

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA INFORMÁTICO PARA LA
GESTIÓN DE EVIDENCIAS EN LOS PROCESOS DE INVESTIGACIÓN Y POSTGRADO
DE LA FACULTAD 4**



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

Autores:

Yanet Monterrey Zayas
Manuel Alejandro Pérez Silva

Tutor:

Ing. Lizardo Ramírez Tabuada

Cotutor(es):

Ing. Leonardo Rodríguez González
MSc. María Caridad Valdés Rodríguez

La Habana, junio 2012

Año 54 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas, así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Yanet Monterrey Zayas

Autor

Manuel Alejandro Pérez Silva

Autor

Ing. Lizardo Ramírez Tabuada

Tutor

Ing. Leonardo Rodríguez González

Co- Tutor

MSc. María Caridad Valdés Rodríguez

Co-Tutora



“Si no existe la organización, las ideas, después del primer momento de impulso, van perdiendo eficacia.”

Yanet

A las personas que más quiero y admiro en el mundo: mi madre, mi abuela Carmen y abuelo Virgilio.

Manuel

Dedico este trabajo a todos aquellos que, por una u otra razón, no se encuentran entre nosotros, me refiero a aquellos que se fueron paulatinamente y que hubieran querido llegar hasta aquí.

A mis padres, mi hermana, mi familia y amigos.



AGRADECIMIENTOS

A mi Familia en primer lugar, a mi Mamá, a mi abuela Carmen, a mis tías Marielis, Mariolis y Yunia, mi tío Robe, a mis primos Omar y Yarisley por creer en mí y por el apoyo que me dieron durante todos estos años.

A Fernando por inculcarme el bichito de estudiar, por su apoyo incondicional y su invaluable ayuda durante mis cinco años de carrera; por ser el sostén espiritual, compañero, amigo y el amor de mi vida.

Gracias también a los padres de Fernando, Ana y el Niño por acogerme como una hija, a Anita y Adairis por ser también mis amigas.

*A Orlando y su familia por acogernos en su familia a mi mamá y a mí.
A Alexeis, Maike!, Emilio por su ayuda, sus consejos y paciencia conmigo.*

A los pipos de la Laguna, Ernesto y Juan Pablo, a mi compañero de Tesis, Lyannet por las horas revisando ortografía.

A mi abuelo Virgilio, por quererme tanto.

A mis profesores de la carrera y a todos aquellos que de una forma u otra contribuyeron a mi formación.

A todos, gracias.

AGRADECIMIENTOS

A nuestro Señor, Dios Todopoderoso, por guiarme y socorrerme en los momentos difíciles.

A mis padres por traerme al mundo, educarme, apoyarme y siempre estar ahí para lo que necesité.

A mis abuelas que han rezado mucho por mí y siempre se preocupan por todo lo que me pasa.

A Gloria, mi eterno amor, por aguantarme, apoyarme estar ahí cuando la necesité, en las buenas como en las malas y peores.

A todos los profesores que durante 5 años supieron instruirme y enseñarme, no solo cosas de la carrera sino formas de enfrentarme a la vida y seguridad ante la toma de decisiones.

A mi familia y amigos por su apoyo y por creer en mí.

A todos, gracias.

RESUMEN

El departamento de Investigación y Postgrado de la Facultad 4 necesita de un sistema para la gestión de las evidencias que se registran en el mismo, debido a que actualmente existen dificultades en la gestión de las mismas. Por esta razón la presente investigación se centra en desarrollar un sistema para la gestión de las evidencias obtenidas por la comunidad profesoral de la facultad. Para lograr los objetivos trazados, se realizó un estudio de soluciones similares para la gestión de documentos en los ámbitos nacional e internacional y se efectuó un análisis de las diferentes herramientas y tecnologías posibles a utilizar, escogiendo a Drupal como sistema gestor de contenido, Geany como entorno de desarrollo integrado, Extreme Programming (XP) como metodología para guiar el proceso de desarrollo del software atendiendo, a las distintas fases que propone. La aplicación de la metodología escogida permitió una buena interacción con el cliente, lo que facilitó el entendimiento de las características que debe poseer el sistema. La planificación de esfuerzo por Historia de Usuario fue de suma importancia para definir el tiempo de entrega, la implementación para el desarrollo del sistema y las pruebas para comprobar que el sistema cumple con todo lo definido por el cliente.

Palabras claves: Drupal, evidencia, gestión documental

Índice

PENSAMIENTO.....	II
DEDICATORIA	III
AGRADECIMIENTOS.....	IV
AGRADECIMIENTOS.....	V
Índice.....	VII
Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Gestión documental.....	5
1.2 Las evidencias en las ARC de Investigación y Postgrado en la Facultad 4.....	7
1.3 Soluciones similares	8
1.4 Metodologías de desarrollo.....	10
1.5 <i>Frameworks</i>	14
1.6 Sistemas de gestión de contenidos.....	14
1.7 Sistemas gestores de bases de datos	17
1.8 Servidor web.....	19
1.9 Lenguajes de programación.....	21
1.10 Entorno de desarrollo integrado.....	24
1.11 Conclusiones parciales	25
Capítulo 2. Propuesta de solución	27
2.1 Descripción del proceso de gestión de evidencias.....	27
2.2 Objetivos del sistema.....	28
2.3 Definición de la audiencia	28
2.4 Descripción del sistema.....	29
2.5 Usuarios relacionados con el sistema.....	29
2.6 Historias de usuario	30
2.7 Estimación de esfuerzo por HU	33
2.8 Iteraciones.....	34
2.9 Plan de entregas.....	35
2.10 Metáfora	36
2.11 Tarjetas CRC.....	36
2.12 Conclusiones parciales	38
Capítulo 3. Implementación y prueba.....	39

3.1	Tema	39
3.2	Módulos.....	41
3.3	Configuración	43
3.4	Nodo.....	45
3.5	Tipos de contenido	46
3.6	Pruebas	48
3.7	Conclusiones parciales	56
	Conclusiones generales.....	57
	Recomendaciones	58
	Referencias bibliográficas	59
	Anexos.....	64
	Glosario de términos.....	71

Introducción

Desde la antigüedad, el hombre ha sentido la necesidad de guardar sus experiencias y recuerdos. Por ello, surgieron medios primitivos para grabar ese conocimiento. Más adelante con el descubrimiento del papel, la humanidad obtuvo una nueva forma para expresar y hacer perdurar sus pensamientos, siendo estas las bases para la confección de los documentos, libros y artículos que conocemos hoy día. La acumulación, cada vez mayor, de documentos producto del propio desarrollo de la sociedad lleva al nacimiento de la gestión documental (GD).

Con el empleo de las Tecnologías de la Información y las Comunicaciones (TIC), se ofrece la posibilidad de almacenar grandes volúmenes de documentos en un espacio reducido, todo en formato digital. Para el tratamiento de los documentos surgen los sistemas de gestión documental o sistemas de gestión de documentos (SGD).

En el ámbito empresarial, institucional y en las organizaciones, sus acciones y conocimiento quedan documentados cada vez más en soportes electrónicos, que constituyen la evidencia de su actividad. Tanto las necesidades de cumplimiento legal, como de auditoría, de preservación del conocimiento o del propio funcionamiento operativo, obligan a plantearse la gestión de los documentos y evidencias como un componente estratégico de negocio y con un máximo aprovechamiento de las tecnologías de la información.

En Cuba, el proceso de informatización de la sociedad, ha conducido en los últimos años a un incremento en el uso de los SGD. La Universidad de las Ciencias Informáticas (UCI) también los ha incorporado para facilitar los procesos que en ella se llevan a cabo.

Dentro de los procesos fundamentales que tienen lugar en la UCI, se encuentran: la producción de software, prestación de servicios y superación de sus profesionales. Estos procesos generan producciones intelectuales y de superación postgraduada, que aportan evidencias a los indicadores de la ciencia, la tecnología y la innovación (CTI).

A pesar de que en la universidad se cuenta con la tecnología necesaria para informatizar los procesos que en ella se llevan a cabo, en la Facultad 4 la gestión de las evidencias en las Áreas de Resultados Claves (ARC) de Investigación y Postgrado, en ocasiones, se realiza de forma manual, por no contar con un sistema informático para ello. Este proceso, que involucra a los docentes, coordinadores y cuadros de diferentes niveles organizativos, genera extensos sistemas de despachos los cuales dificultan las gestiones habituales de

estas dos áreas de trabajo y, a su vez, obstaculizan la entrega ágil y sistemática de la actualización que ellas requieren.

Además se evidencian otras limitaciones como: el no contar con un espacio virtual dónde archivar las evidencias ni con un registro histórico para futuras comparaciones de resultados. La localización y/o búsqueda de una evidencia específica se dificulta por lo disperso de la información.

Es por ello que en estos momentos los cuadros no cuentan con la suficiente viabilidad para el control sistemático que les facilite los reportes e informes del estado de las evidencias en estas dos ARC.

Por lo anteriormente planteado el **problema a resolver** queda resumido en la siguiente interrogante: ¿Cómo dar solución a las dificultades existentes en la gestión de evidencias de los procesos de Investigación y Postgrado en la Facultad 4?

Para dar solución al problema planteado se establece el siguiente **objetivo general**: desarrollar el análisis, diseño e implementación de un sistema que informatice la gestión de evidencias en los procesos de Investigación y Postgrado de la Facultad 4.

De ahí que se defina como **objeto de estudio**, los procesos de la GD y como **campo de acción**, la gestión de las evidencias en los procesos de Investigación y Postgrado en la Facultad 4.

El presente estudio defiende la **idea de que** el desarrollo de un sistema informático contribuirá con la gestión de las evidencias en los procesos de Investigación y Postgrado en la Facultad 4.

El objetivo general de esta investigación deriva en los siguientes **objetivos específicos**:

- Desarrollar el análisis del proceso de gestión de las evidencias en las ARC de Investigación y Postgrado en la Facultad 4.
- Diseñar un sistema informático para la gestión de evidencias en los procesos de Investigación y Postgrado.
- Implementar y probar el sistema informático para la gestión de evidencias en los procesos de Investigación y Postgrado.

Para darle cumplimiento a estos objetivos específicos se definieron las **tareas** siguientes:

1. Elaboración de los aspectos teóricos que sustentan la investigación.
2. Identificación y análisis de las tendencias actuales sobre los SGD.

3. Análisis de las herramientas y componentes para el sistema.
4. Realización de entrevistas al cliente para saber cómo quiere que funcione el sistema a desarrollar.
5. Descripción de las funcionalidades y características de la aplicación.
6. Definición de los artefactos propuestos por la metodología seleccionada.
7. Descripción de las funcionalidades y características que tendrá el sistema.
8. Implementación de las funcionalidades especificadas del sistema.
9. Realización de pruebas al sistema.

En la presente investigación se emplean los métodos investigativos teóricos y empíricos. A continuación se mencionan cada uno de ellos y cómo se ponen de manifiesto en la investigación.

Métodos teóricos

Analítico-Sintético: A través del mismo se realiza una descripción de las principales metodologías, tecnologías y herramientas posibles a utilizar en el desarrollo de la investigación y se seleccionan las más convenientes, en dependencia de sus características.

Histórico-Lógico: Mediante este método se estudia los conceptos y la evolución en el tiempo de la GD y los SGD.

Métodos empíricos

Observación: Con este método es posible recopilar una serie de datos confiables que ayudan a la comprensión del fenómeno y a la definición de lo primordial de la problemática. Al mismo tiempo esto posibilita el planteamiento del problema permitiendo enmarcar el objeto de estudio y el campo de acción, lo cual propicia enfocar la investigación hacia lo que se necesita alcanzar y cómo alcanzarlo.

Para lograr el objetivo general de esta investigación, la misma se estructuró de la siguiente manera:

Capítulo 1- Fundamentación teórica

En este capítulo se describen conceptos y características que se deben tener en cuenta para alcanzar los objetivos del trabajo. Se analizan las aplicaciones similares existentes y, a juicio de los autores de la presente investigación, se justifica la selección de las herramientas y tecnologías a utilizar para dar la solución del problema planteado.

Capítulo 2- Propuesta de solución

En este capítulo se resumen los aspectos relacionados con la fase de planificación y de diseño de la metodología seleccionada. Se realiza una estimación de esfuerzos que permite dar una idea de la duración del proceso de desarrollo. Además se describe la solución propuesta y se definen las Historias de Usuarios que constituyen las características del sistema.

Capítulo 3- Implementación y prueba

En este capítulo son implementadas las Historias de Usuarios definidas con anterioridad. Se describe como se construyó el tema “SGEIP¹”. También se explica cómo se realizó la configuración del sistema una vez implementado todos los módulos, para lograr su correcto funcionamiento. Además se plasman los casos de prueba o *test* de aceptación a las que fue sometida la aplicación en cada una de las iteraciones.

¹ Abreviatura para la aplicación SGEIP (Sistema para la Gestión de Evidencias en Investigación y Postgrado).

Capítulo 1. Fundamentación teórica

Introducción

La revolución tecnológica que vive la humanidad, hoy en día, se debe en parte a los avances significativos de las TIC. Los grandes cambios que caracterizan esta nueva sociedad están dados por el uso de las tecnologías, las redes de comunicaciones, el rápido desenvolvimiento científico y la globalización de la información (1).

Cuba se encuentra inmersa en el proceso de desarrollo de la sociedad caracterizado, entre otros aspectos, por el empleo de los SGD en organismos, instituciones, organizaciones y el ámbito empresarial. La implantación de un sistema de gestión de este tipo es clave para optimizar los procesos, mejorar los servicios y la seguridad de la información.

En el desarrollo de este capítulo se investigan aspectos fundamentales relacionados con la GD, los SGD, la metodología, los lenguajes, herramientas y tecnologías empleadas para el desarrollo de las funcionalidades del Sistema de Gestión de Evidencias. Además, se realiza un estudio de algunos de los de sistemas similares a la propuesta de solución.

1.1 Gestión documental

Antes de abordar la GD se debe definir el concepto de documento. Cuando se refiere a documento se hace alusión al documento archivístico que es la esencia de la GD y que según la norma ISO² 15489 es (2): *“información creada o recibida, conservada como información y prueba, por una organización o un individuo en el desarrollo de sus actividades o en virtud de sus obligaciones legales”*. En lo adelante, cuando se hable de documentos se entenderá que se trata de la definición dada en este apartado.

La GD, tiene sus orígenes en la década del '40, debido a la acumulación de grandes volúmenes documentales por parte de las administraciones de los estados. En los Estados Unidos de América esta preocupación se venía gestando desde finales del siglo XIX, y se fueron desarrollando acciones en este sentido. En el año 1934 se crea el Archivo Nacional y en el año 1950 se institucionaliza la GD en la Ley de Documentos Federales como forma idónea del tratamiento documental (3).

Existen varias definiciones de GD pero a los efectos de este trabajo solo se utilizarán dos de ellas. La primera se toma del Diccionario de Terminología Archivística del Consejo

² Organización Internacional para la Estandarización o International Organization for Standardization.

Internacional de Archivo, (año 1993) para el que la GD es (4): *“un área de la administración general que se encarga de garantizar la economía y eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida”*.

