



Universidad de las Ciencias Informáticas
Facultad 1

Aplicación de soporte a la Metodología de desarrollo de distribuciones GNU/Linux, Nova-OpenUp.

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

Autor(a):

Isabel Barbara Sanchez Quiala

Tutor(a):

**Ms. C. Yusleydi Fernández del Monte
Ing. Mairim Delgado Muñiz**

**La Habana
junio 2014**

Declaración de Autoría

Declaro ser la autora del presente Trabajo de Diploma y se reconoce a la Universidad de las Ciencias Informáticas, los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría en Ciudad de La Habana a los _____ días del mes de _____ del año _____.

Isabel Barbara Sanchez Quiala

Firma del autor(a)

MsC. Yusleydi Fernández del Monte

Firma del Tutor(a)

Ing. Mairim Delgado Muñiz

Firma de Tutor(a)

“Tu trabajo va a llenar gran parte de tu vida, la única manera de estar realmente satisfecho es hacer lo que creas, es un gran trabajo y la única manera de hacerlo es amar lo que haces. Si no lo has encontrado aún, sigue buscando. Como con todo lo que tiene que ver con el corazón, sabrás cuando lo hayas encontrado.”

Steve Jobs

Agradecimientos

Quiero agradecer a todas las personas importantes en mi vida y darle las gracias, ellas son:

Mi mamá, por ser tan especial, por tu amor, tu comprensión y por darme fuerzas cuando más la necesitaba.

A mi papá y a mis hermanos por creer en mi.

Quiero agradecer a mis Erza Sans: Marlen, Mayvis, Dayana, Yamilka y Daynelis, por estar siempre conmigo en los buenos y malos momentos.

A mi equipo de Hunters, el cual está conformado por Oniel, Emil, Guillermo, Pepe, Charlie y todos los chicos de mi grupo, en especial Daniel, Yosvany, Tapia, Yoel, Javier y Hanoi; porque ustedes me han enseñado que los chicos y las chicas pueden ser grandes amigos.

A mis tutoras por haberme ayudado en cada momento que lo he necesitado, durante todo el proceso de desarrollo de la tesis.

Resumen

La metodología Nova-OpenUp constituye el eslabón fundamental en el ciclo de vida de la distribución de GNU/Linux Nova. Está conformada por un conjunto de disciplinas compuestas por actividades, roles y artefactos. Define buenas prácticas de las metodologías pesadas y ágiles, y brinda la posibilidad de configurarla según el producto a construir. Actualmente los desarrolladores de Nova-OpenUp y de las metodologías como OpenUp, RUP y SCRUM, ven el desarrollo web como el escenario para proporcionar una mayor interacción entre los elementos que las conforman y los equipos de proyectos que las utilizan. Además de posibilitar el reconocimiento de la metodología en la industria de software.

La presente investigación describe el desarrollo de una aplicación web para la metodología Nova-OpenUp, que integra servicios informativos y de socialización de la información, con el objetivo de ofrecer un espacio de asesoría y consulta a aquellos usuarios que emplean Nova-OpenUp como guía en el proceso de desarrollo de software. Para la implementación de la propuesta de solución se utiliza el Sistema de Gestión de Contenidos Drupal, PostgreSQL como Sistema Gestor de Base de Datos y Apache como servidor web. Se emplea la metodología SXP para guiar el proceso de desarrollo de la aplicación. Se realizan pruebas de funcionalidad y pruebas de aceptación para verificar el cumplimiento de las funcionalidades. Como principal resultado se obtiene un producto funcional, capaz de apoyar el proceso de desarrollo de la distribución de GNU/Linux Nova, incidiendo en las formas de utilización de la metodología Nova-OpenUp.

Palabras claves: metodología, aplicación web, software libre

Índice de contenido

Introducción.....	8
Capítulo 1: Fundamentación Teórica de las aplicaciones de soporte a las metodologías de desarrollo de software.....	12
1.1 Introducción.....	12
1.2 Definiciones de interés.....	12
1.3 Arquitectura de la Información.....	14
1.4 Metodología de desarrollo de software.....	14
1.5 Tecnologías, herramientas y lenguajes a utilizar.....	18
1.5.1 Sistemas de Gestión de Contenido (CMS).....	18
1.5.2 Lenguajes de desarrollo.....	21
1.5.3 Servidores web.....	22
1.5.4 Sistema de Gestión de Base de datos (SGBD).....	23
1.5.5 Herramientas a utilizar	25
1.6 Proceso de soporte.....	27
1.7 Aplicaciones web para brindar soporte a las metodologías de desarrollo de software.....	28
1.8 Conclusiones del capítulo.....	31
Capítulo 2: Requisitos y Arquitectura de la aplicación web para la Metodología Nova-OpenUp.....	32
2.1 Introducción.....	32
2.2 Descripción de la aplicación.....	32
2.2.1 Modelo de dominio.....	32
2.3 Requisitos funcionales y no funcionales del sistema.....	33
2.3.1 Lista de requisitos funcionales.....	33
2.3.2 Requisitos no funcionales.....	35
2.4 Propuesta de la aplicación de soporte a la metodología Nova-OpenUp.....	37
2.5 Historias de usuarios.....	38
2.6 Descripción de la Arquitectura de software y los patrones de diseño.....	41
2.6.1 Arquitectura de software.....	41
2.6.2 Patrones de diseño.....	42
2.7 Diagrama de Clases.....	44
2.8 Diagrama de paquetes.....	46
2.9 Modelo de despliegue.....	47
2.10 Conclusiones del capítulo.....	48
Capítulo 3: Programación y Evaluación de la aplicación web para la Metodología Nova-OpenUp.....	49
3.1 Introducción.....	49
3.2 Plan de Iteraciones.....	49
3.3 Tareas de Ingeniería.....	50
3.4 Estándar de codificación.....	52
3.5 Casos de Pruebas de aceptación.....	52
3.6 Pruebas de carga y estrés.....	56
3.7 Pantallas de la aplicación.....	58
3.8 Conclusiones del Capítulo.....	59
Conclusiones generales.....	60

Recomendaciones.....	61
Bibliografía.....	62
Glosario de términos.....	65
Anexo 1.....	66
Anexo 2	67
Anexo 3	72
Anexo 4	74
Anexo 5	77
Anexo 6	79
Anexo 7.....	81

Introducción

En el mundo moderno hay un desarrollo cada vez más acelerado de la Informática y las Tecnologías de la Información (TI). Estas constituyen el núcleo central de una transformación multidimensional que experimenta la economía y la sociedad; y es ahí donde los servicios de soporte representan un elemento crucial para garantizar el éxito de los productos informáticos que persiguen obtener una ventaja competitiva y mejorar la calidad.

Los servicios de soporte se desarrollan cada vez más, es por eso que muchos proyectos productores de software se han dado la tarea de crear grupos de soporte con especialistas en el tema, capaces de resolver las inquietudes del cliente, usando los procedimientos y herramientas correctas con la mayor rapidez posible. A nivel internacional muchos de estos proyectos brindan soporte por cada uno de los productos que se ofrecen.

Entre los aspectos fundamentales que debe tener en cuenta el proyecto a la hora de desarrollar un software, se encuentra la selección de la metodología; pues estas determinan un proceso completo y detallado que implica un conjunto de herramientas, procedimientos, técnicas y soporte documental, permitiendo obtener un producto que cumpla las expectativas de los usuarios. En este entorno se debe tener en cuenta que no existe una metodología de desarrollo de software que sea global, ya que la selección de la misma depende de las características de cada proyecto, del equipo de desarrollo, los requisitos y de los recursos que se dispongan.

La Universidad de las Ciencias Informáticas (UCI), como pilar en la construcción de software del país realiza importantes acciones en esta área, tal es el caso de la metodología de desarrollo de distribuciones de GNU/Linux, Nova-OpenUp, creada en el Centro de Desarrollo de Soluciones Libres (CESOL), orientada como su nombre lo indica al desarrollo de distribuciones o una personalización de esta. La metodología posee la característica fundamental de ser altamente configurable por la diferencia de entornos de desarrollo de software que pueden coexistir en el departamento Sistemas Operativos.

Sin embargo actualmente la metodología se encuentra en un documento que no permite la interacción dinámica entre esta y los usuarios que la necesitan; disponible solamente en el repositorio al que no todos los implicados en el desarrollo de la distribución de GNU/Linux Nova pueden acceder. El equipo de desarrollo solo por medio del correo electrónico y a través de

reuniones sincroniza sus intereses, lo que implica una excesiva utilización del correo, tornándose difícil la gestión del conocimiento entre los involucrados. Adicionalmente se carece de medios para proporcionar elementos técnicos, para responder a todas aquellas consultas que requieran de asesoramiento, así como la dificultad para configurar la metodología por equipos inexpertos y noveles en temas de desarrollo de software.

Por lo anteriormente planteado surge el siguiente **Problema a resolver**: ¿Cómo brindar soporte a la metodología Nova-OpenUp para el desarrollo de la distribución de GNU/Linux Nova?

Por tanto **el objeto de estudio** se centra en el proceso de soporte de las metodologías de desarrollo de software y **el campo de acción** está enmarcado en el desarrollo de aplicaciones web destinadas a brindar soporte a las metodologías de desarrollo de software.

Para dar solución a la situación problemática mencionada anteriormente se plantea como **Objetivo general** desarrollar una aplicación web que brinde soporte a la metodología Nova-OpenUp para el desarrollo de la distribución de GNU/Linux Nova, planteando como **objetivos específicos**:

- Caracterizar la fundamentación teórica de la utilización de aplicaciones web que brindan soporte a las metodologías de desarrollo de software.
- Desarrollar una propuesta de solución para la construcción de una aplicación web que permita dar soporte a la metodología Nova-OpenUp.
- Evaluar el correcto funcionamiento de la aplicación de soporte a la metodología Nova-OpenUp.

La investigación se sustenta en la siguiente **Idea a defender**: El desarrollo de una aplicación web puede brindar soporte a la metodología Nova-OpenUp y apoyar el proceso de desarrollo de la distribución de GNU/Linux Nova.

Para cumplir con los objetivos específicos se trazan las siguientes **tareas de investigación**:

- Revisión bibliográfica de las aplicaciones web destinadas a brindar soporte a las metodologías de desarrollo de software para determinar las implicaciones que tienen este tipo de aplicaciones en el proceso de soporte de las metodologías de desarrollo de software.

- Análisis de las herramientas, metodologías y tecnologías actuales de desarrollo de aplicaciones web para seleccionar las más factibles en el desarrollo de la propuesta de solución.
- Obtención de los requisitos funcionales y no funcionales para el desarrollo de la aplicación de soporte a la metodología Nova-OpenUp.
- Programación de la aplicación de soporte a la metodología Nova-OpenUp para lograr su capacidad operacional.
- Ejecución de pruebas para evaluar la correctitud de la aplicación de acuerdo a los requisitos especificados.

A partir de lo antes planteado los **resultados esperados son:**

- Obtener una retroalimentación con los usuarios, que posibilite mejorar la calidad de los productos elaborados por los proyectos que siguen la línea de desarrollo de la distribución de GNU/Linux Nova.
- Contribuir a mejorar los procesos de soporte y atención a los usuarios finales de la metodología de desarrollo de software Nova-OpenUp.

Métodos científicos

Teóricos:

El método Analítico–Sintético es utilizado con el fin de analizar un conjunto de bibliografías, posibilitando la extracción de los elementos más importantes que sustentan la investigación.

El método histórico-lógico es empleado con el objetivo de verificar teóricamente cómo ha evolucionado el tema de las aplicaciones web que se dedican a publicar y difundir información acerca de las metodologías de desarrollo de software y de este modo poder realizar la selección de las herramientas y técnicas que se van a utilizar de acuerdo a las características propias de la aplicación a desarrollar.

El método inductivo-deductivo es utilizado en el estudio del comportamiento de aplicaciones particulares encargadas de contribuir al soporte de metodologías de desarrollo de software y de este modo obtener los requisitos fundamentales que deben estar presentes en la aplicación a desarrollar.

El método de modelado es empleado en la confección de los diagramas que permiten representar la propuesta de solución y a partir de estos se implementan las funcionalidades de la aplicación.

Empíricos:

La entrevista se utiliza con el objetivo de obtener información valiosa sobre los flujos de trabajo de los proyectos involucrados en el desarrollo de la distribución de GNU/Linux Nova y organización de la información de la metodología Nova-OpenUp, para el desarrollo eficiente de esta investigación. Ver anexo 1.

Este documento se encuentra organizado en tres capítulos, a continuación se muestra una breve descripción de los mismos:

En el **primer capítulo** titulado **Fundamentación teórica de las aplicaciones de soporte a las metodologías de desarrollo de software**, se definen los principales conceptos asociados al dominio del problema que son indispensables para el desarrollo y comprensión de esta investigación. Contiene además un análisis completo para definir las herramientas y tecnologías que se van a utilizar en el desarrollo de la aplicación web.

En el **segundo capítulo** titulado **Requisitos y Arquitectura de la aplicación web para la metodología Nova-OpenUp**, se realiza todo lo vinculado con la identificación de los requisitos y la arquitectura del sistema, describiendo de manera más detallada los elementos que se deben obtener en cada una de las fases propuestas por la metodología de desarrollo a utilizar.

En el **tercer capítulo** titulado **Programación y Evaluación de la aplicación web para la metodología Nova-OpenUp**, se explican temas referentes a la implementación de la solución. Además se realizan los casos de pruebas para lograr el desarrollo de un software con la calidad requerida.

Capítulo 1: Fundamentación Teórica de las aplicaciones de soporte a las metodologías de desarrollo de software.

1.1 Introducción

En el presente capítulo se muestran las principales definiciones relacionadas con la investigación, tales como, aplicación web, sitio web y portal web; con el objetivo de lograr una mejor comprensión de la solución al problema. Se realiza un estudio del estado del arte de aplicaciones web dedicadas a brindar soporte a las metodologías de desarrollo de software. Se justifica la selección de las herramientas, tecnologías y lenguajes de programación a emplear para la construcción de la solución.

1.2 Definiciones de interés

Aplicación Web

Existen variedades de definiciones de aplicación web. A continuación se muestran algunas que dan varios autores.

Una aplicación web es un conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor web, incluidas bases de datos. Esta interacción permite implementar características en su sitio como catálogos de productos virtuales, administradores de noticias y contenidos. Adicionalmente se pueden realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real (1).

Las aplicaciones web son programas diseñados para utilizarse íntegramente en el navegador. Permiten crear documentos, editar fotos y escuchar música, entre otras funcionalidades, sin tener que instalar ningún software complejo. En la actualidad, las aplicaciones web pueden realizar las mismas funciones dinámicas que las aplicaciones para ordenador (2).

De manera general estas definiciones encierran características afines, que concretan el desarrollo actual y progresivo de estas tecnologías. Por tanto se puede concretar que las aplicaciones web son programas que permiten la interacción dinámica, entre los usuarios y la información o servicios, estas pueden ser, desde aplicaciones encargadas de gestionar la información hasta aplicaciones complejas, capaces de realizar las mismas funciones que los programas para ordenador.

Sitio Web

Un sitio web es un espacio virtual en Internet. Se trata de un conjunto de páginas web que son accesibles desde un mismo dominio o subdominio de la *World Wide Web* (WWW). Los sitios web incluyen documentos HTML, fotografías, sonidos, videos, animaciones *flash* y otro tipo de contenidos que pueden compartirse en línea (3).

Como medio, los sitios web son similares a las películas, a la televisión o a las revistas, en que también crean y manipulan imágenes digitales y texto, pero un sitio web es también un medio de comunicación. La diferencia principal entre un sitio web y los medios tradicionales es que un sitio web está en una red de ordenadores y está codificado de manera que permite que los usuarios interactúen con él (2).

Se puede definir que un sitio web es un conjunto de páginas web en la Internet, que contiene información de consulta de algún tipo, de estas no se espera interacción mayor con el lector, lo que se espera es que encuentre la información que esté buscando.

Portal Web:

Un portal de Internet es un sitio web cuya característica fundamental es la de servir de puerta de entrada (única) para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente un portal en Internet está dirigido a resolver necesidades de información específica de un tema en particular (2).

Portal es un término, sinónimo de puente, para referirse a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan a la *World Wide Web*. Son sitios que los usuarios tienden a visitar como sitios ancla. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre comunidades (3).

Los portales web están dirigidos principalmente a resolver las necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución pública o privada. Un portal es más bien una plataforma de despegue para la navegación en la Web donde se reúne toda la información que los usuarios necesitan en un único lugar para acceder a ella de forma coherente con el objetivo de aumentar la productividad. El objetivo principal del Portal es ayudar a los usuarios a encontrar lo que necesitan sin salir del mismo, fidelizándoles e

incentivándoles a utilizarlo de forma continuada.

Para dar solución a la problemática planteada se escoge desarrollar una aplicación web, pues debido a las características que encierra, ayuda a los usuarios en el procesamiento de tareas específicas y se logra mayor interacción con la metodología, permitiéndoles seleccionar la variante que más se adapte a las características de su proyecto.

