

Universidad de las Ciencias Informáticas

Facultad 4



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

**“APLICACIÓN WEB PARA LA GESTIÓN DE INFORMACIÓN DE
INVESTIGACIÓN Y POSTGRADO DE LA FACULTAD 4”**

Autor: Emilio Alvarez Muñoz

Tutor: Ing. José A. Soto Pérez

Ciudad de la Habana, 20 de junio de 2011

“Año 53 del Triunfo de la Revolución”

Declaración de autoría

Declaro que soy el único autor de la presente tesis con título: **Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4** y autorizo a la Universidad de las Ciencias Informáticas; como al centro de producción de la facultad 4 para que hagan el uso que estimen pertinente con este trabajo.

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

Autor

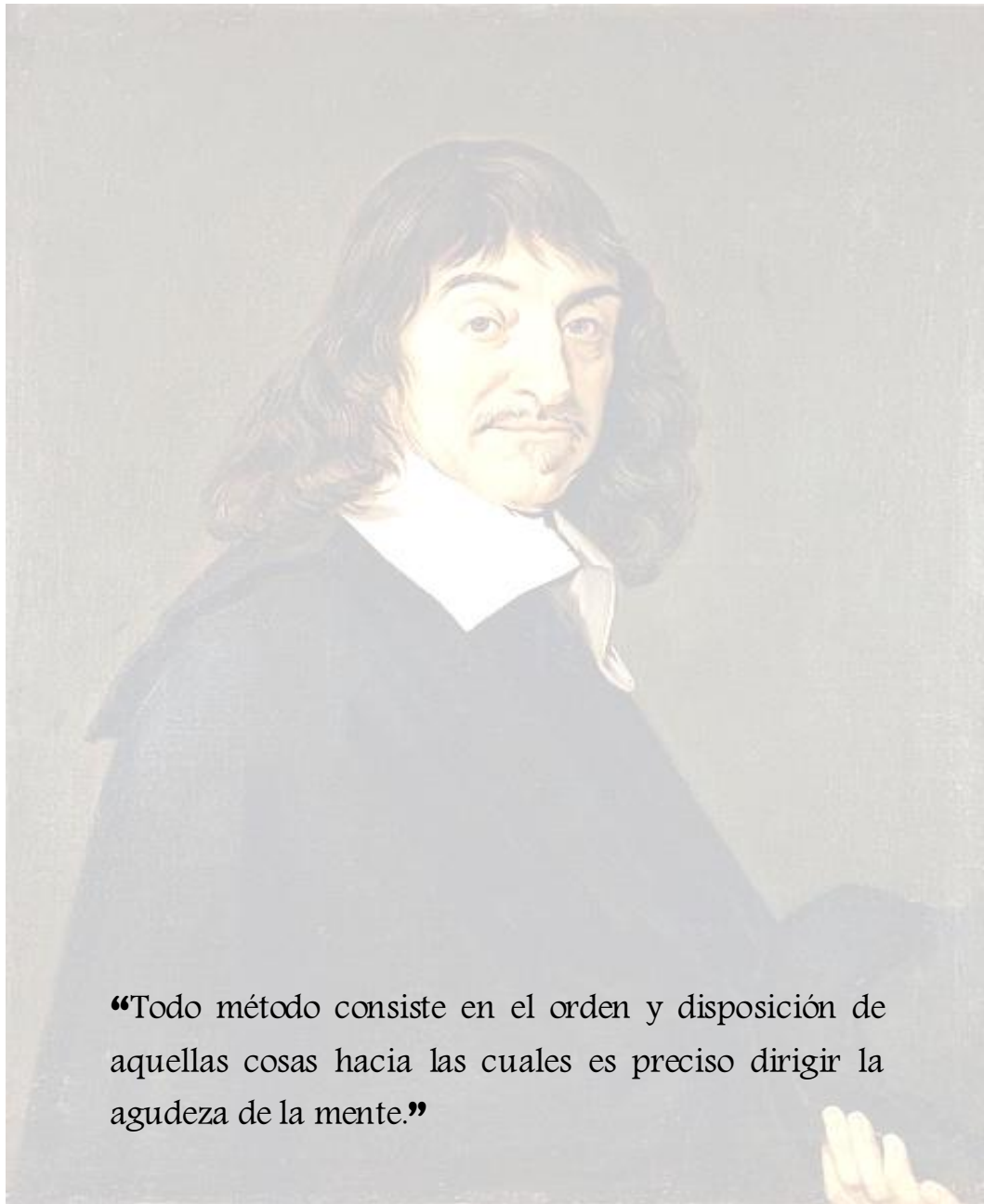
Emilio Alvarez Muñoz

Firma del autor

Tutor

Ing. José Antonio Soto Pérez

Firma del tutor



René Descartes.

Agradecimientos

A Mami Lidia y mima Chefa por ser las madres más dedicadas y brindarme su infinito amor y cariño.

A mi hermana, hermano y sobrinos por apoyarme siempre, les quiero mucho.

A toda mi familia de manera general, en especial a Cacha, Mercedes, tía Luz María, mi prima Magalys, Ismaris.

A Suri la bella, Castríbiri, Mirian (machupichu) y Aballe que los considero mi familia.

A mis amigos todos, que durante todos estos años de estudio me han acompañado en las buenas y en las malas.

A todas aquellas personas que de una forma u otra formaron parte de mi desarrollo como futuro profesional.

A todas las personas que de una forma u otra han hecho posible el desarrollo de esta tesis...

Dedicatoria

Dedico este trabajo a la persona que más quiero y admiro en el mundo “mi mamá”, Lidia Muñoz Genes, a la cual le debo lo que soy y lo que seré en el futuro. Primeramente por haberme dado el privilegio de la vida, en la que me ha entregado, su amor, dedicación y esmero, por haber influido directamente en mi educación y formación, con sacrificio y apoyo incondicional, por haberme dado la oportunidad de contar con ella en todas las etapas de mi vida. Por esto y tantas razones le doy gracias, pues mi mayor anhelo es estar a la altura del ejemplo que constituye para mí.

Resumen

La información es un conjunto de datos que componen un mensaje sobre cierto fenómeno en determinado contexto. Permite resolver problemas y tomar decisiones, ya que su uso es la base del conocimiento. En la actualidad, los volúmenes de información existentes son elevados y la gestión de la misma llega, en ocasiones, a convertirse en un proceso tedioso si no se realiza de forma correcta.

Este trabajo consiste en la creación de una *“Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4”*, en respuesta a la necesidad de mejorar y perfeccionar, la manera en que actualmente se llevan a cabo estas tareas. La utilización de un sistema informatizado para la gestión de la información, significaría una mejora considerable de este proceso, en cuanto a eficiencia, tiempo y organización contribuyendo favorablemente al desarrollo del *“Departamento de investigación y postgrado de la Facultad 4”*.

Con el propósito de elevar el nivel de calidad, en la gestión de información y en las tareas que se realizan en el *“Departamento de investigación y postgrado de la facultad 4”*, se realizó una investigación centrándose principalmente en la implementación de una Aplicación web, que gestionará de manera automática todo este gran cúmulo de información.

En el presente trabajo se plasman los conceptos asociados al dominio del problema. Se explica de forma detallada, el análisis realizado para la selección de las herramientas que se utilizan en la implementación de este gestor de información y se muestra como se realizó la misma.

PALABRAS CLAVES:

Volúmenes de información, proceso, sistema informático, gestión de información, aplicación web.

Índice de tablas

Tabla 1: HU - Gestionar usuario	34
Tabla 2: HU - Autenticar usuario	35
Tabla 3: HU - Gestionar foro	35
Tabla 4: HU - Gestionar artículos	35
Tabla 5: HU - Gestionar documentos	35
Tabla 6: HU - Gestionar cursos	36
Tabla 7: HU - Gestionar eventos	36
Tabla 8: HU - Gestionar solicitud de eventos	36
Tabla 9: HU - Gestionar revisión de trabajos	36
Tabla 10: HU - Revisar trabajos	37
Tabla 11: HU - Gestionar actividad postgraduada	37
Tabla 12: HU - Buscar actividad postgraduada	37
Tabla 13: HU - Generar reportes	37
Tabla 14: Historias de usuarios distribuidas por módulos	38
Tabla 15: Tiempo real de implementación por módulos. Iteración 1	41
Tabla 16: Tarea 1 - Desarrollar CRUD de solicitud de eventos - Módulo investigación	42
Tabla 17: Tarea 2 - Desarrollar CRUD de revisión de trabajos - Módulo investigación	42
Tabla 18: Tarea 3 - Permitir la revisión de trabajos - Módulo investigación	43
Tabla 19: Tarea 4 - Desarrollar CRUD de actividad postgraduada - Módulo postgrado	43
Tabla 20: Tarea 5 - Buscar actividad postgraduada - Módulo postgrado	43
Tabla 21: Tiempo real de implementación por módulo. Iteración 2	44
Tabla 22: Tarea 1 - Generar reportes para el balance - Módulo balance	44
Tabla 23: CP01 - Gestionar solicitar evento	49
Tabla 24: CP02 - Gestionar solicitar evento	50
Tabla 25: CP03 - Gestionar solicitar evento	51
Tabla 26: CP04 - Gestionar solicitar evento	52
Tabla 27: CP01 - Gestionar revisión de trabajos	52
Tabla 28: CP02 - Gestionar revisión de trabajos	53
Tabla 29: CP03 - Gestionar revisión de trabajos	55
Tabla 30: CP01 - Revisar trabajos	55
Tabla 31: CP01 - Gestionar actividad postgraduada	56
Tabla 32: CP02 - Gestionar actividad postgraduada	57
Tabla 33: CP03 - Gestionar actividad postgraduada	58
Tabla 34: CP0 - Buscar información postgraduada	59

Índice de ilustraciones

Ilustración 1: Metodología Extreme Programming.....	16
Ilustración 2: Arquitectura de Drupal	39
Ilustración 3: Modelo de datos del sistema.....	40
Ilustración 4: Diagrama de componentes del CMS Drupal	45
Ilustración 5: Diagrama de componentes de los módulos en Drupal	46
Ilustración 6: Diagrama de despliegue	47

Índice de contenidos

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 ESTADO DEL ARTE	5
1.1.1 ¿Qué es un sistema de gestión de información?	5
1.1.2 Análisis de soluciones similares	6
1.2 TENDENCIAS Y TECNOLOGÍAS ACTUALES	10
1.2.1 Metodologías de desarrollo de software	10
1.2.2 Herramientas de Modelado	17
1.2.3 Lenguajes de programación	19
1.2.4 Servidores web	22
1.2.5 Sistema Gestor de Base de Datos (SGBD)	24
1.2.6 Sistemas de Gestión de Contenidos (CMS)	26
1.2.7 Entorno de Desarrollo Integrado (IDEs)	28
1.2.8 Arquitectura	30
CAPÍTULO 2. CONSTRUCCIÓN Y PRUEBA DE LA SOLUCIÓN	33
2.1 INTRODUCCIÓN	33
2.2 PROPUESTA DE SOLUCIÓN	33
2.2.1 Requerimientos no funcionales	33
2.2.2 Historias de usuario (HU)	34
2.3 ARQUITECTURA DE DRUPAL	38
2.4 DISEÑO DE LA BASE DE DATOS	40
2.5 FASE DE IMPLEMENTACIÓN	41
2.5.1 Plan de iteraciones	41
2.5.2 Diagrama de componentes	45
2.5.3 Diagrama de Despliegue.	47
Conclusiones parciales	48
2.6 FASE DE PRUEBA	48
2.6.1 Pruebas de aceptación	49
Conclusiones parciales	60
CONCLUSIONES GENERALES	61
RECOMENDACIONES	62
REFERENCIAS BIBLIOGRÁFICAS	63
ANEXOS	66
GLOSARIO DE TÉRMINOS	71

Introducción

El incesante avance de las ciencias y las tecnologías ha marcado un paso importante en el progreso de la humanidad, demostrando la necesidad de crear nuevas herramientas y funcionalidades que satisfagan las necesidades existentes y la confección de productos cada vez más complejos. La informática se ha extendido a todos los sectores de la sociedad y la rápida evolución de los sistemas informáticos presiona a mantener el ritmo de dicho desarrollo.

Cuba a pesar de ser un país bloqueado, con escasas posibilidades dentro del amplio mundo del desarrollo de las tecnologías, la información y las comunicaciones, no está exenta al desarrollo tecnológico existente. Pues a pesar de todas las dificultades que afronta, para adquirir medios tecnológicos y avanzar en el mundo de la informática, reconoce la necesidad de lograr un auge en el desarrollo de este campo; es por eso por lo que numerosas sedes universitarias, así como centros de trabajo demandan la optimización de la gestión de la información.

La Universidad de las Ciencias Informáticas (UCI), primera universidad creada al calor de la batalla de ideas, está inmersa en el proceso de informatización que se lleva a cabo en el país, con la utilización de las Tecnologías de la Información y las Comunicaciones (TIC).

En la Facultad 4 específicamente en el Vicedecanato de Investigación y Postgrado, es de vital importancia toda la información referente a eventos y a los cursos de superación postgraduada (que son los estudios de especialización posteriores al título de grado).

Estos cursos constituyen la estrategia principal para la formación de profesionales del más alto nivel. Son llevados a cabo en universidades o cualquier Instituto de Educación Superior, por individuos que poseen previamente un título universitario de primer nivel como antecedente obligatorio. Son desarrollados en dos vertientes, la primera es la Superación Profesional que propicia cursos, entrenamientos, talleres y diplomados; mientras que la otra línea se centra en la Superación Académica con maestrías, especialidades, doctorados y pos-doctorados. Todas estas actividades pueden desarrollarse en diferentes modalidades, como son: a tiempo completo, parciales o diferentes grados de comparecencia, a saber, presencial, semi-presencial o a distancia.

