



Universidad de las Ciencias Informáticas

Facultad 1

Título: Aplicación web para la gestión de la información de metodologías de desarrollo de software para el Centro de Identificación y Seguridad Digital.

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

Autora: Edignis Pérez Darromán

Tutores: Msc. Alina Surós Vicente

Ing. Joe Luis Borrero Senciales

La Habana, 2011

“Año 53 de la Revolución”



"El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que le compete. Ernesto Che Guevara.

Declaración de autoría

Declaro que yo Ediagnis Pérez Darromán soy la única autora de este trabajo y autorizo al Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

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

Autora:

Ediagnis Pérez Darromán

Tutores:

Ing. Joe Luis Borrero Senciales

Msc. Alina Surós Vicente

A mi mamá Ibis, por ser lo más grande que tengo, por su dedicación, amor y cariño en todos los momentos de mi vida, por la confianza que siempre ha depositado en mí, porque gracias a ella he llegado a lograr el sueño más grande de mi vida.

A mi papá que aunque no esté presente físicamente forma parte de mí.

A mis abuelos Raquel y Juan, por formar parte de mi educación y preocuparse siempre por todo mi futuro.

A mi tía Teresa por su apoyo infinito.

A mis dos sobrinos Leani y Alvaro.

Agradecimientos

A mi mamá por ser la mejor madre del universo y dedicar su vida para que la mía sea mejor cada día.

A mis abuelos.

A mi tía.

A mis primos Raudelis, Abel Adrián y Yordanis.

A mis hermanas Adienis y Ediliagnis.

A David, por todo su amor, cariño y comprensión, porque poco a poco se convirtió en una persona especial para mí y lo seguirá siendo para toda la vida.

A todos mis amigos, por hacer de esta universidad como mi propia casa, en especial a Yenisey, Alejandro que me ayudó mucho en este quinto año, Lisbet, Tahimí, Mairelys, Yunieski. Eider, Yadier y Rodney.

A Dairo que más que compañero ha sido amigo durante estos cinco años.

A mis tutores por todo su apoyo.

A todos en general que de una forma u otra contribuyeron a que mi sueño de ser Ingeniera informática hoy se haga realidad.

Muchas gracias.

El objetivo fundamental del Centro Identificación y Seguridad Digital (CISED) es el desarrollo de productos, servicios y soluciones integrales en el campo de la identificación y la seguridad. Durante la producción de software los cambios son frecuentes por lo que se hace necesario las metodologías de desarrollo de software para estructurar, plantear y controlar sus procesos.

El presente trabajo de diploma establece como objetivo desarrollar una aplicación web para la gestión de la información de las metodologías de desarrollo de software para los proyectos del CISED. El desarrollo de esta aplicación incluye el análisis de sistemas informáticos similares a la propuesta de solución. Estuvo guiado por la metodología XP, en la cual se analizaron cada uno de los artefactos generados en cada una de sus fases, tales como exploración y planificación, diseño, implementación y prueba. Después del estudio de las tendencias tecnológicas en la actualidad se realizó una selección de herramientas libres y de código abierto.

Se desarrolló la aplicación dándole cumplimiento a los objetivos específicos, la cual tiene como principales funcionalidades la gestión de metodologías de desarrollo de software, artículos de las metodologías y descargas de herramientas. De esta manera el CISED cuenta con un repositorio de información de metodologías de desarrollo de software lo que trae consigo la aceleración en el área de investigación para el desarrollo de una aplicación, logrando mayor eficiencia, organización y simpleza.

Palabras claves: desarrollo, gestión, información, metodologías

Índice de contenido

Introducción 1

Capítulo I..... 6

1.1. Introducción 6

1.2. Conceptos asociados a la gestión de información..... 6

1.3. Análisis de las soluciones existente 7

1.4. ¿Qué es una Web? 8

1.5. Aplicaciones web 9

1.6. Los Sistemas de Gestión de Contenidos..... **¡Error! Marcador no definido.**

1.6.1. Drupal 12

1.7. Lenguajes de programación a utilizar 13

1.7.1. PHP 13

1.8. Gestor de Base de Datos 14

1.8.1. MySQL..... 15

1.8.2. PostgreSQL 15

1.9. Análisis de las diferentes metodologías de desarrollo de software 16

1.9.1. Metodología Robusta o Pesada..... 17

1.9.2. Metodologías ágiles o ligeras 18

1.9.3. XP 19

1.10. Herramientas utilizadas para el desarrollo del sistema..... 22

1.10.1. Servidor Web Apache..... 22

1.10.2. Visual Paradigm 23

1.11. Conclusiones 23

Capítulo II.....	24
2.1. Introducción	24
2.2. Objetivos estratégicos de la organización	24
2.3. Descripción del sistema propuesto	24
2.3.1. Requerimientos no funcionales del sistema.....	25
2.4. Fase de Exploración. Definición.....	27
2.4.1. Historias de Usuarios (HU)	27
2.5. Fase de Planificación. Definición.	30
2.5.1. Estimación del esfuerzo por historia de usuario.....	30
2.5.2. Plan de Iteraciones.....	32
2.5.2.1. Iteración 1.....	32
2.5.2.2. Iteración 2.....	33
2.5.2.3. Iteración 3.....	33
2.5.3. Duración de las iteraciones	33
2.5.4. Plan de entregas.....	35
2.6. Diseño del sistema.....	35
2.6.1. Estructura del CMS Drupal	36
2.6.2. Principales módulos de soporte a usar	38
2.6.3. Patrones de Diseño en Drupal.....	39
2.6.4. Diagrama de Despliegue	41
2.7. Conclusiones	42
Capítulo III.....	43
3.1. Introducción	43

3.2. Implementación del sistema.....	43
3.2.1. Prácticas de XP.....	43
3.3. Iteración 1	46
3.4. Iteración 2	47
3.5. Iteración 3	48
3.5. Pruebas.....	49
3.5.1. Pruebas Unitarias.....	49
3.5.2. Pruebas de aceptación.....	49
3.6. Conclusiones	60
Conclusiones.....	62
Recomendaciones.....	63
Bibliografía referenciada	64
Glosario de términos	69
Anexos	71
Anexo 1: Operacionalización de las variables.....	71
Anexo 2: Historias de Usuarios.....	72
Anexo 3: Tareas de ingeniería	81
Anexo 4: Casos de Pruebas de Aceptación	103
Anexo 5: Imágenes de la aplicación.....	116

Introducción

El desarrollo de software es una disciplina que todos relacionamos en forma directa con el progreso, las mejoras en la productividad, y mucha gente inteligente trabajando duro y generando importantes beneficios para las empresas y toda la sociedad. Pero al mismo tiempo observamos que muchas veces los proyectos sufren retrasos y no se obtienen los resultados esperados pese al talento y esfuerzo puestos en acción por parte de analistas, programadores y usuarios para que el nuevo sistema funcione correctamente en tiempo y forma.

Para lograr un producto con la calidad requerida, el desarrollo de un *software* depende de una guía. Las metodologías de desarrollo de software nacen para intentar solucionar este conflicto. Su propósito es establecer un contrato social entre todos los participantes en un proyecto para conseguir la solución más eficaz con los recursos disponibles.

La producción de software es uno de los renglones más importante en el desarrollo de la economía de los países. En Cuba existen diversas empresas que se dedican a la creación de software para otras entidades como es el caso de la educación, la salud, el comercio, entre otras.

En la Universidad de las Ciencias Informáticas (UCI) con el objetivo de apoyar el desarrollo de la informatización del país, además del proceso docente educativo existen diferentes proyectos productivos encaminados a disímiles temas.

La producción de *software* se realiza en los centros de desarrollo en la UCI, de los cuales forma parte el Centro de Identificación y Seguridad Digital (CISED), que está conformado por cuatro departamentos: de Identificación, Soluciones Integrales, Seguridad Digital y Biometría, encargados de realizar proyectos asociados a las líneas de investigación seguidas por cada uno de ellos. El proceso de desarrollo de software está organizado por metodologías para estructurar, planear y controlar su proceso.

Actualmente en el CISED la obtención de información acerca de las metodologías de desarrollo de software se realiza por disímiles vías, tales como el Entorno Virtual de Aprendizaje (EVA), Internet, entre otros. Esto dificulta su procesamiento y trae consigo demora al inicio en el área de investigación por lo dispersa que se encuentra la información. No existe un repositorio de información de este tipo en el

centro, condicionado a que la información no se generaliza. Es decir, cada proyecto cuando va a desarrollar un software define la metodología por la que se van a guiar, buscan todas las informaciones referentes a ella, pero luego ocurre que estas informaciones se quedan en ese grupo de trabajo y no se divulga hacia los demás proyectos.

Por lo anteriormente expuesto se formula el siguiente **problema a resolver**:

¿Cómo mejorar la gestión de la información de metodologías de desarrollo de software en el CISED?

El **Objeto de estudio** se centra en los procesos de gestión de información de las metodologías de desarrollo de software.

El **Campo de Acción** se enmarca en los procesos de gestión de información de las metodologías de desarrollo de software para los proyectos del CISED.

Objetivo General: Desarrollar una aplicación informática que permita la gestión de información de metodologías de desarrollo de software para el CISED.

Objetivos específicos:

- ❖ Realizar el análisis de los procesos, tecnologías y herramientas de gestión de información de metodologías de desarrollo de software.
- ❖ Efectuar el análisis de las diferentes tecnologías y herramientas para el desarrollo de la propuesta de solución.
- ❖ Realizar el análisis, diseño e implementación del sistema.
- ❖ Probar el correcto funcionamiento de la propuesta de solución a desarrollar.

Basado en la **hipótesis**: Si se desarrolla una aplicación informática que gestione información de las metodologías de desarrollo de software para el CISED permitirá mejorar la gestión de la información de las metodologías de desarrollo de sus proyectos.

Variable Independiente:

- ❖ Desarrollo de una aplicación informática para el CISED.

Variable Dependiente:

- ❖ Gestionar la información de metodologías de desarrollo de software para el CISED.

Operacionalización de las variables, ver Anexo 1.

En función de darle cumplimiento a los objetivos específicos planteados se han trazado las siguientes **tareas científicas:**

- ❖ Desarrollar un estudio sobre metodologías utilizadas para desarrollar el sistema.
- ❖ Investigar las soluciones similares al sistema a desarrollar.
- ❖ Efectuar un estudio sobre las herramientas a utilizar para el desarrollo de la aplicación para el CISED.
- ❖ Realizar el análisis y diseño del sistema.
- ❖ Desarrollar la implementación de la propuesta de solución.
- ❖ Realizar pruebas del funcionamiento de la propuesta de solución.

Posibles resultados: Se espera aportar al Centro de Identificación y Seguridad Digital una aplicación informática que gestione la información de metodologías de desarrollo de software para el CISED.

Para garantizar el desarrollo óptimo de la aplicación es necesario el empleo de los siguientes métodos de investigación científica:

Métodos empíricos:

Entrevistas: Consiste en realizar una conversación planificada entre el investigador y el entrevistado para obtener información acerca del tema.

Se utiliza para entrevistar a los analistas de los proyectos y determinar las principales dificultades en la gestión de las metodologías.

Experimento: Este método se basa en el estudio de un objeto en el cual el investigador crea las condiciones o adapta las existentes para el esclarecimiento de las propiedades, leyes y relaciones del objeto, para verificar una hipótesis, una teoría o un modelo. El experimento puede ser transformador o comprobador.

Este método se utiliza ya que uno de los objetivos propuestos fue el de garantizar la calidad y el funcionamiento de la propuesta de solución, por lo que es necesario la realización de pruebas para dar cumplimiento a dicho objetivo.

Métodos teóricos:

Analítico-sintético: Consiste en el estudio y análisis de la bibliografía relacionada con el tema para establecer los elementos fundamentales referentes al objeto de investigación.

Luego de haber estudiado y analizando todo lo relacionado al tema específico a abordar se pudo hacer una profundización de todas sus características, dígame objetivos, logros, o deficiencias que se derivan, así como también las mejoras que se pueden proponer para obtener resultados eficientes.

Histórico-lógico: Se emplea para el estudio y profundización de investigaciones ya realizadas relacionadas con el tema de investigación, lo que facilitó la indagación de soluciones al problema planteado.

Se utiliza este método porque uno de los objetivos planteados fue el de analizar las tecnologías, herramientas y aplicaciones de gestión de información.

Modelación: El modelo científico es un instrumento de la investigación de carácter material o teórico, creado por los científicos para reproducir el fenómeno que se está estudiando.

Se utiliza este método ya que uno de los objetivos propuestos fue el de realizar el diseño de una aplicación web por lo que a través del modelado se puede entender la lógica del proceso a automatizar.

El presente trabajo de diploma se encuentra distribuido en el documento de la siguiente manera:

Capítulo 1. Fundamentación teórica.

Se realiza un estudio conceptual de los principales elementos que se deben tener en cuenta para el desarrollo del sistema. Se hace referencia a los lenguajes de programación, sistemas gestores de base de datos, tecnología web así como las metodologías de desarrollo de software.

Capítulo 2. Propuesta de la solución.

Se realiza la propuesta del sistema, describiendo las principales funcionalidades de la propuesta de solución, así como las características mediante los artefactos definidos por la metodología. Se presenta el diseño del sistema.

Capítulo 3. Implementación y pruebas funcionales de la propuesta de solución.

Se describen las pruebas a realizar al sistema implementado, basados en los casos de prueba definidos y se muestra el resultado de su aplicación, comprobando de este modo el correcto funcionamiento de la solución.

Capítulo I.

1.1. Introducción

En el presente capítulo se describen los conceptos fundamentales asociados al dominio del problema y el objeto de estudio, haciéndose un análisis de la situación actual. Se analizarán las soluciones semejantes a la propuesta de solución, así como la fundamentación de las tecnologías que se utilizarán para el desarrollo de la aplicación, las herramientas, lenguajes de programación para el diseño del sistema y las propuestas para su implementación y desarrollo.

1.2. Conceptos asociados a la gestión de información

Gestión de Información: Es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma. [1]

Sistema de Información: Es un conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización. [2]

Metodología de desarrollo de software: Es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. Las metodologías se clasifican en estructuradas entre las que se encuentran las Orientadas a procesos, Orientadas a datos y Mixtas, así como las no estructuradas las Orientadas a Objetos y Sistema de tiempo real. [3]

Una metodología está compuesta por:

- ❖ ¿Cómo dividir un proyecto en etapas?
- ❖ ¿Qué tareas se llevan a cabo en cada etapa?
- ❖ ¿Qué restricciones deben aplicarse?
- ❖ ¿Qué técnicas y herramientas se emplean?
- ❖ ¿Cómo se controla y gestiona un proyecto?

Entre las principales características de las metodologías de desarrollo de software se destacan:

- ❖ Existencia de reglas predefinidas
- ❖ Cobertura total del ciclo de desarrollo
- ❖ Verificaciones intermedias
- ❖ Planificación y control
- ❖ Comunicación efectiva
- ❖ Utilización sobre un abanico amplio de proyectos
- ❖ Fácil formación
- ❖ Herramientas CASE
- ❖ Actividades que mejoren el proceso de desarrollo
- ❖ Soporte al mantenimiento
- ❖ Soporte de la reutilización de software [33]

1.3. Análisis de las soluciones existente

Para la implementación del sistema se hace un análisis de las soluciones similares existentes con el objetivo de profundizar en el desarrollo de la aplicación.