Por otra parte, la norma ISO 15489 refiere que por GD se entiende (2): *“el área de gestión responsable de un control eficaz y sistemático de la creación, la recepción, el mantenimiento, el uso y la disposición de documentos de archivo, incluidos los procesos para incorporar y mantener en forma de documentos la información y prueba de las actividades y operaciones de la organización”*.

Si bien no existen grandes diferencias entre ambas definiciones, se debe tomar como punto de partida una de ellas. En lo adelante cuando se haga referencia a la GD se entenderá que se trata de la segunda definición expuesta.

La evolución de la GD ha estado estrechamente ligada al desarrollo de las TIC. La aparición de las computadoras y su inserción en casi todos los sectores de la sociedad, dio inicio a la gestión electrónica de los documentos. Con la aparición de las redes, internet, las nuevas tecnologías de desarrollo de software y nuevos tipos de soporte para almacenar información, la gestión electrónica de documentos alcanzó un nuevo nivel, surgiendo así los sistemas informáticos para la gestión de documentos.

1.1.1 Sistemas de gestión documental

Para facilitar la gestión de un gran número de documentos, surgen los SGD, el uso de estos permite administrar el flujo de los mismos en una organización, recuperar información desde ellos, determinar el tiempo que deben guardarse, eliminar los que ya no tienen ninguna utilidad y asegurar la conservación indefinida de los más valiosos.

De los conceptos de SGD, a los efectos de esta investigación se asumirá el establecido por la norma ISO 15489 (2): *“sistema de información que incorpora, gestiona y facilita el acceso a los documentos de archivo a lo largo del tiempo”*.

Los primeros SGD soportados en el uso de ordenadores, surgen en la década del '60, en las grandes bibliotecas en Estados Unidos y Gran Bretaña, con el empleo de programas de bases de datos o catalogación por computadora. A partir de la década de los '80, se acelera el desarrollo de las tecnologías informáticas y de las telecomunicaciones y con ellas evolucionan los SGD, favoreciendo la gestión de un volumen cada vez mayor de

información en formato digital en las instituciones, organizaciones y empresas (5). En lo adelante, cuando se mencione a los SGD se refiere a los que hacen uso de las TIC.

1.1.2 Características de los SGD

Las características principales que posee un SGD se resumen en (6):

- Manejo de grandes volúmenes de documentación.
- Garantía de acceso a la información más actual.
- Mantenimiento coherente de la información procedente de diferentes compañías y organizaciones.
- Definición de flujos de trabajo en el sistema para la gestión de procesos operativos entre departamentos y empresas externas.
- Gestión de la información en formato nativo.
- Control de acceso a la información.
- Seguridad ante la posible pérdida de documentación.

1.1.3 Ventajas de los SGD

El empleo de los SGD implica un número importante de ventajas las cuales pueden resumirse en (7):

- Reducción de costos de los procesos de las entidades, mediante el rediseño de procesos, sustitución del trabajo administrativo no productivo y reducción del espacio físico de almacenamiento.
- Reducción de los ciclos de trabajo, aumentando la concurrencia de las distintas actividades necesarias.
- Unificación de los procesos empresariales en los distintos ámbitos departamentales y geográficos, potenciando los canales formales y los procedimientos de trabajo, lo que facilitará el cumplimiento de los requerimientos que los sistemas de calidad imponen.
- Aumento de las capacidades de comunicación en toda la organización, mejorando la integridad y seguridad de la información.

1.2 Las evidencias en las ARC de Investigación y Postgrado en la Facultad 4

Según la Real Academia Española (RAE) una evidencia es: *“certeza clara y manifiesta de una cosa, de tal forma que nadie puede dudar de ella ni negarla”* o la *“prueba determinante en un proceso”* (8), entre otras acepciones.

La evidencia dentro de la Investigación y Postgrado es el documento que respalda el reconocimiento, la certeza, seguridad y convencimiento de una aceptación y/o participación en conferencias científicas, competencias académicas, concursos, eventos, capacitaciones, cursos de Postgrado, publicaciones en diversas fuentes, así como de avales de clientes por resultados introducidos o premiados, registros y patentes.

Las evidencias que se gestionan en la ARC de Investigación y Postgrado en la Facultad 4 se constituyen como: las certificaciones de publicaciones, los premios, los resultados introducidos, registros o patentes³ en el Centro Nacional de Derecho de Autor (CENDA), los trabajos presentados, proyectos de I+D y el empleo de estudiantes. Además, los informes de los cortes de las estadísticas para el procesamiento de los indicadores.

1.3 Soluciones similares

Para informatizar la gestión de las evidencias de Investigación y Postgrado de la Facultad 4 se analizan soluciones existentes en el ámbito nacional e internacional, que tienen funcionalidades similares a las que se desean desarrollar.

1.3.1 OrfeoGPL

Es una herramienta para la GD y de código abierto, que permite gestionar electrónicamente la producción, el trámite, el almacenamiento digital y la recuperación de documentos, evitando su manejo en papel, garantizando la seguridad de la información y la trazabilidad⁴ de cualquier proceso que se implemente mediante su funcionalidad. Esta herramienta puede instalarse en cualquier sistema operativo (GNU⁵/Linux, Unix, *Windows*), con diferentes bases de datos (PostgreSQL, Oracle, MySQL y Ms-SQL) y maneja múltiples tipos de formatos, logrando así independencia de plataforma tecnológica y la reducción de costos en la implantación (9).

1.3.2 AvilaDoc

El SGD AvilaDoc fue desarrollado por la empresa Desoft de la provincia Ciego de Ávila sobre una plataforma de software libre con base de datos centralizada destinado a la gestión, tramitación y resguardo de archivos electrónicos (10).

³ La patente es un derecho, otorgado por el Estado a un inventor o a su causa habiente (titular secundario).

⁴ Capacidad para reconstruir el historial de la utilización o la localización de un artículo o producto mediante una identificación registrada.

⁵ GNU es un acrónimo recursivo que significa **GNU No es Unix** (*GNU is Not Unix*). Proyecto iniciado con el objetivo de crear un sistema operativo libre.

Es una herramienta de trabajo que gestiona el tramitado interno de los documentos a través de la intranet sin necesidad de copias o envíos físicos. Asimismo, permite el control de la entrada y salida de estos, el manejo de anotaciones por usuarios, el tratamiento diferencial para los pendientes, así como la recuperación de la información. Conserva la acción que se hizo sobre el mismo, el usuario que la acometió y la fecha en que lo realizó (10).

1.3.3 Alfresco

Es un SGD desarrollado para empresas de pequeño y mediano tamaño, que posibilita la búsqueda y obtención de documentos, colaborar en ellos y gestionar su ciclo de vida en un repositorio centralizado. Posee un sistema de ficheros virtuales, categorización del contenido, servicios de librería, flujo de trabajo integrado y garantías para la seguridad. Alfresco fue fundado como empresa en el 2005, por John Newton (11).

Síntesis de las principales características y limitaciones de los sistemas presentados

Los tres sistemas analizados anteriormente comparten los siguientes elementos comunes: son desarrollados con tecnologías web y sobre plataformas de software libre. El hecho de emplear las tecnologías web permite un mejor desempeño de la GD en empresas e instituciones que cuenten con redes internas; mientras que al ser distribuidas bajo licencias no privativas facilita su empleo, distribución, modificación e introducción de mejoras. A pesar de contar con esas bondades estos sistemas presentan limitantes que impiden su uso para el desarrollo de la solución, a continuación se describen las mismas.

AvilaDoc y **OrfeoGPL** implementan algunas funcionalidades para la gestión de documentos, pero la dificultad principal que presentan para no ser empleados en esta investigación es que no cuentan con un sistema de reportes que pueda ser personalizado con facilidad.

Por su parte, **Alfresco** es una herramienta que ofrece grandes posibilidades para el control de documentos, pero no se usa porque al estar basado en una arquitectura J2EE⁶ y usar como servidor a Apache Tomcat, según la bibliografía consultada (12), para cincuenta o más usuarios concurrentes se necesita 1 Gb de RAM y 2 servidores CPU solamente para la máquina virtual de java y no se cuenta con los requerimientos de

⁶ *Java 2 Platform Enterprise Edition.*

hardware que exige para su correcto funcionamiento en el entorno donde va a ser desplegada la aplicación.

A continuación se realiza un estudio de las herramientas y tecnologías que serán seleccionadas para el desarrollo de la solución, así como la metodología que guiará el proceso de desarrollo.

1.4 Metodologías de desarrollo

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto informático. Las mismas pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de uno a otro son tan variados y cambiantes que ha dado lugar a que exista una gran variedad de metodologías (13).

Actualmente, existen numerosas metodologías para guiar el proceso de desarrollo. Entre ellas están: las tradicionales, las ágiles y las orientadas a aplicaciones web. Según la bibliografía consultada, las tradicionales o pesadas 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. Se centran, especialmente, en el control del proceso mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y específica documentación (14).

Entre ellas podemos mencionar a *Rational Unified Procces* (RUP), *Microsoft Solution Framework* (MSF) y *Win-Win Spiral Model*, entre otras.

Estas metodologías se recomiendan especialmente para proyectos de grandes dimensiones que necesiten de una precisa documentación. Además, no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un ambiente, donde los requisitos no pueden predecirse o bien pueden variar. Es por esto que no se tienen en cuenta para la presente investigación.

Según la bibliografía consultada ((15) (16)), dentro de las metodologías guiadas al desarrollo de aplicaciones web se encuentran: *UML-Based Web Engineering* (UWE), *Ubiquitous Web Applications* (UWA), *Scenario-based Object-Oriented Hypermedia Design Methodology* (SOHDM), *Hypermedia Flexible Process Modeling* (HFPM), *Web Site Design*

Method (WSDM) y Object Oriented Hypermedia Design Model (OOHDM). Las mismas le prestan poca atención al tratamiento de requisitos y el trabajo de ellas está más enfocado en las fases de análisis, diseño e implementación. Su invención está destinada al uso del paradigma orientado a objeto y se tiende a no usarlas en la actualidad. Por lo que no se emplearán para guiar el proceso de desarrollo de la solución que se realizará.

Por tanto, se decide profundizar en las metodologías ágiles, las cuales dan mayor valor al individuo, la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

1.4.1 Metodologías ágiles

La metodología ágil tiene como factor de éxito de un proyecto de software a las personas. Prefieren crear un buen equipo y que este configure su propio entorno de desarrollo, en base a sus necesidades. No hacen énfasis en construir documentos a menos que sean necesarios, los mismos deben ser cortos y recoger lo imprescindible. Es más importante darle respuesta a un cambio que seguir un plan (13).

Dentro de las más destacadas se encuentran: *Crystal Clear*, *Feature Driven Development* (FDD), *Adaptive Software Development* (ASD), SCRUM⁷ y *Extreme Programming* (XP).

En el presente estudio se establece una comparación entre SCRUM y *Extreme Programming*, ya que son utilizadas frecuentemente para aplicaciones web. Además, promueven el trabajo en equipo y facilitan las relaciones entre el cliente y el equipo de desarrollo.

1.4.1.1 SCRUM

SCRUM es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software, el cuál se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora una nueva funcionalidad. SCRUM se utiliza como marco para otras prácticas de ingeniería de software como *RUP* o *XP* (17).

La siguiente figura muestra el ciclo de vida de SCRUM, el desarrollo del software se realiza a través de iteraciones llamadas *sprint*. La duración de una iteración es de 2 a 4 semanas.

⁷ Procede del vocabulario del rugby y significa melé; es decir, esa "figura" en la que los compañeros del equipo se amontonan, forman una piña y empujan todos en la misma dirección.



Figura 1: Ciclo de vida de la metodología SCRUM

1.4.1.2 XP

XP 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 buen clima de trabajo. Se basa en la realimentación 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. Especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico (18).

Los principios y prácticas son de sentido común pero llevadas al extremo. De ahí, la procedencia de su nombre. Las características esenciales de XP se organizan en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

Según Jeffries R (18), *“un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar, basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo”*.

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor del negocio a implementar. El autor se refiere en este punto: el cliente es el que define lo que desea.
2. El programador estima el esfuerzo necesario para su implementación. El autor se refiere en este punto: el programador es el encargado de definir el tiempo necesario para la implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo. El autor se refiere en este punto: el cliente decide que construir primero según sus necesidades.

4. El programador construye ese valor de negocio. El autor se refiere en este punto: el programador implementará lo deseado por el cliente.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo, tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma, el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

La metodología XP propone cuatro fases en su proceso de desarrollo: planificación, diseño, codificación y pruebas (19), las cuales están relacionadas entre sí, tal y como muestra la figura 2.

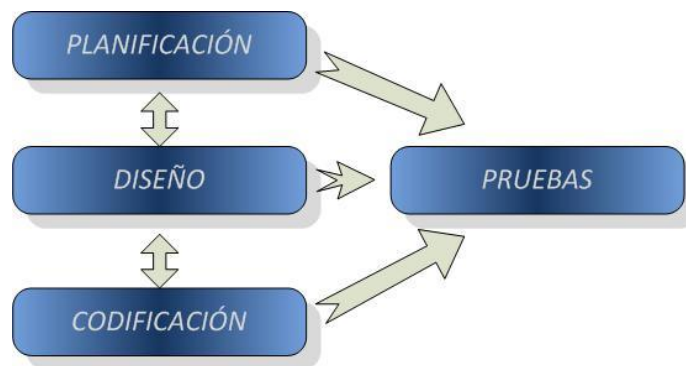


Figura 2: Fases de XP

Fundamentación de la metodología a utilizar

Toda metodología debe ser adaptada al contexto del proyecto en que se trabaja. Después de realizarse un análisis, se seleccionó como metodología para guiar el proceso de desarrollo del software a XP. El proyecto es pequeño y en el transcurso del tiempo pueden ocurrir cambios en los requisitos funcionales. XP permite administrar estos cambios de forma óptima. Además, establece que la comunicación y satisfacción del cliente es lo principal. Es decir, no hace énfasis en la documentación.

Es significativo destacar que XP es muy utilizada para portales web. Asimismo, potencia al máximo el trabajo en equipo donde los jefes del proyecto, los clientes y los desarrolladores están involucrados en el desarrollo del software.

No se utiliza SCRUM porque no hay un jefe de proyecto que oriente al equipo de desarrollo, siendo este uno de los roles más importantes en el proceso. Solo existe un rol

llamado SCRUM MASTER, para solucionar contrariedades y que no dirige. Además no presenta alguna de las prácticas que muestra XP como es la programación en parejas, el modelo de 40 horas de trabajo semanales, los roles y disponibilidad del cliente.

1.5 Frameworks

Un *framework*⁸ es una estructura conceptual y tecnológica de soporte, definida normalmente con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Además define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. Representa, también, una arquitectura de software que modela las relaciones generales de las entidades del dominio. Asimismo, provee una estructura y una metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio (20).

Dentro de los *frameworks* más usados para el desarrollo de aplicaciones web se encuentran (21): Seagull, CakePHP, Zend Framework, CodeIgniter y Symfony, este último es el propuesto por la línea arquitectónica de la Facultad 4.

