corto plazo. Por otra parte, este proceso se encuentra guiado por el modelo de mejora de procesos CMMI en su nivel 2, especificado por la Universidad.

1.3.2 Sistema de Gestión de Contenidos.

El Sistema de Gestión de Contenidos (CMS) es un programa desarrollado para crear, editar, manejar y publicar contenidos. Las ventajas más comunes son: completo control sobre el contenido, se puede actualizar el contenido dinámicamente sin conocimientos de programación, fácil integración de nuevas funcionalidades y completo control de usuarios (14).

Drupal es un sistema de gestión de contenido modular y configurable, el cual ofrece numerosas funcionalidades que son imprescindibles para obtener una calidad del *software* a implementar como por

ejemplo publicar artículos, imágenes, u otros archivos además de servicios añadidos ya sea foros, encuestas, votaciones, blogs, administración de usuarios y permisos. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios.

El sitio principal de desarrollo y coordinación de Drupal es drupal.org (15).

Características de Drupal:

- **Objetos de Contenido (Nodos):** el contenido creado en Drupal es, funcionalmente, un objeto (Nodo). 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.
- **Plataforma:** independencia de la base de datos. Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal 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.
- **Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- **Múltiples idiomas y Localización:** Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilinguaje. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como GNU ettext.
- **Administración y Análisis:** La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún *software* adicional.
- **Análisis, Seguimiento y Estadísticas:** Drupal puede mostrar en las páginas web de administración informes sobre enlaces entrantes, popularidad del contenido, o de cómo los usuarios navegan por el sitio.
- **Encuestas:** Drupal incluye un módulo que permite a los administradores y usuarios crear encuestas on-line totalmente configurables.
- **Sistema de Cache:** el mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

Perfil de Instalación

El perfil de instalación es un aspecto de Drupal que se debe tener en cuenta en la investigación, debido a que se va a desarrollar un perfil para la plataforma electrónica GRECIA con sus propias características, a continuación se describe lo que es un perfil de instalación:

Perfil de instalación, es un conjunto de instrucciones que permite definir qué módulos y opciones se activan durante el proceso de instalación de Drupal. De esta forma se pueden crear portales Web sin necesidad de empezar de cero, ganando en tiempo y productividad.

- Los perfiles de instalación permiten montar sitios temáticos. Por ejemplo, se puede encontrar perfiles de instalación específicos para implementar un periódico digital, una web corporativa, un aula virtual, una tienda online, etc.

Estructura de archivos del perfil de instalación

Un perfil de instalación se compone de los siguientes archivos y carpetas:

- archivo .info. Archivo de definición del perfil de instalación. Su contenido es similar al archivo .info de módulos y temas.
- archivo .profile. Es el archivo donde se implementan las tareas que se llevarán a cabo durante la instalación. Es equivalente al archivo .module de los módulos, y en él podremos usar la mayoría de funciones disponibles de la API de Drupal.
- archivo .install. Permite implementar hook_install () para insertar contenido en la base de datos durante la instalación.
- carpeta /modules. En esta carpeta se añaden módulos que estarán disponibles durante la instalación. Generalmente sólo se usa esta carpeta para añadir módulos creados específicamente para la distribución.
- carpeta /themes. En esta carpeta se pueden añadir temas que estarán disponibles durante la instalación. Generalmente sólo se usa esta carpeta para añadir temas creados específicamente para la distribución.
- carpeta /translations. En esta carpeta se añade las traducciones del núcleo, necesarias para que el proceso de instalación se realice en otro idioma (por ejemplo en español).

1.3.3 Lenguajes programación

Los lenguajes del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

PHP

Para el desarrollo de aplicaciones web existen numerosos lenguajes de programación en el lado del servidor, entre los que destacan PHP, Java, Perl, Ruby, Python, entre otros.

Hypertext Pre-processor (PHP) es uno de los lenguajes de programación más populares, que se caracteriza por la gran fluidez y rapidez de sus script. Es un lenguaje interpretado, diseñado originalmente para la creación de páginas web dinámicas, gratuito e independiente de plataforma, con una gran biblioteca de funciones y numerosa documentación(16).

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y portar el sitio desarrollado de un sistema a otro sin prácticamente ningún trabajo. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Su programación es segura y confiable, compatible con las bases de datos más comunes tales como: MySQL y PostgreSQL. Permite aplicar técnicas de programación orientada a objetos. Este lenguaje de programación se encuentra preparado para realizar muchos tipos de aplicaciones web gracias a la extensa biblioteca de funciones con la que está dotado(16). Debido a que Drupal se encuentra escrito en PHP y se necesita que el módulo a implementar no presente inconvenientes cuando se integre a la plataforma de gestión de eventos científicos, se utilizará PHP 5.3.10.

1.3.4 Lenguajes del lado del cliente.

Los lenguajes de lado cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento, entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML.

HTML

Sus siglas vienen de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares los cuales tienen esta forma o <P>. Cada etiqueta tiene un significado, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir hasta cierto punto la forma o estilo que se aplica a un documento HTML, se propone utilizar HTML 5 porque es la versión más avanzada y por todas las ventajas que brinda para realizar sitios web, un ejemplo es que no tienes de necesidad de utilizar plugins para reproducir audio y video(17).

JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor como por ejemplo, un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente los eventos generados por el usuario. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM sus siglas vienen de Document Object Model (Modelo de Objetos de Documento), para el desarrollo del sistema se va emplear la versión 1.8(17).

CSS

Las hojas de estilo en cascada o (Cascading Style Sheets) son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML. Este estilo propone las ventajas: Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.

Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.

El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea), se propone utilizar CSS 3 porque es el compañero adecuado de html5 para que el sitio web sea un éxito (17).

1.3.5 Servidor Web.

Un servidor web es un programa que implementa el protocolo HTTP sus siglas vienen de Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto) o el protocolo HTTPS (la versión cifrada y autenticada). El protocolo HTTP pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llama hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música (18).

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Servidor Web Apache

El servidor Apache es un *software* que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

Módulos Base: Módulo con las funciones básicas del Apache.

Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.

Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el *software*, se emplear el servidor web apache 2.2.

Ventajas de Apache:

- Modular
- Open source (Código abierto)
- Multiplataforma
- Extensible (19).

1.3.6 Entornos de Desarrollo Integrado (IDE)

Los Entornos de Desarrollo Integrados (en inglés *Integrated Development Environment* o IDE) son programas compuestos por una serie de herramientas que utilizan los programadores para desarrollar código. Estas herramientas pueden estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos(20).

NetBeans

NetBeans IDE es una herramienta para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE, además es un producto libre y gratuito sin restricciones de uso (21).

Para el desarrollo de la solución se ha elegido NetBeans IDE, en su versión 7.2.1, como Entorno de Desarrollo Integrado.

1.3.7 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos (22).

MySQL

MySQL es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del *software* libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de bibliotecas y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Se proponen utilizar MySQL 5.5.

Características de MySQL

- Las principales características de este gestor de bases de datos son las siguientes:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Gestión de usuarios y contraseña, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está (23).

1.3.8 Herramienta de Diseño

Las herramientas de diseño se han convertido en todo una moda entre las empresas productoras de *software* a nivel internacional, en la actualidad gran cantidad de estas han adquirido herramientas CASE², con el propósito de automatizar los aspectos clave durante el proceso de desarrollo de los sistemas informáticos durante todo su ciclo de vida y de esta manera lograr un trabajo más eficiente (19).

Visual Paradigm for UML

Visual Paradigm ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de

² CASE: siglas en ingles que se utilizan para referirse a Ingeniería de *Software* Asistida por Computadora

usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque Orientado a Objetos (24).

Algunas de sus principales características se relacionan a continuación.

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Licencia: gratuita y comercial.
- Soporta aplicaciones Web.
- Compatibilidad entre ediciones.
- Editor de Detalles de Casos de Uso - Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.

Conclusiones del capítulo

- Durante el desarrollo de este capítulo se realizó un análisis de los sistemas informáticos existentes, que comparten características similares con la solución informática que se propone. Centrándose el estudio en cómo se gestiona la solicitud, instalación y configuración de un sitio web, lo cual permitió definir, conjunto con la dirección de investigaciones, las funcionalidades a desarrollar en la solución informática para las deficiencias presentes en la plataforma electrónica GRECIA.
- Las tecnologías seleccionadas para el desarrollo de la solución están acordes a las utilizadas para el desarrollo de la plataforma. Fue definido como lenguaje de modelado UML, el cual ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Como herramienta CASE fue seleccionado el Visual Paradigm que se utilizará para el apoyo del modelado del negocio, así como para la generación de artefactos en las etapas de definición de requisitos.

- La utilización del marco de trabajo Drupal 7 para el desarrollo de la aplicación ofrecen disímiles ventajas para la implementación y el posterior mantenimiento del sistema. Fue definido el sistema gestor de base de datos MySQL que brinda opciones tanto en flexibilidad, rapidez y seguridad.

Capítulo 2. Descripción de la solución propuesta

Introducción

Se describen a continuación las características de la propuesta de solución haciendo hincapié en la arquitectura, diagramas y patrones de diseño que serán utilizados en la misma así como en los requerimientos pactados con el cliente.

2.1 Modelo de Dominio

El modelo de dominio se realiza para tener un mejor entendimiento del sistema en la actualidad, se basa en la representación de los conceptos más importantes y significativos para el desarrollo de un sistema. Este representa clases conceptuales del dominio del problema y conceptos del mundo real, no de los componentes de *software*. Se definen las interrelaciones de los objetos más importantes representados mediante clases (25).