Actualmente existe en la universidad una herramienta llamada *“Alfresco”* que se encarga de la gestión documental, pero no satisface las necesidades del Vicedecanato de Investigación y Postgrado pues no gestiona el tipo de información que realmente se necesita en este local y no cuenta con las funcionalidades requeridas, como por ejemplo, la gestión de eventos: convocatorias, publicaciones, recepción de trabajos, revisión de trabajos en línea, foros; así como, la actualización de la superación postgraduada de los profesores: reportes y actualizaciones hechas por los asesores de los Departamentos.

El hecho de que la totalidad de esta información se obtenga manualmente o vía correo trae como consecuencia la desactualización, la pérdida de datos y la descentralización de la información. Ralentizando su tratamiento, dificultando la obtención de reportes exactos e implicando mucho más esfuerzo, más tiempo y un aumento considerable en el margen de error. Debido a la situación anteriormente expuesta es posible plantear el siguiente **problema a resolver**: ¿Cómo facilitar la gestión de la información de investigación y postgrado de la Facultad 4?

Se define como **objeto de estudio**: Sistemas de gestión de la información.

Campo de acción: Sistemas de gestión de la información en el área de investigación y postgrado de la Facultad 4.

Siendo el **objetivo general** de la presente investigación, desarrollar una *“Aplicación web para la gestión de la información de investigación y postgrado de la Facultad 4”*, que permita una mejor organización en todo lo referente a este proceso.

Para darle cumplimiento al presente trabajo se cuenta con los siguientes **objetivos específicos**:

- » Investigar acerca del proceso informativo en el Vicedecanato de Investigación y Postgrado de la Facultad 4.
- » Analizar las soluciones o aplicaciones similares existentes en la actualidad.
- » Analizar las herramientas de modelado e implementación, así como, la metodología a utilizar en la construcción del sistema.

- » Definir las funcionalidades del sistema a implementar.
- » Efectuar la implementación de la aplicación.
- » Validar la propuesta de solución.

Para encaminar la investigación con vista a resolver el problema planteado se propone la siguiente **idea a defender**: Con el desarrollo de la “*Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4*”, se podrá tener un eficiente control sobre la labor investigativa y de superación postgraduada del profesorado de la Facultad 4.

Para el desarrollo de la aplicación se cuenta con una serie de **tareas** que ayudarán a cumplimentar su construcción:

- » Análisis de soluciones o aplicaciones similares existentes en la actualidad.
- » Selección de las herramientas de modelado e implementación, así como, la metodología a utilizar para la construcción del sistema.
- » Diseño del prototipo de Interfaz de Usuario.
- » Implementación de las funcionalidades del sistema.
- » Evaluación de la calidad del producto a través de pruebas.
- » Estructura y redacción correcta del informe de tesis.

Para darle cumplimiento a las tareas antes mencionadas se hace necesaria la utilización de los siguientes **métodos de investigación**:

Métodos teóricos:

- » **Histórico – lógico**: Para llevar a cabo una mejor comprensión del estado actual de las aplicaciones Web desarrolladas con herramientas libres, así como, otras temáticas estrechamente relacionadas con la investigación.
- » **Analítico – sintético**: Para procesar la información en la elaboración de los fundamentos teóricos y de las conclusiones.

- » **Modelación:** Mediante la utilización del lenguaje de modelado UML, permitirá reflejar la estructura, relaciones internas y características de la solución a través de diagramas. Este método facilitará también el diseño de clases, así como otros artefactos necesarios para la implementación de la aplicación.

Métodos empíricos:

- » **Entrevista:** Para hacerles preguntas a los stakeholders (usuarios a los cuales va dirigido la aplicación) sobre el sistema que utilizan y sobre el sistema a desarrollar. Los requerimientos provienen de las respuestas a estas preguntas.
- » **Encuesta:** Se obtiene información primaria sociológica mediante preguntas orales o escritas al personal investigado.

Al concluir este trabajo se esperan como **posibles resultados:** la realización de una aplicación Web para la gestión de eventos y cursos de postgrado que permitirá; mantener informado a todo el colectivo de profesores, la recepción de trabajos, foros donde los profesores puedan brindar sus comentarios, publicaciones, convocatorias tanto a eventos como a cursos, generar reportes y tener un control acerca de la situación del profesorado en cuanto a participación en eventos y cursos acreditados.

El presente documento se estructura por varias secciones que incluyen dos capítulos. Para lograr una mejor comprensión de estas secciones, a continuación, se brinda un pequeño resumen:

Capítulo 1. Fundamentación Teórica: Se describen los conceptos fundamentales asociados al dominio del problema, se expone el estado del arte del tema tratado, se hace un análisis entre las soluciones ya existentes; tanto a nivel nacional como internacional, explica y justifica además, las tendencias, tecnologías y herramientas en las que se apoya la solución al problema.

Capítulo 2. Implementación y Prueba: En este capítulo se realiza una breve descripción de las historias de usuario definidas por el cliente. Además, se exponen de manera concisa a modo de introducción algunas características y artefactos del diseño, que ayudarán a un mejor entendimiento de la aplicación y de esta forma realizar con la calidad requerida todo el trabajo de construcción de la misma. El sistema, debe realizarse de forma iterativa e incremental, obteniéndose al final de cada iteración un producto funcional que debe ser probado y mostrado al cliente, permitiendo de esta forma lograr una constante retroalimentación entre los desarrolladores y el cliente.

Capítulo 1. Fundamentación teórica

En la actualidad la complejidad de los sistemas de *software* y los exigentes tiempos de desarrollo, manifiestan cada vez más la necesidad de eficiencia y aprovechamiento del tiempo, en cada una de las tareas que se realizan durante el ciclo de vida del *software*. Es así como han surgido en el ámbito computacional, asistentes que ayudan a optimizar tareas y a generar código automáticamente.

En este capítulo se hace referencia a las tecnologías, lenguajes de programación y herramientas utilizadas para el desarrollo de Aplicaciones web, permitiendo realizar una selección correcta que se adapte a las necesidades del sistema propuesto, lográndose identificar el Lenguaje de programación, el Sistema gestor de base de datos, la Metodología de desarrollo de *software* adecuada y las herramientas a utilizar en el desarrollo de la aplicación.

1.1 Estado del arte

1.1.1 ¿Qué es un sistema de gestión de información?

“Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización”. (1) Un Sistema de gestión de información puede ser definido como un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades que se realizan en una organización o para automatizar los procesos de trabajo que se efectúan dentro de esta.

Se puede identificar además, como un conjunto de datos modificados o combinados entre sí, que poseen un significado potencialmente útil para el usuario que los recibe. Teniendo así vital importancia en el cómo y dónde se almacenan estos datos. Obteniendo los mismos, un nivel de organización estricto que permita encontrar todo ese cúmulo de información en el menor tiempo y lo más estructurado posible.

El desarrollo de un Sistema de gestión de información puede proporcionar numerosos beneficios, dentro de los que se pueden mencionar la optimización del tiempo en la búsqueda de documentos, la reducción de los riesgos de pérdida del documento físico original, el acceso concurrente a un documento y el hecho de disponer de la información de forma centralizada y rápidamente accesible.

1.1.2 Análisis de soluciones similares

Los sistemas informáticos facilitan las actividades que deben realizar cualquier empresa o compañía y estos contribuyen a la optimización de los resultados y mejor aprovechamiento de los recursos, buscando mayor eficiencia en los procesos que se realizan y mejores resultados en la producción. Es por eso por lo que hoy en día la demanda es incesante.

Internacional

» **Sistema de Información de Gestión Académica (SIGA)**

SIGA es un producto que permite recopilar y tratar la información de los distintos niveles institucionales, lo que relaciona la labor docente, investigativa, de creación artística y extensión, permitiendo conocer cuantitativa y cualitativamente las iniciativas desarrolladas por los distintos organismos de la Institución.

A partir de los datos e información ingresada por las diversas unidades de la Institución, es posible generar indicadores y reportes de análisis en docencia de pregrado; postgrado, de actividad académica, de investigación y extensión.

Presenta módulos como: personal académico y actividad académica donde trata todo lo referente a docencia y a actividades tanto de extensión como difusión. (27)

Este sistema implementa muchas de las funcionalidades que se requiere en este trabajo, pero, no permite mantener una constante interactividad entre los profesores. Que defina una mejor gestión de lo que debe realizar la dirección de investigación y postgrado, para poder suplir todas las interrogantes establecidas por los profesores involucrados.

» **Sistema de Estudios de Postgrado (SEP)**

SEP fue desarrollado en la Universidad de Costa Rica, se dedica específicamente a organizar, orientar y facilitar el desarrollo de los programas de maestría, doctorado y demás especialidades para una mejor preparación de los profesionales. El SEP construye su estrategia enmarcado en una institución que se redefine y se reorienta, además, rodeado por un entorno altamente dinámico y con importantes perspectivas de crecimiento.

SEP ha desarrollado programas de postgrado que atienden las áreas clave para el desarrollo sostenible de Costa Rica y de la región centroamericana, así como programas multidisciplinarios y otros de carácter interinstitucional. (26)

En este sentido SEP es un sistema bastante completo con respecto a la organización de manera centralizada de toda la información referente a las maestrías, doctorados, el cual podría implantarse en la facultad 4 si no fuera por el hecho de ser un sistema privativo.

Nacional

» Universidad virtual de postgrado (IPLAC-Virtual)

El Instituto Pedagógico Latinoamericano y Caribeño, IPLAC, Universidad Pedagógica de Postgrado, con sede en La Habana, Cuba. Pone a disposición de los profesionales de la educación diversos cursos, en temas de Ciencias Pedagógicas y Ciencias de la Educación que abarcan todas las esferas del quehacer educativo. Los mismos se desarrollan de manera presencial o a distancia con el apoyo de las Tecnologías de la Información y la Comunicación (TIC). (28)

Sus objetivos básicos son:

- » Realizar actividades de docencia, investigación y transferencia de conocimientos en materia pedagógica y cultural, propendiendo a la sistematización y enriquecimiento de un pensamiento regionalmente situado y atento a las necesidades y realidades de sus países y organizaciones. Producir bienes y prestar servicios que sirvan para la jerarquización de la profesión docente y cultural en América Latina y el Caribe.

Objetivos específicos del IPLAC:

- » Propiciar la superación profesional y la formación postgraduada de los educadores de América Latina y el Caribe en lo referente a las Ciencias de la Educación, a través de programas específicos para maestros, profesores, directivos de la educación, mediante la modalidad presencial y a distancia con el empleo de las TIC (en particular la educación virtual).
- » Contribuir al desarrollo de las investigaciones pedagógicas, sociológicas y los estudios de la educación comparada en América Latina y el Caribe. Entre los servicios que se ofrecen están:
 - » Información referente a los diferentes postgrados en ofertas organizados por unidades temáticas.
 - » Tiene un control detallado de todos los docentes integrados a esta Universidad Virtual. Ofrece información específica de los datos profesionales de los mismos.

Capítulo 1. Fundamentación teórica

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

- » Da a conocer el correcto procedimiento para el pago de la matrícula.
- » Ofrece un formulario de registro con los datos necesarios para la inscripción a los postgrados deseados en un período dado.

IPLAC es un sistema que cuenta con muchas funcionalidades que permiten una gestión de la información de investigación y postgrado del profesorado, pero, no se adecua a las necesidades específicas existentes en la actualidad en el departamento de investigación y postgrado de la Facultad 4. El mismo no cuenta con una funcionalidad que permita subir y descargar documentos asociados a todo lo referente a la investigación y postgrado, pero además, se necesita poder publicar todos los artículos que de una forma u otra ayudan a una mejor superación de los profesores involucrados.

» **Sistema de gestión de la información de un Departamento Docente, Universidad de Matanzas**

En la Universidad de Matanzas “Camilo Cienfuegos”, se implementó un sistema de gestión de información de un departamento docente desarrollando una Aplicación web y utilizando PHP como lenguaje de programación. Este sistema está constituido por cuatro módulos: (29)

- » Módulo de Capital Humano
- » Módulo de Formación del Profesional
- » Módulo de Educación de Postgrado
- » Módulo de Ciencia y Técnica

Estos módulos facilitan la gestión de la información del capital humano de un departamento docente, dentro de ellos quedan registrados los datos personales de los trabajadores del departamento, el cumplimiento del plan de trabajo, los objetivos del profesor, de sus evaluaciones, etc. Se controlan los datos referentes a los asesoramientos a otros profesores. También se desarrolla todo el trabajo de planificación de la carga docente de los profesores y se gestiona la documentación de la educación posgraduada: cursos, maestrías, diplomados, doctorados que se ofertan o son recibidos por profesores del departamento, obteniendo reportes como el plan de postgrados del departamento, el estado de la superación del claustro, su planificación y control. (29)

Permite además la inscripción en las diferentes opciones del sistema de postgrado. Además de lo anterior, también se trabaja con la documentación de Ciencia y Técnica a través de proyectos, líneas de investigación, publicaciones, entre otros, de los profesores de un departamento docente. (29)

Sin embargo, este sistema se diseñó específicamente para gestionar información referente a los profesores de los departamentos que trabajan en la Universidad “Camilo Cienfuegos” de Matanzas, mientras que el sistema en desarrollo tiene dentro de su alcance gestionar información investigativa y postgraduada, no sólo de los profesores sino también de los estudiantes, todo se debe a que la Universidad de las Ciencias Informáticas (UCI) posee la característica de ser una universidad docente-productiva lo cual la hace diferente de otras universidades, esto significa que tanto los profesores como los estudiantes están vinculados a la investigación y a la producción de *software* en conjunto.