1.5.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, esto lo logra haciendo uso del patrón Modelo Vista Controlador (MVC). Está desarrollado completamente en PHP5, es multiplataforma e independiente del sistema gestor de bases de datos (22).

1.6 Sistemas de gestión de contenidos

Un sistema de gestión de contenidos (CMS por sus siglas en inglés: *Content Management System*), es una aplicación web que posibilita desde un navegador y sin necesidad de tener grandes conocimientos informáticos, desarrollar sitios Web (23).

⁸ La traducción de *framework* (o en plural *frameworks*) en idioma español es marco de trabajo.

Un CMS permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio, sin que el contenido se vea afectado.

1.6.1 Tipos de CMS

Existen diversos tipos de CMS, según la bibliografía consultada se pueden clasificar teniendo en cuenta los propósitos específicos de su creación en (24):

- **Genéricos:** ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenido, para soluciones de comercio electrónico, blogs, portales, etc. Ejemplos: Zope, OpenCMS, Typo3, Plone, Mambo, Joomla y Drupal.
- **Foros:** permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados.
- **Blogs:** permite la publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión. Ejemplo: WordPress.
- **Wikis:** sitio web dónde todos los usuarios pueden colaborar en los artículos, aportando información o reescribiéndola. También brinda un espacio para discusiones. Es muy útil para material que irá evolucionando con el tiempo como los manuales de usuarios. Ejemplos: MediaWiki, TikiWiki y MoinMoin.
- **e-Commerce:** son sitios web para comercio electrónico. Ejemplo: osCommerce.
- **Portal:** sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad. Ejemplos: Joomla, Drupal, PHP-Nuke y Plone.
- **Galería:** permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual, normalmente imágenes. Ejemplos: Gallery, Flickr, PicasaWeb o Coppermine
- **e-Learning:** sirve para la enseñanza de conocimientos. Los usuarios son los profesores y estudiantes, tienen aulas virtuales donde se pone a disposición el material del curso. Ejemplos: Moodle, Dokeos.

Según la bibliografía consultada (25) los CMS más destacados son: WordPress, Joomla y Drupal. A continuación se establece una comparación entre Joomla y Drupal, en aras de seleccionar el más adecuado para el desarrollo de la solución, ya que posee una buena

documentación, una amplia comunidad de usuarios que aportan soluciones para erradicar errores y ampliar el uso de estos. Conjuntamente están desarrollados sobre plataformas libres lo cual evita el pago de licencia para su uso y modificación. No se analiza WordPress pues está diseñado más bien para blogs.

1.6.2 Joomla 1.7

Joomla es una aplicación de código abierto programada en lenguaje PHP⁹, bajo una licencia GPL y que utiliza una base de datos MySQL para almacenar el contenido y los parámetros de configuración del sitio. Es libre y gratuito. Cuenta con una interfaz amigable con distintas secciones para manejar artículos, bloques, lista de vínculos, multimedia y contenido. Incluye un número de provisiones para hacer que las páginas carguen más rápido, incluyendo compresión cache (26).

1.6.3 Drupal 7.8

Drupal es un sistema de gestión de contenido modular muy configurable, de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. La interfaz es muy simple, con vínculos lógicamente organizados para crear y/o editar contenido nuevo o administrar usuarios y permisos. Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares en la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. Posee una cantidad enorme de módulos disponibles, para añadir funcionalidades más allá del contenido básico de administración (27).

Fundamentación de la herramienta a utilizar para el desarrollo de la solución

Se selecciona como herramienta un CMS, ya que posee una arquitectura de autenticación, autorización y de permisos buenos y sencillos de configurar, lo cual trae consigo ahorro en tiempo de diseño y desarrollo. Con un CMS el manejo de la presentación se controla mediante plantillas fácilmente personalizables. En cambio los frameworks no poseen un principio básico definido para la gestión de roles y usuarios, es decir, no brindan un punto de partida para definir de forma automática estos conceptos en el sistema (28). Con un framework se tendría que implementar lógica de negocio que se podría reutilizar a partir de un gestor de contenido. Además el uso de un framework podría influir en el rendimiento de la aplicación dependiendo de la complejidad de la tarea que se quiere resolver, cuando uno de los requerimientos que debe tener el sistema es que sea

⁹ *Hypertext Preprocessor.*

rápido, flexible y portable. Los procesos a informatizar no requieren de alta complejidad, la lógica relacionada con el negocio es de fácil entendimiento, por lo que es más adecuado un CMS.

El CMS seleccionado para el desarrollo del sistema es Drupal en su versión 7.8, pues este gestor de contenidos es de código abierto, soporta varios gestores de bases de datos como MySQL, SQLite y PostgreSQL. En cambio Joomla solo soporta a MySQL, el cual no es totalmente gratuito. Los módulos son gratuitos a diferencia de Joomla. La comunidad de Drupal es inmensa, activa y puedes obtener realimentación constante en cuanto surge alguna duda. Drupal clasifica mejor el contenido que Joomla gracias al sistema de taxonomía que posee. Además distingue el contenido en distintas categorías garantizando una estructura sólida y más organizada, cuenta con funcionalidad de metadatos, usando vocabularios controlados y publicación de XML¹⁰ de contenido. Permite autenticación mediante el *Lightweight Directory Access Protocol* (LDAP), lo que facilita a todos los usuarios del sistema autenticarse sin necesidad de crear una nueva cuenta. El sistema de roles y permisos que define Drupal, posibilita que los usuarios puedan ver la información según establece su rol, este mecanismo ofrece seguridad en cuanto al acceso a los contenidos. Su amplia variedad de módulos posibilita ampliar las funcionalidades que ofrece el mismo y añadirle otras que no trae por defecto.

1.7 Sistemas gestores de bases de datos

Un sistema gestor de bases de datos es: “*el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos*”. En estos gestores se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad e integridad de los mismos (29).

Existen diversos sistemas gestores de bases de datos. A continuación se describen las características de PostgreSQL y SQLite, ya que el CMS escogido para la solución del problema soporta por defecto estos gestores al igual que MySQL este último no se analiza pues no es totalmente libre.

¹⁰ Siglas en inglés de (*Extensible Markup Language*) lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C).

1.7.1 SQLite 3

SQLite es un proyecto de dominio público, su creador es D. Richard Hipp, el cual implementa una pequeña librería de aproximadamente 500 kb, programado en el lenguaje C, de dominio público, totalmente libre y que tiene como función hacer de un sistema de bases de datos relacional (30).

Uno de las primeras diferencias entre los motores de bases de datos convencionales es su arquitectura cliente/servidor, pues es independiente y simplemente se realizan llamadas a subrutinas o funciones de las propias librerías de SQLite, lo cual reduce ampliamente la latencia en cuanto al acceso a las bases de datos. Con lo cual podemos decir que la base de datos compuesta por la definición de las tablas, índices y los propios datos son guardados por un solo fichero estándar y en un solo ordenador. Entre las características que posee, presenta (30):

- **Tamaño:** tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.

1.7.2 PostgreSQL 8.4

PostgreSQL es un servidor de base de datos relacional libre, bajo la licencia *Berkeley Software Distribution* (BSD). Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema esto significa que: un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (31). Fue desarrollado en la Universidad de California, liderado por el Profesor Michael Stonebraker. Es pionero en los sistemas gestores de bases de datos disponibles actualmente a nivel mundial (32) (31) .

Algunas de sus principales características (32):

- **Altamente extensible:** soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.

- **Soporte SQL comprensivo:** cuenta con la más reciente especificación de SQL orientado a características avanzadas tales como las uniones (joins).
- **Integridad referencial:** presenta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **Multiplataforma:** es funcional en los principales sistemas operativos como: Linux, Unix, Mac OS, Windows.:

Fundamentación del sistema gestor de bases de datos a utilizar

Como sistema gestor de base de datos se ha seleccionado, para el desarrollo del sistema, a PostgreSQL en su versión 8.4, pues garantiza seguridad a la hora de almacenar los datos, en cambio SQLite al no poseer configuración, la seguridad y disponibilidad de la información se ven comprometidas, debido a que no existe un usuario administrador que garantice la integridad de los datos (o sea que no los elimine cualquier persona, sino el usuario que tiene ese permiso). Posee puntos de recuperación en un momento dado, sin embargo SQLite en caso de una caída del sistema no permite adoptar opciones de recuperación o sea una copia de la base de datos. Además, PostgreSQL es multiplataforma, de código abierto y posee una gran escalabilidad, es capaz de ajustarse al número de unidades de procesamiento y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

1.8 Servidor web

Un servidor web puede ser tanto un ordenador de grandes dimensiones y capacidad como un programa informático que utiliza el protocolo de comunicaciones http, para recibir las peticiones de información de un programa cliente (navegador) en el ordenador del usuario. Una vez interpretada la solicitud que el servidor/cliente le hace, (la página web que deseamos navegar), el mismo recupera la información solicitada de sus carpetas y la envía al programa cliente, (al ordenador del usuario vía su navegador), para su visualización (32).

A continuación se analizan dos servidores que son recomendados para el trabajo con Drupal: Apache, en su versión 2.2.12 y Microsoft IIS, en su versión 7.5, de acuerdo a los requerimientos dados en la bibliografía siguiente ((25) (33)).

1.8.1 Apache 2.2.14

Uno de los más ventajosos servidores web es el servidor Apache. Este servidor es software libre de código abierto para plataformas *Unix*, GNU/Linux, etcetera, que implementa el protocolo de transferencia de hipertexto (HTTP) y la noción de sitio virtual. Apache presenta mensajes de error altamente configurables y bases de datos de autenticación. De igual manera, es altamente configurable en la creación y gestión de log y negociado de contenido (34).

Algunas de sus principales ventajas son (35):

- Modular.
- *OpenSource*.
- Multi-plataforma.
- Extensible.
- Popular (fácil conseguir ayuda/suporte).
- Gratuito.

1.8.2 Microsoft IIS 7.5

Microsoft Internet Information Server (IIS), es la plataforma de servidores desarrollada por Microsoft, por lo que no es gratuita ni de código abierto, aunque la misma ha tenido un ascenso significativo por su compatibilidad con los *host*¹¹ y por sus nuevas innovaciones como *.Net*, *Silverlight*¹², entre otros. Uno de los factores más importantes sobre el ascenso de este servidor es que viene preinstalado en la mayoría de los sistemas operativos de Microsoft, por lo que fue el estándar de servidores por mucho tiempo (34).

Fundamentación del servidor web a utilizar

El servidor web seleccionado para el desarrollo del sistema es Apache, en su versión 2.2.14, pues es el más recomendado para trabajar con Drupal, ya que cada nueva versión de Drupal se prueba con el mismo (33). Es estable, confiable y altamente configurable. Puede ejecutarse en modo híbrido, multiproceso-multihebra, lo cual permite mayor escalabilidad para la mayoría de sus configuraciones. Además es multiplataforma, libre y gratuito, no siendo así para IIS.

¹¹ Utilizado a veces como sinónimo de *mainframe*, identifica al ordenador central en un sistema informático complejo.

¹² Es una estructura para aplicaciones web, agrega nuevas funciones multimedia como la reproducción de videos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace *Adobe Flash*.

1.9 Lenguajes de programación

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, que pone a disposición del programador, para que este pueda comunicarse con los dispositivos hardware y software existentes.

Los lenguajes de programación pueden clasificarse, según el paradigma que usan, en: procedimentales, orientados a objetos, funcionales, lógicos, híbridos, etc. Actualmente, existen diferentes lenguajes de programación para desarrollar en la web. Estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Los mismos se pueden separar en lenguajes del lado de servidor y lenguajes del lado del cliente (36).

En el desarrollo de aplicaciones web se utilizan lenguajes de interpretación del lado del servidor y del lado del cliente. Existen numerosos lenguajes de programación en la actualidad para el desarrollo de aplicaciones web entre los que se destacan PHP, Javascript, Python y Ruby. Seguidamente, se mencionan los lenguajes que se utilizarán para el desarrollo del sistema.

1.9.1 Lenguajes de programación del lado del cliente

Los lenguajes del lado del cliente son aquellos que se procesan antes de enviar la solicitud, mediante el protocolo HTTP, hacia el servidor web (37).

1.9.1.1 JavaScript 1.5

JavaScript es el lenguaje de programación del lado del cliente más utilizado por ser compatible con la mayoría de los navegadores modernos. También es un lenguaje interpretado que no requiere compilación, utilizado principalmente en páginas web. No es un lenguaje orientado a objetos y permite crear aplicaciones específicamente orientadas a su funcionamiento en Internet.

Con JavaScript se pueden construir aplicaciones que varían desde la gestión de la información corporativa interna y su publicación en Intranets hasta la gestión masiva de transacciones de comercio electrónico. No requiere un tiempo de compilación, los scripts se pueden desarrollar en un período de tiempo relativamente corto. Aunque este tiene muchas similitudes con Java, no incluye la sintaxis y reglas complejas de Java. Como WWW¹³ es independiente de la plataforma hardware o sistema operativo, los programas

¹³ World Wide Web.

escritos en este lenguaje también lo son, siempre y cuando exista un navegador con soporte JavaScript para la plataforma en cuestión. Los programas JavaScript tienden a ser pequeños y compactos (en comparación con los *applets* de Java), no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML, se reduce el número de accesos independientes a la red (37).

1.9.1.2 XHTML 1.1

XHTML es un ingenioso sistema de incluir información sobre el contenido justo en un documento del texto. La misma, denominada código, puede incluir instrucciones de formato al igual que detalles sobre las relaciones entre las partes del documento. Sin embargo, puesto a que el otro código consta principalmente de texto, el documento es accesible universalmente. Tiene tres tipos principales de código: elementos, atributos y valores. Ayuda a que el código sea coherente, esté bien estructurado y libre de etiquetas no estándar, lo que, a su vez, facilita actualizar y editar, así como aplicar formato con CSS (38).

1.9.1.3 CSS 3

CSS son las siglas de *Cascading Style Sheets*. Estas hojas de estilo en cascada son un lenguaje formal, usado para definir el aspecto de un documento. Está escrito en HTML y XHTML. Sirve tanto para presentarlo en la pantalla como para imprimirlo. El desarrollo de las CSS tiene como objetivo separar los contenidos de la página web de su presentación. Esta característica permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo.

Este lenguaje brinda como resultado una mejor organización del código HTML, menos peso en las páginas y más flexibilidad en los cambios. Uno de los aspectos más importantes a la hora de crear una página web es la accesibilidad.

Aplicar el principio básico de separar estructura de apariencia así como ciertas reglas sencillas para hacer tus páginas perfectamente accesibles, producirá efectos secundarios positivos.

Entre estos efectos se puede dar la posibilidad a un ciego, por ejemplo, de escuchar los contenidos. La facilidad de mantenimiento del sitio web provoca más rapidez de carga y mayor accesibilidad para las diferentes tecnologías de navegación por la Web, por ejemplo, los *robots* indexadores de los motores de búsqueda (39).