Para el sistema a desarrollar se define el siguiente diagrama de dominio:

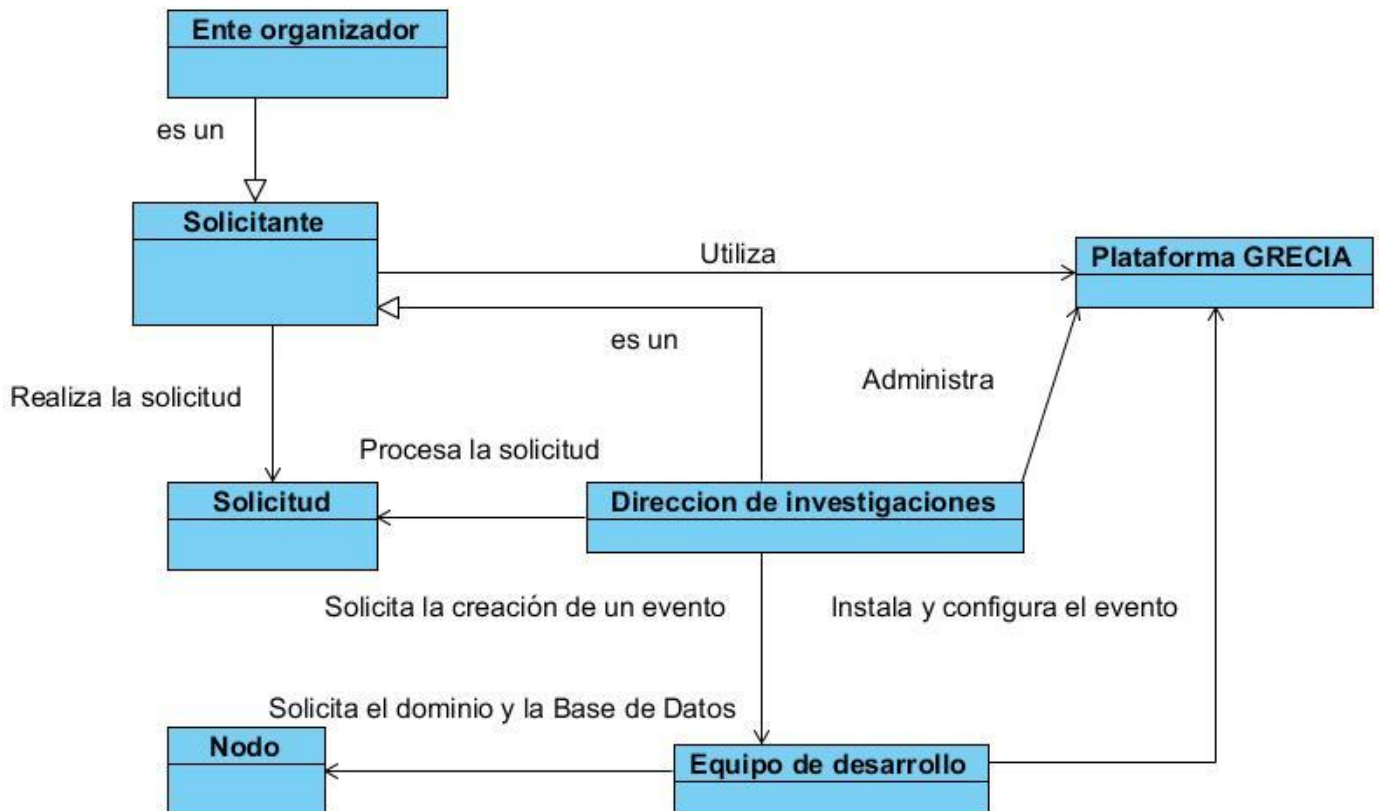


Figura 8. Modelo de dominio

- Solicitante: Es el que realiza la solicitud a la dirección de investigación para realizar un evento, puede pertenecer a una institución interna o externa de la UCI.
- Solicitud: Es el documento que contienen los datos del solicitante y el evento.
- Dirección de Investigación: Se encarga de la organización científica.
- Plataforma: Brinda un conjunto de funcionalidades que son utilizadas por un usuario.
- Desarrolladores: Son los encargados del desarrollo de las funcionalidades a incluir en la plataforma.
- Nodo: Es el área de la Universidad que se encarga de brindar los servicios telemáticos.

El solicitante le solicita a la dirección de investigaciones la creación de un nuevo evento mediante carta a la institución o correo electrónico. El especialista de la dirección de investigaciones procesa la solicitud en caso de ser aprobada solicita al equipo de desarrollo para que realice los procesos en la plataforma, sino se le envía un correo o carta al solicitante informándole el rechazo de la solicitud. El equipo de desarrollo solicita al nodo el dominio para así pasar a instalar y configurar el evento, cuando concluyen el especialista de la dirección de investigaciones le informa al solicitante que ya puede hacer uso de la plataforma.

2.2 Descripción de la propuesta de solución

Debido a como se realiza el proceso en el momento que fue realizada la investigación, este no es factible para los usuarios debido a los múltiples inconvenientes que puede generar, por lo que se tiene como propuesta de solución el desarrollo del módulo gestión de solicitud, la instalación y configuración del perfil que permitan la creación de un nuevo evento a partir de una solicitud realizada por un cliente, que en caso de ser aprobada, podrá realizarse de forma intuitiva por parte del solicitante. De igual forma la plataforma debe brindar la posibilidad de realizar configuraciones sin depender del equipo de desarrollo. Una vez que el solicitante se autentica en la plataforma y da clic en realizar solicitud, llena un formulario con los datos que debe cumplir la misma, luego que se envía la solicitud, el sistema le muestra un mensaje al usuario reafirmando que su solicitud ha sido enviada correctamente, después el especialista de la dirección de investigaciones procesa la solicitud, en caso de ser aprobada, le solicita al nodo el dominio y luego se le envía un correo al solicitante mediante la plataforma, notificándole que ya puede entrar a su sitio enviándole además la URL del mismo. De esta manera, el solicitante accede a instalar y configurar su perfil como el desee en la plataforma.

El diagrama de actividades representa la modelación del flujo de tareas y el modelo de las operaciones, para un mejor entendimiento de como se realiza el flujo de trabajo paso por paso a continuación se describe la propuesta de solución haciendo uso del mismo.

Seguidamente se describe el proceso completo.

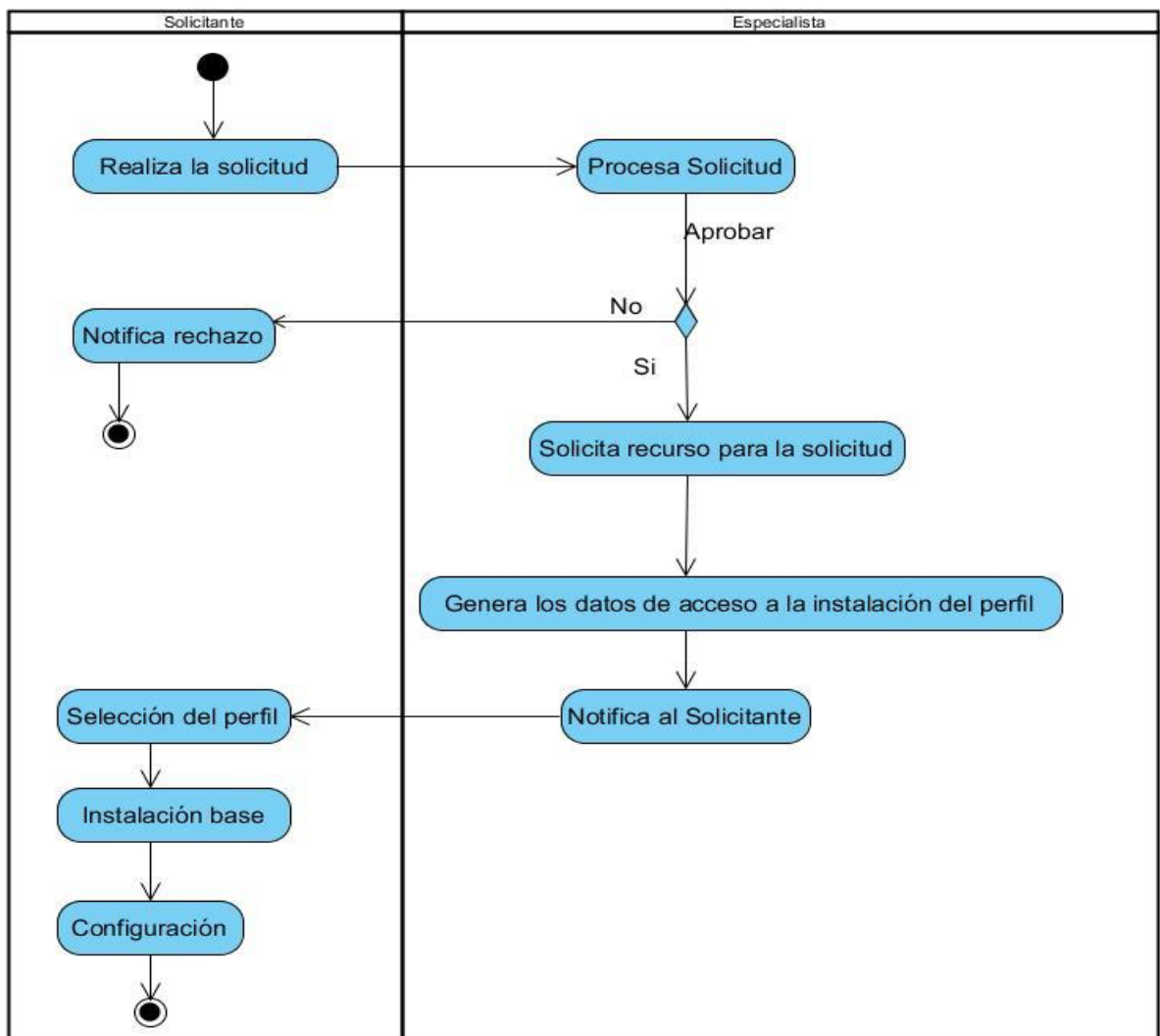


Figura 9. Diagrama de actividades.

El solicitante puede ser una institución tanto interna como externa de la UCI que realiza la solicitud para gestionar la creación de un nuevo evento en la plataforma electrónica. El especialista es la persona capacitada de la dirección de investigaciones, que se encarga de procesar la solicitud, que en caso de ser aprobada, solicita los recursos necesarios para la instalación del perfil y notifica a los solicitantes el

estado en que se encuentra la solicitud. Una vez que el especialista, mediante la plataforma, le informa al solicitante que ya puede instalar y configurar su perfil, el mismo selecciona su perfil y es cuando se realiza la instalación básica y la configuración, de no ser aprobada la solicitud se le notifica al solicitante la negación de la misma.

2.3 Captura de requerimientos del software

El objetivo fundamental de la captura de los requerimientos del *software* es traducir de forma sencilla el problema tal y como lo ve el cliente, logrando un mejor entendimiento entre el usuario final y los desarrolladores. Este análisis sobre lo que debe y no debe hacer el sistema permite identificar las funcionalidades requeridas y las restricciones que se imponen.

2.3.1 Mecanismo empleado para la captura de requisitos

La **entrevista** permite al analista adquirir conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Su estructura abarca cuatro pasos:

- Identificación de los entrevistados.
- Preparación de la entrevista.
- Realización de la entrevista.
- Documentación de los resultados (26).

Resumen de las entrevistas al especialista de la Dirección de Investigaciones (Cliente):

En la entrevista realizada a los especialistas de la Dirección de investigaciones para obtener información acerca de las necesidades y perspectivas sobre el sistema, expresó la importancia de contar con una aplicación sencilla, fácil de utilizar por cualquier persona sin dominio de Drupal; que después que el usuario se autentique pueda acceder a las funcionalidades del sistema; accesible desde la web; multiplataforma; que respete la identidad de la Universidad; que permita almacenar la información de los eventos realizados. Además, que mediante una solicitud se pueda instalar un evento, configurar como el solicitante lo desee y que le brinde la posibilidad de visualizar todos los datos de su evento.

2.4 Especificación de Requisitos

En el artefacto “Especificación de requisitos” se describe los requisitos funcionales y no funcionales del *software* y hardware del sistema a implementar.

2.4.1 Requisitos funcionales.

Los requerimientos funcionales son las capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto y que a su vez cumpla con las necesidades del usuario (27).

