



Trabajo de Diploma para optar por el título de Ingeniero Informático

Subsistema de gestión de contenidos para
aplicaciones con tecnología multimedia.

Prima-Frame

Autores: Eric Manuel Cobas Peña.

José Antonio Soto Pérez.

Tutores: Ing. Michel Miranda Cairo.

Ing. Abel Ernesto Lorente Rodríguez.

Ciudad de La Habana

Abril, 2009

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo con carácter exclusivo.

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

Autores: José Antonio Soto Pérez
Eric Manuel Cobas Peña

Tutor: Ing. Michel Cairo Miranda
Ing. Abel Ernesto Lorente Rodríguez

Firma de los autores

Firma de los tutores

Agradecimientos

A todos aquellos que de una manera u otra han hecho posible la realización de este trabajo, y en especial a mis amigos por estar ahí siempre que los he necesitado.

José Antonio Soto Pérez.

A mi familia y a mi esposa por su apoyo incondicional.

Al equipo Voltus: Jose, Jorge, Fabián y Humberto por guiarme por el camino correcto y ser mi familia en la UCI.

A mis amigos de CICPC por su ayuda por brindarme alegrías en los momentos más difíciles.

A Reyniel Herrera Pereda, gracias peludo.

Eric Manuel Cobas Peña.

Dedicatoria

Dedico este trabajo a mi mamá y a mi papá que han estado siempre presentes sin importar las circunstancias y me han guiado por el camino de la vida con su amor y confianza; a mi vieji por su amor y devoción para conmigo; a mi novia por su paciencia y todo el apoyo que me ha dado; a mis tías que han sido como madres para mí, a mis hermanos por su incondicionalidad, a mis primos, en general a mi familia por ser tan especial. Gracias a todos.

José Antonio Soto Pérez.

A la memoria de mi padre...

Eric Manuel Cobas Peña.

Resumen

En la Universidad de las Ciencias Informáticas (UCI), en el proceso de gestión de contenidos para aplicaciones con tecnología multimedia, no se usa ninguna herramienta que automatice el proceso, para ganar en tiempo y eficacia. El presente trabajo consiste en desarrollo de una herramienta libre para la gestión de los contenidos de las aplicaciones con tecnología multimedia, que agilice el desarrollo de los proyectos, disminuya el gasto de recursos humanos y mejore la calidad del producto final. El documento recoge, en un primer momento, el estudio del estado del arte y la investigación realizada. Luego detalla el proceso de desarrollo de software seguido para la creación de la solución y los aspectos técnicos de la creación del mismo.

Índice

Declaración de autoría	III
Agradecimientos	IV
Dedicatoria.....	V
Resumen	VI
Índice	VII
Introducción	1
Capítulo 1 Fundamentación teórica.....	4
1.1 Introducción	4
1.2 Descripción General del Objeto de Estudio.....	4
1.3 Identificación de la audiencia	5
1.4 Análisis de soluciones existentes.....	5
1.4.1 MALTED 3:	5
1.4.2 Constructor Atenex 3:	6
1.5 Tendencias y Tecnologías	6
1.5.1 Metodología de desarrollo	7
1.5.1.1 Extreme Programing (XP)	7
1.5.1.2 Rational Unified Process	8
1.5.2 Lenguaje Unificado de Modelado 2.0.....	8
1.5.3 Herramientas CASE	9
1.5.4 Visual Paradigm for UML 6.4	10
1.5.5 Lenguajes de Codificación.....	10
1.5.5.1 ActionScript 3.0.....	10
1.5.5.2 PHP 5.2.....	11

1.5.5.3	PostgreSql 8.3.6.....	11
1.5.6	Herramientas de desarrollo.....	12
1.5.6.1	FlashDevelop 3.0	12
1.5.6.2	Flex Builder para GNU/Linux.....	12
1.5.6.3	PHP Development Tools (PDT).....	13
1.5.7	Otras Herramientas para el desarrollo	13
1.5.7.1	OpenOffice Writer 3.0.....	13
1.5.7.2	Gimp 2.6	14
1.5.7.3	Mozilla Firefox 3.0.6	14
1.6	Conclusiones del Capítulo	14
Capítulo 2 Características de la solución propuesta.....		16
2.1	Introducción	16
2.2	Descripción de los procesos vinculados al campo de acción.	16
2.2.1	Flujo actual del proceso	16
2.2.2	Objeto de automatización	17
2.3	Propuesta del sistema	17
2.3.1	Personas relacionadas con el sistema.....	19
2.3.2	Lista de Reserva.....	19
2.3.2.1	Funcionalidades del sistema	19
2.3.2.2	Propiedades del sistema	20
2.4	Conclusiones	21
Capítulo 3 Exploración y Planificación.....		22
3.1	Introducción	22
3.2	Fase de exploración.....	22
3.2.1	Historias de usuario	22
3.3	Fase de planificación	25

3.3.1	Estimación de esfuerzos por historias de usuario	25
3.3.2	Plan de iteraciones	26
3.3.3	Plan de duración de las iteraciones	26
3.4	Conclusiones	27
Capítulo 4 Construcción de la solución propuesta		28
4.1	Introducción	28
4.2	Diseño del sistema.....	28
4.2.1	Diagramas de clases	28
4.2.2	TarjetasCRC.....	30
4.2.3	Diseño de la Base de Datos.....	42
4.3	Fase Implementación.....	45
4.3.1	Tareas de la programación	45
4.3.2	Diagrama de componentes.....	51
4.3.3	Diagrama de despliegue	56
4.3.4	Multiplataforma	57
4.3.5	Estándares de Codificación	57
4.3.6	Arquitectura	59
4.3.7	Patrones	60
4.4	Fase de Prueba	61
4.4.1	Pruebas de Aceptación.....	61
4.5	Conclusiones	65
Conclusiones		66
Recomendaciones		67
Glosario de términos		68
Trabajos citados.....		69
Anexos.....		71

Introducción

La producción de aplicaciones con tecnología multimedia es el proceso que da como resultado un software donde se combinan diferentes tipos de información como texto, imágenes, videos, entre otras, relacionados de manera coherente, todo referente a un tema determinado.

A partir de los cambios que han venido sucediendo en el sistema de educación cubano y la incorporación de nuevas metodologías de enseñanza y el uso intensivo de las tecnologías de la informática y las comunicaciones en las escuelas y centros educacionales ha venido en ascenso la demanda de software educativo, sobre todo el software con tecnología multimedia que por sus características didácticas, metodológicas y educativas sirve de apoyo al proceso docente educativo.

La Universidad de las Ciencias Informáticas es un pilar fundamental en el desarrollo de software con dichas características. Según ha ido en ascenso esta demanda, más recursos, tanto humanos como materiales se han invertido por dicha institución para satisfacer la necesidad existente.

La **situación problemática** viene dada debido a que actualmente en la producción de este tipo de sistemas no se tienen en cuenta, en la mayoría de las ocasiones, la utilización de pautas, estándares y componentes que agilicen el proceso de desarrollo y que posibiliten la entrega a tiempo y con calidad, del producto final al cliente. Además, se malgastan recursos humanos al tener que emplear a los programadores en el montaje de contenidos, tarea que con las herramientas indicadas, se podría realizar por una persona que pudiera no tener conocimientos avanzados sobre el lenguaje de programación.

Otra situación adversa es la utilización de herramientas propietarias, que con las restricciones para adquisición de las licencias, reduce las posibilidades de comercialización debido a la solicitud de algunos clientes de productos desarrollados sobre plataformas libres, además de no ajustarse a las políticas de migración de la UCI.

Teniendo en cuenta lo anteriormente dicho se puede definir como **problema científico** la no existencia de una herramienta libre para la automatización del proceso de gestión de contenidos para aplicaciones con tecnología multimedia, que agilice el desarrollo de los proyectos, disminuya el gasto de recursos humanos y mejore la calidad del producto final.

Como **idea a defender** se tiene que, la creación de una herramienta libre para la automatización del proceso de gestión de contenidos para aplicaciones con tecnología multimedia reducirá el tiempo de producción de las mismas y simplificará el proceso de desarrollo de estas, garantizando un ahorro de

esfuerzo y un producto de mayor calidad.

Este trabajo tiene como **objeto de estudio** el desarrollo de productos con tecnología multimedia y software educativo en la Universidad de las Ciencias Informáticas. Centrándose el **campo de acción** en el desarrollo de herramientas libres para la gestión de los contenidos de este tipo de aplicaciones.

El **objetivo general** del presente trabajo es desarrollar una herramienta libre para la gestión de los contenidos de las aplicaciones con tecnología multimedia, que agilice el desarrollo de los proyectos, disminuya el gasto de recursos humanos y mejore la calidad del producto final.

Para lograr este objetivo se identifican los siguientes **objetivos específicos**:

- Realizar Estado del Arte y Revisión Bibliográfica.
- Definir una arquitectura para la organización de los contenidos en aplicaciones con tecnología multimedia.
- Realizar Análisis, Diseño e Implementación de la Aplicación.
- Integrar a la Micro-Arquitectura Primavera.
- Elaborar la documentación de cómo extender la aplicación a otras plataformas.

Para lograr los objetivos específicos contamos con las siguientes **tareas de la investigación**:

- Realizar una Revisión Bibliográfica.
- Realizar el Estado del Arte
- Selección y fundamentación de una metodología de desarrollo.
- Estudiar los patrones existentes para la organización de los contenidos en aplicaciones con tecnologías multimedia.
- Realizar el levantamiento de Requisitos.
- Realizar el diseño de la aplicación.
- Realizar un estudio de Patrones de Diseño a aplicar.
- Realizar la elaboración de la Interfaz de Usuario.
- Estudiar los estándares existentes para el intercambio de información entre aplicaciones.
- Aplicar el estándar seleccionado.
- Seleccionar la herramienta más adecuada para generar la documentación de la aplicación.

- Generar la documentación.

Capítulo 1

Fundamentación teórica

1.1 Introducción

El uso intensivo de las tecnologías de la informática y las telecomunicaciones (TIC) en los centros educacionales cubanos ha provocado una gran demanda de productos con tecnología multimedia. Cada vez son más los esfuerzos que se realizan tanto en la Universidad de Ciencias Informáticas como fuera de esta para desarrollar este tipo de aplicaciones con menos desperdicio de recursos, a un menor costo, en menos tiempo y con una mayor calidad. Es por ello que surge la necesidad de contar con una herramienta que permita darle solución a estas necesidades. En este capítulo se hará un análisis detallado acerca de las herramientas existentes que pudieran darle cumplimiento a dichas necesidades, teniendo en cuenta las tendencias tecnológicas basadas en alternativas libres para el desarrollo de aplicaciones con tecnología multimedia.

1.2 Descripción General del Objeto de Estudio

El proceso de desarrollo de software educativo con tecnología multimedia en la Universidad de Ciencias Informáticas se lleva a cabo por parte de estudiantes, profesores y especialistas con sólidos conocimientos en el tema, y estos a su vez se encuentran vinculados a proyectos productivos, que tienen como meta la producción de un software que satisfaga las necesidades de un cliente.

Para la realización de estos productos se usan herramientas privativas (por ejemplo Adobe Flash Professional 8, perteneciente a Adobe Systems Incorporated), que están sujetas a restricciones legales que obstaculizan el comercio con otros países, reduciendo los ingresos y en ocasiones se priva el derecho de autor, al no poder decirse que determinado software fue realizado en Cuba porque es un país embargado y no tiene derecho a adquirir este tipo de herramientas. Además no existen herramientas que faciliten la reutilización del código y los componentes, y definan un estándar para el desarrollo de este tipo de aplicaciones, lo que trae consigo grandes periodos de tiempo para la realización de las mismas, aumentando los costos de producción, el desperdicio de capital humano y de recursos que, de alguna manera, interfieren en la calidad del producto final.

1.3 Identificación de la audiencia

La Herramienta a desarrollar estará dirigida a todas aquellas personas o desarrolladores que estén implicados en la creación de aplicaciones con tecnología multimedia, tanto en la UCI como fuera de esta, a lo largo de todo el país. Además, podría extenderse a nivel internacional, debido a las facilidades que brindará durante el proceso de desarrollo de productos multimedia, permitiendo que este trabajo se reduzca a horas o días, en dependencia de su magnitud, y contribuyendo a disminuir el margen de errores.

1.4 Análisis de soluciones existentes

El análisis realizado de las soluciones existentes arrojó como resultado dos tipologías de aplicaciones de las cuales se tomó un representante de cada una y se desglosaron sus características. Todas estas herramientas tienen en común que están dirigidas a un grupo específico de aplicaciones y con características semejantes como son los software educativos destinados a la creación de cursos, ejercicios y contenido digital con el propósito de asistir el proceso docente educativo, pero ninguno va más allá de eso, por lo cual no satisfacen las necesidades que tiene la universidad con respecto a este tipo de herramientas.

1.4.1 MALTED 3:

Es una herramienta de autor para la creación y ejecución de unidades didácticas multimedia e interactivas para ser utilizadas en aulas con el equipamiento tecnológico adecuado. Ha sido desarrollada en particular para la enseñanza de idiomas, pero su uso se puede extender a otras materias del currículo escolar, su uso está basado en plantillas que mejoran y facilitan la creación de contenidos. Al ser una herramienta con tecnología web la convierte en multiplataforma. Su principal desventaja es que solo se puede generar productos para las clases de idiomas, lo que reduce su campo de acción a un reducido grupo de aplicaciones.