1.9.2 Lenguaje del lado del servidor

Los lenguajes del lado del servidor son aquellos que procesan los datos una vez recibidos en el servidor web mediante el protocolo HTTP (37).

PHP 5.3

PHP es un lenguaje sencillo para el desarrollo de aplicaciones web. Actualmente se utiliza por una gran cantidad de usuarios y una de las características que lo ha llevado a estos logros es que permite las técnicas Programación Orientada a Objeto (POO). Es un lenguaje de programación interpretado que posibilita la generación dinámica de contenidos en un servidor web. Entre sus características principales destacan su alto rendimiento, potencia, facilidad de aprendizaje y los pocos recursos que consume (40).

Entre las particularidades que posee este lenguaje y que lo convierten en una potente herramienta están (40):

- Es un lenguaje multiplataforma.
- Soporte sólido POO.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web.
- Es un software de código abierto.
- Soporta bases de datos entre las que se pueden mencionar: MySQL y PostgreSQL.
- Se integra con varias bibliotecas externas: permite desde generar documentos en formato .PDF hasta analizar código XML.

Con las primeras dos versiones de PHP, PHP 3 y PHP 4, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Estas versiones han servido de mucha ayuda para la comunidad de desarrolladores, haciendo posible que PHP sea el lenguaje más utilizado en la Web para la realización de páginas avanzadas.

Sin embargo, todavía existían puntos negros en el desarrollo PHP que se han tratado de solucionar con la versión 5, aspectos que se echaron en falta en la versión 4, casi desde el día de su lanzamiento. Estos están referidos principalmente a la POO (la orientación a objetos es una manera de programar que trata de modelar los procesos de programación de una manera cercana a la realidad, tratando a cada componente de un programa como un objeto con sus características y funcionalidades) que, a pesar de que estaba soportada

a partir de PHP3, solo implementaba una parte muy pequeña de las características de este tipo de programación.

Varias de las ventajas fundamentales que brinda PHP5 son (40):

- Mejor soporte para la POO, que en versiones anteriores era extremadamente rudimentaria.
- Mejoras de rendimiento.
- Mejor soporte para MySQL.
- Mejor soporte a XML.
- Soporte integrado para SOAP.
- Manejo de excepciones.

El principal objetivo de PHP5 es mejorar los mecanismos de POO, para solucionar las carencias de las versiones anteriores (37).

1.10 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (en inglés *Integrated Development Environment* o IDE) “es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código”. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos (41).

En esta sección se comparan los entornos de desarrollo Zend Studio por ser el IDE recomendado para el uso de PHP (42), NetBeans por ser de código abierto, multiplataforma y gratuito, que permite la utilización de varios lenguajes, incluido PHP, además Geany por consumir pocos recursos.

1.10.1 Zend Studio 8.0.1

Zend Studio es un IDE para PHP orientado a desarrollar aplicaciones web que proporciona ayudas para la creación y gestión de proyectos. Por lo tanto, soporta HTML, XML, CSS, JavaScript así como una integración directa con Zend Framework y PHP Unit. Está escrito en Java, por lo tanto es multiplataforma. Viene por defecto con un depurador de código (simple) pero para proyectos avanzados es necesario Zend Server. Consta de un explorador de archivos, un depurador, y el propio editor de código. El mismo es privado (43).

1.10.2 NetBeans 6.9.1

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe, además, un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso (44).

La interfaz de desarrollo brinda la opción de programar en distintos lenguajes, es agradable y sobre todo gratuita. Igualmente, se le pueden adoptar innumerables librerías y *plugins*¹⁴, los cuales brindan muchas opciones al momento de programar (44).

1.10.3 Geany 0.18

Es un IDE pequeño y rápido, con una arquitectura basada en complementos (plugins), para programadores y desarrolladores web, disponible tanto para Windows como Linux. Dispone de las funciones básicas de un editor de estas características. Otras características que posee son: resaltado de sintaxis, cierre de códigos, autocompletados de etiquetas en XML y HTML y soporte de los lenguajes C, Java, PHP, HTML, Python, Perl, PASCAL. Dispone de versiones para Windows, Debian y Fedora Core (45).

Fundamentación del entorno de desarrollo integrado a utilizar

El IDE seleccionado es Geany en su versión 0.18 por su bajo consumo de recursos. Luego de hacerse un estudio en una Mini laptop con un 1 Gb de memoria RAM y un procesador a 1.6 GHz(en el laboratorio 103 del docente 5), se pudo evidenciar que mantenía un consumo de 5 a 10 Mb (Ver anexo 15 y anexo 16). En cambio, el consumo de Netbeans osciló entre 86 y 90 Mb, lo que indica un mayor consumo de recursos del sistema. Geany es gratuito a diferencia de Zend Studio. Además es multiplataforma, cuenta con una interfaz sencilla y legible, requiere poco espacio en disco duro, tiene búsqueda y reemplazo, explorador de ficheros, variables y funciones, autocompletado usando etiquetas de Drupal y soporte multidocumento.

1.11 Conclusiones parciales

El uso de los métodos científicos ha sido la estrategia para desarrollar la teoría que fortalece esta investigación, dando la posibilidad de construir los conceptos que permiten comprender el problema de investigación y llegar a un análisis minucioso para la elección de las herramientas y tecnologías más usadas para el desarrollo de un Sistema de

¹⁴En castellano se le llama complemento. Un *plug-in* es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

Gestión de Evidencias, así como la metodología de desarrollo adecuada para guiar el proceso de desarrollo. La selección de las mismas ayudará a dar cumplimiento al objetivo general y será el resultado de los estudios que se muestran.

Capítulo 2. Propuesta de solución

Introducción

Uno de los pasos fundamentales de la metodología XP es la fase de diseño. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado en el proyecto.

Kent Beck dice que, en cualquier momento, el diseño adecuado para el software es aquel que *“supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de métodos y clases (46).”*

Cada proyecto XP debe estar guiado por una metáfora (47). La misma, según Martin Fowler, consiste en *“formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema (48).”*

Paralelamente con la metáfora, dicha metodología propone el uso de otras prácticas como son las “Historias de Usuarios (HU)” y las “Tarjetas CRC”. En el presente capítulo se abordarán temas relacionados con las características y funcionamiento del sistema describiendo las dos primeras fases de la metodología XP, Planificación y Diseño, detallando cada uno de los artefactos que se generan en las mismas.

2.1 Descripción del proceso de gestión de evidencias

Las evidencias que aportan los profesores de la Facultad 4 a los indicadores a las ARC de Investigación y Postgrado son recopiladas de forma manual. Este proceso se inicia cuando un profesor desea dejar constancia de su desempeño para lo cual debe escanear la evidencia correspondiente y hacerla llegar al responsable de Investigación y Postgrado bien sea de forma directa o mediante el empleo del correo electrónico. Una vez recepcionada la evidencia esta pasa a un proceso de revisión para constatar su veracidad y ajuste al período que se analiza (normalmente año fiscal) donde además se asigna un estado a la misma que puede ser: pendiente, revisada sin aprobar, aprobada o rechazada. El estado de pendiente indica que aún no se ha hecho una revisión, el de revisada sin aprobar significa que tiene errores y debe modificarse para ser aprobada, el estado de rechazada indica que se detectaron errores durante la revisión y requiere la eliminación y el estado de aprobada indica que la misma está lista para su uso en reportes posteriores. Esta etapa es realizada siempre por el responsable de Investigación y Postgrado y los contenidos procesados se van archivando de manera progresiva en

carpetas especializadas por tipo de evidencia y año de creación. De manera gráfica el proceso descrito puede observarse en la figura 3.



Figura 3: Proceso de gestión de evidencias

2.2 Objetivos del sistema

Como paso previo al desarrollo de un producto de software se debe definir con exactitud los objetivos que se persiguen con el mismo. Estos quedan definidos de manera general en los aspectos siguientes:

1. Ampliar los espacios de divulgación sobre temas relacionados con las ARC de Investigación y Postgrado en la comunidad profesoral de la Facultad 4.
2. Ampliar los espacios de intercambio entre los profesores sobre temas relacionados con las ARC, para así conocer su opinión acerca de las diversas temáticas que se ofrecen y que ellos puedan ofrecer.
3. Agilizar y hacer más confortables los procesos relacionados con el control de evidencias de lo profesores así como la generación del balance de ciencia y técnica de la Facultad 4.

2.3 Definición de la audiencia

La audiencia es el público al que va dirigido este producto, estructurado por características necesarias que facilitan el trabajo de sus usuarios. El público que va a

acceder a este producto es la comunidad de profesores de la Facultad 4, además de los usuarios de la comunidad universitaria.

Es importante mencionar que las necesidades informativas de esta solución informática están orientadas a las actividades correspondientes a la Investigación y el Postgrado de la facultad, así como noticias científicas y de investigación.

2.4 Descripción del sistema

Atendiendo al proceso descrito en el epígrafe 2.1, se decide informatizar la gestión de evidencias de los profesores obtenidas por su participación y desempeño en eventos, publicaciones entre otros aspectos relacionados. Para ello se propone desarrollar una aplicación web que contenga avisos y noticias para mantener informados a todos los usuarios, también debe contar con espacio de intercambio entre usuarios mediante foros. Además posibilitará la creación, modificación, listado y eliminación de evidencias. Para validar que las evidencias sean veraces el usuario con los permisos correspondientes podrá cambiar el estado de las evidencias. Para generar reportes el sistema debe verificar que las evidencias tengan el estado de aprobada. El responsable de la actividad generará el balance general de ciencia y técnica por centro, departamento así como por año y adicionalmente creará temas en los foros, noticias, avisos e incluirá documentación de consulta actualizada. Todo reporte generado deberá poderse imprimir y exportarse a formato PDF.

Este sistema, de manera general, cumple con la definición dada de SGD de acuerdo con la norma ISO 15489 aunque no contiene todos sus aspectos (e.g. control de versiones para la gestión de las evidencias o el mantenimiento coherente de la información procedente de diferentes compañías u organizaciones). Como contraparte a esto el sistema realiza operaciones propias de la GD como el manejo de grandes volúmenes de documentos, la aplicación de un flujo documental para las evidencias, el manejo del control y el acceso a las evidencias así como la archivación de las mismas.

2.5 Usuarios relacionados con el sistema

Para un mejor entendimiento de la solución, se dará a conocer los usuarios que estarán relacionados de una forma u otra con el sistema informático.

Tabla 1: Usuarios relacionados con el sistema

Usuarios	Descripción
-----------------	--------------------

Administrador	Realiza la gestión de contenido del sistema y permisos de usuarios. Cuenta con posibilidades ilimitadas para ejecutar todas las funciones administrativas del sistema. Necesariamente debe autenticarse, de lo contrario navega como un usuario anónimo.
Jefe de departamento	Es el encargado de generar los reportes de su departamento. Navega como un usuario autenticado.
Jefe de postgrado	Es el encargado de generar todo los reportes del sistema. Incorporar la documentación y agregar un tema en el foro. Navega como un usuario autenticado.
Usuario autenticado	Accede a toda la información básica del sistema, puede gestionar sus evidencias.
Usuario anónimo	Personas interesadas en ver documentación publicada, sin necesidad de autenticarse en el sistema informático.

2.6 Historias de usuario

El sistema es desarrollado para el cliente. Por tanto, el usuario es quien decide que tareas realizará la aplicación. Las HU son utilizadas como herramientas para dar a conocer los requerimientos funcionales y no funcionales del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que ejecutará el sistema. La redacción de los mismos es hecha bajo la terminología del cliente, no del desarrollador, en un lenguaje que sea claro, sencillo y sin profundizar en detalles (19).

Se puede considerar que las HU, en la metodología XP, juegan un papel similar a los casos de uso en otras metodologías, pero son muy diferentes: sólo muestran la silueta de una tarea a realizarse. Por esta razón, es fundamental que el usuario o un representante del mismo se encuentren disponibles en cualquier momento para solucionar dudas. Las HU no proporcionan información detallada acerca de actividad específica.

Las HU también son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas. En una entrega se puede desarrollar una o más HU. Esto depende del tiempo que demore la implementación de cada una (35).

Una HU está compuesta por: el **número** que la identifica, el **nombre**, los **usuarios** que van a interactuar con la aplicación, los **puntos de estimación** que son las semanas que durará la implementación de dicha HU, la **prioridad** que va a tener para el negocio, la iteración en que será implementada, los **riesgos** en el desarrollo y una breve **descripción** de lo que realizará la misma.

En la presente investigación, a juicio de los autores, se asume un formato para las HU que recogen los requisitos no funcionales del sistema, el mismo está conformado por el **número** que la identifica, el **nombre** y la **descripción**.

Seguidamente, se presentan las principales HU del sistema. El resto se encuentra en: (ver anexos del 1- 6), al igual que todas las que recogen los requisitos no funcionales (ver anexos del 7- 14).

Tabla 2: HU Gestionar evidencia

Historia de Usuario	
Número de HU: 1	Nombre: Gestionar evidencia
Usuarios: Usuario autenticado	
Puntos de estimación: 1	Iteración asignada: 1
Prioridad del negocio: Alta	Riesgos en el desarrollo: Media
Descripción: El usuario puede crear, modificar, eliminar y consultar su evidencia.	
Observaciones: Todos los usuarios deben estar autenticados.	

Tabla 3: HU Gestionar rol

Historia de Usuario	
Número de HU: 2	Nombre: Gestionar rol
Usuarios: Administrador	
Puntos de estimación: 1	Iteración asignada: 1
Prioridad del negocio: Media	Riesgos en el desarrollo: Baja
Descripción: Permite administrar todas la cuentas de usuario existente en el sistema. A	

cada usuario se le asignará el rol que desempeñará dentro de la aplicación.
Observaciones:

Tabla 4: HU Listar evidencias

Historia de Usuario	
Número de HU: 3	Nombre: Listar evidencias
Usuarios: Usuario autenticado, Jefe de postgrado, Jefe de departamento	
Puntos de estimación: 1	Iteración asignada: 1
Prioridad del negocio: Alta	Riesgos en el desarrollo: Media
Descripción: Permite listar todas las evidencias que hay en la aplicación. Los usuarios la pueden ver según sus permisos .	
Observaciones:	

Tabla 5: HU Gestionar documento

Historia de Usuario	
Número de HU: 4	Nombre: Gestionar documento
Usuarios: Usuario autenticado, Jefe de postgrado	
Puntos de estimación: 0.5	Iteración asignada: 2
Prioridad del negocio: Media	Riesgos en el desarrollo: Media
Descripción: El Jefe de Postgrado podrá publicar y eliminar documentos de interés para la superación de los profesores de la Facultad 4. El usuario autenticado podrá descargar o consultar la fuente que sea de su interés.	
Observaciones:	

Tabla 6: HU Generar el balance general

Historia de Usuario	
Número de HU: 5	Nombre: Generar balance general
Usuarios: Jefe de postgrado	
Puntos de estimación: 1	Iteración asignada: 2
Prioridad del negocio: Alta	Riesgos en el desarrollo: Media

Descripción: Permite generar el balance general de las CTI.
Observaciones:

Tabla 7: HU Generar reportes

Historia de Usuario	
Número de HU: 6	Nombre: Generar reportes
Usuarios: Jefe de departamento, Jefe de postgrado	
Puntos de estimación: 2	Iteración asignada: 2
Prioridad del negocio: alta	Riesgos en el desarrollo: media
Descripción: Permite generar reportes de las estadísticas por: departamentos, centro FORTES ¹⁵ , fecha, trimestre, semestre, mensual, anual, autor (permite seleccionar el usuario y/o la categoría de autor), tipo de evidencia y evento (permite seleccionar el nombre del evento y/o alcance).	
Observaciones:	

2.7 Estimación de esfuerzo por HU

Después de definir las HU, los desarrolladores hacen una planificación del tiempo que necesitan para implementarlas y así determinar el tiempo total para el desarrollo de la aplicación.