Seguidamente se dan a conocer los requisitos funcionales que son:

- **RF1-** Registrar solicitud: El sistema debe permitir registrar la solicitud.
- **RF2-** Modificar solicitud: El sistema debe permitir modificar la solicitud.
- **RF3-** Buscar solicitud: El sistema debe permitir buscar la solicitud por el criterio de búsqueda.
- **RF4-** Eliminar solicitud: El sistema debe permitir eliminar la solicitud.
- **RF5-**Listar solicitud: El sistema debe permitir listar las solicitudes. .
- **RF6 -** Aprobar solicitud: El sistema debe permitir aprobar la solicitud.
- **RF7-** Rechazar solicitud: El sistema debe permitir rechazar la solicitud.
- **RF8-**Registrar perfil: El sistema debe permitir registrar el perfil.
- **RF9-**Modificar Perfil: El sistema debe permitir modificar el perfil.
- **RF10-**Buscar perfil: El sistema debe permitir buscar el perfil.
- **RF11-**Eliminar perfil: El sistema debe permitir eliminar el perfil.
- **RF12-**Instalación de perfil: El sistema debe permitir la instalación del perfil.
- **RF13-**Configuración de perfil de instalación: El sistema debe permitir la configuración del perfil de instalación.

2.4.2 Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener como las características que hacen al producto atractivo, usable, rápido o confiable (27).

1- Usabilidad.

- RNF1.1-El sistema debe presentar una interfaz amigable que permita al usuario la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información que se busca.
- RNF1.2-Debe permitir a usuarios con pocos conocimientos informáticos poder interactuar con el mismo.

2- Fiabilidad

- RNF2.1- El sistema debe registrar cualquier fallo en su funcionamiento para que los administradores puedan resolverlo, y de la misma forma mostrarle al usuario un mensaje indicándole que ha ocurrido un fallo en la operación que realice.

- RNF2.2-Los mensajes que se muestren a los usuarios invitados deben mostrar los errores sin dar detalles de información, que puedan comprometer la seguridad e integridad del mismo.

3- Soporte

- RNF3.1- Se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
- RNF3.2-El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.
- RNF3.3-Se le debe dar mantenimiento al sistema y corregir cualquier problema que surja con la aplicación.

4- Restricciones de diseño

Para el correcto funcionamiento del sitio se necesitan las siguientes herramientas:

- RNF 4.1 Bibliotecas de PHP: pdo, pdo- mysql.
- RNF 4.2 Servidor Web Apache 2.2.
- RNF 4.3 Se requiere un navegador capaz de interpretar JavaScript 1.8 y CSS 3, HTML 5, se recomienda Mozilla Firefox en su versión 7.x o superior.
- RNF 4.4 Gestor de base de datos MySQL en su versión 5.5 o superior.
- RNF 4.5 La versión de PHP debe ser la 5.2.5 o superior, recomendándose la versión 5.3.
- RNF 4.6 Cómo Sistema de Gestión de Contenidos Drupal 7.

5- Apariencia o interfaz externa

- RNF 5.1 La interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.
- RNF 5.2 El sistema deberá estar optimizado para una resolución de 1024x768.
- RNF 5.3 El sitio deberá ser accesible desde cualquier punto de la red que se disponga.

- RNF 5.4 Se utilizará el protocolo HTTP³/HTTPS⁴ para la comunicación entre el cliente y el servidor en los procesos de envío de datos entrados por el usuario en la autenticación y en las tareas administrativas y de gestión de contenidos.

6- Legales

- RNF 6.1 El CMS escogido como plataforma para el desarrollo del portal está basado en la licencia GNU/GPL versión 2.

7- Seguridad

- RNF 7.1 Define los roles de cada usuario permitiendo separar las funcionalidades que podrá realizar cada uno.
- RNF 7.2 La carpeta donde se encuentre el sitio solo tendrá permiso de lectura.
- RNF 7.3 Garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.
- RNF 7.4 Se debe implementar un sistema de salvadas de seguridad de la base de datos y de los ficheros del sistema de la forma siguiente: una copia total al año, una diferencial cada semestre y una incremental cada mes.

8- Eficiencia

- RNF 8.1 El sistema debe soportar un mínimo de 100 peticiones.
- RNF 8.2 Debe ser capaz de responder en un tiempo menor de 1 segundo para las páginas de información y 2 segundos para las operaciones de registro de contenidos.

9- Hardware

Servidor web y de base de datos:

- RNF 9.1 Poseer 1 GB de memoria RAM para cumplir con una carga de trabajo media.
- RNF 9.3 El micro procesador debe tener cuatro núcleos o más y no menos de 1.5 GHz.

³ HTTP (Protocolo de transferencia de hipertexto).

⁴ HTTPS (Protocolo seguro de transferencia de hipertexto).

Cliente:

- RNF 9.4 Debe tener mínimo 256 Mb de RAM.

10- Disponibilidad

- RNF 10.1 El sitio debe estar disponible en todas las horas del día.
- RNF 10.2 Se deben realizar tareas de mantenimiento una vez al año para revisar las capacidades óptimas de trabajo del sistema.

11- Portabilidad

- RNF 11.1 El sistema debe poder ser instalado en cualquier sistema operativo ya sea Windows o cualquier distribución de Linux.

2.4.3 Validación de requisitos

La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Se desarrolló mediante las siguientes técnicas:

- Prototipos.

Permiten al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica le al usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

- Revisión técnica de artefactos

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.

La validación de los requisitos permitió comprobar que los requerimientos definidos para el sistema, no presentan ambigüedad, además que se encuentran claramente definidos y poseen objetivos concretos. Los prototipos de interfaz de usuario ofrecieron un mejor entendimiento del sistema para el cliente, ajustándose a las necesidades del mismo.

2.4.4 Actores del sistema

A continuación se describen los Actores del sistema:

Actor	Descripción
Solicitante	Son instituciones tanto internas con externa a la UCI que interactúan con el sistema para realizar una solicitud de un nuevo evento en la plataforma llenado un formulario de solicitud.

Especialista	Es el encargado de procesar la solicitud.
Administrador del sistema	Es el encargado de gestionar los roles de los usuarios del sistema y toda la información del mismo.

Tabla 1. Actores del sistema

2.4.5 Descripción de requisitos

A continuación se muestra la descripción de uno de los requisitos funcionales del sistema.

Precondiciones	Usuario autenticado en la plataforma
Flujo de eventos	
Flujo básico Realizar solicitud	
1	Seleccionar la opción de realizar solicitud.
2	Muestra una pantalla para introducir los datos de la solicitud.
3	Introduce los datos.
4	Envía la solicitud.
5	Valida los datos de la solicitud en caso de algún error ver <u>Flujo alternativo 1.</u>
6	Valida que la solicitud no existe en el sistema, sino <u>Flujo alternativo 2.</u>
7	Registrar la solicitud.
8	Concluye el requisito.
Pos-condiciones	
1	Se guardan los datos de la Solicitud.
Flujos alternativos	
Flujo alternativo 1.a Error en los datos de la solicitud	
1	El sistema señala en rojo el campo con los datos incorrectos.
2	Muestra un mensaje de error.
3	Continúa en el punto 3.
Flujo alternativo 2.a Ya existe la solicitud	
1	Se muestra un mensaje “Ya existe la solicitud”.
2	Continúa en el punto 2.
Pos-condiciones	
1	N/A
Validaciones	
1	N/A
Requisitos	N/A

especiales

Asuntos N/A

pendientes

Tabla 2. Descripción del requisito funcional Realizar solicitud.

Prototipo elemental de interfaz gráfica de usuario.

The screenshot shows a web form titled "Crear Solicitud" (Create Request). The form includes the following elements:

- Nombre ***: A text input field with the placeholder "Escribe su nombre".
- Apellidos ***: A text input field with the placeholder "Escribe su dos apellidos".
- Correo ***: A text input field.
- Afiliación**: A dropdown menu.
- Temática**: A dropdown menu.
- Descripción**: A large text area.
- Documento**: A file upload section with an "Examinar..." button and a "Subir al servidor" button. Below it, a note states: "Los archivos deben ser menores que 1 MB. Tipos de archivo permitidos: doc..."
- Propuesta de URL**: A text input field.
- Estado de la solicitud**: A section with a "Cambiar Estado de la solicitud" button and a radio button selected for "Pendiente".
- At the bottom, there are "Guardar" and "Cancelar" buttons.

Formatos de entrada/salida

Entradas

N/A

Salidas

N/A

2.6 Diagrama de paquete

Un diagrama de paquetes es un conjunto de agrupaciones lógicas de elementos del modelo, mostrando las dependencias entre dichas agrupaciones. Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los

paquetes. Un paquete puede contener paquetes subordinados, así como otros tipos de elementos del modelo. Todo tipo de elementos del modelo UML pueden ser organizados en paquetes.

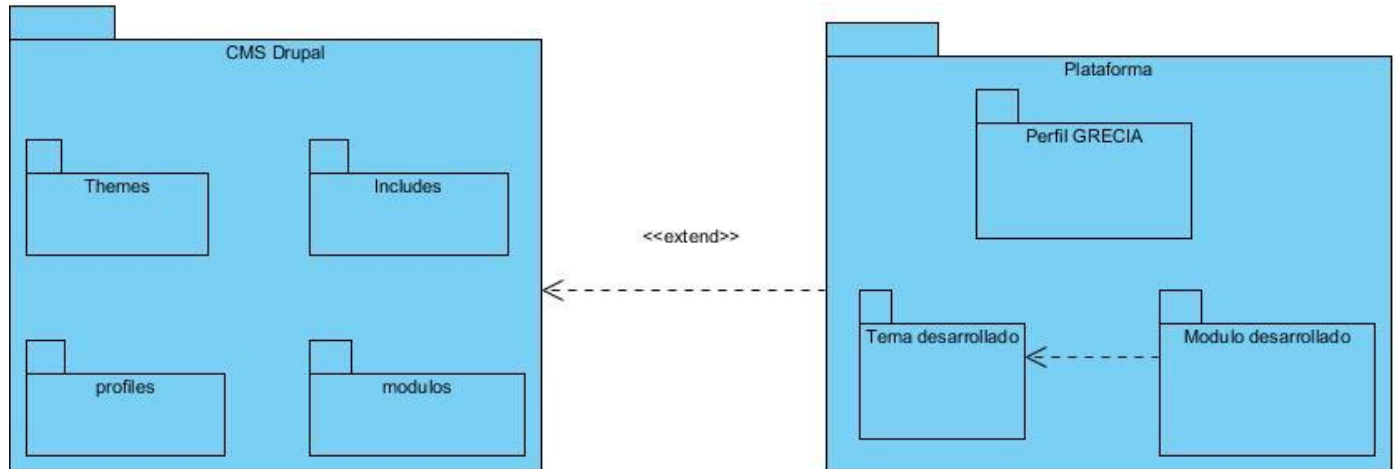


Figura 10. Diagrama de paquete.