» **Sistema para la Gestión de Postgrados en la UCI**

Es un sistema desarrollado en la Universidad de las ciencias informáticas, que se encarga de todo lo referente a la matrícula por parte de los profesores a los diferentes cursos, entrenamientos, talleres y diplomados que se ofertan, así como, controlar sus evaluaciones. Permitiendo a la Dirección de formación postgraduada de la UCI, obtener todo el proceso de control de reportes, que es la respuesta a toda la información que necesite el Director de postgrado.

Como se puede apreciar este sistema solo se centra en las evaluaciones y la matrícula de los profesores a los diferentes cursos de postgrado ofertados. Por lo que no cuenta con las características definidas por la vicedecana de investigación y postgrado de la facultad 4.

» **Akademos**

Es un sistema web desarrollado en la Universidad de las Ciencias Informáticas (UCI). Brinda una interfaz común para todos sus usuarios. Además, ofrece la información actualizada en tiempo real de las facultades de la Universidad. Akademos realiza la gestión de toda la información referente a la formación de pregrado de un estudiante. Además, posee un módulo para profesores donde estos pueden controlar las evaluaciones de sus estudiantes y la asistencia de estos a clases.

» **SGSPG (Sistema de Gestión de Superación Postgraduada)**

Es una aplicación creada en la UCI, desarrollada y administrada mediante el framework Symfony que contiene una cuenta de suscripción al sistema, realiza la publicación de convocatorias de cursos y crea la matrícula a los mismos. Publica materiales de interés y presenta los siguientes módulos: admin, prof, post, act, becas, dipl, cursos y polo. Esta aplicación no se ha puesto en práctica y actualmente está fuera de uso.

De los sistemas anteriormente analizados ninguno cumple con las expectativas del cliente, ya que lo que este desea es un sistema que gestione la información referente a investigación y postgrado de forma específica y detallada de acuerdo con lo que realmente se utiliza en este Departamento, además, que tenga carácter informativo e interactivo.

1.2 Tendencias y tecnologías actuales

En las siguientes secciones se describen las principales tecnologías y herramientas que se utilizan en la actualidad para la construcción de aplicaciones web llegando a la conclusión de cuáles serían más factibles a utilizar para el desarrollo de la solución propuesta.

1.2.1 Metodologías de desarrollo de *software*

Una metodología para el desarrollo de un proceso de *software* es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que los desarrolladores de sistemas informáticos deben seguir para garantizar la calidad de un producto informático. En la actualidad existen diversas metodologías de desarrollo con particularidades específicas, entre las más conocidas están: RUP (Rational Unified Process) y XP (Extreme Programming).

La calidad en el desarrollo y mantenimiento de los sistemas informáticos se ha convertido en uno de los principales objetivos de las organizaciones, debido a que cada vez más los procesos principales dependen de sistemas informáticos para su buen funcionamiento. La metodología que va a guiar el proceso de desarrollo de un *software* es muy importante ya que esta permite definir los pasos a seguir durante todo el ciclo de desarrollo.

» Metodologías tradicionales

Las metodologías tradicionales se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. Sin lugar a dudas ellas imponen una disciplina de trabajo sobre el proceso de desarrollo del *software*, con el fin de conseguir un *software* más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto *software*.

» Proceso Unificado de Desarrollo (RUP)

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de *software* de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational *Software*, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, utilizando UML como lenguaje de modelado. Define cuatro fases esenciales (Inicio, Elaboración, Construcción y Transición) y nueve flujos de trabajos; seis de Ingeniería (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y tres de apoyo (Administración del proyecto, Administración de configuración y cambios y Ambiente). (2)

Presenta las cuatro fases (**Inicio**, **Elaboración**, **Construcción** y **Transición**) y dentro de cada una de ellas el equipo de desarrollo pasa por todos los **Flujos de Trabajo** que son transversales a las fases, inclusive en varias iteraciones. (2)

En la fase de **Inicio** se describe el negocio y se delimita el proyecto, describiendo el alcance con la identificación de los casos de uso del sistema. Durante la **Elaboración** se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. Mediante la fase de **Construcción** se logra obtener un producto listo para ser utilizado, el cual está documentado y cuenta con un manual de usuario. En la fase de **Transición** la versión final ya se

encuentra lista para su instalación en condiciones reales del entorno de trabajo, durante esta fase se pueden reparar algunos errores del producto. (2)

Flujos de trabajo: (2)

Modelación del negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: establece cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba: busca los defectos a lo largo del ciclo de vida.

Instalación: produce un entregable del producto y realiza actividades como empaque, instalación, asistencia a usuarios, entre otras, su propósito es entregar el *software* a los usuarios finales.

Administración del proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización o actualización concurrente de elementos, control de versiones, entre otro.

Ambiente: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como, el procedimiento para implementar el proceso en una organización.

Ventajas:

- » Evaluación en cada fase que permite cambios de objetivos.
- » Funciona bien en proyectos de innovación.

- » Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el *software*.
- » Seguimiento detallado en cada una de las fases.

Desventajas:

- » El cliente debe ser capaz de describir y entender a un gran nivel de detalle para que los desarrolladores puedan acordar un alcance del proyecto con él.
- » No es flexible para el desarrollo de proyectos pequeños que pueden estar sujetos a constantes cambios en sus requerimientos; debido a que un pequeño cambio en uno de ellos, provocaría que se tenga que generar nuevamente toda la documentación y prolongaría el tiempo de desarrollo del proyecto.

» Metodologías ágiles

En febrero de 2001, en una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de *software*. En esta reunión se creó The Agile Alliance 3, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de *software* y ayudar a las organizaciones para que adopten dichos conceptos. Las metodologías ágiles recogen técnicas, “buenas prácticas”, cada una tiene sus características propias y responden mejor a la realidad del *software*. (3)

» SCRUM

Scrum además de una metodología es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Por su naturaleza es iterativo y pretende tener un producto listo al final de cada iteración. Permite la creación de equipos auto-organizados y la comunicación verbal entre todos los miembros del proyecto.

Principios de SCRUM: (4)

- » Los equipos son auto-gestionados.
- » Una vez dimensionadas las tareas, no es posible agregarles trabajo extra.
- » Se realizan reuniones diarias en las que los miembros del equipo se plantean 3 cuestiones:

- » ¿Qué has hecho desde la última revisión?
- » ¿Qué obstáculos te impiden cumplir la meta?
- » ¿Qué vas a hacer antes de la próxima reunión?
- » Las iteraciones de desarrollo tienen una frecuencia inferior a un mes, al final de las cuales se presenta el resultado a los externos del equipo de desarrollo, y se realiza una planificación de la siguiente iteración, guiada por el cliente.

Ventajas:

- » Entrega de un producto funcional al finalizar cada Sprint¹.
- » Posibilidad de ajustar la funcionalidad sobre la base de la necesidad de negocio del cliente.
- » Visualización del proyecto día a día.
- » Alcance acotado y viable.
- » Equipos integrados y comprometidos con el proyecto, toda vez que ellos definieron el alcance y se auto-administran.

Desventajas:

- » Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.
- » Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas

» Crystal

Se trata de un conjunto de metodologías para el desarrollo de *software* caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de *software* se considera un juego cooperativo de invención y comunicación, limitado por los

¹ El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo en base a su propia experiencia.

recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). (5)

Los roles definidos por Crystal son:

- » Patrocinador Ejecutivo (Executive Sponsor)
- » Jefe de Proyecto (Project Manager)
- » Experto en el Dominio (DomainExpert)
- » Experto de uso (UsageExpert)
- » Programador Diseñador (Designer-Programmer)
- » Diseñador de interfaz de usuario (Designer UI)
- » Realizador de Pruebas (Tester)
- » Programador Técnico (Technical)

Ventajas:

- » Son apropiadas para entornos ligeros.
- » Al estar diseñada para el cambio experimenta reducción de costo.
- » Presenta una planificación más transparente para los clientes.
- » Se definen en cada iteración cuales son los objetivos de la siguiente.
- » Permite tener una muy útil realimentación de los usuarios.

Desventajas:

- » Delimita el alcance del proyecto con el cliente.

» eXtremeProgramming (XP)

Extreme Programming se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un excelente clima de trabajo. (5)

Es una de las metodologías ágiles más conocidas. Es utilizada en proyectos de corto plazo y con equipos pequeños. Plantea desarrollar una programación rápida o extrema. (6)

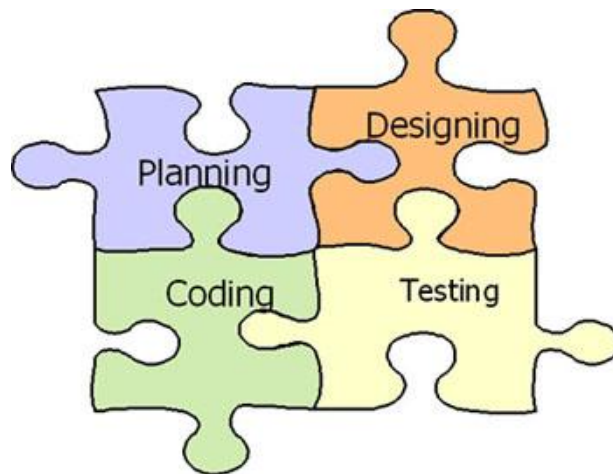


Ilustración 1: Metodología Extreme Programming

Esta metodología se basa en las pruebas realizadas a los principales procesos “Pruebas Unitarias”, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir. Es como adelantarse a los posibles errores. Además de la reutilización de código “Refabricación” para lo cual se crean patrones o modelos estándares que hacen más flexible el cambio, se basa también en la “Programación en pares” que es una particularidad de esta metodología la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa. (6)

Lo fundamental en este tipo de metodología es:

- » La comunicación, entre los usuarios y los desarrolladores.
- » La simplicidad, al desarrollar y codificar los módulos del sistema.
- » La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

» ¿Por qué XP?

Una vez finalizado el estudio realizado sobre las posibles metodologías de desarrollo de *software* que podrían haber guiado el proceso de desarrollo, se seleccionó por parte del equipo de trabajo la metodología ágil XP.

En primer lugar una de la característica fundamental de XP, es que el cliente forme parte del equipo de desarrollo, requisito que se cumple debido a la cercanía constante del cliente; por otra parte, el equipo de trabajo cuenta con dos integrantes, la dimensión del proyecto es pequeña y está basado en el uso de nuevas tecnologías, por lo que se suman los riesgos técnicos asociados a ellas lo cual es perfectamente posible gracias a su naturaleza iterativa e incremental. Por último, se consigue tener un proceso de desarrollo motivado, ya que XP no permite exceso de trabajo y se consigue una mayor integración entre los miembros por la comunicación establecida. Además, está probada en disímiles proyectos y tiene suficiente soporte y documentación accesible en la red. Descartando las anteriormente analizadas principalmente por las desventajas encontradas durante el análisis de cada una de ellas, debido a que se debían tener en cuenta, algunas necesidades por parte de los desarrolladores para la realización de la solución propuesta.

1.2.2 Herramientas de Modelado

Las herramientas CASE (Computer Aided *Software* Engineeringen) nacen para auxiliar a los desarrolladores de sistemas, lo que permite apoyo computarizado en todo o en parte del ciclo de vida de desarrollo de un sistema de *software*.

» Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* de

modelado UML ayuda a una construcción más rápida de aplicaciones con mayor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar la documentación. Además, proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (7)

Esta herramienta tiene las siguientes características: (7)

- » Soporte de UML versión 2.1.
- » Modelado colaborativo con CVS y Subversion (nueva característica).
- » 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.
- » Diagramas EJB - Visualización de sistemas EJB.
- » Generación de código – Desarrollo y despliegue de aplicaciones.
- » Diagramas de flujo de datos.
- » Soporte ORM - Generación de objetos Java desde la base de datos.
- » Generación de base de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- » Ingeniería inversa de base de datos - Desde Sistemas Gestores de Base de datos (DBMS) existentes a diagramas de Entidad-Relación.
- » Generador de informes para generación de documentación, Distribución automática de diagramas, Reorganización de las figuras y conectores de los diagramas UML.
- » Importación y exportación de ficheros XML.
- » Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.

» **Rational Rose Enterprise**

Esta herramienta permite una modelación absoluta de los procesos del negocio y del sistema, además resulta de gran utilidad para los desarrolladores de proyectos, pues ella cubre todo el ciclo de vida del proyecto desde su fase inicial hasta su culminación.

Rational Rose posibilita establecer una trazabilidad entre los modelos (análisis y diseño). Cada rol tiene su propia vista de arquitectura (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y funcionalidad del sistema en construcción.

Cada analista, desarrollador o diseñador puede usar Rational Rose para definir y comunicar el negocio, el diseño y la arquitectura de la aplicación que se esté desarrollando. Es una solución completa para mostrar de forma gráfica el análisis de los procesos del negocio y los requerimientos del sistema.

» **¿Por qué Visual Paradigm?**

Visual Paradigm para UML, además, de ser una herramienta que posee licencia gratuita y comercial, es una herramienta case que permite llevar una buena organización, planificación de los diagramas y modelados de la aplicación a desarrollar, también, el equipo de desarrollo presenta conocimientos básicos de la herramienta lo que posibilita un mejor desempeño con la misma.

1.2.3 Lenguajes de programación

Un lenguaje de programación se puede definir como un conjunto de sintaxis y reglas semánticas que definen los programas del computador. Permite a las personas escribir en un lenguaje que sea más apropiado a las características humanas y se pueda traducir al lenguaje de la máquina. Los lenguajes más conocidos son C#, C++, SQL, Java y PHP. Estos lenguajes son usados para comunicar instrucciones a una computadora y poder ejecutar un programa.