El sistema cuenta con 3 subsistemas o entornos de trabajo denominados:

Editor MALTED (DVE) es un entorno de desarrollo, que genera el código XML de los ejercicios y las unidades de trabajo diseñadas, cuenta con numerosas plantillas que permiten generar un sinnúmero de ejercicios ya predeterminados, además permite la incorporación de medias (textos, imágenes, sonidos), y por otro lado también permite crear las unidades de trabajo que nos son más que el conjunto de ejercicios dotados de una navegación lógica desde el punto de vista organizacional y educativo.

Navegador MALTED este entorno de desarrollo permite ver y completar los ejercicios creados con

anterioridad, además de poseer su propio visor también puede ser visto desde un navegador web. Depurador MALTED este entorno está encaminado a los diseñadores con conocimientos de programación ya que provee un entorno mixto compuesto por un editor de código XML y el visor del programa lo que permite ver directamente las modificaciones realizadas.

El sistema puede enlazarse con 2 complementos uno denominado dispositivo de seguimiento el cual lleva un control del estudiante y el otro una base de recursos lo que posibilita que los diseñadores accedan a una base de datos externa de la que podrán obtener recursos multimedia, permitiendo así la colaboración sobre este entorno. (Desconocido, 09)

1.4.2 Constructor Atenex 3:

Es un poderoso instrumento de creación de contenidos educativos digitales; se trata de una herramienta de autor que se puede instalar en modo local o en un servidor, y tiene versiones para sistemas GNU/Linux y Windows. Por lo tanto, se puede usar en cualquier ordenador personal o en un servidor para ser usado conjuntamente por diferentes usuarios.

Consta de un variado plantillero que incluye 47 modelos diferentes de actividades (crucigrama, sopa de letras, emparejamientos diversos, dictados de palabras o frases, completar u ordenar frases o textos, pirámide, ahorcado, rompecabezas, puzzles, tangram, reconstrucción de figuras, compositor musical,...), además de un conjunto de aplicaciones sumamente avanzadas (calculadoras básica y científica, fórmulas y gráficas matemáticas, y un completo laboratorio virtual de física). Permite además la inclusión de zonas interactivas, documentos de todo tipo, crear presentaciones de imágenes o generar enlaces a materiales externos para el diseño de estructuras de navegación tan complejas como desee el profesor-autor, de modo que acepta tanto la utilización del plantillero como la creación de páginas web.

Posee una arquitectura que facilita la portabilidad de los ejercicios generados a través de él, debido a que toda las soluciones desarrolladas con esta herramienta están basadas en una combinación de XML y SWFs lo cual hace posibles que sean vistas en casi cualquier ordenador del mundo a través de un navegador web ordinario (Educarex, 09)

1.5 Tendencias y Tecnologías

En la investigación realizada sale a relucir que las herramientas “lo que ves es lo que obtienes” (del Inglés “what you see is what you get”, por sus siglas en inglés WYSIWYG) son las que más aceptación tienen entre los usuarios, ya que permiten, según van creando el contenido interactuar con este e ir previsualizando como va a quedar el producto final. Estas herramientas simplifican mucho el trabajo ya

que de una manera gráfica se realizan tareas que de otra manera habría que realizarlas interactuando con el código.

1.5.1 Metodología de desarrollo

1.5.1.1 Extreme Programming (XP)

Recientemente han surgido un nuevo grupo de metodologías en contraposición a las llamadas robustas o también conocidas como “burocráticas” las denominadas metodologías ágiles las cuales se guían por un manifiesto denominado Manifiesto Ágil el cual sigue estos principios:

- “Nuestra prioridad más alta es satisfacer al cliente, mediante tempranas y continuas entregas del software”.
- “Entregar frecuentemente un software ejecutable, en un período de tiempo desde dos semanas hasta dos meses preferiblemente una escala de tiempo pequeña.”
- “El cliente y los desarrolladores deben trabajar en conjunto diariamente a lo largo del tiempo de vida del proyecto.”
- “Software ejecutable es la primera medida de progreso.”

Esta metodología se basa en cuatro valores fundamentales Comunicación, Retroalimentación, Simplicidad y Coraje, sobre la base de estos valores construye doce prácticas que se deben cumplir en los proyectos desarrollados con la metodología XP.

- El juego de la planificación. Hay una comunicación frecuente entre el cliente y los programadores.
- Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas.
- Metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo.
- Diseño simple. Se debe diseñar la solución más simple que pueda funcionar.
- Pruebas. La producción de código está dirigida por las pruebas unitarias.
- Refactorización (Refactoring). Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores.
- Propiedad colectiva del código. Cualquier programador puede cambiar cualquier parte del

código en cualquier momento.

- Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista.
- Se debe trabajar un máximo de 40 horas por semana.
- Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo.
- XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. (Fowler, 2003; Don, 2006)

Una de estas prácticas consideradas por muchos la más importante y revolucionaria consiste en la realización de pruebas constantes, estableciendo así la comprobación como el fundamento del proyecto. La principal suposición de esta metodología es la posibilidad de disminuir la curva del costo del cambio a lo largo del proyecto.

1.5.1.2 Rational Unified Process

Es una de las metodologías consideradas como robustas. Tiene 3 características fundamentales: dirigido por Casos de Uso; iterativo incremental y centrado en la arquitectura. En ella se definen 4 fases principales las cuales son: Inicio; Elaboración; Construcción y Transición, además posee nueve flujos de trabajos, seis de Ingeniería Modelado del Negocio: Requerimientos; Análisis y Diseño; Implementación; Prueba y Despliegue y tres de apoyo: Gestión de la Configuración; Gestión de Proyecto y Ambiente. Su principal desventaja consiste en la generación excesiva de documentación, la cual a veces, en proyectos pequeños con equipos de desarrollo pequeños resulta innecesaria y lejos de ayudar en el desarrollo de este, lo entorpece y demora.

De las dos metodologías se escoge XP, debido a que esta es la que mejor se ajusta y aprovecha las condiciones de trabajo del equipo de desarrollo, cliente in situ, requisitos cambiantes, equipo de trabajo pequeño, tiempo de desarrollo relativamente corto por lo que es necesario centrarse en la implementación y no en el modelado del sistema.

1.5.2 Lenguaje Unificado de Modelado 2.0

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language), proviene de la unión de 3 técnicas diferentes las cuales son: el método de Grady Booch, la Técnica de Modelado de Objetos de James Rumbaugh y "Objetory", de Ivar Jacobson. En 1994 Booch, Rumbaugh y Jacobson dieron forma a la primera versión del UML y en 1997 fue aceptado por la OMG, fecha en la que fue lanzada la versión v1.1 del UML. Desde entonces, UML atravesó varias revisiones y

refinamientos hasta llegar a la versión actual: UML 2.0.

El UML está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, incluye aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Este lenguaje sirve para especificar y no para describir métodos o procesos.

En UML 2.0 hay trece tipos diferentes de diagramas:

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado: (Valerio, 2005)

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x)
- Diagrama de tiempos (UML 2.0)
- Diagrama de vista de interacción (UML 2.0)

Con el uso de este lenguaje, el modelado del software se realiza de una manera más sencilla y entendible para el equipo de desarrollo. (Valerio, 2005)

1.5.3 Herramientas CASE

Ingeniería de Software Asistida por Ordenador, (**C**omputer **A**ided **S**oftware **E**ngineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden

ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación y detección de errores.

1.5.4 Visual Paradigm for UML 6.4

Puede generar y realizar ingeniería inversa a más de 10 lenguajes entre los que se encuentran Java, C++, C#, ActionScript, entre otros; además está optimizado para proyectos de gran escala a través de un acercamiento del paradigma orientado a objetos, sus requerimientos son bastante aceptables, además apoya los más recientes estándares de UML. Entre sus principales características se encuentran:

- Soporte para la versión UML 2.1.
- Documento Modelado colaborativo con CVS y Subversión.
- Interoperabilidad con modelos UML2.
- Código a modelo.
- Código a diagrama.
- Ingeniería inversa ActionScript.
- Generación de código ActionScript.
- Modelo a código.
- Diagrama a código.
- Diagramas de flujo de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas.
- Reorganización de las figuras y conectores de los diagramas UML. (Visual Paradigm, 08)

1.5.5 Lenguajes de Codificación

1.5.5.1 ActionScript 3.0

En la actualidad un sinnúmero de aplicaciones web basan su funcionamiento o parte de este en la plataforma flash y en su lenguaje de programación ActionScript, recientemente Adobe liberó una nueva versión de su lenguaje el cual es más poderoso y ha significado una importante evolución de las potencialidades del Flash Player, el ActionScript 3.0, éste provee un alto rendimiento y las facilidades para crear aplicaciones altamente complejas, manejar juegos de datos más grandes y la reutilización de código, lo que permite incrementar la productividad y el rendimiento de los desarrolladores de

aplicaciones para Flash Player.

El código ActionScript es ejecutado por la Máquina Virtual de ActionScript que se encuentra embebida en el Flash Player, para esta nueva versión, también se liberó una nueva Máquina Virtual para ActionScript, la cual excede dramáticamente en rendimiento a su anterior versión, se dice que puede ser 10 veces más rápida aún así ambas conviven en el Flash Player por problemas de compatibilidad con aplicaciones anteriores.

Esta nueva versión también cumple con los nuevos estándares de la especificación 3 del lenguaje ECMAScript, y contiene funcionalidades de la especificación 4 de ECMAScript. Cumplir con estos estándares le permite una mejor integración con XML haciendo mucho más sencilla y eficaz la manipulación de dichos datos. (Grossman, et al., 2006)

1.5.5.2 PHP 5.2

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa ***PHP Hypertext Pre-processor*** (inicialmente PHP Tools, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Es un lenguaje de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (PHP, 07)

1.5.5.3 PostgreSql 8.3.6

Es el motor de base de datos libre más avanzado hasta estos momentos, al que se le puede acceder a su código fuente. Es usado para manejar grandes cantidades de información y está basado en el modelo relacional, aunque incorpora conceptos del modelado orientado a objeto. Se pueden definir consultas anidadas, vistas, crear funciones por el usuario, no sólo en el lenguaje natural SQL, sino en

varios más, entre ellos C, lenguaje nativo PostgreSQL, Perl, PHP y Java. Es multiplataforma, soporta múltiples transacciones, integridad de datos, presenta una estabilidad muy alta, gran seguridad de los datos, soporta la réplica y procedimientos almacenados. Es un gestor de base de datos robusto, y con grandes funcionalidades. Presenta como desventaja que por sus grandes potencialidades consume muchos recursos y carga el sistema. (Drake, 2009)

1.5.6 Herramientas de desarrollo

1.5.6.1 FlashDevelop 3.0

FlashDevelop es un entorno de desarrollo integrado (IDE, por sus siglas en Inglés Integrated Development Environment) para aplicaciones basadas en ActionScript, MXML y HAXE muy potente y con muchas facilidades para los programadores entre las que se encuentran las siguientes:

- Soporta 3 lenguajes: Actionscript 2, Actionscript 3 & MXML y HAXE.
- Brillante completamiento de código de Actionscript y generadores de código.
- Completamiento de código para XML, MXML and HTML
- Navegación por el código (saltar a la declaración).
- Fácil integración con Flash y compiladores a línea de comando.
- Integrado gestor de proyectos para controlar propiedades y archivos.

Este entorno de desarrollo, además de las características mencionadas también es libre y open source, pero tiene como limitante que solo está disponible para Windows y requiere del Microsoft Windows .NET 2.0 framework, lo cual nos obliga a pagar las licencias de este software. (Palu, 2008)

1.5.6.2 Flex Builder para GNU/Linux

Es un IDE destinado al desarrollo de aplicaciones basadas en ActionScript 3.0. Este software es desarrollado por la empresa Adobe y aunque es un software privativo cuenta con una versión para los sistemas operativos Gnu/Linux, que por estar en versión de prueba es gratis, recientemente esta empresa liberó, bajo licencia Mozilla Public License (MPL, por sus siglas en inglés) los compiladores de ActionScript y de MXML, las librerías de clases del núcleo y de los componentes, permitiendo usar, extender y distribuir estas partes del software sin ningún problema legal.

Este IDE posee características que facilitan mucho la creación de aplicaciones, como son:

Incluye editor de código ActionScript, MXML y CSS, los cuales cuentan con coloreo de sintaxis, completamiento de código, contracción de código y posibilidades de depurar el código.

Permite cambiar de manera automática todas las referencias a una clase en caso de tener que

modificar su nombre, esta característica se denomina refactorización de código.

Posee herramientas para monitorear el rendimiento de la aplicación y su comportamiento en cuanto a consumo de memoria RAM y uso del procesador.

Gestor de proyectos que permite controlar los archivos que se van creando durante la realización de la solución.