Scrum.es: En Scrum.es se dispone de la información necesaria para poder integrar este sistema de gestión de la producción dentro de una empresa. Introducir Scrum en la gestión y control de la producción supone aumentar la productividad sin tener que aumentar el número de horas de trabajo. Scrum se puede aplicar a cualquier tipo de trabajo de producción que realice un equipo humano. [4]

Risk Technology Development: *Risk Technology Development* (RTD) es una División de *Risk Technology*, que provee servicios de desarrollo de software a la medida de los procesos de negocio, implementación de Sistemas de Gestión Empresarial y Sistemas de Gestión de Servicios [5]

Incluye el diagnóstico de necesidades e identificación de riesgos financieros e informáticos, que permite diseñar soluciones a la medida de las empresas e instituciones, que incluyen diseño de procesos, definición de recurso humano, recursos tecnológicos, definición de planes estratégicos, desarrollo de las tecnologías necesarias para abordar las necesidades del cliente.

La metodología de desarrollo de proyectos, incluye la aplicación de las mejores prácticas basadas en RUP, arquitectura de aplicaciones, desarrollo de manuales de usuario y documentación técnica utilizando UML.

Estos sistemas son aplicaciones web las cuales tienen en común que brindan informaciones acerca de una metodología de desarrollo de software específicamente, no de todas las metodologías, además no cuentan con descargas de las herramientas que se utilizan para el desarrollo de un producto, por lo que no cumplen con los objetivos de la propuesta de solución. Por lo anteriormente expuesto surge la necesidad de un sistema que permita mostrar las informaciones de todas las metodologías utilizadas en el CISED, así como la descarga de sus herramientas.

1.4. ¿Qué es una Web?

Por éste término se suele conocer a WWW (World Wide Web), creado por el Centro Europeo de Investigación Nuclear como un sistema de intercambio de información y que Internet ha estandarizado. Supone un medio cómodo y elegante, basado en multimedia e hipertexto, para publicar información en la red. Inicial y básicamente se compone del protocolo http y del lenguaje HTML. [6]

Como principales características de la web se destacan las siguientes:

- ❖ La World Wide Web se basa en la tecnología llamada hipertexto. Esta tecnología aumenta la accesibilidad a los documentos vinculados en Internet y ayuda al usuario a navegar entre documentos muy fácilmente.
- ❖ Permite buscar, recorrer muchos tipos de información en numerosos lugares y en múltiples formas. Esta interfaz llama navegador.
- ❖ Se destina a ayudar a la gente a compartir recursos, información y servicios con la más amplia comunidad de usuarios posible. Así, un usuario puede acceder a la Web de Apple, UNIX, Macintosh, DOS, Windows y otros sistemas operativos.
- ❖ Al igual que Internet, la Web tiene un protocolo, que se conoce como Protocolo de Transferencia de Hipertexto (HTTP). Este protocolo actúa como una interfaz entre el software de cliente Web, como Netscape Navigator.

- ❖ Una ventaja importante de la World Wide Web es que también soporta TCP/IP, tales como Gopher, FTP y Archie, además de HTTP.

1.5. Aplicaciones web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. [7]

La aplicación web contiene un conjunto de páginas web estáticas y dinámicas. Una página web estática es aquella que no cambia cuando un usuario la solicita: el servidor web envía la página al navegador solicitante sin modificarla. Por el contrario, el servidor modifica las páginas dinámicas antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica.

Existen dos grupos de lenguajes de programación para la Web clasificándose en dependencia de dónde se implementan siguiendo la arquitectura Cliente/Servidor. De esta forma un grupo es llamado lenguajes de lado del Cliente y el otro grupo lenguajes del lado del Servidor.

La modalidad o arquitectura Cliente/Servidor se define como aquella en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes, una requiere servicios y la otra los brinda, pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. Esas dos categorías son justamente cliente y servidor.

Existen lenguajes de programación tanto en el lado del Cliente como en el lado del servidor. En el lado del Cliente se encuentran: JavaScript, HTML, Visual Basic Script, y en el lado del servidor PERL, ASP, JSP, PHP, Java.

Ventajas de las aplicaciones web

Las aplicaciones web tienen alta disponibilidad, ya que pueden realizar consultas en cualquier parte del mundo donde tenga acceso a Internet. No se necesita instalar nada, no depende de algún software, por lo general, también tiene ventajas tales como:

- ❖ Se puede acceder a ella desde cualquier ordenador con conexión a Internet.
- ❖ Brindan privacidad con acceso (usuario y contraseña) para acceder a tus datos.
- ❖ No necesitas actualizarlo.
- ❖ No hay discriminación generalmente acerca del sistema operativo del usuario, aunque exista por parte del servidor. [29]

Características de las aplicaciones web

- ❖ Son aplicaciones basadas en el modelo Cliente-Servidor que gestionan datos almacenados en un servidor web.
- ❖ Utilizan como interfaces páginas en formato HTML, conteniendo datos hipermedia.
- ❖ El usuario se comunica con la aplicación desde cualquier cliente conectado a la red.
- ❖ Una aplicación web es más que un conjunto de páginas entrelazadas entre sí, supone la existencia de código ejecutable.
- ❖ Las aplicaciones web se caracterizan porque están formadas por diferentes módulos, independientes entre sí, que se pueden ejecutar en el servidor o en el cliente y pueden estar implementados utilizando lenguajes o tecnologías diferentes.
- ❖ Para organizar la información se suelen utilizar Bases de Datos.

1.6. Sistema Gestor de Contenidos

Los Sistema Gestor de Contenidos (CMS) con las siglas en inglés *Content Management System*. Como el propio nombre indica, es un sistema que permite gestionar contenidos. En líneas generales, un CMS admite administrar contenidos en un medio digital y para el caso particular un CMS permite gestionar los contenidos de una web. [8]

CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza una base de datos, incluyendo nueva información o editando la existente.

Tipos de CMS

Capítulo I: Fundamentación teórica

Existen en la actualidad diferentes tipos de CMS, estos se pueden agrupar en dependencia del tipo de sitio que permiten gestionar. A continuación se muestran los fundamentales:

- ❖ **Genéricos:** Ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenidos, para soluciones de comercio electrónico, blogs, portales entre otros, ejemplos: Zope, OpenCMS, Typo3, Apache Lenya.
- ❖ **Foros:** Sitio que permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados. Ejemplos: phpBB, MyBB.
- ❖ **Blogs:** Publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión. Ejemplos: Wordpress, Typo.
- ❖ **Wikis:** Sitio web dónde todos los usuarios pueden colaborar en los artículos, aportando información o reescribiéndola. También permite espacio para discusiones. Indicado para material que irá evolucionando con el tiempo. Ejemplos: Mediawiki, Tikiwiki.
- ❖ **eCommerce:** Son Sitios web para comercio electrónico.
- ❖ **Portal:** Sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad. Ejemplos: PHPNuke, Postnuke, Drupal, Plone.
- ❖ **Galería:** Permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual, normalmente imágenes. Ejemplo: Gallery.
- ❖ **e-Learning:** Sirve para la enseñanza de conocimientos. Los usuarios son los profesores y estudiantes, tenemos aulas virtuales donde se ponen a disposición el material del curso. La publicación de un contenido por un profesor es la puesta a disposición de los estudiantes, en un aula virtual, de ese contenido. Ejemplo: Moodle.
- ❖ **Publicaciones digitales:** Son plataformas especialmente diseñadas teniendo en cuenta las necesidades de las publicaciones digitales, tales como periódicos, revistas, etc. Ejemplo: ePrints. [34]

Criterio de selección de Drupal

Drupal es uno de los tipos de CMS más reconocidos, ofrece una plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas y para construir soluciones de gestión de contenidos. Es destacado entre los CMS de tipo genérico. Se optó por el uso de este CMS

para el sistema de gestión ya que es uno de los más completos actualmente, acorde a los requerimientos de la aplicación y flexible a las modificaciones.

1.6.1. Drupal

Drupal es un CMS que permite, tanto a individuos como a comunidades de usuarios, la publicación de contenidos en Internet, de manera sencilla, y sin tener conocimientos de lenguaje de marcas. Desarrollado dentro de la filosofía del Software Libre y de código fuente abierto bajo licencia GPL (Licencia Pública General), Drupal es un sistema dinámico. [9]

Desarrollado en PHP y mantenido por una amplia comunidad, permite múltiples usuarios y distintos lenguajes de marcas. Destaca la calidad de su código, el respeto por los estándares web, un énfasis especial en la usabilidad y consistencia de todo el sistema. Su flexibilidad y adaptabilidad, así como su estructura modular, existe gran cantidad de módulos adicionales disponibles, hacen que sea adecuado para el desarrollo de sitios web con funcionalidades distintas.

Entre sus características se destacan:

- ❖ **Ayuda on-line:** Un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- ❖ **Búsqueda:** Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- ❖ **Módulos:** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante jabber, mensajes privados, bookmarks, etc.
- ❖ **Personalización:** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- ❖ **URLs amigables:** Drupal usa el *mod_rewrite* de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.
- ❖ **Independencia de la base de datos:** Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de

datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

- ❖ **Posee interfaz** de administración integrada en la propia Web, a diferencia de otros CMS, además es multiplataforma. El sistema de temas de Drupal permite generar plantillas completamente basadas en CSS y XHTML y lo mejor de todo es que los módulos normalmente generan código válido. [30]

1.7. Lenguajes de programación a utilizar

Para alcanzar el objetivo de la investigación se hace necesario utilizar PHP como lenguaje de programación ya que es un lenguaje implícito en el funcionamiento del CMS Drupal.

1.7.1. PHP

El PHP (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl, con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas”. [10]

PHP permite realizar diversas funciones, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos articulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales, donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día con códigos comparables en otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores.

Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. Es utilizado en aplicaciones web relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson y NASA.

Características de PHP

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web.

- ❖ Soporte para una gran cantidad de gestores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- ❖ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ❖ Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- ❖ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- ❖ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. [31]

1.8. Gestor de Base de Datos

Un Sistema Gestor de Base de Datos(SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. [11]

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. [12]

Un Sistema Gestor de Base de Datos (SGBD) permite realizar acciones tales como: definición de los datos, mantenimiento de la integridad de los datos dentro de la base de datos, control de la seguridad y privacidad de los datos, manipulación de los datos.

MySQL y PostgreSQL son los principales Sistemas Gestores de Base Datos que soporta el CMS Drupal.

1.8.1. MySQL

MySQL es un sistema de administración para bases de datos relacionales que provee una solución robusta a los usuarios con poderosas herramientas multi-usuario, utiliza soluciones SQL (del inglés *Structured Query Language*, Lenguaje de Consulta Estructurado). Es rápido, robusto y fácil de utilizar. Actualmente MySQL es la base de datos de código abierto más usada del mundo desarrollado. [13]

Entre las principales características podemos destacar:

- ❖ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ❖ Soporta gran cantidad de tipos de datos para las columnas.
- ❖ Gran portabilidad entre sistemas.
- ❖ Soporta hasta 32 índices por tabla.
- ❖ Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

1.8.2. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (Distribución de Software Berkeley) la cual; pertenece al conjunto de las licencia de software libre.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de desinteresadamente, libre y/o apoyados por organizaciones comerciales. [14]

Utiliza el lenguaje SQL para llevar a cabo sus búsquedas de información, las bases de datos generadas dentro de servidores de SQL relacionales. Utiliza un modelo Cliente-Servidor y usa multiprocesos en vez

de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Es considerado actualmente el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Por las características que posee PostgreSQL antes mencionadas se decide escoger este gestor de bases de datos para el desarrollo de la propuesta de solución.

1.9. Análisis de las diferentes metodologías de desarrollo de software

El desarrollo de software en ocasiones es un poco riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, se obtiene clientes insatisfechos con el resultado y desarrolladores aún más.

Las metodologías ágiles surgen como una extensión a las metodologías tradicionales para mejorar el desarrollo de sistemas, según el tipo de proyecto y empresa, añadiendo y mejorando, en otras palabras optimizando las prácticas de desarrollo de software. Es por eso que es necesario definir metodologías para guiar el proceso de desarrollo de un producto de software.

Las metodologías se definen según los pasos a seguir para el cumplimiento de un objetivo. El objetivo fundamental dentro del desarrollo software es producir un producto con la calidad requerida del cliente que cumpla con los requerimientos.

Entre las principales metodologías más usadas en el desarrollo de un software se destacan:

- ❖ XP (eXtreme Programming)
- ❖ FDD (Feature Driven Development)
- ❖ MSF (Microsoft Solution Features)
- ❖ RUP (Rational Unified Process)
- ❖ DSDM(Dynamic Systems Development Method)
- ❖ ASD(Adaptive Software Development)
- ❖ Scrum
- ❖ Crystal
- ❖ Métrica V3

1.9.1. Metodología Robusta o Pesada

RUP

Rational Unified Process con sus siglas en inglés RUP es un proceso de desarrollo de software. Asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos. [15]

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

En la **Ilustración 1** se muestran las fases de RUP.

La metodología RUP divide en 4 fases el desarrollo del software:

- ❖ **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- ❖ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- ❖ **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ❖ **Transmisión:** El objetivo es llegar a obtener el release del proyecto.

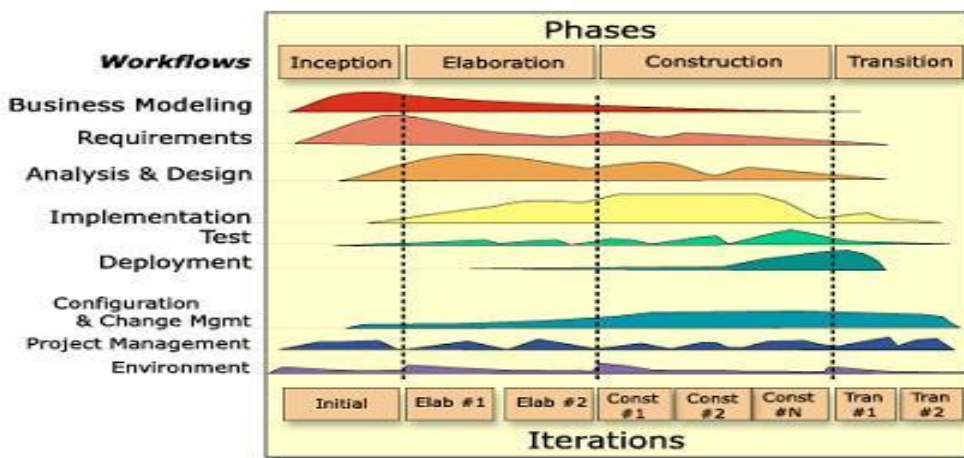


Ilustración 1- Fases de flujo de trabajo de RUP [35]

Los elementos de RUP son:

- ❖ **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- ❖ **Trabajadores:** Vienen hacer las personas involucrados en cada proceso.
- ❖ **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Características de RUP:

- ❖ **Dirigido por los casos de uso:** Los CU son una forma abstracta de representar lo que los usuarios finales necesitan, por lo cual ellos son los que deben guiar el proceso de desarrollo. Esto se garantiza con la obtención de los diferentes modelos que son el resultado de los flujos de trabajo que propone RUP.
- ❖ **Centrado en la arquitectura:** La arquitectura de un proyecto muestra una visión común del sistema completo, visión en la que deben estar de acuerdo tanto el equipo de desarrollo como los clientes.
- ❖ **La arquitectura:** Describe los elementos del modelo que son más importantes para la construcción del sistema, así como los cimientos para comprenderlo y desarrollarlo de forma económica.
- ❖ **Iterativo e incremental:** RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto, y las cuales se definen según el nivel de madurez que alcanzan los productos que se van obteniendo con cada actividad ejecutada.