» **PHP Hypertext Processor (PHP)**

PHP es utilizado para la generación de páginas web dinámicas embebidas en páginas HTML y ejecutada en el servidor, es un lenguaje de script interpretado en el lado del servidor. Para su funcionamiento

necesita tener instalado Apache o Internet Information Server (IIS) con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Este lenguaje brinda una serie de ventajas las cuales se citan a continuación: (8)

- » Las aplicaciones desarrolladas con este lenguaje se caracterizan por tener un tiempo de ejecución corto.
- » Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros.
- » Capacidad de expandir su potencial utilizando módulos.
- » Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- » Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- » Incluye gran cantidad de funciones.
- » No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

» **Active Server Pages (ASP)**

Es una tecnología del lado del servidor desarrollada por Microsoft para el desarrollo de sitios web dinámicos. ASP fue liberado por Microsoft en 1996. No necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP, el lenguaje más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. (8)

Ventajas:

- » Usa Visual Basic Script, siendo fácil para los usuarios.
- » Comunicación óptima con SQL Server.
- » Soporta el lenguaje JavaScript de Microsoft.

Desventajas:

- » Código desorganizado.
- » Se necesita escribir mucho código para realizar funciones sencillas.
- » Tecnología propietaria.

» Hospedaje de sitios web costosos.

» Python

Es un lenguaje de programación creado en el año 1990 por Rossum. Es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web. Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma², lo cual fuerza a que los programadores adopten un estilo de programación particular.

Ventajas: (8)

- » Lenguaje de propósito general.
- » Gran cantidad de funciones y librerías.
- » Multiplataforma.
- » Licencia de código abierto (Open source).
- » Orientado a Objetos.
- » Portable.

Desventaja:

- » Lentitud por ser un lenguaje interpretado.

» ¿Por qué PHP?

Después de analizar los distintos lenguajes de programación web se puede apreciar que todos son recomendables para desarrollar aplicaciones web con una alta calidad. Se escogió PHP para llevar a cabo el desarrollo de esta, principalmente por sus ventajas ya que es muy fácil de aprender, se caracteriza por

² Un **lenguaje de programación multiparadigma** es el cual soporta más de un paradigma de programación. Según lo describe Bjarne Stroustrup, permiten crear “programas usando más de un estilo de programación”.

ser muy rápido, es multiplataforma y tiene capacidad de conexión con la mayoría de los manejadores de base de datos. También se puede decir que tiene la posibilidad de expandir su potencial mediante módulos y además presenta una comunidad bastante activa.

1.2.4 Servidores web

Un servidor web es conocido como programa que atiende y responde a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). (9)

» Apache

Apache, es el servidor web por excelencia. Presenta características que hacen que cada vez millones de desarrolladores aumenten su confianza en este programa, destacándose: su configuración, robustez y estabilidad. Es una tecnología gratuita, de código fuente abierto, que posee una licencia descendiente de las licencias BSD (Distribución de *Software* Berkeley), la cual permite hacer lo que se quiera con el código fuente. A continuación se enuncian algunas características de este servidor: (10)

- » Corre sobre todos los Sistemas Operativos (SO).
- » Tiene integración en estándar del protocolo de seguridad SSL (Secure Sockets Layer).
- » Provee interfaz a todas las bases de datos.
- » Es muy configurable en la creación y gestión de log, ya que permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.
- » Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- » Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades de este servidor, ya que se pueden escribir módulos para realizar determinadas funciones, lo que implica que haya gran cantidad de ellos disponibles para su utilización.

- » Apache posee varios módulos, los cuales pueden clasificarse en tres categorías:
 - »Módulos Base: Módulo con las funciones básicas del Apache.
 - »Módulos Multiproceso (para manejar las peticiones): 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. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y la rapidez del código.
 - »Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor. Simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el *software*.

» Internet Information Server (IIS)

Este servidor web permite que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones web y a la vez la reducción de sus costes administrativos. Es reconocido como un potente servidor web que brinda una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web sobre todas las versiones de Windows Server 2003 sin dejar de mencionar que IIS 0.6 soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI) con monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas. (11)

IIS engloba un conjunto de herramientas destinadas al control de servicios de Internet como el Web, FTP, correo y servidores de noticias. Además, incluye el soporte necesario para la creación de páginas dinámicas en el servidor mediante el lenguaje ASP.

Las características en cuanto a seguridad se aprovechan de las últimas tecnologías de cifrado y métodos de autenticación mediante certificados de cliente y servidor. Una de las formas que tiene IIS de asegurar los datos es mediante SSL. Esto proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario.

» ¿Por qué Apache?

El estudio de los servidores web más usados en la actualidad ha demostrado que Apache es la mejor opción y la que más se adecua a las necesidades del sistema a implementar. Además de las posibilidades que brinda para que el usuario pueda ampliar sus capacidades, este servidor es compatible con todos los SO, su carácter modular permite que pueda ser adaptado a diferentes entornos y necesidades con los diferentes módulos de apoyo que proporciona, presenta soporte para el trabajo con archivos grandes (más de 2 GB) en plataformas de 32 bits, es de código abierto, extensible y muy popular.

1.2.5 Sistema Gestor de Base de Datos (SGBD)

Un Sistema Gestor de Base de datos (SGBD) o Data Base Management System (DBMS) puede ser clasificado como una colección de programas. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Se compone de un lenguaje de definición de datos, manipulación de datos y de consulta teniendo como propósito servir de interfaz entre la base de datos, el usuario y las aplicaciones. (12)

Entre los diferentes tipos de gestores de base datos existentes se pueden encontrar Oracle, PostgreSQL, MySQL, MS SQL Server, por mencionar algunos. Dado que muchos de estos gestores son propietarios y su costo es elevado en el mercado, una alternativa libre sería el uso de gestores como MySQL y PostgreSQL que en los últimos años han alcanzado un nivel de madurez equivalente a los proporcionados por los sistemas propietarios en cuanto a seguridad y eficiencia.

» PostgreSQL

PostgreSQL es un sistema de gestión de base de datos objeto-relacional (ORDBMS), es una derivación libre (OpenSource) de este proyecto. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. (13)

A continuación se enumeran las principales características de este gestor de base de datos: (14)

- » Incorpora una estructura de datos array.

- » Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de base de datos se le incluye entre los gestores objeto-relacionales.
- » Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- » Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- » Es altamente confiable en cuanto a estabilidad se refiere.

Tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa. A pesar de ello, el primer encuentro con este gestor es un poco "duro", ya que la sintaxis de algunos de sus comandos no es nada intuitiva. También resulta engorroso las pequeñas variaciones que presenta este gestor en alguno de los tipos de datos que maneja, siendo el problema más comentado el referente al tipo "serial".

» MySQL

Este gestor de base 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 librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. MySQL es un sistema de gestión de base de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. (13)

Las principales características de este gestor de base de datos son las siguientes: (13)

- » Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- » Soporta gran cantidad de tipos de datos para las columnas.
- » Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.)
- » Gran portabilidad entre sistemas.
- » Soporta hasta 32 índices por tabla.
- » Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

MySQL Community Edition es una versión de descarga gratuita de la base de datos más popular de código abierto que posee una comunidad activa de desarrolladores. Esta versión de MySQL de código abierto está bajo la licencia GPL. (30)

» ¿Por qué MySQL?

El Sistema Gestor de Base de Datos seleccionado es MySQL Community Edition el cual supera desde hace tiempo a muchas soluciones comerciales en velocidad, dispone de un sistema de permisos elegante y potente y usa el lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información. MySQL Community Edition es muy rápido y capaz de almacenar grandes cantidades de datos. Muestra una interfaz amena y los comandos para gestionar la base de datos son más intuitivos que los de PostgreSQL, siendo muchos de ellos sentencias SQL. Por otro lado, la API de PHP para acceder a MySQL Community Edition es más sencilla de usar, teniendo un estilo mucho más natural y por último está bajo la licencia GPL.

1.2.6 Sistemas de Gestión de Contenidos (CMS)

El Sistema Gestor de Contenido es una herramienta que permite crear y mantener una Web con facilidades, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las Web. Es el grupo de procesos y tecnologías que apoyan el ciclo de vida evolutivo de información digital. Esta información digital a menudo es llamada como contenido digital. El contenido digital puede tomar la forma de texto, como documentos, archivos multimedia, archivos de audio de video o cualquier otro tipo de archivo que siga un ciclo de vida de contenido, que requiere de dirección.

» Drupal

Drupal es un Sistema Gestor de Contenidos modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web y hace un énfasis especial en la usabilidad y consistencia de todo el sistema.

Su flexibilidad y adaptabilidad, así como, la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitio web. (15)

Drupal tiene un buen diseño y un código de muy buena calidad, que lo hace fácilmente extensible. Su usabilidad y accesibilidad son excelentes en comparación con otros CMS. Es abierto a PHP permitiendo que en cualquier tipo de contenido se pueda agregar código PHP en dependencia de las necesidades, además, genera un código HTML extremadamente limpio y bien estructurado. Permite tener aplicaciones multilinguaje, así como, hacer cualquier diseño deseado.

Tiene un control total para los administradores, desde permisos para usuario y módulo hasta el manejo del panel de control.

Por último, se debe señalar que cuenta con un soporte de comunidad muy amplio, donde sus usuarios colaboran en un fin común y esa sea quizás la principal ventaja de Drupal frente a Joomla, pues Drupal tiene una comunidad muy buena, Drupal.org, que es un punto de encuentro entre sus seguidores donde cualquier duda está documentada y explicada. (15)

» Joomla

Joomla es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenido se utiliza para publicar en Internet e Intranets utilizando una base de datos MySQL. En Joomla se incluyen características como: versiones imprimibles de páginas, flash con noticias, blogs, foros, encuestas, calendarios, búsquedas en el sitio web e internacionalización del lenguaje.

Características:

- » Organización del sitio web.
- » Publicación de contenidos.
- » Escalabilidad e implementación de nuevas funcionalidades.
- » Administración de usuario.
- » Diseño y aspecto estético del sitio.
- » Disposición de módulos modificable.
- » Encuestas.

» Estadísticas de visitas.

» ¿Por qué Drupal?

Teniendo en cuenta las características de los CMS descritos anteriormente, se ha decidido utilizar el CMS Drupal como plataforma de publicación de la aplicación propuesta. Drupal brinda un gran número de facilidades, que lo hacen adecuado para realizar muchos tipos de sitios web. Entre esas facilidades se puede destacar su rápido y bajo consumo del servidor. Además, ofrece muchos recursos y módulos incluidos muy sencillos de administrar.

Por tanto, se decide utilizar el CMS Drupal para el desarrollo de la aplicación web ya que se necesita alto rendimiento, escalabilidad³, buen gestor de contenido y una flexibilidad de diseño significativa.

1.2.7 Entorno de Desarrollo Integrado (IDEs)

Un Entorno de Desarrollo Integrado (IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación. Consiste en un editor de código, compilador, depurador y un constructor de Interfaz Gráfica (GUI), estos pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación y es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

» Zend Studio

Zend Studio es un editor de texto esencialmente para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. Es un programa orientado a desarrollar aplicaciones web en lenguaje PHP. Este IDE consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Zend Studio es un completo Entorno de Desarrollo Integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. (16)

³ Capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

» PHP Designer

PHP Designer es una potente herramienta para aquellos programadores que trabajen con PHP. También permite trabajar en otros lenguajes de programación como HTML, XHTML, CSS y SQL. PHP Designer permite ahorrar tiempo gracias a sus múltiples herramientas y utilidades. El entorno de desarrollo de esta herramienta es bastante funcional, incluye los scripts y códigos prediseñados más utilizados, además de disponer de la función de autocompletar código. Ofrece una serie de asistentes y diálogos integrados, además permite el acceso directo a librerías de código o scripts de uso habitual y utilidades diversas, todo ello en una interfaz de diseño sencillo y elegante que puedes personalizar con nada menos que dieciocho temas distintos. (17)

» NetBeans

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. La plataforma NetBeans es un IDE de código abierto escrito completamente en Java. Tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS y en su versión 6.8 tiene soporte para el framework Symfony. Es un producto libre y gratuito sin restricciones de uso. Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados módulos. Además, es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. (18)

» ¿Por qué NetBeans?

El IDE seleccionado es el NetBeans principalmente por ser un producto libre, con todos los beneficios del *software* disponible en forma gratuita. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados módulos los cuales dan la posibilidad de ser extendidas agregándole nuevos módulos facilitando y reduciendo el tiempo de desarrollo de la aplicación.

1.2.8 Arquitectura

Arquitectura del *Software* es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- » Definir los módulos principales.
- » Definir las responsabilidades que tendrá cada uno de estos módulos.
- » Definir la interacción que existirá entre dichos módulos.
- » Control y flujo de datos.
- » Secuenciación de la información.
- » Protocolos de interacción y comunicación.
- » Ubicación en el hardware.

La Arquitectura del *Software* aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. La definición oficial de Arquitectura del *Software* es la IEEE Std 1471-2000 que reza así: “La Arquitectura del *Software* es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (19)

» **Arquitectura en tres capas**

La Arquitectura en tres capas es un estilo de programación, se basa en la división en el nivel de acceso a datos, nivel de lógica de negocio y nivel de presentación o aplicación.

Capas de la arquitectura:

Capa de presentación.- Esta capa es la que ve el usuario, le presenta el sistema, le comunica la información y captura la información minimizando el uso de procesos. Esta capa se comunica únicamente con la capa de un negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario. Generalmente se presentan como formularios. (20)

Capa de negocio.- Aquí es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos, para almacenar o recuperar datos de él.