Estas son las características que propician que esta sea la herramienta escogida para la realización de la solución propuesta en este trabajo. (Adobe, 08)

1.5.6.3 PHP Development Tools (PDT)

Como su nombre lo indica, es una herramienta de desarrollo de código abierto que ofrece todas las capacidades básicas de edición de código necesario para crear aplicaciones con PHP, basado en el popular proyecto Eclipse y usando Plataforma Web. PDT brinda las siguientes facilidades:

- Editar el código PHP.
- Completamiento de código.
- Uso de las plantillas de código de PHP.
- Controlar el formato del código.
- Navegar a través de los elementos de PHP (Zend, 2008)

1.5.7 Otras Herramientas para el desarrollo

1.5.7.1 OpenOffice Writer 3.0

La suite ofimática que se utilizará para la realización del documento de tesis, así como para la documentación pertinente a la investigación y desarrollo de la aplicación será OpenOffice.org por contar con las características que se exponen a continuación:

OpenOffice.org es software libre y código abierto de distribución gratuita que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para muchas plataformas. OpenOffice está pensado para ser altamente compatible con Microsoft Office. Soporta el estándar ISO OpenDocument con lo que es fácil el intercambio de documentos con muchos otros programas, y puede ser utilizado sin costo alguno.

OpenOffice.org posee como base inicial a StarOffice, la cual fue desarrollada por StarDivision y adquirida por Sun Microsystems en agosto de 1999. El código fuente de la herramienta fue liberado en julio de 2000, proporcionando una alternativa abierta, gratuita y de alta calidad. (Sun Microsystem, 07)

1.5.7.2 Gimp 2.6

GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la Licencia pública general de GNU.

La primera versión de GIMP se desarrolló inicialmente en sistemas Unix y fue pensada especialmente para GNU/Linux. Existen versiones totalmente funcionales para Windows, para Mac OS X, y se incluye en muchas Distribuciones Linux (como Ubuntu). También se ha portado a otros sistemas operativos, haciéndolo el programa de manipulación de gráficos disponible en más sistemas operativos. Se le puede considerar como una alternativa firme, potente y rápida a Photoshop para muchos usos, aunque no se ha desarrollado como un clon de él y posee una interfaz bastante diferente.

Existen también versiones portátiles de GIMP que pueden ser transportadas y usadas directamente desde una memoria USB sin necesidad de instalarse en el ordenador, tanto bajo Windows como bajo Mac OS X. (Steiner, et al., 08)

1.5.7.3 Mozilla Firefox 3.0.6

Mozilla Firefox es un navegador de Internet libre y de código abierto, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos. Para visualizar páginas web, Firefox usa el motor de renderizado Gecko, incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además, se pueden añadir funciones a través de complementos desarrolladas por terceros. Este programa es multiplataforma. (Mozilla Foundation, 09)

1.6 Conclusiones del Capítulo

En este capítulo se ha realizado un estudio minucioso de las soluciones existentes teniendo en cuenta tendencias tecnológicas actuales, llegando a la conclusión que las soluciones de software libre existentes para la gestión de contenidos para aplicaciones con tecnología multimedia no cubren completamente las necesidades. Para el desarrollo de la nueva solución se usará como herramienta case Visual Paradigm, siguiendo la metodología Extreme Programming, debido que es la que más se ajusta al equipo de desarrollo y a las características en las que se está desarrollando el software, como son los requisitos cambiantes y el hecho de que el cliente trabaje en conjunto con el equipo de desarrollo. Como herramientas para la implementación PHP Development Tools y Flex Builder, para trabajar con los lenguajes de programación PHP y Actionscript 3.0 respectivamente, usando como

Gestor de bases de datos PostgreSQL

Capítulo 2

Características de la solución propuesta

2.1 Introducción

Este capítulo tiene como principal objetivo ofrecer una valoración general de las características que tendrá el sistema a desarrollar, centrándose en la situación problemática que dio origen al mismo. También se abordarán las funcionalidades que con el sistema se pretenden automatizar; además, se presentará una propuesta del software a implementar y se especificarán detalladamente los requerimientos funcionales y no funcionales, con los que debe cumplir el sistema una vez implementado.

2.2 Descripción de los procesos vinculados al campo de acción.

Actualmente en la Universidad de las Ciencias Informáticas el proceso de gestión de contenidos para aplicaciones con tecnologías multimedia, no está bien definido, lo que propicia que no se aplican pautas, estándares, tampoco se cuenta con una herramienta que automatice este proceso, todo esto trae consigo atraso en la entrega del producto final, trayendo aparejado inconformidades por parte del cliente, situación que es muy adversa para una empresa, por lo que se hace necesario revertir este escenario.

2.2.1 Flujo actual del proceso

El desarrollo de este proceso se realiza de la siguiente manera: una vez que se firma el contrato del software que se va a desarrollar, empiezan a generarse los contenidos, que muchos son brindados por los clientes y otros son generados en la universidad, ya sea por los especialistas en diseño, o cualquier otro encargado de esta tarea. Estos contenidos son entregados al jefe del proyecto, este después se encarga de hacérselos llegar al equipo de desarrollo, los cuales deben preparar dichos contenidos, para poder ser utilizados en el producto, es decir, deben realizar los cambios de formato, tamaño y estructuración requeridos.

2.2.2 Objeto de automatización

Del análisis realizado a la actual actividad, definimos varios procesos que deben ser automatizados, debido a que su ejecución de forma manual, aparte de ser una tarea tediosa y expuesta a errores, tiene que ser llevada a cabo por los desarrolladores o por personal que tenga conocimientos de las técnicas de programación y utilización del lenguaje XML, lo cual consume mucho tiempo del equipo de desarrollo.

Gran parte de estos procesos poseen una gran importancia e influyen directamente en el tiempo de realización del producto final y en la calidad del mismo. De los procesos analizados se definieron como automatizables: la gestión, modificación, vinculación e incorporación de los contenidos, tarea que se realizará mediante una aplicación Web, la cual dará la posibilidad de gestionar, modificar, incorporar y vincular todos los contenidos de una aplicación con tecnología multimedia, sin la intervención de personal con conocimientos informáticos, permitirá una vez terminado todo el proceso, exportar el proyecto bajo estándares de organización y de formatos de media; las imágenes se exportarán en formato PNG debido a que este soporta colores de hasta 32 bit de profundidad y a la vez transparencias, los sonidos, en MP3, este es el único formato comprimido que acepta la plataforma Adobe Flash, los videos se exportarán en FLV por su alta integración con Adobe Flash y fácil manejo de streaming, los textos en XML, por las posibilidades de estructuración y organización del contenido que brinda y las facilidades que brinda el lenguaje AS3 para el manejo de estos.

2.3 Propuesta del sistema

El sistema contará con una interfaz web como forma de presentación al usuario, la cual tendrá una página de bienvenida que brindará información general sobre el sistema y desde la cual los usuarios se podrán autenticar, en los proyectos que estén creados en el sitio. Una vez que el usuario esté autenticado podrá tener acceso a las funcionalidades del sistema, que se correspondan con su rol, las cuales estarán agrupadas en la parte superior en una barra de menús, contendrá de manera general las opciones de Proyecto, Usuarios, Contenidos, Ayuda, que a su vez tendrán dentro otras funcionalidades como son:

- Proyecto
 - Crear Nuevo Proyecto (Permitirá crear un proyecto nuevo, a esta opción solo tendrá acceso el administrador del sistema).
 - Modificar Proyecto (Permitirá modificar un proyecto, a esta opción solo tendrá acceso el administrador del sistema).

- Eliminar Proyecto (Permitirá eliminar un proyecto, a esta opción solo tendrá acceso el administrador del sistema).
- Cerrar Proyecto (Permitirá cerrar un proyecto en el cual se esté trabajando, a esta opción tendrán acceso todos los usuarios del sistema).
- Salir (Permitirá salir de la aplicación).
- Usuarios
 - Agregar Nuevo Usuario (Permitirá agregar un usuario nuevo al proyecto, a esta opción solo tendrá acceso el Jefe de Proyecto).
 - Modificar Datos de Usuario (Permitirá modificar la información de un usuario autenticado en el sistema, a esta opción tendrán acceso todos los usuarios del sistema).
 - Eliminar Usuario (Permitirá eliminar un usuario perteneciente a ese proyecto, a esta opción solo tendrá acceso el Jefe de Proyecto).
 - Cambiar de Usuario (Permitirá cambiar de usuario, a esta opción tendrán acceso todos los usuarios del sistema).
- Contenidos
 - Crear Nuevo Contenido (Permitirá crear un contenido nuevo en el proyecto, a esta opción tendrán acceso el Jefe de Proyecto y los desarrolladores).
 - Modificar Contenido (Permitirá modificar un contenido existente en el proyecto, a esta opción tendrán acceso el Jefe de Proyecto y los desarrolladores).
 - Eliminar Contenido (Permitirá eliminar un contenido del proyecto, a esta opción tendrán acceso el Jefe de Proyecto y los desarrolladores).
- Ayuda
 - Ayuda Online
 - Acerca de

El sistema contará además con un área de trabajo en la cual se desarrollarán todas las funcionalidades, esta tendrá la característica de ser WYSIWYG, lo cual, dota a la aplicación de una gran usabilidad y economía de tiempo, ya que, no tiene que estar exportando el proyecto y chequeando como se verá este, porque estará en tiempo real verificando como resultará su trabajo una vez que lo publique.

En el área inferior, la aplicación contará con una barra de estado en la cual se brindará información adicional al usuario sobre el sistema y proyecto en el cual está trabajando y otras como posibles notificaciones a realizar.

El sistema permitirá al usuario exportar una salva del proyecto en cualquier momento que lo solicite, el

cual generará la estructura de carpeta, contenidos, y una salva de la base de datos y configuraciones del proyecto.

2.3.1 Personas relacionadas con el sistema

Se definen como personas relacionadas al sistema, aquellas que interactúan con este obteniendo un resultado de su interacción con al menos uno de sus procesos, además de todas aquellas que están de una forma u otra vinculadas a la realización de los mismos, ya que participan, pero no obtienen ningún resultado.

Tabla 2.1 Personas relacionadas con el sistema

Personas relacionadas con el sistema.	Justificación
Usuario	Es la persona que accede al sistema pero aún no se ha autenticado, por lo que no goza de ningún privilegio en la aplicación.
Desarrollador	Es la persona que se encuentra autenticada en el sitio cumpliendo este rol. Dicha persona solo tendrá acceso a los proyectos en los que figure como desarrollador, podrá agregar, modificar y eliminar contenido.
Administrador	Es la persona que se encarga del mantenimiento del sistema y el registro de usuarios, tendrá acceso a todos los proyectos creados pero no podrá realizar cambios en ellos.

2.3.2 Lista de Reserva

A partir del análisis de los conceptos asociados al objeto de estudio y los objetivos planteados al inicio de esta investigación, se puede realizar un análisis sobre las posibles prestaciones que el sistema deber tener para cumplirlos, las cuales se definirán como requisitos funcionales de la aplicación a implementar. En ellos están incluidas las acciones ejecutadas por los usuarios y las propias del sistema. El sistema debe ser capaz de:

2.3.2.1 Funcionalidades del sistema

Tabla 2.2 Funcionalidades del sistema

Número	Descripción	Prioridad
1	Crear cuenta en el sistema	Alta
2	Ver perfil de usuario	Alta
3	Modificar cuenta del usuario	Alta
4	Eliminar cuenta de usuario	Alta
5	Autenticar	Alta
6	Publicar contenidos	Alta
7	Crear nuevo contenido	Alta
8	Mostrar contenidos	Alta
9	Modificar contenidos	Alta
10	Eliminar contenidos	Alta
11	Crear nuevo proyecto	Alta
12	Eliminar proyecto	Alta
13	Exportar proyecto	Alta

2.3.2.2 Propiedades del sistema

Una vez determinado las funcionalidades del sistema, se deben analizar los requerimientos no funcionales, que no son más que aquellos que permiten darle usabilidad, atractivo, confiabilidad y reusabilidad a la aplicación, permitiendo así, que sea un software de calidad.

Diseño e implementación

- 3.1 Aplicación Web escrita sobre el lenguaje de programación PHP 5.2.
- 3.2 Usar el gestor de Base de Datos PostgreSQL 8.3.6.
- 3.3 Utilizar como servidor web Apache 2.4.
- 3.4 Utilizar como metodología de desarrollo XP.
- 3.5 El sistema debe ser multiplataforma.

Apariencia o interfaz externa

- 3.6 Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- 3.7 Diseño perfectamente encuadrado para resoluciones de 1024X768, pero preparado para verse en otras resoluciones.
- 3.8 El diseño garantizará la selección de un esquema de colores atractivo pero que no canse,

conjugando equilibrio entre los mismos y un contraste suficiente pero no agresivo.

Tabla 2.3 Colores de la aplicación

Colores	Codificación hexadecimal
Naranja	#FF3300
Azul	#0489D0

3.9 Verificación sobre acciones irreversibles (eliminaciones).

Ayuda y documentación en línea.

3.10 El sistema contará con una ayuda online y documentación que permita conocer como extender la plataforma y hacer uso de ella.

2.4 Conclusiones

Con este capítulo se ha comenzado el desarrollo de la propuesta de solución que se desea implementar, después de realizar el análisis del flujo actual del proceso y de definir las posibles automatizaciones de este con el fin de agilizar, asistir y guiarlo, definiéndose así las funcionalidades del sistema lo cual constituye la base de todo proceso de desarrollo de software.

Capítulo 3

Exploración y Planificación

3.1 Introducción

En este capítulo se hace alusión a las fases de exploración y planificación, propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone. Se exponen además los artefactos generados durante el transcurso de las mismas.

3.2 Fase de exploración

Esta es la primera fase de la metodología de desarrollo Programación Extrema (o como se conoce en inglés Extreme Programming), en la cual se definen las historias de usuario, lo que posteriormente servirá para realizar una planificación del ciclo de vida del proyecto.