1.9.2. Metodologías ágiles o ligeras

Scrum

Scrum es una metodología de desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Es un framework para mejorar la forma en la que trabajan las personas, es basado en equipos para desarrollar sistemas y productos complejos. Scrum utiliza un proceso iterativo en donde cada

iteración (llamadas sprint) es lo más corta posible, manteniendo un ritmo parejo a medida que se avanza con planificación, ejecución y reflexión. Estas cajas de tiempo estrictas y rítmicas de Scrum permiten descubrir disfunciones en la organización. [16]

Scrum define tres roles (Scrum Master, Dueño del Producto y Equipo), un conjunto de objetivos priorizados, un compromiso por cada sprint, y una forma sencilla para medir el progreso. Scrum usa ceremonias de tiempo acotado para planificar, para inspeccionar/adaptar a diario, y para inspeccionar/adaptar durante sprints.

Características de Scrum

- ❖ Scrum permite la creación de equipos auto organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.
- ❖ Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan.
- ❖ Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido.
- ❖ Se centra en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.
- ❖ Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

1.9.3. 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.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Capítulo I: Fundamentación teórica

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [17]

Entre las principales características de XP se destacan en que la metodología se basa en:

Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- ❖ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ❖ El manejo del cambio se convierte en parte sustantiva del proceso.
- ❖ El costo del cambio no depende de la fase o etapa.
- ❖ No introduce funcionalidades antes que sean necesarias.
- ❖ El cliente o el usuario se convierte en miembro del equipo.

Derechos del Cliente

- ❖ Decidir que se implementa.
- ❖ Saber el estado real y el progreso del proyecto.
- ❖ Añadir, cambiar o quitar requerimientos en cualquier momento.
- ❖ Obtener lo máximo de cada semana de trabajo.

- ❖ Obtener un sistema funcionando cada 3 o 4 meses.

Derechos del Desarrollador

- ❖ Decidir cómo se implementan los procesos.
- ❖ Crear el sistema con la mejor calidad posible.
- ❖ Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- ❖ Estimar el esfuerzo para implementar el sistema.
- ❖ Cambiar los requerimientos en base a nuevos descubrimientos.

En la **Ilustración 2** se muestran las fases de la metodología XP.

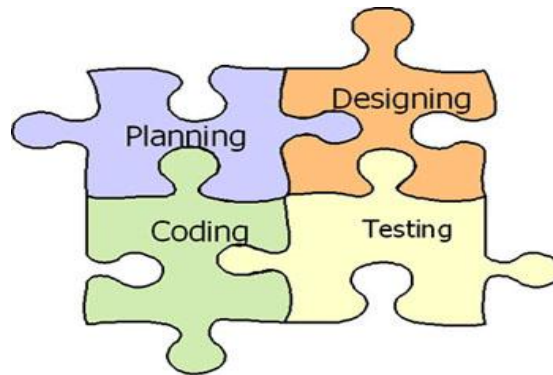


Ilustración 2- Fases de XP [36]

A partir del análisis de las diferentes metodologías de desarrollo de software y teniendo en cuenta las características del sistema propuesto se define como metodología a utilizar XP para el desarrollo del sistema. A continuación se describen las principales razones que permitieron seleccionar esta metodología.

- ❖ **Pocos roles:** Esta metodología está dirigida a grupos de desarrollo pequeños y con pocos roles como este caso.
- ❖ **El proyecto es pequeño:** XP está concebida para ser utilizada dentro de proyectos pequeños y de desarrollo rápido se adapta perfectamente a este caso.

- ❖ **Empieza en pequeño y añade funcionalidad con retroalimentación continua:** El desarrollo del sistema comienza a partir de los requerimientos básicos y a partir de ahí se van añadiendo funcionalidades que tanto el desarrollador como el cliente entiendan necesarias.
- ❖ **El manejo del cambio se convierte en parte sustantiva del proceso:** A medida que el proyecto avanza pueden surgir nuevas expectativas o ideas que pueden ser incorporadas fácilmente permitiéndole mayor adaptabilidad al producto, con la metodología XP esto es completamente factible pues esta se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.

1.10. Herramientas utilizadas para el desarrollo del sistema

A continuación se describen las herramientas utilizadas en el desarrollo de la propuesta de solución.

1.10.1. Servidor Web Apache

Apache es una aplicación gratuita que convierte cualquier ordenador en un servidor web. Es flexible, rápido y eficiente. Apache es de código abierto, lo que significa que muchas personas han contribuido a su desarrollo. Además se puede descargar desde el sitio del desarrollador y utilizarlo libre y gratuitamente sin pagar ninguna licencia. Apache permite negociar protocolos HTTP entre una máquina que haría de servidor web y los otros ordenadores que deseen ver determinado sitio web alojado aquí.

[18]

Características del Servidor Web Apache

- ❖ Continuamente actualizado y evoluciona a mayor velocidad.
- ❖ Multiplataforma.
- ❖ Se desarrolla de forma abierta.
- ❖ Modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.
- ❖ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos. [32]

1.10.2. Visual Paradigm

Visual Paradigm es una herramienta UML (Lenguaje Unificado de Modelado) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos. Presenta licencia gratuita y comercial. Es fácil de instalar, actualizar y compatible entre ediciones. [19]

Por las características mencionadas que presentan esta herramienta y la metodología de desarrollo de software que se usará para guiar el desarrollo del producto, la herramienta más idónea para el modelado es *Visual Paradigm*, por tres razones principales:

- ❖ Es una herramienta libre, lo que significa que no es necesario tener licencia para ser uso de ella.
- ❖ Posee una curva de aprendizaje menos prolongada en el tiempo en comparación con otras herramientas de modelado.
- ❖ Genera código, esto es de gran utilidad pues ahorra tiempo a los desarrolladores y reduce las posibilidades de cometer errores.

1.11. Conclusiones

En el transcurso de este capítulo se ha desarrollado un estudio de las tecnologías actuales que serán empleadas durante el desarrollo del sistema, especificando sus características y ventajas.

Se propone utilizar Drupal como sistema gestor de contenido para el desarrollo de la aplicación, PHP como lenguaje del lado del servidor, por ser un producto de código abierto, ofrece soluciones simples y universales, además es el lenguaje de programación que está creado Drupal. Se propone igualmente XP como metodología de desarrollo de software, como sistema gestor de base de datos PostgreSQL, por ser considerado actualmente el sistema de código abierto más potente del mercado. Se escogió como servidor Web el Apache y como herramienta de modelado en UML el Visual Paradigm, por ser una herramienta libre.

Capítulo II.

2.1. Introducción

En el presente capítulo se da una breve descripción del sistema a implementar, así como de los requisitos no funcionales del mismo. Además se realiza el análisis de las primeras fases de la metodología utilizada en el desarrollo del sistema. Se describe y diseña la estructura del CMS Drupal para mejor entendimiento, así como los principales módulos utilizados para el desarrollo del sistema.

2.2. Objetivos estratégicos de la organización

El Centro de Identificación y Seguridad Digital cuenta con varios proyectos los cuales brindan servicios de asesoría para entidades nacionales y extranjeras, en torno a la definición y adopción de procesos y soluciones integrales adaptadas a las necesidades legales, políticas y socioculturales de los clientes en el campo de la identificación y la seguridad digital.

Estos proyectos no cuentan con un repositorio generalizado donde puedan obtener información sobre las metodologías de desarrollo de software, tales como sus herramientas, documentos, etc. Por lo que se hace necesaria una aplicación, que brinde este tipo de información para el desarrollo de un software.

2.3. Descripción del sistema propuesto

El sistema propuesto es una aplicación web, la cual tiene como objetivo fundamental el acceso a las diferentes informaciones relacionadas con las metodologías de desarrollo de software.

Esta aplicación mostrará una serie de secciones que les permitirá a los usuarios las siguientes informaciones:

- ❖ Información específica de cada metodología de desarrollo de software utilizada en el Centro.
- ❖ Artículos sobre metodologías de desarrollo de software.
- ❖ Expedientes de proyectos para cada metodología.
- ❖ Descarga de las herramientas utilizadas por las metodologías de desarrollo de software.
- ❖ Información sobre Modelo de Boehm y Turner.
- ❖ Tecnologías utilizadas por las metodologías de desarrollo de software.

Las informaciones que mostrará esta aplicación son accesibles para todos los usuarios. Le permitirá al usuario la opción de escoger su propia metodología para desarrollar un software de acuerdo a las características específicas de cada equipo de desarrollo, así como descargar las herramientas utilizadas por cada una de ellas, además de consultar otros sitios de interés.

La aplicación consta con un sistema de administración, el cual permite actualizar toda su información.

Teniendo en cuenta la descripción anterior el sistema consta con los siguientes roles de usuarios:

❖ Usuario invitado

- ✚ Es el usuario que utiliza la aplicación para obtener todo tipo de información referente a las metodologías de desarrollo de software.

❖ Usuario editor

- ✚ Es el usuario que tiene acceso a la gestión de ciertas funcionalidades de la aplicación excepto las administrativas.

❖ Usuario administrador

- ✚ Es el usuario que configura y controla todo lo referente al comportamiento y soporte de la aplicación.

2.3.1. Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

Apariencia o interfaz externa:

- ❖ Optimizado para una resolución de 1024x768.
- ❖ Cumplir con los estándares de la W3C.

Usabilidad:

- ❖ Para utilizar el sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente Web en sentido general.
- ❖ Tener buena visibilidad en los principales navegadores.

Soporte:

- ❖ Sistema multiplataforma.
- ❖ Gestor de base de datos con soporte para grandes volúmenes de datos y alta velocidad de procesamiento.

Software:

Estaciones de trabajo (PC Cliente)

- ❖ Sistema operativo: Multiplataforma.
- ❖ Navegador web: Internet Explorer, Mozilla Firefox, Chrome, Opera, Safari, etc.

Servidor de Aplicaciones

- ❖ Servidor Web Apache 2.0.
- ❖ Versión de PHP 5.2.13.

Servidor de Base de Datos

- ❖ Sistema Gestor de Base de Datos PostgreSQL 8.2.4.

Seguridad:

- ❖ Chequear que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.
- ❖ Garantizar que las funcionalidades del sistema se muestren de acuerdo al tipo de usuario que esté activo.
- ❖ La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- ❖ Realizar salvadas periódicas de la información en otros dispositivos.

Disponibilidad:

- ❖ El sistema deberá tener un 100% de disponibilidad por lo que podrá ser usado las 24 horas del día por todos los usuarios.

2.4. Fase de Exploración. Definición

La metodología de desarrollo XP comienza con la fase de exploración, en esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. [20]

2.4.1. Historias de Usuarios (HU)

Las HU son unos de los artefactos más importantes que genera la metodología XP. Estas tienen el mismo propósito que los casos de uso y son escritas por el propio cliente, tal y como ven ellos las necesidades del sistema. [21]

Las historias de usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conllevará la implementación de dicha historia de usuario. Su nivel de detalle debe ser el mínimo posible, de manera que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Durante el análisis en la fase de exploración fueron identificadas diez HU, cada una de ellas corresponden a las diferentes funcionalidades, además proporcionan una idea al equipo de desarrollo de cómo debe ser su posterior implementación. A continuación se describen las HU más significativas. Para consulta las restantes HU, ver Anexo 2.

Tabla 1: HU Autenticar Usuario

<i>Historia de Usuario</i>	
Número: 1	Nombre: Autenticar Usuario.
Usuario: Administrador, editor, Invitado	

Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1.
Descripción: El usuario podrá autenticarse para acceder a las informaciones del sistema según los privilegios que tengan cada uno de ellos.	
Observaciones:	

Tabla 2: HU Adicionar metodologías de desarrollo

<i>Historia de Usuario</i>	
Número: 2	Nombre: Adicionar metodologías de desarrollo.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El usuario podrá adicionar los datos necesarios de la metodología.</p> <ul style="list-style-type: none"> • Clasificación (Ágil o Pesada). • Tamaño de Proyectos. • Descripción. 	

- Flujo de trabajo.
- Fases.

Observaciones:

Tabla 3: HU Modificar metodologías de desarrollo

<i>Historia de Usuario</i>	
Número: 3	Nombre: Modificar metodologías de desarrollo.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario podrá modificar los datos necesarios de la metodología.	
Observaciones:	

Tabla 4: HU Eliminar metodologías de desarrollo

<i>Historia de Usuario</i>	
Número: 4	Nombre: Eliminar metodologías de desarrollo.

Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario podrá eliminar los datos necesarios de la metodología.	
Observaciones:	

2.5. Fase de Planificación. Definición.

En la fase de planificación el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. [21]

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

2.5.1. Estimación del esfuerzo por historia de usuario.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

Tabla 5: Plan de estimación del esfuerzo por historia de usuario

Historias de Usuarios	Puntos Estimados
Autenticar Usuario.	1
Adicionar metodologías de desarrollo.	1
Modificar metodologías de desarrollo.	1
Eliminar metodologías de desarrollo.	1
Adicionar artículos.	1
Modificar artículos.	1
Eliminar artículos.	1
Adicionar herramienta.	1
Modificar herramienta.	1
Eliminar herramienta.	1
Adicionar información sobre Modelo de Boehm y Turner.	1
Modificar información sobre Modelo de Boehm y Turner.	1
Eliminar información sobre Modelo de Boehm y Turner.	1
Sugerir metodología de desarrollo según Modelo de Boehm y Turner.	1
Realizar reportes de todas las informaciones del sistema.	1
Adicionar enlaces a otros sitios.	1
Modificar enlaces a otros sitios.	1
Eliminar enlaces a otros sitios.	1

Realizar búsquedas de la información existente en el sistema.	1
Descargas de las herramientas.	1
Total	20

2.5.2. Plan de Iteraciones.

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de cinco semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Las HU son seleccionadas de acuerdo al orden preestablecido para cada entrega, son desarrolladas y probadas en un ciclo de iteración, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración para maximizar el valor de negocio. Al final de la última iteración el sistema estará listo para entrar en producción. Así mismo, para cada HU se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. [21]

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

2.5.2.1. Iteración 1

El objetivo principal de esta iteración es la implementación de las historias de usuarios seleccionadas de mayor prioridad, de esta forma se obtienen las primeras funcionalidades tales como la autenticación de usuario, adicionar, modificar y eliminar las metodologías de desarrollo de software, artículos y herramientas de metodologías de desarrollo de software, así como la primera versión del sistema.

2.5.2.2. Iteración 2

En esta iteración se implementan las historias de usuario de prioridad alta correspondientes con el adicionar, modificar y eliminar de información del modelo de Boehm y Turner, proponer la metodología según los datos entrados por el usuario, así como los reportes de todas las informaciones existentes en el sistema, Además se corrigen los errores de las HU de la iteración anterior. De esta forma se tiene la segunda versión del producto.

2.5.2.3. Iteración 3

En esta iteración se implementan las HU de prioridad media, relacionadas con las descargas de las herramientas, el adicionar, modificar y eliminar los enlaces a otros sitios y la realización de búsqueda de información existente en el sistema. De esta forma se obtiene la versión 1.0 final del producto. Esta versión se pondrá en funcionamiento en el Centro de Identificación y Seguridad Digital para la evaluación de su comportamiento y rendimiento.