Tabla 8: Estimación de esfuerzo por HU

Historias de usuarios	Estimación (semanas)
Autenticar usuario	1
Gestionar rol	1
Gestionar evidencia	1
Listar evidencias	1
Gestionar noticia	0.5
Gestionar documento	0.5
Generar reporte	2
Generar balance general	1

¹⁵ Centro de Tecnologías para la Formación.

Exportar a PDF	1
Gestionar tema del foro	0.5
Vistazo	0.5
Realizar búsquedas	1
Gestionar perfil de usuario	1
Total	12

2.8 Iteraciones

Luego de haber realizado las HU y una estimación previa de esfuerzo, se procede a realizar la planificación de la implementación del sistema, especificando la prioridad con que se irán implementando las HU, organizadas por iteraciones así como posibles fechas de liberación.

Una iteración no es más que una pequeña versión del proyecto que se muestra como resultado, con un valor para el cliente. La versión completa del mismo no se obtiene hasta que no concluyan todas las iteraciones.

Iteración 1:

En esta iteración se dará cumplimiento de implementación a las historias de usuario de mayor prioridad. Al concluir esta iteración, se obtendrá el resultado algunas de las funcionalidades básicas del sistema.

Iteración 2:

En esta iteración se continuará dando cumplimiento a las restantes HU de mayor prioridad y alguna de prioridad media. Con la implementación de las mismas, se obtendrán los resultados de las funcionalidades que cubren las necesidades especificadas por el cliente.

Iteración 3:

En esta iteración se implementan las funcionalidades con prioridad baja. Las mismas brindan al cliente la comodidad en la gestión de otras tareas asociadas a las de mayor prioridad.

2.8.1 Plan de duración de las iteraciones

Según define la metodología XP se crea el plan de duración de las iteraciones. En el mismo se especifican más detalladamente el orden de las HU dentro de cada iteración así como la iteración completa de la misma.

Tabla 9: Plan de duración de iteraciones

Iteraciones	Historias de Usuarios a implementar	Duración
1	Autenticar usuario Gestionar rol Gestionar evidencia Listar evidencias	4 semanas
2	Generar reporte Generar el balance general Gestionar documento	3.5 semanas
3	Gestionar noticia Exportar a PDF Gestionar tema del foro Gestionar perfil de usuario Vistazo Realizar búsquedas	4.5 semanas

2.9 Plan de entregas

El cronograma de entrega establece cuáles son las HU que serán agrupadas para conformar las entregas y el orden de las mismas. Este cronograma es resultado de una reunión con todos los actores del proyecto.

Tabla 10: Plan de duración de las entregas

Historia de Usuario	Fin 1ra Iteración 3ra Semana de febrero	Fin 2da iteración 1ra Semana de Abril	Fin 3ra Iteración 4ta Semana de Abril
Autenticar usuario	Versión 1.0	Finalizado	Finalizado
Gestionar rol	Versión 1.0	Finalizado	Finalizado
Gestionar evidencia	Versión 1.0	Finalizado	Finalizado
Listar evidencias	Versión 1.0	Finalizado	Finalizado
Generar reporte	-	Versión 1.0	Finalizado
Generar balance general	-	Versión 1.0	Finalizado
Gestionar documento	-	Versión 1.0	Finalizado
Gestionar noticia	-	-	Versión 1.0
Exportar a PDF	-	-	Versión 1.0
Gestionar tema del foro	-	-	Versión 1.0

Vistazo	-		Versión 1.0
Realizar búsquedas	-	-	Versión 1.0
Gestionar perfil de usuario	-	-	Versión 1.0

2.10 Metáfora

Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto y guiar la estructura y arquitectura del mismo. Por ejemplo, puede ser una guía para la nomenclatura de los métodos y las clases utilizadas en el diseño del código. Tener nombres claros, que no requieran de mayores explicaciones, redundará en un ahorro de tiempo. Es muy importante que el cliente y el grupo de desarrolladores estén de acuerdo y compartan esta “metáfora”, para que puedan dialogar en un “mismo idioma”. Una buena metáfora debe ser fácil de comprender para el cliente y, a su vez, debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto (49).

En la presente investigación se definió como metáfora: “Un portafolio donde se crean, archivan, consultan evidencias y se generan reportes”.

2.11 Tarjetas CRC

Las tarjetas CRC (*Class, Responsibilities and Collaboration*) trabajan con la técnica de modelado basada en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades (49). Las tarjetas están compuestas por el **nombre de la clase** colocado como **título**, en la parte izquierda se colocan las **responsabilidades** (funcionalidades) y en la parte derecha las **clases** que se implican en cada responsabilidad.

La creación de las mismas tiene gran utilidad, pues da una idea clara de la arquitectura del sistema, distribución de las clases, paquetes y la ubicación de las diferentes responsabilidades sobre la lógica del negocio (49).

Drupal no maneja la POO, pero sí usa algunos conceptos de la misma como son: objeto, herencia, polimorfismo, encapsulamiento y abstracción. Por tanto, en esta investigación se asumirá que cada tarjeta CRC es un módulo, sus responsabilidades son las funciones que realiza y sus colaboradores los métodos que se implementan en el módulo. A continuación se muestra la tarjeta CRC que representa el módulo creado.

Tabla 11: Tarjeta CRC Generar reportes

Tarjeta CRC 1 Generar reporte

Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Generar reporte de los departamentos • Generar reporte del centro FORTES • Generar reporte personal • Generar reporte por fecha • Generar reporte por trimestre • Generar reporte por semestre • Generar reporte por año • Generar reporte mensual • Generar Balance General • Generar reporte por premios • Generar reportes por publicaciones • Generar reportes por trabajos presentados • Generar reportes 	<ul style="list-style-type: none"> • function reportes_generar_reportes_form() • function reportes_generar_reportes_form_validate() • function reportes_generar_reportes_form_submit() • function reportes_reporte_personal_form() • function reportes_reporte_personal_form_validate() • function reportes_reporte_personal_form_submit() • function reportes_resultado() • function reportes_generar_pdf()

<p>por capacitación recibida</p> <ul style="list-style-type: none"> • Generar reportes por capacitación ofertada • Generar reportes por proyecto de I+D • Generar reportes por patentes y registros • Generar reportes por Resultados Introducidos • Generar reportes por empleo de estudiantes 	
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

2.12 Conclusiones parciales

La metodología de desarrollo seleccionada para guiar el proceso de software posibilitó definir una metáfora, que sirve de guía para la arquitectura del sistema. Se describieron los usuarios que interactuarán con el sistema conociendo así la función que cumple cada uno, para garantizar la seguridad de acceso al contenido. En conjunto con el cliente se definieron las HU para la especificación funcional, en un lenguaje del cliente, facilitando la comunicación del programador con el mismo a la hora de implementarlas, teniendo en cuenta, según el plan de iteraciones, el orden de su implementación. Se estimó el tiempo que tomará llevar a cabo el desarrollo de la misma, considerándose que debe estar listo en el tiempo establecido. Además, permitió la construcción de una tarjeta CRC que traduce los requisitos en entidades (módulos), dando paso a la fase de codificación.

Capítulo 3. Implementación y prueba

Introducción

En XP, desde un principio, se inicia la codificación favoreciendo el logro del objetivo de estar haciendo entregas frecuentes al cliente ((19) (13)).

Alguno de los elementos más importantes en cuanto a la codificación son: que el cliente siempre debe estar presente, se debe trabajar en pareja y debe haber una propiedad colectiva del código ((49) (18)). Todos estos elementos representan paradigmas nuevos en lo que a la ingeniería de software se refiere, planteando un entorno de discusión sobre la conveniencia de adoptarlas.

En el presente capítulo, se detallan conceptos y temas importantes sobre Drupal que fueron de gran utilidad para la construcción del Sistema de Gestión de Evidencia, centrándose en las fases de Codificación y Prueba de la metodología XP, describiendo el tema usado y los módulos desarrollados., además de las pruebas de aceptación efectuadas sobre el sistema.

3.1 Tema

Drupal tiene mucho cuidado de mantener los elementos del contenido y la presentación de un sitio independiente, haciendo uso del patrón MVC. Esto le permite llegar a los diseños múltiples para el mismo contenido. Los mismos son denominados temas. Cada tema puede constar de uno o más archivos que trabajan juntos, para presentar el contenido que se guarda en la base de datos de una manera única (50).

Los temas son los que definen el aspecto visual de la web. Reciben el contenido a mostrar en una página y le dan formato, incluyendo, entre otras cosas, los colores, los tipos de letra y la ubicación del contenido mostrado en las páginas (51). En la práctica son un conjunto de ficheros que se encuentran en el directorio (**themes**) del servidor, y que incluyen plantillas con la estructura del sitio, hojas de estilo e imágenes. Mediante las plantillas es posible sobrescribir o definir el formato visual de regiones, bloques, comentarios e incluso campos. Se pueden utilizar temas ya existentes, realizarles personalizaciones si lo desea o crear uno nuevo definiendo las regiones, bloques y todos los elementos necesarios.

3.1.1 Tema SGEIP

Para implementar el tema “SGEIP” se utilizó una plantilla que se ajustaba bien al diseño deseado por el cliente, se siguió la filosofía de Drupal crear temas, se copió con el nombre de “SGEIP” al directorio de temas **sites/all/themes** con el mismo nombre **sites/all/themes/SGEIP** y se empezó a modificar. Utilizando CSS e imágenes para el estilo. Para darle formato al contenido, Drupal cuenta con el motor de temas phptemplate por defecto. Este utiliza plantillas para definir la estructura global de las páginas y de ciertos tipos de contenido en particular (52). Las plantillas son ficheros de texto con extensión **.tpl.php** que definen los elementos que deben aparecer en las páginas y su ubicación dentro de las mismas (53). El número de ficheros totales necesarios para un tema dependerá de lo que se quiera conseguir. A continuación se mencionarán los ficheros más importantes que contiene el tema:

html.tpl.php: plantilla principal que tiene el diseño del sistema.

page.tpl.php: plantilla que define todo entre las etiquetas <body>, donde se construye la estructura de la página mediante el uso de etiquetas <div>.

node.tpl.php: plantilla de los nodos. Se utiliza para crear la estructura visual que tendrán los contenidos.

comment.tpl.php: plantilla que corresponde al módulo de comentarios. Define la estructura de los comentarios para los contenidos.

style.css: este fichero es una hoja de estilos, en él se incluyeron todos los identificadores y clases con sus atributos, utilizados en el tema.

3.1.2 Instalación del tema SGEIP

Para instalar el tema “SGEIP” en Drupal fue suficiente con crear un subdirectorio con el nombre “SGEIP” en el directorio (**themes**), copiar dentro los ficheros de plantillas y hojas de estilo. Para que la web lo utilice, hubo que dirigirse al menú **administrar->temas** seleccionar la opción activar, seleccionándolo como tema por defecto.

Drupal cuenta con un gran número de módulos que pretenden extender sus funcionalidades. Los mismos se pueden clasificar en: módulos del núcleo y módulos contribuidos. En el siguiente subepígrafe se mencionan los módulos que se utilizaron para la solución.

3.2 Módulos

Los módulos en Drupal permiten extender la funcionalidad de la web a través de la unión de varias funciones escritas en PHP. Estas funciones actúan como enganche a la hora de construir una página Web y gestionar el contenido. Los módulos para reducir la carga del servidor permiten la opción de ser desactivados temporalmente. Algunos módulos vienen con la instalación de Drupal; mientras, otros pueden ser instalados y descargados de los sitios (52).

Para realizar la instalación de nuevos módulos se deben tener los conocimientos necesarios para acceder al servidor y copiar los ficheros del módulo en el directorio **sites/all/modules**. Su complejidad depende de los requerimientos de cada módulo en particular, siendo por lo general muy sencillo. La activación del módulo es mediante el vínculo módulo, que brinda el menú con los privilegios administrativos en la parte superior del sistema.

3.2.1 Módulos de Drupal

La versión 7.8 de Drupal tiene una amplia variedad de módulos, que facilitan el funcionamiento del sistema. Para el desarrollo del mismo se utilizaron los siguientes módulos personalizables.

1. **forum**: proporciona foros de discusión con temas propuestos por el administrador del sistema.
2. **search**: mediante la activación de este módulo se realizan las búsquedas avanzadas o básicas en cualquier página del sistema.

3.2.2 Módulos contribuidos de Drupal

Una de las ventajas de Drupal son las contribuciones de la comunidad ya que las mismas promueven e incrementan la diversidad y las características de cualquier proyecto y además reducen la carga de desarrollo para los creadores de sitios web. Esto significa que se puede utilizar una variedad de módulos que se han hecho por alguien o mejorado por otra persona y elegir la mejor. Para poder desarrollar una solución que cumpliera con todos los requisitos especificados, se necesitó el apoyo de varios módulos.

Estos módulos son:

1. **SMTP**: este módulo sirve de soporte al protocolo que lleva el mismo nombre, el cual se utiliza para el envío de notificaciones por correo electrónico.
2. **LDAP**: este módulo sirve de soporte al protocolo que lleva el mismo nombre, el cual se utiliza para garantizar la autenticación de los usuarios en la aplicación web.
3. **Date**: este módulo se utiliza como soporte para facilitar la inserción de fechas en la aplicación.
4. **CCK**¹⁶: herramienta para la construcción de campos de contenido. Añade características a este módulo que ya trae incluido el núcleo de Drupal.
5. **Views**: sirve para configurar los listados de contenidos.
6. **Workflow**¹⁷: permite la creación y asignación de los flujos de trabajo arbitrarios a los tipos de nodos. En la solución se utiliza para establecer los estados de aprobada o rechazada al tipo de contenido Evidencia.

3.2.3 Paquetes de módulos del sistema

Generar reportes

Mediante este módulo se generarán todos los reportes de la actividad postgraduada de los profesores de la Facultad 4. Se obtienen reportes estadísticos en formato PDF, a nivel de: departamento, centro, facultad, por fechas y balance general de las CTI. Para el desarrollo de este módulo se utilizó para exportar, en formato PDF, la librería FPDF.

3.2.4 Instalación de módulos

Lo primero que se debe hacer es elegir un nombre para el módulo "reportes". A continuación, se almacena con el mismo nombre en el directorio **sites/all/modules/reportes**. El primer archivo que se creará para el nuevo módulo es el archivo de reportes.info. Cada módulo en Drupal 7 debe tener un archivo **.info**. En ese archivo se guarda el nombre del módulo, la descripción, la versión, el paquete al que pertenece. La activación del módulo es mediante el vínculo módulo que brinda el menú con los privilegios **administrar -> módulo->activado** ((52) (51)).