1.3 Arquitectura de la Información

Con el objetivo de obtener toda la información posible de la metodología Nova-OpenUp y del producto a desarrollar se tuvieron en cuenta elementos importantes de la Arquitectura de la Información.

“La Arquitectura de la Información se refiere al diseño, organización, etiquetado, navegación y sistemas de búsqueda que ayudan a los usuarios a encontrar y gestionar la información de manera efectiva”. (Lou Rosenfeld y Peter Morville).

“El arte y la ciencia de estructurar y organizar el entorno informativo para ayudar a los usuarios eficientemente a satisfacer sus necesidades informativas.” (Steve Toub)

Entre los aspectos fundamentales que atiende esta profesión se encuentra:

- Determinar la funcionalidad que la aplicación web va a tener.
- Especificar como los usuarios van a encontrar la información.
- Representar como la aplicación web se va a acomodar al cambio y crecimiento en el tiempo.

Beneficios que brinda la Arquitectura de la Información

- Generar estructuras que soporten el cambio y el crecimiento en el tiempo.
- Asegurar la consistencia y localización de la información.
- Crear sistemas de navegación intuitivos. (16)

La arquitectura de la información es un punto importante en cada una de las fases de elaboración de aplicaciones web, ya que la misma se encarga de clarificar la misión y visión del producto, la navegación y organización de la información en el sistema, influyendo en el impacto que pueda tener la aplicación en los usuarios.

1.4 Metodología de desarrollo de software

Las metodologías de desarrollo de software están conformadas por un conjunto de procedimientos y pasos que ayudan al correcto desarrollo de un software de gran prestigio. Actualmente existen varias metodologías, cada una de ellas con características particulares que las hacen diferenciarse, son clasificadas en dos grupos: metodologías ágiles y tradicionales.

Las metodologías tradicionales se basan en la idea de que el éxito del producto se puede lograr si se tiene todo correctamente documentado, mientras que, las ágiles defienden la idea de que el proceso de desarrollo del software, se centra en el software como tal y no en la documentación alrededor de este, sino que se toma en cuenta sólo la documentación necesaria y de forma muy sencilla. Se deben tener en cuenta varios aspectos en el momento de escoger la que se va a utilizar en el proceso de desarrollo de un proyecto (21). A continuación se hace un estudio de metodologías de desarrollo de software con la intención de seleccionar la que más se adapte a las características del proyecto.

Rational Unified Process (RUP)

El proceso unificado de *Rational* RUP, llamado así por sus siglas en inglés *Rational Unified Process*, divide en cuatro 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.
- Transición: el objetivo es llegar a obtener la liberación del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Modelado de Negocio: entender las necesidades del negocio.
- Requerimientos: trasladar las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: trasladar los requerimientos dentro de la arquitectura de software.

- Implementación: crear software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: asegurar que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- Configuración y administración del cambio: guardando todas las versiones del proyecto.
- Administrando el proyecto: administrando horarios y recursos.
- Ambiente: administrando el ambiente de desarrollo.
- Despliegue: realizando todo lo necesario para la salida del proyecto.

Los elementos principales del RUP son:

- Actividades: son los procesos que se llegan a determinar en cada iteración.
- Trabajadores: vienen hacer las personas o entes involucrados en cada proceso.
- Artefactos: un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de este proceso 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 (22).

Metodología ágil para el desarrollo de software(SXP)

SXP es un híbrido cubano de metodologías ágiles que tiene como base las metodologías SCRUM y XP, ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto. Se compone de cuatro fases:

- Planificación-Definición: donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

- Desarrollo: es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega: se despliega y se pone en marcha.
- Mantenimiento: donde se realiza el soporte para el cliente.

SXP es la más apropiada para proyectos de corta duración con requisitos cambiantes o no bien definidos, donde prevalezca la retroalimentación entre el cliente y el equipo de trabajo. El desarrollo con SXP se realiza en iteraciones cortas a lo largo de cuatro fases, dándole cumplimiento a un grupo de actividades, de las que se generan una serie de artefactos, que documentan el proceso de desarrollo, obteniendo una versión del producto con nuevas funcionalidades (23).

Metodología seleccionada

Luego de un estudio realizado se decide que la metodología SXP es la óptima a utilizar, ya que es utilizada para proyectos de corto plazo y con un equipo de trabajo pequeño. Dada las condiciones, facilidades que brinda y la propuesta de desarrollo que se tiene de la aplicación, se determina que sus características se asocian más al tipo de proyecto que se lleva a cabo durante la construcción de una aplicación web.

A continuación se presentan las razones fundamentales que se tomaron en cuenta para escoger la metodología:

- El proyecto es pequeño: SXP está concebida para ser utilizada dentro de proyectos pequeños y de desarrollo rápido.
- Pocos roles: esta metodología está dirigida a grupos de desarrollo pequeños y con pocos roles.
- Cuenta con una fase de exploración que le permite al equipo de trabajo elaborar las historias de usuarios y familiarizarse con las herramientas y tecnologías que se van a utilizar en el desarrollo del sistema.
- Existe una fuerte interacción entre el cliente y el equipo de desarrollo durante toda la realización de la aplicación, lo que implica que disminuya la cantidad de documentación y se definan en conjunto las funcionalidades.

- 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 SXP esto es completamente factible pues se adapta a los proyectos cuyos requerimientos cambian a menudo.

1.5 Tecnologías, herramientas y lenguajes a utilizar.

1.5.1 Sistemas de Gestión de Contenido (CMS)

Un Sistema de Gestión de Contenido (CMS) es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente trabajan contra una base de datos, que es actualizada por un editor, incluyendo nueva información o editando la existente (6).

Se clasifican interiormente en varias categorías, de acuerdo con la información que manejan:

Blogs: contienen noticias o artículos organizados cronológicamente, que no pueden ser editados, pero sí hacerse comentarios acerca de ellos.

Foros: contiene foros de discusión en línea, en los que las personas opinan sobre temas de su interés.

Portales: son sitios que combinan los blogs, foros, artículos, noticias y buscadores, los cuales tienen un gran tráfico de personas que persiguen fines comunes (pueden ser centros de atención a clientes, revistas o empresas que ofrecen servicios).

Wiki: cualquier persona puede colaborar o escribir sobre algún tema en el que pueda aportar sus conocimientos. También pueden contener espacios de discusión (foros) (6).

Ventajas de los Sistemas de Gestión de Contenidos (CMS).

- Tienen la capacidad de manejar eficientemente gran cantidad de páginas web.
- Trabajan en un ambiente de páginas web interactivas, es decir, que se generan según las peticiones de los usuarios.
- Controlan el acceso de los usuarios al sistema, mediante su contraseña y permisos asignados a cada uno de ellos.
- Mantienen el orden en el sistema, ya que se le puede asignar un mismo estilo a todas las

páginas generadas.

- Permiten tener una apariencia y navegación uniforme en todo el sitio.
- Aumentan la seguridad.
- Reducen los gastos de mantenimiento del sitio. (7)

Representación de los Sistemas de Gestión de Contenido más actuales.

Wordpress

Inicialmente enfocado a la creación de blogs (sitios web periódicamente actualizados), hoy día es usado para multitud de sitios web. Desarrollado en PHP y MySQL, bajo licencia GPL (Licencia Pública General) y código modificable. Posee como principales características, su licencia, el extenso repositorio de plugins y su facilidad de uso. Su estructura está basada en la arquitectura tres capas, compuesto por las capas de la base de datos, la base de programación y la base de diseño.

Otro motivo a considerar sobre su éxito y extensión, es la enorme comunidad de desarrolladores y diseñadores. WordPress pone especial atención a la estética, estándares web y usabilidad, sin embargo posee vulnerabilidades relacionadas con la inyección SQL.

Funcionalidades

- Garantiza la posibilidad de actualizar automáticamente el sistema.
- Permite ordenar artículos y páginas estáticas en categorías, subcategorías y etiquetas.
- Permite comentarios y herramientas de comunicación entre blogs.
- Permite subir y gestionar adjuntos y archivos multimedia (8).

Drupal

Es un software de código abierto, mantenido y desarrollado por una comunidad de más de 630 000 usuarios y desarrolladores, permite organizar y publicar su contenido con una infinita variedad de personalización. Es distribuido bajo los términos de la Licencia Pública General (o por sus siglas en inglés, GNU/GPL). Los principios del proyecto Drupal son fomentar la modularidad, la colaboración y la facilidad de uso. El secreto de Drupal para conseguir su reconocida flexibilidad y facilidad en la creación de sitios web es la abstracción y organización en capas que aplica en el tratamiento de los contenidos (9).

Ventajas

- Cuenta con una gran comunidad de usuarios y desarrolladores.
- Ofrece gran seguridad.
- Tiene un potente motor de plantilla.
- Permite la tenencia de varios sitios con una sola instalación.
- Es flexible y sin limitaciones.
- Un robusto sistema de ayuda *on-line* y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- Soporta varios lenguajes de base datos (PostgreSQL, MySQL).
- Trabaja sobre servidores Apache e Internet Information Server.
- Permite la activación de Módulos para los sistemas de recomendación.

Desventajas

- Es complicado aprender a usarlo.
- Necesita invertir tiempo para desarrollar todas sus posibilidades.

Joomla

Permite construir sitios web de manera rápida y eficiente, su facilidad de uso y extensibilidad han hecho de Joomla uno de los CMS más populares en el mundo, gracias, en parte, a ser de código abierto y gratuito. Joomla es usado para la construcción de sitios web corporativos, intranet corporativa, revistas, periódicos, tiendas virtuales, sitios de reservaciones en línea, páginas personales entre otros. Su arquitectura está separada en tres capas. Joomla sigue el modelo de desarrollo más avanzado y actual de Software (Modelo Vista Controlador). Además cuenta con una comunidad de desarrolladores muy talentosa y con más de 200 000 usuarios y colaboradores. (10)

Ventajas

- Fácil de instalar y configurar con sencillos clic.
- Es fácil de aprender.
- Administración intuitiva y amigable para los usuarios.

Desventajas

- Permite crear solo un sitio por instalación.
- No se le pueden integrar otros script a tu sitio.
- La utilización de recursos del servidor es mayor en comparación con la de Drupal.
- Los módulos adicionales no son gratis hay que comprarlos.
- Solo conoce un lenguaje de base datos (MySQL).
- Solo trabaja sobre servidores Apache.

Fundamento de la selección del CMS

Estos CMS comparten muchas similitudes, incluyen la mayoría de las funcionalidades generales de los CMS, son de código abierto, cuentan con un gran número de usuarios y colaboradores que contribuyen al mejoramiento y a la corrección de fallas. Después de haber analizado las ventajas y desventajas de estos CMS, se decide escoger para la propuesta de solución al problema planteado el CMS Drupal; pues el mismo presenta flexibilidad y extensibilidad para el desarrollo de aplicaciones web. Otro elemento a tener en cuenta es que soporta una amplia variedad de estructuras web y ampliamente adaptable a cualquier necesidad. En la universidad es el CMS más utilizado por lo que existe una gran comunidad y documentación, lo cual es una gran ventaja en las fases de desarrollo y soporte de la aplicación. Se propone su utilización en su versión 7.26.

1.5.2 Lenguajes de desarrollo

HTML

Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, videos y enlaces. Utilizando HTML se puede definir la apariencia de las páginas creadas de diferentes formas. Contiene una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje para presentación con soporte de aspectos visuales (12).

PHP

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Se conecta a servidores de bases de datos tales como MySQL, Oracle, Informix,

PostgreSQL, entre otros. Debido a su amplia distribución, PHP está perfectamente soportado por una gran comunidad de desarrolladores. Es un lenguaje de programación de alto nivel, totalmente compatible con los modernos métodos orientados a objetos, prácticas y principios (5).

CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML que separa los contenidos y su presentación por lo que es imprescindible para crear páginas web complejas. En la actualidad se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc (13).

JavaScript

Es un lenguaje de programación que fue creado con el objetivo de hacer páginas dinámicas. Es interpretado, no necesita ser compilado, solo es necesario un navegador para su interpretación por lo que se ejecuta del lado del cliente. Es multiplataforma, orientado a objetos y guiado por eventos, que produce el mismo usuario. Permite el control de las ventanas del navegador, contenidos y el control de los datos que se inserten en cualquier formulario (27).

1.5.3 Servidores web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada).

Apache

Es un servidor de red para el protocolo HTTP, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Apache se distribuye como software libre de código abierto, modular multiplataforma, extensible, popular (fácil de conseguir ayuda/soporte) y gratuito. Su licencia es una descendiente del tipo BSD que permite el uso comercial y no comercial de Apache (14).

Características:

- Altamente configurable de diseño modular y en la creación y gestión de *logs*(registros).

- Personaliza la respuesta ante los posibles errores que puedan ocurrir.

Internet Information Server (IIS).

Es un potente servidor web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones web sobre todas las versiones de Windows Server 2003. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones web y a la vez reducir sus costes administrativos. IIS 0.6 soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI) con monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas (15).

Selección del servidor

Teniendo en cuenta las características de ambos servidores, es seleccionado el servidor web Apache; pues el mismo es un proyecto libre y gratuito, además es uno de los más utilizados a nivel mundial, mientras que el servidor Internet Information Server a pesar de ser muy potente solo funciona bajo servidores Microsoft y por una licencia. Se propone la utilización del servidor Apache en su versión 2.2.

1.5.4 Sistema de Gestión de Base de datos (SGBD)

MySQL

Dentro de los gestores de bases de datos relacionales, MySQL es uno de los utilizados para el desarrollo de aplicaciones web, el mismo posee dos licencias, una GPL para cualquier uso compatible y otra para aquellas empresas que quieran incorporarlo en productos privativos. Entre las características principales de MySQL se encuentran:

Escalabilidad: Tiene la capacidad de manejar grandes volúmenes de datos, llegando a la cifra de las seis mil tablas, alrededor de cincuenta millones de registros y hasta 32 índices por tabla. MySQL está desarrollado en C y C++, probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes que permiten a las aplicaciones escritas en diversos lenguajes de programación acceder a dicho gestor de base de datos.

Conectividad: Permite múltiples conexiones independientemente de la cantidad de computadoras, usuarios y de sistemas operativos que estén haciendo uso del protocolo TCP/IP.

Plataforma: Posee versiones para varios sistemas operativos, por lo que es común ver servidores GNU/Linux o Unix, usando MySQL, que a su vez estos datos pueden ser utilizados en

ordenadores con sistemas operativos Windows, Linux, Solaris, etc. Posee una buena integración con PHP, lenguaje usado para el desarrollo web.

Seguridad: Ofrece un control de acceso de usuarios mediante un sistema de contraseñas seguras y privilegios mediante verificación basada en el host, así como el tráfico de contraseñas que está cifrado al conectarse a un servidor.

Manejo de datos: Permite manejar multitudes de tipos de datos para columnas, además de utilizar registros de longitud fija o variable (18).

PostgreSQL

Se trata de un sistema de administración de base de datos que incorpora el modelo relacional para sus bases de datos y es compatible con el lenguaje de consulta SQL estándar. Resulta ser muy capaz y muy confiable, y tiene buenas características de rendimiento. Es un sistema multiplataforma por lo que opera en varios sistemas operativos como Unix, Mac OS, Windows, Linux, es de código abierto lo que hace posible que los usuarios puedan realizar las modificaciones pertinentes al código fuente según la necesidad de estos. Este sistema soporta potentes lenguajes de programación como Java, C, C++, Perl, Python, PHP y Ruby.

Entre sus principales características resaltan la replicación asincrónica y sincrónica, transacciones anidadas, copias de seguridad, múltiples métodos de autenticación, MVCC (Control de Concurrencia Multi-Versión) el cual permite que se ejecuten sobre una tabla varias transacciones a la vez, pues para cada transacción se muestra una versión de la tabla y no la original. Este gestor es robusto y por ende hoy en día es muy usado con respecto a gestores libres existentes como *SQLite*, *MySQL*, *FireBird*, entre otros (11).

Selección del Sistema de gestor de base de datos

Teniendo en cuenta las características antes expuestas de los sistemas gestores de base de datos, es seleccionado PostgreSQL, pues a pesar de que ambos son proyectos de base de datos que brindan múltiples ventajas. PostgreSQL, está distribuido bajo términos más permisivos, resulta ser más potente que la versión libre de MySQL y garantiza la integridad de los datos, provee a la base de datos de una mayor consistencia, además presenta una mayor escalabilidad y rendimiento bajo grandes cargas de trabajo, otro factor importante es la experiencia en el trabajo con PostgreSQL por parte del personal encargado de desarrollar la aplicación web. Se

propone la utilización de este en su versión 9.1.

1.5.5 Herramientas a utilizar

PgAdminIII

PgAdminIII es una aplicación gráfica para el uso del gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Es capaz de gestionar bases de datos en las versiones más recientes de PostgreSQL, así como otras versiones comerciales. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP, y puede encriptarse mediante SSL (acrónimo de *Secure Sockets Layer*-Protocolo de Capa de Conexión Segura) para mayor seguridad. Tiene características interesantes como son: construcción gráfica de consultas, inclusión del framework *PgScript* para el desarrollo de scripts y así ejecutar las consultas, buscador de objetos, opciones para habilitar o deshabilitar reglas y borrar o reasignar rol a determinadas bases de datos (30). Se propone la utilización de este en su versión 1.14.0.