Capa de datos.- Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (20)

» Patrón de arquitectura Modelo Vista Controlador (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de *software*. Los patrones expresan el esquema fundamental de organización para sistemas de *software*. Su concepto se basa en separar el modelo de datos de la aplicación de su representación ante el usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales: (21)

- » Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- » Vista (View): Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- » Controlador (Controller): Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests") para el modelo o la vista. (21)

» ¿Por qué Modelo Vista Controlador (MVC)?

Después de un estudio se determinó que la arquitectura a utilizar es Modelo Vista Controlador (MVC) el cual hace más factible el desarrollo de la aplicación teniendo como ventaja una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado. Además, hay una API muy bien definido; cualquiera que use la API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad y sin dejar de mencionar que la conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Conclusiones parciales

En este capítulo se realizó un estudio de las tecnologías y tendencias actuales en cuanto a desarrollo de Aplicaciones web y teniendo en cuenta que el país está inmerso en el proceso de migración a *software* libre, las herramientas seleccionadas se identifican por ser libres y en su mayoría multiplataforma. De ahí que se decidió seleccionar para el desarrollo y construcción del sistema, como lenguaje del lado del servidor, PHP, ya que se tiene un poco más de experiencia en su uso comparado con el resto de los mencionados en el capítulo. Como servidor web se escogió Apache destacándose por ser personalizable, configurable y flexible. Por último, MySQL en su versión 5.0 Community Edition fue seleccionado como gestor de base de datos ya que cumple con las necesidades planteadas.

La metodología de desarrollo del *software* Extreme Programming (XP) es la indicada, debido a que se adapta perfectamente al equipo de desarrollo. Visual Paradigm se utiliza para el modelado del sistema. Por las ventajas que brinda el sistema de gestión de contenidos Drupal, fue seleccionado como plataforma de publicación para la creación de la aplicación y el patrón Modelo Vista Controlador (MVC) para la estructuración y organización de la aplicación.

Capítulo 2. Construcción y prueba de la solución

2.1 Introducción

El presente capítulo se centra en la construcción de la “Aplicación para la gestión de información de investigación y postgrado de la Facultad 4”, se tomó como punto de partida todos los artefactos y características definidos por el analista. En este capítulo se fue en todo momento consistente con el diseño, se construyó el sistema en términos de componentes: ejecutables, ficheros de código fuente, scripts, entre otros. Además, se determinaron y se describieron las distintas pruebas de las funcionalidades para lograr la calidad requerida para la aplicación.

2.2 Propuesta de solución

Como propuesta de solución que fundamenta esta investigación se planteó la creación de un sistema que permita concentrar toda la información referente a Investigación y Postgrado de la Facultad 4, el mismo será una Aplicación web, administrada por el CMS Drupal que tenga los módulos Investigación, Postgrado y Balance, que permitirán la gestión y un manejo adecuado de toda la información referente a estos procesos.

2.2.1 Requerimientos no funcionales

Los requerimientos no funcionales se refieren directamente a las características que este debe tener. A continuación, damos una descripción breve y concisa de las propiedades con que debe contar esta Aplicación web. (24)

El diseño de la interfaz visual no debe tener animaciones ni imágenes pesadas, que repercuta en la rapidez de la aplicación. Debe ser amigable para los usuarios que interactúan con la aplicación, permitiendo un fácil entendimiento de las funcionalidades que el mismo brinda. Debe ser diseñada de forma tal, que los usuarios que harán uso de la misma obtengan los conocimientos necesarios en el menor tiempo posible para una mejor explotación de sus funcionalidades.

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Para un mejor funcionamiento de los servicios que brinda este sistema gestor de información, se debe nutrir con un Sistema Operativo Windows 98 o superior, o Linux y haciendo uso de un navegador web como; Mozilla Firefox (versión 3.5.0 o superior) o Internet Explorer (versión 7.0 o superior). Esto permitirá una mayor rapidez y eficiencia tanto en los tiempos de respuesta como en la velocidad de procesamiento y de navegación a nivel de aplicación. También va a proporcionar tiempos de respuesta entre 5 y 10 segundos para los procesos en línea del sistema. (24)

2.2.2 Historias de usuario (HU)

Las historias de usuario describen las funcionalidades del *software* y son escritas por el cliente empleando un lenguaje claro. Estas permiten responder rápidamente a los requerimientos cambiantes y son la base para las pruebas funcionales del sistema. Además, se emplean para hacer estimaciones de tiempo y para realizar el plan de entregas. A continuación se listan todas las HU realizadas por el cliente durante la fase de exploración del proyecto, que abarcan las funcionalidades del sistema propuesto y una breve descripción de cada una de ellas. (24)

Tabla 1: HU - Gestionar usuario

Número: 1	Nombre: Gestionar Usuario	Usuario: administrador
Descripción: Permite administrar las cuentas de usuarios en el sistema, posibilitando incluir, modificar, eliminar y visualizar los datos de las mismas. A cada usuario se le asignará el rol correspondiente a sus responsabilidades dentro de la aplicación.		

(Para un mejor entendimiento de esta HU. Ver anexo 1)

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Tabla 2: HU - Autenticar usuario

Número: 2	Nombre: Autenticar Usuario	Usuario: administrador, asesor por departamento, profesor, estudiantes.
Descripción: El personal perteneciente a la facultad, debe introducir su cuenta de usuario y contraseña para autenticarse, según sus privilegios accederá a la sesión correspondiente de acuerdo con su rol en la aplicación.		

(Para un mejor entendimiento de esta HU. Ver anexo 2)

Tabla 3: HU - Gestionar foro

Número: 3	Nombre: Gestionar foros	Usuario: administrador
Descripción: Realizar las acciones de incluir, modificar, eliminar y visualizar los temas y comentarios en la aplicación. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 3)

Tabla 4: HU - Gestionar artículos

Número: 4	Nombre: Gestionar artículos	Usuario: administrador
Descripción: Realizar acciones de incluir, modificar, eliminar y listar artículos. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 4)

Tabla 5: HU - Gestionar documentos

Número: 5	Nombre: Gestionar documentos	Usuario: administrador
Descripción: Realizar las acciones de incluir, modificar, eliminar y listar documentos de diferentes tipos ya sean útiles, de consultas o trabajos realizados. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 5)

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Tabla 6: HU - Gestionar cursos

Número: 6	Nombre: Gestionar cursos	Usuario: administrador
Descripción: Realizar las acciones de incluir, modificar, eliminar y listar los cursos. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 6)

Tabla 7: HU - Gestionar eventos

Número: 7	Nombre: Gestionar eventos	Usuario: administrador
Descripción: Realizar las acciones de incluir, modificar, eliminar y visualizar los eventos. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 7)

Tabla 8: HU - Gestionar solicitud de eventos

Número: 8	Nombre: Gestionar solicitud de eventos	Usuario: administrador, asesor por departamento, profesor
Descripción: Realizar las acciones de incluir, modificar, eliminar y listar las revisiones de trabajos por usuarios involucrados. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 8)

Tabla 9: HU - Gestionar revisión de trabajos

Número: 9	Nombre: Gestionar revisión de trabajos	Usuario: administrador
Descripción: Realizar las acciones de inclusión, modificación, eliminación y listado de las revisiones de trabajos por usuarios involucrados. Todos los usuarios pueden ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 9)

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Tabla 10: HU - Revisar trabajos

Número: 10	Nombre: Revisar trabajos	Usuario: administrador, asesor por departamento, profesor
Descripción: Realizar la acción de revisar la documentación de los trabajos asignados.		

(Para un mejor entendimiento de esta HU. Ver anexo 10)

Tabla 11: HU - Gestionar actividad postgraduada

Número: 11	Nombre: Gestionar actividad postgraduada	Usuario: administrador
Descripción: Realizar las acciones de incluir, modificar, eliminar y visualizar toda la actividad de superación postgraduada de los profesores.		

(Para un mejor entendimiento de esta HU. Ver anexo 11)

Tabla 12: HU - Buscar actividad postgraduada

Número: 12	Nombre: Buscar actividad postgraduada	Usuario: administrador
Descripción: Realizar la acción de búsqueda. La búsqueda se realiza según el/ los parámetro(s) introducido(s) siempre y cuando exista como indicador. Todos los usuarios (excepto el usuario estudiante) podrán ver la información publicada.		

(Para un mejor entendimiento de esta HU. Ver anexo 12)

Tabla 13: HU - Generar reportes

Número: 13	Nombre: Generar reportes	Usuario: administrador
Descripción: Mostrar reportes generales generados a partir de las planillas y de la información obtenida tanto de investigación como de la superación postgraduada.		

(Para un mejor entendimiento de esta HU. Ver anexo 13)

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

2.2.3 Historias de usuario distribuidas por módulos

Para un mejor entendimiento de las funcionalidades, estas se unificaron según los temas analizados en módulos. Esto permite un mayor entendimiento de las mismas.

Tabla 14: Historias de usuarios distribuidas por módulos

Módulos	Historias de usuario
Investigación	Gestionar eventos Gestionar solicitud de eventos Gestionar revisión de trabajos Revisar trabajos
Postgrado	Gestionar actividad postgraduada Buscar actividad postgraduada
Balance	Generar reportes

2.3 Arquitectura de Drupal

El secreto de Drupal para conseguir su reconocida flexibilidad y facilidad en la creación de sitios web es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (nodes), módulos (modules), bloques y menús (blocks & menus), permisos de usuario (user permissions) y plantillas (templates). (25)

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

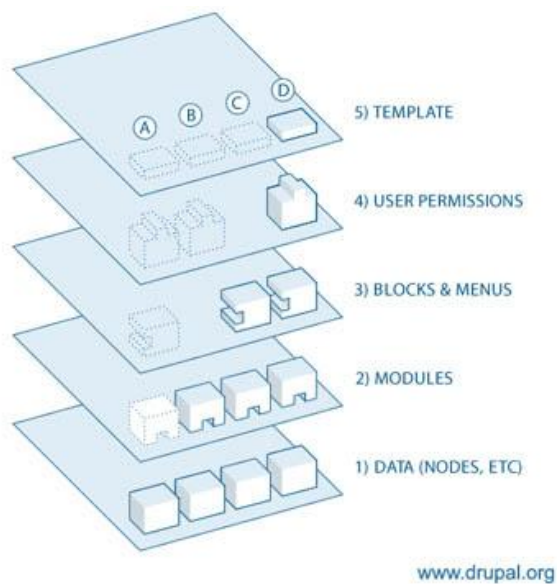


Ilustración 2: Arquitectura de Drupal

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

2.4 Diseño de la Base de Datos

El diseño de la base de datos modela el tratamiento de la información con carácter persistente dentro del sistema. Los servicios que se implementaron están desarrollados con Drupal, y para el almacenamiento de la información de los mismos se utilizó la base de datos de este CMS, agregándose 2 tablas nuevas investigación y postgrado.

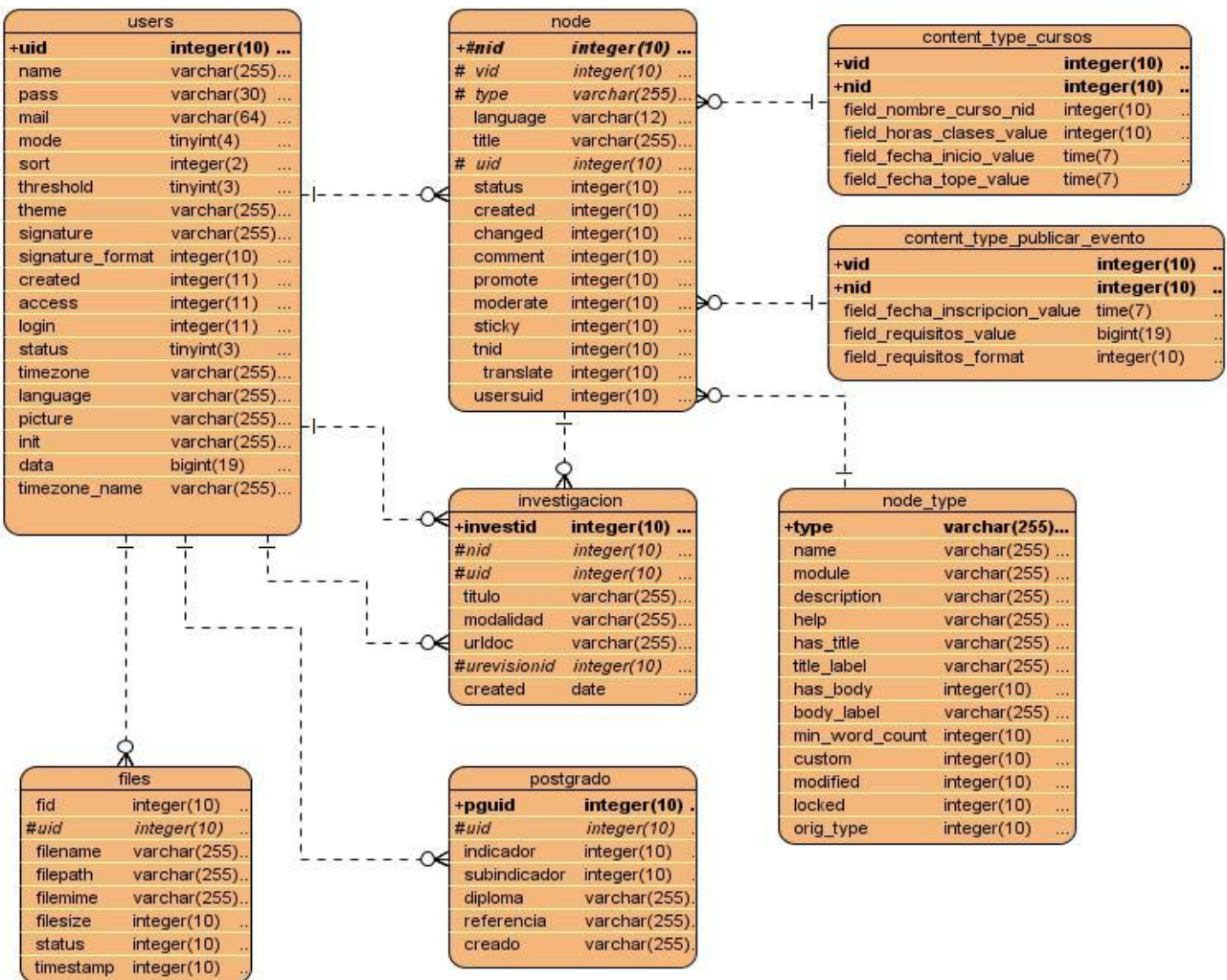


Ilustración 3: Modelo de datos del sistema

2.5 Fase de Implementación

En esta fase se realizó la implementación de las HU que fueron seleccionadas por cada iteración. Antes de iniciar en esta fase el equipo de trabajo del proyecto en reunión técnica, se analizaron y discutieron las diferentes interacciones de esta fase, distribuyéndose las tareas a realizar y adecuación de la planificación inicial realizada, se tuvo en cuenta para la misma el grado de madurez y desempeño de los integrantes del grupo de trabajo que facilitó la organización y cumplimiento de esta fase. Como parte de este plan se crean tareas de programación para ayudar a organizar la implementación exitosa de las HU.