3.2.1 Historias de usuario

En la metodología XP las historias de usuario las escriben los propios clientes, tal y como ven ellos las necesidades del sistema. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Durante este proceso se identifican 5 historias de usuarios las cuales se detallan a continuación.

Historia de Usuario	
No.: 1	Nombre: Gestionar cuenta de usuarios
Usuario: Administrador	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realizarán las acciones de crear, modificar y eliminar las cuentas de usuario, así como mostrar la información de cada uno y permitir que se autenticuen en el sistema, el que debe ser capaz de brindarle, la posibilidad de ingresar los datos necesarios para cada acción; en el caso de eliminación se debe comprobar que el usuario que intente realizar la acción, tenga los privilegios necesarios (administrador).	
Información adicional (Observaciones): Se hace referencia a los requisitos R1, R2, R3, R4,R5	

Historia de Usuario	
No.: 2	Nombre: Importar contenidos
Usuario: Desarrollador, Administrador	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Medio
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Se realizarán las acciones de publicar, mostrar, modificar y eliminar contenidos, el sistema tiene que permitirle al usuario incorporar contenidos nuevos, que posteriormente se utilizarán en la aplicación, como imágenes, música, textos, videos, y además debe permitir ingresar información de cada uno de estos contenidos.	
Información adicional (Observaciones): Se hace referencia a los requisitos R6, R8, R9, R10	

Historia de Usuario	
No.: 3	Nombre: Crear nuevo contenido
Usuario: Desarrollador, Administrador	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alto
Puntos de Estimación: 3	Iteración Asignada: 2
Descripción: El sistema debe ser capaz de proveerle al usuario las herramientas y medios necesarios para la creación de contenidos nuevos como menús, temas, palabras calientes, galerías de imágenes, visores de videos, pop-up, reproductores de música y las respectivas configuraciones de cada contenido.	
Información adicional (Observaciones): Se hace referencia a los requisitos R7	

Historia de Usuario	
No.: 4	Nombre: Gestionar Proyectos
Usuario: Administrador	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: El sistema debe ser capaz de: Permitir la creación y eliminación de proyectos. Entrar los datos referentes a cada proyecto. Eliminar dichos datos en caso que sea necesario.	
Información adicional (Observaciones): Se hace referencia a los requisitos R11, R12	

Historia de Usuario	
No.: 5	Nombre: Exportar proyectos
Usuario: Desarrollador, Administrador	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Puntos de Estimación: 2	Iteración Asignada: 3
Descripción: El sistema tiene que ser capaz de crear una copia del proyecto deseado, donde el usuario solicite, incluyendo todos los contenidos de este, así como sus configuraciones.	
Información adicional (Observaciones): Se hace referencia a los requisitos R13	

3.3 Fase de planificación

En la fase de planificación se realiza el Plan de Entregas (Release plan, como se conoce en inglés) que no es más que la planificación que se debe realizar sobre la entrega de pequeños ejecutables para la retroalimentación del cliente. Esto se lleva a cabo basándose en la estimación del esfuerzo que costará implementar cada historia de usuario, para eso se utilizan los puntos de estimación los cuales se toman como una semana de trabajo ideal (5 días laborables, 8 horas de trabajo por día).

3.3.1 Estimación de esfuerzos por historias de usuario

A continuación se muestra un resumen de la estimación de esfuerzo de implementación para cada historia de usuario, basado en los puntos de estimación.

Tabla 3.1 Estimación de esfuerzos por HU

Historia de Usuario	Puntos de Estimación
Gestionar cuenta de usuarios	1
Importar contenidos	2
Crear nuevo contenido	3
Gestionar Proyectos	1
Exportar proyectos	2

3.3.2 Plan de iteraciones

Posteriormente a describir e identificar las historias de usuarios y haber realizado la estimación de esfuerzo, se puede proseguir con la planificación de la etapa de implementación del sistema, que no es más que precisar qué historias de usuario se van a implementar en cada iteración y las posibles fechas de liberación de dichas versiones, por lo que no queda más que detallar los ciclos de desarrollo del sistema, como sigue a continuación:

Iteración 1

En esta iteración se pretende implementar las historias de usuario que más importancia tienen para el cliente, las cuales son: No1. Gestionar cuenta de usuarios y No2. Importar contenidos, estas le darán cumplimiento a las funcionalidades: crear, modificar, eliminar cuentas de usuarios, autenticar en el sistema, ver perfil de usuarios, publicar, mostrar, modificar y eliminar contenidos. Una vez que se terminen de implementar todas las funcionalidades de estas iteraciones, se procederá a liberar un ejecutable con la finalidad de retroalimentarnos del cliente y solventar posibles errores.

Iteración 2

En esta iteración se programarán las historias de usuario que le siguen en importancia a las tareas críticas que fueron implementadas en la primera iteración, como es la historia de usuario No3. Crear nuevo contenido, esta dota al sistema de sus características más relevantes como son las de proveer al desarrollador, de herramientas y los medios necesarios para la creación de contenidos nuevos. Al finalizar esta iteración se liberará un ejecutable con la finalidad de retroalimentarnos del cliente y solventar posibles errores.

Iteración 3

En esta iteración se implementarán las historias de usuario restantes, las cuales están diseñadas con el objetivo de lograr la organización de la información y exportación de los proyectos, tareas que son un poco engorrosas de realizar manualmente; estas están contenidas en las historias de usuarios: No4. Gestionar Proyectos y No5. Exportar Proyectos, las cuales permiten crear, eliminar y exportar un proyecto. Una vez que se termine de implementar estas funcionalidades ya tendremos la versión 1.0 de la aplicación, disponiéndola al servicio de la comunidad, con el objetivo de ponerla a prueba para evaluar su funcionamiento, rendimiento y recuperación ante posibles errores.

3.3.3 Plan de duración de las iteraciones

Este plan sirve para plasmar de una forma resumida, en qué orden y qué tiempo nos tomará la implementación de cada una de las historias de usuario con las que cuenta nuestra aplicación.

Tabla 3.2 Plan de duración de las iteraciones (Equipo de desarrollo)

Iteraciones	Orden de las Historias de usuario a implementar	Duración total de las iteraciones
Iteración 1	Gestionar cuenta de usuarios Importar contenidos	3 semanas(La entrega se realizará en la segunda semana de marzo)
Iteración 2	Crear nuevo contenido	3 semanas(La entrega se realizará en la primera semana de abril)
Iteración 3	Gestionar Proyectos Exportar proyectos	3 semanas(La entrega se realizará en la última semana de abril)

3.4 Conclusiones

En este capítulo se trató todo lo concerniente a la fase de exploración y planificación del sistema y se documentó todos los artefactos generados en el transcurso de esta. Como quedó expresado en el capítulo, se adoptó un desarrollo por iteraciones y programación en parejas tal y como lo propone la metodología XP.

Capítulo 4

Construcción de la solución propuesta

4.1 Introducción

En este capítulo se describe detalladamente el proceso de desarrollo e implementación de la aplicación, teniendo en cuenta la metodología XP que plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éstos. También se mostraran los diagramas correspondientes a esta etapa, las tareas generadas por cada historia de usuario y las pruebas de aceptación efectuadas sobre el sistema para demostrar que funciona según lo esperado.

4.2 Diseño del sistema

La metodología de desarrollo XP establece prácticas especializadas, que inciden directamente en la realización y elaboración del diseño de un software, sin embargo no requiere que la representación del sistema sea mediante diagramas de clases basados en UML, sino que pueden emplearse indistintamente sencillos esquemas descritos en pizarras u otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). No obstante el empleo de los diagramas UML pueden ser utilizados siempre y cuando contribuyan en el mejoramiento de la comunicación del equipo de desarrollo, no sean muy extensos y no requieran de mucho tiempo para su creación. (Don, 2006)

4.2.1 Diagrama de clase

4.2.2 TarjetasCRC

Clase Responsabilidad Colaborador (CRC) son una herramienta de reflexión en el diseño de software orientado a objetos. Fueron propuestas por Ward Cunningham y Kent Beck. Se utilizan normalmente cuando primero se determinan las clases que se necesitan y cómo van a interactuar y después se implementa la solución.

Las tarjetas CRC suelen ser creados a partir de tarjetas de índice en el que están escritos:

- El nombre de la clase.
- Su clase padre y sus clases.
- Las responsabilidades de la clase.
- Los nombres de otras clases con los que colaborará la clase para cumplir con sus responsabilidades.
- Autor.

El uso de una pequeña tarjeta mantiene la complejidad del diseño al mínimo. Se centra el diseño de las bases fundamentales de la clase y le impide entrar en detalles y su funcionamiento interno en un momento en que el detalle es probablemente contraproducente. También obliga a los diseñadores a que se abstengan de dar a la clase demasiadas responsabilidades. Dado que las tarjetas son portátiles, que pueden fácilmente ser expuesto en una mesa y reformularse el diseño en el debate con otras personas.

Un método común para determinar qué tarjetas se debe crear es leer las especificaciones del programa que se está diseñando y examinar si cada sustantivo debe ser una clase y si cada verbo debe ser una responsabilidad del sustantivo o clase a la que pertenece. Naturalmente, la existencia de un sustantivo o un verbo no requiere una clase o responsabilidad en el programa, pero se considera un buen punto de partida.

Tabla 4.2.1 Plantilla para las Tarjetas CRC

Tarjeta CRC	
Clase: <i>Nombre de la clase que se está modelando.</i>	
Súper Clase: <i>Nombre de la clase padre en la herencia.</i>	
Sub Clase(s): <i>Nombre de la(s) clase(s) hija en la herencia.</i>	
Responsabilidades: <i>Es una descripción de</i>	Colaboraciones: <i>Indica con cuáles otras clases</i>

<i>alto nivel del propósito de la clase.</i>	<i>se requiere relación para cumplir la responsabilidad.</i>
--	--

Tabla 4.2.2 Tarjeta CRC CargarDatos

Tarjeta CRC	
Clase: CargarDatos	
Súper Clase: EventDispatcher	
Sub Clase(s): -	
Responsabilidades: Carga datos XML a partir de una URL que se le pase y cuando termina lanza un evento a través de la clase EventDispatcher. Carga una imagen a partir de una URL que se le pase y cuando termina lanza un evento a través de la clase EventDispatcher.	Colaboraciones: EventDispatcher EventoXML

Tabla 4.2.3 Tarjeta CRC EventoXML

Tarjeta CRC	
Clase: EventoXML	
Súper Clase: Event	
Sub Clase(s): -	
Responsabilidades: Esta clase es lanzada como evento cuando se termina de cargar los datos XML.	Colaboraciones: Event

Tabla 4.2.4 Tarjeta CRC Parseo

Tarjeta CRC	
Clase: Parseo	
Súper Clase: -	
Sub Clase(s): -	
<p>Responsabilidades:</p> <p>Genera un XML con toda la información de proyecto que ha sido creado en caso de no poder, lanza un error.</p> <p>Genera un XML con toda la información de tema que ha sido creado en caso de no poder, lanza un error.</p> <p>Genera un XML con toda la información de Pop_Up que ha sido creado en caso de no poder, lanza un error.</p> <p>Genera un XML con toda la información de galería que ha sido creado en caso de no poder, lanza un error.</p> <p>Genera un XML con toda la información de video que ha sido creado en caso de no poder, lanza un error.</p> <p>Genera un XML con toda la información de música que ha sido creado en caso de no poder, lanza un error.</p>	<p>Colaboraciones:</p> <p>Clase CProyecto</p> <p>Clase Tema</p> <p>Clase PopUP</p> <p>Clase Galeria</p> <p>Clase Video</p> <p>Clase Musica</p>

Tabla 4.2.5 Tarjeta CRC ParseoInverso

Tarjeta CRC
Clase: ParseoInverso
Súper Clase: -

Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad generar un proyecto a partir de un archivo XML que contenga la información, para esto genera cada uno de los objetos que contiene un proyecto.	Colaboraciones: Clase CSS Clase Musica Clase Video Clase Galeria Clase Imagen Clase PopUP Clase Texto Clase Tema Clase CProyecto

Tabla 4.2.6 Tarjeta CRC Navegación

Tarjeta CRC	
Clase: Navegacion	
Súper Clase: -	
Sub Clase(s): NavegacionLineal, NavegacionGlobal	
Responsabilidades: Tiene por responsabilidad generar la navegación de un proyecto creado.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.7 Tarjeta CRC NavegacionLineal

Tarjeta CRC	
Clase: NavegacionLineal	
Súper Clase: Navegacion	
Sub Clase(s): -	
Responsabilidades:	Colaboraciones:

Tiene por responsabilidad generar la navegación lineal de un proyecto creado.	Clase IConfiguracion
---	----------------------

Tabla 4.2.8 Tarjeta CRC NavegacionGlobal

Tarjeta CRC	
Clase: NavegacionGlobal	
Súper Clase: Navegacion	
Sub Clase(s): -	
Responsabilidades: Tiene por responsabilidad generar la navegación global de un proyecto creado.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.9 Tarjeta CRC CSS

Tarjeta CRC	
Clase: CSS	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de una CSS asociada a un tema.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.10 Tarjeta CRC Musica

Tarjeta CRC	
Clase: Musica	
Súper Clase: -	

Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de una Música asociada a un tema.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.11 Tarjeta CRC Video

Tarjeta CRC	
Clase: Video	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de un Video asociado a un tema.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.12 Tarjeta CRC Galeria