Geany

Geany es un editor de texto pequeño y ligero basado en *Scintilla* con características básicas de entorno de desarrollo integrado (IDE). Utiliza librerías GTK para su funcionamiento. Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y *Microsoft Windows*. Es distribuido como software libre bajo la Licencia Pública General de GNU.

Algunas de sus funcionalidades son:

- Resaltado de sintaxis.
- Plegado de código.
- Autocompletado.
- Cierre automático de etiquetas XML y HTML.
- Muestra de consejos.
- Muchos tipos de archivos soportados tales como C, *Java*, PHP, *Python*, *Perl* y Pascal.
- Listas de símbolos.

- Código de navegación.
- Construir un sistema para compilar y ejecutar el código.
- Fácil gestión de proyectos.
- Soporte para plugins.(31)

Teniendo en cuenta los elementos anteriormente mencionados y que la implementación de la aplicación no requiere de un potente Entorno de Desarrollo Integrado (IDE), se propone la utilización de este en su versión 0.21.

Visual Paradigm

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computador o en inglés *Computer Aided Software Engineering*) profesional para UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Garantiza una rápida construcción de aplicaciones con una mayor calidad, y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. Es una herramienta multiplataforma distribuida bajo licencia privativa en la *Visual Paradigm Suite*. Es orientada a objetos y se puede integrar con otras herramientas de desarrollo. Proporciona un entorno ágil y eficiente para el diseño. Permite el diseño de base de datos y el diseño del sistema con el diagrama de clases UML (24).

Se decide utilizar Visual Paradigm como herramienta de diseño UML, pues permitirá modelar los diagramas que se generen durante todo el desarrollo de la solución propuesta. Esta herramienta, aunque no es libre constituye la mejor opción para realizar las tareas antes descritas. Se propone la utilización de este en su versión 8.0 para GNU/Linux.

Bootstrap

Bootstrap es el *framework* de *Twitter* que permite la creación de diseños web combinando CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice, es decir, automáticamente se adapta al tamaño de un ordenador, de una *Tablet* o de un teléfono móvil sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design. Es un *framework* potente con numerosos componentes webs como: botones, menús desplegables, formularios entre otros que nos ahorrarán mucho esfuerzo y tiempo. (26)

Características principales de Bootstrap:

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo jQuery.
- Funciona con todos los navegadores.

Firebug

Firebug es una extensión del navegador web Firefox que brinda un paquete de utilidades para el desarrollo de páginas y aplicaciones web. Permite examinar todos los elementos HTML, y además los estilos CSS y los códigos de scripting en *JavaScript* que está ejecutando la página. Se distribuye bajo la *Mozilla Public License 1.1* lo cual significa que es de código abierto y gratuito (25). Se propone la utilización de este en su versión 1.10.3. Entre las características que posee se pueden destacar las siguientes:

- Se puede inspeccionar cualquier etiqueta HTML y editarla desde el propio Firebug, para ver los resultados en la página al instante.
- Se puede ver qué estilos afectan a cada área o elemento de la página y editarlos en el instante por medio de la consola Firebug, viendo los resultados en la propia página.
- Tiene herramientas para ver cómo se maquetan los elementos de la página con las propiedades CSS.

1.6 Proceso de soporte

El proceso de soporte, representa una actividad interna generalmente horizontal que asegura el buen funcionamiento de la empresa. Los procesos de soporte generalmente son invisibles para el cliente (beneficiario). Estos procesos incluyen administración financiera, administración de recursos humanos, capacitación, etc. (19).

Por su lado, el soporte técnico es aquel tipo de servicio que proporciona una asistencia directa sobre el hardware o software de una computadora o de cualquier otro dispositivo electrónico, cuya principal misión resulta ser la de ayudar al usuario a resolver cualquier tipo de problemática que surja en el uso del mismo; en vez de entrenar o personalizar. En la actualidad, existen

múltiples lugares libres en la web respecto a soporte técnico, en los cuales los usuarios más experimentados ayudan a los novatos (20).

Cuando el soporte está debidamente organizado, se pueden dar varios niveles de soporte, donde el soporte de primer nivel consiste en dar al cliente orientación telefónica a nivel de usuario y operacional. El soporte de segundo nivel es un servicio que consiste en proveer soporte técnico en los problemas de uso de aplicaciones, su objetivo es mantener recursos especializados disponibles, con cobertura horaria, con el fin de mejorar las aplicaciones del cliente. El soporte de tercer nivel está relacionado con la intervención e investigación por parte de los especialistas a nivel de arquitectura del software (20).

El costo del soporte puede variar. Algunas empresas ofrecen soporte gratuito limitado cuando se compra su hardware o software; otros cobran por el servicio de soporte telefónico. Algunos son gratuitos mediante foros, salas de charla, correo electrónico y algunos ofrecen contratos de soporte (20).

En general las aplicaciones de soporte a las metodologías de desarrollo de software tienen como objetivo principal, apoyar a los equipos de desarrollo en las fases de planificación, diseño, implementación y pruebas de sus soluciones; de manera que se ejecuten de una forma más eficiente y óptima. En este tipo de aplicaciones los usuarios tienen acceso a personal con amplia experiencia en el manejo de la metodología con la que su aplicación está desarrollada, así como tecnologías complementarias que ayudan a construir una visión mucho más global de su sistema.

1.7 Aplicaciones web para brindar soporte a las metodologías de desarrollo de software

Teniendo en cuenta el estudio de varias aplicaciones web que dedican su funcionamiento a la publicación de toda la información relacionada con las metodologías de desarrollo de software, se logró observar como se ha ampliado la utilización de estas para la capacitación de los equipos de desarrollo y reconocer las principales funcionalidades que debe tener la solución propuesta. Entre las aplicaciones visitadas se encuentra la metodología SCRUM, metodología OpenUp, metodología XP, metodología RUP, la aplicación VersionOne y la metodología Merinde.

Aplicación Web de la metodología SCRUM (<https://www.scrum.org>)

Esta aplicación cuenta con una página principal, que incluye banners publicitarios, posee seis secciones que se definen en su menú principal (Comunidad, Guía de la metodología, recursos, cursos que se encuentran en línea, valoraciones de la metodología. Presenta un formulario de registro, enlaces de contacto, términos de uso e información sobre la empresa, tiene la opción de cambiar la guía de Scrum a 30 idiomas, una sección en donde se muestran algunas noticias asociadas a la metodología, tales como, conferencias que se imparten y productos que se han desarrollado sobre la base de la misma, y además posibilita la interacción entre los usuarios a partir de un servicio de foro.

Aplicación Web de la metodología XP (<https://sites.google.com/site/xpmetodologia>)

Esta aplicación presenta un diseño más sencillo, cuenta con un banner que no ofrece mucha información y un menú principal con varias secciones en donde se exponen cada uno de los aspectos de la metodología, tiene una sección de búsqueda y de encuesta, sin embargo no cuenta con formularios de registros, no posee otros servicios como foros de discusión o sala de chat que permita la interacción entre los usuarios.

Aplicación Web de la metodología OpenUp (epf.eclipse.org/wikis/openup)

La aplicación OpenUp presenta un banner principal con un diseño simple, posee un panel lateral donde se sitúa por temas toda la información referente a la metodología, muestra a los usuarios ejemplos de los artefactos que se generan en cada fase, en el banner se sitúan otras secciones que dan acceso a un glosario de términos, información sobre la empresa, la posibilidad de imprimir las páginas, cuenta con un foro de discusión para lograr la interacción entre los clientes y los expertos en la metodología. Presenta un formulario de registro, sin embargo no necesariamente debe registrarse si el propósito del usuario es solamente de lectura.

Aplicación de la metodología RUP (Rational Unified Process)

(www-01.ibm.com/software/rational/rup)

La aplicación de la metodología RUP posee un banner principal que cuenta con varias secciones relacionadas con los servicios que se brindan en la aplicación, una sección de productos y servicios, de soporte y descargas, soluciones e industrias por temas en donde los clientes de acorde al tema que seleccionen se les brinda ayuda sobre el ciclo de vida del software que se esté desarrollando o se vaya a desarrollar y cuenta con una sección de registro. Contiene un

panel lateral derecho en donde aparecen datos de contacto; que los clientes pueden emplear para comunicarse con los grupos de expertos en el tema y presenta un área en donde se encuentran documentos relevantes que los usuarios deseen descargar. Esta aplicación cuenta con múltiples clientes; desde empresas respetadas como *Oracle* y *Microsoft* hasta empresas que se inician en el desarrollo de software.

Aplicación Web VersionOne para el manejo de proyectos basados en metodologías ágiles (www.versionone.com)

La aplicación VersionOne cuenta con un banner principal en donde aparece un menú que presenta varias secciones relacionadas con las ventas de las ediciones que prefieran los clientes, recursos, soporte e información sobre las metodologías ágiles. Posee otras secciones que involucran a los clientes, ofertas de cursos, productos que desarrollan, información sobre la empresa y una sección de búsqueda. Presenta un espacio en donde se detalla cada una de las fases del desarrollo de software, ofrece herramientas para generar los artefactos que requiere el producto. VersionOne desde sus inicios en el año 2002 hasta la fecha; cuenta con múltiples empresas asociadas, debido a la capacidad de simplificar el proceso de desarrollo de los productos por medio del enfoque ágil.

Aplicación Web de la metodología Merinde (Metodología de la Red Nacional de Integración y Desarrollo de Software Libre) (merinde.net)

La aplicación Merinde cuenta con un banner principal en donde se encuentran localizadas varias secciones relacionadas con la disciplina de la metodología, los artefactos que se generan, las actividades, tareas, descripción de roles, glosario de términos, videos y una sección de ayuda, en la parte superior del banner aparecen otras secciones tales como, el mapa del sitio, zona de descargas y de contacto con los responsables de la aplicación. Presenta además un panel a la izquierda en donde aparece un buscador y otro conjunto de secciones vinculadas con las buenas prácticas, marco de desarrollo, mantenimiento, aportes, ventajas y desventajas, el número de visitas por usuario a la aplicación. Posee un diseño más llamativo en comparación a las aplicaciones estudiadas.

Conclusiones sobre el estudio de aplicaciones web utilizadas para brindar soporte a las metodologías de desarrollo de software.

Del estudio realizado se concluyó que la aplicación de soporte a la metodología Nova-OpenUp debe contar con los siguientes requisitos, siendo estos comunes en aplicaciones de soporte a las metodologías de desarrollo de software:

- Utilización de *Banners* para dar publicidad.
- Enlaces para términos de uso, información sobre el proyecto encargado de la metodología, contacto, formulario de búsqueda y mapa del sitio.
- Permitir múltiples conexiones de usuarios y que sea accesible para todos los que navegan en él.
- Debe poseer un servicio de foro y notificaciones de correo para permitir la interacción entre los usuarios y los profesionales encargados de la aplicación.
- Un centro de descarga donde aparezca documentación sobre la metodología Nova-OpenUp.

1.8 Conclusiones del capítulo

En este capítulo se realizó el estudio de algunas aplicaciones web destinadas a brindar soporte a las metodologías de desarrollo de software, las cuales fueron utilizadas como referencia en cuanto a funcionalidades y diseño de la aplicación que se desarrolla. La aplicación de soporte a la metodología de desarrollo de distribuciones de GNU/Linux, Nova-OpenUp, se centrará en el soporte de tercer nivel para proporcionar la capacitación de los usuarios que la utilizan y proveer los recursos necesarios en el desarrollo de las aplicaciones de los clientes.

Para incrementar el uso de la de la metodología Nova-OpenUp es preciso desarrollar una aplicación web dirigida por la metodología SXP, ya que se ajusta a las características del sistema. Se definió el Sistema de Gestión de contenido Drupal, para agilizar el proceso de desarrollo de la aplicación. Se seleccionaron los lenguajes de programación HTML, PHP y JavaScript para la implementación de las funcionalidades del sistema; los cuales están fundamentados por las tecnologías que soporta el CMS Drupal. La selección del Sistema Gestor de Base de datos PostgreSQL, dotará al sistema de una base de datos estable, capaz de soportar una gran cantidad de peticiones simultáneas.

Capítulo 2: Requisitos y Arquitectura de la aplicación web para la Metodología Nova-OpenUp.

2.1 Introducción

El siguiente capítulo está enmarcado en las dos primeras fases definidas en la metodología SXP. Se abordan temas relacionados con el funcionamiento del sistema. Se muestra la lista de reserva del producto, las historias de usuarios y se explica la Arquitectura de la aplicación a desarrollar.

2.2 Descripción de la aplicación

2.2.1 Modelo de dominio

En el Modelo de Dominio se representan los conceptos más importantes y significativos en el desarrollo de un sistema. Su objetivo fundamental es definir las interrelaciones de los objetos más importantes representados mediante clases. Además, desempeña un papel central en la comprensión del entorno actual y en la planificación futura de la posible aplicación.

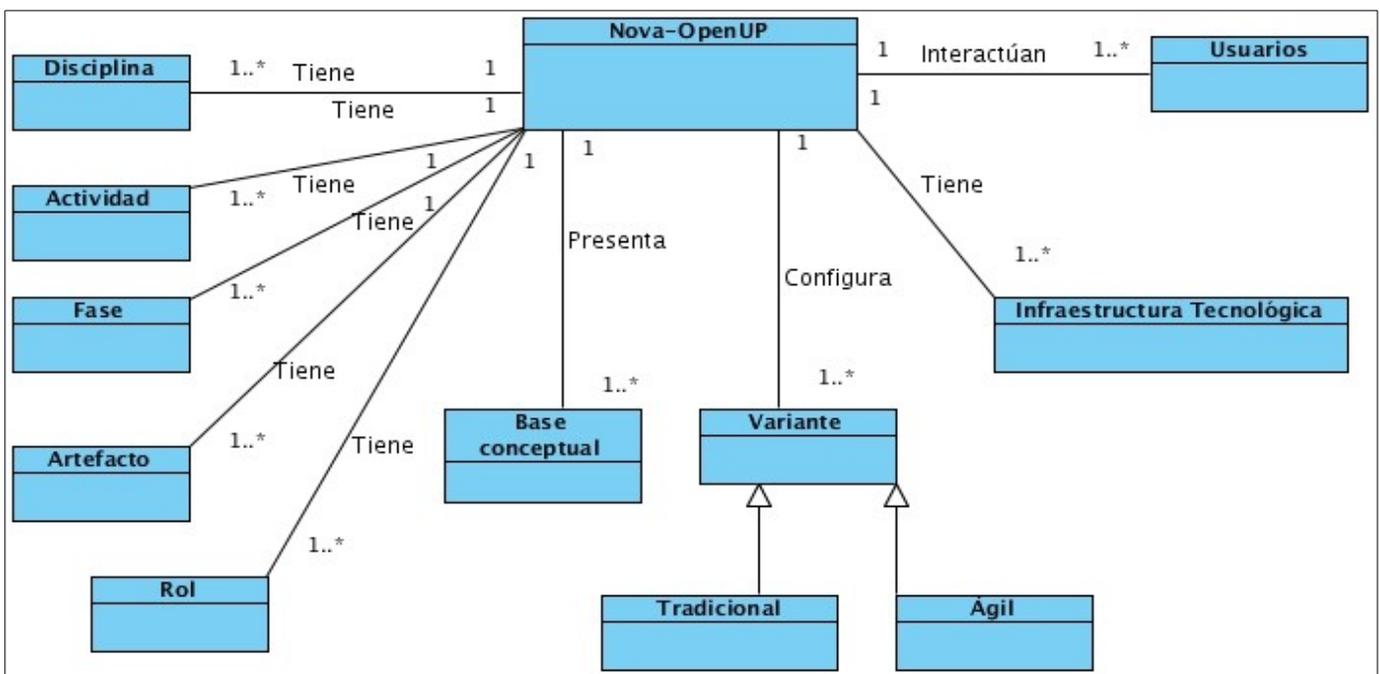


Figura 1. Modelo de dominio

A continuación se describen los diferentes objetos y relaciones que conforman el proceso.

Nova-OpenUp: Documento que contiene cada uno de los elementos que conforman la metodología Nova-OpenUp, tales como: actividades, artefactos, infraestructura tecnológica, base

conceptual, disciplinas, fases y roles.

Base conceptual: Contiene los elementos teóricos por los cuales se utiliza la metodología para el desarrollo de la distribución de GNU/Linux Nova.

Infraestructura Tecnológica: Representa las herramientas que los usuarios pueden emplear para agilizar el proceso de desarrollo de software.

Variante: Se refiere a la configuración que puede adoptar la metodología, atendiendo a las características de los proyectos, ya sea metodología ágil o tradicional.

Usuarios: Personas que utilizan la metodología para guiar el desarrollo de sus aplicaciones.

Artefactos: Conjunto de planillas con las que cuenta la metodología para documentar la información del software.