2.5.1 Plan de iteraciones

En cada iteración se implementaron las HU previamente definidas. Como se explicó en el párrafo inicial de este acápite para cada iteración planificada se realizó encuentro técnico para su revisión, replanificación y/o reajustes de la misma. En este plan se desglosaron las HU en tareas de desarrollo, que realizaron los programadores para llevar a cabo la construcción del sistema. Estas tareas fueron para el uso estricto de los programadores, por lo que fueron escritas en lenguaje técnico para su comprensión y fácil entendimiento. (24)

2.5.1.1 Iteración 1

En esta iteración para la construcción del *software* se implementaron las HU que conforman la estructura básica del sistema. A partir de ellas se realizaron las demás funcionalidades requeridas por la aplicación.

Tabla 15: Tiempo real de implementación por módulos. Iteración 1

Módulos	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
	Gestionar solicitud de eventos	2	2

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Investigación	Gestionar revisión de trabajos	2	2
	Revisar trabajos	1	2
Postgrado	Gestionar actividad postgraduada	2	1
	Buscar actividad postgraduada	2	2

A continuación se muestran las tareas efectuadas para los módulos implementados en esta iteración:

Tabla 16: Tarea 1 - Desarrollar CRUD de solicitud de eventos - Módulo investigación

Tarea	
Número de la tarea: 1	Número Historia (Nro.8): Gestionar solicitud de eventos
Nombre de la tarea: Desarrollar CRUD de solicitud de eventos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Emilio Álvarez Muñoz	
Descripción: Implementar las funcionalidades de listar, incluir, modificar y eliminar los datos de las solicitudes a eventos.	

Tabla 17: Tarea 2 - Desarrollar CRUD de revisión de trabajos - Módulo investigación

Tarea	
Número de la tarea: 2	Número Historia (Nro.9): Gestionar revisión de trabajos
Nombre de la tarea: Desarrollar CRUD de revisión de trabajos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Emilio Álvarez Muñoz	
Descripción: Implementar las funcionalidades de listar, incluir, modificar y eliminar los datos para las revisiones de trabajos.	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Tabla 18: Tarea 3 - Permitir la revisión de trabajos - Módulo investigación

Tarea	
Número de la tarea: 3	Número Historia (Nro.10): Revisar trabajos
Nombre de la tarea: Permitir la revisión de trabajos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Emilio Álvarez Muñoz	
Descripción: Implementar la funcionalidad que permita a los usuarios revisar los trabajos presentados a partir de las solicitudes de eventos.	

Tabla 19: Tarea 4 - Desarrollar CRUD de actividad postgraduada - Módulo postgrado

Tarea	
Número de la tarea: 4	Número Historia (Nro.11): Gestionar actividad postgraduada.
Nombre de la tarea: Desarrollar CRUD de actividad postgraduada	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Emilio Álvarez Muñoz	
Descripción: Implementar las funcionalidades de listar, adicionar, modificar y eliminar los datos de la actividad postgraduada por usuario.	

Tabla 20: Tarea 5 - Buscar actividad postgraduada - Módulo postgrado

Tarea	
Número de la tarea: 5	Número Historia (Nro.12): Buscar actividad postgraduada
Nombre de la tarea: Buscar actividad postgraduada	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Emilio Álvarez Muñoz	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Descripción: Crear filtros de búsqueda para las informaciones postgraduadas almacenadas, teniendo como base los campos: indicador y nombre usuario.

2.5.1.2 Iteración 2

Esta iteración estuvo centrada en desarrollar las HU con prioridad alta de vital importancia para el funcionamiento de la aplicación y en su mayoría poseían un nivel medio de riesgo para el programador.

Tabla 21: Tiempo real de implementación por módulo. Iteración 2

Módulos	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Balance	Generar reportes	2	3

A continuación se muestran las tareas efectuadas para los módulos implementados en esta iteración:

Tabla 22: Tarea 1 - Generar reportes para el balance - Módulo balance

Tarea	
Número de la tarea: 1	Número Historia (Nro.13): Generar reportes
Nombre de la tarea: Generar reportes para el balance	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Programador responsable: Emilio Álvarez Muñoz	
Descripción: Se genera reportes a partir de la información almacenada de información postgraduada en formato pdf.	

2.5.2 Diagrama de componentes

Un diagrama de componentes muestra cómo se encuentra dividido un *software* y las relaciones entre los componentes. Los diagramas de componentes presentan un nivel de abstracción más elevado que los diagramas de clases, aunque los componentes usualmente se encuentran implementados por clases.

Se utilizan para modelar las diferentes vistas de un sistema (estática y dinámica) y para mostrar que componentes pueden ser utilizados en común por varios sistemas o entre las diferentes partes de este.

A continuación se muestran los diagramas de componentes de las HU más críticos que se implementaron, los demás diagramas de componentes se incorporan en la sección Anexos del presente trabajo.

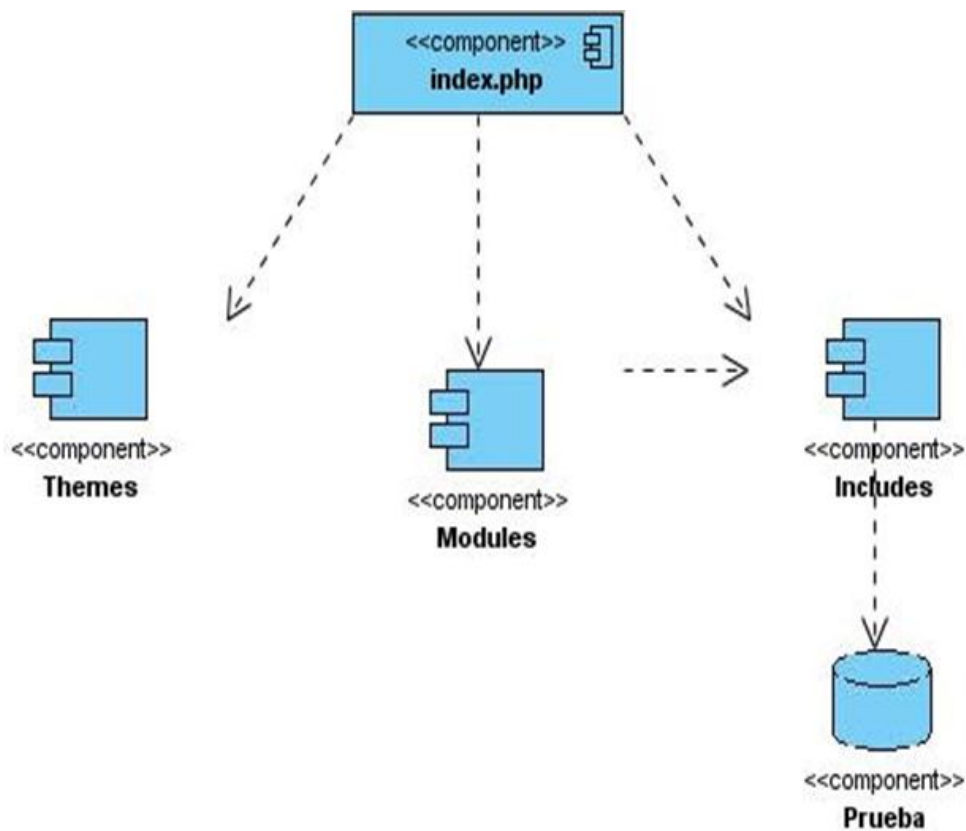


Ilustración 4: Diagrama de componentes del CMS Drupal

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

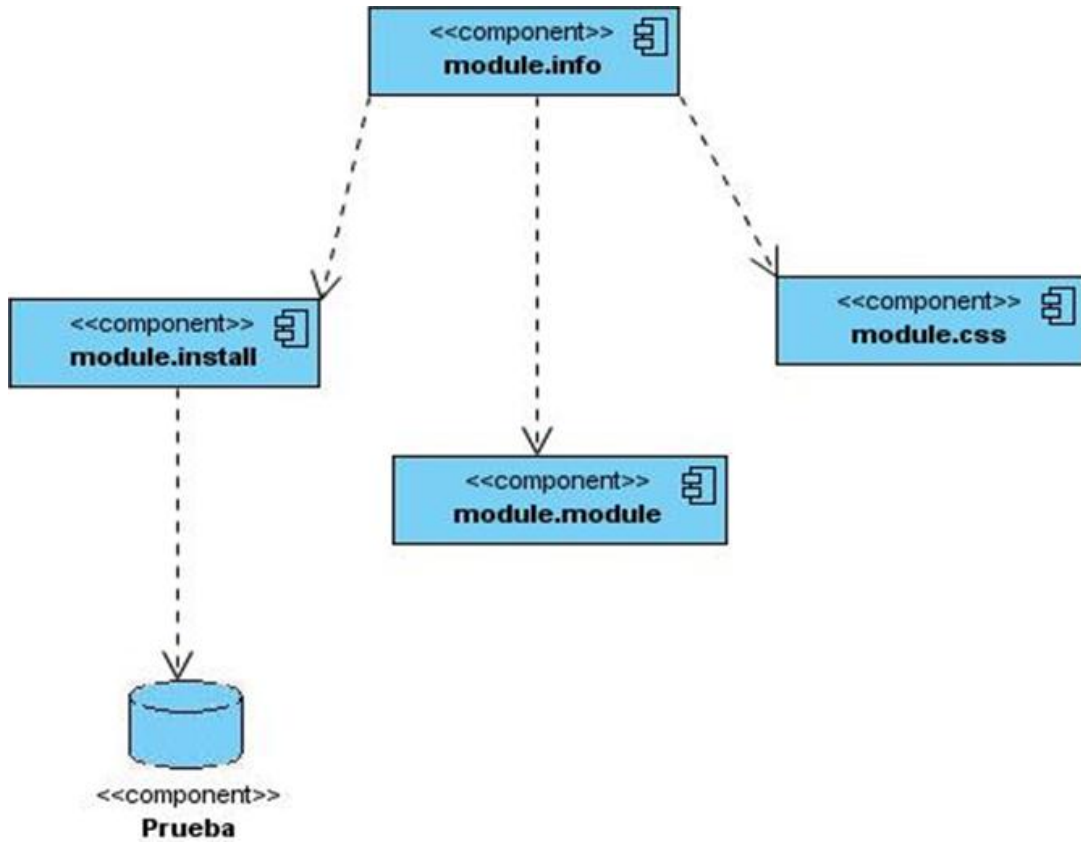


Ilustración 5: Diagrama de componentes de los módulos en Drupal

2.5.3 Diagrama de Despliegue.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos, siendo esta una unidad de computación de algún tipo (computadora, servidor). En este caso la aplicación se encuentra hospedada en un servidor Web (apache) y maneja los datos persistentes sobre un sistema gestor de base de datos (MySQL).