Tarjeta CRC	
Clase: Galeria	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de una Galería asociada a un tema, el cual va a contener todas las Imágenes asociadas a ella.	Colaboraciones: Clase IConfiguracion Clase Imagen

Tabla 4.2.13 Tarjeta CRC Imagen

Tarjeta CRC	
Clase: Imagen	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de una Imagen asociada a una Galería.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.14 Tarjeta CRC PalabraCaliente

Tarjeta CRC	
Clase: PalabraCaliente	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de una palabra caliente asociada a un Texto.	Colaboraciones:

Tabla 4.2.15 Tarjeta CRC PopUp

Tarjeta CRC	
Clase: PopUp	
Súper Clase: Texto	
Sub Clase(s): -	
Responsabilidades:	Colaboraciones:

Esta clase tiene como responsabilidad contener la información de un Pop_up asociado a un Tema.	Clase Texto Clase IConfiguracion
--	-------------------------------------

Tabla 4.2.16 Tarjeta CRC Texto

Tarjeta CRC	
Clase: Texto	
Súper Clase: -	
Sub Clase(s): PopUp	
Responsabilidades: Esta clase tiene como responsabilidad contener la información de un Texto asociado a un Tema.	Colaboraciones: Clase IConfiguracion

Tabla 4.2.17 Tarjeta CRC Tema

Tarjeta CRC	
Clase: Tema	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase va a contener toda la información de un Tema, dígame: CSS; Música; Video; Galería; Imagen; PopUp; Texto y demás informaciones asociadas a un tema.	Colaboraciones: Clase CSS Clase Musica Clase Video Clase Galeria Clase Imagen Clase PopUP Clase Texto Clase IConfiguracion

Tabla 4.2.18 Tarjeta CRC IConfiguracion

Tarjeta CRC	
Clase: IConfiguracion	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase es una interface que se define para el trabajo con las configuraciones de cada objeto.	Colaboraciones:

Tabla 4.2.19 Tarjeta CRC CProyecto

Tarjeta CRC	
Clase: CProyecto	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta será la clase controladora de la aplicación y tendrá como objetivo contener todos los temas asociados a un Proyecto.	Colaboraciones: Clase Tema Clase IConfiguracion

Tabla 4.2.20 Tarjeta CRC FactoryService

Tarjeta CRC	
Clase: FactoryService	
Súper Clase: -	

Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad construir los objetos para acceder a las clases controladoras. En esta se implementan los patrones Singleton y Factory.	Colaboraciones: Clase ProyectoService Clase UsuarioService Clase ContenidoService

Tabla 4.2.21 Tarjeta CRC FactoryDao

Tarjeta CRC	
Clase: FactoryDao	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad construir los objetos para acceder a las clases de acceso a dato. También implementa los patrones Singleton y Factory.	Colaboraciones: Clase ProyectoDao Clase UsuarioDao Clase ContenidoDao

Tabla 4.2.22 Tarjeta CRC ProyectoService

Tarjeta CRC	
Clase: ProyectoService	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad la implementación de la lógica del negocio del módulo proyecto. Entre sus principales	Colaboraciones:

funcionalidades tiene: cargar, eliminar, guardar y actualizar proyectos.
--

Tabla 4.2.23 Tarjeta CRC UsuarioService

Tarjeta CRC	
Clase: UsuarioService	
Súper Clase: -	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad la implementación de la lógica del negocio del módulo administración. Entre sus principales funcionalidades tiene: cargar, eliminar, guardar y actualizar Usuario, además de garantizar los parámetros para garantizar la autenticación.	Colaboraciones:

Tabla 4.2.24 Tarjeta CRC ContenidoDao

Tarjeta CRC	
Clase: ContenidoDao	
Súper Clase: - Conexion	
Sub Clase(s): -	
Responsabilidades: Esta clase tiene como responsabilidad la implementación del acceso a datos del módulo contenido. Es la principal responsable de garantizar la persistencia de los datos en la	Colaboraciones: Clase Conexion

<p>base de datos, debido a que es la encargada de mandar a ejecutar las consultas SQL correspondientes a insertar, modificar, ver y eliminar contenidos.</p>	
--	--

Tabla 4.2.25 Tarjeta CRC UsuarioDao

Tarjeta CRC	
Clase: UsuarioDao	
Súper Clase: - Conexion	
Sub Clase(s): -	
<p>Responsabilidades: Esta clase tiene como responsabilidad la implementación del acceso a datos del módulo administración. Es la principal responsable de garantizar la persistencia de los datos en la base de datos, debido a que es la encargada de mandar a ejecutar las consultas SQL correspondientes a insertar, modificar, ver y eliminar usuarios.</p>	<p>Colaboraciones: Clase Conexion</p>

Tabla 4.2.26 Tarjeta CRC ProyectoDao

Tarjeta CRC	
Clase: ProyectoDao	
Súper Clase: - Conexion	
Sub Clase(s): -	
Responsabilidades:	Colaboraciones:

<p>Esta clase tiene como responsabilidad la implementación del acceso a datos del módulo Proyecto. Es la principal responsable de garantizar la persistencia de los datos en la base de datos, debido a que es la encargada de mandar a ejecutar las consultas SQL correspondientes a insertar, modificar, ver y eliminar proyectos.</p>	<p>Clase Conexion</p>
--	-----------------------

Tabla 4.2.27 Tarjeta CRC Conexion

Tarjeta CRC	
<p>Clase: Conexion</p>	
<p>Súper Clase: -</p>	
<p>Sub Clase(s): -</p>	
<p>Responsabilidades: Esta clase tiene como responsabilidad conectarse al gestor de base de datos y ejecutar las consultas SQL, de esta heredan las clases DAO.</p>	<p>Colaboraciones: Clase: ProyectoDao Clase: UsuarioDao Clase: ContenidoDao</p>

4.2.3 Diseño de la Base de Datos

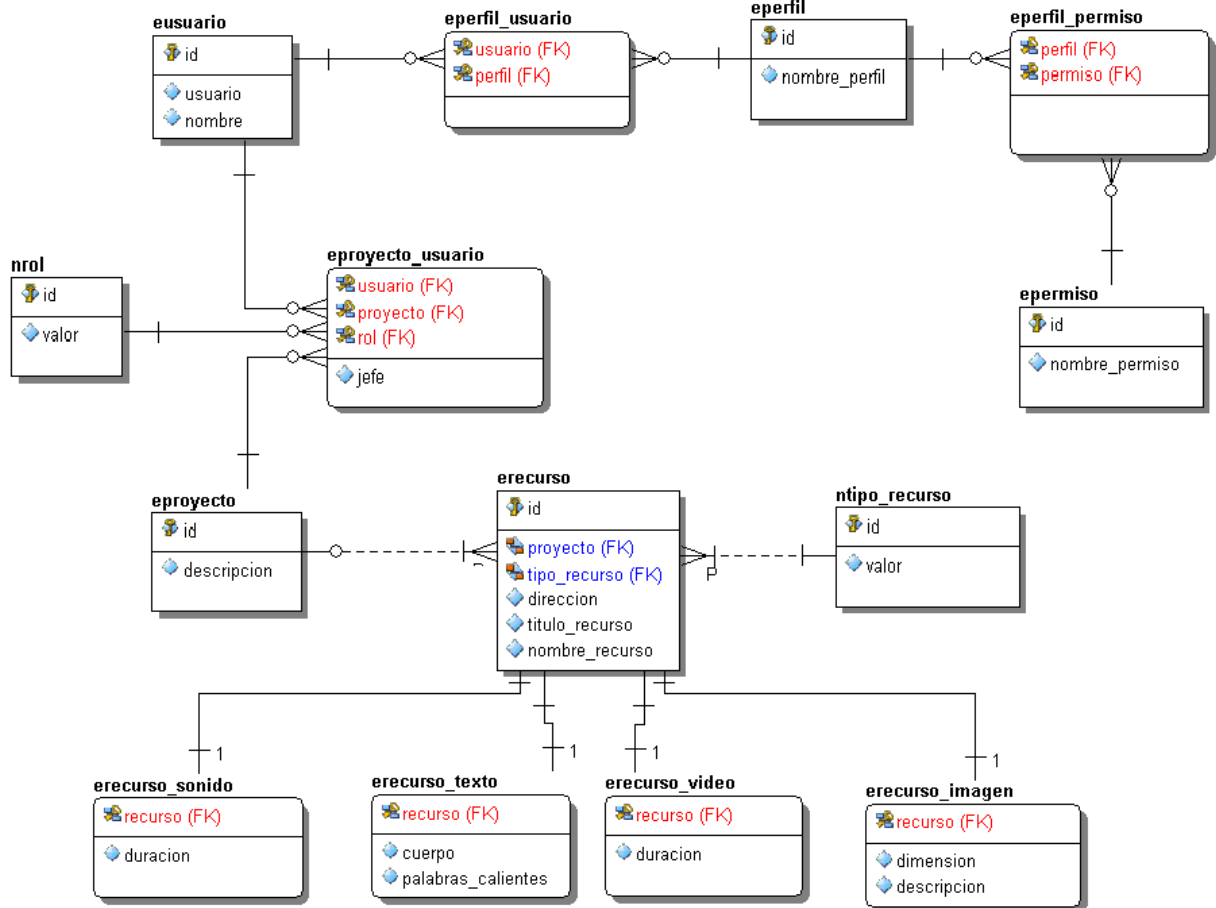
El diseño de la base de datos juega un papel fundamental para cualquier aplicación que gestione información, sobre todo para las aplicaciones web; es por ello que se hace necesario que los datos se almacenen de forma coherente y organizada, para evitar que dicha información se pierda.

El principal objetivo para realizar el diseño de la base de datos en cuestión, es lograr la persistencia de los datos de los usuarios, así como la información de los recursos que se gestionan por los mismos, controlando que esta información no tenga redundancia y no existan datos innecesarios que no cumplen ningún objetivo al ser guardados.

El modelo de datos para la aplicación posee un nivel de complejidad bajo, debido a que no se

almacenarán grandes volúmenes de información. A continuación se muestra cómo está organizado el modelo de datos del sistema.

Modelo de Datos



4.3 Fase Implementación

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o una persona), responsable de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan cada una de las iteraciones.

4.3.1 Tareas de la programación

Iteración 1

En esta iteración se pretende implementar las historias de usuario que más importancia tienen para el cliente, las cuales son: la No 1 Gestionar cuenta de usuarios y la No 2 Importar contenidos, las cuales le darán cumplimiento a las funcionalidades crear, modificar, eliminar cuentas de usuarios, autenticar en el sistema, ver perfil de usuarios, publicar, mostrar, modificar y eliminar contenidos. Una vez que se terminen de implementar todas las funcionalidades de estas iteraciones, se procederá a liberar un ejecutable con la finalidad de retroalimentarnos del cliente y solventar posibles errores.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar cuenta de usuarios	1	0.8
Importar contenidos	2	2.4

Tabla 4.3.1 1ra Tarea de la HU #-1

Tarea	
Número de tarea: 1	Número de HU: 1

Nombre de la tarea: Implementación del módulo de gestión de usuarios.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.8
Fecha inicio: 23/02/09	Fecha fin: 27/02/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases que permitan gestionar los usuarios	

Tabla 4.3.2 2da Tarea de la HU #-1

Tarea	
Número de tarea: 2	Número de HU: 1
Nombre de la tarea: Implementación del módulo de Importar contenido.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2.4
Fecha inicio: 02/03/09	Fecha fin: 17/03/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases que permitan importar contenido	

Iteración 2

En esta iteración se implementarán las historias de usuario que le siguen en importancia a las tareas críticas que fueron implementadas en la primera iteración como es la historia de usuario No 3 Crear nuevo contenido, con la cual se implementará la funcionalidad de crear nuevo contenido, esta dota al sistema de sus características más relevantes como son las de proveer al desarrollador de herramientas y los medios necesarios para la creación de contenidos nuevos. Al finalizar esta iteración se liberará un ejecutable con la finalidad de retroalimentarnos del cliente y solventar posibles errores.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Crear nuevo contenido	3	5

Tabla 4.3.3 1ra Tarea de la HU #-3

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación de la clase controladora CProyecto	

Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha inicio: 18/03/09	Fecha fin: 19/03/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán la clase CProyecto.	

Paquete GestionNavegacion

Este paquete contiene las clases encargadas de controlar la navegación del proyecto, dichas clases son: Navegacion; NavegacionLineal; NavegacionGlobal.

Tabla 4.3.4 1ra Tarea de la HU #-3 Paquete *GestionNavegacion*.

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Gestión de Navegación	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.3
Fecha inicio: 20/03/09	Fecha fin: 23/03/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Paquete GestionDatos

Este paquete contiene las clases encargadas de gestionar la carga de todo los datos del proyecto, dichas clases son: EventoXML; CargarDatos; Parseo; ParesoInverso.

Tabla 4.3.5 1ra Tarea de la HU #-3 Paquete *GestionDatos*

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Gestion de Datos	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.6
Fecha inicio: 24/03/09	Fecha fin: 27/03/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Paquete Interface

Este paquete contiene las clases interface del proyecto, dichas clases son: IConfiguration.