En el diagrama se representa un paquete nombrado “CMS Drupal”, que incluye los paquetes contenedores de los componentes del CMS y otro paquete de los más generales es el denominado “Plataforma”, que incluye los elementos que conforman la aplicación. A continuación se describe el contenido de cada uno:

Paquete CMS Drupal:

- **incluidos (*includes*):** Contiene un conjunto de ficheros indispensables para el funcionamiento de los módulos y el sistema Drupal, como las API de conexión a la base de datos.
- **módulos (*modules*):** Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.
- **temas (*themes*):** Incluye los temas que incluye de forma predefinida Drupal, permiten cambiar el estilo de la vista que se le muestra al usuario, cuando desee incorporar un nuevo tema sólo se tiene que copiar en dicha carpeta.
- **perfiles (*profiles*):** Es un conjunto de instrucciones que permite definir qué módulos y opciones se quieren activar durante el proceso de instalación de Drupal.

Paquete Plataforma:

- **tema desarrollado:** Contiene el tema desarrollado para el sistema.

- **módulo desarrollado:** Contiene el módulo que fue necesario implementar para el desarrollo de la aplicación.
- **perfil Gracia:** Contiene el perfil.

Estos dos paquetes generales (CMS Drupal y Plataforma) lógicamente se encuentran relacionados mediante un *extend*, pues la Plataforma se implementa haciendo uso de dicho CMS y por tanto utiliza las características y funcionalidades que este brinda.

2.6 Seguridad de la solución propuesta

Con el propósito de garantizar la protección de los datos, Drupal implementa la seguridad a través de la autenticación de los usuarios utilizando sesiones y estableciendo un sistema de control de acceso basado en roles. La asignación de permisos se realiza a través de los roles del sistema, de esta forma, el administrador no tiene que establecer los permisos para cada usuario, sino, se asignan los permisos a un determinado rol y se agrupan los usuarios por roles. Esta asignación por roles permite controlar a qué funcionalidades el usuario tiene acceso, de manera que la información accedida esté limitada y protegida en dependencia de los distintos niveles de usuarios, obteniendo como resultado un sistema más seguro. El sistema define los roles de invitado, usuario autenticado y el administrador.

Otro mecanismo implementado por Drupal para brindar seguridad al sistema son las consultas parametrizadas que define la capa de abstracción de base de datos de Drupal, que evitan los ataques de inyecciones SQL. Además, Drupal cuenta con el módulo *Database Logging* el cual se encarga de guardar registro de la actividad de los usuarios en el sistema, lo que permite auditar los fallos que ocurran en el sistema.

2.7 Descripción de estilos arquitectónicos.

Para el desarrollo de la solución se emplea la arquitectura de Drupal, que es una arquitectura modular. El núcleo de Drupal es el encargado de la integración de los módulos que sean instalados. Este le envía la información procesada de los módulos al componente de la lógica del tema, que se encarga de la visualización de la información en las vistas del sistema. Finalmente, los temas que se desarrollen utilizan los datos, permitiendo pre-procesar o alterar los mismos antes de ser mostrados. A su vez estos pueden utilizar las funciones del API de JavaScript y AJAX para brindar una interfaz más intuitiva a los usuarios.

Drupal incluye además un conjunto de bibliotecas que brindan un grupo de servicios como conexión y administración de la base de datos, gestión de procesos de envío de correos, tratamiento de imágenes e internacionalización. Drupal cuenta además con un entorno de integración de utilidades que posibilita ampliar las funcionalidades del sistema a través de métodos uniformes de desarrollo e integración de nuevos módulos.

La figura 9 muestra los elementos de la arquitectura que conforman el sistema:

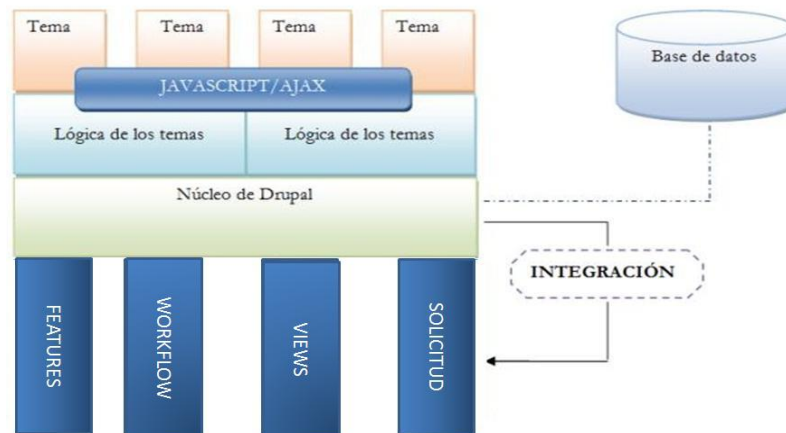


Figura 11. Arquitectura de GRECIA.

La reutilización de los módulos que componen el núcleo de Drupal, es un aspecto a tener en cuenta para el desarrollo de las funcionalidades en la propuesta de solución.

- **Features:** Permite agrupar funcionalidades y reutilizarlas. Básicamente las features son grupos de funcionalidades y objetos que pueden instalarse, activarse y desactivarse en un único paso.

A continuación se enuncian algunos que representan componentes indispensables de las funcionalidades del sistema:

- **Nodo (Node):** Se incluye en el núcleo de Drupal. Permite asociar tipos de contenidos como entidades del sistema para brindar la posibilidad de representar cada una de las entidades de información que se gestiona. En la investigación su utiliza a la hora de realizar los contenidos, en los nodos es donde se va aguardar dicha información creada en cada vista.
- **API de campos (Field API):** Permite definir nuevos atributos (campos) a los tipos de contenidos, para lograr una mayor organización y estructura de la información a la hora del almacenamiento y para facilitar las vistas y búsquedas sobre esa información. Se emplea a la hora de crear los

atributos de un tipo de contenido ejemplo de esto es cuando se crean los atributos con que cuenta la solicitud ya que la solicitud es un tipo de contenido.

- **Campos de validación (Field Validation):** Este módulo contribuye a especificar las reglas de validación de los campos definidos a los contenidos.
- **Vista (Views):** Módulo encargado de brindar la posibilidad de crear cada una de las vistas del sistema para visualizar la información de los diversos tipos de contenidos que sean definidos, de una forma fácil e intuitiva para los desarrolladores. La vista se utiliza para mostrar la información de los contenidos creados, por ejemplo se utiliza para visualizar los elementos del tipo de contenido solicitud.

La propuesta de solución se basa en la utilización de módulos previamente incluidos por el núcleo de Drupal y por otros que son contribuidos por la comunidad de Drupal. Además se desarrolla el módulo gestión de solicitud para agregar esta funcionalidad al sistema.

El núcleo de Drupal es el encargado del funcionamiento de manera integrada de los módulos desarrollados con el resto de los componentes del mismo. El módulo de gestión de solicitud es una evidencia de dicha integración. Este es el encargado de la gestión de la solicitud para realizar la creación de un evento en la plataforma, para la construcción de la interfaz utiliza el módulo *Views* lo que posibilita definir a través de filtros expuestos los campos para que se puedan filtrar los resultados.

2.8 Patrones de diseño.

A continuación se describen los patrones de diseño que se emplea en el desarrollo de la solución:

El núcleo de Drupal 7 está diseñado haciendo uso de una arquitectura modular y de la misma forma utiliza recursos de la Programación Orientada a Objetos como el manejo de objetos para los datos, la herencia, el encapsulamiento, la abstracción y el polimorfismo. Las características estructurales presentes en este CMS son resultados de la aplicación de patrones de diseño. A continuación se describen ejemplos de patrones GOF⁵ que son evidenciados en Drupal.

⁵ La Banda de los cuatro. Es el nombre con el que se conoce comúnmente a los autores del libro Design Patterns.

- **Instancia única (Singleton)**

Este patrón se evidencia si se asocian los módulos y los temas como objetos, cada uno de ellos en si no almacena datos, sino que representan un conjunto de funciones que extienden el núcleo para añadirle funcionalidades al mismo por separado; de esta forma se pueden considerar como una clase con una instancia singleton. Se puede representar al módulo Solicitud como el conjunto de funciones encargadas de la manipulación de los datos del solicitante almacenados en la base de datos, realizando la interacción con estos a través de la misma instancia del módulo.

- **Decorador (Decorator)**

La evidencia de este patrón se encuentra en la definición de hooks por parte de los módulos del núcleo de Drupal y otros módulos contribuidos, que posibilitan que otros puedan extender el comportamiento de dichas funciones. Los hooks son funciones que permiten la comunicación con todo el núcleo de Drupal, pueden ser eventos u operaciones con campos. De esta forma se brinda la flexibilidad de que nuevos módulos puedan modificar el comportamiento del núcleo en cuanto al tratamiento de los datos y en cada uno de los eventos del sistema. Este comportamiento podría ser imitado por el uso de decoradores, por ejemplo con el hook_views_exposed_form_alter definido por el módulo Views de Drupal, el módulo de Solicitud implementa ese hook para alterar la presentación de los filtros de las vistas de búsqueda de solicitantes.

- **Observador (Observer)**

El comportamiento anterior descrito en la utilización del patrón decorator, puede ser visto también como evidencia de la aplicación del patrón observer. Este patrón se evidencia en el sistema en la implementación del hook_entity_delete para eliminar las referencias de los elementos de las listas a la hora de eliminar.

- **Puente (Bridge)**

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón Puente. Cada módulo debe ser 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 nuevas siguiendo la API definida, para añadir soporte para nuevos sistemas gestores de base de datos, sin necesidad de modificar el código del módulo.

- **Cadena de responsabilidades (Chain of responsibility)**

El sistema de menús de Drupal es la evidencia del patrón Chain of responsibility. En cada petición de una página, el sistema de menús 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 trasmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos. De esta forma se continúa la cadena hasta que un módulo atienda la petición, hasta que un módulo deniegue el acceso o hasta que la cadena se agote. Todas las peticiones de rutas definidas por el módulo desarrollado evidencian este patrón.

- **Comando (Command)**

Muchos de los hooks definidos por el núcleo de Drupal usan el patrón Command para reducir número de funciones que son necesarias implementar, pasando la operación como parámetro conjuntamente con los argumentos. De esta forma se evidencia en el sistema de hooks, que cada módulo no necesariamente debe definir cada hook, sino solo los que tienen que ver con su propósito y funcionamiento. En el desarrollo de la solución propuesta se evidencia este patrón ya que el módulo implementado no tiene que definir cada hook, sino los que vayan a aplicar.

Conclusiones del capítulo

- Durante el desarrollo de este capítulo se obtuvo los artefactos generados durante la etapa de diseño de la solución. Donde la descripción del modelo de dominio permitió representar cómo se realizan los procesos en la actualidad, lo que permite desarrollar con mayor facilidad la solución propuesta.
- Fueron presentadas además las técnicas de captura de requisitos, que permitieron dejar claramente expuestas las funciones con las que debe contar el *software* para que el producto cumpla con las expectativas del cliente. Con la validación de los requisitos fue comprobada la calidad con que se capturaron los mismos, arrojando resultados alentadores, los cuales se evidenciarán en la etapa de implementación.
- El desarrollo de la propuesta de solución basado en la arquitectura modular y los patrones de diseño con que cuenta Drupal brindaron la flexibilidad y adaptabilidad presentes en el sistema ante posibles cambios en la implementación de las funcionalidades.