2.5.3. Duración de las iteraciones

La metodología XP para que exista mayor organización a la hora de desarrollar un software crea un plan de duración de las iteraciones, en este caso se realizaría un solo plan ya que existe un único equipo de desarrolladores.

El plan tiene como objetivo fundamental mostrar la iteración en que será implementada cada HU según el orden correspondiente a cada una de ellas, así como el tiempo que se ha destinado para cada iteración.

Tabla 6: Plan de duración de las iteraciones

Iteración	Orden de la HU a Implementar	Duración total de la iteración
-----------	------------------------------	--------------------------------

Iteración # 1	<ol style="list-style-type: none">1. Autenticar Usuario.2. Adicionar metodología de desarrollo.3. Modificar metodología de desarrollo.4. Eliminar metodología de desarrollo.5. Adicionar artículo.6. Modificar artículo.7. Eliminar artículo.8. Adicionar herramientas.9. Modificar herramientas.10. Eliminar herramientas.	5
Iteración # 2	<ol style="list-style-type: none">1. Adicionar información sobre Modelo de Boehm y Turner.2. Modificar información sobre Modelo de Boehm y Turner.3. Eliminar información sobre Modelo de Boehm y Turner.4. Sugerir metodología de desarrollo según Modelo de Boehm y Turner.5. Realizar reportes de todas las informaciones del sistema	5
Iteración # 3	<ol style="list-style-type: none">1. Adicionar enlaces a otros sitios.2. Modificar enlaces a otros sitios.3. Eliminar enlaces a otros sitios.	3

	<ol style="list-style-type: none">4. Realizar búsquedas de la información existente en el sistema.5. Descargas de las herramientas.	
--	------------------------------------------------------------------------------------------------------------------------------------------------------------	--

2.5.4. Plan de entregas

En el plan de entrega que se plantea a continuación se hace una propuesta de la fecha aproximada en que se harán versiones (rleases) al sistema, al finalizar cada iteración en la fase de implementación.

Tabla 7: Plan de entrega

Producto a Entregar	Final 1ra Iteración 1ra semana de abril	Final 2da Iteración 2da semana de mayo	Final 3ra Iteración Última semana de mayo
InfoMet	Versión 0.1	Versión 0.2	Versión 1.0

2.6. Diseño del sistema

La metodología XP propone un diseño simple, donde siempre se intenta tener el código más simple, menos redundante y con las funcionalidades estrictamente necesarias en el presente.

En este epígrafe se dará una breve explicación del funcionamiento de Drupal como Sistema de Gestión de Contenidos.

Existen varios tipos de contenidos que se pueden publicar usando Drupal. Muchos de estos están organizados en lo que llamamos nodos. La palabra nodo es sinónimo de tipos de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo, que puede ser variable e incluir artículos, historias, encuestas, imágenes, libros colaborativos, reseñas, recetas, etcétera. Este nuevo concepto nos permite estandarizar la información asignándoles las mismas características a distintos objetos y la posibilidad de tener toda la información catalogada.

Drupal posee un potente sistema de seguridad basado en roles, lo cual permite controlar el acceso a los diferentes módulos según los permisos que el administrador le haya definido a cada rol.

2.6.1. Estructura del CMS Drupal

Drupal para la configuración propia de un sitio cuenta con 4 ficheros principales: Includes, Scripts, Themes y Modules, cada uno de estos ficheros contiene a su vez gran cantidad de archivos los cuales se hará referencia a los mismos con el término paquete.

Paquete Includes: Incluye archivos que contienen código PHP, y que serán llamados por el core de drupal para la realización de tareas diversas. Además se encuentra las clases de acceso a dato.

Paquete Scripts: Contiene un conjunto de ficheros indispensables para el funcionamiento de la aplicación, fundamentalmente orientados al aspecto visual, CSS y JavaScript.

Paquete Themes: Este paquete contiene las distintas plantillas que permiten configurar el diseño de toda la capa de presentación del sistema; cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

Paquete Modules: En él están ubicados todos los módulos que permiten extender las distintas funcionalidades del Drupal, para incorporar un nuevo módulo sólo se tiene que copiar en el directorio donde se encuentran los mismos.

A continuación se dará una breve descripción de los módulos que se incluyeron para el desarrollo del sistema.

- ❖ **CCK:** Para la creación de tipos de contenido personalizado.
- ❖ **Views:** Para obtención de vistas de los contenidos existentes.
- ❖ **LDAP_Integration:** Para la autenticación de un dominio externo.
- ❖ **Image:** Permite el tratamiento de imágenes. Para agregar un campo de tipo imagen para la creación de contenidos.
- ❖ **Search:** Para realizar búsquedas de contenidos existentes.
- ❖ **Menú:** Para gestionar los diferentes menús.

❖ **Taxonomy:** Para la categorización del contenido.

En el presente trabajo de diploma se modelará solamente el paquete Modules del CMS Drupal, ya que este es el que contiene las principales funcionalidades.

Básicamente Drupal está estructurado por una única página servidora. Esta página servidora muestra todo el contenido generada en la página final, esto está en dependencia de los argumentos con que se realicen la petición. Estas páginas generadas por Drupal pueden o no contener formularios, esto depende del módulo sobre el que se esté trabajando y el propósito del mismo. A continuación se representa en la **Ilustración 3** el funcionamiento de Drupal como aplicación.

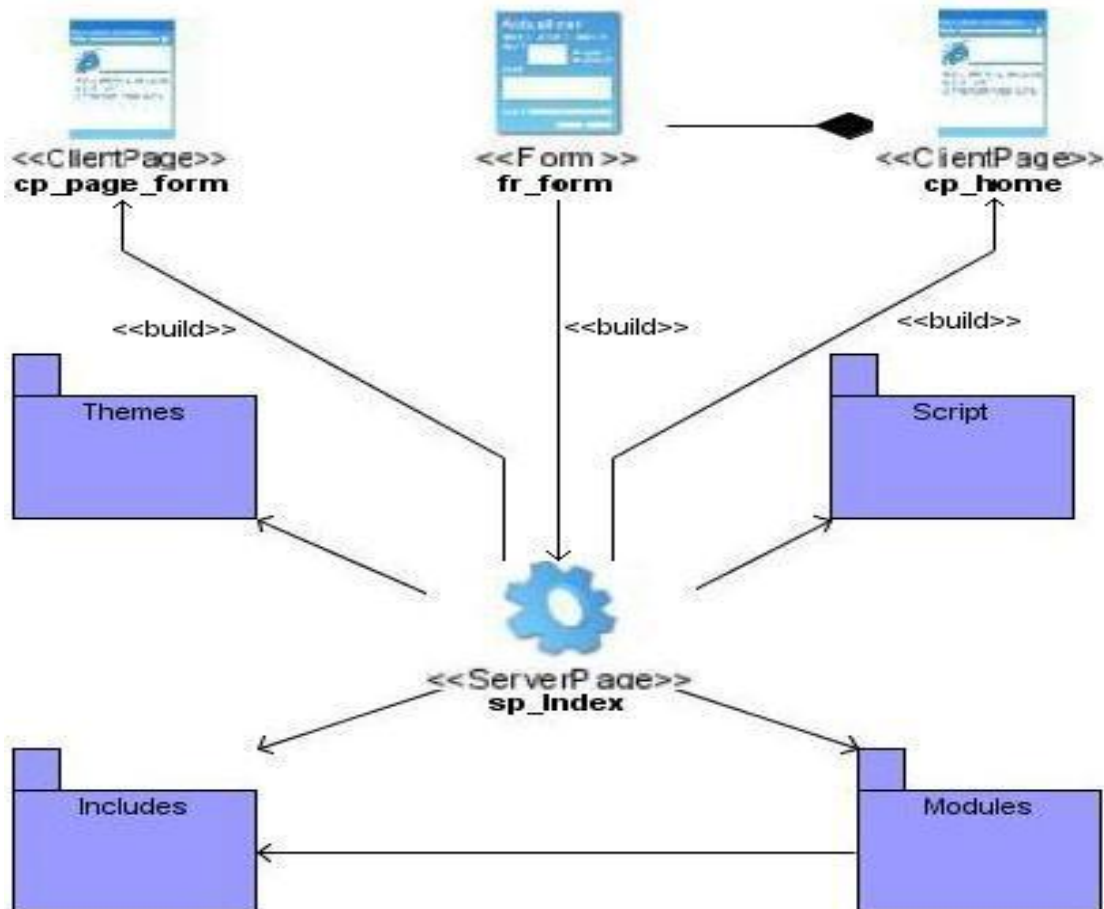


Ilustración 3- Diagrama de clases paquete Drupal.

2.6.2. Principales módulos de soporte a usar

CCK (Content Construction Kit): Permite al administrador del sitio, a través de la interfaz web, crear campos personalizados. Incluye campos como email, dirección, referencia a nodos, y muchos más para cualquier tipo de contenido. Esto significa que podemos crear fácilmente un campo de texto. Este nuevo campo de texto permite, al creador de tener un mecanismo más centrado de entrada y permite a los usuarios finales un formato de visualización.

Views: El módulo Views nos permite generar vistas para poder mostrar el contenido que queramos, filtrado y organizado a través de una interfaz que hace la tarea relativamente sencilla. Este módulo es esencialmente un constructor de consultas inteligentes, que proporcionándole la suficiente información, es capaz de construir la consulta adecuada, ejecutarla y mostrar los resultados.

Para el desarrollo de este trabajo se propone el uso de este módulo ya que es muy útil para mostrar listados de contenidos.

LDAP_Integration: Este Modulo cuenta con todas las funciones necesarias para el trabajo con el protocolo LDAP y se integra perfectamente con el dominio de la Universidad. Se usa para controlar el acceso y autenticación al sistema y los niveles de permisos.

Esto no quiere decir que la autenticación y la seguridad en cuanto a los niveles de privilegios en el sistema dependen del directorio activo, en caso de no existir, la plataforma de Drupal tiene sus propios mecanismos de autenticación, módulos que brindan todas las funcionalidades necesarias para proporcionar una seguridad adecuada, como por ejemplo el módulo User.

En el desarrollo de la presente investigación se tendrá en cuenta este módulo, ya que el CMS Drupal debe integrarse con el directorio activo de la universidad mediante el protocolo LDAP.

Search: Permite indexar el contenido del sitio de forma que se puedan realizar búsquedas sobre el mismo. Recupera y visualiza la información referida a contenido, usuarios y otros criterios de búsqueda.

Taxonomy: Constituye uno de los módulos más fuertes y más populares de Drupal debido a sus características y los beneficios que brinda. Permite a los usuarios que estén autorizados etiquetar

contenido usando etiquetas personalizadas y clasificar automáticamente nuevos contenidos basados en esta taxonomía.

2.6.3. Patrones de Diseño en Drupal

Los patrones del diseño tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes. [22]

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Drupal para su funcionamiento hace uso de algunos patrones de diseño que son propiamente característicos de sistemas orientados a objetos, particularmente los patrones GOF que contribuyen a la construcción de un diseño más elegante y robusto.

Dentro de los patrones GOF existen varias categorías, a continuación se describen las que destacan en la propuesta de solución.

- **Creacionales:** Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. [22]
 - ❖ Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Si se piensan los módulos como objetos entonces siguiendo este patrón, se puede mencionar que en general estos objetos no encapsulan datos, lo que separa un módulo de otro es el conjunto de funciones que contiene, garantizando de esta manera la existencia de una única instancia de este objeto y la creación de un mecanismo de acceso global a dicha instancia, por lo que debe ser tratado como una clase con una instancia *singleton*.

- **Estructurales:** Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.[22]

- ❖ *Decorator* (Envoltorio): Añade funcionalidad a una clase dinámicamente.

EL uso de *hook_nodeapi*, permite a los módulos extender arbitrariamente el comportamiento de todos los nodos. Para los archivos cargados y adjuntados a un nodo, se podría diseñar un nuevo tipo de nodo con las demás características del nodo, que brinde la posibilidad de cargar archivos con la concesión de que cada nodo solicite la posibilidad de tener archivos adjuntos.

- **Comportamiento:** Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. [22]

- ❖ *Observer* (Observador): Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

Cuando se hace una modificación a un vocabulario del sistema de taxonomía de Drupal, el gancho taxonomía se llama en todos los módulos que lo implementan. Por la implementación del gancho, estos módulos quedan como observadores del objeto vocabulario; cualquier cambio que sufra este objeto, los módulos dependientes del objeto, actúan tanto como sea apropiado.

- ❖ *Chain of Responsibility* (Cadena de responsabilidad): Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada

En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, y qué función se llama para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa a otro eslabón de la cadena. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado.

Otro de los patrones determinante es el Reflection Pattern, que describe el comportamiento de todo sitio basado en Drupal. Mediante este patrón es que el sistema se convierte en una aplicación muy adaptable a diferentes entornos ya que permite prácticamente la modificación de todo su comportamiento a través de los módulos instalables, sin la necesidad de modificar el núcleo.

2.6.4. Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: [23]

A continuación se representa en la **Ilustración 4** el diagrama de despliegue de la aplicación.

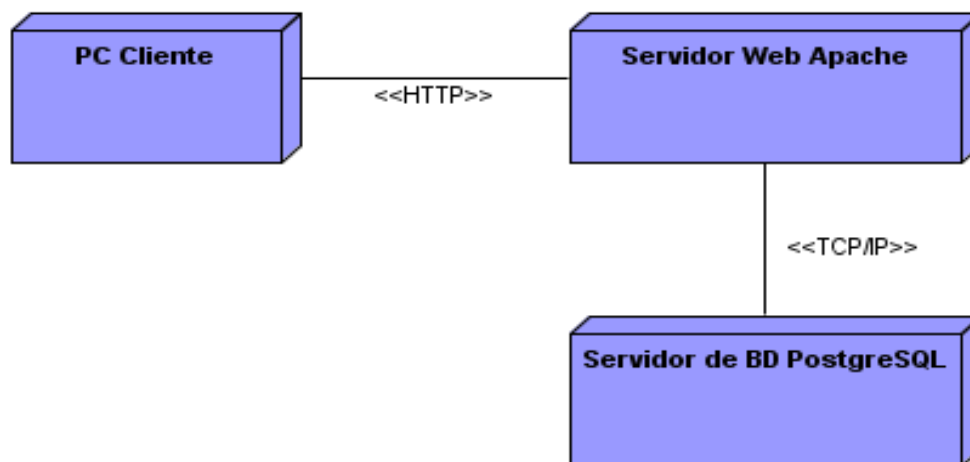


Ilustración 4- Diagrama de despliegue

2.7. Conclusiones

Como parte del presente capítulo se abordó todo lo referente a las primeras fases de la metodología de desarrollo de software XP, exploración y planificación, describiendo cada uno de los artefactos generados durante el transcurso de las mismas. Se hizo referencia a la etapa de diseño del sistema, describiéndose la estructura del CMS Drupal, así como los principales módulos a usar en la propuesta de solución y los patrones de diseño que utiliza este CMS para su funcionamiento.

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Capítulo III.

3.1. Introducción

En el presente capítulo se hace referencia a las prácticas de la metodología XP y lo importante que es seguir cada uno de sus pasos, así como a las tareas de ingeniería implementadas por cada historia de usuario. Además se describen las pruebas funcionales, las que permitieron evaluar que el sistema cumpla con los objetivos propuestos y funcione con la calidad requerida.

3.2. Implementación del sistema

La implementación es la parte más importante en la programación extrema. XP sugiere un modelo de trabajo usando repositorios de código donde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones. Para mantener un código consistente, publicar un código en un repositorio es una acción exclusiva para cada pareja de programadores.