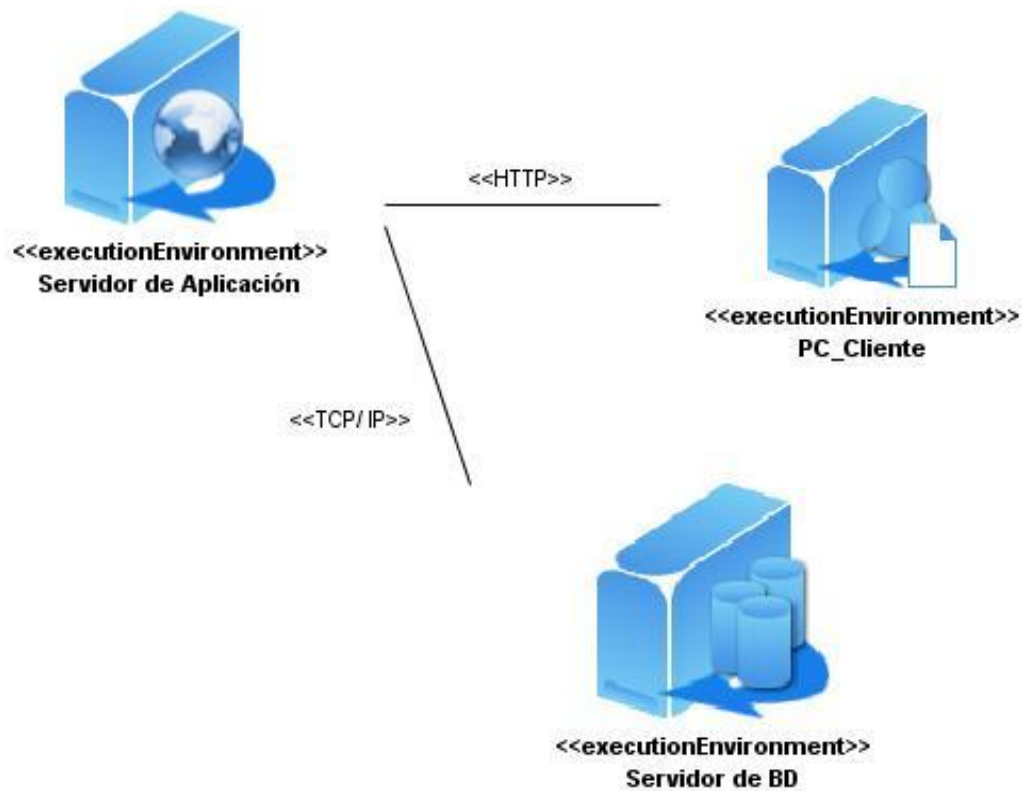


Ilustración 6: Diagrama de despliegue

Conclusiones parciales

Durante el desarrollo de este epígrafe se realizó el Modelo de Despliegue (ver Ilustración 7) a través del cual se muestra la distribución física del sistema y sus conexiones. Mediante la realización del Modelo de Datos (ver Ilustración 8) se logró describir los elementos que intervienen en el problema dado y la forma en que se relacionan estos elementos entre sí. Finalmente, se expuso el Modelo de Implementación, donde se especificaron los distintos componentes creados para desarrollar la aplicación y la relación entre ellos, lográndose traducir el diseño en términos de componentes ejecutables. De esta forma, se cuenta con la base principal en la cual apoyar la construcción del sistema.

2.6 Fase de prueba

En cualquier obra que emprenda el ser humano siempre se debe tener en cuenta que es posible la introducción accidental de fallos en los objetivos que se quieren alcanzar. En el desarrollo del *software* la detección y corrección de estos errores es vital. Este es un proceso que tiene lugar durante todo el desarrollo del producto y en todas sus etapas de construcción.

Una vez que se termina el producto, se realizan pruebas sistemáticas, para buscar fallos usando un criterio determinado. La prueba corresponde a una actividad en la cual un componente o el sistema en general son probados ejecutándolos en ciertas condiciones, y sobre la base de los requerimientos especificados para la construcción del mismo, los resultados son observados y registrados para hacer una evaluación y corregir lo que falla o mejorar lo que ya es funcional.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

2.6.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crearon a partir de las HU. Durante las iteraciones las HU seleccionadas fueron traducidas a pruebas de aceptación. En ellas se especificaron, desde la perspectiva del cliente, los escenarios para probar que una HU fue implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos fueron cumplidos y que el sistema es aceptable. Una HU no fue considerada completa hasta que no pasó por sus pruebas de aceptación. (23)

Tabla 23: CP01 - Gestionar solicitar evento

Caso de Prueba de Aceptación	
Código Caso de Prueba: 08PA01	Nombre Historia de Usuario: Gestionar solicitar evento
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Insertar los datos de una solicitud de eventos, inicialmente introducir incorrectamente la información para verificar las validaciones del sistema, luego de forma correcta para comprobar que los datos sean almacenados.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: Seleccionar un evento que supero su etapa de inscripción. Insertar el campo "Título del trabajo" en blanco. Insertar los datos correctos. Verificar que el nuevo producto aparece en el listado.	
Resultado Esperado:	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

<p>El sistema solo debe mostrar los eventos que aun están fecha de solicitud.</p> <p>El sistema debe alertar al usuario cuando el campo “Título del trabajo” este en blanco. <i>“El campo título del trabajo es obligatorio.”</i></p> <p>El sistema debe almacenar los datos y mostrarle al usuario un mensaje de solicitud enviada correctamente.</p> <p>El sistema debe mostrar los datos en el “Listar solicitudes de eventos”.</p>
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 24: CP02 - Gestionar solicitar evento

Caso de Prueba de Aceptación	
Código Caso de Prueba: 08PA02	Nombre Historia de Usuario: Gestionar solicitar evento
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Modificar los datos de una solicitud de eventos seleccionada previamente por el usuario. Inicialmente se modificarán incorrectamente los datos para verificar las validaciones del sistema, luego de forma correcta para comprobar que los datos se almacenan correctamente.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: Seleccionar la opción Modificar de una solicitud de eventos. Modificar el campo “Título del trabajo” y ponerlo en blanco. Modificar los datos correctamente. Verificar que la solicitud de eventos modificada sea almacenada y mostrada correctamente en el “Listar solicitudes de eventos”.	
Resultado Esperado: El sistema muestra los campos con los datos de la solicitud de evento seleccionada.	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

<p>El sistema debe alertar al usuario que hay campos vacíos.</p> <p>El sistema debe almacenar los datos modificados y posteriormente informarle al usuario que los datos fueron modificados satisfactoriamente.</p> <p>El sistema debe mostrar los datos con las modificaciones realizadas.</p>
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 25: CP03 - Gestionar solicitar evento

Caso de Prueba de Aceptación	
Código Caso de Prueba: 08PA03	Nombre Historia de Usuario: Gestionar solicitar evento
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Listar todas las solicitudes de eventos realizadas por el usuario. Se visualiza toda la información referente a las solicitudes.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: <p>Seleccionar la opción Listar eventos.</p> <p>No existen solicitudes en la Base de datos para este usuario.</p> <p>El usuario tiene varias solicitudes hechas y almacenadas en la Base de datos.</p>	
Resultado Esperado: <p>El sistema debe mostrar el nodo Listar eventos.</p> <p>El sistema debe mostrar el siguiente mensaje <i>No existen eventos solicitados por usted, si desea solicitar un nuevo evento, por favor, diríjase a la sección de solicitar evento en el menú o a través de este link</i>. El sistema debe mostrar un link que le permite ir al nodo Solicitar eventos.</p> <p>El sistema debe mostrar en forma de tabla todas las solicitudes de eventos hechas por el usuario, con cada uno de los datos que la misma contiene.</p>	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Evaluación de la Prueba: *Satisfactoria*

Tabla 26: CP04 - Gestionar solicitar evento

Caso de Prueba de Aceptación	
Código Caso de Prueba: 08PA04	Nombre Historia de Usuario: Gestionar solicitar evento
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se elimina una solicitud de evento.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: Seleccionar la opción Eliminar de una solicitud de evento. Verificar que la solicitud de evento eliminada no aparece en el Listar solicitud.	
Resultado Esperado: El sistema debe mostrar una ventana de aviso al usuario preguntando si realmente desea eliminar la solicitud de evento. Después de eliminado, la solicitud no debe aparecer en la lista.	
Evaluación de la Prueba: <i>Satisfactoria</i>	

Tabla 27: CP01 - Gestionar revisión de trabajos

Caso de Prueba de Aceptación	
Código Caso de Prueba: 09PA01	Nombre Historia de Usuario: Gestionar revisión de trabajos.
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se inserta una revisión de trabajo.	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Condiciones de Ejecución: Solo interviene el usuario administrador.
Entrada / Pasos de ejecución: Seleccionar la opción Gestionar revisión. Seleccionar la opción asignar revisión. No existen solicitudes validas almacenadas para el evento seleccionado. Insertar el campo usuario en blanco. Insertar un usuario que no está registrado. No seleccionar ninguna solicitud de evento. Insertar todos los datos correctamente.
Resultado Esperado: El sistema debe mostrar un nodo con una lista de todos los eventos que están en convocatoria. El sistema debe mostrar los campos necesarios para que se lleve a cabo la asignación de trabajos a revisar. El sistema debe mostrar un mensaje diciéndole al usuario que no existen aún solicitudes válidas para ese evento. El sistema debe mostrar un mensaje advirtiéndole al usuario que el campo usuario es obligatorio. El sistema debe mostrar un mensaje advirtiéndole al usuario que el usuario entrado no existe. El sistema debe mostrar un mensaje advirtiéndole al usuario que debe elegir al menos una solicitud de eventos. El sistema almacena los datos insertados y muestra un mensaje de almacenamiento realizado satisfactoriamente.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 28: CP02 - Gestionar revisión de trabajos

Caso de Prueba de Aceptación

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Código Caso de Prueba: 09PA02	Nombre Historia de Usuario: Gestionar revisión de trabajos
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se modifica una revisión de trabajo.	
Condiciones de Ejecución: Solo interviene el usuario administrador.	
Entrada / Pasos de ejecución: Seleccionar la opción Gestionar revisión de trabajos. Seleccionar la opción modificar revisión de trabajos. No existen revisiones de trabajos. Seleccionar la revisión de trabajo a modificar. Modificar los datos del campo usuario en blanco. Modificar los datos del campo usuario con un usuario que no está registrado. Modificar los datos del campo usuario correctamente. Listar la revisión de trabajo modificada correctamente.	
Resultado Esperado: El sistema debe mostrar un nodo con una lista de todos los eventos que están en convocatoria. El sistema debe mostrar un nodo con una lista de todas las solicitudes realizadas a ese evento en convocatoria. El sistema debe mostrar un mensaje diciéndole al usuario que no existen revisiones de trabajo para ese evento. El sistema debe mostrar los campos con los datos de la revisión de trabajo seleccionada. El sistema debe mostrar un mensaje advirtiéndole al usuario que el campo usuario es obligatorio. El sistema debe mostrar un mensaje advirtiendo que el usuario entrado no existe. El sistema almacena los datos modificados y muestra un mensaje de	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

almacenamiento realizado satisfactoriamente. El sistema debe listar la solicitud con las modificaciones realizadas correctamente.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 29: CP03 - Gestionar revisión de trabajos

Caso de Prueba de Aceptación	
Código Caso de Prueba: 09PA03	Nombre Historia de Usuario: Gestionar revisión de trabajos
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se listan todas revisiones de trabajos por cada evento en convocatoria. Se publican eventos con no contengan solicitudes ni revisiones.	
Condiciones de Ejecución: El usuario debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: Seleccionar la opción Listar revisión de trabajos.	
Resultado Esperado: El sistema debe mostrar todas las revisiones de trabajos asignadas al usuario con todos los datos referentes a la solicitud de eventos. En caso que no contenga ninguna revisión asignada el sistema debe mostrarle un mensaje de que no existen revisiones asignadas.	
Evaluación de la Prueba: <i>Satisfactoria</i>	

Tabla 30: CP01 - Revisar trabajos

Caso de Prueba de Aceptación	
Código Caso de Prueba: 10PA01	Nombre Historia de Usuario: Revisar trabajos

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz
Descripción de la Prueba: Se realiza la revisión de los trabajos asignados
Condiciones de Ejecución: Solo interviene el usuario administrador.
Entrada / Pasos de ejecución: Seleccionar la opción revisar trabajo de una solicitud de evento. No subir ningún documento. La URL no se encuentra. Subir documento revisado.
Resultado Esperado: El sistema muestra un nodo con los campos correspondientes. El sistema debe mostrar un mensaje advirtiéndole al usuario que debe subir un documento. El sistema debe mostrar un mensaje advirtiéndole al usuario que la dirección del documento no es correcta. El sistema debe mostrar un mensaje informando al usuario que se ha subido la revisión correctamente.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 31: CP01 - Gestionar actividad postgraduada

Caso de Prueba de Aceptación	
Código Caso de Prueba: 11PA01	Nombre Historia de Usuario: Gestionar actividad postgraduada
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se insertan los datos una actividad postgraduada. Inicialmente se insertarán incorrectamente para verificar las validaciones del sistema, luego de forma correcta para comprobar que los datos sean almacenados.	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Condiciones de Ejecución: Solo interviene el usuario administrador.
Entrada / Pasos de ejecución: Se selecciona la opción información postgraduada. Insertar campos en blanco. Insertar el usuario en blanco y los otros datos correctamente. Insertar las referencias en blanco y los otros datos correctamente. No subir el documento y los otros datos correctamente. Insertar los datos correctamente y la fecha mal. Insertar un usuario y referencia y los otros datos bien.
Resultado Esperado: El sistema debe mostrar los indicadores por los que se puede realizar la asignación de una información postgraduada. El sistema debe alertar al usuario cuando se inserten datos en blanco y o incorrectamente escritos en los campos (usuario, referencia, diploma, indicador y fecha). Cuando se inserten los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos en un listado.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 32: CP02 - Gestionar actividad postgraduada

Caso de Prueba de Aceptación	
Código Caso de Prueba: 11PA02	Nombre Historia de Usuario: Gestionar actividad postgraduada
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se modifican los datos de una actividad postgraduada. Inicialmente se modificarán incorrectamente para verificar las validaciones del sistema, luego de forma correcta para comprobar que los datos sean almacenados y	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

cargados.
Condiciones de Ejecución: Solo interviene el usuario administrador.
Entrada / Pasos de ejecución: Seleccionar la opción Modificar en una información postgraduada específica. Modificar campos en blanco. Modificar el usuario en blanco y los otros datos correctamente. Modificar las referencias en blanco y los otros datos correctamente. Modificar los datos correctamente y la fecha mal. Modificar de forma correcta los datos. Verificar que el elemento aparece en el listado con los nuevos datos.
Resultado Esperado: El sistema debe alertar al usuario cuando se modifiquen datos en blanco y o incorrectamente en los campos (usuario, referencia, diploma, indicador y fecha). Cuando se modifiquen los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos en un listado.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 33: CP03 - Gestionar actividad postgraduada

Caso de Prueba de Aceptación	
Código Caso de Prueba: 11PA03	Nombre Historia de Usuario: Gestionar actividad postgraduada
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se elimina una información postgraduada.	
Condiciones de Ejecución: Solo interviene el usuario administrador.	
Entrada / Pasos de ejecución: Seleccionar la opción Eliminar de una información postgraduada.	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Verificar que el usuario eliminado no aparece en el listado.
Resultado Esperado: El sistema debe mostrar una ventana de aviso al usuario preguntando si realmente desea eliminar. Cuando se acepte, debe eliminar la información postgraduada. Después de eliminado, el elemento no debe aparecer en la lista.
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 34: CP0 - Buscar información postgraduada

Caso de Prueba de Aceptación	
Código Caso de Prueba: 11PA03	Nombre Historia de Usuario: Buscar información postgraduada
Nombre de la persona que realiza la prueba: Emilio Álvarez Muñoz	
Descripción de la Prueba: Se busca la información postgraduada a partir de indicadores predefinidos.	
Condiciones de Ejecución: Solo interviene el usuario administrador.	
Entrada / Pasos de ejecución: Seleccionar la opción buscar información postgraduada. Filtrar la búsqueda. Verificar todos los datos encontrados.	
Resultado Esperado: El sistema debe mostrar un nodo con los indicadores a través de los cuales se realizará la búsqueda. El sistema debe mostrar todos los datos encontrados filtrando la búsqueda y mostrando todos sus datos en forma de tabla.	
Evaluación de la Prueba: <i>Satisfactoria</i>	

Capítulo 2. Construcción y prueba de la solución

Aplicación web para la gestión de información de investigación y postgrado de la Facultad 4.