Capítulo 3. Implementación y validación

Introducción

En este capítulo se describen los elementos establecidos por el proceso de desarrollo empleado, que responden a la implementación de la solución. Este capítulo concluye con la realización de pruebas de caja negra a la solución obtenida.

3.1 Estándares de codificación.

La implementación del módulo y perfil utilizando el marco de trabajo Drupal, se emplea los siguientes estándares de código para hacer más entendible la implementación de los mismos.

Drupal cuenta con estándares de codificación que permiten lograr una implementación limpia y organizada para cualquier desarrollador que utilice el código desarrollado (28).

Indentación: La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos. En Drupal la indentación debe ser de 2 espacios. En la siguiente imagen se muestra un fragmento de código con las indentaciones realizadas, de 2 espacios cada una (→→), y los saltos de línea o Entre (↵) al final de cada línea (sin dejar espacios).

```

<?php↵
function forum_block_view($delta = '') {↵
→→$query = db_select('forum_index', 'f')↵
→→→→→fields('f')↵
→→→→→addTag('node_access');↵
→→switch ($delta) {↵
→→→→case 'active':↵
→→→→→$title = t('Active forum topics');↵
→→→→→$query↵
→→→→→→→orderBy('f.last_comment_timestamp', 'DESC')↵
→→→→→→→range(0, variable_get('forum_block_num_active', '5'));↵
→→→→→→→break;↵
↵
→→→→case 'new':↵
→→→→→$title = t('New forum topics');↵
→→→→→$query↵
→→→→→→→orderBy('f.created', 'DESC')↵
→→→→→→→range(0, variable_get('forum_block_num_new', '5'));↵
→→→→→→→break;↵
→→}↵
↵
→→$block['subject'] = $title;↵
→→// Cache based on the altered query.↵
→→$block['content'] = drupal_render_cache_by_query($query,↵
'forum_block_view');↵
→→$block['content']['#access'] = user_access('access content');↵
→→return $block;↵
}↵
?>↵

```

Figura 12. Indentación.

Etiquetas de apertura y cierre de PHP: Se utiliza cuando estemos escribiendo en PHP, siempre se deben utilizar las etiquetas `<?php y?>`, y en ningún caso la versión corta `<? y?>`. En la siguiente figura se muestra cómo se utilizan las etiquetas de apertura y cierre de PHP en los archivos.

```
<?php if ($forums_defined): ?>
<div id="forum">
  <?php print $forums; ?>
  <?php print $topics; ?>
</div>
<?php endif; ?>
```

Figura 13. Etiquetas de apertura y cierre de PHP.

Funciones: Deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, etc., para evitar así duplicidad de funciones. 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.

```
function forum_help($path, $arg) {
  En la llamada a la función se aplican las mismas reglas anteriores con respecto a
  los parámetros, como se muestra en el siguiente ejemplo:
  $field = field_info_instance('node', 'taxonomy_forums', $node->type);
```

Figura 14. Funciones.

Documentando ganchos: Que permiten la comunicación entre los distintos módulos del sistema, esto posibilita que cuando una función es la implementación de un gancho, puede no ser necesario dar más detalles sobre la misma, ya que los comentarios se habrán definido en la función gancho original. Se muestra un ejemplo de documento gancho.

```
/**
 * Implements hook_help().
 */
function forum_help($path, $arg) {
  ...
}
```

Figura 15. Documentando ganchos.

Operadores: Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Los operadores unarios como ++, -- no deben tener separación.

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. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

Estructuras de control: Con respecto a las estructuras de control, se debe tener en cuenta las siguientes normas:

- Debe haber un espacio entre el comando que define la estructura (if, while, for) y el paréntesis de apertura permitiendo no confundir las estructuras de control con la nomenclatura de las funciones.
- La llave de apertura ({) se sitúa en la misma línea que la definición de la estructura, separada por un espacio.
- Se recomienda usar siempre las llaves { } aún en los casos en que no sea obligatorio su uso.
- Las estructuras else y elseif se deben escribir en la línea siguiente al cierre de la sentencia anterior.

Arreglos: Los valores dentro de un arreglo se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del arreglo supera 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 debe escribir, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

Constantes: Los nombres de las constantes deben escribirse en mayúsculas, con guiones bajos para separar palabras. Al igual que ocurre con las funciones, los nombres de las constantes deben tener como prefijo el nombre del módulo (o tema) en el que se utilizan, para evitar errores de duplicidad de constantes. Este prefijo también se debe escribir en mayúsculas.

Variables globales: Estas se deben declarar utilizando un guión bajo inicial, seguido del nombre del módulo o tema y otro guión bajo antes del nombre de la variable.

Nombres de archivos: Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tienen extensión .txt y el nombre es en mayúsculas.

URLs de ejemplo: Por convención, siempre que haya que indicar una *URL* de ejemplo, se debe utilizar "example.com".

3.2 Modelo de Implementación

Durante la etapa de diseño se genera los artefactos que propician el correcto desarrollo de la implementación del sistema, y que posee como principal propósito desarrollar la arquitectura y el sistema en conjunto.

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan, como ficheros de código fuente o ejecutables, describe cómo se organizan el módulo y el perfil de acuerdo con los mecanismos de estructura disponibles en el entorno de implementación y en el lenguaje de programación utilizados.

A continuación se describe los datos que contiene el módulo **Gestionar solicitud:**

Contenido	Tipo de campo
Nombre y Apellidos	Texto
Correo Electrónico	Texto
Temática	Autocompletable
Afiliación	Autocompletable
Proponer URL	Texto
Descripción	Texto largo
Adjunto	Archivo
Fecha	Calendario

Tabla 3.1. Módulo Gestionar solicitud.

- Nombre y Apellidos: Nombre y Apellidos del solicitante.
- Correo Electrónico: Correo Electrónico del solicitante.
- Temática: Nombre del evento.
- Afiliación: Centro al que pertenece.
- Proponer url: Proponer una url para el evento.
- Descripción: Descripción de la solicitud.
- Adjunto: Archivo que respalda la descripción de la solicitud.

- Fecha: El día que se realiza la solicitud.

Para el desarrollo de la solicitud se utiliza el form para crear el formulario de la solicitud junto al mismo se implementa el submit para enviar, la función hook_node_validate que se utiliza para verifica que cada campo cumpla con las reglas definidas. El formulario solicita el valor de un campo de texto obligatorio, el valor introducido será almacenado en una variable de la base de datos, desde la función de envío (submit), para cancelar se utiliza hook_user_cancel_methods_alter, la función hook_default_rules_configuration() permite definir las reglas por defecto y hook_views_default_views() la vista por defecto.

Desarrollo del Perfil de instalación:

Para el desarrollo del perfil de instalación se llevó acabo teniendo en cuenta los perfiles de Drupal, para el desarrollo del perfil GRECIA se modificó las operaciones y los módulos que se activan durante un perfil definido para Drupal, crean su propia características centrado en satisfacer la necesidad de un cliente en específico para que pueda instalar y configura su evento si necesidad de tener algún conocimiento de Drupal y de forma amigable.

Descripción de la Configuración del perfil.

Contenido	Datos del Contenido
General	<ul style="list-style-type: none"> -Nombre del evento -Slogan(tema) -Correo electrónico -Opciones Regionales(país) -Patrocinador -Datos de contacto
Evento	<ul style="list-style-type: none"> -Tipo de evento(abierto o cerrado) -Forma de evaluación (simple o por pares a ciega) -Fechas del evento <ul style="list-style-type: none"> • Inicio del evento • Fin de plazo de subida de los trabajos • Fin de plazo de aceptación del trabajo • Fin de plazo de evaluación parcial de un evento

	<ul style="list-style-type: none"> Fin de plazo de evaluación final de un evento Fin de plazo para entregar los resultados finales del evento.
Vista	<ul style="list-style-type: none"> -Banner -Color -Fuente -Layout (Estructura)

Para el desarrollo del perfil se emplea la función `hook_install_tasks ()` que permite modificar el perfil y en `grecia_install_tasks()` es donde se implementa las opciones que va a tener el perfil, además cada elemento del perfil tiene su función de implementación ejemplo `grecia_datos_generales_form` se va encontrar todos los datos de la opción General. También se emplea `hook_install ()` que es quien permite la instalación del perfil.

3.3 Diagrama de componentes

La relación existente entre los elementos físicos es descrita mediante el diagrama de componentes que muestra la forma en que se encuentran entrelazados los diferentes componentes que son utilizados en el desarrollo de la solución. Son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar la relación entre ellos. A continuación se presenta el diagrama de componentes de la solución, en el mismo se representa los principales componentes por los que está formado el sistema.

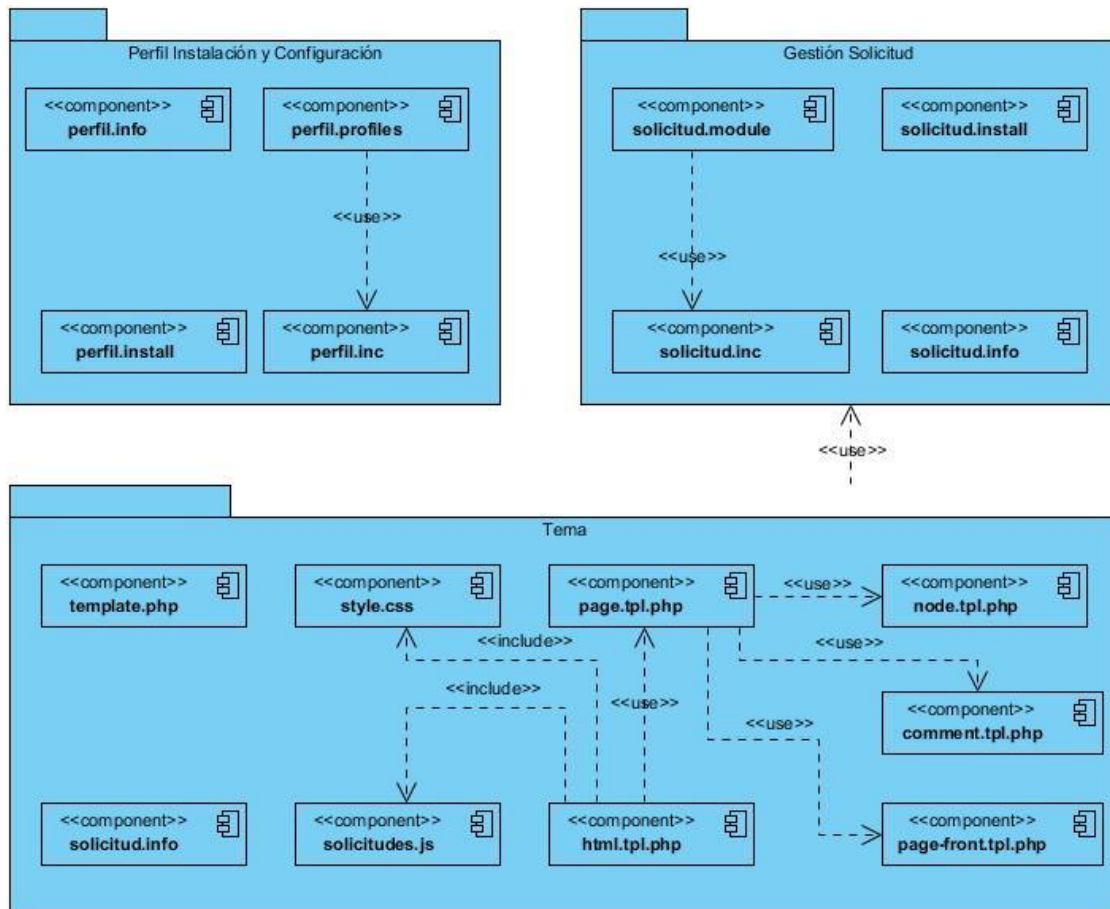


Figura 17 Diagrama de componentes.