Fases: Contiene las fases que conforman la metodología Nova-OpenUp.

Disciplinas: Presenta las disciplinas que define la metodología.

Actividades: Contiene las actividades que se deben desarrollar durante el ciclo de vida del software.

Roles: Contiene los roles que propone la metodología para asignar las responsabilidades a los integrantes del proyecto.

2.3 Requisitos funcionales y no funcionales del sistema.

Con la especificación de los requisitos de software se obtiene una descripción detallada de las necesidades del producto informático a desarrollar. Estos se dividen en dos grupos para una mejor especificación, los requisitos funcionales y los no funcionales. A continuación se muestra la lista de los 47 requisitos funcionales con que cuenta la aplicación, de los cuales 21 son de prioridad alta, 17 de prioridad media y 9 de prioridad baja.

2.3.1 Lista de requisitos funcionales

RF1 Autenticar usuario (alta)	RF31 Insertar variante (media)
RF2 Crear cuenta de usuario (alta)	RF32 Editar variante (media)
RF3 Editar cuenta de usuario (alta)	RF33 Eliminar variante (media)
RF4 Eliminar cuenta de usuario (alta)	RF34 Mostrar variante (media)
RF5 Insertar fase(alta)	RF35 Insertar preguntas y respuestas relacionadas con

Aplicación de soporte a la metodología Nova-OpenUp

RF6 Editar fase (alta)	el funcionamiento de la aplicación y la metodología.
RF7 Eliminar fase (alta)	(media)
RF8 Mostrar fase (alta)	RF36 Editar preguntas y respuestas relacionadas con
RF9 Insertar artefacto (alta)	el funcionamiento de la aplicación y la metodología.
RF10 Editar artefacto (alta)	(media)
RF11 Eliminar artefacto (alta)	RF37 Eliminar preguntas y respuestas relacionadas
RF12 Mostrar artefacto (alta)	con el funcionamiento de la aplicación y la metodología.
RF13 Descargar artefacto (alta)	(media)
RF14 Insertar disciplina (alta)	RF38 Mostrar preguntas y respuestas relacionadas con
RF15 Editar disciplina. (alta)	el funcionamiento de la aplicación y la metodología.
RF16 Eliminar disciplina (alta)	(media)
RF17 Mostrar disciplina (alta)	RF39 Enviar mensajes vía correo a los usuarios. (baja)
RF18 Insertar rol (alta)	RF40 Insertar vínculos (baja)
RF19 Editar rol (alta)	RF41 Editar vínculos (baja)
RF20 Eliminar rol (alta)	RF42 Eliminar vínculos (baja)
RF21 Mostrar rol (alta)	RF43 Mostrar vínculos (baja)
RF22 Insertar base conceptual (media)	RF44 Mostrar lista de usuarios
RF23 Editar base conceptual (media)	conectados (baja)
RF24 Eliminar base conceptual (media)	RF45 Publicar artículos (baja)
RF25 Mostrar base conceptual (media)	RF46 Eliminar artículos (baja)
RF26 Insertar actividad	RF47 Descargar artículos (baja)
(media)	
RF27 Editar actividad	
(media)	
RF28 Eliminar actividad	
(media)	
RF29 Mostrar actividad	

(media)	
RF30 Calcular variante (media)	

Tabla 1: Requisitos funcionales

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que debe tener el producto. Entre los requisitos no funcionales del sistema propuesto se encuentran:

Usabilidad

RnF 1. < Requisito de usabilidad >

Tipo de aplicación informática: Aplicación web

RnF 2 < Requisito de usabilidad > Finalidad

La aplicación de soporte para la metodología Nova-OpenUp, persigue informar sobre cada uno de los elementos que componen a la metodología, en aquellos proyectos que la emplean como guía en el proceso de desarrollo de software.

Debe existir una buena distribución y categorización de los contenidos para facilitar la navegación al usuario.

Para hacer uso del sistema no es necesario poseer conocimientos avanzados de la metodología o sobre el ambiente web en sentido general.

RnF 3. < Requisito de usabilidad > Ambiente

Características de Software

Sistema operativo: Nova 2011 o superior, Ubuntu 11.04 o superior, Windows XP, o superior.

Navegador web: Internet Explorer 7.0 o superior, Mozilla Firefox 2.0 o superior.

Servidor web: Apache 2.0 o superior.

Versión de PHP: 5.0 o superior.

Sistema Gestor de Base de Datos: PostgreSQL 8.4 o superior.

Características de Hardware (PC cliente)

512 MB de memoria RAM

80 GB de disco duro

Microprocesador: Intel Pentium IV o superior

Características de Hardware (Servidores)

Servidor web Apache: El servidor contará con 2GB de memoria RAM, un microprocesador Intel core 2 Duo y un disco duro de 250GB.

Servidor Base de datos: El servidor contará con 2GB de memoria RAM, un microprocesador Intel core 2 Duo y un disco duro de 160GB.

Tiempo de respuesta:

El sistema debe demorar como promedio en una transición, de dos a cinco segundos aproximadamente.

Soporte

RnF 4. < Restricción de soporte>

El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarios.

Restricciones de diseño

RnF 5. < Restricción de diseño>

Lenguajes de programación

Lenguaje de programación PHP 5 o superior.

Lenguaje de marcas de hipertexto (HTML 5).

Hojas de estilos en cascada (CSS 3).

RnF 6. < Restricción de diseño>

Herramientas de desarrollo

Framework Bootstrap

Sistema de Gestión de Contenidos (CMS) Drupal 7.26

Visual paradigm 8.0.

RnF 7. < Restricción de diseño>

Debe brindar una interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario, facilitando en todo momento la interacción de este con el sistema.

Componentes adquiridos

RnF 8. < Licencia de los Componentes adquiridos>

Uso de la licencia GNU/GPL para el CMS Drupal

Uso de la licencia BSD de PostgreSQL.

Uso de la licencia PHP *License*.

Todas estas licencias son consideradas libres y flexibles, además de haber sido aprobadas por la Fundación de Software Libre (*Free Software Foundation*).

2.4 Propuesta de la aplicación de soporte a la metodología Nova-OpenUp.

La estructura de la aplicación se describe a continuación, así como las diferentes secciones con las que cuenta, para una mejor comprensión por parte del usuario. La navegación no excede los tres niveles de profundidad.

Inicio: En esta sección se expondrá de manera general todos los elementos relacionados con la Metodología Nova-OpenUp.

- Fases
- Roles
- Disciplinas
- Actividades
- Artefactos

Variante: En esta opción se visualizan los elementos de la variante ágil que puede adoptar la Metodología Nova-OpenUp según el producto a construir.

Nova-OpenUp: Se muestra toda la información básica de la metodología al estilo OpenUp.

Autenticar: Cada usuario tiene permisos específicos para acceder a las secciones de la aplicación.

Administración: Sección de administración que posibilita la actualización de todas las informaciones contenidas en la aplicación. Teniendo en cuenta dicha descripción, el sistema dispone de los siguientes roles de usuarios:

- Usuario invitado: Es la persona que navega en la aplicación sin haberse registrado aún y tiene la posibilidad de visualizar las diferentes opciones que se brindan.
- Usuario registrado: Es la persona que tiene acceso a cualquier opción de la aplicación, una vez que se haya autenticado, excepto a la sección de administración.
- Usuario administrador: Es el usuario que tiene control total sobre la aplicación.

- **Usuario editor:** Es el usuario con permisos para crear, revisar y publicar la información en la aplicación.

Publicación de artículos: Se publican artículos relacionados con la metodología y el proceso de desarrollo de software.

Infraestructura Tecnológica: En esta sección se publican las herramientas que se emplean en el desarrollo de la distribución de GNU/Linux Nova.

Ayuda: La sección está compuesta por un foro de discusión en donde los usuarios podrán aclarar dudas relacionadas con el trabajo en la aplicación y la metodología.

Usuarios conectados: Se muestra los usuarios que se encuentran en línea en todo momento, permitiendo el control de acceso a la aplicación.

2.5 Historias de usuarios

Uno de los artefactos más importantes que genera la metodología SXP son las historias de usuario. Durante el análisis en la fase de planificación se identificaron 14 historias de usuarios, cada una de ellas respondiendo a las diferentes funcionalidades solicitadas por el cliente y dando una idea al resto del equipo de trabajo de cómo debe ser su posterior implementación. A continuación se ofrece la descripción de algunas historias de usuarios de prioridad alta para el cliente, el resto de las historias de usuarios se encuentran en el Anexo 2.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar Usuario
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel B. Sanchez Quiala	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 0.2
Riesgo en Desarrollo: Medio	Puntos Reales: 0.2
Descripción: El usuario debe poder autenticarse para así acceder a las principales opciones que le brinda la aplicación.	

Observaciones: Para acceder a la aplicación el usuario debe con anterioridad, crearse una cuenta en el sistema.

Para acceder a la aplicación el usuario debe completar los campos: nombre de usuario y contraseña. En dependencia de los privilegios del usuario, puede acceder a las funcionalidades del sistema.

Prototipo de interfaz: Anexo 4

Tabla 2: Historia de usuario. Autenticar usuario

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar fases
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 0.4
Riesgo en Desarrollo: Medio	Puntos Reales: 0.4
Descripción: Permite adicionar, editar, eliminar y mostrar la información de cada una de las fases de la metodología Nova-OpenUp.	
Observaciones:	
Para adicionar una fase se deben completar los campos siguientes: título y cuerpo. Para modificar la información de las fases, se debe seleccionar la fase y escoger la opción Editar. En el caso de eliminar la información de las fases, se debe seleccionar previamente la fase y escoger la opción Eliminar.	
Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores. Para mostrar las fases, se debe localizar y presionar clic sobre su nombre o la imagen que las identifica.	
Prototipo de interfaz: Anexo 4	

Tabla 3: Historia de usuario. Gestionar fases

Historia de usuario	
Número: 4	Nombre: Gestionar artefacto
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel B Sanchez Quiala	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados (semanas): 1.2
Riesgo en desarrollo: Alta	Puntos reales (semanas): 1.2
Descripción: Permite insertar, modificar, eliminar y mostrar cada uno de los artefactos que propone la metodología.	
Observación: Para adicionar un artefacto se deben completar los campos siguientes: nombre del	

artefacto, clasificación, descripción, disciplina a la que pertenece, planilla del expediente de proyecto del artefacto, herramientas que permiten la confección del artefacto y si el mismo permanece en la variante ágil.

Para modificar la información de los artefactos, se debe seleccionar previamente el artefacto y escoger la opción Editar.

Para eliminar la información de los artefactos, se debe seleccionar previamente el artefacto y escoger la opción Eliminar y Confirmar su eliminación.

Los usuarios deberán tener permisos de administración o edición para realizar las acciones anteriores.

Para mostrar los artefactos, se debe localizar y presionar clic sobre su nombre o la imagen que lo identifica.

Para descargar un artefacto, se localiza y se presiona clic sobre la planilla de la vista de artefactos.

Prototipo de Interfaz: Anexo 4

Tabla 4: Historia de usuario. Gestionar artefacto

Historia de usuario	
Número: 5	Nombre: Gestionar disciplinas
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel B Sanchez Quiala	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados (semanas): 0.4
Riesgo en desarrollo: Medio	Puntos reales (semanas): 0.4
Descripción: Permite insertar, editar, eliminar y mostrar la información de las disciplinas que propone la metodología.	
Observación: Para adicionar las disciplinas se deben completar los campos título y cuerpo; en este campo se adiciona la descripción de la disciplina, la relación entre los roles y los artefactos, y la relación entre las actividades, roles y artefactos.	
Para modificar la información de las disciplinas, se debe seleccionar la disciplina y escoger la opción Editar.	
Para eliminar la información de las disciplinas, se debe seleccionar previamente la disciplina y escoger la opción Eliminar.	
Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores.	
Para mostrar las disciplinas, se debe localizar y presionar clic sobre su nombre o la imagen que las identifica.	
Prototipo de interfaz: Anexo 4	

Tabla 5: Historia de usuario. Gestionar disciplinas

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar roles
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4

Riesgo en Desarrollo: Medio	Puntos Reales: 0.4
Descripción: Permite adicionar, modificar, eliminar y mostrar la información de cada uno de los roles que propone la metodología Nova-OpenUp	
Observaciones: Para adicionar los roles se debe llenar los campos siguientes: nombre del rol, habilidades y responsabilidades. Para modificar la información de los roles, se debe seleccionar el rol y escoger la opción Editar. Para eliminar la información de los roles, se debe seleccionar previamente el rol y escoger la opción Eliminar. Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores. Para mostrar los roles, se debe localizar y presionar clic sobre su nombre o la imagen que los identifica.	
Prototipo de interfaz: Anexo 4	

Tabla 6: Historia de usuario. Gestionar roles

2.6 Descripción de la Arquitectura de software y los patrones de diseño

2.6.1 Arquitectura de software

Al ser utilizado el CMS Drupal para el desarrollo de la aplicación web, la arquitectura y los patrones son heredados del mismo. La arquitectura de Drupal es un tema muy polémico en estos momentos, la afirmación de que Drupal presenta una arquitectura de n-capas, viene dada por la combinación de dos elementos fundamentales. Siendo el primero que esta arquitectura es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo (17) y el segundo, es su adaptabilidad mediante la adición de módulos, que son al final porciones de la aplicación que se modifican, sin tener que realizar cambios en el resto de la aplicación, además de dividir su contenido en una serie de elementos básicos: nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuarios (*user permissions*) y plantillas (*templates*).

Los nodos son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de nodos, los cuales van formando un “depósito de nodos” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este “depósito de nodos”. Los módulos son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. La siguiente capa de Drupal la constituyen los “*Blocks &*

Menus". Estos permiten estructurar y organizar los contenidos en la página web. Es decir que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos. La siguiente capa importante en Drupal es la de control de usuarios y permisos. Actualmente, la mayor parte de sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario. Es decir que las operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal. La última capa, es la capa de *"Templates"* y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia (29).

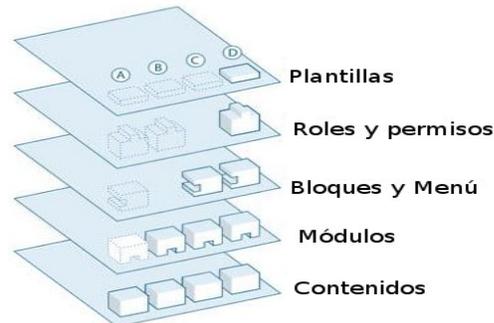


figura 2: Arquitectura del CMS Drupal

2.6.2 Patrones de diseño

De los patrones de diseño que presenta Drupal, la aplicación utiliza los siguientes:

Singleton

Este patrón está diseñado para que solo permita que se haga una instancia de un elemento cuando este no existe. Dentro del núcleo de Drupal se utiliza este patrón de diseño pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos elementos,

pues Drupal solo crea una instancia de un nodo, cuando este nodo no existe (28).

En la aplicación puede evidenciarse este patrón cuando el editor crea los tipos de contenido “Artefacto” y “Actividad”. Primeramente verifica que no existan en la base de datos y luego los crea.

Observer

Este patrón es usado en Drupal para controlar la forma en que se realizan las interacciones entre nodos, pues cuando se hace una modificación en un nodo, Drupal notifica de este cambio a todos los nodos que se vean afectados por él (28).

La utilización del patrón Observer se demuestra en la aplicación cuando el editor modifica el tipo de contenido “Actividad” que pertenece a una sección, entonces el sistema informa a todas sus dependencias de la modificación, como por ejemplo la vista que lista las actividades de la sección a la que pertenece.

Decorator

Este patrón se usa en Drupal, por ejemplo, de la siguiente manera: un nodo necesita tener la opción de adjuntar archivos y para esto, Drupal se vale del módulo *upload*, que brinda a los nodos la posibilidad de adjuntar, en vez de implementar esta funcionalidad en cada nodo (28).

En la aplicación, este patrón se utiliza cuando se adjunta un archivo a los contenidos de tipo “Artefacto”, pues esta funcionalidad no se implementa en cada nodo, sino que se le añade a través del módulo “*File*”.

Chain of Responsibility

El sistema de menús de Drupal sigue este patrón. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud y si el usuario tiene acceso a los recursos solicitados. 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. 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 (28).

Un ejemplo en la aplicación puede verse cuando se selecciona el vínculo “Variante”. El sistema detecta el módulo de búsqueda que puede encargarse de la petición, verifica si este tiene permisos para usar esta función y entonces permite acceder a su contenido.

Bridge

La capa de abstracción de bases de datos de Drupal se aplica de una forma similar a este patrón. Los módulos son escritos en una forma que es independiente del sistema que se está utilizando en la base de datos, y de esta forma, se puede añadir soporte para distintas bases de datos sin tener que modificar el código de un módulo (28).

En la aplicación puede apreciarse este patrón, en la abstracción de los módulos con respecto al diseño, pues puede cambiarse la estructura de la aplicación web o los tipos de contenido sin que por ello se afecte un módulo. Por ejemplo, la eliminación del tipo de contenido "Artículo" no provoca que el módulo "*CKEditor*" cese su funcionamiento en otro tipo de contenido donde se emplee, como "Disciplina".