Conclusiones parciales

Se cumplió con cada uno de los objetivos trazados para la validación y prueba del sistema como un todo ya que, cada uno de los releases obtenidos en las diferentes iteraciones, resultaron de evaluación satisfactoria por las pruebas de aceptación aplicadas. Por tanto, la culminación del presente capítulo representa la garantía, calidad y satisfacción conjunta del cliente y el equipo de desarrollo con el producto final.

Conclusiones generales

Con el desarrollo de la investigación se lograron cumplir los objetivos propuestos al principio de la misma, “Desarrollar una Aplicación que gestione la información de investigación y postgrado de la facultad 4”. Logrando de esta forma dotar al mismo de flexibilidad. A continuación se mencionan los resultados obtenidos durante la investigación:

- » Se realizó el estudio del arte el cual nos permitió conocer acerca de los diferentes *softwares* existentes en el mundo que permiten la realización exitosa de la aplicación.
- » Se investigaron los principales sistemas de gestión de información y posibles soluciones que podían darle solución al problema planteado, aunque se llegó a la conclusión de que ninguno cumplía las condiciones necesarias.
- » Se justificaron las herramientas, metodologías y tecnologías que se utilizaron en el desarrollo del trabajo las cuales permitieron la implementación de un sistema que cumpla con los requisitos necesarios para una eficaz gestión de información.
- » Se realizó la implementación del sistema atendiendo a la propuesta arquitectónica y se realizaron las pruebas pertinentes para la validación de la solución propuesta mediante el diseño y aplicación de las pruebas de aceptación para validar la calidad de la solución propuesta arrojando resultados positivos.

Por todo lo antes mencionado se evidencia el cumplimiento de los objetivos propuestos en el presente trabajo de diploma, lo cual conlleva a un cumplimiento del objetivo general.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida y sin dejar de tener en cuenta que este trabajo solo es la primera versión de este sistema y que pueden surgir nuevas necesidades del cliente se proponen las siguientes recomendaciones:

- » Garantizar que el sistema implementado tenga debida atención y actualización para mantener de forma confiable su carácter de fuente fiable de información calidad relacionados con la investigación y postgrado.
- » Ampliar las funcionalidades del módulo con los nuevos requerimientos que surjan por necesidades del cliente.
- » Incluir una ayuda con el objetivo de viabilizar posibles dudas que puedan tener usuarios comunes respecto a las funcionalidades de los servicios desarrollados.
- » Migrar a la última versión existente del CMS Drupal que incluye otras mejoras funcionales.

Referencias bibliográficas

1. The British Standards Institution. [En línea] BSI, 2009. [Citado el: 25 de 11 de 2010.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion>.
2. **Ivar Jacobson, Grady Booch y James Rumbaugh.** El Proceso Unificado de Desarrollo. *La guía completa del Proceso Unificado*. Madrid Addison Wesley : s.n., 2000.
- 3.
4. **José H. Canós, Patricio Letelier y Carmen Penadés.** Metodologías Ágiles en el Desarrollo de *Software*. [En línea] [Citado el: 14 de 12 de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
5. Morón, Universidad. Metodologías Ágiles. [En línea] [Citado el: 15 de 12 de 2010.] http://noqualityinside.com/nqi/nqifiles/Metodologias_Agiles.pdf.
6. **Calderon, Anyelin.** Sistematizando Aprendizajes. [En línea] [Citado el: 15 de 12 de 2010.] <http://anyelincalderon.blogspot.com/2010/04/algo-sobre-metodologias-agiles.html>.
7. **Mendoza Sanchez, María A.** [En línea] 6 de 7 de 2004. [Citado el: 16 de 12 de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
8. Visual Paradigm for UML . [En línea] 2007. [Citado el: 7 de 1 de 2011.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/).
9. **Valdés, Damián Pérez.** Maestros de la web. [En línea] [Citado el: 28 de 11 de 2010.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.
10. Conceptos básicos del servidor web. [En línea] [Citado el: 1 de 12 de 2010.] http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
11. The Apache *Software* Foundation. [En línea] [Citado el: 4 de 12 de 2010.] <http://www.apache.org>.

12. Internet Information Services. [En línea] [Citado el: 6 de 12 de 2010.] <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.
13. cavi. [En línea] [Citado el: 6 de 12 de 2010.] <http://www.cavi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>.
14. **Daniel, Pecos.** PostgreSQL. [En línea] [Citado el: 9 de 12 de 2010.] http://danielpecos.com/docs/mysql_postgres/x15.html.
15. **Mora, Marc Gibert Ginestà.Oscar Pérez.** [En línea] [Citado el: 12 de 12 de 2010.] http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf.
16. **GERSBACH, M.** Principales características de Drupal. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.factoriadigital.com/ir.php/a/aplicacionesweb/drupal>.
17. Tufuncion.com . [En línea] 2007. [Citado el: 6 de 1 de 2011.] <http://www.tufuncion.com/zend-studio>.
18. Softonic.com . [En línea] 2010. [Citado el: 7 de 1 de 2011.] <http://www.php-designer.softonic.com>
19. Netbeans.org. [En línea] 2008. [Citado el: 7 de 1 de 2011.] http://netbeans.org/index_es.html.
20. desarrolloweb.com. [En línea] [Citado el: 8 de 1 de 2011.] [desarrolloweb.com. http://www.desarrolloweb.com/articulos/1622.php](http://www.desarrolloweb.com/articulos/1622.php).
21. slideshare.net. [En línea] [Citado el: 8 de 1 de 2011.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.
22. oness.sourceforge.net. [En línea] [Citado el: 9 de 1 de 2011.] [oness.sourceforge.net. http://oness.sourceforge.net/proyecto/html/ch03s02.html](http://oness.sourceforge.net/proyecto/html/ch03s02.html).
23. **Villegas, A.A.** A Propósito de la programación extrema XP(eXtreme Programming). [En línea] [Citado el: 14 de 2 de 2011.] <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml>.
24. **Beck, Kent. 2000.** *Extreme Programming Explained*. Addison-Wesley Professional. [En línea] 2009. [Citado el: 8 de marzo de 2010.] <http://www.goodreads.com>

25. **Peña, Lizandra.** Analisis y diseño de la aplicación para la gestión de información de investigación y postgrado de la Facultad 4. s.l. : UCI, 2011.
26. Drupal. [Online] [Cited: Mayo 15, 2011.] <http://www.drupal.org>.
27. Sistema Centroamericano de Investigación y Postgrados. [En línea] [Citado el: 16 de Mayo de 2011.] <http://sicar.csuca.org/>.
28. SISTEMA DE INFORMACIÓN DE GESTIÓN ACADÉMICA. [En línea] [Citado el: 10 de Mayo de 2011.] <https://www.siga.usm.cl/pag/>..
29. [En línea] Instituto Pedagógico Latinoamericano y Caribeño. [Citado el: Febrero de 13 de 2011.] <http://www.iplac.rimed.cu/#>.
30. <http://monografias.umcc.cu>. [En línea] [Citado el: Marzo de 25 de 2011.] <http://monografias.umcc.cu/monos/2006/Informatica/Sistema%20Automatizado%20para%20la%20Gestin%20de%20la%20Informacin%20Ciencia%20y%20Tcnica.pdf>.
31. **Corporation, Oracle.** Mysql. [En línea] 2010. [Citado el: 9 de Marzo de 2011.] <http://www.mysql.com/downloads/mysql/>.

Anexos

Anexo 1: Historia de Usuario Gestionar usuarios.

Historia de Usuario Gestionar usuarios

Historia de Usuario	
Número: 1	Nombre: Gestionar usuarios
Usuario: Administrador, asesor por Departamento	
Modificación de la Historia de Usuario:	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: Permite administrar las cuentas de los usuarios del sistema, posibilitando incluir, modificar, eliminar y ver los datos de las mismas. A cada usuario se le darán los permisos que necesite en dependencia de su responsabilidad en el sistema.	

Anexo 2: Historia de Usuario Autenticar usuarios.

Historia de Usuario Autenticar usuarios

Historia de Usuario	
Número: 2	Nombre: Autenticar usuarios
Usuario: Administrador, asesor por Departamento, profesor	
Modificación de la Historia de Usuario:	Iteración Asignada: 1

Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: El individuo debe introducir para autenticarse, su usuario y contraseña, según sus privilegios accederá a la sesión correspondiente con su rol en la aplicación.	

Anexo 3: Historia de Usuario Gestionar foros.

Historia de Usuario Gestionar foros

Historia de Usuario	
Número: 3	Nombre: Gestionar foros
Usuario: Administrador	
Modificación de la Historia de Usuario:	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: Se realizan las acciones de incluir, eliminar, modificar y ver los temas y comentarios en la aplicación. Todos los usuarios pueden ver la información mostrada.	

Anexo 4: Historia de Usuario Gestionar artículos.

Historia de Usuario Gestionar artículos

Historia de Usuario	
Número: 4	Nombre: Gestionar artículos

Usuario: Administrador	
Modificación de la Historia de Usuario:	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: Se realizan las acciones de incluir, eliminar, modificar y ver los artículos en la aplicación. Todos los usuarios pueden ver los artículos publicados.	

Anexo 5: Historia de Usuario Gestionar documentos.

Historia de Usuario Gestionar documentos

Historia de Usuario	
Número: 5	Nombre: Gestionar documentos
Usuario: Administrador	
Modificación de la Historia de Usuario:	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgos en Desarrollo: Media	Puntos Reales: 1
Descripción: Se realizan las acciones de incluir, eliminar, modificar y ver documentos de diferentes tipos ya sean útiles, de consultas o trabajos realizados.	

Anexo 6: Historia de Usuario Gestionar cursos.

Historia de Usuario Gestionar cursos

Historia de Usuario	
Número: 6	Nombre: Gestionar cursos
Usuario: Administrador	
Modificación de la Historia de Usuario:	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: Se realizan las acciones de incluir, eliminar, modificar y ver los cursos. El usuario podrá obtener toda la información necesaria acerca de los cursos que se desarrollarán.	

Anexo 7: Historia de Usuario Gestionar eventos.

Historia de Usuario Gestionar eventos

Historia de Usuario	
Número: 7	Nombre: Gestionar eventos
Usuario: Administrador	
Modificación de la Historia de Usuario:	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1

Riesgos en Desarrollo: Baja	Puntos Reales: 1
Descripción: Se realizan las acciones de incluir, eliminar, modificar y ver los eventos. El usuario podrá obtener toda la información necesaria acerca de los eventos que se desarrollarán.	

Anexo 8: Historia de Usuario Solicitar eventos.

Historia de Usuario Solicitar eventos.

Historia de Usuario	
Número: 8	Nombre: Gestionar solicitud de eventos
Usuario: Profesor	
Modificación de la Historia de Usuario:	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgos en Desarrollo: Media	Puntos Reales:
Descripción: Inicia cuando el profesor selecciona el evento en el que desea participar y adjunta un resumen quedando avalada su futura participación. Permite eliminar, modificar y listar solicitudes de eventos.	

Glosario de términos

UML: Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language).

CVS: Sistema de Control de Versiones (CVS, por sus siglas en inglés, Concurrent Versions System).

ORM: Mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping).

XML: Siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible').

EJB: Enterprise JavaBeans (también conocidos por sus siglas EJB).

HTML: siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

ABC: lenguaje de programación de alto nivel desarrollado a principios de los años 1980.

GPL GNU: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License. Está orientada principalmente a proteger la libre distribución, modificación y uso de software.

CSS: El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

API: Una interfaz de programación de aplicaciones o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

CRUD: Acrónimo de Create-Read-Update-Delete (Crear, leer, actualizar, eliminar)