XP también propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo. La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

3.2.1. Prácticas de XP

XP para facilitar una producción efectiva plantea una serie de prácticas básicas, las cuales se deben seguir disciplinadamente, ya que son necesarias para el desarrollo exitoso de un proyecto. Es imprescindible que estas sean todas realizadas, porque si se deja de implementar por lo menos alguna de ellas se corre el riesgo de que fracase el proyecto. [24]

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

A continuación se describen estas prácticas.

- ❖ **El juego de la planificación:** Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente y Programador.
- ❖ **Entregas pequeñas:** La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero sí que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- ❖ **Metáfora:** En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe como debería funcionar el sistema.
- ❖ **Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. 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 clases y métodos.
- ❖ **Pruebas:** La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

- ❖ **Refactorización:** La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo.
- ❖ **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores. las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema.
- ❖ **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.
- ❖ **Integración continua:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.
- ❖ **Semana de 40 horas:** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

- ❖ **Cliente in-situ:** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.
- ❖ **Estándares de programación:** XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

La programación extrema propone una forma iterativa para la implementación de un software junto a estas prácticas, da como resultado que al terminar cada iteración se obtenga una versión del producto funcional, éste debe ser probado y mostrado al cliente sirviendo de retroalimentación para el equipo de trabajo.

Este capítulo expone detalladamente las tres iteraciones generadas por la planificación descrita anteriormente, así como las tareas de ingeniería que se plantearon para la realización de cada una de las historias de usuarios y las pruebas que se efectuaron al sistema.

3.3. Iteración 1

En esta iteración se implementan las historias de usuarios del sistema que presentan mayor prioridad con el objetivo de obtener la primera versión del producto, así como las principales funcionalidades.

Para consultar las Tareas de Ingeniería generadas por cada historia de usuario de esta primera iteración, ver **Anexo 3**.

Tabla 8: Historias abordadas en la primera iteración

No.	Historia de Usuario	Estimación	Real
1	Autenticar Usuario.	1	1
2	Adicionar metodología de desarrollo.	1	1

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

3	Modificar metodología de desarrollo.	1	1
4	Eliminar metodología de desarrollo.	1	1
5	Adicionar artículos.	1	1
6	Modificar artículos.	1	1
7	Eliminar artículos.	1	1
8	Adicionar herramientas.	1	1
9	Modificar herramientas.	1	1
10	Eliminar herramientas.	1	1
Total		10	10

3.4. Iteración 2

Durante esta iteración se implementan las historias de usuarios que presentan nivel de prioridad alto para la construcción del sistema. De esta forma se agrega nuevas funcionalidades al sistema por lo que se obtiene una versión más completa.

Para consultar las Tareas de Ingeniería generadas por cada historia de usuario en esta segunda iteración, ver **Anexo 3**.

Tabla 9: Historias abordadas en la segunda iteración

No.	Historia de Usuario	Estimación	Real
11	Adicionar información sobre Modelo de Boehm y Turner.	1	1
12	Modificar información sobre Modelo de Boehm y Turner.	1	1

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

13	Eliminar información sobre Modelo de Boehm y Turner.	1	1
14	Sugerir metodología de desarrollo según Modelo de Boehm y Turner.	1	1
15	Realizar Reportes.	1	1
Total		5	5

3.5. Iteración 3

En esta última iteración se implementan las últimas funcionalidades del sistema, brindando una versión final del producto listo para pasar a ser probado, y posteriormente ponerlo en funcionamiento.

Para consultar las Tareas de Ingeniería generadas por cada historia de usuario en esta tercera iteración, ver **Anexo 3**.

Tabla 10: Historias abordadas en la tercera iteración

No.	Historia de Usuario	Estimación	Real
8	Adicionar enlaces a otros sitios.	1	1/2
	Modificar enlaces a otros sitios.	1	1/2
	Eliminar enlaces a otros sitios.	1	1/2
9	Realizar búsquedas de la información existente en el sistema.	1	1/2
10	Descargas de Herramientas.	1	1
Total		5	3

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

3.5. Pruebas

Uno de los pilares de XP es el proceso de pruebas. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. XP divide las pruebas del sistema en dos grupos: pruebas unitarias y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. [25]

3.5.1. Pruebas Unitarias

Son las encargadas de verificar el código y diseñada por los programadores. Consisten en comprobaciones manuales o automatizadas que se realizan para verificar que el código correspondiente a un módulo concreto de un sistema software, funciona de acuerdo con los requisitos del sistema. [26]

3.5.2. Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.

Se llevan a cabo sobre la interfaz del software y son completamente indiferentes al comportamiento interno y a la estructura del programa. Los casos de prueba de la caja negra pretende demostrar que:

- ❖ Las funciones del software son operativas.
- ❖ La entrada se acepta de forma adecuada.
- ❖ Se produce una salida correcta.
- ❖ La integridad de la información externa se mantiene.

Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa. La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- ❖ Funciones incorrecta o ausente.
- ❖ Errores de interfaz.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y de terminación. [27]

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Las pruebas de caja negra permiten confirmar que la HU ha sido implementada correctamente al final de cada iteración. Se definirán las entradas al sistema y los resultados esperados de estas entradas. Una HU puede tener todas las pruebas que necesite para asegurar su correcto funcionamiento.

A continuación se describen los casos de pruebas realizados a las principales funcionalidades. Para consultar los restantes casos de pruebas, ver **Anexo 4**.

Pruebas realizadas para las HU del sistema:

Tabla 11- Caso de prueba: Autenticar Usuario

Caso de Prueba	
Código: HU1_P1	Historia de Usuario: Autenticar Usuario.
Nombre: Autenticar Usuario.	
Descripción: Prueba de funcionalidad de la autenticación del usuario.	
Condiciones de ejecución: Debe estar previamente activado y configurado el módulo LDAP_Integration y el sub-módulo <i>Idapauth</i> . Los usuarios que tienen acceso al sistema deben estar creados y configurados con sus respectivos permisos.	
Entrada / Pasos de Ejecución: El usuario procede a la autenticación en el sistema.	
Resultado esperado: Una vez autenticado correctamente, el usuario accede al sistema y a la información que este tendrá acceso en dependencia del rol que tenga.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Autenticar Usuario			No: 1
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Introducción correcta de las credenciales del	Una vez entrados los datos del usuario que intenta acceder al sistema,	Satisfactorio	

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

usuario: Usuario: rmora Contraseña: *****	al presionar en el botón aceptar, se mostrará la página principal de la aplicación con las opciones que el usuario tendrá acceso en dependencia de su rol.		
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Introducción incorrecta de las credenciales del usuario: Usuario: vacío Contraseña: vacío	Muestra el mensaje de error "El campo contraseña es obligatorio".	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 12- Caso de prueba: Adicionar metodología

Caso de Prueba	
Código: HU2_P1	Historia de Usuario: Adicionar metodologías de desarrollo.
Nombre: Adicionar metodologías	
Descripción: Prueba de funcionalidad de adicionar metodología.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a introducir los datos específicos de cada metodología y se ejecuta la acción de adicionar metodología.	
Resultado esperado: La metodologías se ha adicionado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Caso de prueba: Adicionar metodologías			No: 2
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta los datos de la metodología que va a adicionar.	El sistema muestra un mensaje: La metodología se ha creado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario deja campos vacíos al insertar los datos de la metodología que va a adicionar.	El sistema muestra un mensaje: Todos los campos son obligatorios.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 13- Caso de prueba: Modificar metodología de desarrollo

Caso de Prueba	
Código: HU3_P1	Historia de Usuario: Modificar metodologías de desarrollo.
Nombre: Modificar metodologías de desarrollo de software.	
Descripción: Prueba de funcionalidad de modificar metodología.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a modificar los datos de la metodología y se ejecuta la acción de modificar metodología.	
Resultado esperado: La metodologías se ha modificado correctamente.	

Caso de prueba: Modificar metodologías de desarrollo de software.	No: 3
--------------------------------------------------------------------------	--------------

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar la metodología e introduce los datos que serán modificados.	El sistema muestra un mensaje: La metodología ha sido actualizada.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar la metodología y deja campos vacíos.	El sistema muestra un mensaje: Todos los campos deben ser llenados.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 14- Caso de prueba: Eliminar metodología de desarrollo

Caso de Prueba	
Código: HU4_P1	Historia de Usuario: Eliminar metodologías de desarrollo.
Nombre: Eliminar metodologías.	
Descripción: Prueba de funcionalidad de eliminar metodología.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a eliminar los datos de la metodología y se ejecuta la acción de eliminar metodología.	
Resultado esperado: La metodologías se ha eliminado correctamente.	

Caso de prueba: Eliminar metodologías.			No: 4
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

El usuario selecciona la opción de eliminar metodología y posteriormente la metodología específica que desea eliminar.	Se muestra el mensaje: ¿Desea eliminar la metodología? Una vez eliminada se muestra el mensaje: "Metodología ha sido borrada."	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 15- Caso de prueba: Adicionar artículos

Caso de Prueba	
Código: HU5_P1	Historia de Usuario: Adicionar artículo.
Nombre: Adicionar artículo.	
Descripción: Prueba de funcionalidad de adicionar artículo.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a introducir los datos específicos de cada artículo y se ejecuta la acción de adicionar artículo.	
Resultado esperado: El artículo se ha adicionado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Adicionar artículos.			No: 5
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta los datos del artículo que va a adicionar.	El sistema muestra un mensaje: El artículo se ha creado	Satisfactorio	

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario deja campos vacíos al insertar los datos del artículo.	El sistema muestra un mensaje: Todos los campos son obligatorios.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 16- Caso de prueba: Mostrar artículos

Caso de Prueba	
Código: HU6_P1	Historia de Usuario: Mostrar artículos.
Nombre: Mostrar artículos.	
Descripción: Prueba de funcionalidad de mostrar artículos.	
Condiciones de ejecución: Debe estar creada previamente la vista que permite el proceso.	
Entrada / Pasos de Ejecución: El usuario accede a la sección de reportes de artículos y selecciona el artículo deseado. Se ejecuta la acción y se muestra el artículo.	
Resultado esperado: El artículo se ha mostrado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Mostrar artículos.			No: 6
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona el menú Artículo y luego el artículo específico deseado.	Se muestra artículo seleccionado y una breve descripción.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Tabla 17- Caso de prueba: Modificar artículos

Caso de Prueba	
Código: HU6_P2	Historia de Usuario: Modificar artículos.
Nombre: Modificar artículos.	
Descripción: Prueba de funcionalidad de modificar artículos.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a modificar los datos necesarios de los artículos. Se ejecuta la acción de modificar artículos.	
Resultado esperado: El artículo se ha modificado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Modificar artículos.			No: 7
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar el artículo e introduce los datos que serán modificados.	El sistema muestra un mensaje: El artículo ha sido actualizado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar el artículo y deja campos vacíos.	El sistema muestra un mensaje: Todos los campos deben ser llenados.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Tabla 18- Caso de prueba: Eliminar artículos

Caso de Prueba	
Código: HU7_P1	Historia de Usuario: Eliminar artículos.
Nombre: Eliminar artículos.	
Descripción: Prueba de funcionalidad de eliminar artículos.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a eliminar los artículos. Se ejecuta la acción de eliminar artículos.	
Resultado esperado: El artículo se ha eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Eliminar artículos.			No: 8
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de eliminar artículo y posteriormente el artículo específico que desea eliminar.	Se muestra el mensaje: ¿Desea eliminar el artículo? Una vez eliminado se muestra el mensaje: "Artículo ha sido borrado."	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 19- Caso de prueba: Adicionar herramienta

Caso de Prueba	
Código: HU8_P1	Historia de Usuario: Adicionar herramienta.

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Nombre: Adicionar herramienta.
Descripción: Prueba de funcionalidad de adicionar herramienta.
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.
Entrada / Pasos de Ejecución: El usuario procede a adicionar los datos necesarios de la herramienta, incluyendo los archivos. Se ejecuta la acción de adicionar herramienta.
Resultado esperado: La herramienta se adiciona correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Adicionar herramienta.			No: 9
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta los datos de la herramienta que va a adicionar.	El sistema muestra un mensaje: La herramienta se ha creado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario deja campos vacíos al insertar los datos de la herramienta.	El sistema muestra un mensaje: Todos los campos son obligatorios.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 20- Caso de prueba: Modificar herramienta

Caso de Prueba	
Código: HU9_P1	Historia de Usuario: Modificar herramienta.
Nombre: Modificar herramienta.	
Descripción: Prueba de funcionalidad de modificar herramienta.	

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.
Entrada / Pasos de Ejecución: El usuario procede a modificar los datos necesarios de la herramienta seleccionada, incluyendo los archivos. Se ejecuta la acción de modificar herramienta.
Resultado esperado: Los datos de la herramienta se modifican correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Modificar herramienta.			No: 10
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar la herramienta e introduce los datos que serán modificados.	El sistema muestra un mensaje: La herramienta ha sido actualizada.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar la herramienta y deja campos vacíos.	El sistema muestra un mensaje: Todos los campos deben ser llenados.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 21- Caso de prueba: Eliminar herramienta

Caso de Prueba	
Código: HU10_P1	Historia de Usuario: Eliminar herramienta.
Nombre: Eliminar herramienta.	
Descripción: Prueba de funcionalidad de eliminar herramienta.	

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.
Entrada / Pasos de Ejecución: El usuario procede a eliminar la herramienta seleccionada. Se ejecuta la acción de eliminar herramienta.
Resultado esperado: La herramienta se elimina correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Eliminar herramienta.			No: 11
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de eliminar herramienta, y posteriormente la herramienta específica que desea eliminar.	Se muestra el mensaje: ¿Desea eliminar la herramienta? Una vez eliminada se muestra el mensaje: "Herramienta ha sido borrada."	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Resultados de las pruebas

Durante la etapa de prueba el sistema fue sometido a una primera iteración, realizándose once casos de prueba y se encontraron siete no conformidades, luego de corregir los errores encontrados se realizó una segunda iteración con un total de veinticinco casos de prueba en la cual no se encontraron no conformidades, lo que permite que el sistema esté listo para pasar por el departamento de calidad del centro.

3.6. Conclusiones

En el capítulo recién concluido se hizo referencia a las prácticas de la XP que debe seguirse disciplinadamente para el buen funcionamiento de un proyecto, se abordaron los artefactos generados por la metodologías de desarrollo de software XP, relacionados con la implementación del sistema, tales

Capítulo III: Implementación y pruebas funcionales de la propuesta de solución

como las Tareas de ingeniería, finalmente se realizaron las pruebas de caja negra al sistema para garantizar su correcto funcionamiento.

Conclusiones

Producto de la investigación que se llevó a cabo, se creó una aplicación web, para la gestión de información de metodologías de desarrollo de software con el objetivo de solucionar la gestión de información para los proyectos del CISED. Los objetivos planteados fueron cumplidos de manera satisfactoria obteniendo a su vez una serie de resultados:

- ❖ Con el análisis del estado del arte, se determinó la necesidad del desarrollo del sistema para la gestión de información de metodologías de desarrollo de software para el CISED.
- ❖ Se realizó un estudio de las principales herramientas y tecnologías más usadas a nivel mundial, lo que permitió desarrollar la aplicación utilizando PHP como lenguaje de programación y PostgreSQL como Sistema Gestor de Base de Datos.
- ❖ Con el estudio y análisis de las principales herramientas candidatas para el desarrollo de aplicación se llegó a la conclusión de utilizar el CMS Drupal.
- ❖ El proceso de desarrollo de software se realizó de manera satisfactoria guiado por la metodología XP. Se logró finalizar con el desarrollo del sistema propuesto, el cual fue probado durante su creación y validado mediante las pruebas de caja negra, fue desarrollado con tecnologías libres y proveído de elementos de seguridad garantizando la integridad y disponibilidad de la información asociada al sistema.