2.7 Diagrama de Clases

El diagrama mostrado a continuación permite visualizar los paquetes que genera el CMS Drupal. Para alcanzar una mayor comprensión de la aplicación a desarrollar, es necesario tener en cuenta el funcionamiento del CMS que será utilizado, debido a esto se realiza una breve descripción de los paquetes generados por Drupal.

Includes: Contiene los ficheros de conexión a la base de datos.

Themes: Es el espacio donde se encuentran ubicadas las plantillas que serán utilizadas como tema para la interfaz de Drupal.

Modules: Contiene los módulos que permiten el funcionamiento del CMS.

Scripts: Almacena los ficheros que permiten que sean visualizados correctamente los datos.

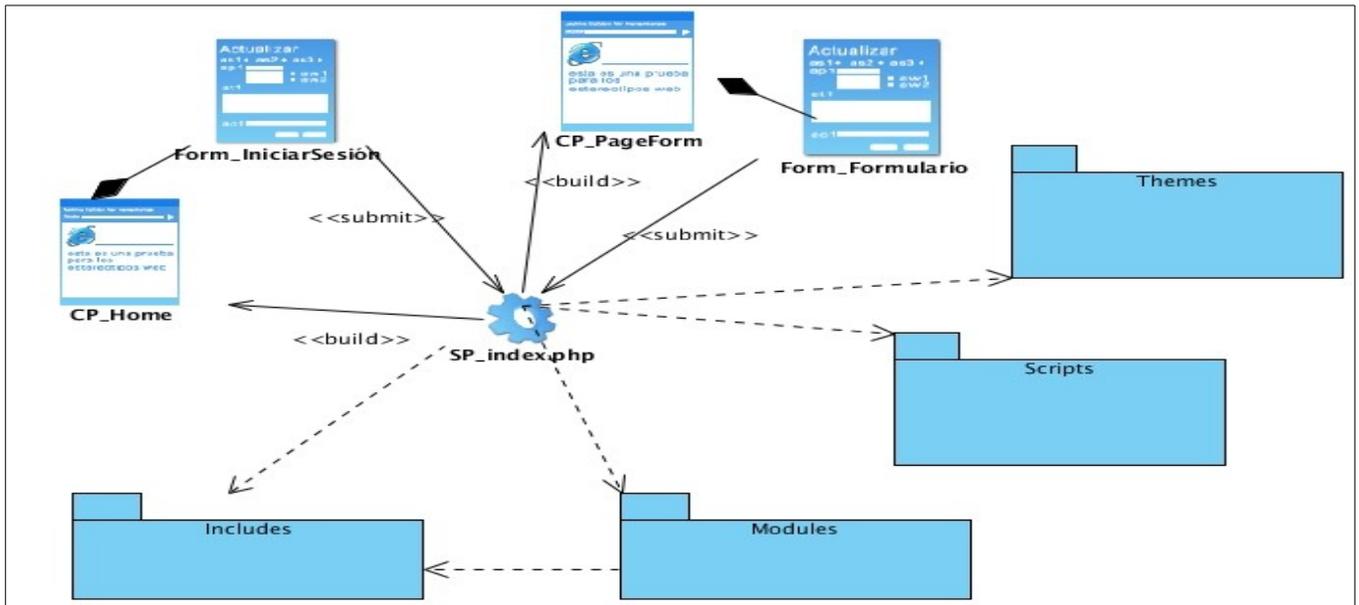


Figura 3: Diagrama de clases del CMS Drupal

Una vez confeccionado el diagrama de clases del diseño por paquetes, se modela el diagrama de clases del diseño para cada subpaquete como se muestra a continuación. Los restantes diagramas se ubican en el Anexo 3.

Paquete Gestionar artefactos

Para insertar un artefacto la página servidora `SP_Gestionar Artefacto` construye la página cliente `CP_Insertar artefacto`, la última se compone de un formulario `Form_Insertar artefacto` donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para editar un artefacto la página servidora construye la página cliente `CP_Editar artefacto`, la última se compone de un formulario `Form_Editar artefacto` donde se van a modificar los datos y va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora `CC_Gestionar artefacto`. Para eliminar un artefacto la página servidora construye la página cliente `CP_Eliminar artefacto`, la última se compone de un formulario `Form_Eliminar artefacto` donde se van a eliminar los datos y va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para mostrar un artefacto la página servidora construye la página cliente `CP_Mostrar artefacto`, la última se compone de un formulario `Form_Mostrar artefacto` donde se van a mostrar los datos y que va a ser ejecutado por

la página servidora; para acceder a los datos se hace mediante la clase controladora.

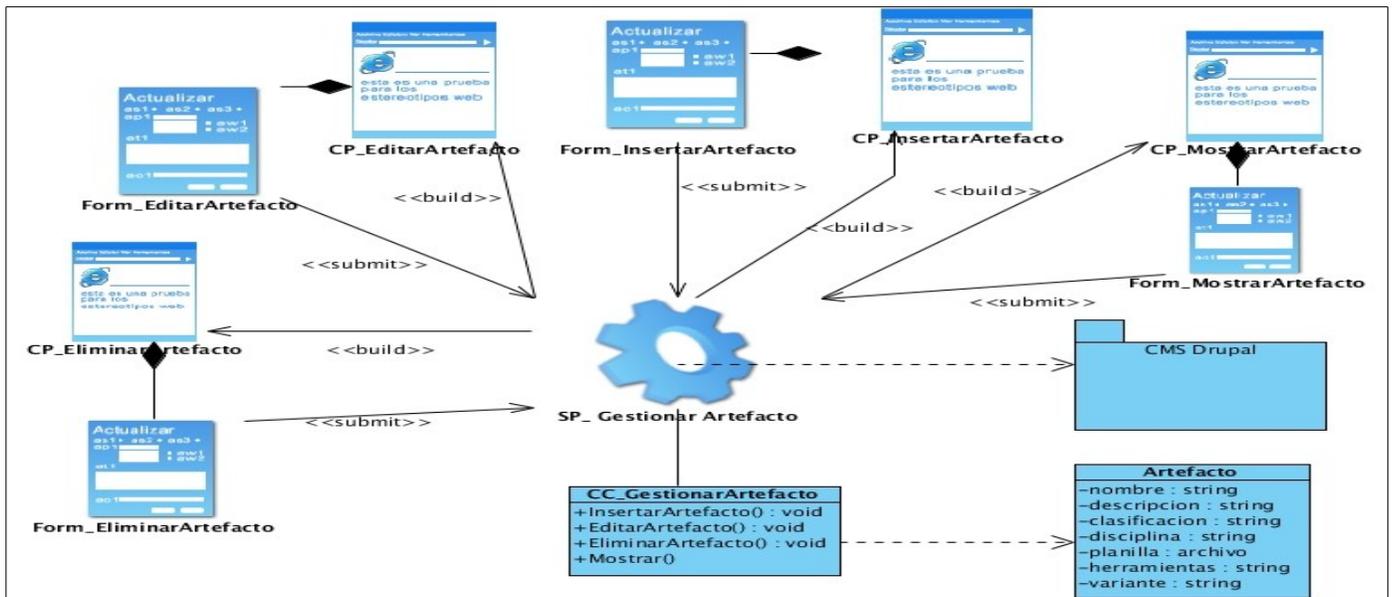


Figura 4. Diagrama de clases del paquete Gestionar artefacto

2.8 Diagrama de paquetes

Con el objetivo de lograr un mayor entendimiento de como funciona la aplicación, se hace necesario describir su estructura física. Permitiendo que las mejoras que se realicen posteriormente, estén acopladas a la estructura que se define, proporcionando uniformidad al sistema. En la siguiente figura se representa como va a quedar organizada la aplicación de soporte a la metodología Nova-OpenUp, donde se evidencia la arquitectura seleccionada.

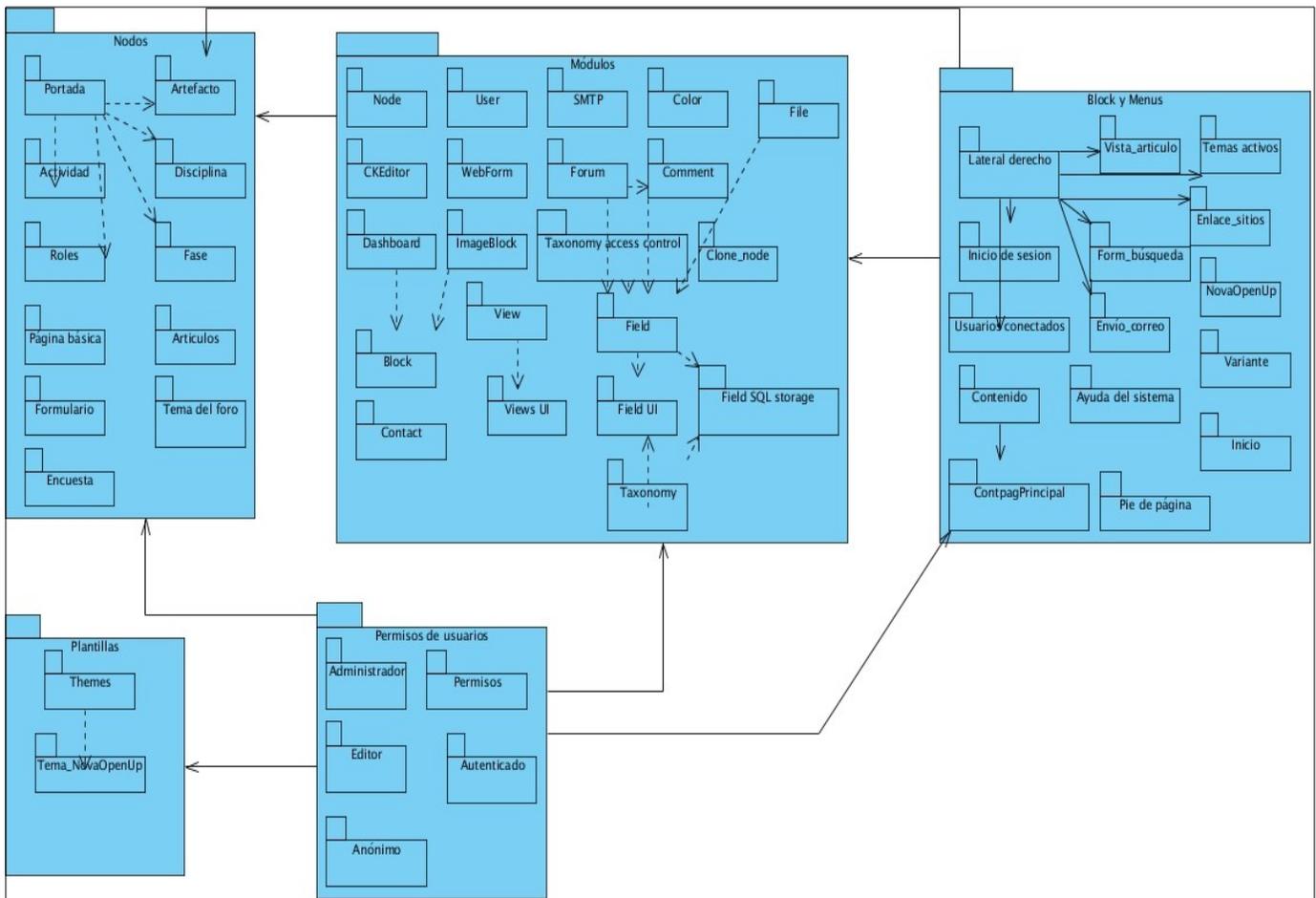


Figura 5. Diagrama de paquetes

2.9 Modelo de despliegue.

El Diagrama de Despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre dichos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. Entre los nodos existen relaciones que se representan a través de conectores.

En la figura se muestra el diagrama de despliegue que corresponde a la solución propuesta. La PC Cliente representa las computadoras de los usuarios que se conectan al sistema, las cuales realizan peticiones al Servidor Web mediante el protocolo HTTPS. Este servidor mantiene una conexión mediante el protocolo TCP/IP al servidor de bases de datos y mediante el protocolo SMTP con el servidor de correo.

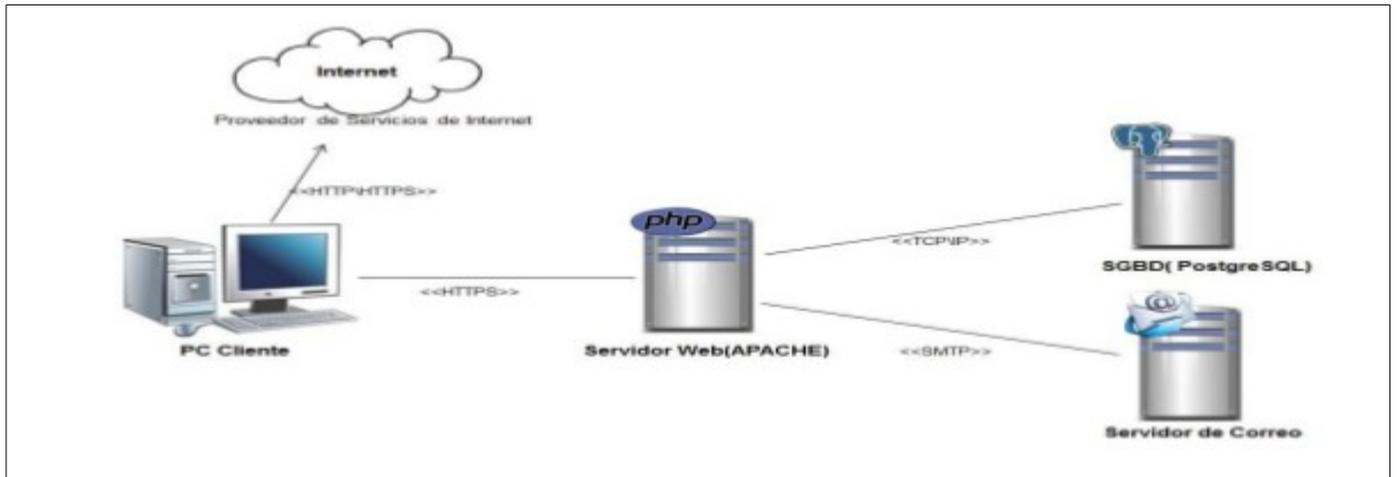


Figura 6. Modelo de despliegue

2.10 Conclusiones del capítulo

Con la caracterización realizada, se describieron los elementos significativos de la arquitectura de software, así como la definición de los requisitos funcionales y no funcionales, que permitieron establecer las funcionalidades de la aplicación. El sistema está basado en la arquitectura de n capas, definida por el CMS Drupal, separando los diferentes elementos del desarrollo y de este modo si ocurren cambios, no tenga que ser modificado completamente. Las actividades contempladas en el análisis posibilitaron refinar su estructura y profundizar en el análisis de este, mediante los diagramas realizados. Se obtuvo como resultado una representación de la estructura del sistema con el nivel de detalle suficiente para proporcionar una vista general del sistema que diera paso a su posterior implementación.

Capítulo 3: Programación y Evaluación de la aplicación web para la Metodología Nova-OpenUp.

3.1 Introducción

El presente capítulo comprende la implementación de los componentes de la aplicación para la metodología Nova-OpenUp, para ello se presenta el plan de iteraciones y las tareas de ingeniería por historia de usuario. También se van a describir los casos de prueba de aceptación a las que fueron sometidas las funcionalidades del sistema en cada una de las iteraciones.

3.2 Plan de Iteraciones

El Plan de Entrega está compuesto por tres iteraciones, tomando como base cada una de las historias de usuarios y el esfuerzo que se requiere para el desarrollo de estas, se procede a fragmentar el trabajo en distintas iteraciones obteniendo un trabajo incremental. Al final de la última iteración el sistema debe estar listo para incorporarlo al desarrollo de la distribución de GNU/Linux Nova.

Iteración 1

En la primera iteración se implementan las historias de usuario con mayor prioridad, obteniendo al final de la misma una primera versión de prueba y dando al sistema las primeras funcionalidades.

Iteración 2

En la segunda iteración se realiza la implementación de las historias de usuario con prioridad media, además se corrigen errores en las historias de usuario implementadas durante la primera iteración. De esta forma se obtiene la segunda versión de pruebas del software. Esta segunda versión se muestra a los clientes con el único objetivo de realizar cambios en base a la aceptación del mismo.

Iteración 3

En la tercera iteración, ya implementadas las funcionalidades especificadas, se realiza el desarrollo de las últimas historias de usuario con prioridad baja. De esta manera se obtiene la versión 1.0 del producto final.

Duración de iteraciones

Para una mayor organización del trabajo, se crea un plan de duración de las iteraciones. Este plan se realiza con el objetivo de reflejar cuáles serán las historias de usuario que serán implementadas en cada una de las iteraciones, así como el tiempo destinado a cada una de ellas y el orden en que se implementarán, lo que ayuda a obtener una idea general del tiempo que durará la confección total de la aplicación.