Tabla 4.3.6 1ra Tarea de la HU #-3 Paquete *Interface*

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Interface	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha inicio: 30/03/09	Fecha fin: 31/03/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Paquete Gestión de Temas

Este paquete contiene las clases y paquetes encargados de la gestión de los temas de un proyecto. Su estructura se refleja debajo definiéndose como paquetes: GestionGaleria; GestionTextos, Contenidos y los demás elementos que lo componen son las clases agrupadas en estos paquetes.

- GestionGaleria
 - Galeria
 - Imagen
- GestionTextos
 - PopUp
 - Texto
 - PalabraCaliente
- Contenidos
 - CSS
 - Musica
 - Video
- Tema

Tabla 4.3.7 1ra Tarea de la HU #-3 Paquete *Gestión de Temas*

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación de la clase Tema	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha inicio: 01/04/09	Fecha fin: 02/04/09

Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña
Descripción: Se implementará la clase Tema

Tabla 4.3.8 2da Tarea de la HU #-3 Paquete *Gestión de Temas*

Tarea	
Número de tarea: 1	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Gestión de galerías	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 03/04/09	Fecha fin: 10/04/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Tabla 4.3.9 3ra Tarea de la HU #-3 Paquete *Gestión de Temas*

Tarea	
Número de tarea: 3	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Gestión de Textos	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.7
Fecha inicio: 13/04/09	Fecha fin: 16/04/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Tabla 4.3.10 4ta Tarea de la HU #-3 Paquete *Gestión de Temas*

Tarea	
Número de tarea: 4	Número de HU: 3
Nombre de la tarea: Implementación del Paquete Gestión de Contenidos	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.6
Fecha inicio: 17/04/09	Fecha fin: 22/04/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases pertinentes que componen dicho paquete.	

Tabla 4.3.1 2da Tarea de la HU #-3

Tarea	
Número de tarea: 2	Número de HU: 3
Nombre de la tarea: Implementación de las Interfaces Visuales	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.4
Fecha inicio: 23/04/09	Fecha fin: 04/05/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las interfaces visuales para trabajo con el sistema.	

Iteración 3

En esta iteración se implementarán las historias de usuario restantes, las cuales están diseñadas con el objetivo de lograr la organización de la información y exportación de los proyectos, tareas que son un poco engorrosas de realizar manualmente; estas tareas están contenidas en las historias de usuarios: No 4 Gestionar Proyectos y No 5 Exportar Proyectos, las cuales permiten crear, eliminar y exportar un proyecto.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar Proyectos	1	0.7
Exportar Proyectos	2	1.0

Tabla 4.3.2 1ra Tarea de la HU #-4

Tarea	
Número de tarea: 1	Número de HU: 4
Nombre de la tarea: Implementación del Módulo Gestión de Proyectos	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha inicio: 04/05/09	Fecha fin: 05/05/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases que permitan Gestionar proyectos.	

Tabla 4.3.3 2da Tarea de la HU #-5

Tarea	
Número de tarea: 2	Número de HU: 5
Nombre de la tarea: Implementación del Módulo Exportar proyectos	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.0
Fecha inicio: 06/05/09	Fecha fin: 12/05/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases que permitan Exportar Proyectos.	

Tabla 4.3.4 3ra Tarea de la HU #-5

Tarea	
Número de tarea: 3	Número de HU: 5
Nombre de la tarea: Implementación del Módulo Administrativo	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.5
Fecha inicio: 13/05/09	Fecha fin: 15/05/09
Programador responsable: José Antonio Soto Pérez-Eric Manuel Cobas Peña	
Descripción: Se implementarán las clases que generen el módulo de administración.	

4.3.2 Diagramas de componentes

Un diagrama de componentes representa cómo se divide el software en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Los diagramas de componentes son usados para modelar la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes, aunque uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

Con el objetivo de hacer más detallado el estudio de la aplicación, se presentan a continuación los diagramas de componentes de la misma.

Diagrama de Componentes AS3

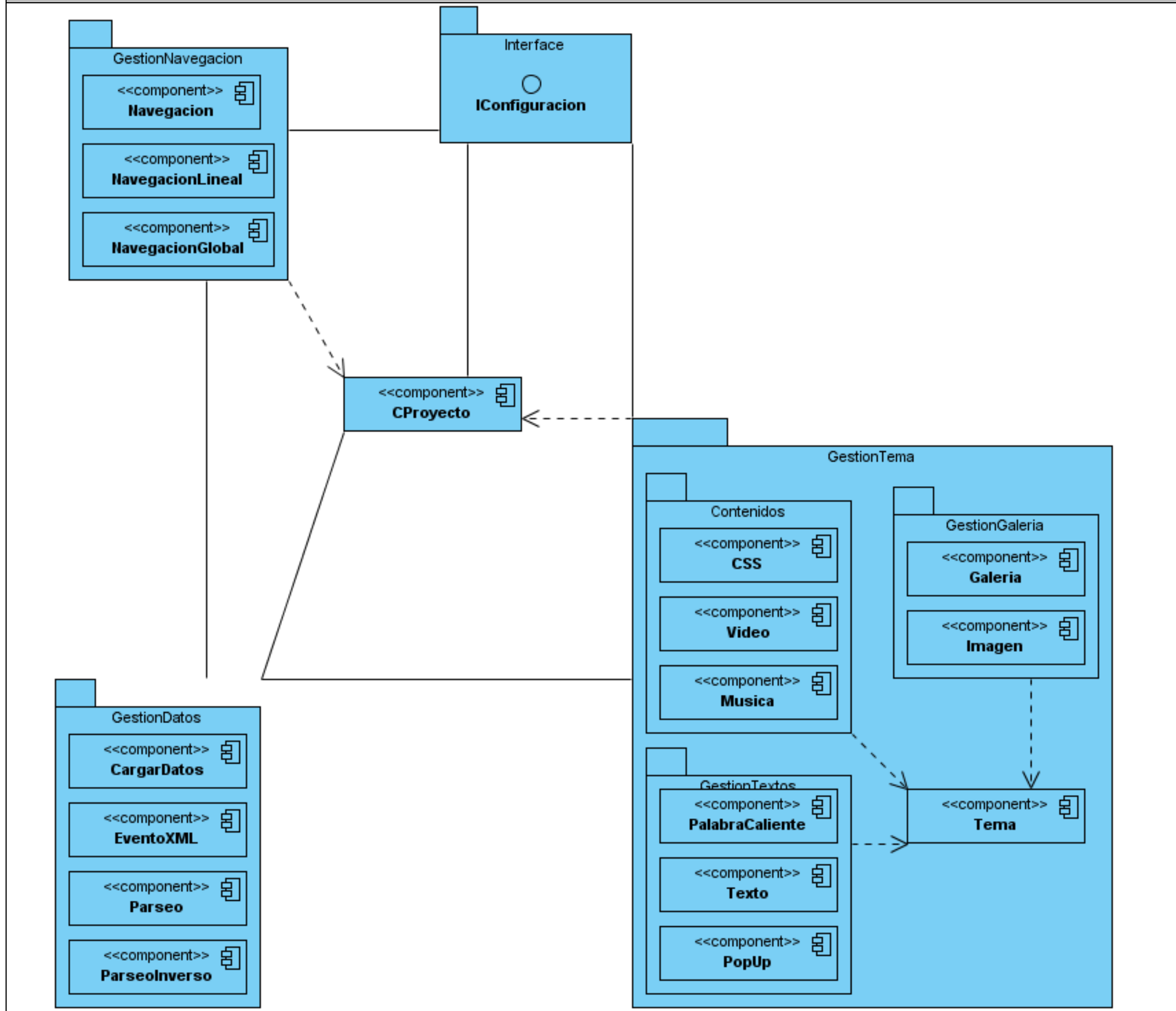
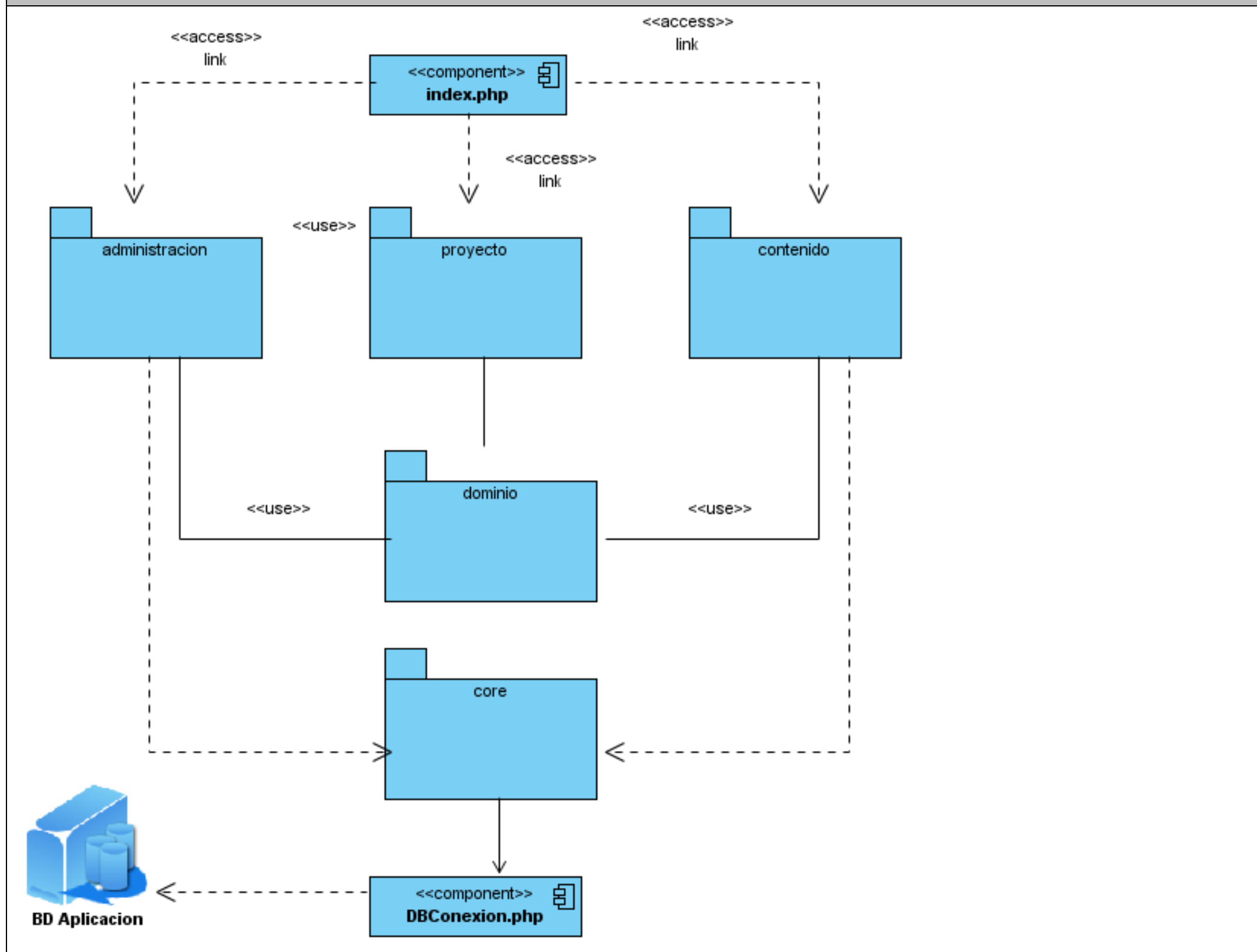


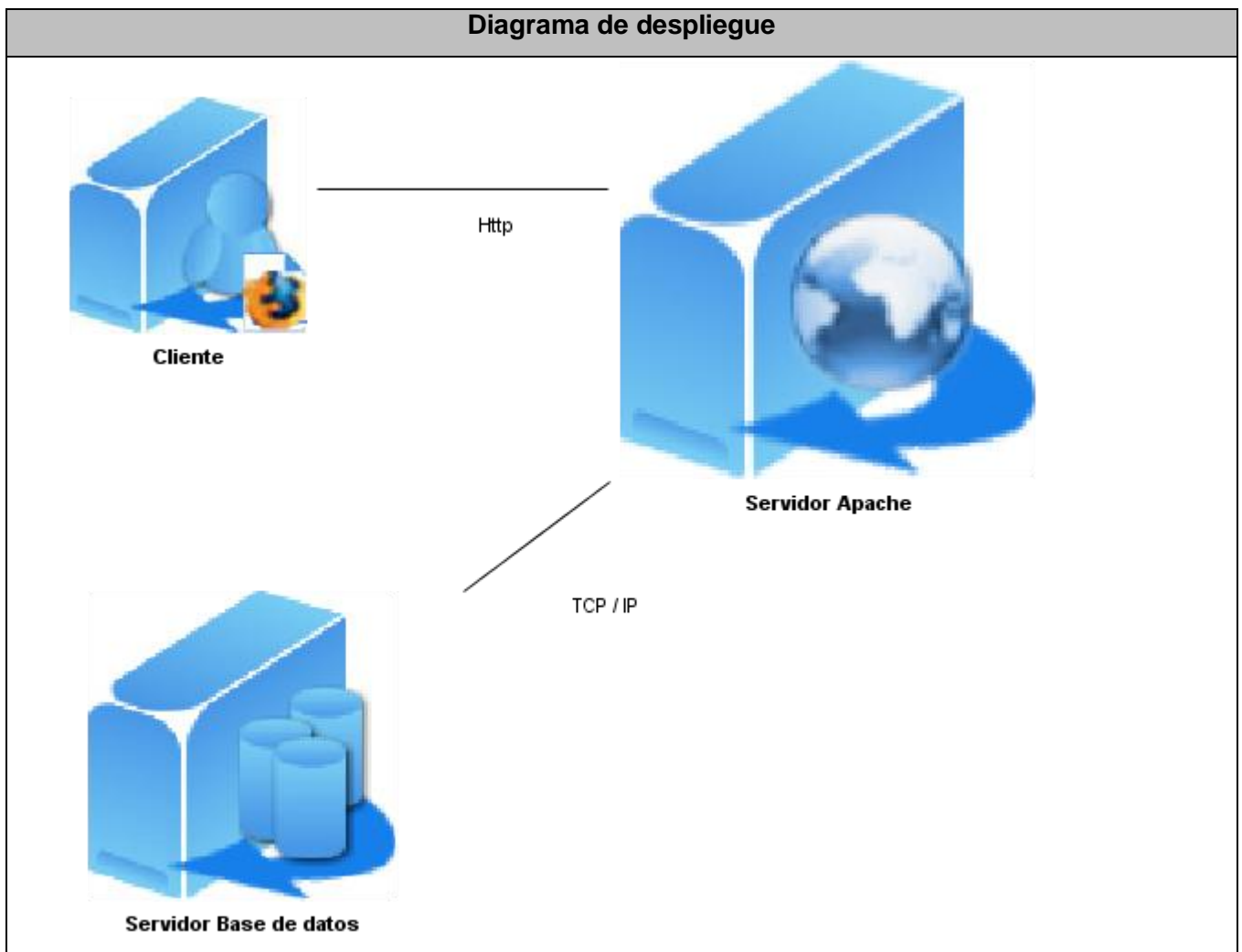
Diagrama de Componentes Php



4.3.3 Diagrama de despliegue

El Diagrama de Despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. La mayoría de las veces el modelado de la vista de despliegue permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación.