La aplicación Web muestra todas las informaciones relacionadas con las metodologías de desarrollo de software, descripciones, fases, expedientes de proyectos, artículos, así como descargas de herramientas. Posibilita al usuario la búsqueda de informaciones mediante el buscador, cuenta con una serie de vínculos a otros sitios y brinda la facilidad a editores de la aplicación, editar todos los contenidos necesarios. De esta manera se concluye que se cumplió el objetivo de la investigación, pues se desarrolló la aplicación web para la gestión de la información de metodologías de desarrollo de software.

Recomendaciones

Teniendo en cuenta que el principal objetivo de todo proyecto de software es entregar un producto con la calidad requerida para ser capaz de satisfacer las necesidades de los clientes y usuarios; se proponen las siguientes recomendaciones para asegurar el proceso de mejora continua las que contribuirían a un producto de mayor calidad.

- ❖ Desarrollar otras versiones de este sistema incorporando nuevas funcionalidades que se crea pueda ser útil para lograr un producto de mayor calidad.
- ❖ Implementar un módulo para proponer la metodología de desarrollo de software aumentándole nuevos criterios.
- ❖ Poner en funcionamiento el sistema para que los proyectos del centro lo utilicen.

Bibliografía referenciada

1. informationmanagement.wordpress. [En línea] 5 de Marzo de 2011. <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>.
2. definicion.de. [En línea] 5 de Marzo de 2011. <http://definicion.de/sistema-de-informacion/>.
3. informatizate.net. [En línea] 6 de marzo de 2011. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
4. scrum.es. [En línea] 6 de marzo de 2011. <http://scrum.es/>.
5. caraujo334. [En línea] 6 de marzo de 2011. <http://caraujo334.blogspot.es/1194206280/>.
6. mastermagazine.info. [En línea] 7 de Marzo de 2011. <http://www.mastermagazine.info/termino/7216.php>.
7. buenastareas.com. [En línea] 10 de Marzo de 2011. <http://www.buenastareas.com/ensayos/Definici%C3%B3n-y-Tipos-De-Aplicaciones-Web/317130.html>.
8. desarrolloweb.com. [En línea] 15 de marzo de 2011. <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>.
9. hugodonairemiles.blogspot.com. [En línea] 20 de marzo de 2011. <http://hugodonairemiles.blogspot.com/2006/11/accesibilidad-y-cms-drupal-una.html>.
10. maestrosdelweb. [En línea] 20 de marzo de 2011. <http://www.maestrosdelweb.com/editorial/phpintro/>.
11. www.desarrolloweb.com. [En línea] 25 de marzo de 2011. <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
12. .masadelante. [En línea] 1 de Abril de 2011. <http://www.masadelante.com/faqs/base-de-datos>.
13. terra.com. [En línea] 2 de Abril de 2011. <http://www.terra.com.mx/servicioacientes/articulo/118449/>.

14. scribd.com. [En línea] 2 de Abril de 2011. <http://es.scribd.com/doc/36570462/postgreSQL-investigacion>.
15. google.com. [En línea] 5 de Abril de 2011. http://www.google.com/url?sa=t&source=web&cd=10&ved=0CGwQFjAJ&url=http%3A%2F%2Fapuntes-utn.com.ar%2Fapuntes%2Fdocumento.aspx%3FIdDocumento%3D787%26bajar%3D1&rct=j&q=define%20rup&tbs=lr%3Alang_1es&ei=WFrdTZqNN8Tw0gHJ66z3Dw&usg=AFQjCNH39LMINX_9OBY-6KhM-1rSl.
16. google.com. [En línea] 5 de Abril de 2011. http://www.google.com/url?sa=t&source=web&cd=10&ved=0CGwQFjAJ&url=http%3A%2F%2Fapuntes-utn.com.ar%2Fapuntes%2Fdocumento.aspx%3FIdDocumento%3D787%26bajar%3D1&rct=j&q=define%20rup&tbs=lr%3Alang_1es&ei=WFrdTZqNN8Tw0gHJ66z3Dw&usg=AFQjCNH39LMINX_9OBY-6KhM-1rSl.
17. dosideas.com. [En línea] 5 de Abril de 2011. <http://www.dosideas.com/noticias/metodologias/729-scrum-en-pocas-palabras.html>.
18. willydev. [En línea] 9 de Abril de 2011. <http://www.willydev.net/descargas/masyxp.pdf>.
19. hostarting. [En línea] 12 de Abril de 2011. <http://blog.hostarting.es/entender-conceptos-servidor-web-apache>.
20. monografias. [En línea] 15 de Abril de 2011. <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
21. sourceforge. [En línea] 18 de Abril de 2011. <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
22. nfo-ab.uclm.es. [En línea] 21 de Abril de 2011. <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
23. ciberaula. [En línea] 23 de Abril de 2011. http://java.ciberaula.com/articulo/disenio_patrones_j2ee/.

24. buenastareas. [En línea] 24 de Abril de 2011. <http://www.buenastareas.com/ensayos/Diagrama-De-Despliegue/1023607.html> .
25. willydev. [En línea] 26 de Abril de 2011. <http://httpwww.willydev.netdescargasmasypdf>.
26. lsi.us.es. [En línea] 27 de Abril de 2011. http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf .
27. profepineda.wikispaces. [En línea] 10 de Mayo de 2011. <http://profepineda.wikispaces.com/file/view/file.TiposPruebasSoftware.pdf>.
28. monografias. [En línea] 29 de Abril de 2011. <http://www.monografias.com/trabajos36/pruebas-de-aceptacion/pruebas-de-aceptacion2.shtml> .
29. carlosmontalvo. [En línea] 10 de Abril de 2011. <http://www.carlosmontalvo.com/2010/12/aplicaciones-web-ventajas-y-desventajas>.
30. pilos.com. [En línea] 12 de Abril de 2011. <http://www.pilos.com.co/drupal/27-caracteristicas-de-drupal/>.
31. linuxcentro.ne. [En línea] 21 de Abril de 2011. <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
32. acsblog.es. [En línea] 23 de Abril de 2011. <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.
33. scribd.com. [En línea] Abril de 2011. <http://es.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
34. luisitob.wordpress. [En línea] Abril de 2011. <http://luisitob.wordpress.com/2008/07/22/tipos-de-cms/>.
35. [En línea] Mayo de 2011. http://www.google.com/imgres?imgurl=http://edn.embarcadero.com/article/images/33319/RUP.JPG&imgrefurl=http://carlitosdv.blogspot.com/2009/10/3-metodologia-rup.html&usq=__xEF2UQyf8_HhnKBfzxl2wkmOUkY=&h=420&w=623&sz=45&hl=es&start=7&zoom=1&itbs=1&tbnid=bDN7.

36 [En línea] mayo de 2011.
http://www.google.com/search?hl=es&biw=1024&bih=548&gbv=2&tbm=isch&sa=1&q=fases+metodologia+xp&oq=fases+metodologia+xp&aq=f&aql=&gs_sm=e&gs_upl=79941128211011211210101011016811320810.4.4.1.0.2.

Bibliografía consultada

- **Falguera, Benet Campderrich.** *Ingeniería de software.* Barcelona : UOC, 2003.
- **Amo, Fernando Alonso.** *Introducción ala Ugeniería de software.* Madrid : s.n., 2005.
- **Marcer, David.** *Creating Blogs, Forums, Portals, And Community Websites.* 2006.
- **VanDyk, John K.** *Pro Drupal Development.* 2008.
- **L., Netbiblo S.** *Técnicas Cuantitativas para la Gestión de Ingeniería de software.* 2007.
- **Weitzenfeld, Alfredo.** *Ingeniería de software.*
- **Laurie, Ben.** *Apache.* 2003.
- **Castillos, Carlos.** *Características de la Web Chilena.* 2007.
- **Bakore, Amit.** *Professional Apache Tomcat.* 2002.
- *DIGITAL CERTIFICATES.* **Jalal Fegghi, Jalil Fegghi y Peter Williams.**
- **Rojas, Juan Carlos Olivares.** *Patrones de Diseño.*
- **Johansen, Ernst.** *Patrones de codiseño para control avanzado .*
- **Tello, Jesús Cáceres.** *Patrones de diseño.* 2009.
- **García Ibáñez, Carlos.** *Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural.* 2004.
- **Molpeceres, Alberto.** *Procesos de desarrollo RUP, XP y FDD.* 2002.
- **Amaro Calderon, Sarah Damaris.** *Metodologías Ágiles.* Perú : s.n., 2007.

Glosario de términos

Blogs: también llamado *weblog* o bitácora, es un sitio Web fácil de crear y utilizar, el cual te permite, entre otras muchas cosas, publicar, compartir e interactuar contenidos que van desde tus conocimientos, noticias, artículos u opiniones sobre una determinada temática hasta simplemente tu propia historia de vida.

CASE: *Computer Aided Software Engineering* y en su traducción al Español significa Ingeniería de Software Asistida por Computación.

CCS: Hoja de estilos en cascada con sus siglas en inglés (*Cascading Style Sheets*). Se utiliza para dar estilo a documentos.

db_query: Es la función que utiliza Drupal para realizar sus consultas a la Base de Datos.

FTP: (*File Transfer Protocol*) es un medio específico de conexión de un Sitio web para cargar y descargar ficheros.

GOF: Gang Of Four es el nombre con el que se conoce comúnmente a los autores del libro *Design Patterns*. De esta forma de le llama a los patrones de diseño.

HTML: *Hyper Text Markup Language* (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto diseñado para la creación de páginas web.

HTTP: Siglas inglesas de *Hypertext Transfer Protocol*. Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la *World Wide Web*.

LDAP: *Lightweight Directory Access Protocol*, es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Mod_rewrite: Es un módulo de Apache que nos permite convertir URLs dinámicos a estáticos haciendo una redirección interna.

Mozilla Firefox: Es un navegador de Internet libre y de código abierto.

PHP: Lenguaje de programación.

UML: Lenguaje de modelado, es la notación principalmente gráfica que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.

URL: URL significa *Uniform Resource Locator*, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

WEB: Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

XHTML: *eXtensible Hypertext Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje pensado para sustituir a HTML como estándar para las páginas web.

XML: *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas, una manera de definir lenguajes para diferentes necesidades.

Anexos

Anexo 1: Operacionalización de las variables

Tabla 22-Operacionalización de las variables

Variable Independiente	Dimensión	Indicadores	Unidad de Medida
Desarrollo de una aplicación informática para el CISED.	Factibilidad	Tiempo de desarrollo	Extenso, Moderado Breve
		Costo	Costoso, Moderado Barato
		Esfuerzo	Alto, Moderado, Despreciable

Variable Independiente	Dimensión	Indicadores	Unidad de Medida
Gestionar la información de metodologías de desarrollo de software para el CISED.	Rendimiento	Complejidad	Alta, Media, Baja
		Control	Bueno, Malo
		Organización del Trabajo	Bueno, Malo
		Importancia	Alta, Media, Baja
		Disponibilidad	Alta, Media, Baja
		Seguridad	Alto, Moderado, Baja

Anexo 2: Historias de Usuarios

Tabla 23: HU Adicionar artículos

<i>Historia de Usuario</i>	
Número: 5	Nombre: Adicionar artículos.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1.
<p>Descripción: El usuario podrá adicionar los datos necesario del artículo:</p> <ul style="list-style-type: none"> • Nombre del artículo. • Autor del artículo. • Breve descripción del contenido del artículo. • Documento. 	
Observaciones:	

Tabla 24: HU Modificar artículos

<i>Historia de Usuario</i>	
Número: 6	Nombre: Modificar artículos.
Usuario: Editor	

Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1.
Descripción: El usuario modificar los datos necesarios del artículo:	
Observaciones:	

Tabla 25: HU Eliminar artículos

<i>Historia de Usuario</i>	
Número: 7	Nombre: Eliminar artículos.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1.
Descripción: El usuario eliminar los datos necesarios del artículo:	
Observaciones:	

Tabla 26: HU Adicionar herramienta

<i>Historia de Usuario</i>	
Número: 8	Nombre: Adicionar herramienta.

Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario podrá adicionar los datos de las herramientas. <ul style="list-style-type: none"> • Nombre de las herramientas. • Breve descripción de la herramienta. • Archivo adjunto de la herramienta. 	
Observaciones:	

Tabla 27: HU Modificar herramienta

<i>Historia de Usuario</i>	
Número: 9	Nombre: Modificar herramienta.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario podrá modificar los datos necesarios de las herramientas.	

Observaciones:

Tabla 28: HU Eliminar herramienta

<i>Historia de Usuario</i>	
Número: 10	Nombre: Eliminar herramienta.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario podrá eliminar los datos necesarios de la herramienta.	
Observaciones:	

Tabla 29: HU Adicionar información del Modelo de Boehm y Turner

<i>Historia de Usuario</i>	
Número: 11	Nombre: Adicionar información del Modelo de Boehm y Turner.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2

<p>Descripción: El usuario podrá adicionar los datos sobre Modelo de Boehm y Turner.</p> <ul style="list-style-type: none"> • Breve descripción del modelo • Grafico
<p>Observaciones:</p>

Tabla 30: HU Modificar información del Modelo de Boehm y Turner

<i>Historia de Usuario</i>	
Número: 12	Nombre: Modificar información del Modelo de Boehm y Turner.
Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
<p>Descripción: El usuario podrá modificar los datos sobre Modelo de Boehm y Turner.</p>	
<p>Observaciones:</p>	

Tabla 31: HU Eliminar información del Modelo de Boehm y Turner

<i>Historia de Usuario</i>	
Número: 13	Nombre: Eliminar información del Modelo de Boehm y Turner.

Usuario: Editor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Descripción: El usuario podrá eliminar los datos sobre Modelo de Boehm y Turner.	
Observaciones:	

Tabla 32: HU Sugerir metodología de desarrollo según Modelo de Boehm y Turner

<i>Historia de Usuario</i>	
Número: 14	Nombre: Sugerir metodología de desarrollo según Modelo de Boehm y Turner.
Usuario: Sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Descripción: El usuario procede a introducir los datos necesarios para según Modelo de Boehm y Turner realizar el cálculo correspondiente y el sistema sugerir la metodología de desarrollo a utilizar.	
Observaciones:	

Tabla 33: HU Realizar reportes de todas las informaciones del sistema

<i>Historia de Usuario</i>	
Número: 15	Nombre: Realizar reportes de todas las informaciones del sistema.
Usuario: Invitado	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Descripción: El usuario podrá acceder a todas las informaciones que brinda el sistema, tanto de las metodologías como de las herramientas, así como los artículos entre otras.	
Observaciones:	

Tabla 34: HU Adicionar enlaces a otros sitios

<i>Historia de Usuario</i>	
Número: 16	Nombre: Adicionar enlaces a otros sitios.
Usuario: Editor	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario podrá adicionar los datos de los enlaces.	