Iteración	Descripción de la iteración	Orden de las HU a implementar	Duración total
1	Desarrollo de las historias de usuarios de alta prioridad	1,2,3,4,5,6	6 semanas
2	Desarrollo de las historias de usuarios de media prioridad.	7, 8, 9, 10	5 semanas
3	Desarrollo de las historias de usuarios de baja prioridad.	11,12,13, 14	2 semanas

Tabla 7: Plan de iteraciones

3.3 Tareas de Ingeniería

A continuación se muestran las tareas de ingeniería realizadas para desarrollar las historias de usuarios de prioridad alta, el resto se puede encontrar en el Anexo 5.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Utilización del módulo User	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 24/02/2014	Fecha Fin: 26/02/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se utiliza el módulo User para el acceso de los usuarios en la aplicación.	

Tabla 8: Tarea de ingeniería. Utilización del módulo User

Tarea de Ingeniería

Aplicación de soporte a la metodología Nova-OpenUp

Número Tarea: 2	Número Historia de Usuario: 2
Nombre Tarea: Instalación y configuración del módulo SMTP	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 4/03/2014	Fecha Fin: 4/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se activa y se configura el módulo SMTP para el envío de correos y crear las cuentas de los usuarios en la aplicación.	

Tabla 9: Tarea de ingeniería. Instalación y configuración del módulo SMTP

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 3, 4, 5, 6, 7, 8, 9
Nombre Tarea: Instalación y Configuración del editor de texto Ckeditor.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 10/03/2014	Fecha Fin: 12/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se instala y configura el editor de texto Ckeditor para generar páginas con contenidos ricos en estilo sin necesidad de conocer el lenguaje HTML.	

Tabla 10: Tarea de ingeniería. Instalación y configuración del módulo Ckeditor

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 4, 8, 13
Nombre Tarea: Prueba del módulo View.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 día
Fecha Inicio: 25/03/2014	Fecha Fin: 27/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se prueba el módulo View, para la construcción de páginas dinámicas	

Tabla 11: Tarea de ingeniería. Prueba del módulo View

Tarea de Ingeniería

Número Tarea: 6	Número Historia de Usuario: 4, 8,13
Nombre Tarea: Instalación y configuración del módulo View.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 10/03/2014	Fecha Fin: 10/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se activa y se hace uso del módulo View para mostrar las vistas de artefactos y de actividades según la disciplina a la que correspondan.	

Tabla 12: Tarea de ingeniería. Instalación y configuración del módulo View

3.4 Estándar de codificación

- Se sigue el estándar de programación sugerido por la arquitectura de Drupal.
- Cuando se escribe en PHP, siempre se deben utilizar las etiquetas `<?php y ?>`, y en ningún caso la versión corta `<? y ?>`.
- Cada sentencia condicional debe llevar sus respectivas llaves.
- En los ficheros `.module` y `.inc` se omite la etiqueta de cierre de PHP (`?>`).
- En los archivos de plantilla `.tpl.php`, cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.
- Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.
- Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo.

3.5 Casos de Pruebas de aceptación

Las pruebas de software son aquellos procedimientos que se realizan para verificar la calidad de un producto de software y pueden ser aplicadas periódicamente. Estas tienen como objetivo fundamental la identificación de posibles errores que se presenten en la aplicación. En la metodología SXP se realizan pruebas de aceptación, estas son las pruebas que se realizan en conjunto con los usuarios finales antes del despliegue del sistema, los cuales emitieron un aval de conformidad, que se encuentra en el Anexo 7. Su objetivo es verificar que el software está listo

y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

A continuación se describen los casos de pruebas, basados en el método de pruebas de caja negra para algunos requisitos de prioridad alta. El resto de los casos de prueba se pueden encontrar en el Anexo 6.

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-1-1	Nombre Historia de Usuario: Autenticar usuario
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Esta prueba consiste en permitir autenticar al usuario en el sistema.	
Condiciones de Ejecución: El usuario debe haberse creado una cuenta con anterioridad en el sistema.	
Entrada / Pasos de ejecución: La entrada consiste en la introducción de los siguientes campos en el formulario de inicio de sesión: nombre de usuario: Mairim contraseña : mairim12 Luego de que el usuario seleccione el botón iniciar sesión, el sistema verifica en la base de datos que el usuario y contraseña existan y sean correctos, para luego habilitar los permisos que tienen el usuario autenticado sobre la aplicación.	
Resultado Esperado: El usuario entra a su sesión.	
Evaluación de la Prueba: Satisfactoria	

Tabla 13: Caso de prueba de aceptación. Autenticar usuario

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-3-3	Nombre Historia de Usuario: Gestionar fase
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Insertar fase.	
Condiciones de Ejecución: Para que esto sea posible el administrador o editor debe estar autenticado en el sistema.	

<p>Entrada / Pasos de ejecución:</p> <p>Para insertar una fase ir a la opción Fases en el menú Contenido de la vista Administración. La entrada consiste en la introducción de los campos que definen una fase:</p> <p>Título: Inicio</p> <p>Características: "La fase de inicio tiene como meta fundamental conocer y entender los objetivos..."</p> <p>Presionar el botón guardar.</p> <p>Resultado Esperado: El sistema verifica que los campos no estén vacíos y guarda los datos de la fase en la base de datos.</p> <p>Evaluación de la Prueba: Satisfactoria</p>

Tabla 14: Caso de prueba de aceptación. Gestionar fase

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-3-4	Nombre Historia de Usuario: Gestionar fase
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Editar fase.	
Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.	
<p>Entrada / Pasos de ejecución:</p> <p>La entrada consiste en la selección por parte del usuario de una fase y la opción Editar.</p> <p>El sistema muestra una interfaz con los campos para editar la fase.</p> <p>Título: Construcción</p> <p>Características: "La fase de construcción logra dar al producto capacidad operacional cumpliendo con los objetivo"</p> <p>El usuario selecciona la opción guardar.</p> <p>Resultado Esperado: El sistema verifica que los campos no estén vacíos y guarda los datos en la base de datos.</p> <p>Evaluación de la Prueba: Satisfactoria</p>	

Tabla 15: Caso de prueba de aceptación. Gestionar fase

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-4-7	Nombre Historia de Usuario: Gestionar artefacto
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Insertar artefacto	

<p>Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.</p>
<p>Entrada / Pasos de ejecución: Para insertar un artefacto ir a la opción Artefacto en el menú Contenido de la vista Administración. La entrada consiste en la introducción de los campos que definen un artefacto: Título: artefacto_01 Nombre: Especificación de requisito de software Clasificación: Obligatorio Disciplina: Requisitos Planilla de proyecto: 010113_Especificacion_de_requisitos_de_software.doc Herramientas o alternativas: La herramienta GESPRO permite especificar... El usuario selecciona la opción guardar.</p>
<p>Resultado Esperado: La aplicación verifica que los datos estén correctos y los almacena en la base de datos, de no ser así indica el error.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>

Tabla 16: Caso de prueba de aceptación. Gestionar artefacto

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-4-9	Nombre Historia de Usuario: Gestionar artefacto
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Eliminar artefacto.	
Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.	
Entrada / Pasos de ejecución: La entrada consiste en la selección por parte del usuario de un artefacto. El usuario selecciona la opción eliminar.	
Resultado Esperado: La aplicación muestra un mensaje de confirmación de la acción: "¿ Seguro desea eliminar el artefacto_02?. Esta acción no se puede deshacer. El usuario selecciona la opción "eliminar" y la aplicación elimina el artefacto_02 de la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 17: Caso de prueba de aceptación. Gestionar artefacto

Resultados de las pruebas a las funcionalidades

Se probaron todos los casos de pruebas que responden a las funcionalidades de la Aplicación de

soporte a la metodología Nova-OpenUp. En total fueron detectadas 26 no conformidades, las cuales fueron resueltas, donde los principales errores detectados se relacionan con errores ortográficos, textos en idioma inglés y cambios en el diseño.

A continuación se muestra una gráfica donde se desglosan las no conformidades detectadas en las tres iteraciones realizadas. En la primera iteración se obtuvieron 16 no conformidades, de las cuales se resolvieron 9 y las 7 restantes quedaron pendientes para la segunda iteración, en donde se detectaron 6 no conformidades y fueron resueltas 10. En la tercera iteración se resolvieron las que habían quedado pendientes en la segunda iteración y se detectaron 4 no conformidades.

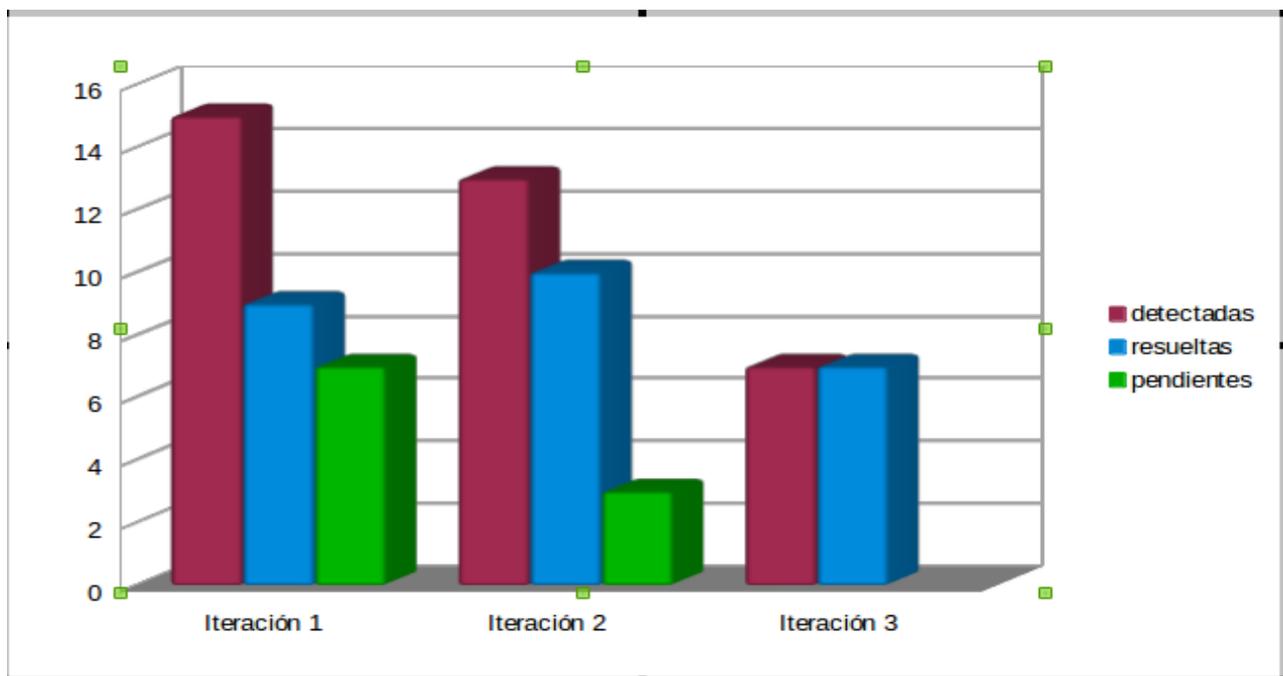


Figura 7. Resultados de las pruebas funcionales

3.6 Pruebas de carga y estrés

Las pruebas de rendimiento fueron realizadas en la herramienta Apache JMeter en su versión 2.9, diseñada para pruebas de carga de comportamientos funcionales y la medición del rendimiento. Prueba la resistencia y analiza el rendimiento en diferentes tipos de carga. El entorno en que fueron realizadas cumple con las siguientes características: 1 PC cliente, con un procesador Intel Core Dos Duo y 1GB de RAM.

A continuación se presenta el plan de pruebas de rendimiento para las diferentes acciones que los usuarios pueden realizar al conectarse a la aplicación.

Para una muestra de 50 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo que espera cada usuario para realizar una petición) la aplicación generó los siguientes reportes:

Funcionalidad	URL	Máx	Kb/s	Rendimiento	Avg
Portada	NovaOpenUp/inicio	54979	65.01	1.8/sec	37573.0
Nova-OpenUp	NovaOpenUp/uno	43013	64.80	2.3/sec	28992.0
VarianteAgil	NovaOpenUp/agil	46739	65.00	2.1/sec	31757.0
Artefactos	NovaOpenUp/artefactos	40572	66.61	2.4/sec	28310.0
Calcular variante	NovaOpenUp/calcular	45300	87.63	2.2/sec	41548.0

Tabla 18: Resultados de pruebas de rendimiento

Los datos de la tabla 18 muestra como resultado, que para las pruebas de rendimientos realizadas a las funcionalidades, los tiempos máximos de respuestas de la aplicación fueron de 46121 milisegundos, con un rendimiento de 2.16 segundos por cada petición. La interpretación de estos resultados demuestra que la aplicación se comporta satisfactoriamente, atendiendo a los tiempos de respuestas trazados por los desarrolladores, el cual debe oscilar de 2 a 5 segundos por petición.

3.7 Pantallas de la aplicación



Fig 8: Portada de la aplicación de soporte a la metodología Nova-openUp

3.8 Conclusiones del Capítulo

En este capítulo se describieron las tareas de ingeniería y el plan de iteraciones a seguir en el desarrollo de la aplicación. Se definió el estándar de código a utilizar en la implementación de la solución. Se confeccionaron los casos de pruebas de aceptación con el fin de comprobar el correcto funcionamiento del sistema y demostrar la conformidad de los requisitos especificados.

Se obtuvo una aplicación web que posibilita la interacción dinámica entre los usuarios y la metodología Nova-OpenUp. Cumple las características de permitir descargar las planillas de los artefactos y de indicar la variante de la metodología que deben utilizar los equipos de desarrollo, a través de la selección de las características del proyecto a desarrollar.

Conclusiones generales

Como el resultado final de esta investigación se ha determinado que los objetivos específicos fueron cumplidos satisfactoriamente, arribándose a las siguientes conclusiones:

- En esta investigación se realizó un estudio de varias aplicaciones web destinadas a brindar soporte a las metodologías de desarrollo de software, dicho estudio logró sentar las bases para la creación de una aplicación de soporte, que permite mostrar las variantes que puede adoptar la metodología Nova-OpenUp.
- Se desarrolló una aplicación web basada en el CMS Drupal que permite gestionar desde el punto de vista metodológico, cada uno de los componentes que conforman a la metodología Nova-OpenUp, posibilitando mejorar la productividad, la calidad de las acciones y las decisiones de los equipos de desarrollo que la utilizan.
- Las pruebas de software efectuadas al sistema permitieron mejorar la calidad de la aplicación.

Recomendaciones

Existen algunas funcionalidades que quedan fuera del alcance de la investigación y que se considera que pueden aumentar el valor agregado de la aplicación de soporte a la metodología Nova-OpenUp. Por tanto, se recomienda:

- Añadir Comercio Electrónico a la aplicación debido al gran auge de esta técnica en los sistemas dedicados a la promoción, comercialización y servicios de productos.
- Trazar una política de actualizaciones periódicas utilizando el módulo de actualización del CMS Drupal, con el objetivo de evitar fallas de seguridad o mal funcionamiento del sistema.