En este caso la aplicación se encuentra hospedada en un servidor Web y este se encuentra conectado a un servidor de base de datos configurado con PostgreSQL, como se puede apreciar en el siguiente diagrama.



4.3.4 Multiplataforma

Las aplicaciones multiplataforma son aquellas que pueden funcionar en diferentes sistemas operativos u ordenadores, pero el código fuente es el mismo, o sea, una aplicación que podría ejecutarse en un ordenador con sistema operativo Windows con un procesador x86, en un ordenador con sistema operativo GNU/Linux con un procesador x86, o en uno con sistema operativo Mac OS X.

La aplicación en cuestión (Prima-Frame) está diseñada para que funcione en cualquier servidor Web independientemente del sistema operativo en el que esté funcionando dicho servidor. Actualmente es muy fácil encontrar un servidor dónde hospedar una aplicación web, sobre todo si se usa PHP como lenguaje de programación.

En cuanto al gestor de bases de datos no hay razones para preocuparse debido a que PostgreSQL lo podemos encontrar en una gran cantidad de los servidores Web que usan PHP, pero si no estuviera instalado, la aplicación cuenta con un patrón de bajo acoplamiento que permite, con solo cambiar una clase se pueda usar el gestor de base de datos que se necesite, por el momento solo está diseñado para trabajar con PostgreSQL y MySQL, pero se puede extender hasta donde se desee.

4.3.5 Estándares de Codificación

Los estándares de codificación son reglas específicas a un lenguaje que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, su función es impedir que ocurran errores. Los errores frecuentes en programas pueden ser detectados mucho anterior o pueden incluso ser evitados totalmente. Durante el desarrollo de un software determinado, los estándares de codificación ayudan a los programadores a producir un código de alta calidad, de fácil entendimiento y altamente reutilizable. También realzan considerablemente la capacidad de mantenimiento y reutilización a largo plazo del producto final. Tal práctica del control de errores en el proceso del desarrollo mejora la calidad del producto final, mientras que reduce el tiempo de desarrollo, el coste, y el esfuerzo. Para la implementación del sistema se definieron las siguientes pautas de codificación:

Idioma

Todos los nombres de clases, funciones y variables se definen en español.

Nombres de Clases

Las clases comenzarán con letra inicial mayúscula y en caso de tener nombre compuesto dichas palabras comenzarán con mayúscula también.

Ejemplo: class PopUp

Tipos de Clases

Clases Interfaz: comienzan siempre con el prefijo I, seguido por el nombre de la clase.

Clases Controladoras: comienzan con la letra C, seguida por el nombre de la clase.

Ejemplos: IConfiguracion, CProyecto.

Nombres de Atributos de Clases

Todos los nombres de variables se definen en minúsculas e inician con un “_”.

Deben utilizarse nombres descriptivos que permitan una comprensión rápida de la función de dicha variable.

Nombres de Funciones

Deben utilizarse nombres descriptivos que permitan identificar rápidamente el objetivo de la función, sin importar cuán largo sea el nombre.

Comienzan con letra mayúscula, y en caso de existir, el resto de las palabras que forman el nombre comenzarán con mayúsculas.

Nombres de Argumentos de Funciones

Comienzan con letra minúscula “p”, y el resto de las palabras que forman el nombre comenzarán con mayúsculas.

En los prototipos de funciones se incluirá el nombre de los argumentos debido a que esto hace más legible aún las etiquetas de completamiento de código a la hora de programar, obteniendo así una mayor comprensión de qué argumentos se requieren.

Nombres de Variables Temporales

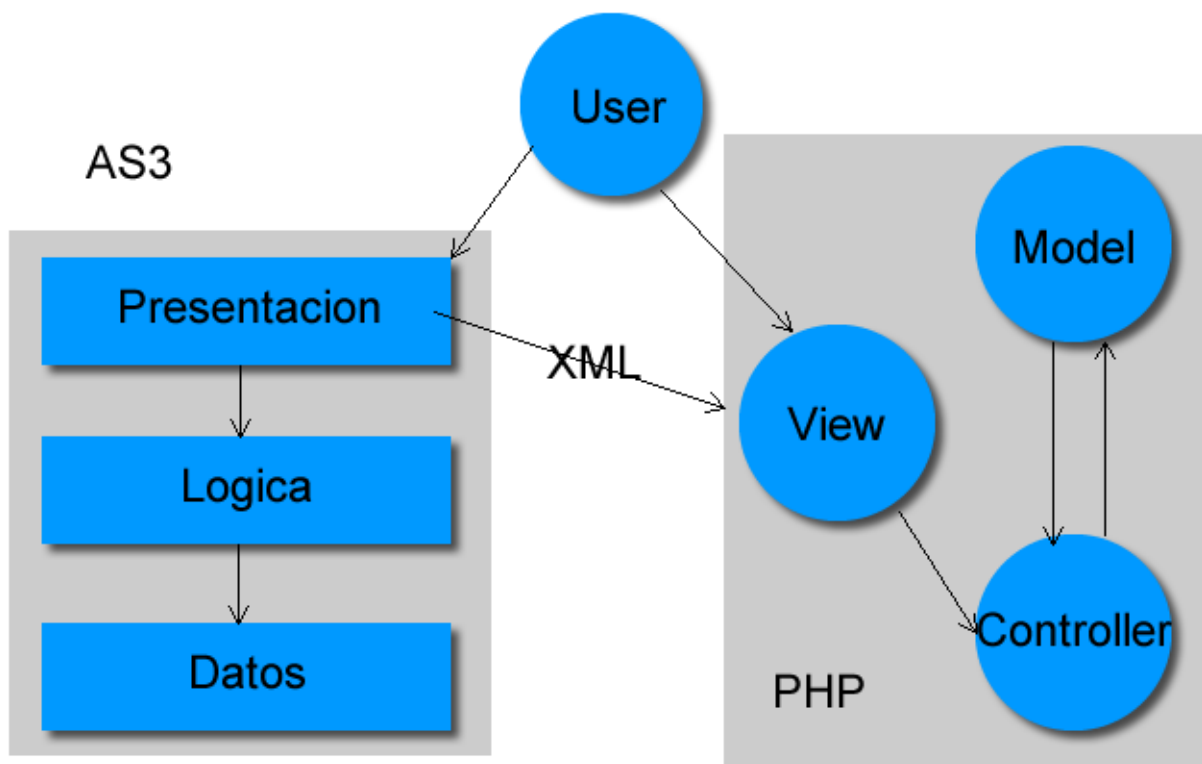
Se utilizarán nombres lo más cortos posible, pero que transmitan bien su significado y todos deben empezar con el sufijo “temp”, los demás nombre deben comenzar con mayúscula.

Ejemplo: tempCantTemas:Number.

4.3.6 Arquitectura

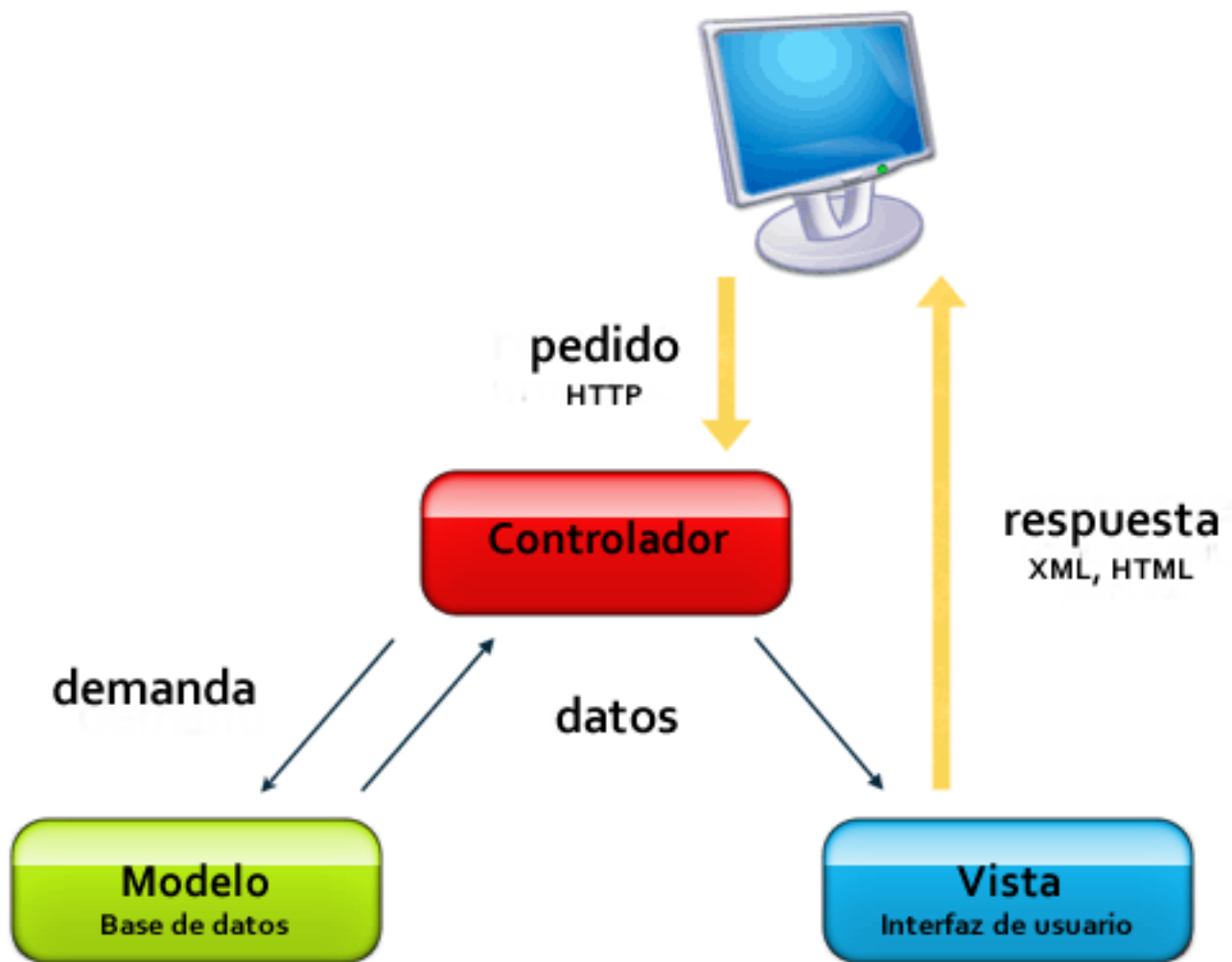
La Arquitectura del Software aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. La definición oficial del término Arquitectura del Software, fue pronunciado por la IEEE Std 1471-2000: “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (IEEE, 2000) El objetivo principal de la Arquitectura del Software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto. Para conseguirlo, la Arquitectura del Software construye abstracciones, materializándolas en forma de diagramas comentados.

Para la aplicación se decidió usar una arquitectura basada en el patrón modelo-vista-controlador, aunque también se usan otros patrones como el Singleton y el Factory. En la siguiente figura se muestra como queda estructurada la arquitectura que implementa la aplicación.



4.3.7 Patrones

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. La siguiente figura muestra detalladamente la estructura del patrón.



La aplicación cuenta con varios módulos, cada uno de ellos cuenta con esta estructura, por ejemplo en el módulo proyecto, que es el encargado de gestionar todo lo referente a los proyectos, se pueden encontrar una carpeta llamada acceso a datos que contiene la clase contenidoDao que es la encargada de gestionar los datos en el modelo, otra carpeta llamada negocio, que en esta se almacena la clase contenidoService, que es la que controla la lógica del negocio de módulo, y por último se tiene otra carpeta llamada web, que es donde se almacenan las páginas que conforman la interfaz de usuario del módulo.