¹⁶ Content Construction Kit.

¹⁷ Flujo de trabajo.

3.3 Configuración

Después de realizarse la instalación del tema, es necesario hacer varios arreglos en las opciones de configuración.

- Registrar la información del sistema.
- Establecer el español como idioma.
- Fijar la disposición de los bloques y los menús.
- Crear roles.

De estas opciones de configuración, algunas que se pueden hacer a través de la interfaz gráfica de Drupal y otras requieren de más complejidad. En los siguientes subepígrafes se explicarán algunas de las más complicadas.

3.3.1 Menús

Existen tres menús estándar en Drupal: Enlaces primarios, Enlaces secundarios y Menú de navegación. Los primarios y secundarios son construidos a criterio de los administradores y mostrados automáticamente. También se pueden crear menús personalizados mediante la creación de bloques ((54) (51)). El tema SGEIP cuenta con 3 menús. El **menú usuario** tiene como enlaces: Mi cuenta y Cerrar sesión. El **menú principal** tiene como enlace: Inicio, Añadir evidencia, Añadir noticias, Balance general, Generar reportes, Subir documento y la documentación. El **menú de navegación** tiene como enlaces: Añadir evidencia, Añadir noticias, Buscar, Foros, Generar balance general, Generar reportes y Solicitud de eliminación de evidencia.

3.3.2 Bloques

Los bloques son fragmentos de texto o la funcionalidad que se pueden colocar en las regiones definidas en la plantilla. Los bloques pueden ser cualquier cosa de un solo nodo, una lista de nodos, un calendario, un video, una forma, una encuesta en línea, etc. Ellos se definen dentro de los módulos mediante **hook_block_info ()**, y un módulo puede aplicar múltiples bloques dentro de este anzuelo simple.

Una vez que un bloque se define, se muestra la administración del mismo ((52) (50)). Los bloques pueden mostrarse en diferentes zonas de la página, que se denominan regiones. Los bloques pueden mostrarse en todas o algunas páginas, y/o solo a los usuarios de ciertos roles, y/o solo en las páginas que muestren contenidos de un tipo concreto

(páginas, artículos,...), por defecto los bloques son lo único cuya visibilidad podemos configurar en función de los roles.

La visualización de los menús depende de su bloque asociado. Al crear un menú se crea automáticamente un bloque, aunque inicialmente está desactivado. **Apariencia->Temas:** Las regiones posibles, sus posiciones, nombres y demás características están determinadas por el tema asignado al sitio. Para descubrir el nombre de las regiones en el tema activo seleccionaremos **Estructura->Bloques->Demostrar regiones de bloques.**

El tema SGEIP define 26 regiones. Las regiones principales son Menú, *Top 1*, *Top 2*, *Top 3*, *User 1*, *User 2*, *Contenido*, *User 3*, *User 4*, *Bottom 1*, *Bottom 2*, *Bottom 3*. En la región Menú: se sitúa el **bloque menú principal**. Región *Top 1*: se sitúa el **bloque user menu**. Región *Top 2*: se sitúa el **bloque destacados en el mes**. Región *Top 3*: se sitúa el **bloque formulario de búsqueda**. Región *User 2*: se sitúa el **bloque navegación**. Región *User 1*: se sitúa el **bloque contenido reciente**. Región *User 3*: se sitúa el **bloque comentarios recientes**. Región *User 4*: se sitúa el **bloque quién está conectado**. Región *Bottom 1*: se sitúa el **bloque enlaces externos**. Región contenido: se sitúa el **bloque contenido de la página principal**.

3.3.3 Roles de usuario

El sistema dispondrá con varios roles que varían según el tipo de usuario, estos roles poseen distintos privilegios de acceso al contenido. Drupal permite agrupar aquellos usuarios que comparten niveles de acceso o de edición, es decir, aquellos con los mismos permisos, bajo un rol. Sobre estos roles se debe definir los permisos que tendrán los usuarios pertenecientes a cada rol.

Inicialmente, Drupal tiene tres roles: usuario anónimo, usuario autenticado y administrador **Personas->Permisos->Permisos:** Drupal nos permite configurar los permisos de cada rol para cada módulo. Cada usuario puede tener asignado uno o más roles; cada rol posee una configuración específica de permisos sobre los módulos. Consecuentemente, los roles de un usuario determinan qué permisos tiene en el sistema. Por defecto, los permisos determinan qué se puede hacer, pero no qué se puede ver. A continuación se muestran los roles definidos para la aplicación (Ver Figura 6), estos roles están organizados desde el que más privilegios posee hasta el de menos privilegios.

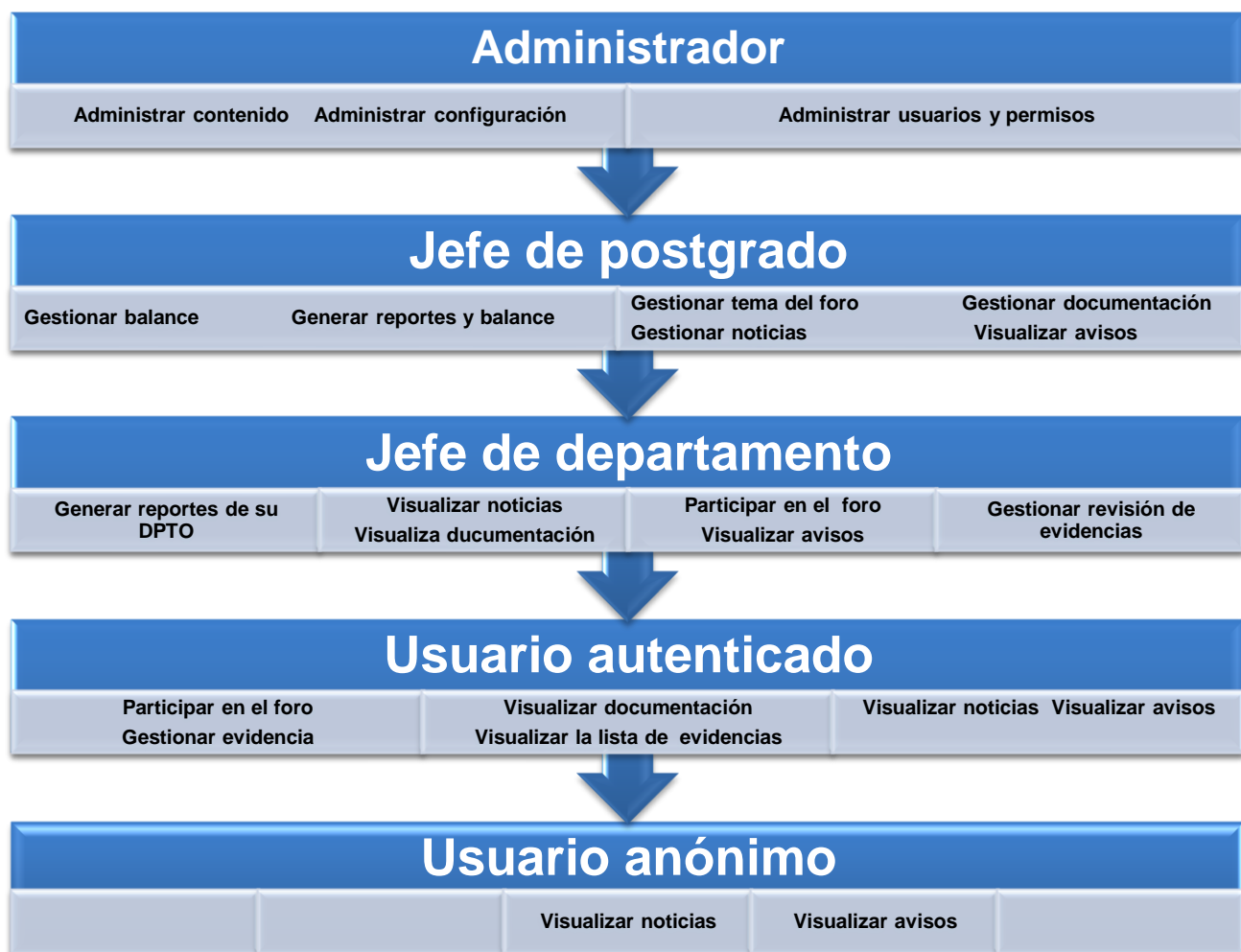


Figura 6: Roles y permisos

Administrar contenido: este permiso administra todos los contenidos del sistema, estos son Foro, Aviso, Noticia, Evidencia y Documento.

3.4 Nodo

La reconocida flexibilidad y facilidad en la creación de aplicaciones web que caracteriza a Drupal 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: los nodos.

El nodo es el bloque de construcción básico de Drupal donde se almacena información. Durante la planificación de una aplicación web es necesario saber con qué tipos de contenido se va a trabajar, cada uno de esos será un tipo diferente de nodo. Todos los nodos comparten una serie de propiedades básicas:

- Autor.
- Fecha de creación.
- Título.
- Contenido.

3.5 Tipos de contenido

Aquellos objetos de la realidad que presentan características comunes están representados en Drupal con los distintos tipos de contenido. Cada tipo tiene unas propiedades que lo definen, que son las mismas que caracterizan al objeto en el mundo real. Cada instancia de un tipo de contenido es un nodo en Drupal.

Para la creación de los tipos de contenido necesarios y para la definición de sus características, es necesario utilizar un módulo de Drupal, llamado CCK. Este módulo permite añadir nuevos tipos de contenido y definir sus propiedades mediante la adición de nuevos campos. Cada campo lleva asociado un tipo de dato que son los valores que puede tomar, algunos de estos tipos son: número, fecha, texto, etc.; aunque los tipos crecen con la adición de nuevos módulos. A partir de la implantación se crearon dos tipos de contenidos lo cuales generaron las tablas adicionales del sistema, así como los atributos asociados a estos tipos de contenidos.

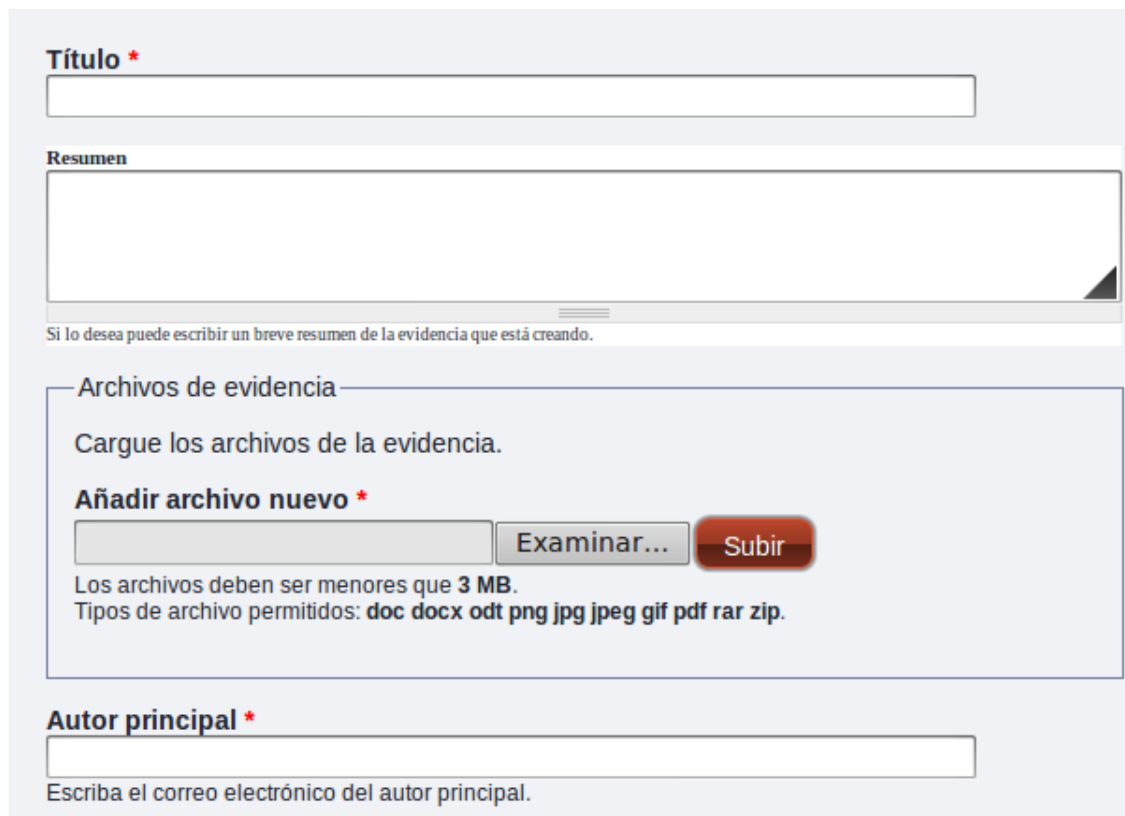
Evidencia

La evidencia es el tipo de contenido que constituyen los indicadores científicos de cada profesor. Esta se clasifica mediante tipos de evidencia, los cuales son: premios, publicaciones, trabajos presentados, capacitaciones, proyectos de investigación y desarrollo, resultados introducidos y empleo de estudiantes; de acuerdo a los indicadores establecidos por la UCI.

Cada tipo de evidencia ofrece una ramificación específica, por ejemplo, el campo "Premios" tiene un tipo específico de premio que puede ser: otorgado por la Academia de Ciencias de Cuba (ACC), entre otros. En el caso del campo "Capacitación", las mismas se pueden clasificar en "Recibida" u "Ofertada", siendo obligatorio especificar el tipo de "Recibida" u "Ofertada" (Ver anexo 17) el mismo es un balance general donde se puede observar como son clasificados los tipos de evidencia mencionados en el párrafo anterior.

A continuación se muestra la interfaz para la creación del tipo de contenido Evidencia.

Crear evidencia



Título *

Resumen

Si lo desea puede escribir un breve resumen de la evidencia que está creando.

Archivos de evidencia

Cargue los archivos de la evidencia.

Añadir archivo nuevo *

Los archivos deben ser menores que **3 MB**.
Tipos de archivo permitidos: **doc docx odt png jpg jpeg gif pdf rar zip**.

Autor principal *

Escriba el correo electrónico del autor principal.

Figura 4: Crear evidencia paso 1

Documento

Este tipo de contenido ofrece la posibilidad de que el usuario con el rol Jefe de Postgrado, pueda subir documentos de interés para los profesores de la facultad. A continuación se muestran los campos a llenar para la creación del tipo de contenido Documento.

Crear documento

Título *

Archivo

Añadir archivo nuevo *

Los archivos deben ser menores que **16 MB**.
Tipos de archivo permitidos: **doc ppt pps pdf ods odp**.

Figura 5: Crear documento

3.6 Pruebas

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 ((18) (19) (55)).

3.6.1 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de la metodología XP (49). Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados.