El diagrama de componentes está compuesto por:

- **solicitud.info:** Fichero que contiene la información del nombre, versión y dependencias del módulo o tema.
- **template.php:** Contiene las funciones de pre-procesamiento del tema determinado.
- **style.css:** Donde se definen las fuentes, colores y los estilos.
- **solicitud.js:** Funciones programadas en el lenguaje JavaScript empleadas para brindar una interfaz más intuitiva a los usuarios.
- **Gestión Solicitud:** Módulo encargado de gestionar la solicitud.
 - solicitud.module:** Incluye todas las funciones definidas en el módulo.
 - solicitud.info:** Incluye la definición del módulo.
 - solicitud.install:** Se ejecuta durante la instalación del módulo.

-solicitud.inc: Se encuentran funciones complementarias que contiene el módulo.

- **Perfil instalación y configuración:** Es donde se encuentra los componentes que conforman el perfil GRECIA.

-perfil.info: Definición del perfil de instalación.

-perfil.porfiles: Es el archivo donde se implementan las tareas que se llevarán a cabo durante la instalación.

-perfil.install: Permite implementar `hook_install ()` para insertar contenido en la base de datos durante la instalación.

-perfil.inc: Se encuentran funciones complementarias que contiene el perfil.

3.4 Diagrama de despliegue del sistema

Los diagramas de despliegue proveen una vista de cómo los componentes se despliegan en el transcurso de la infraestructura del sistema. Permite comprender la correspondencia entre la arquitectura de *software* y la de *hardware*. En la Figura 17 se muestra la forma en que se van a comunicar los componentes del sistema mediante el Diagrama de Despliegue, donde la PC cliente se va a comunicar por el protocolo HTTP/HTTPS para utilizar las funcionalidades que se encuentran en el servidor de aplicaciones; este a su vez se comunica con el servidor de base de datos por medio del protocolo TCP/IP para la obtención de la información que maneja el sistema.

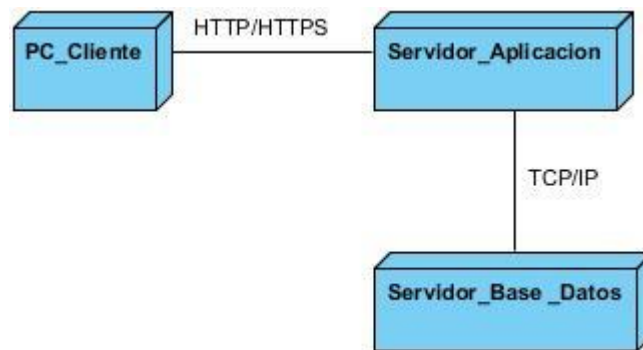


Figura 17. Diagrama de Despliegue

Protocolos:

- **TCP/IP (Protocolo de comunicación y transmisión/ Protocolo de Internet):** Se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.
- **HTTP (Protocolo de transferencia de hipertexto):** Establece un esquema de comunicación cliente – servidor. El cliente es el navegador web que realiza las peticiones a las que el servidor se encarga de dar respuesta.
- **HTTPS (Protocolo seguro de transferencia de hipertexto):** Es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

3.5 Validación

Seguidamente se describe el proceso de validación de los resultados obtenidos en la construcción de la solución propuesta, con el objetivo de corroborar el correcto funcionamiento a partir de los requerimientos definidos para el desarrollo del sistema.

Según Pressman las pruebas de *software* son un elemento crítico para la garantía de calidad del *software* y constituyen una revisión final de las especificaciones del diseño y de la codificación (29).

3.5.1 Pruebas de *software*

En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que el *software* cumple con los requerimientos. Las pruebas de *software* constituyen un conjunto de herramientas, técnicas y métodos que evalúan el desempeño de un programa (30). Estas involucran las operaciones del sistema, evaluando los resultados bajo condiciones controladas, lo que hace que la realización de pruebas al *software* sea un factor de vital importancia. La etapa de pruebas implica los siguientes aspectos (30):

- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos sean implementados correctamente.
- Identificar y asegurar que los defectos encontrados sean corregidos antes de entregar el *software* al cliente.

3.5.2 Pruebas funcionales

Se denominan pruebas funcionales 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 (31).

Dentro de las pruebas funcionales se encuentra el método de caja negra, las cuales están diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento del programa. La prueba de caja negra permite al ingeniero del *software* obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Estas pruebas no son una alternativa a las técnicas de prueba de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca no son capaces de identificar.

La prueba de caja negra intenta encontrar errores de las siguientes categorías (31):

- Las funcionalidades que están incorrectas o ausentes.
- Los errores de interfaz.
- Los errores en estructuras de datos o en accesos a las bases de datos externas.
- Los errores de rendimiento.
- Los errores de inicialización y terminación.

3.5.3 Descripción de caso de pruebas

Los casos de pruebas pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. Permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Deben verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.
- Si el producto se comporta tal y como se describe en las especificaciones funcionales del diseño.

Para poder comprobar que todos los requisitos de una aplicación fueron revisados debe existir un caso de prueba para cada requisito y si el requisito tiene requisitos secundarios se debe hacer un caso de prueba para cada uno de los requisitos secundarios. Existen muchos casos de prueba para determinar que un requisito es satisfactorio, a continuación se presenta una muestra del diseño de casos de pruebas realizados al sistema (Tabla 3.1).

Escenario	Descripción	Nombre de solicitante	Apellidos del solicitante	Afiliación	Correo electrónico	Temática	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Realizar solicitud correctamente insertando los datos correctamente	Mediante este escenario se realiza la solicitud de un evento.	V Arianna	V Galán Quijala	V Facultad 3	V aquiala@estudiantes.uci.cu	V Mi Web x Cuba	V Esta es mi solicitud	El sistema actualiza los datos y muestra un mensaje de confirmación: "Se enviado la solicitud".	Luego de acceder al sistema el usuario escoge la opción realizar solicitud, el sistema muestra el formulario de solicitud. El usuario llena los campos correctamente y selecciona el botón "enviar".
EC 1.2 Realizar solicitud	Mediante este escenario	i Vacío	V Galán Quijala	V Facultad 3	V aquiala@estu	V Mi Web x Cuba	V Esta es mi	El sistema	Luego de acceder

incorrectamente insertando los datos correctamente	o se realiza la solicitud de un evento.	V	I	V	V	V	solicitud	muestra un mensaje de alerta "Por favor verifique que los datos introducidos estén correctos".	al sistema el usuario escoge la opción realizar solicitud, el sistema muestra el formulario de solicitud. El usuario llena los campos correctamente y selecciona el botón "enviar".
		Arianna	Vacío	Facultad 3	aquiala@estudiantes.uci.cu	Mi Web x Cuba	Esta es mi solicitud		
		V	V	I	V	V	V	solicitud	
		Arianna	Galán Quijala	Vacío	aquiala@estudiantes.uci.cu	Mi Web x Cuba	Esta es mi solicitud		
		V	V	V	I	V	V	solicitud	
Arianna	Galán Quijala	Facultad 3	Vacío	Mi Web x Cuba	Esta es mi solicitud				
V	V	V	V	I	V	solicitud			
Arianna	Galán Quijala	Facultad 3	aquiala@estudiantes.uci.cu	Vacío	Esta es mi solicitud				
V	V	V	V	V	I	solicitud			
Arianna	Galán Quijala	Facultad 3	aquiala@estudiantes.uci.cu	Mi Web x Cuba	Vacío				

Tabla 3.1 Descripción de caso de prueba. Realizar solicitud.

Nota: Las celdas de la tabla contienen V, I. V indica válido, I indica inválido.

Durante la realización de las pruebas al sistema se realizaron 13 Casos de Pruebas (uno para cada requisito funcional). Luego de una detallada revisión se encontraron 4 no conformidades en la primera iteración, las cuales fueron corregidas, comprobándose la eliminación de las mismas en una segunda iteración y obteniéndose como resultado de la misma 0 no conformidades.

3.5.4 Pruebas de carga y estrés

Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada. Para la realización de las pruebas de carga y estrés se define la utilización de la herramienta Jmeter⁶, la misma es una herramienta desarrollada en el lenguaje Java por el proyecto Apache. Esta aplicación brinda la posibilidad de realizar pruebas del rendimiento de la interfaz a través de la construcción de peticiones HTTP/HTTPS con cantidad de usuarios concurrentes variables, lo que posibilita obtener un resumen de los niveles de estrés del sistema y los límites de trabajo del mismo en condiciones extremas. La versión que se utiliza es la 2.3.4, que su instalación se encuentra en los repositorios de las distribuciones de Ubuntu.

Los requerimientos no funcionales que conforman el basamento de las pruebas realizadas son los siguientes:

- RNF 8.1 El sistema debe soportar un mínimo de 100 peticiones.
- RNF 8.2 Debe ser capaz de responder en un tiempo menor de 1 segundo para las páginas de información y 2 segundos para las operaciones de registro de contenidos.

Entorno de prueba

Previo a la realización de las pruebas se define el ambiente de prueba, con las características de hardware que poseen las estaciones de trabajo, las cuales se enuncian a continuación:

- microprocesador Intel Dual Core a 2.33 GHz
- memoria RAM de 1Gb

⁶ <http://jmeter.apache.org/>.

- disco duro de 250 GB
- sistema operativo Ubuntu 12.04.

Para realizar las pruebas de carga y estrés con la herramienta propuesta, se elabora un plan de pruebas que contiene todos los casos de pruebas necesarios a ejecutar, permitiendo la generación de reportes con los resultados obtenidos.

Las pruebas de carga y estrés se centran en la realización de peticiones HTTP a la interfaz, peticiones que son definidas a partir de los parámetros de entrada de las pruebas siguientes:

- Cantidad de usuarios conectados concurrentemente (CU): permite obtener los límites operables del núcleo bajo conexiones concurrentes.
- Número de repeticiones de las pruebas (NR): permite obtener rangos de tiempos de respuesta del sistema bajo un mismo número de usuarios concurrentes.
- Como resultados de la ejecución de las pruebas se definen los siguientes elementos de salida:
- Cantidad de muestras realizadas (CM): el número resultantes de la multiplicación de la cantidad de usuarios por el número de repeticiones de las pruebas.
- Tiempo promedio de respuesta (TPR): el tiempo promedio de los tiempos de respuesta de las peticiones que se realizan en milisegundos.
- Tiempo mínimo de respuesta (TMiR): el menor valor de respuesta obtenido de todas las muestras que se realizan en milisegundos.
- Tiempo máximo de respuesta (TMaR): el mayor valor de respuesta obtenido de todas las muestras que se realizan en milisegundos.
- Porcentaje de error (PE): el porcentaje del número de peticiones que se realizan en las que se obtuvo errores.
- Rendimiento (R): valor que indica el número de peticiones por segundo que se realizan.