<ul style="list-style-type: none"> Nombre del enlace. Ruta del enlace.
Observaciones:

Tabla 35: HU Modificar enlaces a otros sitios

<i>Historia de Usuario</i>	
Número: 17	Nombre: Modificar enlaces a otros sitios.
Usuario: Editor	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario podrá modificar los datos de los enlaces. <ul style="list-style-type: none"> Nombre del enlace. Ruta del enlace. 	
Observaciones:	

Tabla 36: HU Eliminar enlaces a otros sitios

<i>Historia de Usuario</i>	
Número: 18	Nombre: Eliminar enlaces a otros sitios.

Usuario: Editor	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario podrá eliminar los enlaces.	
Observaciones:	

Tabla 37: HU Realizar búsquedas de la información existente en el sistema

<i>Historia de Usuario</i>	
Número: 19	Nombre: Realizar búsquedas de la información existente en el sistema.
Usuario: Invitados, Administrador, Editor.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario podrá realizar búsquedas de cualquier información que exista en el sistema.	
Observaciones:	

Tabla 38: HU Descargas de las herramientas

<i>Historia de Usuario</i>	
Número: 20	Nombre: Descargas de las herramientas.
Usuario: Invitados	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario podrá realizar descargar de las herramientas que se utilizan en el desarrollo de un software.	
Observaciones:	

Anexo 3: Tareas de ingeniería

Tareas de ingeniería para la primera iteración.

HU Autenticar usuario

Tabla 39: Tarea#1: HU Autenticar usuario

Tarea	
Número de la tarea: 1	Número de la historia: 1
Nombre de la tarea: Crear el usuario Editor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/7	Fecha fin: 2011/03/9

Programador responsable: Ediagnis Pérez Darromán
Descripción: Se creará un usuario Editor para asignarle los permisos de gestionar toda la información referente a las metodologías a través de la creación de un rol editor.

Tabla 40: Tarea#2: HU Autenticar usuario

Tarea	
Número de la tarea: 2	Número de la historia: 1
Nombre de la tarea: Agregar el módulo <i>LDAP_Integration</i> y activar el sub_módulo <i>ldapauth</i> .	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/9	Fecha fin: 2011/03/11
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se agregará el módulo <i>LDAP_Integration</i> en la carpeta de módulos del CMS y se activará el <i>sub_módulo ldapauth</i> . De esta forma los usuarios podrán registrarse por el dominio UCI.	

HU Adicionar metodologías de desarrollo.

Tabla 41: Tarea#1: HU Adicionar metodologías de desarrollo

Tarea	
Número de la tarea: 1	Número de la historia: 2

Nombre de la tarea: Crear el tipo de contenido metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/14	Fecha fin: 2011/03/14
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se creará el tipo de contenido metodologías de desarrollo para publicar de esta forma las informaciones de las metodologías utilizadas en el del Centro de Identificación y Seguridad Digital. Deberá estar activado previamente el módulo <i>CCK</i> .	

Tabla 42: Tarea#2: HU Adicionar metodologías de desarrollo

Tarea	
Número de la tarea: 2	Número de la historia: 2
Nombre de la tarea: Añadir los campos del tipo de contenido metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/14	Fecha fin: 2011/03/14
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se agregarán los campos necesarios del tipo de contenido metodologías de desarrollo insertando las informaciones específicas de cada una de ellas. Deberá estar activado previamente el módulo <i>CCK</i> .	




HU Modificar metodologías de desarrollo.

Tabla 43: Tarea#1: HU Modificar metodologías de desarrollo

Tarea	
Número de la tarea: 1	Número de la historia: 3
Nombre de la tarea: Crear vistas para modificar las metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/15	Fecha fin: 2011/03/15
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crean una vista para modificar las metodologías de desarrollo y se procederá a la configuración de la misma. Deberá estar activado previamente el módulo <i>Views</i> .	

Tabla 44: Tarea#2: HU Modificar metodologías de desarrollo

Tarea	
Número de la tarea: 2	Número de la historia: 3
Nombre de la tarea: Configurar la vista para modificar las metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/15	Fecha fin: 2011/03/15

Programador responsable: Ediagnis Pérez Darromán
<p>Descripción: Se configurará la vista para modificar las metodologías de desarrollo de software a través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de metodología a mostrar).  Campos a mostrar.




HU Eliminar metodologías de desarrollo.

Tabla 45: Tarea#1: HU Eliminar metodologías de desarrollo

Tarea	
Número de la tarea: 1	Número de la historia: 4
Nombre de la tarea: Crear la vista para eliminar las metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/16	Fecha fin: 2011/03/16
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crean una vista para modificar las metodologías de desarrollo y se procederá a la configuración de la misma. Deberá estar activado previamente el módulo <i>Views</i> .	

Tabla 46: Tarea#2: HU Eliminar metodologías de desarrollo

Tarea

Número de la tarea: 2	Número de la historia: 4
Nombre de la tarea: Configurar las vistas para eliminar las metodologías de desarrollo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/16	Fecha fin: 2011/03/16
Programador responsable: Ediagnis Pérez Darromán	
<p>Descripción: Se configurará la vista para eliminar las metodologías de desarrollo de software a través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de metodología a mostrar).  Campos a mostrar. 	

HU Adicionar artículos.

Tabla 47: Tarea#1: HU Adicionar artículos

Tarea	
Número de la tarea: 1	Número de la historia: 5
Nombre de la tarea: Crear el tipo de contenido artículos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/17	Fecha fin: 2011/03/17
Programador responsable: Ediagnis Pérez Darromán	

Descripción: Se creará el tipo de contenido artículos para publicar de esta forma las informaciones de los artículos de las metodologías de desarrollo utilizadas en el del Centro de Identificación y Seguridad Digital. Deberá estar activado previamente el módulo CCK.

Tabla 48: Tarea#2: HU Adicionar artículos

Tarea	
Número de la tarea: 2	Número de la historia: 5
Nombre de la tarea: Añadir los campos del tipo de contenido artículos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/17	Fecha fin: 2011/03/17
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se agregarán los campos necesarios del tipo de contenido artículos insertando las informaciones específicas de cada uno de ellos. Deberá estar activado previamente el módulo CCK.	




HU Modificar artículos.

Tabla 49: Tarea#1: HU Modificar artículos

Tarea	
Número de la tarea: 1	Número de la historia: 6
Nombre de la tarea: Crear vistas para modificar los artículos.	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/117	Fecha fin: 2011/03/117
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crea una vista para modificar los artículos y se procederá a la configuración de las mismas. Deberá estar activado previamente el módulo Views.	

Tabla 50: Tarea#2: HU Modificar artículos

Tarea	
Número de la tarea: 2	Número de la historia: 6
Nombre de la tarea: Configurar las vistas para modificar los artículos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/17	Fecha fin: 2011/03/17
Programador responsable: Ediagnis Pérez Darromán	
<p>Descripción: Se configurará la vista para modificar los artículos través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de artículo a mostrar).  Campos a mostrar. 	

HU Eliminar artículos.




Tabla 51: Tarea#1: HU Eliminar artículos

Tarea	
Número de la tarea: 1	Número de la historia: 7
Nombre de la tarea: Crear las vistas para eliminar los artículos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/18	Fecha fin: 2011/03/18
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crea una vistas para eliminar los artículos y se procederá a la configuración de las mismas. Deberá estar activado previamente el módulo Views.	

Tabla 52: Tarea#2: HU Eliminar artículos

Tarea	
Número de la tarea: 2	Número de la historia: 7
Nombre de la tarea: Configurar las vistas para eliminar los artículos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/18	Fecha fin: 2011/03/18
Programador responsable: Ediagnis Pérez Darromán	

Descripción: Se configurará la vista para eliminar los artículos.

-  Información básica.
-  Filtro (aquí se define el tipo de artículo a mostrar).
-  Campos a mostrar.

HU Adicionar herramienta.

Tabla 53: Tarea#1: HU Adicionar herramienta

Tarea	
Número de la tarea: 1	Número de la historia: 8
Nombre de la tarea: Crear el tipo de contenido herramienta.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/19	Fecha fin: 2011/03/19
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se creará el tipo de contenido herramienta para publicar de esta forma las informaciones de las mismas. Deberá estar activado previamente el módulo CCK.	

Tabla 54: Tarea#2: HU Adicionar herramienta

Tarea	
Número de la tarea: 2	Número de la historia: 8




Nombre de la tarea: Añadir los campos del tipo de contenido herramienta.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/19	Fecha fin: 2011/03/19
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se agregarán los campos necesarios del tipo de contenido herramienta insertando las informaciones específicas de cada una de ellas. Deberá estar activado previamente el módulo <i>CCK</i> .	

HU Modificar herramienta.

Tabla 55: Tarea#1: HU Modificar herramienta

Tarea	
Número de la tarea: 1	Número de la historia: 9
Nombre de la tarea: Crear vistas para modificar las herramientas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/20	Fecha fin: 2011/03/20
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crea una vistas para modificar las herramientas y se procederá a la configuración de las mismas. Deberá estar activado previamente el módulo <i>Views</i> .	

Tabla 56: Tarea#2: HU Modificar herramienta

Tarea	
Número de la tarea: 2	Número de la historia: 9
Nombre de la tarea: Configurar las vistas para modificar las herramientas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/20	Fecha fin: 2011/03/20
Programador responsable: Ediagnis Pérez Darromán	
<p>Descripción: Se configurará la vista para modificar las herramientas a través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de herramienta a mostrar).  Campos a mostrar. 	



HU Eliminar herramienta.

Tabla 57: Tarea#1: HU Eliminar herramienta

Tarea	
Número de la tarea: 1	Número de la historia: 10
Nombre de la tarea: Crear las vistas para eliminar las herramientas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/21	Fecha fin: 2011/03/21

Programador responsable: Ediagnis Pérez Darromán
Descripción: Se crea una vistas para eliminar las herramientas y se procederá a la configuración de las mismas. Deberá estar activado previamente el módulo <i>Views</i> .

Tabla 58: Tarea#2: HU Eliminar herramienta

Tarea	
Número de la tarea: 2	Número de la historia: 10
Nombre de la tarea: Configurar las vistas para eliminar herramienta.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/04/21	Fecha fin: 2011/04/21
Programador responsable: Ediagnis Pérez Darromán	
<p>Descripción: Se configurará la vista de las herramientas de desarrollo de software a través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de herramienta a mostrar). 	

Tareas de ingeniería para la segunda iteración.

HU Adicionar información sobre Modelo de Boehm y Turner.

Tabla 59: Tarea#1: HU Adicionar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 1	Número de la historia: 11
Nombre de la tarea: Crear el tipo de contenido Modelo de Boehm y Turner.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/04/22	Fecha fin: 2011/04/22
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se creará el tipo de contenido Modelo de Boehm y Turner para publicar informaciones sobre las mismas. Deberá estar activado previamente el módulo CCK.	

Tabla 60: Tarea#2: HU Adicionar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 2	Número de la historia: 11
Nombre de la tarea: Añadir los campos al tipo de contenido Modelo de Boehm y Turner.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/04/22	Fecha fin: 2011/04/22
Programador responsable: Ediagnis Pérez Darromán	

Descripción: Se añaden los campos necesarios al tipo de contenido Modelo de Boehm y Turner, para especificar lo que se va a mostrar de este contenido. Deberá estar activado previamente el módulo CCK.



HU Modificar información sobre Modelo de Boehm y Turner.

Tabla 61: Tarea#1: HU Modificar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 1	Número de la historia: 12
Nombre de la tarea: Crear vistas para modificar el tipo de contenido Modelo de Boehm y Turner.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/04/23	Fecha fin: 2011/04/23
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crean una vista para contenido Modelo de Boehm y Turner, posteriormente de procederá a su configuración. Deberá estar activado previamente el módulo Views.	

Tabla 62: Tarea#2: HU Modificar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 2	Número de la historia: 12
Nombre de la tarea: Configurar las vistas para modificar el tipo de contenido Modelo de Boehm y Turner	
Tipo de tarea: Desarrollo	Puntos estimados: 1




Fecha inicio: 2011/04/23	Fecha fin: 2011/04/23
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se configurará la vista del Modelo de Boehm y Turner a través de las siguientes opciones: <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de Modelo a mostrar). 	

HU Eliminar información sobre Modelo de Boehm y Turner.

Tabla 63: Tarea#1: HU Eliminar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 1	Número de la historia: 13
Nombre de la tarea: Crear las vistas para eliminar la información sobre Modelo de Boehm y Turner.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/24	Fecha fin: 2011/03/24
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crearán dos vistas para eliminar la información sobre Modelo de Boehm y Turner, se procederá a la configuración de las mismas. Deberá estar activado previamente el módulo <i>Views</i> .	

Tabla 64: Tarea#2: HU Eliminar información sobre Modelo de Boehm y Turner

Tarea	
Número de la tarea: 2	Número de la historia: 13
Nombre de la tarea: Configurar las vistas para eliminar la información sobre Modelo de Boehm y Turner.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/03/24	Fecha fin: 2011/03/24
Programador responsable: Ediagnis Pérez Darromán	
<p>Descripción: Se configurará la vista para eliminar la información sobre Modelo de Boehm y Turner a través de las siguientes opciones:</p> <ul style="list-style-type: none">  Información básica.  Filtro (aquí se define el tipo de modelo a mostrar).  Campos a mostrar. 	

HU Sugerir metodología de desarrollo según Modelo de Boehm y Turner.

Tabla 65: Tarea#1: HU Sugerir metodología de desarrollo según Modelo de Boehm y Turner

Tarea	
Número de la tarea: 1	Número de la historia: 14
Nombre de la tarea: Llenar los campos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha inicio: 2011/04/24	Fecha fin: 2011/04/24
Programador responsable: Ediagnis Pérez Darromán	
Descripción: El usuario debe llenar todos los campos para que el sistema le proponga una metodología de desarrollo a utilizar.	

HU Realizar reportes de todas las informaciones del sistema.

Tabla 66: Tarea#1: HU Realizar reportes de todas las informaciones del sistema

Tarea	
Número de la tarea: 1	Número de la historia: 15
Nombre de la tarea: Crear menú.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2011/05/2	Fecha fin: 2011/05/4
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se creará un menú donde se le adiciona un reporte por cada contenido que se va a mostrar en el sistema, configurando las opciones de cada uno.	

Tabla 67: Tarea#2: HU Realizar reportes de todas las informaciones del sistema

Tarea

Número de la tarea: 2	Número de la historia: 15
Nombre de la tarea: Crear vistas para los reportes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/4	Fecha fin: 2011/05/6
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crean las vistas para mostrar los reportes, posteriormente se procedes a su configuración.	

Tareas de ingeniería para la tercera iteración.

HU Adicionar enlaces a otros sitios.

Tabla 68: Tarea#1: HU Adicionar enlaces a otros sitios

Tarea	
Número de la tarea: 1	Número de la historia: 16
Nombre de la tarea: Crear menú.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/9	Fecha fin: 2011/05/11
Programador responsable: Ediagnis Pérez Darromán	

Descripción: Se creará un menú donde se le adiciona un enlace por cada sitio a referenciar, configurando las opciones de cada uno.

HU Modificar enlaces a otros sitios.

Tabla 69: Tarea#1: HU Modificar enlaces a otros sitios

Tarea	
Número de la tarea: 1	Número de la historia: 17
Nombre de la tarea: Modificar enlaces.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/9	Fecha fin: 2011/05/11
Programador responsable: Ediagnis Pérez Darromás	
Descripción: Se modifican los enlaces creados anteriormente en el menú.	