Uno de los elementos más importantes en estas es que deben ser construidas antes que los métodos mismos, permitiéndole al programador tener amplia claridad sobre lo que va a programar antes de hacerlo, así como conocer cada uno de los casos de pruebas que deberá pasar, lo que optimizará su trabajo y su código será de mejor calidad.

3.6.2 Pruebas de aceptación

Las pruebas de aceptación se utilizan para mostrar que las características que el cliente deseaba están funcionando correctamente. Las mismas son consideradas pruebas de caja negra ((13) (55)). Estas marcan el camino a seguir indicando las funcionalidades más importantes, permiten que el cliente sepa cuándo el sistema informático está funcionando y que los programadores conozcan qué les resta por hacer.

Estas pruebas se crean a partir de las HU. Una HU puede tener una o más pruebas de aceptación con el objetivo de asegurar su buen funcionamiento.

Tabla 12: Caso de Prueba Crear evidencia

Caso de Prueba de Aceptación	
Número de CP: 1	HU 1: Gestionar evidencia
Nombre: Crear evidencia	
Descripción: Se insertan los datos para crear evidencia, inicialmente se insertarán incorrectamente para verificar las validaciones del sistema, luego de forma correcta para comprobar que los datos sean almacenados.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. 	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Para crear una evidencia se debe seleccionar en el menú principal: <u>Añadir evidencia/Guardar</u>, 2. Dejar los campos en blanco. 3. Insertar en el campo título de la evidencia que el campo no esté en blanco y que no exista en la base de datos. 4. El campo resumen no es obligatorio. 5. El campo tipo de evidencia es obligatorio seleccionarlo. 6. En el campo archivo es obligatorio subir un archivo.
<p>Resultado esperado</p> <ul style="list-style-type: none"> • El sistema debe alertar al usuario cuando se deje campos en blancos. • El sistema debe alertar al usuario cuando existe una evidencia con ese título. • El sistema debe alertar al usuario cuando el campo autor tenga menos de 3 palabras. • Cuando se inserten los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos en el listado. <u>CP 5: Listar evidencias</u>.
<p>Evaluación de prueba:</p> <p>Satisfactoria</p>

Tabla 13: Caso de Prueba Ver evidencia

Caso de Prueba de Aceptación	
Número de CP: 2	HU 1: Gestionar evidencia
Nombre: Ver evidencia	
Descripción: Se muestran los datos de la evidencia.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. 	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Para ver una evidencia se debe seleccionar en el menú principal: <u>Listar evidencia/Ver</u>, o la opción <u>Buscar/parte del título de la evidencia /Ver</u>. 2. El sistema permite las opciones de modificar <u>CP 3: Modificar evidencia</u>
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Ver los datos de la evidencia seleccionada
<p>Evaluación de prueba:</p> <p>Satisfactoria</p>

Tabla 14: Caso de Prueba Modificar evidencia

Caso de Prueba de Aceptación	
Número de CP: 3	HU 1: Gestionar evidencia
Nombre: Modificar evidencia	
Descripción: Se modifican los datos de una evidencia. Inicialmente, se modificarán incorrectamente para verificar las validaciones del sistema, luego de forma correcta y se muestra los datos actualizados.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. 	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Para modificar una evidencia se debe seleccionar del menú principal la opción: <u>Listar evidencia/Modificar/Actualizar</u>, o la opción <u>Añadir evidencia/Editar evidencia/Actualizar</u>. 2. No dejar los campos en blanco. 3. Insertar en el campo título de la evidencia que el campo no esté en blanco y que no exista en la base de datos. 4. El campo resumen no es obligatorio. 5. El campo tipo de evidencia es obligatorio seleccionarlo. 6. En el campo archivo es obligatorio subir un archivo. 7. El campo correo debe ser obligatorio.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • El sistema debe alertar al usuario cuando se dejen campos en blancos. • Cuando se actualicen los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos inmediatamente que se presione el botón Actualizar <u>CP 2: Ver evidencia</u> y en el listado <u>CP 5: Listar evidencias</u>.
<p>Evaluación de prueba:</p> <p>Satisfactoria</p>

Tabla 15: Caso de Prueba Eliminar evidencia

Caso de Prueba de Aceptación	
Número de CP: 4	HU 1: Gestionar evidencia
Nombre: Eliminar evidencia	
Descripción: Se elimina una evidencia.	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado o tener el rol de Administrador. • Los usuarios solo pueden modificar la evidencia creada por ellos.
<p>Entrada/Pasos de ejecución:</p> <p>1. Para eliminar una evidencia se debe seleccionar en el menú principal la opción: <u>Listar evidencia/Eliminar</u> o la opción <u>Buscar/parte del título de la evidencia/Eliminar</u></p> <p>Verificar que la evidencia eliminada no aparece en el listado <u>CP 5: Listar evidencia</u>.</p>
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado o tener el rol de Profesor. • Los usuarios solo pueden modificar la evidencia creada por ellos.
<p>Evaluación de prueba:</p> <p>Satisfactoria</p>

Tabla 16: Caso de Prueba Listar evidencia

Caso de Prueba de Aceptación	
Número de CP:5	HU 3: Listar evidencias
Nombre: Listar evidencias	
Descripción: Se muestran todas las evidencias almacenadas en el sistema ordenadas por la fecha de creación.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado o tener el rol de Administrador. 	
<p>Entrada/Pasos de ejecución:</p> <p>1. Para listar las evidencias se debe seleccionar en el menú principal la opción: <u>Listar evidencias</u>.</p>	

<p>Resultado esperado:</p> <ul style="list-style-type: none"> • El sistema debe mostrar mensaje de alerta en caso de no tener permisos. • Se muestran todas las evidencias almacenadas en el sistema.
<p>Evaluación de prueba:</p> <p>Satisfactoria</p>

Tabla 17: Caso de Prueba Reporte

Caso de Prueba de Aceptación	
Número de CP: 6	HU 6: Generar reporte
Nombre: Reporte	
Descripción: Se selecciona según el usuario desee los criterios para realizar el reporte.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe tener el rol de Jefe de departamento, Jefe de postgrado o Administrador. 	
Entrada/Pasos de ejecución:	
1. Para realizar reportes se selecciona en el menú principal o el menú de navegación la opción: <u>Generar reportes</u> .	
Resultado esperado:	
<ul style="list-style-type: none"> • El sistema debe mostrar una tabla con la cantidad de categorías, según los filtros que realice el usuario. 	
Evaluación de prueba:	
Satisfactoria	

Tabla 18: Caso de Prueba Balance general

Caso de Prueba de Aceptación	
Número de CP: 7	HU 5: Generar balance general

Nombre: Balance general
Descripción: Se genera el balance general de la facultad.
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe tener el rol Jefe de postgrado.
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Para realizar reportes se selecciona en el menú principal o el menú de navegación la opción: <u>balance general</u>.
Resultado esperado: <ul style="list-style-type: none"> • El sistema debe mostrar una tabla con el balance general.
Evaluación de prueba: Satisfactoria

Tabla 19: Caso de Prueba Exportar a PDF

Caso de Prueba de Aceptación	
Número de CP: 8	HU 9: Exportar a PDF
Nombre: Exportar a PDF	
Descripción:	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe tener el rol de Jefe de departamento, Jefe de Postgrado o Administrador. 	
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Para realizar reportes se selecciona en el menú principal o el menú de navegación la opción: <u>Generar reportes/VersiónPDF</u>. 	
Resultado esperado: <ul style="list-style-type: none"> • El sistema debe permitir mostrar los reportes en formato PDF. 	

Evaluación de prueba:

Satisfactoria

3.6.3 Resultados de las pruebas

Se realizaron tres iteraciones de las pruebas de aceptación con el cliente, para conocer el grado de satisfacción hacia el producto. Durante la primera iteración se encontraron once no conformidades (NC) de prioridad Media y Baja, relacionadas con la redacción y validación de campos en la aplicación, las cuales fueron resueltas por el equipo de desarrollo. Se realizó una segunda iteración, en la cual se encontraron dieciseis NC de prioridad Media y Baja, relacionadas con la redacción y validación de campos en la aplicación, asignación de responsabilidades a los roles, las cuales fueron resueltas por el equipo de desarrollo. Se realizó una tercera iteración, en la cual se encontraron cinco NC de prioridad Media, relacionadas con el formato de algunos reportes y la redacción, las cuales fueron resueltas por el equipo de desarrollo, evaluándose de esta manera todos los Casos de Prueba como Satisfactorios.

3.7 Conclusiones parciales

La propuesta de diseño realizado en el Capítulo 2 fue la base para llevar a cabo el desarrollo de las funcionalidades especificadas por el cliente, haciendo uso de las herramientas y tecnologías seleccionadas en el Capítulo 1, dándole solución a las dificultades que presentaba el departamento de Investigación y Postgrado en la gestión de sus evidencias. Además, se aplicaron las pruebas de aceptación con el objetivo que lo que desea el cliente esté funcionando de manera correcta.

Conclusiones generales

La realización de este trabajo satisfizo en su totalidad los objetivos trazados, pudiéndose destacar, de manera general, las conclusiones siguientes:

- El estudio del arte y el empleo de los métodos científicos proporcionó fundamentar la teoría que sustenta la investigación, así como la correcta selección de las herramientas y tecnologías.
- El análisis de las soluciones existentes que pueden ser utilizadas para la gestión de documentos, permitió determinar el desarrollo de una herramienta que se adapte a las características propias de los proceso de la Investigación y el Postgrado.
- La metodología seleccionada permitió satisfacer las necesidades del cliente, mediante los artefactos generados por la misma, facilitando así una buena comunicación entre el cliente y el equipo de desarrollo.
- El diseño propuesto fue la base para definir las funcionalidades que le dieron solución a las deficiencias en la gestión de las evidencias.
- La validación del sistema facilitó valorar la calidad y el correcto funcionamiento del mismo, pudiendo asegurar que las funcionalidades alcanzadas se encuentran en correspondencias con las necesidades del cliente.

Recomendaciones

Una vez concluida la presente investigación, se recomienda:

- A la Dirección de Investigación y Postgrado de la Facultad 4 tener en cuenta para futuras versiones realizar un currículum vitae del profesor utilizando sus evidencias.
- A la Dirección de Investigación y Postgrado de la Facultad 4 tener cuenta para futuras versiones realizar comparaciones de resultados.

Referencias bibliográficas

1. "INCLUSIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN EN LAS DIERENTES MATERIAS". **ELENA P. VARO MARTÍNEZ, MARIA TERESA TORRES PINTOR**. Andalucía : s.n., 2009, Vol. 18.
2. ISO . [En línea] 2001. [Citado el: 23 de enero de 2012.] <http://www.iso.org>.
3. **Mugica, Mayra Mena**. *Gestión Documental y organización de Archivos*. s.l. : Felix Varela, 2005.
4. International council on archives. [En línea] [Citado el: 12 de enero de 2012.] <http://www.ica.org>.
5. **Torres, Irima Campillo**. Tesis Doctoral. *Sistema de Gestión de Documetos de archivos para empresas de la construcción del territorio de Camaguey*. 2010.
6. LA GESTIÓN DEL CONOCIMIENTO. GESTIÓN DOCUMENTAL. [En línea] [Citado el: 8 de enero de 2012.] <http://www.gestion-conocimiento.com/contenido/gestiondoc.asp>.
7. Aplicaciones Web a la medida México. [En línea] [Citado el: 3 de diciembre de 2011.]
8. REAL ACADEMIA ESPAÑOLA. [En línea] [Citado el: 4 de diciembre de 2011.] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=evidencia.
9. Sitio Oficial OrfeoGPL. [En línea] [Citado el: 17 de enero de 2012.] <http://www.Orfeo.com>.
10. Desoft Ficha TécnicaAvilaDoc. [En línea] [Citado el: 12 de enero de 2012.] <http://www.desoft.cu/Portals/0/FichaTecnicaAvilaDoc.pdf>.
11. Sitio Oficial de Alfresco. [En línea] [Citado el: 7 de enero de 2012.] <http://www.alfresco.com>.
12. *Alfresco ECM Version 3.0. Installing and Configuring Alfresco ECM Version 3.0*. 2008.
13. **Figuroa, Roberth G, Solís, Camilo J. y Cabrera, Armando A**. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*.
14. Metodologías tradicionales y Metodologías ágiles. [En línea] [Citado el: 12 de diciembre de 2011.]

<http://www.eumed.net/libros/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>.

15. Metodologías para la Ingeniería Web. [En línea] [Citado el: 23 de marzo de 2012.] <http://www.lsi.us.es/docencia/get.php?id=2086>.

16. **María José Escalona, Nora Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web Un estudio comparativo.* 2002.

17. **Sutherland, Dr. Jeff.** El Método SCRUM. [En línea] [Citado el: 20 de noviembre de 2011.] http://www.mastersoft.com.ar/MsWeb/otros_archivos/NotaScrumPCUsers.pdf.

18. **Jeffries, R., Anderson, A., Hendrickson, C.** *"Extreme Programming Installed"*. 2001.

19. **Beck, K.** *"Una explicación de la programación extrema. Aceptar el Cambio"*. 1999.

20. Framework. What is Framework? [En línea] [Citado el: 4 de marzo de 2012.] <http://www.framework.com/>.

21. Los framework más destacados. [En línea] 22 de julio de 2011. [Citado el: 21 de enero de 2012.] http://www.rpp.com.pe/2011-07-22-los-5-frameworks-php-mas-destacados-noticia_387351.html.

22. ¿Qué es symfony? [En línea] 2012. [Citado el: 23 de enero de 2012.] <http://www.symfony.es/que-es-symfony/>.

23. **Chalef, D.** CMS_SPAIN.com. [En línea] 2009. [Citado el: 8 de enero de 2012.] <http://www.ecm-spain.com/noticia.asp?IdItem=7881>.

24. **Aldama, Julio Santillán.** *Sistemas de Gestión de Contenido CMS.* febrero 2007.

25. Curso de Drupal 7. [En línea] [Citado el: 12 de enero de 2012.] <http://www.adrformacion.com/cursos/drupal7/leccion1/tutorial2.html>.