Prima-Frame cuenta además con un módulo llamado core (núcleo) que es usado por el resto de los módulos, debido a que en este se implementan los patrones Factory y Singleton, para ello se crean las clases FactoryDao y FactoryService, la primera es la encargada de crear los objetos de acceso a datos de cada módulo y la segunda tiene como tarea crear objetos para acceder a la capa de negocio, garantizando al mismo tiempo que dichos objetos se creen una sola vez.

4.4 Fase de Prueba

4.4.1 Pruebas de Aceptación

Uno de los pilares fundamentales de XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Crear cuenta en el sistema	
Descripción: Prueba para la funcionalidad de crear una nueva cuenta en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta crear una cuenta por un usuario que contiene sus datos válidos, es decir, que cumple con el rol administrador.	
Resultado Esperado: La cuenta es creada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1

Nombre: Ver perfil de usuario
Descripción: Prueba para la funcionalidad de ver perfil de usuario
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.
Entrada/ Pasos de ejecución: Se intenta mostrar el perfil de un usuario, la acción puede ser realizada por cualquier usuario del sistema
Resultado Esperado: El perfil es mostrado sin errores.
Evaluación de la Prueba: Prueba satisfactoria.

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Modificar cuenta de usuario	
Descripción: Prueba para la funcionalidad de modificar cuenta de usuario.	
Condiciones de Ejecución: El usuario debe estar autenticado y debe contener el perfil administrador. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta modificar la cuenta de un usuario, por un usuario que contiene el perfil administrador.	
Resultado Esperado: La cuenta es modificada sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Importar contenidos	
Descripción: Prueba para la funcionalidad de importar contenidos en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta importar contenidos por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de desarrollador o administrador.	
Resultado Esperado: Los contenidos son importados sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación

Código: HU3_P1	Historia de Usuario: 3
Nombre: Crear nuevo contenido	
Descripción: Prueba para la funcionalidad de crear nuevo contenido en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta crear un nuevo contenido por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de administrador o como desarrollador.	
Resultado Esperado: El contenido es creado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Mostrar contenidos	
Descripción: Prueba para la funcionalidad de Mostrar contenidos del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta mostrar los contenidos de un tipo en específico, por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de administrador o como desarrollador.	
Resultado Esperado: Los contenidos son mostrados sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Modificar contenidos	
Descripción: Prueba para la funcionalidad de modificar contenidos del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta modificar un contenido por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de desarrollador o administrador.	
Resultado Esperado: Los contenidos son modificados sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Eliminar contenidos	
Descripción: Prueba para la funcionalidad de eliminar contenidos del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta eliminar un contenido del sistema por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de desarrollador o administrador.	
Resultado Esperado: El contenido es eliminado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Crear nuevo proyecto	
Descripción: Prueba para la funcionalidad de crear nuevo proyecto en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y debe contener el perfil de administrador. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta crear nuevo proyecto por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de administrador.	
Resultado Esperado: El proyecto es creado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Eliminar proyecto	
Descripción: Prueba para la funcionalidad de eliminar proyecto del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y debe contener el perfil administrador. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta eliminar un proyecto por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de administrador	
Resultado Esperado: El proyecto es eliminado sin errores.	

Evaluación de la Prueba: Prueba satisfactoria.

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Exportar proyecto	
Descripción: Prueba para la funcionalidad de exportar proyecto del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada/ Pasos de ejecución: Se intenta exportar un proyecto por un usuario que contiene sus datos válidos, es decir, que cumple con el rol de desarrollador.	
Resultado Esperado: El proyecto es exportando sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

4.5 Conclusiones

En este capítulo se construyó el modelo necesario para desarrollar el proceso de implementación del sistema. Se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y las pruebas de aceptación que propician al cliente conformidad y seguridad ante el sistema. Con el fin de este capítulo se da por terminada la propuesta.

Conclusiones

Con la finalización del presente trabajo se cumplieron los objetivos y metas planteadas al inicio de la investigación:

1. Se realizó un estudio profundo del estado del arte de las herramientas libres existentes en el mundo del software libre para la gestión de contenido para aplicaciones con tecnología multimedia, seleccionándose aquellas que podrían utilizarse como referencia de la solución.
2. Se obtuvo una lista de las funcionalidades del sistema y de propiedades del sistema que definieron la aplicación final, así como su comportamiento.
3. Con la aplicación del XP como metodología para el desarrollo y con la aplicación de su principal práctica, que es el desarrollo dirigido por pruebas, se obtuvo una arquitectura sólida y flexible.
4. La evaluación de varios lenguajes para la programación permitió seleccionar el más adecuado de acuerdo a los objetivos y requerimientos de la aplicación.
5. Se cumplió con el objetivo principal de implementar un software multiplataforma para gestión de contenido para aplicaciones con tecnología usando software libre, el cual fue programado usando PHP y AS3 como lenguajes para la implementación, PHP para el subsistema de gestión y AS3 para el editor WYSIWYG, Visual Paradigm como herramienta CASE, y Eclipse PDT y Adobe Flex Builder como entornos de desarrollo.

Se concluye que el sistema brinda una solución factible a la problemática inicial y que su desarrollo y explotación significará una mejora considerable en las facilidades de producción de software educativo con tecnología multimedia para la UCI.

Recomendaciones

Como resultado del proceso de investigación y realización de la aplicación han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas:

1. Desarrollar un módulo para exportar proyecto que le permita al usuario personalizar los XML que se van a exportar, con el objetivo que sea ajustable a cualquier proyecto o framework con el que se trabaje.
2. Agregar nuevas funcionalidades en la medida del avance y de acuerdo con las expectativas de los usuarios finales, recomendadas por el equipo de desarrollo:
 - a. Añadir las animaciones como un nuevo tipo de media.
 - b. Eliminar las referencias que se encuentren en los XML a medias, que hayan sido eliminadas de la base de datos.
3. Se recomienda su uso en la producción de software educativo con tecnología multimedia, por los beneficios que provee el sistema.

Glosario de términos

Aplicaciones con tecnología multimedia: dicese de aquellas aplicaciones que combinan el uso de diferentes medias, dígase textos, sonidos e imágenes.

GNU/Linux: sistema operativo open source y es el mejor ejemplo de software libre, todo su código puede ser compartido, distribuido y modificado libremente.

Plantilla: son pautas que permiten guiar, portar o construir un diseño o esquema predefinido

Herramientas WYSIWYG: "lo que ves es lo que obtienes" (en inglés, What You See Is What You Get) son un conjunto de herramientas las cuales cumplen con la característica de poder visualizar en tiempo de edición la solución final.

Manifiesto Ágil: no es más que el resumen de los principios sobre los que se basan los métodos alternativos (metodologías ágiles) en cuatro postulados. (Beck, et al., 2001)

Metodologías Robustas: son las metodologías tradicionales de desarrollo de software como son el Proceso Unificado de Desarrollo y el Microsoft Solution Framework.

OMG: Grupo de Gestión de Objetos (de sus siglas en inglés Object Management Group), grupo dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA. (OMG, 08)

Java, C++, C#, ActionScript: lenguajes de programación.

Máquina Virtual de ActionScript (AMV): es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en código ActionScript (Grossman, et al., 2006)

Open Source: código abierto (en inglés open source) es el término con el que se conoce al software que cumple con las 4 libertades del usuario.

MXML: lenguaje de etiquetado extensible para multimedia (Multimedia eXtensible Markup Language), es un lenguaje de etiquetado similar al HTML, pero con una sintaxis menos ambigua y además se puede expandir. (Grossman, et al., 2006)

CSS: las hojas de estilo en cascada (CSS, por sus siglas en inglés Cascading Style Sheets), son un lenguaje formal utilizado para definir la presentación de un documento escrito en HTML o XML

Trabajos citados

- Adobe. 08.** Flex Builder for Linux. *Adobe Labs*. [En línea] 14 de 08 de 08. [Citado el: 07 de 02 de 09.] http://labs.adobe.com/technologies/flex/flexbuilder_linux/.
- Beck, Kent, y otros. 2001.** Manifiesto ágil para el desarrollo de software. *Manifiesto ágil para el desarrollo de software*. [En línea] 2001. [Citado el: 07 de 02 de 09.] <http://www.agilemanifesto.org/>.
- Desconocido. 09.** Presentacion. *Instituto Superior de Formación y Recursos en Red para el Profesorado*. [En línea] 07 de 02 de 09. [Citado el: 07 de 02 de 09.] <http://malted.cnice.mec.es/presentacion/QMalted.htm>.
- Don, Wells. 2006.** A gentle introduction. *Extreme Programming*. [En línea] 17 de 02 de 2006. [Citado el: 09 de 02 de 2009.] <http://www.extremeprogramming.org/>.
- Drake, Joshua D. 2009.** About. *PostgreSQL*. [En línea] 06 de 02 de 2009. [Citado el: 09 de 02 de 2009.] <http://www.postgresql.org/about/contact>.
- Educarex. 09.** Constructor Atenex. *Sitio Oficial de La Junta de Extremadura*. [En línea] 07 de 02 de 09. [Citado el: 07 de 02 de 09.] <http://constructor.educarex.es/>.
- Fowler, Martin. 2003.** La Nueva Metodología. [En línea] 04 de 2003. [Citado el: 09 de 02 de 2009.] <http://www.programacionextrema.org/articulos/newMethodology.es.html>.
- Grossman, Gary y Huang, Emmy. 2006.** Adobe. [En línea] 27 de Junio de 2006. [Citado el: 09 de 02 de 2009.] http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html.
- Mike. 2009.** UML. *OMG*. [En línea] 16 de 01 de 2009. [Citado el: 09 de 02 de 2009.] <http://www.uml.org/>.
- Mozilla Foundation. 09.** Features. *Mozilla Foundation*. [En línea] 03 de 02 de 09. [Citado el: 07 de 02 de 09.] <http://www.mozilla-europe.org/es/firefox/features/>.
- OMG. 08.** About the Object Management Group™ (OMG™). *OMG*. [En línea] 17 de 10 de 08. [Citado el: 07 de 02 de 09.] <http://www.omg.org/gettingstarted/index.htm>.
- Palu, Mika. 2008.** Features. *Flash Develop*. [En línea] 17 de 11 de 2008. [Citado el: 09 de 02 de 2009.] <http://www.flashdevelop.org/wikidocs/index.php?title=Features>.
- PHP. 07.** News Archive. *PHP*. [En línea] 09 de 11 de 07. [Citado el: 07 de 02 de 09.] <http://www.php.net/archive/2007.php>.
- Steiner, Jakub, y otros. 08.** Features. *Gimp*. [En línea] 09 de 11 de 08. [Citado el: 07 de 02 de 09.] <http://www.gimp.org/features/>.
- Sun Microsystem. 07.** Open Office. *Open Office*. [En línea] 04 de 02 de 07. [Citado el: 07 de 02 de 09.]

09.] <http://www.openoffice.org/>.

Valerio, Adrián Anacleto. 2005. Introducción a UML 2.0. [En línea] 04 de 10 de 2005. [Citado el: 09 de 02 de 2009.] http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.

Visual Paradigm. 08. Visual Paradigm para UML. *Visual Paradigm*. [En línea] 05 de 10 de 08. [Citado el: 07 de 02 de 09.] <http://www.visual-paradigm.com/product/vpum/>.

Zend. 2008. <http://www.zend.com>. *http://www.zend.com*. [En línea] Zend, 2008. [Citado el: 08 de Febrero de 2009.] <http://www.zend.com/en/community/pdt>.

Anexo 1: Estructura del XML de un proyecto

```
<tema id="Titulo del tema" bus="2">
  <css>Url donde se encuentra la css</css>
  <videos>
    <video>
      <titulo>Título con que se mostrara el video en la aplicación</titulo>
      <nombre>Nombre del archivo de video</nombre>
      <duracion>Tiempo de duración</duracion>
      <url>Url donde se encuentra el video</url>
    </video>
  </videos>
  <musicas>
    <musica>
      <titulo>Título con que se mostrara la música en la aplicación</titulo>
      <nombre>Nombre del archivo de música</nombre>
      <duracion>Tiempo de duración</duracion>
      <url>Url donde se encuentra la canción</url>
    </musica>
  </musicas>
  <contenido>
    <titulo>Título que mostrara la aplicacion para este texto</titulo>
    <cuerpo>Aquí iría el contenido del tema en cuestión</cuerpo>
  </contenido>
  <popups>
    <popup>
      <titulo>Título que mostrara la aplicacion para este pop up</titulo>
      <cuerpo>Aquí iría el contenido del pop up en cuestión</cuerpo>
    </popup>
  </popups>
  <galeria>
    <cantThumb>Cantidad de thumb por pagina</cantThumb>
  <imagen>
```

<thumbHeight>Dimensiones de la vista previa</thumbHeight>
<thumbWidth>Dimensiones de la vista previa</thumbWidth>
<width>Dimensiones de la imagen</width>
<height>Dimensiones de la imagen</height>
<nombreArchivo>Nombre de la imagen</nombreArchivo>
<tituloImagen>Título con que se mostrara la imagen en la aplicación/tituloImagen>
<descripcion>Descripción que se mostrará de la imagen</descripcion>
<url>Url donde se encuentra la imagen</url>
</imagen>
</galeria>
</tema>