La Tabla 3.2 muestra los resultados obtenidos de la ejecución de las pruebas de rendimiento a las funcionalidades de las interfaces de aplicación, Realizar solicitud y Mis solicitud, con los valores de los parámetros de entrada descritos anteriormente. Esta selección se justifica con la necesidad de poner a prueba los procesos de recuperación, inserción y actualización de las descripciones de los recursos, para validar el cumplimiento de los requerimientos establecidos en el diseño.

Funcionalidad	CU	NR	CM	TPR	TMiR	TMaR	PE	R
Realizar solicitud	1	10	10	28	25	30	0.00	34.5
	10	10	100	30	18	104	0.00	86.6
	100	10	1000	920	45	10265	0.00	76.3
	120	5	600	4006	46	63163	0.88	72.1
	150	5	510	7489	54	63165	3.66	67.9
Mis solicitud	1	10	10	69	55	185	0.00	14.3
	10	10	100	105	46	264	0.00	55.3
	100	10	1000	2216	17	31943	0.00	129.7
	110	5	550	3845	29	34854	0.00	132.2

Tabla 3.2 Descripción de los resultados de las pruebas de carga y estrés.

Del análisis de los resultados de las pruebas se puede concluir que la interfaz Realizar solicitud con 120 usuarios realizando 5 peticiones que equivalen a 600 conexiones concurrentes, en un tiempo promedio de 46 milisegundos observa un 0.88 % de peticiones fallidas incrementándose este valor a medida que se eleva el número de usuarios; mientras que la interfaz Mis solicitud con el acceso concurrente de 110 usuarios en un tiempo promedio de 29 milisegundos no presenta errores. Partiendo de estos resultados y teniendo en cuenta las características del servidor utilizado en las mismas se demuestra que las interfaces cumplen satisfactoriamente con los requerimientos establecidos.

3.5.5 Validación de las variables

A continuación se realiza la prueba de la validación de las variables para verificar que el problema planteado inicialmente ha sido solucionado.

En la actualidad	Después de terminar la investigación
-La solicitud se realiza mediante carta a la institución o por vía correo electrónico. -El especialista procesa la solicitud y decide si la aprueba o la rechaza, en caso de ser aprobada, solicita	-La solicitud se hace mediante la plataforma. -El especialista procesa la solicitud y decide si la aprueba o la rechaza, en caso de ser aprobada le notifica al solicitante mediante la plataforma, luego le

<p>al equipo de desarrollo para que instale y configure el nuevo evento en la plataforma.</p> <p>-El equipo de desarrollo le solicita al nodo el dominio y la base de datos para poder realizar la instalación y configuración del nuevo evento en la plataforma.</p> <p>-Cuando concluye el proceso el especialista le informa mediante correo electrónico o por carta al solicitante que ya su evento está montado en la plataforma y es cuando pasaría a ser uso de la misma.</p> <p>-En caso de no ser aprobada se le informa al solicitante mediante carta o por correo electrónico.</p>	<p>solicita al nodo la base de datos y el dominio, luego le informa al solicitante los datos para que instale y configure su perfil en la plataforma, de esta manera el solicitante instala y configura su perfil.</p> <p>- En caso de no ser aprobada se le notifica al solicitante mediante la plataforma.</p>
---	--

Tabla 3.3 Validación de las variables.

Con la validación de las variables se arribó que con el desarrollando de las funcionalidades para la gestión de solicitudes, instalación y configuración de los eventos científicos en la Plataforma Electrónica GRECIA, se permite realizar solicitudes de forma más eficiente e instalar y configurar un nuevo evento sin ser parte del equipo de desarrollo.

Conclusiones del capítulo

- Durante el desarrollo del capítulo fueron generados los artefactos correspondientes a la etapa de implementación realizándose el diagrama de despliegue y de componentes.
- Se definieron los estándares de codificación que se utilizaron durante la implementación del módulo y el perfil que posibilitaron crear un código de fácil comprensión.
- Para la validación del sistema se realizó pruebas a los requisitos funcionales y no funcionales a través del método de caja negra y la prueba de carga y estrés. El análisis de estos datos arrojó que la solución desarrollada se encuentra en condiciones para ser implantada.

Conclusiones

- Durante el desarrollo de la primera etapa de la investigación se definieron las herramientas y tecnologías a utilizar para el desarrollo de la solución, además se realizó un estudio de algunos de los sistemas existentes relacionados con el objeto de estudio, análisis que posibilitó fundamentar las bases teóricas de la solución propuesta.
- En un segundo período del desarrollo de la investigación, mediante el uso de técnicas para la captura de requisitos, fueron identificadas las principales funcionalidades de la solución donde se examinaron y seleccionaron los patrones de diseño a utilizar en la solución planteada, los cuales definieron fundamentalmente la calidad con que se desarrollaría la implementación del sistema.
- Durante la implementación de la solución propuesta se utilizó herramientas, tecnologías y patrones que se definieron en fases anteriores de la investigación. Los resultados de esta etapa se evidenciaron mediante el correcto funcionamiento de la aplicación, la que fue validada a través el análisis de los resultados obtenidos de pruebas funcionales realizadas al sistema mediante las pruebas de caja negra y de cargas y estrés obteniéndose resultados satisfactorios. La realización de estas pruebas permitió además encontrar y corregir funcionalidades defectuosas en la aplicación, las que fueron solucionadas en su totalidad.
- De forma general fueron solucionados todos los problemas existentes al inicio de la investigación, mediante la realización de las tareas definidas, obteniéndose como resultado que mediante una solicitud se pueda instalar y configurar un nuevo evento en la plataforma electrónica de gestión de eventos científicos GRECIA sin depender del equipo de desarrollo.

Recomendaciones

- Incorporar en futuras versiones del módulo configuración funcionalidades que permita gestionar los mensajes de registros en los sitios que alojan eventos.
- Incorporar en futuras versiones del Módulo Instalación funcionalidades que permitan, de forma no supervisada, crear nuevas bases de datos y dominios en el servidor de aplicación.

Bibliografías Referenciadas

1. Eventos virtuales [online]. S.l.: s.n. Available from: <http://www.marketresearchmedia.com/?p=421>.
2. Qué son los eventos virtuales [online]. S.l.: s.n. Available from: <http://eventosvirtuales.wordpress.com/2010/02/09/que-son-los-eventos-virtuales/>.
3. La UCI es fruto de la Batalla de Ideas | Atenas. Available from: <http://www.atenas.cult.cu/node/3098>.
4. Universidad de Ciencias Informáticas. Available from: <http://www.arqhys.com/arquitectura/universidad-ciencias-informaticas.html>.
5. SANCHÉZ RIVERO, JOSE MANUEL; ENRIQUEZ PALOMINO, ANTONIO; ABRIL SANCHEZ, CRISTINA. Guía para la integración de sistemas de gestión. S.l.: s.n., [no date]. ISBN 9788493961879.
6. Solicitud [online]. S.l.: s.n. Available from: <http://bitacoras.com/anotaciones/la-solicitud-concepto-caracteristicas-estructura-ejemplo/76196>.
7. JOSE LUIS RAYA CABRERA. Instalación y Configuración. S.l.: s.n., [no date]. ISBN 9788492650910.
8. FRAN GIL RODRÍGUEZ. Experto en drupal 7. S.l.: s.n., [no date].
9. Estudio de la Producción Científica y Tecnológica. In: [online]. Available from: <http://eventos.emagister.com/presentaciones/presentacion-del-andquot;estudio-de-la-produccion-cientifica-y-tecnologica-en-colaboracion-universi/643>.
10. Sistema de gestión de eventos universitarios [online]. S.l.: s.n. Available from: http://eventos.um.es/public_docs/presentacion-symposium.pdf.
11. Sistema de Conferencia EasyChair [online]. S.l.: s.n. Available from: <http://www.easychair.org/>.
12. Proceso de desarrollo de *software*. En Laboratorio de sistemas de información. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. S.l.: s.n.
13. J. Gestión y procesos en empresas de *software*. S.l.: s.n.
14. CMS. [online]. S.l.: s.n. Available from: <http://www.merca-tech.com.mx/articulos/47-ventajas-de-usar-un-cms.html>.
15. Sobre Drupal [online]. S.l.: s.n. Available from: <http://drupal.org.es/drupal>.
16. Qué es PHP [online]. S.l.: s.n. Available from: <http://www.desarrolloweb.com/articulos/392.php>.

17. HTML, CGIs, Java, JavaScript [online]. S.l.: s.n. Available from: <http://sopa.dis.ulpgc.es/zoso/info.html.html>.
18. PRESSMAN, R. S. Ingeniería del *Software*: Un Enfoque Práctico. quinta. S.l.: s.n., [no date]. ISBN 8448132149.
19. Apache [online]. S.l.: s.n. Available from: http://linux.ciberaula.com/articulo/linux_apache_intro.
20. Entornos integrados de desarrollo [online]. S.l.: s.n. Available from: <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>.
21. NetBeans [online]. S.l.: s.n. Available from: http://netbeans.org/index_es.html.
22. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [online]. S.l.: s.n. Available from: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
23. MYSQL [online]. S.l.: s.n. Available from: <http://www.mysql.com>.
24. Visual Paradigm [online]. S.l.: s.n. Available from: <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm>.
25. ROQUE, D. M. Y PERAZA, A. G. Modelo de dominio [online]. S.l.: s.n. Available from: http://www.ecured.cu/index.php?title=Modelo_de_dominio&oldid=1481134.
26. IAN SOMMERVILLE. Ingeniería de *Software*. Séptima. S.l.: s.n., [no date]. ISBN 84-7829-074-5.
27. *Experto en Drupal 7. Nivel avanzado*. Forcontu S.L, 2012b. 43-44 p. ISBN 978-84-939410-5-5.
28. PRESSMAN, R. S. Ingeniería del *Software*: Un Enfoque Práctico. 6ta. Nueva York, E.U.A: s.n., [no date].
29. SOMMERVILLE. Ingeniería del *software*. Séptima. Madrid: s.n., [no date].
30. Tipos de prueba [online]. S.l.: s.n. Available from: http://issuu.com/diegoma363/docs/2.tipos_de_pruebas.
31. PRUEBAS FUNCIONALES [online]. S.l.: s.n. Available from: http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

Anexo. Entrevista

Identificar los problemas existentes en la plataforma de gestión de eventos científicos y comprender el funcionamiento de la misma, así como los mecanismos que emplean para darle solución a las dificultades.

Preguntas:

¿Quién se encarga de velar y controlar la actividad científica en la UCI, específicamente en la gestión de eventos?

¿Quiénes dentro de la Universidad gestiona eventos?

¿Cuáles son los eventos, revistas, etc.?