26. Joomla. [En línea] [Citado el: 11 de noviembre de 2011.] <http://www.joomla.org/que-es-joomla>.

27. Drupal. [En línea] [Citado el: 11 de noviembre de 2011.] <http://drupal.org.es/caracteristicas>.

28. ¿Framework o CMS? [En línea] 18 de junio de 2008. [Citado el: 23 de noviembre de 2011.] <http://jpromerobx.blogspot.com/2008/06/framework-o-cms.html>.
29. **Ansejo, Jorje Sanchez.** [UNIDAD 2] gestores de bases de datos. [En línea] [Citado el: 10 de enero de 2012.] <http://es.scribd.com>.
30. SQLite, el motor de bases de datos ágil y robusto. [En línea] 1 de julio de 2008. [Citado el: 12 de diciembre de 2011.] <http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>.
31. PostgreSQL. [En línea] [Citado el: 9 de enero de 2012.] http://www.postgresql.org.es/sobre_postgresql.
32. Guía para la personalización de PostgreSQL 8.4. [En línea] [Citado el: 23 de enero de 2012.] <http://postgresql.uci.cu>.
33. Consumoteca. [En línea] [Citado el: 23 de marzo de 2012.] <http://www.consumoteca.com/diccionario/servidor-web>.
34. OBSERVATORIO TECNOLÓGICO. [En línea] [Citado el: 12 de diciembre de 2011.] <http://recursostic.educacion.es/observatorio/web/es/component/content/article/1022-monograficodrupal-7?start=3>.
35. taringa. [En línea] [Citado el: 12 de enero de 2012.] http://www.taringa.net/posts/info/2688390/Apache-vs-IIS-_cual-elegir_.html.
36. civeraula. [En línea] [Citado el: 21 de enero de 2012.] http://linux.ciberaula.com/articulo/linux_apache_intro.
37. MiTecnológico. [En línea] [Citado el: 23 de diciembre de 2011.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.
38. Lenguajes del lado servidor o cliente. [En línea] [Citado el: 26 de enero de 2012.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
39. **Castro, Elisabeth.** *HTML con XHTML y CCS.*
40. Guía de referencia css 2.1. [En línea] [Citado el: 1 de marzo de 2012.] <http://www.w3c.es/Divulgacion/GuiasReferencia/CSS21>.

41. PHP: Hypertext Preprocessor. [En línea] [Citado el: 21 de enero de 2012.] <http://php.net>.
42. Entornos de Desarrollo Integrado. [En línea] [Citado el: 12 de enero de 2012.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
43. todoroms. [En línea] 27 de mayo de 2011. [Citado el: 12 de enero de 2012.] <http://www.todoroms.com/zend-studio-professional-edition-8-0-1>.
44. **A, Alvarez M.** Zend Studio. [En línea] [Citado el: 21 de diciembre de 2011.] <http://www.desarrolloweb.com/articulos/1178.php>.
45. ¿Qué es NetBeans? [En línea] [Citado el: 23 de enero de 2012.] <http://ayuda-java.blogspot.com/2007/07/qu-es-netbeans.html>.
46. Geany editor de texto y entorno integrado de desarrollo. [En línea] [Citado el: 2 de marzo de 2012.] http://www.thob.org/articulo/geany_editor_y_entorno_de_desarrollo_integrado-215933.html.
47. **Beck, Kent.** *Extreme Programming Explained: Embrace Change*. 1 edición. s.l. : Addison-Wesley Pub Co, 1999.
48. Metodologías ágiles para el desarrollo de software. [En línea] [Citado el: 1 de enero de 2012.] <http://www.willydev.net>.
49. **M, Flower.** "Is Design Dead?". [En línea] [Citado el: 23 de marzo de 2012.] <http://www.martinfowler.com/articles/designDead.html>.
50. **Delgado, Luis Miguel Echeverry Tobón Luz Elena.** *Caso práctico de la metodología ágil XP al desarrollo de software*. s.l. : España.
51. **Mercer, David.** *Drupal 7*. s.l. : Packt Publishing, Septiembre, 2010.
52. **Miles, Earl, Miles, Lynette.** *Drupals Building Blocks*. s.l. : Addison Wesley, Diciembre, 2010.
53. **Noble, Mark.** *Drupal 7 First Look*. s.l. : Packt Publishing, Noviembre, 2010.

54. **Shreves, Ric.** **Drupal 7. Themes.** *s.l. : Packt_Publishing, Mayo, 2011.* Mayo, 2011.
55. **Tomlinson, Todd y VanDyk, John K.** **Drupal 7 Development** ,3ra.Edition.
56. **Jockowicz, Jose.** **REGLAS Y PRÁCTICAS EN EXTREMING PROGRAMMING.**
España : s.n., 2008.

Anexos

Anexo 1: HU Autenticar usuario

Tabla 20: Autenticar usuario

Historia de Usuario	
Número de HU: 7	Nombre: Autenticar usuario
Usuarios: Usuario autenticado	
Puntos de estimación: 1	Iteración asignada: 1
Prioridad del negocio: Alta	Riesgos en el desarrollo: Media
Descripción: Permite a todos los profesores de la facultad que introduzcan su cuenta de usuario y contraseña dentro de la aplicación.	
Observaciones:	

Anexo 2: HU Gestionar noticia

Tabla 21: Gestionar noticia

Historia de Usuario	
Número de HU: 8	Nombre: Gestionar noticia
Usuarios: Jefe de postgrado	
Puntos de estimación: 1	Iteración asignada: 3
Prioridad del negocio: Baja	Riesgos en el desarrollo: Baja
Descripción: Permite crear, modificar, eliminar y mostrar una noticia de interés para la superación de los profesores de la Facultad 4.	
Observaciones:	

Anexo 3: HU Exportar a PDF

Tabla 22: Exportar a PDF

Historia de Usuario	
Número de HU: 9	Nombre: Exportar a PDF
Usuarios: Jefe postgrado, Jefe de departamento	
Puntos de estimación: 0.5	Iteración asignada: 3

Prioridad del negocio: media	Riesgos en el desarrollo: Baja
Descripción: Permite hacer reportes en formato PDF, según los permisos en la aplicación.	
Observaciones:	

Anexo 4: HU Gestionar tema del foro

Tabla 23: Gestionar tema del foro

Historia de Usuario	
Número de HU: 10	Nombre: Gestionar tema del foro
Usuarios: Jefe de postgrado	
Puntos de estimación: 0.5	Iteración asignada: 3
Prioridad del negocio: Media	Riesgos en el desarrollo: Baja
Descripción: Permite crear, eliminar y modificar un tema donde se retroalimentará información de importancia para los usuarios. Después de creado el tema será publicado.	
Observaciones:	

Anexo 5: HU Vistazo

Tabla 24: Vistazo

Historia de Usuario	
Número de HU: 11	Nombre: Vistazo
Usuarios: Administrador	
Puntos de estimación: 0.5	Iteración asignada: 3
Prioridad del negocio: Media	Riesgos en el desarrollo: Baja
Descripción: Muestra el listado de usuarios conectados en la aplicación. Destacados en el mes.	
Observaciones:	

Anexo 6: HU Realizar búsquedas

Tabla 25: HU Realizar búsquedas

Historia de Usuario	
Número de HU: 12	Nombre: Realizar búsquedas
Usuarios: Usuario autenticado	
Puntos de estimación: 0.5	Iteración asignada: 3
Prioridad del negocio: Media	Riesgos en el desarrollo: Baja
Descripción: Permite a los usuarios realizar búsquedas sobre información contenida en el sistema.	
Observaciones:	

Anexo 7: HU Software

Tabla 26: HU Software

Historia de Usuario	
Número de HU: 1	Nombre: Software
Descripción: <ul style="list-style-type: none">• Cliente Sistema operativo Linux o Windows (98 o superior). Navegador web Mozilla Firefox 3.0 o superior. Navegador web Internet Explore 6.0 o superior• Servidor Servidor web Apache 2.2.14 PHP5.3.2 o superior. PosgreSQL 8.4 o superior.	

Anexo 8: HU Hardware

Tabla 27: HU Hardware

Historia de Usuario	
Número de HU: 2	Nombre: Hardware
Descripción: <ul style="list-style-type: none">• Microprocesador Intel Pentium III o superior.• Disco duro con capacidad de 160 GB como mínimo.• Tarjetas de red cableada, inalámbrica o modem.	

Anexo 9: HU Apariencia o interfaz externa

Tabla 28: HU Apariencia o interfaz externa

Historia de Usuario	
Número de HU: 3	Nombre: Apariencia o interfaz externa
Descripción: <ul style="list-style-type: none">• El sistema debe contar con una interfaz amigable, permitiéndole así al usuario una cómoda interacción con la aplicación.• Las páginas deben tener pocas imágenes.	

Anexo 10: HU Usabilidad

Tabla 29: HU Usabilidad

Historia de Usuario	
Número de HU: 4	Nombre: Usabilidad
Descripción: <ul style="list-style-type: none">• El sistema contará con un ambiente simple y manejable para sus usuarios, incluso para aquellas personas que no tienen experiencia en el uso con ordenadores.	

Anexo 11: HU Funcionalidad

Tabla 30: HU Funcionalidad

Historia de Usuario	
Número de HU: 5	Nombre: Funcionalidad
Descripción: <ul style="list-style-type: none">• Mínima cantidad de páginas para ejecutar todas la funciones.	

Anexo 12: HU Seguridad

Tabla 31: HU Seguridad

Historia de Usuario	
Número de HU: 6	Nombre: Seguridad
Descripción: <ul style="list-style-type: none">• Se realizarán validaciones a la información contenida en el sistema. La subdirectora, los jefes de departamento y profesores se comportarán como usuarios, todos deben de estar autenticados para ver la información referente a su rol, según lo definido en el sistema.	

Anexo 13: HU Disponibilidad

Tabla 32: HU Disponibilidad

Historia de Usuario	
Número de HU: 7	Nombre: Disponibilidad
Descripción: <ul style="list-style-type: none">• La aplicación estará disponible las 24 horas del día, los 7 días de la semana y garantizará un acceso de forma fácil y rápida para los usuarios.	

Anexo 14: HU Portabilidad

Tabla 33: HU Portabilidad

Historia de Usuario	
Número de HU:8	Nombre: Portabilidad
Descripción:	
<ul style="list-style-type: none">El sistema podrá ejecutarse en varios sistemas operativos como Linux y Windows.	

Anexo 15: Consumo de memoria de Netbeans

Nombre del proceso	Estado	% CPU	Prioridad	ID	Memoria
indicator-applet-session	Durmiendo	0	0	1454	2,9 MiB
indicator-application-service	Durmiendo	0	0	1467	768,0 KiB
indicator-me-service	Durmiendo	0	0	1474	932,0 KiB
indicator-messages-service	Durmiendo	0	0	1464	800,0 KiB
indicator-session-service	Durmiendo	0	0	1472	940,0 KiB
indicator-sound-service	Durmiendo	0	0	1469	896,0 KiB
java	Durmiendo	0	0	6227	88,6 MiB
nautilus	Durmiendo	0	0	1355	21,2 MiB
nm-applet	Durmiendo	0	0	1347	3,9 MiB
notification-area-applet	Durmiendo	0	0	1439	2,0 MiB
notify-osd	Durmiendo	0	0	1477	3,5 MiB
polkit-gnome-authenticatio	Durmiendo	0	0	1351	1,4 MiB
pulseaudio	Durmiendo	0	-11	1350	1,5 MiB
python	Durmiendo	0	0	1484	6,8 MiB

Figura 6: Consumo de memoria de Netbeans

Anexo 16: Consumo de memoria de Geany

Nombre del proceso	Estado	% CPU	Prioridad	ID	Memoria
firefox	Durmiendo	0	0	1630	95,2 MiB
gconfd-2	Durmiendo	0	0	1327	2,1 MiB
gconf-helper	Durmiendo	0	0	1371	588,0 KiB
gdu-notification-daemon	Durmiendo	0	0	1475	1,7 MiB
geany	Durmiendo	0	0	5868	5,8 MiB
gedit	Durmiendo	0	0	4170	13,4 MiB
gnome-commander	Durmiendo	0	0	1520	5,3 MiB
gnome-keyring-daemon	Durmiendo	0	0	1268	460,0 KiB
gnome-panel	Durmiendo	0	0	1356	6,4 MiB
gnome-power-manager	Durmiendo	0	0	1353	3,7 MiB
gnome-pty-helper	Durmiendo	0	0	5869	716,0 KiB
gnome-pty-helper	Durmiendo	0	0	3744	716,0 KiB
gnome-screensaver	Durmiendo	0	0	1375	1,5 MiB
gnome-session	Durmiendo	0	0	1286	1,5 MiB

Figura 7: Consumo de memoria de Geany

Anexo 17: Indicadores de desempeño en Investigación y Postgrado

Balance 2011 Indicadores , 7 de sep

DATOS DE LA PLANTILLA	tercer corte			Balance	DHE	DME	DIST	tercer corte	
	1 er Corte	2do Corte	3er C 7 sep					D P Aud	Grupos
Investigadores									
Titulares y Auxiliares	2	2	3	3			1		2
Asistentes	5	5	7	7	2			2	1
Instructores	37	37	54	54	28	11	4	4	4
IRG	38	38	21	21		17	3		1
PREMIOS									
Otorgados por la ACC									
Otorgados por el Fórum de Ciencia y Técnica		4	9	13	5	2	1		1
Internacionales:									
Nacionales			1	1				1	
Provinciales									
Concurso de las BTJ y Exposición Forjadores del Futuro									
Sellos Forjadores del Futur									
De la reserva en el FCT									
CITMA									
PUBLICACIONES									
En la Web of Science									
En revistas referenciadas	1	6	6	8			2		3
En revistas nacionales arbitradas no referenciadas									
Libros									
No arbitradas		6	6	5			1		4
En memorias de eventos	24	33	37	54	13	22	9		5
En repositorios institucionales		1	1	1					1
En la serie científica interna		1	5		1 pp	2 Pp			
TRABAJOS PRESENTADOS									
En eventos internacionales en el extranjero		1	3	6		2	1		3
En eventos internacionales en Cuba	24	25	86	47	17	14	7		4
En eventos nacionales	2	28	17	18	1	5	11	1	
Aceptados y no expuestos en eventos internac	3	3	4	3	3				
Aceptados y no expuestos en eventos nacionales									
Ingresos por eventos desarrollados en el área									

Figura 8: Indicadores de desempeño en Investigación y Postgrado

Glosario de términos

PDF: acrónimo de *Portable Document Format* (en español: Formato de Documento Portátil), es un formato de almacenamiento de documentos, desarrollado por la empresa *Adobe System*. Y falta poner el concepto objeto, abstracción, encapsulación, herencia y dos pas que hay en la parte de las tarjetas CRC de programación orientada a objeto.

ISO: es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ONs) nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como normas ISO.

Información: es lo que se obtiene del procesamiento de datos, o sea, el resultado final.

Objeto: En el paradigma de programación orientada a objetos (POO, o bien *OOP* en inglés), un objeto se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico se define como la instancia de una clase.

Clase: es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten. Un objeto de una determinada clase se denomina una instancia de la clase.

Herencia: es uno de los mecanismos de los lenguajes de programación orientada a objetos basados en clases, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad. La clase de la que se hereda se suele denominar *clase base*, *clase padre*, *superclase*, *clase ancestro* (el vocabulario que se utiliza suele depender en gran medida del lenguaje de programación).

Polimorfismo: en programación orientada a objetos el polimorfismo se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía.

Encapsulamiento: se denomina encapsulamiento al ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto. El aislamiento protege a los datos asociados a un objeto contra su modificación por quien no tenga derecho a acceder a ellos, eliminando efectos secundarios e interacciones. De esta forma el usuario de la clase puede obviar la

implementación de los métodos y propiedades para concentrarse sólo en cómo usarlos. Por otro lado se evita que el usuario pueda cambiar su estado de maneras imprevistas e incontroladas.

Abstracción: desde el punto de vista de la programación orientada a objetos expresa las características esenciales de un objeto, las cuales distinguen al objeto de los demás.