HU Eliminar enlaces a otros sitios.

Tabla 70: Tarea#1: HU Eliminar enlaces a otros sitios

Tarea	
Número de la tarea: 1	Número de la historia: 18
Nombre de la tarea: Eliminan enlaces.	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/9	Fecha fin: 2011/05/11
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se eliminan los enlaces creados anteriormente en el menú.	

HU Realizar búsquedas de la información existente en el sistema.

Tabla 71: Tarea#1: HU Realizar búsquedas de la información existente en el sistema

Tarea	
Número de la tarea: 1	Número de la historia: 19
Nombre de la tarea: Activar módulo Search.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/11	Fecha fin: 2011/05/11
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se activará el módulo Search para permitir la búsqueda de la información existente en el sistema. Previamente deberá estar activado el módulo poormanscron.	

HU Descargas de Herramientas.

Tabla 72: Tarea#1: HU Descargas de Herramientas

Tarea	
Número de la tarea: 1	Número de la historia: 20
Nombre de la tarea: Crear menú de descargas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/12	Fecha fin: 2011/05/15
Programador responsable: Ediagnis Pérez Darromán	
Descripción: Se crea un menú donde se adicionan las herramientas para descargar.	

Tabla 73: Tarea#2: HU Descargas de Herramientas

Tarea	
Número de la tarea: 2	Número de la historia: 20
Nombre de la tarea: Descargar las herramientas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2011/05/15	Fecha fin: 2011/05/17
Programador responsable: Ediagnis Pérez Darromán	

Descripción: El usuario accede a la descargar de la herramienta seleccionada. Debe estar previamente adicionada la herramienta.

Anexo 4: Casos de Pruebas

Tabla 74- Caso de prueba: Adicionar modelo

Caso de Prueba	
Código: HU11_P1	Historia de Usuario: Adicionar información sobre Modelo de Boehm y Turner.
Nombre: Adicionar Modelo.	
Descripción: Prueba de funcionalidad de adicionar modelo.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a adicionar los datos pertenecientes al modelo. Se ejecuta la acción de adicionar modelo.	
Resultado esperado: El modelo es adicionado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Adicionar Modelo.			No: 12
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta los datos del modelo que va a adicionar.	El sistema muestra un mensaje: El modelo se ha creado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario deja campos vacíos al insertar los datos.	El sistema muestra un mensaje: Todos los campos son obligatorios	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 75- Caso de prueba: Modificar modelo

Caso de Prueba	
Código: HU12_P1	Historia de Usuario: Modificar información sobre Modelo de Boehm y Turner.
Nombre: Modificar Modelo.	
Descripción: Prueba de funcionalidad de modificar modelo.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a modificar los datos pertenecientes al modelo. Se ejecuta la acción de modificar modelo.	
Resultado esperado: El modelo es modificado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Modificar Modelo.			No: 13
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar el modelo e introduce los datos que serán modificados.	El sistema muestra un mensaje: El modelo ha sido actualizado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar el modelo y deja campos vacíos.	El sistema muestra un mensaje: Todos los campos deben ser llenados.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 76- Caso de prueba: Eliminar modelo

Caso de Prueba	
Código: HU13_P1	Historia de Usuario: Eliminar información sobre Modelo de Boehm y Turner.
Nombre: Eliminar Modelo.	
Descripción: Prueba de funcionalidad de eliminar modelo.	
Condiciones de ejecución: El usuario debe tener los permisos pertinentes para acceder a esta funcionalidad.	
Entrada / Pasos de Ejecución: El usuario procede a eliminar el modelo. Se ejecuta la acción de eliminar modelo.	
Resultado esperado: El modelo es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Eliminar Modelo.			No: 14
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de eliminar modelo y posteriormente el modelo específico que desea eliminar.	Se muestra el mensaje: ¿Desea eliminar el modelo? Una vez eliminado se muestra el mensaje: "Modelo ha sido borrado."	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 77: - Caso de prueba: Sugerir metodología

Caso de Prueba	
Código: HU14_P1	Historia de Usuario: Sugerir metodología de desarrollo según Modelo de Boehm y Turner.

Nombre: Sugerir metodología.
Descripción: Prueba de funcionalidad de proponer metodología.
Condiciones de ejecución: El usuario debe llenar todos los campos para que se ejecute la acción de proponer la metodología.
Entrada / Pasos de Ejecución: El usuario procede a llenar los campos y luego se propone una metodología de desarrollo de software.
Resultado esperado: La propuesta de la metodología se hace correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Sugerir metodología.			No: 15
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario llena todos los campos propuestos para que el sistema sugiera una metodología.	El sistema muestra una lista de metodologías según los criterios seleccionados.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario deja campos sin llenar.	El sistema muestra un mensaje: Seleccione los criterios para generar una propuesta de metodología	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 78- Caso de prueba: Reporte de metodología

Caso de Prueba	
Código: HU15_P1	Historia de Usuario: Realizar reportes de todas las informaciones del sistema.
Nombre: Reporte de metodología.	

Descripción: Prueba de funcionalidad de reporte de metodología.
Condiciones de ejecución: Las informaciones de las metodologías deben estar publicadas.
Entrada / Pasos de Ejecución: El usuario accede a la sección de Reportes, selecciona la opción de Reporte de metodologías, se ejecuta esta acción mostrándose las informaciones de las metodologías de desarrollo de software.
Resultado esperado: El reporte de las metodologías se muestra correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Reporte de metodología.			No: 16
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario accede al menú de Reportes, selecciona la opción de Metodologías.	Se muestran todas las metodologías existentes en el sistema.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 79- Caso de prueba: Reporte de artículos

Caso de Prueba	
Código: HU15_P2	Historia de Usuario: Realizar reportes de todas las informaciones del sistema.
Nombre: Reporte de Artículos.	
Descripción: Prueba de funcionalidad de reporte de artículos.	
Condiciones de ejecución: Los artículos deben estar publicados.	
Entrada / Pasos de Ejecución: El usuario accede a la sección de Reportes, selecciona la opción de Reporte de artículos, se ejecuta esta acción mostrándose los artículos.	
Resultado esperado: El reporte de los artículos se muestra correctamente.	

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Reporte de Artículos.			No: 17
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario accede al menú de Reportes, selecciona la opción de Artículos.	Se muestran todos los artículos existentes en el sistema.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 80- Caso de prueba: Reporte de modelo

Caso de Prueba	
Código: HU15_P3	Historia de Usuario: Realizar reportes de todas las informaciones del sistema.
Nombre: Reporte de Modelo.	
Descripción: Prueba de funcionalidad de reporte de modelo.	
Condiciones de ejecución: Las informaciones del modelo deben estar publicadas.	
Entrada / Pasos de Ejecución: El usuario accede a la sección de Reportes, selecciona la opción de Reporte de modelo, se ejecuta esta acción mostrándose las informaciones del Modelo de Boehm y Turner.	
Resultado esperado: El reporte del Modelo de Boehm y Turner se muestra correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Reporte de Modelo.			No: 18
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario accede al menú de Reportes, selecciona la	Se muestran todos los modelos existentes en el	Satisfactorio	

opción Modelo.	sistema.		
Evaluación de la prueba:	Satisfactoria		

Tabla 81- Caso de prueba: Adicionar enlace

Caso de Prueba	
Código: HU16_P1	Historia de Usuario Adicionar enlaces a otros sitios.
Nombre: Adicionar enlace.	
Descripción: Prueba de funcionalidad de adicionar enlace.	
Condiciones de ejecución: El usuario autenticado debe tener los permisos pertinentes.	
Entrada / Pasos de Ejecución: El usuario procede a insertar los datos necesarios y se ejecuta la acción adicionar enlace.	
Resultado esperado: El enlace es adicionado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Adicionar enlace.			No: 19
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta la ruta del enlace que va a adicionar.	El sistema muestra un mensaje: El enlace se ha creado.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario inserta la ruta incorrecta del enlace.	El sistema muestra un mensaje: La ruta del enlace no existe.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 82-Caso de prueba: Mostrar enlace

Caso de Prueba	
Código: HU17_P1	Historia de Usuario Modificar enlaces a otros sitios.
Nombre: Mostrar enlace.	
Descripción: Prueba de funcionalidad mostrar enlace.	
Condiciones de ejecución: Los enlaces deben estar activados.	
Entrada / Pasos de Ejecución: El usuario accede a la sección Enlaces a Sitios y se muestran los enlaces.	
Resultado esperado: Los enlaces son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Mostrar enlace.			No: 20
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona el enlace a otro sitio deseado.	El sistema muestra el sitio escogido.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 83- Caso de prueba: Modificar enlace

Caso de Prueba	
Código: HU17_P2	Historia de Usuario Modificar enlaces a otros sitios.
Nombre: Modificar enlace.	
Descripción: Prueba de funcionalidad de modificar enlace.	
Condiciones de ejecución: El usuario autenticado debe tener los permisos pertinentes.	

Entrada / Pasos de Ejecución: El usuario procede a modificar los datos necesarios y se ejecuta la acción modificar enlace.
Resultado esperado: El enlace es modificado correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba: Modificar enlace.			No: 21
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona la opción de modificar el enlace y luego procede a modificar los datos del mismo.	El enlace se modifica correctamente.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 84- Caso de prueba: Mostrar contenido del enlace

Caso de Prueba	
Código: HU17_P3	Historia de Usuario Modificar enlaces a otros sitios.
Nombre: Mostrar contenido del enlace.	
Descripción: Prueba de funcionalidad mostrar contenido del enlace.	
Condiciones de ejecución: Se debe acceder a un enlace previamente.	
Entrada / Pasos de Ejecución: El usuario accede a un enlace deseado y se visualiza el contenido del enlace.	
Resultado esperado: Los enlaces son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Mostrar contenido del enlace.	No: 22
------------------------------------------------------	---------------

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona el enlace que desea ver.	El sistema muestra el sitio del enlace seleccionado.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 85- Caso de prueba: Eliminar enlace

Caso de Prueba	
Código: HU18_P1	Historia de Usuario Eliminar enlaces a otros sitios.
Nombre: Eliminar enlace.	
Descripción: Prueba de funcionalidad eliminar enlace.	
Condiciones de ejecución: El usuario autenticado debe tener los permisos pertinentes.	
Entrada / Pasos de Ejecución: El usuario procede a eliminar el enlace, se ejecuta la acción eliminar enlace.	
Resultado esperado: El enlace es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Eliminar enlace.			No: 23
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona el enlace que desea borrar y luego la opción eliminar.	Se muestra el mensaje: ¿Desea eliminar el enlace? Una vez eliminado se muestra el mensaje: "Enlace ha sido borrado."	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 86- Caso de prueba: Buscar contenido

Caso de Prueba	
Código: HU19_P1	Historia de Usuario: Realizar búsqueda de la información existente en el sistema.
Nombre: Buscar contenido.	
Descripción: Prueba de funcionalidad buscar contenido.	
Condiciones de ejecución: El usuario debe tener los permisos de acceso a este servicio.	
Entrada / Pasos de Ejecución: El usuario procede a ingresar la palabra clave relacionada con el contenido a buscar y se ejecuta la acción buscar contenido.	
Resultado esperado: La búsqueda es realizada correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Buscar contenido.			No: 24
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario introduce la palabra que desea buscar en la aplicación.	El sistema muestra el contenido que desea el usuario.	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario escribe incorrecta la palabra que desea buscar.	El sistema muestra un mensaje: Debe incluir palabra calve.	Satisfactorio	
Evaluación de la prueba:	Satisfactoria		

Tabla 87- Caso de prueba: Mostar contenidos encontrados

Caso de Prueba	
Código: HU19_P2	Historia de Usuario: Realizar búsqueda de la información existente en el sistema.

Nombre: Mostrar contenidos encontrados.
Descripción: Prueba de funcionalidad mostrar contenidos encontrados.
Condiciones de ejecución: El usuario debe ingresar la palabra clave previamente.
Entrada / Pasos de Ejecución: El usuario visualiza el resultado de la búsqueda realizada.
Resultado esperado: Los contenidos encontrados se muestran correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 88- Caso de prueba: Descargas

Caso de Prueba	
Código: HU20_P1	Historia de Usuario: Descargas de herramientas.
Nombre: Descargas.	
Descripción: Prueba de funcionalidad de descargas.	
Condiciones de ejecución: El usuario debe tener los permisos para realizar esta acción.	
Entrada / Pasos de Ejecución: El usuario procede a la descarga de la herramienta seleccionada.	
Resultado esperado: La herramienta ha sido descargada.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba: Descargas.			No: 25
Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
El usuario selecciona el menú Descargas y posteriormente la herramienta que desea descargar.	La herramienta es descargada correctamente.	Satisfactorio	

Evaluación de la prueba:	Satisfactoria
---------------------------------	---------------

Anexo 5: Imágenes de la aplicación

Inicio de sesión

Usuario: *

Contraseña: *

Iniciar sesión

[Crear nueva cuenta](#)

[Solicitar una nueva contraseña](#)

Quién está en línea

Actualmente hay 0 usuarios y 0 invitados en línea.

Enlaces a Sitios

- Akademos
- FEU
- Facultad 1

Bienvenido al Sistema de Información de Metodologías

El siguiente sistema gestiona la información referente a las metodologías de desarrollo de software para los proyectos del CISED.

El sistema muestra todas las informaciones relacionadas con las metodologías de desarrollo de software, descripciones, fases, expedientes de proyectos, artículos, así como descargas de herramientas. Posibilita al usuario la búsqueda de informaciones mediante el buscador, cuenta con una serie de vínculos a otros sitios y la da la facilidad a editores de la aplicación de editar todos los contenidos necesarios.

Ilustración 5: Página principal de la aplicación para la gestión de información de metodologías



InfoMet
InfoMet

Buscar

Inicio

Buscar en este sitio:

Buscar

Quién está en línea

Actualmente hay 1 usuario y 0 invitados en línea.

Usuarios en línea

- edarroman

Reportes

- Reporte de Metodologías
- Reporte de Artículos
- Reporte de Modelo

Descargas

- Descargas de Herramientas

eXtreme Programming (XP)

Lun, 06/06/2011 - 09:53 | Edignis

Tamaño de Proyecto

Proyecto:
Cortos

Clasificación de la Metodología

Metodología:
Ágil

Descripción:
Esta es una de las metodologías de desarrollo de software más utilizada en la actualidad, para proyectos de corto plazo y corto equipo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final como uno de los requisitos para llegar al éxito del proyecto. En esta metodología, el cliente se convierte en un miembro más del equipo de trabajo y es el encargado de decidir que se implementa, puede añadir, cambiar o quitar requerimientos en cualquier momento para lo cual debe estar enterado constantemente del estado real y el progreso del proyecto obteniendo lo máximo de cada semana de trabajo.

Fases:

Ilustración 6: Página de la metodología XP

InfoMet
InfoMet

[Inicio](#)

Metodología

Seleccione los criterios para generar una propuesta de metodología

Personal: ...Seleccione...
 Críticidad: ...Seleccione...
 Tamaño: ...Seleccione...
 Cultura: ...Seleccione...
 Dinamismo: ...Seleccione...

Logo	Metodología
	XP (eXtreme Programming)
	Scrum
	MSF for agil software developmet (Microsoft Solutions Framework)
	FDD (Feature Driven Development)
	MSF for CMMI (Microsoft Solutions Framework)

Ilustración 7: Página para sugerir las metodologías