¿Qué problemas presenta la gestión de eventos en la UCI y qué acciones se realizan actualmente para mitigar esas deficiencias?

¿Cuáles son los errores más frecuentes detectados por la Dirección de Investigaciones?

¿Qué es lo que necesita que el sistema realice?

¿Cuáles son sus expectativas del sistema para los conceptos fácil de usar, confiable, rendimiento?

¿Cuáles son las funciones que le gustaría agregar?

¿Cuáles son las funciones que le gustaría mejorar con relación a sistemas similares?

¿Quiénes son los usuarios del sistema?

¿A qué funcionalidades tendrá acceso cada usuario?

Anexo. Descripción de los requisitos.

Precondiciones	Usuario autenticado en la plataforma
Flujo de eventos	
Flujo básico Registrar Incidencia	
1	Seleccionar la opción de realizar solicitud.
2	Muestra una pantalla para introducir los datos de la solicitud.
3	Introduce los datos.
4	Envía la solicitud.
5	Valida los datos de la solicitud en caso de algún error ver <u>Flujo alternativo 1.</u>
6	Valida que la solicitud no existe en el sistema, sino <u>Flujo alternativo 2.</u>
7	Registrar la solicitud.
8	Concluye el requisito.
Pos-condiciones	
1	Se guardan los datos de la Solicitud.
Flujos alternativos	
Flujo alternativo 1.a Error en los datos de la solicitud	
1	El sistema señala en rojo el campo con los datos incorrectos.
2	Muestra un mensaje de error.
3	Continúa en el punto 3.
Flujo alternativo 2.a Ya existe la solicitud	
1	Se muestra un mensaje “Ya existe la solicitud”.
2	Continúa en el punto 2.
Pos-condiciones	
1	N/A
Validaciones	
1	N/A
Requisitos especiales	N/A
Asuntos	N/A

pendientes

Prototipo elemental de interfaz gráfica de usuario

Crear Solicitud

Nombre *
Escribe su nombre

Apellidos *
Escribe su dos apellidos

Correo *
Escribe su correo electrónico

Afiliación

Temática

Descripción

Documento
Examinar... Subir al servidor
Los archivos deben ser menores que 1 MB.
Tipos de archivo permitidos: doc...

Propuesta de URL

Estado de la solicitud
Cambiar Estado de la solicitud
Pendiente

Guardar Cancelar

Formatos de entrada/salida

Entradas

N/A

Salidas

N/A

Precondiciones Usuario autenticado en la plataforma

Flujo de eventos

Flujo básico Modificar solicitud.

- 1 Seleccionar la opción de modificar solicitud.
- 2 Seleccionar la opción de editar.
- 3 Modifica los datos.

-
- 4 Envía la solicitud.
 - 5 Valida los datos de la solicitud en caso de algún error ver Flujo alternativo 1.
 - 6 Modifica la solicitud.
 - 7 Concluye el requisito.
-

Pos-condiciones

- 1 Se guardan los datos de la Solicitud.
-

Flujos alternativos**Flujo alternativo 1.a Error en los datos de la solicitud**

- 1 El sistema señala en rojo el campo con los datos incorrectos.
 - 2 Muestra un mensaje de error.
 - 3 Continúa en el punto 3.
-

Pos-condiciones

- 1 N/A
-

Validaciones

- 1 N/A
-

Requisitos especiales N/A

Asuntos pendientes N/A

Prototipo elemental de interfaz gráfica de usuario

Editar Solicitud Solicitud #12

Ver Editar

Nombre *
Robin

Apellido *
Senciel Terrero

Correo *
rsenciel@estudiantes.uci.cu

Afiliación
Universidad de las Ciencias Informáticas

Temática
Ciencias de la Computación

Descripción
Una invitación

Documento
Examinar... Subir al servidor

Propuesta de URL
http://www.cubadebate.cu

Guardar Eliminar Cancelar

Formatos de entrada/salida

Entradas

N/A

Salidas

N/A

Precondiciones Usuario autenticado en la plataforma

Flujo de eventos

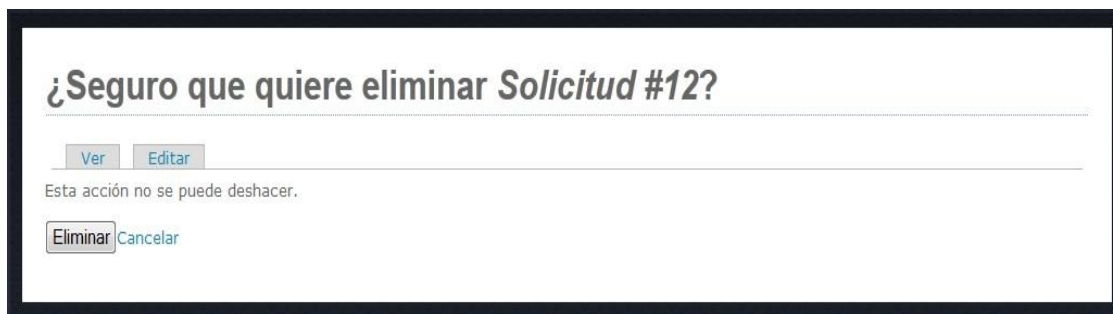
Flujo básico Eliminar solicitud.

- 9 Seleccionar la opción de Eliminar solicitud
- 10 Muestra la solicitud.
- 11 Selecciona elimina solicitud.
- 12 Elimina la solicitud
- 13 Concluye el requisito.

Pos-condiciones

- 2 Se borra la Solicitud.

Pos-condiciones	
2	N/A
Validaciones	
2	N/A
Requisitos especiales	N/A
Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	



Formatos de entrada/salida

Entradas

N/A

Salidas

N/A

Precondiciones	Usuario autenticado en la plataforma
Flujo de eventos	
Flujo básico Listar solicitud.	
14	Seleccionar la opción de Listar solicitud
15	Muestra una pantalla con la opción buscar solicitud.
16	Introduce el criterio de búsqueda de la solicitud.
17	Busca las solicitudes en caso de que no exista <u>Flujo alternativo 1</u> .
18	Muestra la lista de solicitudes.

19 Concluye el requisito.

Pos-condiciones

N/A

Flujos alternativos

Flujo alternativo 1.a Las solicitudes no existe

4 El sistema muestra un mensaje “No hay solicitud que mostrar”.

5 Continúa en el punto 3.

Pos-condiciones

3 N/A

Validaciones

3 N/A

Requisitos especiales N/A

Asuntos pendientes N/A

Prototipo elemental de interfaz gráfica de usuario

Listado de solicitudes pendientes

PENDIENTES APROBADAS RECHAZADAS LISTAS

Nombre Afiliación Temática

Buscar

No hay solicitudes que mostrar.

[XLS](#)

Formatos de entrada/salida

Entradas

N/A

Salidas

N/A

Precondiciones

Usuario autenticado en la plataforma

Flujo de eventos

Flujo básico Aprobar solicitud.

- 20 Seleccionar la opción de Aprobar solicitud
- 21 Envía un correo al usuario donde se le notifica que se le aprobó la solicitud
- 22 Concluye el requisito.

Pos-condiciones

N/A

Pos-condiciones

4 N/A

Validaciones

4 N/A

Requisitos especiales N/A

Asuntos pendientes N/A

Prototipo elemental de interfaz gráfica de usuario

Listado de solicitudes aprobadas

PENDIENTES APROBADAS RECHAZADAS LISTAS

Nombre Afiliación Temática

Buscar

NO.	TÍTULO	CORREO	FECHA DE REGISTRO	SOLICITANTE	OPERACIONES
1	Solicitud #12	rsencial@estudiantes.uci.cu	Viernes, Junio 21, 2013 - 11:20	Arianna	Ver detalles Editar Eliminar Registrar configuración

XLS

Formatos de entrada/salida**Entradas**

N/A

Salidas

N/A

Precondiciones

Usuario autenticado en la plataforma

Flujo de eventos**Flujo básico Rechazar solicitud.**

- 23 Seleccionar la opción de Rechazar solicitud.
- 24 Envía un correo al usuario donde se le notifica que se le rechaza la solicitud
- 25 Concluye el requisito.

Pos-condiciones

N/A

Pos-condiciones

5 N/A

Validaciones

5 N/A

Requisitos especiales N/A

Asuntos pendientes N/A

Prototipo elemental de interfaz gráfica de usuario

Listado de solicitudes rechazadas

PENDIENTES APROBADAS RECHAZADAS LISTAS

Nombre Afiliación Temática

Buscar

NO.	TÍTULO	CORREO	FECHA DE REGISTRO	SOLICITANTE	OPERACIONES
1	Solicitud #13	aquiala@estudiantes.uci.cu	Viernes, Junio 21, 2013 - 17:56	webmaster	Ver detalles Editar Eliminar

XLS

Formatos de entrada/salida**Entradas**

N/A

Salidas

N/A

Anexo. Descripción de casos de pruebas

Escenario	Descripción	Nombre de solicitante	Apellidos del solicitante	Afiliación	Correo electrónico	Temática	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Realizar solicitud correctamente insertando los datos correctamente	Mediante este escenario se realiza la solicitud de un evento.	V <i>Arianna</i>	V <i>Galán Quijala</i>	V <i>Facultad 3</i>	V <i>aquijala@estudiantes.uci.cu</i>	V <i>Mi Web x Cuba</i>	V <i>Esta es mi solicitud</i>	<i>El sistema actualiza los datos y muestra un mensaje de confirmación: "Se enviado la solicitud".</i>	<i>Luego de acceder al sistema el usuario escoge la opción realizar solicitud, el sistema muestra el formulario de solicitud. El usuario llena los campos correctamente y selecciona el botón "enviar".</i>
EC 1.2 Realizar solicitud incorrecta	Mediante este escenario se	<i>i</i> <i>Vacío</i>	V <i>Galán Quijala</i>	V <i>Facultad 3</i>	V <i>aquijala@estudiantes</i>	V <i>Mi Web x Cuba</i>	V <i>Esta es mi solicitud</i>	<i>El sistema muestra</i>	<i>Luego de acceder al</i>

mente insertand o los datos correctam ente	realiza la solicitud de un evento.				.uci.cu			un mensaje de alerta "Por favor verifique que los datos introduc os estén correctos ".	sistema el usuario escoge la opción realizar solicitud, el sistema muestra el formulari o de solicitud. El usuario llena los campos correcta mente y seleccion a el botón "enviar".
		V Arianna	I Vacío	V Faculta d 3	V aquiala @estu diantes .uci.cu	V Mi Web x Cuba	V Esta es mi solicitud		
		V Arianna	V Galán Quiala	I Vacío	V aquiala @estu diantes .uci.cu	V Mi Web x Cuba	V Esta es mi solicitud		
		V Arianna	V Galán Quiala	V Faculta d 3	I Vacío	V Mi Web x Cuba	V Esta es mi solicitud		
		V Arianna	V Galán Quiala	V Faculta d 3	V aquiala @estu diantes .uci.cu	I Vacío	V Esta es mi solicitud		
V Arianna	V Galán Quiala	V Faculta d 3	V aquiala @estu diantes .uci.cu	V Mi Web x Cuba	I Vacío				