Bibliografía

1. Principales Definiciones de Términos más Usados en Internet. 2009 [online]. [Accessed 16 Septiembre 2013]. Available from: <http://www.informaticamilenium.com.mx/es/temas.html>
2. Cuadro comparativo: Página Web, Sitio Web y Portal Web | Histointernet's Blog. 2010 [online]. [Accessed 16 Septiembre 2013]. Available from: <http://histointernet.wordpress.com/2010/10/07/cuadro-comparativo-pagina-web-sitio-web-y-portal-web/>
3. Características de los portales. 2010 [En línea]. [Citado el 17 de septiembre de 2013]. Disponible en: <http://www.eumed.net/tesis/2007/cavl/caracteristicas%20de%20los%20portales.html>
4. Desarrolloweb.com. "Programación en PHP". [En línea]. [Citado el: 16 de septiembre de 2013.] Disponible en: <http://www.desarrolloweb.com/manuales/>
5. Bakken, Sæther, Schmid, Stig. y Egon. Manual de Php. PHP Documentation Group. [En línea] 2003. [Citado el: 16 de septiembre de 2013.]. Disponible en: <http://www.php.net/docs.php>.
6. Miguel Ángel Álvarez. Definición y ejemplos de sistemas CMS: Content Manager System (Sistema gestor de contenidos). [En línea]. [Citado el 24 de septiembre de 2013]. Disponible en: <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>
7. Sistema de gestión de Contenidos: Artículo: "CMS Quick Guide". [En línea] [Citado septiembre de 2013.]. Disponible en: <http://www.opensourcecms.com/index.php?option=content&task=view&id=388>.
8. SANCHEZ, J. ¿Qué es WordPress? [En línea] TodoWordpress, 2010. [Consultado el: 23 de septiembre de 2013]. Disponible en: <http://www.todowp.org/que-es-wordpress/>
9. MIFSUD, E. MONOGRÁFICO: Drupal 7. [En línea] Observatorio Tecnológico, 2011. [Consultado el: 23 de septiembre de 2013]. Disponible en: <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1022-monograficodrupal-7?start=2>.
10. JARAMILLO, D. R. Sistema de Gestión de contenidos Joomla. [En línea] Monografias, 2010. [Consultado el: 23 de septiembre de 2013]. Disponible en: <http://www.monografias.com/trabajos81/sistema-gestion-contenidos-joomla/sistema-gestion->

[contenidos-joomla.shtml](#)

11. POSTGRESQL. Sobre PostgreSQL. [En línea] PostgreSQL.es, 2009. [Consultado el: 18 de septiembre de 2013]. Disponible en: http://www.postgresql.org.es/sobre_postgresql
12. MUSCIANO, Ch.; KEMEDY, B. HTML la guía completa. 2da edición. México, McGRAW-HILL INTERAMERICANA EDITORES, 1999.
13. EGUÍLUZ, J. Introducción a CSS. [En línea] LibrosWeb, 2009. [Citado el 22 de septiembre de 2013.]. Disponible en: <http://www.librosweb.es/css/index.html>
14. MÁRQUEZ, J.; SAMPEDRO, L.; VARGAS, F. Instalación y configuración de Apache, un servidor Web gratis. Red de Revista Científica de América Latina y el Caribe, España y Portugal, 2002, No. 012, p. 10–23. . Citado el 23 de septiembre de 2013. Disponible en: <http://redalyc.uaemex.mx/pdf/852/85201202.pdf>
15. Internet Information Services. 2009 [En línea]. [Citado el 23 de septiembre de 2013]Disponible en: <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>
16. Universidad de Ciencias Informáticas. Entorno Virtual de Aprendizaje. Curso Optativo de Arquitectura de la Información 2011-2012;citado el 24 de septiembre de 2013. Disponible en: http://eva.uci.cu/file.php/193/Tema_1/Documentos/Conferencia_1/Conferencia_1.pdf.
17. Software y Aplicaciones Web.[En línea] [Citado el: 21 de Febrero de 2014.] <http://jtentor.com.ar/post/Arquitectura-de-N-Capas-y-N-Niveles.aspx>.
18. Gilfillan, Ian. La Biblia de MySQL. [Citado el 24 de septiembre de 2013]
19. Definición de Soporte » Concepto en Definición ABC. [online]. 2009. [Accessed 21 May 2014]. Available from: <http://www.definicionabc.com/general/soporte.php>
- 20 JUAN CAMILO MONTOYA VARGAS. EL SOPORTE TECNICO. [online]. 1 October 2009. [Accessed 21 May 2014]. Available from: <http://www.slideshare.net/guestfe3d71c/el-soporte-tecnico>
21. CEDEÑO, A.; GARCÍA, E. A. Portal Web de la Unión Nacional de Juristas de Cuba. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas. Ciudad Habana, 2011
22. MENDOZA, M.A.s. Metodologías de Desarrollo de Software. 2004; [Citado el 13 de noviembre de 2013] Available from: <http://www.willydev.net/Descargas/cualmetodologia.pdf>.

23. PEÑALVER, GLADYS MARCI. 2008. *SXP, metodología ágil para proyectos de Software Libre*. La habana : s.n., 2008.
24. HERNANDIS, J.A. *Why Visual Paradigm for UML?*. 2010; [Citado el 23 de enero de 2014]. Disponible en: <http://www.versionzero.com/noticia/210/visual-paradigm-for-uml>.
25. WEB TALLER. Firebug: extensión para desarrolladores web. [En línea] Web Taller.com, 2012. [Consultado el: 28 de febrero de 2014]. Disponible en: <http://www.webtaller.com/maletin/articulos/firebugextension-desarrolladores-web.php>.
26. FONTELA, Á. ¿Qué es Bootstrap?. [En línea] Open Web CMS, 2013. [Consultado el: 3 de marzo de 2014]. Disponible en: <http://openwebcms.es/2013/que-es-bootstrap/>
27. ZAKAS, N. C. *Professional JavaScript® for Web Developers*. 2da Edición. Indianápolis, Wiley Publishing, Inc. ,2009. p.960
28. Patrones de diseño en Drupal. 2010 [En línea] [Citado el: 22 de Febrero de 2014.] <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.
29. Tutorial Drupal. [En línea]. [Citado el: 22 de febrero de 2014] <http://www.cursosdrupal.com/content/arquitectura>.
30. CIDI, Centro de Ideoinformática. Base Tecnológica. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2012.
31. Geany : Home Page. [online]. 2009 [Accessed 22 abril 2014]. Available from: <http://geany.org/>
32. **Robert T. Douglass, Mike Little, and Jared W. Smith.** *Building Online Communities with Drupal, phpBB, and WordPress*. 2009.
33. **Haney, Andrew.** *Drupal 100 Success Secrets*.
34. **Matt Butcher, Greg Dunlap, Matt Farina, Larry Garfield, Ken Rickard, John Albin Wilkins.** *Drupal 7 Module Development*.
35. **Noble, Mark.** *Drupal 7 First Look*.
36. librosweb.es *Introducción a XHTML*. 2008 [En línea] <http://librosweb.es/xhtml/index.html>
37. librosweb.es *Introducción a CSS*. 2008 [En línea] <http://librosweb.es/css/index.html>
38. **EGIGUREN, R.** *Introducción a Drupal*. [En línea] slideshare.net, 2010. [Consultado el: 12 de noviembre de 2013]. Disponible en: <http://www.slideshare.net/keopx/introduccion-a-drupal-2971232>.

Glosario de términos

API: Interfaz de programación de aplicaciones o API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

GPL: Es el acrónimo en inglés de *General Public License* (Licencia Pública General). Regula los derechos de autor de los programas de software libre promovido por la Fundación de Software Libre.

HTTPS: Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la web.

Módulo: En informática un módulo es una parte autónoma de un programa de ordenador.

Repositorio: El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

Web: Sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web, páginas a las que se puede acceder usando un navegador.

World Wide Web (WWW): Red global mundial, es un sistema de hipervínculos que son accesibles a través de Internet, estos hipervínculos permiten a las personas que mediante el uso de un navegador, se pueda ir de una página a otra siguiendo las direcciones que indican los hipervínculos (o hipertexto).

Framework: Un framework Web, es un framework diseñado para apoyar el desarrollo de sitios web dinámicos. Ofrece un conjunto de componentes para acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Scripts: Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos. Estos pueden estar embebidos en otro lenguaje para aumentar las funcionalidades, como es el caso los scripts PHP o JavaScripts en código HTML.

Anexo 1

Guía de preguntas de la entrevista

1. ¿Cualquier desarrollador en el área de Software Libre tiene acceso a la utilización de la Metodología Nova-OpenUp?
2. ¿El empleo de la Metodología Nova-OpenUp resulta un proceso complejo para los proyectos con experiencia?
3. ¿Cuáles son las principales dificultades que se observan en la configuración de la Metodología, por equipos inexpertos en temas de desarrollo de software?
4. ¿Actualmente que cantidad de proyectos pertenecientes al Centro de Software Libre (CESOL) emplean la Metodología Nova-OpenUp?
5. ¿Que elementos son los que deben tener en cuenta los equipos de trabajo a la hora de seleccionar la variante de la Metodología Nova-OpenUp?

Entrevista realizada a : Ms.C Yusleydi Fernández del Monte

Anexo 2

Historias de usuarios

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar usuarios
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 0.5
Riesgo en Desarrollo: Medio	Puntos Reales: 0.5
Descripción: El usuario administrador puede crear, editar y eliminar las cuentas de usuario para así brindarle una serie de privilegios a las cuentas de usuario en dependencia del rol asignado a las mismas.	
Observaciones: La creación de las cuentas las puede realizar el usuario administrador o el usuario anónimo, para crear una cuenta el usuario debe completar los campos: Nombre de usuario, contraseña, dirección de correo, nombre, apellidos, categoría, proyecto, módulo. Para editar la cuenta el usuario selecciona su cuenta una vez que esté en el sistema, selecciona la opción Editar y accede a sus datos. Para eliminar la cuenta, tanto el usuario autenticado como el administrador tienen permisos para realizar esta acción.	
Prototipo de interfaz:	

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar roles
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4
Riesgo en Desarrollo: Medio	Puntos Reales: 0.4
Descripción: Permite adicionar, modificar, eliminar y mostrar la información de cada uno de los roles que propone la metodología Nova-OpenUp.	
Observaciones: Para adicionar los roles se debe llenar los campos siguientes: nombre del rol, habilidades y responsabilidades. Para modificar la información de los roles, se debe seleccionar el rol y escoger la opción Editar. Para eliminar la información de los roles, se debe seleccionar previamente el rol y escoger la opción Eliminar.	

Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores.
Para mostrar los roles, se debe localizar y presionar clic sobre su nombre o la imagen que los identifica.

Prototipo de interfaz: Anexo 4

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Gestionar base conceptual
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 0.4
Riesgo en Desarrollo: Medio	Puntos Reales: 0.4
Descripción: Permite adicionar, modificar, eliminar y mostrar los conceptos básicos de la metodología Nova-OpenUp.	
Observaciones: Para adicionar la base conceptual de la metodología se debe llenar los campos título y cuerpo. Para modificar la base conceptual, se selecciona la opción Editar. Para eliminar la información de la base conceptual se selecciona la opción Eliminar. Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores. Para mostrar la base conceptual se debe localizar en el menú inicial la opción Nova-OpenUp.	
Prototipo de interfaz: Anexo 4	

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Gestionar actividades
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1.2
Riesgo en Desarrollo: Medio	Puntos Reales: 1.2
Descripción: Permite insertar, modificar, eliminar y mostrar cada una de las actividades que propone la metodología Nova-OpenUp	
Observaciones: Para adicionar las actividades se debe llenar los campos título, cuerpo, tipo y ligera. Para modificar la información de las actividades, se debe seleccionar la actividad y escoger la opción Editar. Para eliminar la información de las actividades, se debe seleccionar previamente la actividad y escoger	

la opción Eliminar.

Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores. Para mostrar las actividades, se debe localizar, presionar clic sobre su nombre o la imagen que las identifica y luego seleccionar las actividades de la disciplina que desea ver el usuario.

Prototipo de interfaz: Anexo 4

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Gestionar variante
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1.4
Riesgo en Desarrollo: Medio	Puntos Reales: 1.4
Descripción: Permite calcular, adicionar, editar, eliminar y mostrar la variante ágil de la metodología Nova-OpenUp.	
Observaciones: Para calcular la variante los usuarios completan un cuestionario con las principales características del proyecto, el sistema le devuelve si debe utilizar la variante ágil o la variante tradicional de la metodología. Para adicionar la variante se debe llenar los campos título, características de la variante y las actividades. Para modificar su información, se debe seleccionar la variante y la opción Editar. Para eliminar la variante, se debe seleccionar previamente y escoger la opción Eliminar. Los usuarios deben tener permisos de administración o edición para realizar las acciones anteriores. Para mostrar la variante el usuario debe estar autenticado y luego seleccionar en el menú principal la opción Variante.	
Prototipo de interfaz: Anexo 4	

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Gestionar ayuda
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 2
Prioridad en Negocio: media	Puntos Estimados: 0.4
Riesgo en Desarrollo: Medio	Puntos Reales: 0.4

<p>Descripción: Permite insertar, editar, eliminar y mostrar las consultas que realizan los usuarios en la aplicación.</p> <p>Observaciones: Para insertar preguntas y respuestas se selecciona en la sección de navegación la opción Ayuda, luego se escoge el tema sobre el cual se quiere debatir y se añade un nuevo comentario. Para modificar o eliminar las preguntas y respuestas el usuario debe poseer el rol de administrador o editor en la aplicación, selecciona su comentario y la opción Eliminar o Editar en el caso correspondiente.</p> <p>Para realizar las acciones anteriores, el usuario debe estar previamente registrado en la aplicación. Para mostrar los comentarios se selecciona en la sección de navegación la opción Ayuda y luego el Tema que se desea ver.</p> <p>Prototipo de interfaz: Anexo 4</p>

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Enviar mensajes vía correo
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 3
Prioridad en Negocio: baja	Puntos Estimados: 0.2
Riesgo en Desarrollo: Medio	Puntos Reales: 0.2
<p>Descripción: Permite el envío de mensajes de correo entre los usuarios que visitan la aplicación por medio de un formulario al que se le completan los campos: nombre, dirección de correo, título del mensaje, mensaje, archivo que desee adjuntar.</p> <p>Observaciones:</p> <p>Prototipo de interfaz: Anexo 4</p>	

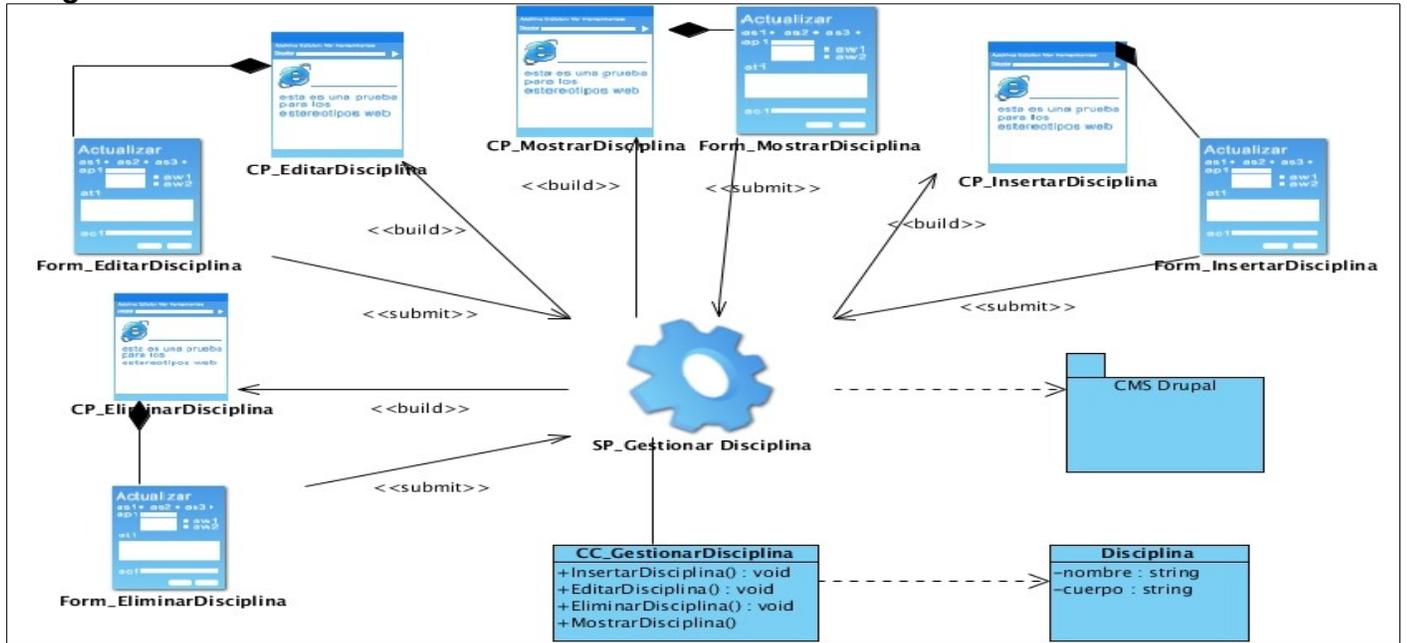
Historia de Usuario	
Número: 12	Nombre Historia de Usuario: Gestionar vínculos
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 3
Prioridad en Negocio: baja	Puntos Estimados: 0.5
Riesgo en Desarrollo: Medio	Puntos Reales: 0.5
<p>Descripción: Permite adicionar, eliminar y mostrar enlaces a otros sitios.</p> <p>Observaciones: Para adicionar un enlace se selecciona en el menú principal la opción infraestructura tecnológica y se añaden los vínculos relacionados con las herramientas que utiliza la metodología. En la sección Enlaces Externos, se pueden adicionar otros vínculos de interés relacionados con el proceso de desarrollo de software.</p> <p>Prototipo de interfaz: Anexo 4</p>	

Historia de Usuario	
Número: 13	Nombre Historia de Usuario: Visualizar usuarios conectados
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 3
Prioridad en Negocio: baja	Puntos Estimados: 0.1
Riesgo en Desarrollo: Medio	Puntos Reales: 0.1
Descripción: Permite mostrar un listado con los usuarios que se encuentran en línea.	
Observaciones: Para mostrar el listado de usuarios conectados se crea una vista con los usuarios autenticados en el sistema.	
Prototipo de interfaz: Anexo 4	

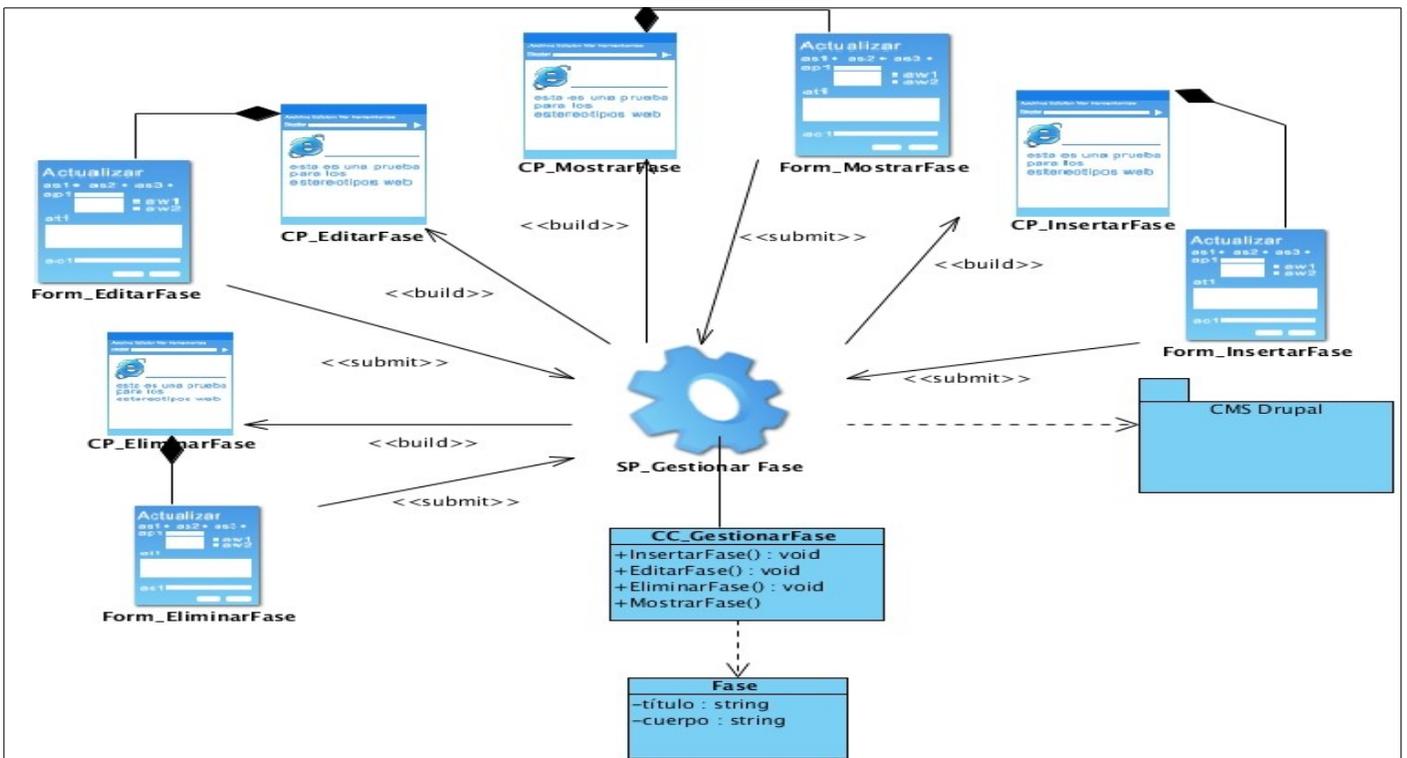
Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Publicar artículos
Modificación de Historia de Usuario Número: ninguna	
Usuario: Isabel Barbara Sanchez Quiala	Iteración Asignada: 3
Prioridad en Negocio: baja	Puntos Estimados: 0.4
Riesgo en Desarrollo: Medio	Puntos Reales: 0.4
Descripción: Permite adicionar, editar y eliminar artículos relacionados con el proceso de desarrollo de software y elementos relevantes relacionados con la metodología.	
Observaciones: Para adicionar un artículo se deben llenar los campos título, cuerpo, una lista de palabras para describir el artículo, imagen y archivo adjunto. En caso de editar un artículo, primeramente este debe ser seleccionado, junto a la opción Editar que posee el artículo. Para eliminar un artículo, se debe seleccionar el artículo y la opción Eliminar. El sistema le preguntará al usuario si esta seguro que desea realizar esta acción, en caso de que el usuario acepte, se elimina de la base de datos.	
Prototipo de interfaz: Anexo 4	

Anexo 3

Diagramas de clases del diseño

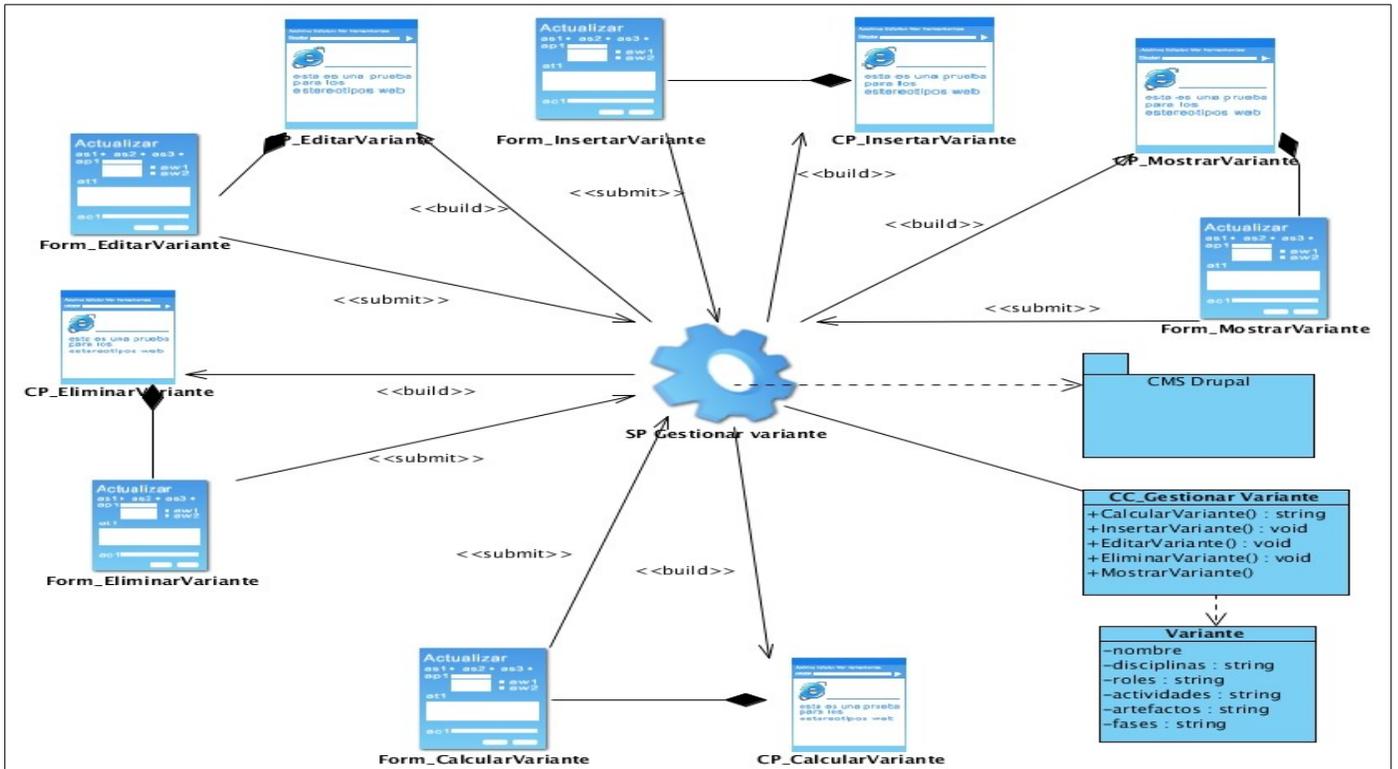


Paquete Gestionar disciplinas

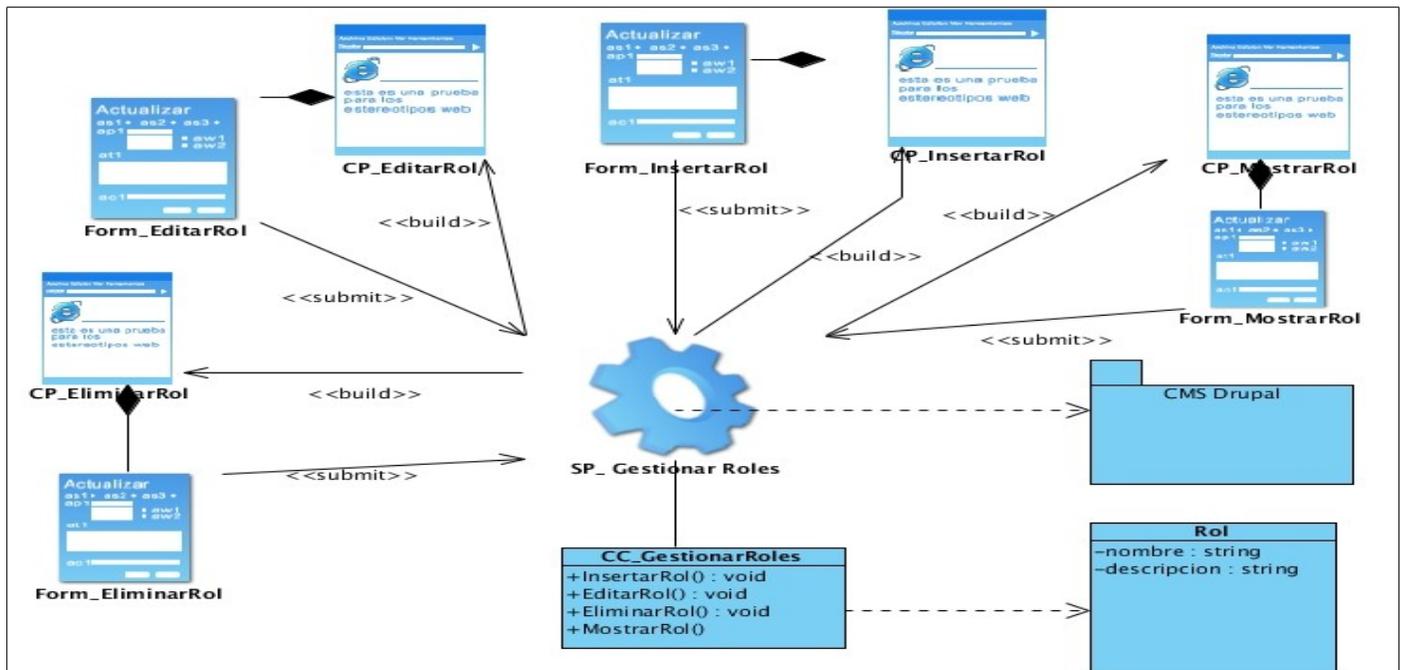


Paquete Gestionar fases

Aplicación de soporte a la metodología Nova-OpenUp



Paquete Gestionar Variante



Paquete Gestionar rol

Aplicación de soporte a la metodología Nova-OpenUp

Crear Artefactos ↻

Inicio » Agregar contenido

Title *

Nombre

Clasificación

Descripción

Disciplina

Plantilla de proyecto

No se ha seleccionado ningún archivo.

Los archivos deben ser menores que **30 MB**.
Tipos de archivo permitidos: **txt pdf doc odt docx xls**.

Herramientas o alternativas

variante

Prototipo artefacto

Crear actividad ↻

Inicio » Agregar contenido

Title *

Body (Editar resumen)

Formato de texto

- Las direcciones de las páginas web y las de correo se convierten en enlaces automáticamente.
- Etiquetas HTML permitidas: <a> <cite> <blockquote> <code> <div> <dt> <dd>

- Saltos automáticos de líneas y de párrafos.

tipo

ligera

Prototipo actividad

Aplicación de soporte a la metodología Nova-OpenUp

Envíe este formulario para contactar con nosotros. Nos pondremos en contacto con usted lo antes posible

Escriba su Nombre

Dirección de correo electrónico

Título del mensaje

Mensaje

Subir algún archivo

No se ha seleccionado ningún archivo.

Enviar notificaciones de correo

Crear Artículo <

[Inicio](#) > [Agregar contenido](#)

Title *

Body (Editar resumen)

Fuente HTML

Formato

B **I** **U** **X₂** **X₃**

Switch to plain text editor

Formato de texto

- Las direcciones de las páginas web y las de correo se convierten en enlaces automáticamente.
- Etiquetas HTML permitidas: <a> <code> <div> <div> <div>

- Saltos automáticos de líneas y de párrafos.

Tags

Escriba una lista de palabras separadas por comas para describir su contenido.

Image

No se ha seleccionado ningún archivo.

Suba una imagen para acompañar este artículo.
Los archivos deben ser menores que **30 MB**.
Tipos de archivo permitidos: **png gif jpg jpeg**.

archivo adjunto

No se ha seleccionado ningún archivo.

Los archivos deben ser menores que **30 MB**.
Tipos de archivo permitidos: **txt pdf doc odt docx xls xlsx rar tagz zip**.

Prototipo artículos

Anexo 5

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 4, 8, 13
Nombre Tarea: Creación de vistas	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 día
Fecha Inicio: 10/03/2014	Fecha Fin: 12/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se desarrollan las vistas de artefactos, de actividades y el listado de los últimos usuarios conectados.	

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 14
Nombre Tarea: Prueba del módulo Comment.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 23/03/2014	Fecha Fin: 24/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se realiza un estudio sobre el funcionamiento del módulo Comment y se utiliza para permitir a los usuarios comentar y discutir los artículos que se publican en la aplicación.	

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 10
Nombre Tarea: Instalación y configuración del módulo Forum.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 23/03/2014	Fecha Fin: 23/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se realiza la instalación y configuración del módulo Forum para crear el foro de discusión, como herramienta de comunicación en la aplicación.	

Aplicación de soporte a la metodología Nova-OpenUp

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 10
Nombre Tarea: Prueba del módulo Field.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 día
Fecha Inicio: 23/03/2014	Fecha Fin: 24/03/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se realiza un estudio y se realizan pruebas sobre el funcionamiento del módulo Field UI para adicionar campos a los contenidos que se presentan en la aplicación.	

Tarea de Ingeniería	
Número Tarea: 13	Número Historia de Usuario: 9, 11
Nombre Tarea: Instalación y configuración del módulo Webform.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 día
Fecha Inicio: 8/04/2014	Fecha Fin: 10/04/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Se instala y configura el módulo WebForm para elaborar formularios de encuesta y contacto.	

Tarea de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: 9, 11
Nombre Tarea: Creación de formularios.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 4 día
Fecha Inicio: 24/04/2014	Fecha Fin: 27/04/2014
Programador Responsable: Isabel Barbara Sanchez Quiala	
Descripción: Creación de un formulario de encuesta para calcular la variante y un formulario de contacto para el envío de correos entre los usuarios.	

Anexo 6

Casos de pruebas de aceptación

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-3 -4	Nombre Historia de Usuario: Gestionar fase
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Editar fase.	
Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.	
Entrada / Pasos de ejecución: La entrada consiste en la selección por parte del usuario de una fase y la opción Editar. El sistema muestra una interfaz para editar la fase. El usuario selecciona la opción guardar.	
Resultado Esperado: El sistema verifica que los datos sean correctos y guarda los datos en la base de datos	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-3 -5	Nombre Historia de Usuario: Gestionar fase
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Eliminar fase.	
Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.	
Entrada / Pasos de ejecución: La entrada consiste en la selección por parte del usuario de una fase. El usuario selecciona la opción eliminar.	
Resultado Esperado: La aplicación muestra un mensaje de confirmación de la acción: "¿ Seguro desea eliminar la fase Construcción?. Esta acción no se puede deshacer. El usuario selecciona la opción "eliminar" y la aplicación elimina la fase de la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-3 -6	Nombre Historia de Usuario: Gestionar fase
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Mostrar fase.	
Condiciones de Ejecución: Ninguna	
Entrada / Pasos de ejecución: La entrada consiste en la selección por parte del usuario de la imagen que representa las fases en la página de Inicio. El usuario selecciona la fase que desea ver	
Resultado Esperado: La aplicación muestra la información de la fase seleccionada.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: ASMN-5-11	Nombre Historia de Usuario: Gestionar disciplina
Nombre de la persona que realiza la prueba: Isabel Barbara Sanchez Quiala	
Descripción de la Prueba: Prueba a la funcionalidad Insertar disciplina	
Condiciones de Ejecución: Para que esto sea posible el usuario administrador o editor debe estar autenticado en el sistema.	
Entrada / Pasos de ejecución: Para insertar una disciplina ir a la opción disciplina en el menú Contenido de la vista Administración. La entrada consiste en la introducción de los campos que definen una disciplina: Título: Requisitos Cuerpo: Esta disciplina proporciona las tareas necesarias para llevar a cabo la obtención... El usuario selecciona la opción guardar.	
Resultado Esperado: La aplicación verifica que los datos estén correctos y los almacena en la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Anexo 7

Aval de conformidad

La Habana, 2 de junio de 2014

Se avala satisfactoriamente, la aplicación web de soporte a la metodología Nova-OpenUp, teniendo en cuenta las mejoras positivas que se esperan, desde el punto de vista que debe proveer mejores formas de comunicación e interacción entre la metodología y los usuarios que la utilizan.

Para que así conste firma Yusleydi Fernández del Monte, cliente de la investigación desarrollada.

Firma:

