



Universidad de las Ciencias
Informáticas

Facultad 6

Desarrollo de un editor WYSIWYG de plantillas de presentación de tipologías de archivos multimedia para la plataforma VideoWeb

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
INFORMÁTICA**

Autor: Frank Rafael Pacareu De La Cruz

Tutor: Frank Torres Rodríguez

Ciudad de la Habana, Mayo de 2011

Año 53 de la Revolución

Declaración de Autoría

Declara ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor

Frank Rafael Pacareu de la Cruz

Firma del Tutor

Frank Torres Rodríguez

Agradecimientos

Agradecer primeramente todo el esfuerzo de mis padres, mi novia Diana y mi tutor porque fueron las personas a las que siempre pude acudir cuando se complicaban las cosas. Muchas gracias por estar ahí en todo momento.

Quisiera resaltar grandemente la posición de mi tutor Frank Torres que estuvo disponible, incluso sábados y domingos para que pudiera trabajar con él en mi tesis, de todo corazón muchas gracias.

A Diana mi novia que me ayudó y no falló cuando la necesité. Ha sido muy importante para mí en este periodo de mi vida.

Existen muchas personas que de una manera u otra pusieron su granito de arena para que pudiera cumplir mi sueño de ser universitario. Mis Amigos Inseparables y hermanos de vida Camilo, Eddy, Vento y mi hermano Javier. Mis socios del aula, el proyecto y el apartamento, Axel, Jose, Jeans, Oslaniel y todos los demás... son muchos, muchos. A mis coleguitas del vicio que los voy a extrañar mucho, el clan del los [F9] mis pupilos Julio y Daniel además del resto de los pésimos Daniel Rivail, Isidro, Nelsito y el clan de los PaZ.

Agradecer también al colectivo de trabajadores del proyecto Plataforma VideoWeb Angel, Albrecht y Eridniel. Al tribunal y el oponente por exigirme y hacer que la investigación quedara lo mejor posible.

A todos, gracias les pido.

Dedicatoria

Tengo mis propios dioses, mis dioses que me guían me encaminan y que también me castigan porque me desean lo mejor. Tengo una luz que me ilumina siempre, no falla, no se detiene, no se resquebraja. Tengo la suerte infinita de tener a mis padres, ellos son todo en lo que pienso cuando a mí alrededor todo se empieza a nublar. Siempre serán mi inspiración y mi meta, siempre serán las personas más importantes en mi vida. A mis padres dedico este trabajo, a ellos que tanto sacrificio y sudor han depositado en mí.

Resumen

El proyecto Plataforma VideoWeb es una aplicación web que permite gestionar archivos multimedia. En esta gestión se realizan procesos fundamentales como son **agregar** una nueva media al sistema y publicarla. Para agregar una nueva media se define a que **tipología** pertenece (ejemplo película, serie entre otras) y así conocer sus **metadatos**. Puede darse el caso de que un archivo multimedia que se quiera adicionar no pertenezca a ninguna de las tipologías existentes. En este caso se puede recurrir a un gestor dinámico que se encuentra implementado en la Plataforma VideoWeb el cual permite crear nuevas tipologías con sus metadatos. Cuando esto sucedía el proceso de **publicación** de estos archivos el cual se realiza haciendo uso de plantillas PHP, se veía afectado producto de que no existían ninguna de las plantillas que se encargan de dicha publicación. Esto implicaba que se tuvieran que construir las mismas directamente en el código lo cual se consideraba una deficiencia del proyecto Plataforma VideoWeb. En el presente trabajo se recoge una investigación hecha con el fin de desarrollar un módulo que permita gestionar plantillas PHP de manera dinámica sin necesidad de ir a la codificación de las mismas.

Para el desarrollo de este componente se utilizó RUP como metodología de desarrollo y Visual Paradigm v6.4 como herramienta CASE para el modelado de los artefactos necesarios. Como Entorno de Desarrollo Integrado se utilizó NetBeans v6.9. El Sistema Gestor de Bases de Datos fue PostgreSQL v8.4 haciendo uso de la herramienta pgAdmin III v1.6.3, el lenguaje de programación en que se desarrolló fue PHP v5.3.2 y Javascript. Como servidor web se utilizó Apache v2.2. Todas herramientas y tecnologías libres.

Con el desarrollo de este módulo se garantizó la construcción de plantillas PHP que permitieron publicar los archivos multimedia del proyecto Plataforma VideoWeb correctamente. Con las pruebas hechas al sistema se pudo corroborar su fortaleza y se concluyó que esta herramienta permite dar solución a la problemática existente en dicho proyecto respecto a la publicación de archivos multimedia según su tipología.

Palabras claves: archivo multimedia, agregar, publicación, tipología, metadatos

Índice

Declaración de Autoría	I
Agradecimientos.....	II
Dedicatoria	III
Resumen.....	IV
Índice	V
Índice de Figuras.....	IX
Índice de Tablas	X
Introducción.....	1
Capítulo 1: “Fundamentación Teórica”	6
1.1. Introducción.....	6
1.2. Conceptos asociados	6
1.2.1. Internet	6
1.2.2. Plataforma Web	7
1.2.3. Sistemas Gestores de Contenido.....	8
1.2.4. CMS Drupal	9
Gestión de usuarios	10
Gestión de contenido.....	10
Plataforma.....	11
Administración y Análisis	11
Características de comunidad	11
Rendimiento y escalabilidad	12
1.3. Editores WYSIWYG disponibles para Drupal	12
1.4. Análisis de los principales editores WYSIWYG para Drupal	12
1.5. Fundamentación de CKEditor como editor a utilizar	16
1.6. Conclusiones del capítulo.....	16
Capítulo 2: “Tendencias y Tecnologías Actuales”	17
2.2. Metodologías.....	17

2.2.1. Metodología de desarrollo de software	17
2.2.2. Fundamentación de RUP como metodología a utilizar	18
2.2.3. Fundamentación de UML como lenguaje de modelado a utilizar	19
2.2.4. Herramientas CASE.....	20
2.2.5. Fundamentación de Visual Paradigm como herramienta CASE a utilizar.....	21
2.3. Tecnologías de desarrollo presentes en los editores WYSIWYG estudiados	21
2.3.1. HTML.....	21
2.3.2. XML y XHTML.....	22
2.3.3. CSS	23
2.3.4. Javascript.....	24
2.3.5. JQuery	24
2.3.6. PHP	25
2.4. Base de Datos.....	27
2.4.1. Sistemas Gestores de Base de Datos.....	28
2.4.2. PostgreSQL y pgAdmin III.....	28
2.5. IDE de desarrollo	29
2.5.1. Fundamentación de NetBeans como IDE de desarrollo a utilizar	30
2.6. Conclusiones del capítulo.....	31
Capítulo 3: “Construcción de la Solución Propuesta”	32
3.1. Introducción.....	32
3.2. Modelo de Dominio	32
3.3. Breve descripción del sistema:.....	33
3.4. Diagrama de Clases del Dominio	34
3.4.1. Descripción de las clases del dominio.....	34
3.5. Captura de Requisitos	35
3.5.1. Requerimientos funcionales	36
3.5.2. Requerimientos no funcionales	36
3.6. Modelo de Sistema.....	37
3.6.1. Definición de casos de uso del sistema	37

3.6.2. Patrón CRUD.....	37
3.6.3. Diagrama de caso de uso del sistema	38
3.6.4. Descripción de los actores	39
3.6.5. Descripción del caso de uso Gestionar Plantilla.....	39
Análisis y Diseño	49
3.7. Análisis.....	49
3.7.1. Modelo de análisis	50
3.7.2. Diagrama de clases del análisis.....	50
3.7.3. Patrón Arquitectónico Modelo Vista Controlador.....	51
3.8. Diseño.....	52
3.8.1. Patrones	52
Patrones GRASP	52
Patrones GoF.....	53
3.8.2. Modelo de diseño.....	55
3.8.3. Diagrama de Clases del Diseño.....	55
3.8.4. Diagrama de Secuencia.....	58
3.9. Modelo de implementación.....	63
3.9.1. Diagrama de Despliegue.....	63
3.9.2. Diagrama de Componentes	64
3.9.3. Conclusiones del Capítulo	67
Capítulo 4: Pruebas al Sistema	68
4.1. Introducción.....	68
4.2. Pruebas de Software.....	68
4.2.1. Objetivos de las pruebas de software	69
4.2.2. Estrategia de pruebas para aplicaciones web	70
4.2.3. Técnicas de Pruebas de Software. Pruebas de Caja Negra.....	70
4.2.4. Casos de pruebas.....	71
4.2.5. Pruebas de Caja Negra. Caso de uso Gestionar Plantilla	72
Sesiones a probar en el caso de uso.....	72

Descripción de las variables	74
Matriz de datos.....	74
Análisis de los resultados obtenidos.....	77
4.2.6. Conclusiones del capítulo	77
Conclusiones Generales	78
Bibliografía Citada	81
Bibliografía Consultada	82
Anexos	85

Índice de Figuras

Figura 1: Esquema del funcionamiento de las páginas PHP.	26
Figura 2: Diagrama de Clases del Dominio	34
Figura 3: Patrón CRUD completo.....	38
Figura 4: Diagrama de Casos de Uso del Sistema.....	39
Figura 5: Diagrama de Clases del Análisis CU Gestiona Plantilla	50
Figura 6: Diagrama de Clases del Diseño.....	56
Figura 7: Diagrama de Clases del Diseño Web. CU Gestionar Plantilla	57
Figura 8: Diagrama de Secuencia. Adicionar Plantilla.....	60
Figura 9: Diagrama de Secuencia. CU Eliminar Plantilla.....	61
Figura 10: Diagrama de Secuencia. CU Editar Plantilla.	62
Figura 11: Diagrama de Despliegue.....	63
Figura 12: Diagrama de Componentes	65
Figura 13: Diagrama de Clases del Análisis. CU Buscar Plantilla.	88
Figura 14: Diagrama de Secuencia. CU Buscar Plantilla.....	89
Figura 15: Diagrama de Clases del Diseño. CU Buscar Plantilla.....	90

Índice de Tablas

Tabla 1: Comparación entre editores WYSIWYG.....	13
Tabla 2: Descripción de Caso de Uso Gestionar Usuario.....	39
Tabla 4: Páginas Clientes y Formularios de la aplicación.....	58
Tabla 5: Sesiones a probar en el caso de uso	72
Tabla 6: Descripción de las variables.....	74
Tabla 7: Matriz de datos. SC 1 Adicionar nueva plantilla.....	74
Tabla 8: Matriz de datos. SC 1 Eliminar plantilla	76
Tabla 9: Matriz de datos. SC 1 Editar plantilla.....	76
Tabla 10: Descripción del Caso de Uso Buscar Plantilla	86

Introducción

Con el actual desarrollo de las tecnologías y el uso de las redes de ordenadores, se ha creado un nuevo espacio donde el hombre moderno puede interactuar entre sí. El internet constituye una de las herramientas más utilizadas en la actualidad por muchas personas en el planeta, debido a que a través de él pueden realizar varias de las actividades de su vida cotidiana. Esto sin lugar a duda crea en la red de redes, el internet, un flujo continuo de información, traído por la interacción que existe entre las personas que acceden a él.

Toda la información que hay en Internet se publica desde las computadoras que conforman esta gigantesca red y lo hacen en gran parte mediante sitios web.

Se puede tomar como definición de sitio web la publicada en el sitio Definicion.org donde se plantea que un sitio web es: “*Colección de páginas web dotada de una dirección web única*” (Definición.org, 2010). Se podría decir entonces que un sitio web es una colección de páginas web que se relacionan entre sí de la misma forma que un libro es un conjunto de páginas de texto. La página web es un documento HTML o XHTML (ver página 21 y 22) los cuales son lenguajes interpretados por aplicaciones tales como: Mozilla Firefox, Internet Explorer y otros. Estas aplicaciones son los llamados “navegadores web” y como su nombre indica son usados para navegar dentro de la red y visitar los distintos sitios web.

Para la construcción de estas páginas web se utiliza en muchos casos código PHP (ver página 25). El PHP es usado para generar de manera dinámica código HTML o XHTML y proporcionarle un grupo de funcionalidades. Una de estas funcionalidades es la posibilidad de que el sitio interactúe con una base de datos donde pueda almacenar información de manera persistente y posibilite la gestión de la misma.

En la actualidad la necesidad de hacer sitios web aumenta. Cada vez es más importante para las empresas tener una localización en la red donde puedan acceder para promocionar sus productos y servicios. Esto ha propiciado el perfeccionamiento de las tecnologías para el desarrollo web, creando aplicaciones para facilitar el trabajo a los desarrolladores. Estas aplicaciones se han ido caracterizando por traer ya implementados procesos y funcionalidades que reducen el tiempo de generación del código. Con la necesidad de crear herramientas que permitan diseñar aplicaciones web de forma más rápida, e incluso sin tener mucho conocimiento del lenguaje HTML comienza a verse en la industria del software los llamados editores web WYSIWYG (**What You See Is What You Get** por sus siglas en inglés) el cual traducido al español significa **lo que ves es lo que obtienes**. Los editores web WYSIWYG permiten que las personas puedan crear sitios web de forma rápida, así podrán publicarlos, sin tener grandes

conocimientos de programación web. Estos editores permiten al desarrollador ir creando su página dinámicamente e ir viendo como quedaría finalmente su diseño; mientras tanto el editor se encarga de generar todo el código HTML que se crea a partir del diseño hecho por el desarrollador. Ejemplo de estos editores WYSIWYG son: Dreamweaver, NVU/Kompozer, las versiones de Composer de Netscape y Mozilla entre otros.

Estos ejemplos constituyen casos de editores web WYSIWYG, pero en realidad el término WYSIWYG se comenzó a usar en el mundo de la informática para referirse a la tecnología de los procesadores de palabras que hacen que lo que está en pantalla corresponda a lo que se imprime en papel. En la actualidad existen aplicaciones que hacen esto como los programas del office de Windows y open office en Linux. En sentido general en ambos casos, los editores web y los editores de texto cumplen el mismo principio del significado de “Editor WYSIWYG”. Ambas tecnologías dan la posibilidad a los usuarios de crear un diseño (tanto web como de un texto) de manera dinámica para realizarle un procesamiento interno y devolver al usuario un producto que corresponda con dicho diseño, o sea, un documento con el código HTML o un documento impreso según sea el caso.

La innovación en los sitios web y la creación de nuevos complementos como los foros y chats unido al fuerte impacto que tuvo la aparición de las redes sociales, trajo consigo que surgieran los llamados editores de campos de texto estilo WYSIWYG. Estos editores constituyen una mezcla de las cualidades de los editores mencionados anteriormente.

En la actualidad con los editores de campo de texto estilo WYSIWYG las personas pueden redactar cualquier mensaje en los campos de texto de los sitios utilizando las funciones de editores WYSIWYG de texto y guardan la información como un documento HTML tal y como lo hacen los editores web WYSIWYG. De esta manera el texto que se redacte podrá cambiar la fuente, tamaño de letra, estilos, poner colores de letra y fondo y muchas otras operaciones y al final será visualizado en una página HTML de la misma manera que se diseñó.

Actualmente existen muchos editores de campo de texto estilo WYSIWYG que son utilizados en diferentes sitios y plataformas web. En el proyecto Plataforma VideoWeb del Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 de la Universidad de las Ciencias Informáticas UCI, se trabaja en una aplicación web que se encarga de proveer a los usuarios el trabajo con medias audiovisuales, o sea, archivos de video y sonido. Este sitio está siendo desarrollado usando el CMS Drupal el cual tiene varios módulos que constituyen editores de campos de texto estilo WYSIWYG. Uno de estos módulos deberá

utilizarse en la presente investigación para dar solución a la problemática que existe en el proyecto Plataforma VideoWeb, respecto a la publicación de los contenidos de los archivos multimedia que se gestionan en dicho sitio.

Como se mencionó anteriormente, a través de la aplicación del proyecto Plataforma VideoWeb se podrá almacenar, gestionar y publicar archivos multimedia. Durante el almacenamiento de las medias uno de los pasos a realizar es el almacenamiento de los datos referentes a dichas medias, los cuales serán usados en la publicación de éstas. Estos datos se pueden conocer o no, todo depende del tipo de media que se quiera almacenar.

Si la media que se desea guardar es de un tipo definido en la aplicación entonces sí se conocerán los datos que se van a almacenar de ella. Para el caso en que se tenga una nueva media para almacenar, que sea distinta tipológicamente a todas las medias almacenadas hasta ese momento, se necesitará crear una nueva tipología de media que recoja los datos que se desean almacenar de la misma.

En la aplicación del proyecto Plataforma VideoWeb, con el objetivo de poder almacenar las medias en cualquiera de los dos casos anteriores, se implementó una funcionalidad la cual consiste en un sistema de gestión dinámico que permite crear nuevas tipologías.

Cada tipología difiere en los datos que almacena y muestra a los usuarios, por tanto la página donde se publican los datos de las medias no podrá ser estándar si las tipologías no lo son. Teniendo en cuenta esto, se determina que cada tipología tendrá un modelo de página web para poder mostrar su información y deberá además cargar de la base de datos las informaciones asociadas a la media.

Si se desea incorporar a la base de datos de medias audiovisuales un nuevo grupo de medias que corresponda con una nueva tipología, entonces no existirá una plantilla de página web que publique sus datos. Esto implica que las personas que deseen hacer uso de la aplicación Plataforma VideoWeb deberán, en estos casos, desarrollar una página web utilizando técnicas informáticas que les permita la construcción de la misma. Esto no es, necesariamente, del conocimiento de los clientes que acceden a dicho sitio y en el caso de que estas personas tengan dicho conocimiento se encontrarán con una deficiencia grande dentro de la aplicación Plataforma VideoWeb.

Lo anteriormente planteado conduce a la formulación del siguiente **problema científico**: ¿Cómo facilitar la construcción de diseños de plantillas para la presentación de los contenidos asociados a las medias que se gestionan en la Plataforma VideoWeb según su tipología?

Para la realización de la investigación se define como **objeto de estudio** los editores web WYSIWYG para la creación de contenido HTML. Como el proyecto VideoWeb del Centro de Geoinformática y Señales Digitales está desarrollado sobre el Sistema de Gestión de Contenidos Drupal se formula como **campo de acción**: Diseño de plantillas para la presentación de los datos almacenados en las tipologías de archivo multimedia haciendo uso de los editores WYSIWYG definidos para Drupal.

La **idea a defender** de la siguiente investigación plantea que: El desarrollo de un editor WYSIWYG que permita la creación de plantillas para las tipologías de la plataforma VideoWeb facilitará la presentación de la información asociada a los archivos multimedia.

Partiendo de esta temática se plantea el siguiente **objetivo general**: El desarrollo de un editor web WYSIWYG para el proyecto Plataforma VideoWeb que permita la presentación de la información asociada a los contenidos audiovisuales según su tipología.

Se trazan como **tareas investigativas**:

1. Analizar las características de los principales editores de páginas estilo WYSIWYG.
2. Seleccionar y argumentar la Metodología de Desarrollo de Software a usar en el proceso.
3. Identificar las funcionalidades que debe brindar el editor.
4. Definir las herramientas y tecnologías a utilizar en la construcción de la solución.
5. Realizar el modelado de los artefactos necesarios para la implementación del editor.
6. Implementar el editor.
7. Realizar las pruebas al editor.
8. Realizar manual de instalación y uso del editor.

Durante el desarrollo del trabajo se utilizarán varios **métodos de investigación**, tales como la **observación** y **experimentación** dentro de los **métodos empíricos** y como **métodos teóricos** están el **histórico-lógico** y **analítico-sintético**.

El análisis Histórico-lógico permitirá conocer el desarrollo actual y las nuevas tendencias en los editores WYSIWYG. La observación y experimentación serán útiles para caracterizar los editores con el fin de determinar que editor se va a utilizar, además de permitir que se comprueben y evalúen los resultados alcanzados.

El análisis analítico-sintético permitirá dividir el problema general en partes más pequeñas para su análisis más detallado y luego integrarlas; esto será útil para definir de qué manera se puede dar solución al problema con el editor seleccionado.

En el presente trabajo de diploma se organizan los contenidos en cuatro capítulos:

Capítulo 1 - Fundamentación Teórica: Se hace un estudio de un grupo de conceptos asociados a la investigación con el fin de llegar al objeto de estudio definido en la misma. Se describen las nuevas tendencias en las tecnologías que constituyen editores de campo de texto estilo WYSIWYG disponibles para Drupal.

Capítulo 2 – Tendencia y Tecnologías Actuales: En este capítulo se define y justifica la metodología, las herramientas y otras tecnologías que serán utilizadas en la presente investigación.

Capítulo 3 – Construcción de la Solución Propuesta: En este capítulo se estudia el entorno en que se desarrolla la problemática de la investigación y se definen los requerimientos a implementar. Además se realizan los modelos de sistema, análisis, diseño e implementación.

Capítulo 4 – Pruebas al Sistema:

En este capítulo se realiza un estudio sobre las pruebas del software con el fin de conocer la importancia y los objetivos de las mismas. Además se hace un análisis de los resultados de las pruebas hechas a la aplicación.

Capítulo 1: “Fundamentación Teórica”

1.1. Introducción

En este capítulo se hace un análisis de los diferentes editores web WYSIWYG que hay disponibles para el CMS Drupal. Con este fin se hace necesario un previo estudio de algunos temas relacionados que faciliten el entendimiento de los contenidos. Temáticas tales como: Inicios y desarrollo del internet, aparición de las plataformas web y el desarrollo de los sistemas gestores de contenidos sirven como preámbulo para analizar las características del CMS Drupal. Una vez culminada dicha caracterización se podrán analizar algunos editores web WYSIWYG que hay disponibles para el CMS Drupal y de esta manera realizar una fundamentación teórica para la presente investigación

1.2. Conceptos asociados

1.2.1. Internet

El concepto de redes interconectadas, o internet, tiene sus orígenes en el año 1969 con la búsqueda, por parte del ejército de los Estados Unidos, de una manera de conectar centros de investigación sobre una red de conmutación de paquetes, así la ARPANET original evolucionó hacia Internet. Internet se basó en la idea de que habría múltiples redes independientes, de diseño casi arbitrario, empezando por ARPANET como la red pionera de conmutación de paquetes, pero que pronto incluiría redes de paquetes por satélite, redes de paquetes por radio y otros tipos de red. Internet como ahora se conoce encierra una idea técnica clave, la de arquitectura abierta de trabajo en red. Bajo este enfoque, la elección de cualquier tecnología de red individual no respondería a una arquitectura específica de red sino que podría ser seleccionada libremente por un proveedor e interactuar con las otras redes a través del metanivel de la arquitectura de Internetworking (trabajo entre redes). (Barry M. Leiner, 2003)

En la década de los 70 el proyecto DARPA del Departamento de Defensa de los Estados Unidos desarrolló el modelo multicapa de protocolos de redes el cual evolucionó en el internet actual. La Organización de Estándares Internacionales (International Standards Organization, ISO) posteriormente aumentó los protocolos de comunicación, los cuales evolucionaron en el modelo OSI (Open Systems Interface). Internet no se adhiere completamente al modelo OSI pero sus principios son similares. (Austerberry, 2003)

Se puede decir entonces que Internet es un conjunto de redes de ordenadores que se comunican e intercambian información en forma de paquetes haciendo uso de protocolos y estándares. Con la aparición del protocolo TCP/IP (Transfer Control Protocol/Internet Protocol, Protocolo para el Control de Transferencia/Protocolo de Internet) el 1ro de enero de 1983 se garantizó que las redes físicas heterogéneas que componen el internet funcionen como una red lógica única, de alcance mundial.

Así, en 1985, Internet estaba firmemente establecida como una tecnología que ayudaba a una amplia comunidad de investigadores y desarrolladores, y empezaba a ser empleada por otros grupos en sus comunicaciones diarias entre ordenadores. Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW, o "la Web"), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Ésta fue desarrollada en 1989 por Tim Berners Lee y utiliza Internet como medio de transmisión. Otro de los servicios que se desarrollaron con el internet es el correo electrónico a través del protocolo SMTP. La interconexión entre los diversos sistemas de correo demostraba la utilidad de las comunicaciones electrónicas entre personas.

Existen, muchos otros servicios y protocolos en Internet, aparte de la Web y el envío de correo electrónico, la transmisión de archivos (FTP y P2P), las conversaciones en línea (IRC), la mensajería instantánea, la transmisión de contenido y comunicación multimedia-telefonía (VoIP), televisión (IPTV), los boletines electrónicos (NNTP), el acceso remoto a otras máquinas (SSH y Telnet) o los juegos en línea.

Internet desde su creación en 1989 hasta la fecha ha ido variando y se ha convertido en una herramienta de uso diario para muchas personas las cuales utilizan el mismo para visitar los miles de sitios web que se encuentran publicados en él. Con su uso aquellos que navegan en internet han encontrado nuevas formas de relacionarse, comerciar, entre otras actividades influyendo en la sociedad de manera, económica, política, social, cultural entre otras.

1.2.2. Plataforma Web

Desde el punto de vista de compatibilidad, internet es usado por cualquier ordenador independientemente del sistema operativo (SO) que tenga. Haciendo uso de esta compatibilidad se trabajó sobre la idea de crear un grupo de servicios que fueran brindados a través de internet y que pudieran ser consumidos desde cualquier ordenador en la red sin depender del SO que use. Las plataformas web fueron creadas con el fin de brindar este tipo de servicios.

Para definir el concepto de plataforma web se toma en el presente trabajo las ideas planteadas por el grupo de autores de

Una plataforma Web es un sistema formado por un conjunto de componentes hardware y software que proporcionan capacidades (servicios) sobre las que se deberá apoyar cualquier aplicación software y cuyo funcionamiento es a través de internet. (Aportela Rodríguez, 2007)

El resultado final debe ser un conjunto (plataforma) formado por componentes horizontales o herramientas de colaboración (gestor de usuarios, gestores de contenido y gestores documentales) que pueden reutilizarse en las distintas aplicaciones específicas que se introduzcan. (Sarduy Domínguez, 2006.)

Haciendo un análisis de los conceptos anteriores, se puede concluir diciendo que una plataforma web es un conjunto de dispositivos de hardware y software que funcionan entre sí para brindar un grupo de servicios a través de la red por lo que independientemente del sistema operativo en que se trabaje se va a poder hacer uso del mismo. O sea cuando se desarrolla un programa, se hace para que funcione en una plataforma específica. Esto se refiere a que el programa va a funcionar en un determinado sistema operativo, normalmente hay versiones para Windows, MacOs, Linux, etc. Pero cuando se refiere a una plataforma web, quiere decir que el sistema funcionará desde un navegador web, independientemente del SO que se esté utilizando.

Esto tiene la ventaja de que no es necesario crear el mismo software para cada tipo de SO. Pero tiene desventajas en el sentido de que no puede ser tan potente como un sistema que funciona usando los recursos del SO.

Un ejemplo de lo anterior, son los clientes de correo. Programas como Outlook, Eudora, Firebird, etc. son clientes de correo que funcionan de manera local, o sea, necesitan del SO para funcionar. Esto los convierte en aplicaciones potentes con muchas opciones y capacidades. Por el otro lado, están los WebMail como Gmail, Hotmail, etc. los cuales son aplicaciones que funcionan en una plataforma web y que no dependen del SO de la computadora para funcionar. Esto los vuelve más versátiles y universales, aunque muestran una disminución en las características que pueden ofrecer a los usuarios si se les compara con los sistemas locales antes mencionados.

1.2.3. Sistemas Gestores de Contenido

Con las plataformas web entre otras cosas se pudo impulsar el desarrollo de los sitios web y cada vez era más necesario para toda empresa, entidad o pequeño negocio tener una representación en la web. Esto

unido al poco conocimiento por parte de muchas personas respecto a la programación web trajo consigo que se crearan herramientas que facilitaran este proceso de realización de sitios web, los CMS (por sus siglas en inglés Content Management System lo cual traducido al español significa Sistema Gestor de Contenido) son un ejemplo de ellas.

La gestión de contenidos proviene del término en inglés Content Management (CM), que es una expresión que se asocia a un nuevo método para el diseño y desarrollo de portales Web que conlleva:

- La inclusión de elementos digitales de diferentes tipos (textuales, gráficos y sonoros).
- El desarrollo de forma cooperativa y descentralizada.
- El paso de un modelo estático a otro mucho más dinámico. La reutilización de los contenidos.

Los CMS identifican un conjunto de programas informáticos destinados a gestionar la presentación de los contenidos de una sede Web. Estos productos requieren un sistema de gestión de bases de datos vía Web, un servidor Web y un lenguaje de programación de las páginas. (Caraballo Pérez, 2007)

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

Una herramienta CMS generalmente contendrá una interfaz basada en formularios, a los que habitualmente se accede con el navegador, donde se pueden dar de alta a los contenidos fácilmente. Esos contenidos luego aparecerán en la página en los lugares donde se ha indicado al darlos de alta. Por lo tanto, un CMS estará compuesto de dos partes, un back y un front, siendo el back la parte donde los administradores publican las informaciones y el front la parte donde los visitantes visualizan las mismas. Como ejemplos de CMS están: Drupal, Joomla, Vignette, OsCommerce, entre otros.

1.2.4. CMS Drupal

Como se planteó en la introducción de la investigación en el proyecto Plataforma VideoWeb se está trabajando en una plataforma web que se encarga en sentido general del trabajo con medias audiovisuales. Esta plataforma está siendo desarrollada haciendo uso del CMS Drupal.

CMS Drupal es un Sistema de Administración de Contenidos muy potente que se puede utilizar para el desarrollo de sitios web. En el sitio oficial de Drupal (Drupal, 2010) se mencionan varias características

tanto generales como más particulares respecto a su gestión de contenido, gestión de usuarios, etc. alguna de ellas son:

- Ayuda on-line: Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.
- Búsqueda: Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- Código abierto: El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Módulos: La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante jabber, mensajes privados, bookmarks, etc.
- Personalización: Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- URLs amigables: Drupal usa el mod_rewrite de Apache para crear URLs (ver Anexos – glosario de términos) que son manejables por los usuarios y los motores de búsqueda.

Gestión de usuarios

Drupal cuenta además con una gestión de usuarios potente que se basa en la autenticación de usuarios y en permisos de usuarios basados en roles. De esta manera los usuarios pueden autenticarse y ocupar un rol, mientras los administradores pueden asignar permisos a estos roles.

Gestión de contenido

Drupal garantiza que el contenido se encuentre bien gestionado debido a que presenta:

- Control de versiones: Todo el contenido que se cree o modifique va a tener la fecha, nombre del usuario además de otros aspectos de quien ha cambiado tal contenido.
- Enlaces permanentes: Todo contenido en Drupal va a tener un enlace permanente que va a permitir ser localizado de manera directa.
- Contenidos basados en Nodos: Cada contenido en Drupal constituye un nodo. Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de

diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto.

- Plantillas: El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

Plataforma

- Independencia de la base de datos: Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL (ver Anexos glosario de términos), existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.
- Multiplataforma: Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- Múltiples idiomas y Localización: Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como GNU ettext

Administración y Análisis

- Administración vía Web: La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.
- Análisis, Seguimiento y Estadísticas: Drupal puede mostrar en las páginas web de administración informes sobre enlaces primarios, popularidad del contenido o de cómo los usuarios navegan por el sitio.
- Registros e Informes: Toda la actividad y los sucesos del sistema son capturados en un 'registro de eventos', que puede ser visualizado por un administrador.

Características de comunidad

- Comentarios enlazados: Proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.
- Encuestas: Incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.

- Foros de discusión: Incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.
- Libro Colaborativo: Esta característica es única de Drupal y permite crear un proyecto o "libro" a ser escrito y que otros usuarios contribuyan contenido. El contenido se organiza en páginas cómodamente navegables.

Rendimiento y escalabilidad

- Control de congestión: Incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.
- Sistema de Cache: El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

1.3. Editores WYSIWYG disponibles para Drupal

En la actualidad Drupal.org (el sitio oficial de Drupal) tiene más de 100 módulos para agregarle funcionalidades al mismo después de la instalación. Alrededor de una docena de ellos son editores WYSIWYG (What You See Is What You Get) que es el término que se usa en informática para describir un sistema en el que, el desarrollador va editando un contenido y mientras lo hace este sistema le va mostrando un contenido muy similar al que va a obtener al final del trabajo. Estos editores WYSIWYG son utilizados en campos de texto y modifican estas áreas para dar varios formatos al texto que se edite en él. El núcleo de Drupal no cuenta actualmente con un editor. Entre los editores WYSIWYG que hay para Drupal se encuentran: BUEditor, FCKEditor, CKEditor, TinyMCE, XStandard, NiceEdit, widgEditor entre otros.

1.4. Análisis de los principales editores WYSIWYG para Drupal

Como se planteó anteriormente existen varios módulos disponibles para Drupal que constituyen editores WYSIWYG de campo de texto. Para dar solución a la problemática planteada en la presente investigación se pretende tomar alguno de éstos y modificarlo.

Con el fin de determinar con qué editor se trabajará en esta investigación se realizó una encuesta al grupo de trabajo del sitio drupaleros en la Universidad de las Ciencias Informáticas, además de consultar el sitio oficial de Drupal. A partir de este estudio se eligieron los siguientes editores en sus respectivas versiones:

BUEditor6.x-1.3, CKEditor6.x-1.2, TinyMCE6.x-1.1-dev y widgEditor6.x-1.0 debido a que estos editores WYSIWYG presentan las mejores prestaciones y son los más utilizados.

A continuación se muestra una tabla donde se hace una comparación entre estos editores con el objetivo final de definir cuál de ellos será utilizado en la presente investigación.

Tabla 1: Comparación entre editores WYSIWYG

Aspectos	BUEditor6.x-1.3	CKEditor6.x-1.2	TinyMCE6.x-1.1-dev	widgEditor6.x-1.0
Instalación	Fácil: No se necesita ir a la página del editor para descargar un nuevo fichero, se ejecuta al ser activado dentro de los módulos de Drupal.	Complicada: Sí se necesita ir a la página del editor para descargar un nuevo fichero, además requiere de algunas modificaciones en su configuración así como la creación de perfiles para el uso de la herramienta.	Complicada: Sí se necesita ir a la página del editor para descargar un nuevo fichero y se debe configurar un perfil que permita utilizar el editor, todo esto requiere modificaciones complicadas.	Fácil: No se necesita ir a la página del editor para descargar un nuevo fichero, se ejecuta al ser activado dentro de los módulos de Drupal.
Funcionalidades	<ol style="list-style-type: none"> 1. Negritas 2. Cursiva 3. Encabezados 4. Listar con números 5. Listar sin números 6. Dar un break 7. Insertar 	<ol style="list-style-type: none"> 1. Negritas 2. Cursiva 3. Subrayado 4. Tachado 5. Encabezados 6. Listar con números 7. Listar sin números 8. Dar un break 9. Insertar 	<ol style="list-style-type: none"> 1. Negritas 2. Cursiva 3. Subrayado 4. Tachado 5. Encabezados 6. Listar con números 7. Listar sin números 8. Dar un break 9. Insertar imagen 10. Insertar vínculo 11. Ver código HTML 	<ol style="list-style-type: none"> 1. Negritas 2. Cursiva 3. Encabezados 4. Listar con números 5. Listar sin números 6. Insertar imagen 7. Insertar vínculo 8. Ver código HTML

	<p>imagen</p> <p>8. Insertar vínculo</p> <p>9. Vista predeterminada</p> <p>10. Ayuda</p>	<p>imagen</p> <p>10. Insertar vínculo</p> <p>11. Ver código HTML</p> <p>12. Insertar Anclaje</p> <p>13. Buscar palabra</p> <p>14. Eliminar formato</p> <p>15. Insertar Símbolo</p> <p>16. Insertar flash</p> <p>17. Insertar Smiley</p> <p>18. Maximizar el editor en el navegador</p> <p>Y otras 10 funcionalidades menos significativas.</p>	<p>12. Insertar Anclaje</p> <p>13. Buscar palabra</p> <p>14. Eliminar formato</p> <p>15. Insertar Símbolo</p> <p>16. Insertar flash</p>	
Tecnologías de desarrollo	<ol style="list-style-type: none"> 1. JavaScript 2. Php 3. Jquery 4. HTML 	<ol style="list-style-type: none"> 1. JavaScript 2. Php 3. Css 4. HTML 5. Jquery 	<ol style="list-style-type: none"> 1. JavaScript 2. Php 3. Css 4. HTML 5. Jquery 	<ol style="list-style-type: none"> 1. JavaScript 2. Php 3. Css 4. Jquery 5. HTML
Observaciones	<ul style="list-style-type: none"> • Interfaz sencilla. • Los botones 	<ul style="list-style-type: none"> • Interfaz atractiva. • El revisor de 	<ul style="list-style-type: none"> • Interfaz atractiva • Muy complicado de instalar 	<ul style="list-style-type: none"> • Interfaz sencilla. • Se trabaja de manera dinámica en un

	<p>agregan en la región editable las etiquetas correspondientes a la opción que se ha solicitado, o sea el usuario trabaja en HTML prácticamente.</p> <ul style="list-style-type: none"> • La funcionalidad de insertar imagen no funciona. • La ayuda consiste en mencionar la función de cada botón (poco significativo) 	<p>ortografía necesita descargar el idioma.</p> <ul style="list-style-type: none"> • El resto de las funcionalidades operaron correctamente. 	<ul style="list-style-type: none"> • Operan bien la gran mayoría de sus funcionalidades. En el caso de cargar imagen y flash no se obtuvo un buen resultado al intentar trabajar con estas funcionalidades, quizás por la complejidad que conlleva su instalación. • Su estructura interna muy complicada y no se encontró buena documentación para conocer más sobre su funcionamiento interno. 	<p>entorno sin código HTML el cual puede ser visto eficientemente si se da en esa opción.</p> <ul style="list-style-type: none"> • La funcionalidad de insertar imagen no funciona.
--	--	---	--	--

Como característica general de los editores estudiados, se puede decir que poseen control de versiones, autocompletado de código, resaltado de sintaxis, soporte de lenguajes de servidor, depuración y documentación en línea. Por lo que dependiendo del entorno en el que se trabaje y el trabajo específico que se realice, se dará valor a su uso.

1.5. Fundamentación de CKEditor como editor a utilizar

Teniendo en cuenta la comparación realizada entre dichos editores, sus características y la problemática existente en el proyecto Plataforma VideoWeb del Centro GEYSED se ha determinado como herramienta a utilizar el editor CKEditor en la versión antes mencionada para la realización de la presente investigación. Esta elección fue hecha atendiendo a que este editor presenta, respecto al resto de los editores, un mejor equilibrio entre complejidad interna y potencialidad de sus funcionalidades. Primeramente, plantear que son incomparables las potencialidades del CKEditor respecto al widgEditor y el BUEditor por tanto queda descartada la opción de utilizar uno de estos dos editores para el presente trabajo. Por otra parte las operaciones que se pueden hacer con este editor son variadas, útiles y trabajan correctamente, características que comparte parcialmente con el TinyMC, debido a que este último presentó problemas con algunas funcionalidades durante las pruebas que se le hicieron (ver Tabla 1 página 13).

En sentido general el CKEditor no solo supera al resto en cuanto a funcionalidades sino que además se pudo comprobar que presenta una estructura interna sencilla si se tiene en cuenta todas las operaciones que permite hacer. Además CKEditor cuenta con un sitio oficial donde se publican ayudas, ejemplos de códigos y otras utilidades que servirán como ayuda para el desarrollo de esta investigación.

1.6. Conclusiones del capítulo

Con los estudios realizados en este capítulo se pudo conocer acerca de Internet, los sitios y plataformas web así como de los CMS. El conocimiento visto en estas actividades sirvió de base y punto de partida para el estudio de las herramientas que se van a utilizar en ésta investigación. De esta manera se pudo conocer los principales aspectos del CMS Drupal y más específicamente de los módulos que hay para este CMS. que constituyen editores de campo de texto estilo WYSIWYG. Como se planteó en la introducción del presente documento, estos editores WYSIWYG constituyen el objeto de estudio de este trabajo, por lo que a partir de su estudio, se llegó a la conclusión de que el CKEditor es el editor que reúne las mejores características y que será por tanto el editor a utilizar en la presente investigación.

Capítulo 2: “Tendencias y Tecnologías Actuales”.

2.1. Introducción:

En este capítulo se define la metodología de desarrollo de software a utilizar en la investigación. Además se hace un análisis de las principales tecnologías que son necesarias para desarrollar los editores web WYSIWYG estudiados en el capítulo anterior. Esto es necesario para entender mejor su funcionamiento.

2.2. Metodologías

Metodología es una palabra compuesta por tres vocablos griegos *metá* (“más allá”), *odós* (“camino”) y *logo* (“estudio”). El concepto hace referencia a los métodos de investigación que permiten lograr ciertos objetivos en una ciencia. (...). Por lo tanto, la metodología es el conjunto de métodos que rigen una investigación científica o una exposición doctrinaria. (Definición.de, 2011)

Una metodología es el conjunto de métodos por los cuales se regirá una investigación científica y un método es el procedimiento que se llevará a cabo en orden a la consecución de determinados objetivos.

Desde el punto de vista de gestión las metodologías implican ventajas por:

- Facilitar la tarea de planificación.
- Facilitar la tarea de control y seguimiento de un proyecto.
- Mejorar la relación coste/beneficio.
- Optimizar el uso de recursos disponibles.
- Facilitar la evaluación de resultados y cumplimiento de los objetivos.
- Facilitar la comunicación efectiva entre usuarios y desarrolladores.
- Ayudar a la gestión del proyecto.

2.2.1. Metodología de desarrollo de software

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Por tanto los métodos dentro de este tipo de metodología serían las técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas)

actividades del proceso de desarrollo. Por tanto en un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

El uso de alguna metodología dentro del desarrollo de software proporciona ventajas tales como:

- Determinar las fases dentro del ciclo de vida especificando su orden de ejecución.
- Definir los resultados intermedios y finales.
- Proporcionar un conjunto de métodos, herramientas y técnicas para facilitar la tarea del ingeniero del software y aumentar su productividad.

En la actualidad existen varias metodologías Orientada a Objeto, basadas en UML (por sus siglas en ingles Unified Modeling Language lo que significa en español Lenguaje Unificado de Modelado): Rational Unified Process (RUP), OPEN y MÉTRICA 3, entre otras.

2.2.2. Fundamentación de RUP como metodología a utilizar

Como metodología para el desarrollo de software se propone RUP por sus siglas en ingles Rational Unified Process lo cual traducido al español significa Proceso Unificado de Rational. RUP ofrece una gran cantidad de ventajas; además el grupo de desarrollo cuenta con suficiente experiencia en el uso de esta metodología, cualidad que dará rapidez al desarrollo del producto.

RUP define para cada etapa: el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. Su meta es asegurar la producción de software con la más alta calidad, que cumpla con las necesidades de los usuarios dentro del cronograma planeado y la inversión prevista.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Además, el Proceso Unificado utiliza el Lenguaje Unificado de Modelado para expresar gráficamente todos los esquemas de un sistema software. Los aspectos más importantes que definen este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. (Ivar Jacobson, 2000)

RUP se divide en 4 fases:

- Inicio: El objetivo de esta etapa es determinar la visión del proyecto.
- Elaboración: El objetivo es determinar la estructura óptima.
- Construcción: El objetivo es obtener la capacidad operacional inicial.

- Transición: El objetivo es obtener la primera versión del proyecto (release).

Dentro de las ventajas que ofrece RUP se encuentra que es una metodología que brinda un desarrollo de software bien documentado lo que permite un futuro entendimiento de cada componente del producto. Esto es de vital importancia para el proyecto Plataforma VideoWeb porque dicha documentación servirá de bibliografía a consultar para la realización de nuevas versiones del producto.

Cada año varía el grupo de trabajo del proyecto debido a problemas internos de funcionamientos muy particulares del marco en que se desenvuelven las actividades productivas en el proyecto Plataforma VideoWeb. Estos problemas se dan principalmente con los estudiantes que se encuentran vinculados al proyecto desempeñando algún rol y que se van del centro o pasan a otros proyectos. Están además los nuevos integrantes que cada año se incorporan a las filas del grupo de trabajo de este proyecto. Todos estos cambios implican que siempre sea necesario tener una buena documentación referente a las actividades de desarrollo que se realizan en el proyecto por lo que es necesario de que sea documentado todo el proceso de desarrollo de esta investigación.

Otra ventaja de RUP es que propone varias iteraciones en el proceso de desarrollo de software posibilitando la corrección de errores y mejoras del software a medida que avanza el proyecto, esto ayuda a que al final del mismo se tenga un producto con buena calidad.

2.2.3. Fundamentación de UML como lenguaje de modelado a utilizar

En el epígrafe anterior se resaltaron las principales características de RUP y dentro de ellas se comentó que el UML (UML por sus siglas en inglés Unified Modeling Language lo cual significa en español Lenguaje Unificado de Modelado) se utiliza para expresar gráficamente todos los esquemas de un sistema software.

De forma más concreta se puede plantear que el UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. (Schmuller, 2000)

Otras de las características que definen al UML y que representan cualidades de gran peso por las cuales se decidió hacer uso de este lenguaje de modelado en la presente investigación son:

- UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.
- El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.
- Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

2.2.4. Herramientas CASE

Las Herramientas CASE (por su nombre en inglés Computer-Aided Software Engineering lo que traducido al español significa Ingeniería de Software Asistida por Ordenador) son básicamente diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. (Together, 2005)

Estas herramientas posibilitan la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas al comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación. Representan una forma que permite el modelado de procesos de negocios y el desarrollo de sistemas. (Perissé, 2005)

Las herramientas CASE suelen incluir:

- Un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos.
- Herramientas de diseño para dar apoyo al análisis de datos.
- Herramientas que permitan desarrollar el modelo de datos corporativo, así como los esquemas conceptual y lógico.
- Herramientas para desarrollar los prototipos de las aplicaciones.

A través de las herramientas CASE se brindará un conjunto de métodos, utilidades y técnicas que facilitaran la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o

en alguna de sus fases de proceso. Algunos ejemplos de herramientas CASE son: Oracle Designer, EasyCASE, Rational Rose Enterprise, Visual Paradigm entre otros.

2.2.5. Fundamentación de Visual Paradigm como herramienta CASE a utilizar.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (International, 2007)

Se escoge Visual Paradigma 6.4 para el modelado del sistema no solo por las facilidades que brinda para el diseño UML del ciclo de vida del proceso de desarrollo de software. Visual Paradigma 6.4 mejora el tiempo de construcción de aplicaciones, posibilitando el modelado de todo tipo de diagramas de clases los cuales van a ser necesarios para la documentación de la investigación.

Esta herramienta CASE soporta PHP y facilita la codificación desde diagramas, esto va a ser positivo durante la elaboración de la solución de la investigación porque PHP es una de las herramientas a utilizar. Posee una amplia bibliografía tanto en materiales audiovisuales como documentación digital. El grupo de trabajo del proyecto Plataforma VideoWeb tiene años de experiencia en la utilización de esta herramienta. Este aspecto influirá positivamente durante el desarrollo del modelado de los diagramas necesarios en esta investigación.

2.3. Tecnologías de desarrollo presentes en los editores WYSIWYG estudiados

Teniendo en cuenta los resultados obtenidos en la Tabla 1, los editores que han sido escogidos para el desarrollo de la presente investigación se han implementado haciendo uso de HTML, PHP y Javascript con la librería JQuery aunque en su mayoría se utilizaron otras tecnologías como CSS. Por tanto se hace necesario profundizar en algunas características de estas tecnologías.

2.3.1. HTML

HTML (HyperText Markup Language por sus siglas en inglés el cual traducido al español significa Lenguaje de Marcado de Hipertexto) es el lenguaje con el que se definen las páginas web. A criterio de

Javier Eguíluz Pérez una definición de HTML podría ser: “HTML es lo que se utiliza para crear todas las páginas web de Internet”. Más concretamente, HTML es el lenguaje con el que se “escriben” la mayoría de páginas web.” (Pérez, 2008)

Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que componen una página web. El HTML se creó en un principio con objetivos divulgativos de información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de personas que lo utilizarían en un futuro. (Alvarez, 2001)

Sin embargo, pese a esta deficiente planificación, se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML: HTML 3.0, HTML 3.2, HTML 4.0, y otros.

Además hay tecnologías como XHTML, CSS, Javascript, PHP entre otras que complementan el funcionamiento de los sitios web en la red minimizando las carencias del HTML.

2.3.2. XML y XHTML

A partir de las deficiencias del HTML se trabajó sobre la idea de buscar un lenguaje que fuera interpretado por los navegadores y que lo sustituyera. Surge así el XHTML (por sus siglas en inglés eXtensible Hypertext Markup Language lo cual significa en español Lenguaje Extensible de Marcado de Hipertexto). El XHTML no es más que la versión XML del HTML.

El XML (Extended Markup Language por sus siglas en inglés las cuales significan en español Lenguaje de marcado Extensible) proviene de un lenguaje que inventó IBM por los años 1970. El lenguaje de IBM se llama GML (por sus siglas en inglés General Markup Language lo cual traducido al español significa Lenguaje de Marcado General) y surgió por la necesidad que tenían en la empresa de almacenar grandes cantidades de información de temas diversos.

En el 1998 se comenzó el desarrollo del XML con el cual se pretendían solucionar las carencias del HTML en lo que respecta el tratamiento de la información en problemas tales como:

- El contenido se mezcla con los estilos que se le quieren aplicar.
- No permite compartir información con todos los dispositivos, como pueden ser ordenadores o teléfonos móviles.
- La presentación en pantalla depende del visor que se utilice. (Alvarez, 2001)

XML es un lenguaje que funciona por etiquetas, igual que HTML, pero cuyas reglas de creación de documentos son mucho más estrictas que las del propio lenguaje HTML, lo que da pie a la posibilidad de procesarlos automáticamente por programas informáticos. Ese pequeño detalle, realmente es la base para toda una serie de ventajas que hacen que XML sea una herramienta ideal para el momento actual, donde la información circula sin límite por las redes globales y XHTML intenta aprovecharse de esto en aras de conseguir una normalización de los estándares.

EL código XHTML trae muchas ventajas por mencionar algunas.

- Puede ser procesado por muchos programas informáticos igual que ocurre con el XML
- Permite que los navegadores no tengan tantos problemas para interpretar lo que el desarrollador ha querido escribir y tengan entonces que solucionar los posibles errores de código cometidos, como ocurría con HTML. (Alvarez, 2011)

Por tanto, como se plantea anteriormente XHTML es la versión XML de HTML. Desde el punto de vista del desarrollador XHTML hereda la rigidez de XML con lo cual no se puede escribir documentos XHTML de cualquier manera, como ocurría con HTML, sino atendiendo a unas normas. Ese detalle, que en principio pueda parecer una limitación, en realidad tiene muchas ventajas que hace del XHTML una potente herramienta.

2.3.3. CSS

El CSS (por sus siglas en inglés Cascading Style Sheets las cuales traducidas al español significan Hojas de Estilo en Cascada) surgió un tiempo después que el HTML por el hecho de que este último no cubría todas las necesidades para crear sitios web. La problemática principal era de que a raíz del desarrollo web se incrementaba la necesidad de crear sitios cada vez más complejos.

En el caso particular de las Hojas de Estilo en Cascada se puede tomar como referencia a la definición hecha por Javier Eguíluz Pérez en su libro “Introducción a CSS” donde plantea que: “CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.” (Pérez, 2009)

En sentido general CSS es un código que se escribe dentro del código HTML de la página web o se pueden escribir en un archivo aparte y enlazar la página con ese archivo.

CSS, es una tecnología que permite crear páginas web de una manera más exacta. Con esta tecnología se hacen muchas cosas que no se podían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, entre otros. (Lazaro, 2009)

Una de las características más potentes de la programación con hojas de estilo consiste en definir los estilos de todo un sitio web. Esto se consigue enlazando todas las páginas del sitio con un archivo que se debe crear y en el cual se colocará las declaraciones de estilos de la página. De este modo, todas las páginas comparten una misma declaración de estilos y, por tanto, si se cambia dicha declaración, cambiarán todas las páginas con ese estilo.

2.3.4. Javascript

EL desarrollo web seguía creciendo y ya por los años 1990 con el HTML, CSS y otras tecnologías se podían hacer páginas estáticas con estilos pero no se lograba aun satisfacer necesidades mayores como que la página interactuara con el usuario. Nació entonces el Javascript, y así los autores de sitio web pudieron crear páginas que permitieran intercambiar con los usuarios para construir webs de mayor complejidad.

El desarrollo del Javascript comenzó en 1990 de manera muy sencilla pero efectiva, esto mismo propició una gran aceptación en el mundo del desarrollo web. En 1996 se comienza con su estandarización en conjunto con ECMA y en el 1997 la SIO asume dicha estandarización. El Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se pueden crear diferentes efectos e interactuar con los usuarios. (Sitio Oficial de Javascript, 2007)

En sentido general este es un lenguaje donde gran parte de la programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, utilización de teclas, cargas de páginas entre otros. Se puede encontrar o crear códigos javascript para insertarlos en las páginas como: reloj, contadores de visitas, fechas, calculadoras, validadores de formularios, detectores de navegadores e idiomas. El Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

2.3.5. JQuery

JQuery es un framework para el lenguaje Javascript con el fin de estandarizar dicho lenguaje en los varios navegadores que existen. Cuando un desarrollador utiliza Javascript, generalmente tiene que hacer scripts compatibles con varios navegadores y para ello necesita incorporar mucho código que lo único que hace es detectar el navegador del usuario. Esto es necesario para definir un conjunto de operaciones en dependencia a si dicho navegador es Internet Explorer, Firefox, Opera u otro. jQuery ayuda en este sentido puesto que implementa una serie de clases (de programación orientada a objetos) que permiten programar sin importar el navegador con el que está visitando el usuario. (Alvarez, 2009)

Este framework ofrece una infraestructura con la que existe mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery se puede obtener ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando se programa Javascript con jQuery se obtiene una interfaz que funcionará para todos los visitantes sin importar el navegador que usen. Simplemente se debe conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para lograr esto. (Alvarez, 2009)

Además, todas estas ventajas con jQuery se obtienen de manera gratuita, porque el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Para ello simplemente se tiene que incluir en las páginas web un script Javascript que contiene el código de jQuery. Este script que se puede descargar de la propia página web del producto y así comenzar a utilizar el framework.

JQuery no es el único framework que existe en el mercado. Existen varias soluciones similares que también funcionan muy bien y que básicamente sirven para hacer lo mismo. Como es normal, cada uno de los frameworks tiene sus ventajas e inconvenientes, pero jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otro aspecto muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.

2.3.6. PHP

EL PHP (por sus siglas en inglés) PHP Hypertext Pre-processor lo cual llevado al español significa PHP Pre-procesador de Hipertexto es un lenguaje que se encarga de generar el código HTML que recibirá el

usuario luego de haber ejecutado alguna acción en un sitio web. Se toma la definición de PHP de FAQ extraído de la página PHP.net donde se plantea que el PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

El PHP es un lenguaje del lado del servidor porque se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

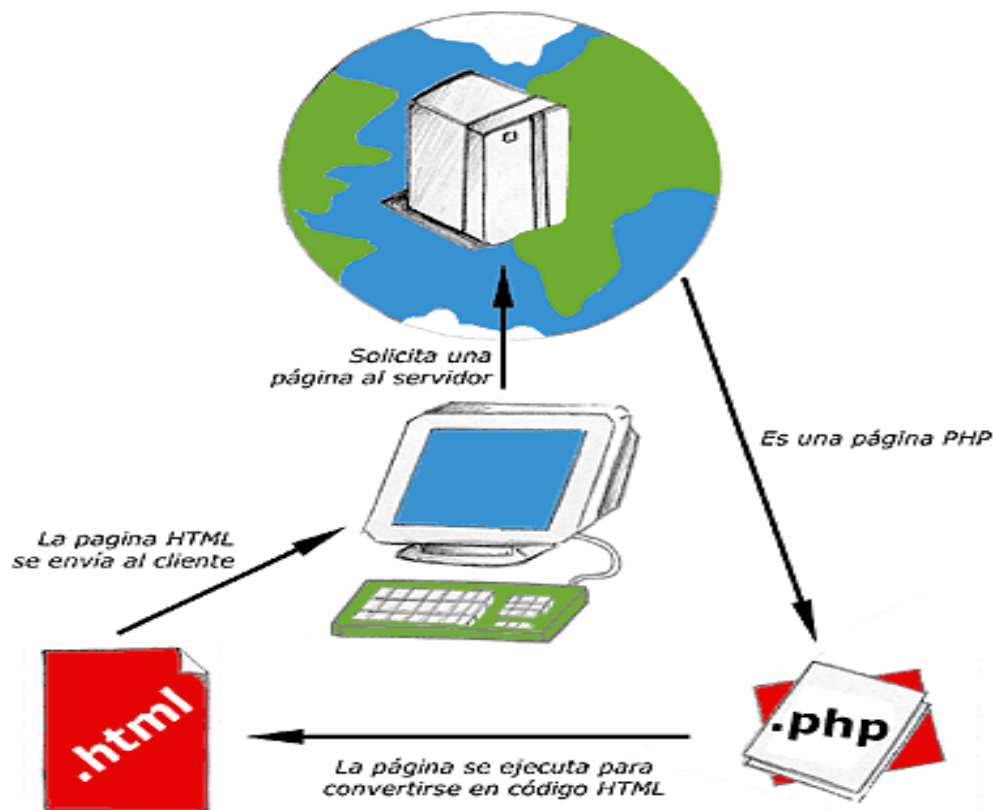


Figura 1: Esquema del funcionamiento de las páginas PHP.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Puede descargarse a través de la página principal de PHP www.php.net y de manera gratuita, un módulo que hace que nuestro servidor web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para muchos servidores web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

2.4. Base de Datos

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como PostgreSQL, MySQL, mSQL, Oracle, Informix, y ODBC. Esto es muy importante para crear sitios web dinámicos vinculados a bases de datos donde la información que se muestra en las páginas web es el resultado entre otras cosas de la utilización del código PHP para hacer consultas a la base de datos y generar el código HTML.

Para entender mejor a que se refiere cuando se habla de base de datos se toma en cuenta para este trabajo lo planteado por C.J. Date “Un sistema de bases de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar; es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.” (C.J.Date, 2001)

A criterio de la Lic. Rosa María Mato García la definición de bases de datos es: “Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo.” (R.M.M.García, 1999)

Tomando como referencia ambas definiciones anteriores se puede decir que, una base de datos no es más que un conjunto de datos referentes a algo y relacionados entre sí que se almacenan con el objetivo principal de ser utilizados en otro momento. Hoy día es muy importante que un sitio web tenga alguna vinculación con una base de datos para que pueda almacenar datos referentes a la información de la misma página o a los usuarios que la visitan y hacer que estos se sientan en la necesidad de consultar dicha información.

2.4.1. Sistemas Gestores de Base de Datos

Después de conocer qué es en sí una base de datos se puede ver entonces qué son los Sistemas Gestores de Base de Datos (SGBD). Debido al gran cúmulo de información que debían manejar las base de datos y lo complicado que se hacía su gestión se hizo necesario el uso de algún software especializado que permitiera el manejo de los datos, así como que garantizara el cumplimiento de ciertas reglas. De esta forma surgen los Sistemas Gestores de Bases de Datos (SGBD) que no son más que sistemas que permiten la utilización y/o actualización de los datos almacenados en una o varias Bases de Datos, desde diferentes puntos de vista a la vez. (C.J.Date, 2001)

Entre las principales ventajas de los SGBD se encuentran:

- Eliminan las inconsistencias en los datos.
- Permiten compartir los mismos datos entre diferentes aplicaciones con distintas necesidades.
- Ahorran espacio de almacenamiento al no existir redundancia o ser ésta escasa.
- Mejoran la seguridad de los datos, pues normalmente incorporan mecanismos de seguridad en el propio SGBD.
- Permiten la creación de entornos de alta disponibilidad, pues con algunos SGBD es posible llegar a disponer de aplicaciones funcionando ininterrumpidamente
- Mejora en los servicios de copias de seguridad y de recuperación ante fallos.

2.4.2. PostgreSQL y pgAdmin III

Teniendo en cuenta que en el proyecto Plataforma VideoWeb para el cual está dirigido la realización del producto de la presente investigación se trabaja con el sistema de bases de datos PostgreSQL haciendo uso pgAdmin como aplicación de gestión de base de datos. Se ha decidido entonces utilizar dichas tecnologías con el fin de mantener el entorno de trabajo existente considerando además las ventajas que brindan por si solas estas técnicas las cuales se mencionan a continuación.

PostgreSQL: Es un avanzado sistema de bases de datos relacionales basado en Open Source. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. PostgreSQL está bajo licencia BSD.

Un sistema de base de datos relacionales es un sistema que permite la manipulación de acuerdo con las reglas del álgebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras. (www.postgresql.net, 2011)

pgAdmin III: Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source o sea de código abierto. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. (guia-ubuntu.org, 2009)

pgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL (por sus siglas en inglés Structured Query Language lo cual traducido al español significa Lenguaje de Consulta Estructurado) simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP (ver página 64) o Unix Domain Sockets y se puede encriptar mediante SSL (por sus siglas en inglés Secure Socket Layer lo que podría traducirse al español como Enchufe Seguro de Capas) para mayor seguridad.

2.5. IDE de desarrollo

Un IDE (por su siglas en inglés Integrated Development Environment las cuales traducidas al español significan Entorno de Desarrollo Integrado) es un conjunto de herramientas de desarrollo de software para programadores. Generalmente están compuestos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario. (Malvaez, 2008)

Los IDEs son creados para el desarrollo de aplicaciones en un sólo lenguaje de programación sin embargo, hay algunos en los que se puede desarrollar en más de un lenguaje como son el caso de Netbeans, Eclipse y Visual Studio entre otros.

Como se ha planteado anteriormente en dicha investigación se va a utilizar el editor WYSIWYG CKEditor el cual ha sido desarrollado haciendo uso de PHP, HTML, CSS y muy especialmente Javascript con

Jquery. Por tanto se hace necesario utilizar un IDE que integre la programación con todas estas tecnologías además de que sea compatible con postgresql.

2.5.1. Fundamentación de NetBeans como IDE de desarrollo a utilizar

NetBeans IDE además de integrar las tecnologías necesarias planteadas en el epígrafe anterior es un entorno de desarrollo integrado modular de base estándar (normalizado).

El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. (netbeans.org, 2011)

Algunas de las características más relevantes del IDE NetBeans para la presente investigación encontradas en el sitio oficial de NetBeans netbeans.org son:

- Propone un esqueleto para organizar el código fuente ya que el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS.
- Soporte JavaScript
 - Sintaxis Resaltada
 - Completamiento de Código y Análisis de Típo
 - Soluciones Rápidas (Quick Fixes) y Verificación de Sintaxis
- Soporte de Edición para PHP: Completamiento de código, sintaxis resaltada, navegación, depuración y mucho más.
- Línea de comandos de depuración para PHP: La salida del programa PHP aparece en una pantalla de línea de comandos en el IDE y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.
- Cuenta con un depurador JavaScript

Por tanto en la presente investigación se ha decidido utilizar como IDE de desarrollo a NetBeans IDE debido a las grandes ventajas que ofrece y a que el grupo de trabajo tiene amplio conocimiento y años de experiencia en el uso de esta herramienta, cualidad que agiliza el proceso de desarrollo. Además es muy potente para la programación con PHP, Javascript, HTML, PostgreSQL y todas las otras tecnologías vinculadas a la investigación.

Otros IDE de desarrollo como Eclipse tienen iguales prestaciones que el IDE seleccionado para utilizar en la investigación, pero no cuentan con otras características de vital importancia con las que cuenta

NetBeans IDE. El IDE Eclipse es una herramienta muy potente pero es más compleja de configurar para comenzar el trabajo con ella, además usa más recursos de la computadora. Partiendo de la idea de que el desarrollador de la aplicación y el grupo de trabajo no tiene conocimientos de esta herramienta, el uso del IDE Eclipse podría traer grandes problemas de adaptabilidad y esto acarrearía a largo plazo un retraso en la codificación del sistema.

2.6. Conclusiones del capítulo

Con el desarrollo de este capítulo se pudo definir que tecnologías se utilizarán en la presente investigación. A partir de que se definió en el capítulo anterior al CKEditor como el editor WYSIWYG disponible para Drupal que reúne mejores condiciones para ser usado en el desarrollo de la investigación. Se pudo conocer que tecnologías están vinculadas con dicho editor y se pudo hacer un estudio de las mismas, por ejemplo PHP, HTML, CSS, Javascript y JQuery.

Además se eligieron y fundamentaron otras técnicas que se utilizarán tales como:

- RUP como metodología.
- Visual Paradigm como herramienta CASE
- PostgreSQL como sistema gestor de base de datos
- PgAdmin como aplicación para gestionar el sistema gestor de base de datos PostgreSQL
- IDE NetBean como entorno de desarrollo

Capítulo 3: “Construcción de la Solución Propuesta”

3.1. Introducción

El presente capítulo está dedicado a la realización de los pasos que define la metodología RUP para el desarrollo de software comenzando por el análisis del dominio del problema hasta la fase del modelo de implementación.

A partir de realizar un estudio inicial del entorno en el cual se enmarca el problema a resolver. Se brinda información referente a los actores y casos de uso del dominio y del sistema, de manera que se pueda entender mejor el producto que se desea obtener. Además se detallan los requisitos funcionales y no funcionales. Luego de culminada estas tareas se podrá realizar el modelado del análisis, diseño e implementación. Estas actividades permitirán realizar una correcta construcción de la solución propuesta para la presente investigación.

3.2. Modelo de Dominio

Para la presente investigación se toma lo planteado por el colectivo de autores del libro “El proceso unificado de desarrollo” donde se define que “Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan “cosas” que existen o los eventos que suceden en el entorno en que se trabaja en el sistema”. (Jacobson, 2000)

Se puede decir entonces que un modelo de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Es un modelo de clases conceptuales del dominio del problema que representa conceptos del mundo real, no de los componentes de software y que trae como objetivo: “comprender y describir las clases más importantes dentro del contexto del sistema”.

Las clases del dominio aparecen en tres formas típicas:

- Objetos del negocio que representan cosas que se manipulan en el negocio, como pedidos, cuentas y contactos.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento, como la aviación enemiga, misiles y trayectorias.
- Sucesos que ocurrirán o han ocurrido, como llegada de un avión, su salida y la hora de comida. (Jacobson, 2000)

El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML para representar conceptos, el Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión.

3.3. Breve descripción del sistema:

Como se menciona en la introducción del presente documento en el proyecto Plataforma VideoWeb se trabaja en una aplicación web que se encarga de gestionar el trabajo con las medias audiovisuales. El proceso de almacenado de los archivos multimedia se lleva a cabo por el **administrador** del sistema el cual se encarga de guardar el **fichero multimedia** y de definir a que tipología corresponde dicha media.

La **tipología** es un concepto que se utiliza en el proyecto Plataforma VideoWeb con el fin de definir conjuntos distintos para poder agrupar las medias audiovisuales de similares características y así diferenciarlas del resto de las medias. El aspecto que diferencia una tipología de otra son los **metadatos** referentes a las medias audiovisuales que se registran para cada una de ellas. Los metadatos referentes a las medias audiovisuales no son más que los datos que se desean almacenar para cada media. Por tanto cada tipología es almacenada en la **base de datos** y está compuesta por un conjunto distinto de metadatos lo cuales van a ser los datos que se van a almacenar en cada caso.

En un caso normal del proceso de almacenado de medias se realizarían los siguientes pasos:

Caso 1:

1. Asignarle a la media audiovisual una tipología.
2. Registrar los metadatos de la media que defina dicha tipología.

De este modo se podría continuar normalmente con el proceso de publicación llevado a cabo por el **revisor** del sistema.

En el caso de que la media que se va a almacenar no corresponda con ninguna de las tipologías definidas en ese momento entonces se realizarían las siguientes operaciones:

Caso 2:

1. Crear nueva tipología
 - 1.1. Establecer que metadatos se recogerán de la tipología
2. Asignar tipología a la media audiovisual.

3. Registrar los metadatos de la media que defina dicha tipología.

Como se observa, en el caso 1, no existe ningún problema en el proceso de publicación de la media audiovisual. Esto es porque al pertenecer a una tipología previamente definida se conocen los datos que se van a almacenar y estos mismos datos van a ser los que se van a publicar en el sitio a través de **plantillas** web ya construidas. Por tanto cada tipología tiene un conjunto de plantillas ubicadas físicamente en el **servidor web** de la aplicación y tienen un diseño distinto para mostrar de diversas formas los contenidos de los archivos multimedia. Esto quiere decir que para el caso 2 planteado anteriormente las medias audiovisuales carecerán de plantillas para su publicación.

3.4. Diagrama de Clases del Dominio

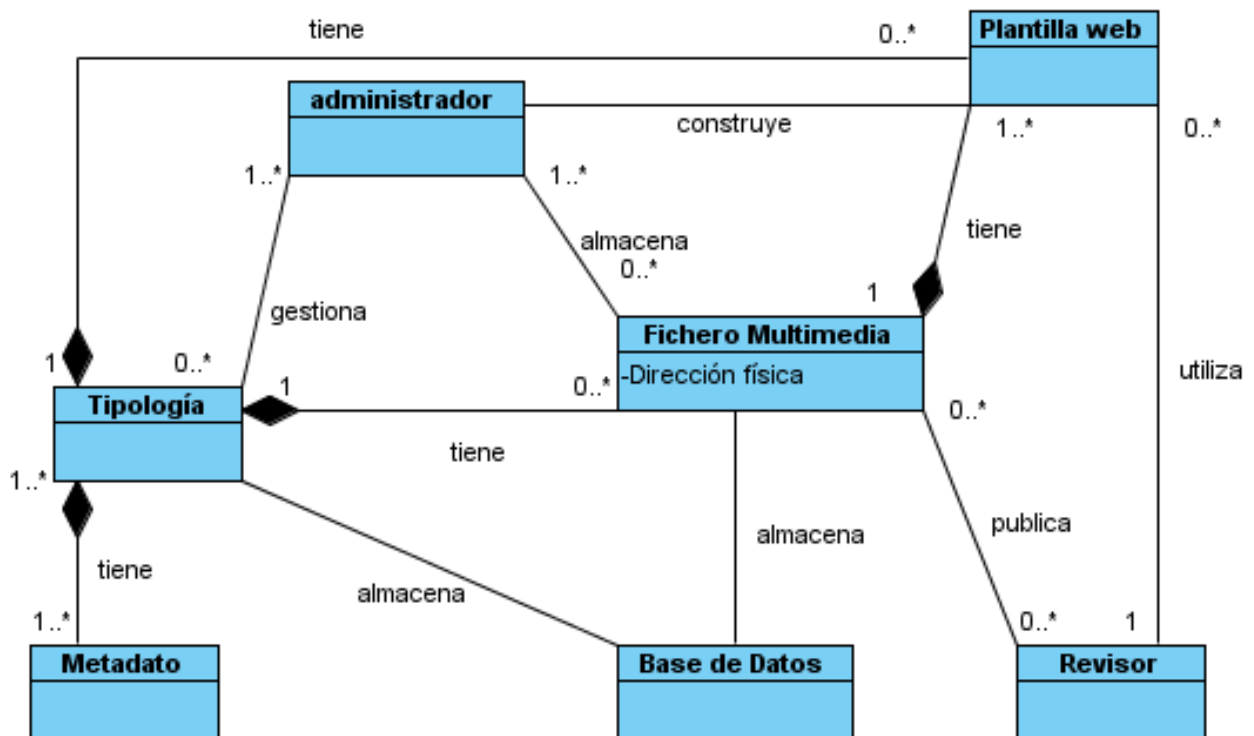


Figura 2: Diagrama de Clases del Dominio

3.4.1. Descripción de las clases del dominio

Administrador: Persona encargada de almacenar las medias y de gestionar las tipologías.

Revisor: Persona encargada de publicar las medias.

Fichero Multimedia: Fichero multimedia de audio o video que será publicado en la plataforma y posteriormente reproducido por los usuarios.

Tipología: Parámetro que se utiliza para definir los grupos de medias que se gestionan en el sistema. Este parámetro define los metadatos de las medias de un mismo tipo.

Ejemplo: películas, series, documentales, video musical entre otros.

Metadatos: Constituyen los datos referentes a las medias los cuales son los que se recogen según la tipología a la que pertenece dicha media. Un ejemplo de metadatos son los referentes a la tipología película los cuales son: nombre, director, poster, sinopsis entre otros.

Base de Datos: Entidad en la cual se almacenan de manera estructurada y con la menor redundancia posible los datos referentes a los artículos de contenido, los contenidos audiovisuales, los usuarios de la plataforma de VideoWeb entre otros datos.

Plantilla web: Fichero que se utiliza para publicar los datos de las tipologías. Normalmente una tipología tiene varias plantillas web para mostrar de diversas formas sus metadatos. Estos ficheros son creados e implementados por los administradores cada vez que se incorpore una nueva tipología de archivo multimedia al sistema.

3.5. Captura de Requisitos

Roger Pressman en su libro “Ingeniería de Software. Un enfoque práctico” plantea: “La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan difícil de rectificar posteriormente”. (Pressman, 2003)

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema. (Jacobson, 2000)

Para la presente investigación la captura de requisitos se va a realizar a partir de la entrevista al grupo de desarrollo que trabaja en el proyecto Plataforma VideoWeb. Esta tarea posibilitará la correcta captura de los requisitos tanto funcionales como no funcionales debido a que dicho grupo de trabajo tiene todos los conocimientos necesarios para que el producto final de esta investigación cumpla con las capacidades y cualidades necesarias.

A continuación se listan los requisitos funcionales y no funcionales del sistema.

3.5.1. Requerimientos funcionales

El sistema debe ser capaz de:

RF 1 - Adicionar plantilla: El sistema debe permitir adicionar una plantilla nueva para cualquiera de las tipologías existentes.

RF 2 - Editar plantilla: El sistema debe permitir modificar el contenido de alguna de las plantillas creadas hasta ese momento. Modificar el contenido implica cambiar el diseño, introducir nuevos elementos, eliminar elementos insertados anteriormente o cualquier otra operación que permita el editor original.

RF 3 - Eliminar plantilla: Se debe dar la posibilidad de eliminar cualquiera de las plantillas creadas hasta ese momento.

RF4 - Buscar plantillas: EL sistema debe permitir realizar una búsqueda a partir de un grupo de filtros que definan tipo de plantilla y tipología.

3.5.2. Requerimientos no funcionales

El sistema debe caracterizarse por:

Requerimientos de apariencia o interfaz externa.

- Interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario.

Requerimientos de confidencialidad.

- La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.
- Acceder a la información según el rol del usuario.

Requerimientos de disponibilidad.

- El sistema debe de estar disponible las 24 horas del día.

Requerimientos de Soporte.

- El soporte y/o mantenimiento del sitio no debe detener el servicio.

Restricciones en el diseño e implementación.

- El sistema estará implementado en lenguaje php5, utilizando como IDE de desarrollo NetBeans.
- Para la modelación de la arquitectura se utilizará el Visual Paradigm.

Restricciones de usabilidad.

- El sistema podrá ser utilizado por usuarios que no necesariamente tengan conocimientos informáticos por lo que todas sus funcionalidades deben ser claras, sencillas y accesibles de manera intuitiva.

3.6. Modelo de Sistema

Luego de realizar el modelado del dominio se pudo conocer el funcionamiento del sistema en los aspectos relevantes para la presente investigación. Se definieron entonces que elementos del sistema son de relevancia para este trabajo y los requerimientos tanto funcionales como no funcionales. Todas estas actividades permiten conocer cómo funciona el sistema y qué capacidades y cualidades debe tener la aplicación a desarrollar. De esta manera se da paso al modelado del nuevo sistema buscando que éste de solución a los requerimientos establecidos.

3.6.1. Definición de casos de uso del sistema

Los casos de uso del sistema modelan el sistema desde el punto de vista del usuario y son creados durante la obtención de requisitos. En general, un caso de uso es, simplemente, un texto escrito que describe el papel de un actor que interactúa con el acontecer del sistema. Estos facilitan una descripción de como el sistema se usará. (Pressman, 2001)

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establecen un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

3.6.2. Patrón CRUD

Este patrón se utiliza en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre CRUD por su las siglas en ingles de Create Read Update Delete lo cual traducido al español significa Crear Leer Modificar Eliminar. Dentro del patrón CRUD existe el CRUD

Completo el cual consiste en un caso de uso para administrar la información que permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y eliminar. Este patrón deberá ser usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples. (ECURED, 2005)

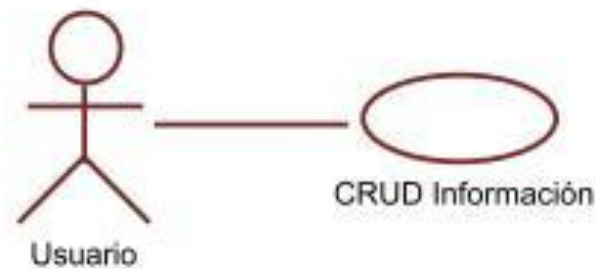


Figura 3: Patrón CRUD completo.

En sentido general este patrón es de mucha importancia utilizarlo en la presente investigación porque simplificará los requisitos que se encargan de gestionar el trabajo con las plantillas. Esto traerá un mejor entendimiento del modelado del sistema y así se logrará un mejor refinamiento del mismo.

3.6.3. Diagrama de caso de uso del sistema

Un diagrama de casos de uso describe parte del modelo de caso de uso y muestra un conjunto de casos de usos y actores con una asociación entre cada par *actor/caso de uso* que interactúan. (Pressman, 2001) Por tanto con su realización se ayuda a que los clientes, los usuarios y los desarrolladores establezcan un acuerdo sobre cómo utilizar el sistema. En el epígrafe anterior se definió la utilización del patrón *CRUD Completo* el cual se identifica en el diagrama de caso de uso del sistema donde el CU Gestionar Plantilla engloba los requerimientos funcionales: Adicionar, editar y eliminar plantilla. Las interrelaciones entre las acciones y los actores de la aplicación son agrupadas en el diagrama de casos de uso del sistema que se muestra a continuación.

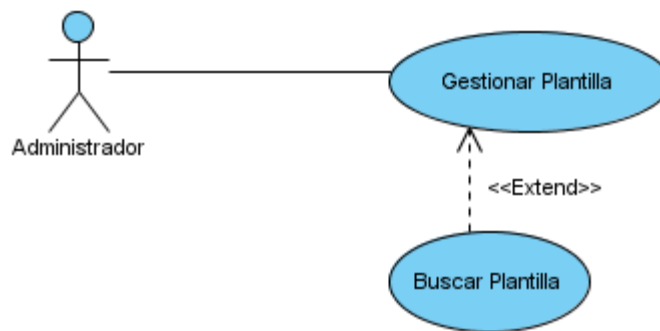


Figura 4: Diagrama de Casos de Uso del Sistema

3.6.4. Descripción de los actores

EL modelo de caso de uso describe lo que hace el sistema para cada tipo de usuario. Cada uno de éstos se representa mediante uno o más actores. Por tanto, los actores representan terceros fuera del sistema que colaboran en el mismo. (Jacobson, 2000)

Actor:

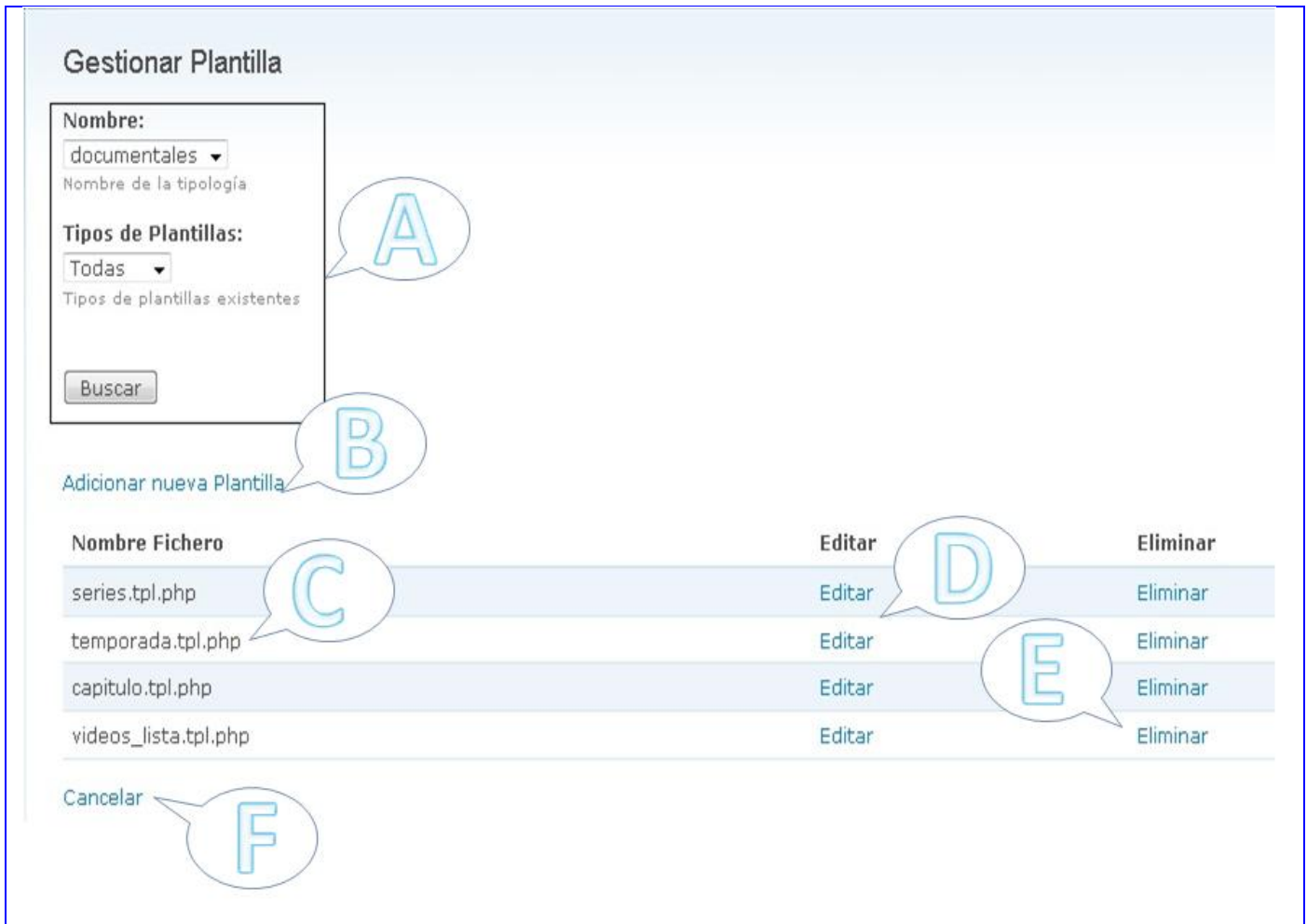
Administrador: Rol que se encarga de gestionar las plantillas de presentación de contenido de los archivos multimedia.

3.6.5. Descripción del caso de uso Gestionar Plantilla

Tabla 2: Descripción de Caso de Uso Gestionar Usuario

Caso de uso	Gestionar Plantillas
Actores	Administrador
Resumen	Este caso se inicia cuando el Administrador desde la opción de Gestionar Plantilla, en el menú de trabajo, podrá crear, modificar o eliminar plantillas. El caso de uso termina con la creación o modificación de plantillas, la eliminación de una plantilla o la cancelación de cualquier operación de las anteriores.
Precondiciones	Se ha autenticado como administrador.
Referencia	RF 1, RF 2, RF 3
Prioridad	Crítico
Flujo normal de eventos	

Acción del Actor	Respuesta del Sistema
1. El Administrador presiona el vínculo “Gestionar plantilla” de su interfaz de trabajo.	2. El sistema muestra un formulario con una serie de filtros para buscar una plantilla específica (A)(Ver CU Buscar Plantilla anexo 2), un vínculo a la página para adicionar plantillas (B), y una tabla donde se muestran un listado de todas las plantillas que están creadas (C) donde se define para cada plantilla un vínculo para editarla (D) y otro para eliminarla (E) .Está además el vínculo Cancelar (F) para ir a la página inicio del sitio (véase Prototipo de Interfaz Gestionar Plantilla).
3. El Administrador escoge una de las opciones: a. Adicionar plantilla, seleccionando el vínculo “Adicionar nueva Plantilla”.(B) Sección Adicionar Plantilla. b. Eliminar plantilla, seleccionando el vínculo “Eliminar”(E) Sección Eliminar Plantilla. c. Editar Plantilla, seleccionando el vínculo “Editar”(D). Sección Editar Plantilla. d. Cancelar seleccionando el vínculo <u>Cancelar</u> (F).	
Prototipo de Interfaz Gestiona Plantilla	



Sección: “Adicionar Plantilla”

Acción del Actor	Respuesta del Sistema
	4. El sistema busca y muestra las tipologías existentes (A) y los tipos de plantillas (B) en la interfaz seleccionar tipo de plantilla.(véase Prototipo Seleccionar Tipo). Además se encuentra un botón “Enviar” (C) y un vínculo “Cancelar” (D)
5. El administrador elige la tipología (A) y el tipo de plantilla (B) que va a adicionar y selecciona la opción “Enviar” (C)	6. El sistema busca los metadatos asociados a la tipología y visualiza la interfaz adicionar plantilla.(véase Prototipo Adicionar Plantilla 1) . En

Capítulo 3: “Construcción de la Solución Propuesta”

	esta interfaz se muestran los metadatos de la tipología buscada (D) el campo de texto modificado con el CKEditor (S) además de un Botón “Guardar” (G) y un vínculo”Cancelar” (H).
7. El administrador realiza un diseño cualquiera de la plantilla haciendo uso de los metadatos de la tipología (D) y de las operaciones que le permite realizar el CKEditor (E). Además utiliza le nuevo botón implementado (F) con el cual se incluyen la variable PHP del metadato seleccionado en la opción (D).	8. El sistema va visualizando el diseño creado por el administrador en el cuerpo de texto del CKEditor (M). Para hacer referencia a la entrada de variables PHP el sistema coloca en el lugar donde esté el cursor una referencia a esta variable. Para esto por cada variable se insertan dos íconos para marcar el inicio (I) y el fin (J) de un texto alternativo (K) que en dependencia del tipo de dato del metadato insertado tendrá variada longitud y/o formato.
9. El administrador selecciona un metadato correspondiente a una imagen (D) y oprime el botón Insertar Imagen (L)	10. El sistema muestra una nueva interfaz (véase Prototipo Propiedades de Imagen) donde hay varias opciones para darle propiedades a la imagen. Además de un nuevo botón (N) agregado para insertar la variable PHP correspondiente a la imagen.
11. El administrador selecciona el nuevo botón (N) para adicionar la variable PHP correspondiente al metadato.	12. El sistema comprueba que el tipo de dato del metadato seleccionado corresponda a “url”. Luego coloca en el cuadro de texto (O) de la URL una dirección de una imagen de prueba.
13. El administrador selecciona la opción “Aceptar” (P)	14. El sistema cierra la ventana para insertar imágenes y visualiza en el lugar donde esté el cursor una imagen (R) de muestra para que el administrador pueda ver como quedan las propiedades de imagen que aplicó en esta.

15. Al finalizar el diseño de la plantilla el administrador selecciona la opción “Guardar” (G)	16. El sistema crea un fichero y lo guarda con un nombre compuesto por el tipo de plantilla y la tipología seleccionada. Luego pasa al paso 2 del flujo normal de eventos.
--	--

Flujos Alternos

Acción del Actor	Respuesta del Sistema
15.1.a.- El administrador selecciona la opción “Cancelar” (G).	15.1.b.- El sistema regresa al paso 2 del flujo normal de eventos.
13.1.a.- El Administrador Selecciona la opción “Cancelar” (Q).	13.1.b.- El sistema regresa al paso 8 del flujo normal de eventos.
	12.1.b.- El sistema muestra el mensaje de error “ La variable debe ser de tipo url.” Y vuelve al paso 10 del flujo normal de eventos.
5.1.a El administrador selecciona la opción “Cancelar” (H).	5.1.b.- El sistema regresa al paso 2 del flujo normal de eventos.

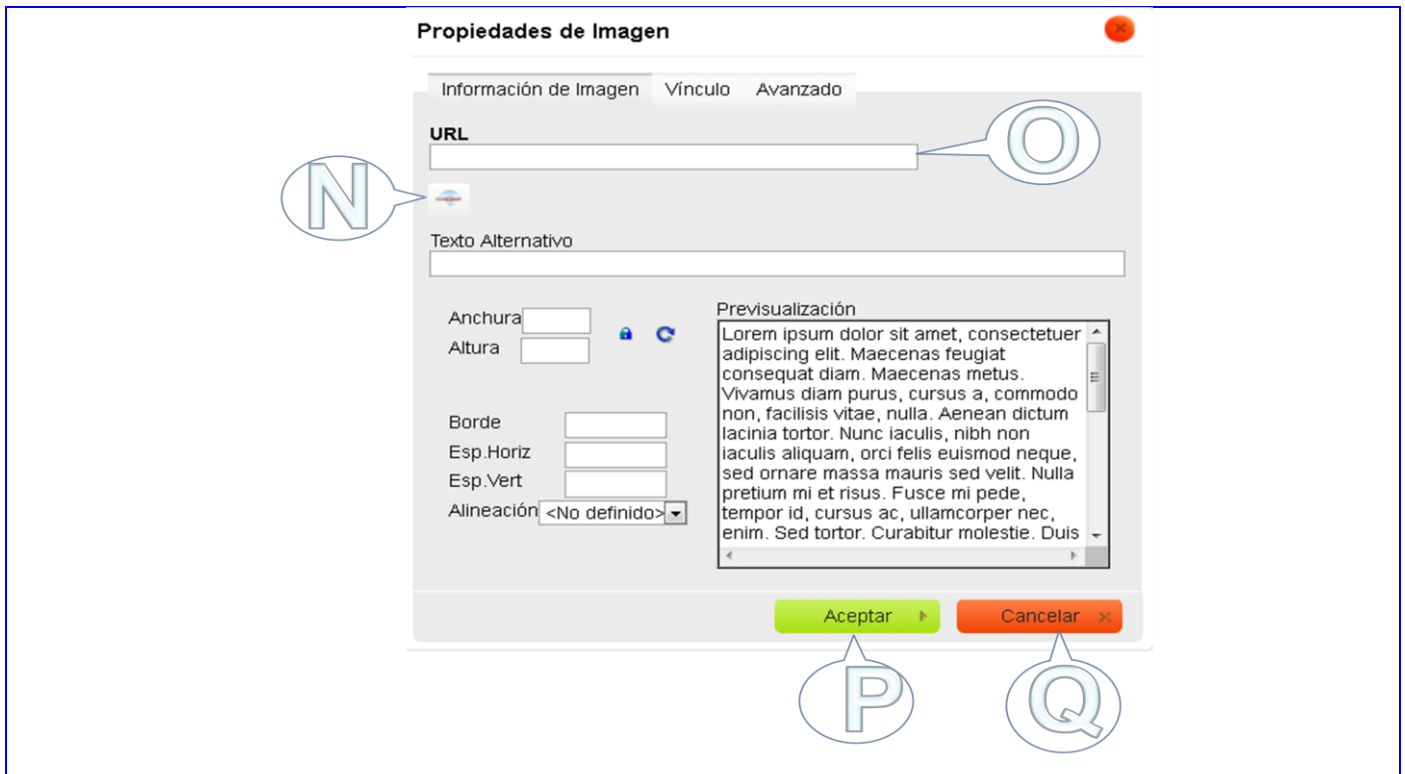
Prototipo de Interfaz Seleccionar Tipo



Prototipo de Interfaz Adicionar Plantilla



Prototipo de Interfaz Propiedades de Imagen



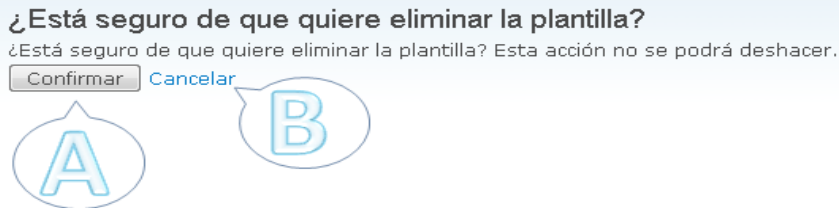
Sección: “Eliminar Plantilla”

Acción del Actor	Respuesta del Sistema
	4. El sistema muestra un formulario de confirmación y las opciones “Eliminar” (A) y “Cancelar” (B).
5. El administrador selecciona la opción “Eliminar” (A).	6. El sistema elimina el fichero físico correspondiente a la plantilla y regresa al paso 2 del flujo normal de eventos.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
5. El administrador selecciona la opción “Cancelar” (B).	6. El sistema regresa al paso 2 del flujo normal de eventos.

Prototipo de Interfaz

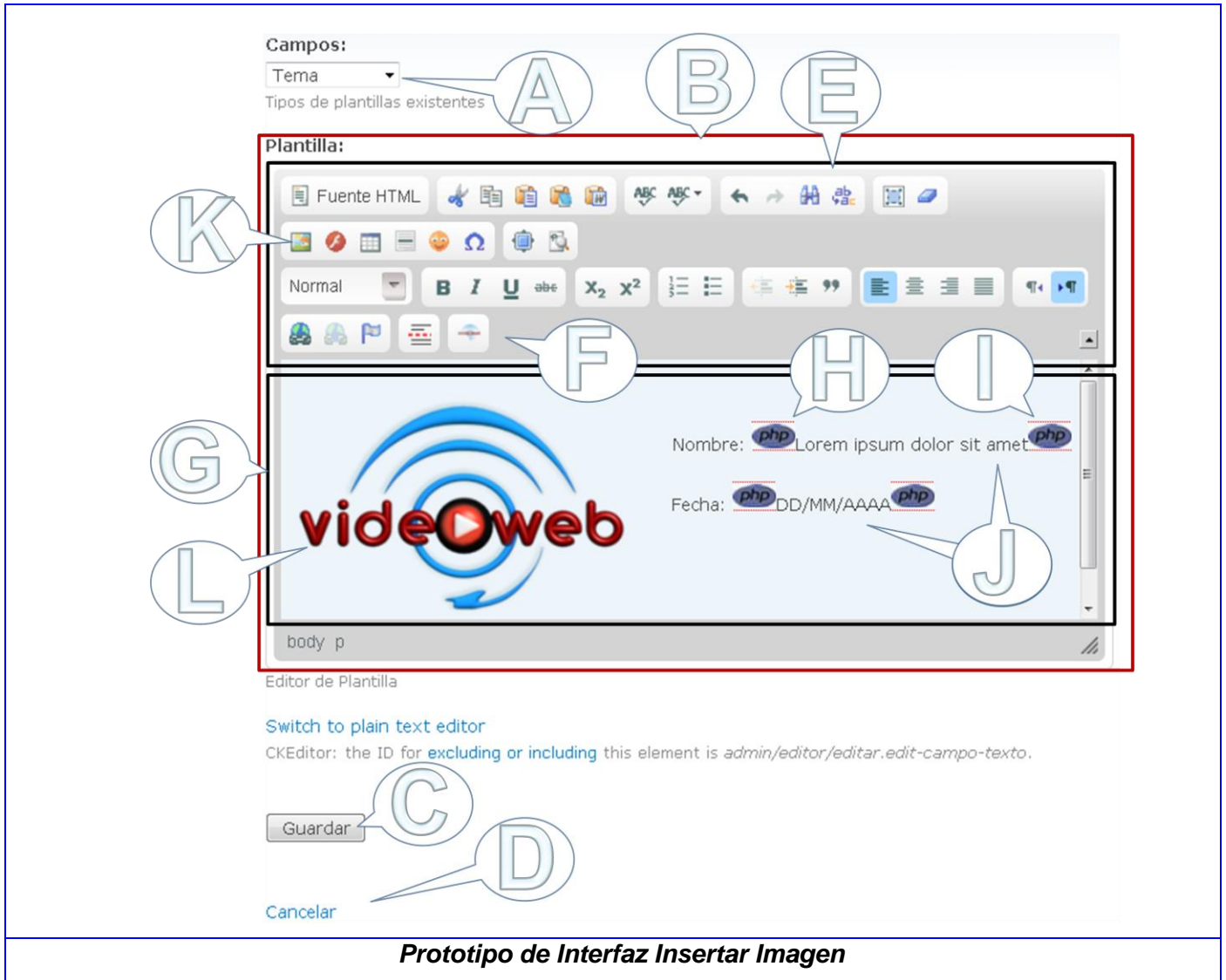


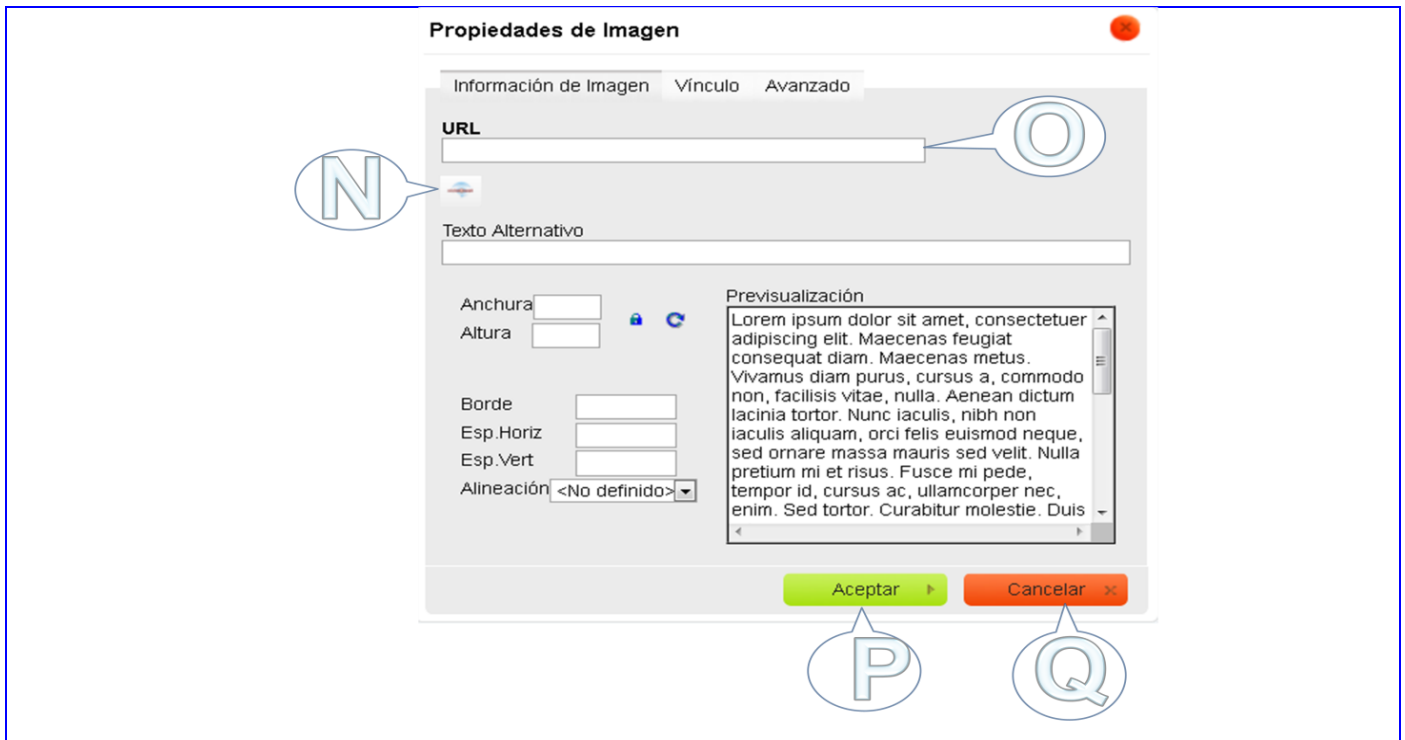
Sección: “Editar Plantilla”

Acción del Actor	Respuesta del Sistema
	4. El sistema busca el fichero correspondiente a la plantilla que ha sido solicitada para editar y muestra la interfaz editar plantilla (véase Prototipo de Interfaz Editar Plantilla), donde se mostrarán los datos de la tipología a la que pertenece (A). Luego carga el contenido de la plantilla (el código) y lo muestra en el CKEditor (B). Tendrá además un botón “Guardar” (C) y un vínculo “Cancelar” (D).
5. El administrador realiza un diseño cualquiera de la plantilla haciendo uso de los metadatos de la tipología (A) y de las operaciones que le permite realizar el CKEditor (E). Además utiliza le nuevo botón implementado (F) con el cual se incluyen la variable PHP del metadato seleccionado en la opción (D).	6. El sistema va visualizando el diseño creado por el administrador en el cuerpo de texto del CKEditor (G). Para hacer indicar la entrada de variables PHP el sistema coloca en el lugar donde esté el cursor una referencia a esta variable. Para esto por cada variable se insertan dos íconos para marcar el inicio (H) y el fin (I) de un texto alternativo (J) que en dependencia del tipo de dato del metadato insertado tendrá variada longitud y/o formato.
7. El administrador selecciona un metadato (A) correspondiente a una imagen y oprime el botón Insertar Imagen (K)	8. El sistema muestra una nueva interfaz (véase Prototipo Propiedades de Imagen) donde hay varias opciones para darle propiedades a la imagen. Además de un nuevo botón (N) agregado

Capítulo 3: “Construcción de la Solución Propuesta”

	para insertar la variable PHP correspondiente a la imagen
9. El administrador selecciona el nuevo botón (N) para adicionar la variable PHP correspondiente al metadato.	10. El sistema comprueba que el tipo de dato del metadato seleccionado corresponda a “url. Luego coloca en el cuadro de texto (O) de la URL una dirección de una imagen de prueba un texto alternativo.
11. El administrador selecciona la opción “Aceptar” (P)	12. El sistema cierra la ventana para insertar imágenes y visualiza en el lugar donde esté el cursor una imagen (L) de muestra para que el administrador pueda ver como quedan las propiedades de imagen que aplicó en esta.
13. Al finalizar el diseño de la plantilla el administrador selecciona la opción “Guardar” (C)	14. El sistema crea un fichero y lo guarda con un nombre compuesto por el tipo de plantilla y la tipología seleccionada. Luego pasa al paso 2 del flujo normal de eventos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
13.1.a.- El administrador selecciona la opción “Cancelar” (D).	13.1.b.- El sistema regresa al paso 2 del flujo normal de eventos.
11.1.a.- El Administrador Selecciona la opción “Cancelar” (Q).	11.1.b.- El sistema regresa al paso 6 del flujo normal de eventos.
	10.1.b.- El sistema muestra el mensaje de error “ La variable debe ser de tipo url.” Y vuelve al paso 10 del flujo normal de eventos.
Prototipo de Interfaz Editar Plantilla	





Análisis y Diseño

El flujo de trabajo de análisis y diseño cuenta con una gran cantidad de actividades de vital importancia para el desarrollo de la aplicación. A criterio del conjunto de autores del libro *El proceso unificado de desarrollo de software* “Durante el análisis y el diseño, se transforma el modelo de casos de uso del sistema mediante un modelo de análisis en un modelo de diseño, es decir, en una estructura de clasificadores y realizaciones de casos de uso. El objetivo es realizar los casos de uso de una forma económica de manera que el sistema ofrezca un rendimiento adecuado y pueda evolucionar en el futuro.” (Ivar Jacobson, 2000)

Se puede plantear entonces que en esta parte de la investigación se realizará el análisis y diseño de los casos de usos tal y como indica RUP que como se planteó en el capítulo anterior es la metodología de desarrollo escogida para esta investigación.

3.7. Análisis

Durante esta fase, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. El objetivo es conseguir una descripción más precisa para una mejor comprensión de

los mismos. Esto ayudará a estructurar todo el sistema, incluyendo su arquitectura. El objetivo del análisis es comprender perfectamente los requisitos del software.

3.7.1. Modelo de análisis

El lenguaje que se utiliza en el análisis se basa en un modelo de objetos conceptual, que se denomina modelo de análisis. El modelo de análisis ayuda a refinar los requisitos y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos. Estos recursos internos pueden figurar como un objeto en el modelo de análisis. Además el modelo de análisis nos ofrece un mayor poder expresivo y una mayor formalización. (Jacobson, 2000)

3.7.2. Diagrama de clases del análisis

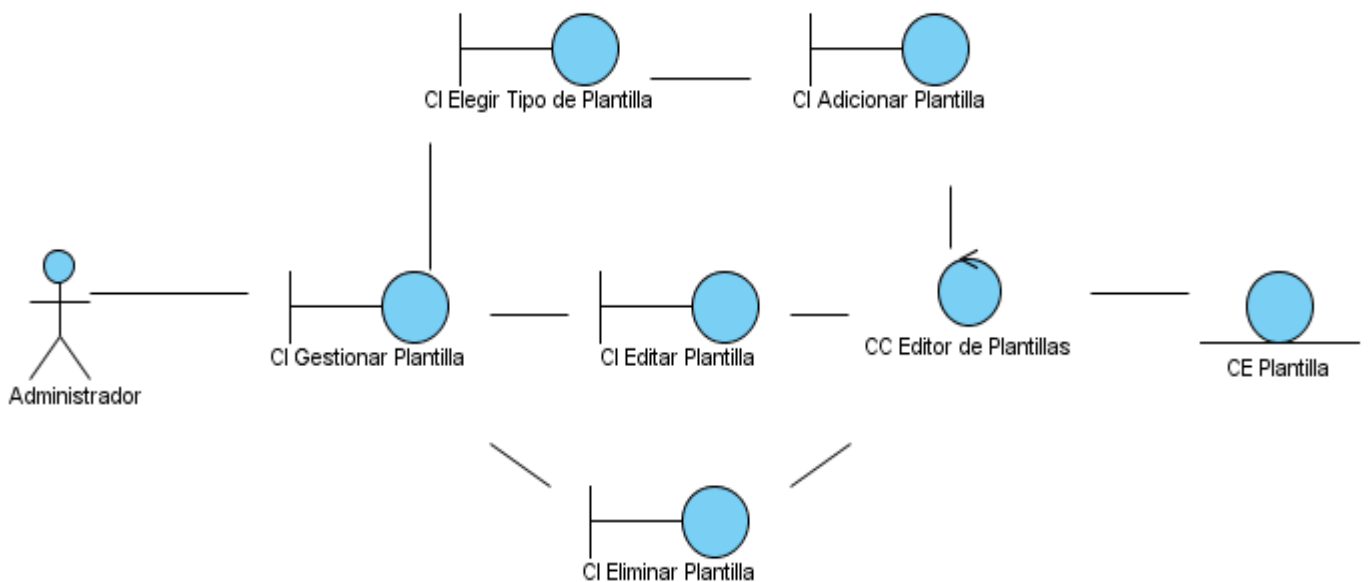


Figura 5: Diagrama de Clases del Análisis CU Gestiona Plantilla

Para el desarrollo del diagrama de clases del análisis se establecen tres estereotipos que definen los principales tipos de clases que pueden interactuar en estos diagramas. Estos tipos de clases son las Entidad, Interfaz y Controladora.

Clase Interfaz: Se utiliza para moldear la interacción entre el sistema y los actores, representan abstracciones de formularios, paneles, interfaces de comunicación etc.

Clase Entidad: Se utiliza para moldear información que es persistente y referente a algún fenómeno o concepto como una persona, objeto del mundo real o suceso del mundo real.

Clase Controladora: Representan coordinación, secuencia, transacciones y control de objetos y se usan con frecuencia para representar el control de un caso de uso en concreto.

3.7.3. Patrón Arquitectónico Modelo Vista Controlador

Con el desarrollo del análisis del sistema se logró una mejor comprensión de cómo debe estar estructurado el mismo. A partir de este análisis se define utilizar en la presente investigación como patrón arquitectónico el Modelo Vista Controlador (MVC). Este patrón divide la programación para el acceso a datos, la generación de las vistas y la lógica de la programación en tres capas por separado: la capa de Modelo, la Vista y la Controladora respectivamente.

Controlador:

Es el responsable de recibir las peticiones de los clientes y de:

- Procesar estas peticiones, determinando qué clase es la que tiene que responder.
- Modificar el modelo en función de los parámetros que recibe ejecutando la lógica de negocio que el desarrollador haya definido.
- Redirigir a la vista en función de la lógica de negocio.

Modelo:

Es la parte responsable de la gestión de la información. En este punto se suelen incluir aquellas clases, herramientas, librerías, etc., que permiten el acceso a los datos, así como el modelado de los mismos.

Vista:

Es la responsable de la percepción que tienen los usuarios finales de la aplicación. Se incluyen las páginas HTML, Javascript, PHP, etc, y en general todos los elementos de visualización de la aplicación. Se nutre de la información que el controlador ha captado del modelo para conformar las páginas finales. (Studios, 2004)

En otras palabras, se puede decir entonces que utilizando esta arquitectura se puede realizar un diseño de manera tal que:

- La capa del modelo va a estar compuesta por la clase que se encargue de realizar las consultas a la base de datos para obtener la información necesaria de las tipologías existentes.
- La vista va a estar compuesta por la clase que se encargue de generar los formularios necesarios para poder realizar la gestión en la aplicación a realizar.
- El controlador estará compuesto por la clase que se encargará de controlar el flujo de datos entre la capa modelo y la vista según las solicitudes de los usuarios durante el uso de la aplicación.

3.8. Diseño

El modelo del análisis proporciona una comprensión detallada de los requisitos e impone una estructura del sistema que se debe conservar. Con su terminación quedan sentadas las bases para realizar el diseño de la aplicación. En el diseño se modela el sistema con el fin de encontrar una estructura que soporte los requisitos tanto funcionales como no funcionales y alguna otra restricción que se haya detectado. Para comenzar con el diseño es esencial tener los resultados del análisis.

3.8.1. Patrones

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (Gracia, 2005)

Patrones GRASP

GRASP por sus siglas en inglés General Responsibility Assignment Software Patterns lo cual traducido al español significa Patrones Generales de Software para Asignación de Responsabilidades. Estos patrones GRAPS son bien generales, dan un marco amplio de cómo puede desarrollarse el diseño del sistema y por tanto sirven de guía para encontrar que patrones de diseño se van a utilizar. Dentro de los patrones GRAPS existentes a continuación se menciona cuáles de ellos fueron utilizados en la presente investigación.

Patrón Bajo Acoplamiento:

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas.

Patrón Alta Cohesión:

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.

Patrón Experto:

Asignar una responsabilidad al experto en información

Estos tres patrones fueron utilizados al construir las clases editor.module editor_vista y editor_controladora de manera que

- La clase editor_vista se encarga únicamente de conformar la estructura de los formularios de las páginas PHP que se muestran en el módulo
- La clase editor_controladora se encarga únicamente de gestionar la información que solicita el usuario y lleva el control de las operaciones. Establecer la conexión entre los diferentes módulos que se utilizan en la investigación, o sea, editor, CKEditor y tipologia_am.
- La clase editor.module se encarga únicamente de establecer la conexión entre el módulo Editor y el Drupal.

De esta manera las clases tienen funcionalidades para realizar actividades completamente diferentes (bajo acoplamiento) por tanto esta característica definió entonces la responsabilidad de cada clase para poder gestionar una información determinada (patrón experto) e hizo que fuera necesario una amplia conexión entre las clases para poder acceder a funciones específicas (alta cohesión).

Patrones GoF

Teniendo en cuenta que en la presente investigación se está desarrollando una aplicación en el entorno del CMS Drupal, se debe entonces diseñar la solución propuesta con los patrones de diseños utilizados en dicho CMS. El diseño de Drupal implementa una serie de Patrones de Diseño GoF (por sus siglas en inglés Gang Of Four lo cual traducido al español significa Banda de Cuatro), de los cuales se obtuvo una amplia documentación en el sitio oficial del CMS Drupal. A continuación se relacionan aquellos patrones que fueron utilizados para implementar el módulo Editor y modificar otros ficheros, todo esto con el fin de dar solución a la problemática de esta investigación.

Patrones Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. (Tedeschi, 2011)

Drupal utiliza internamente este patrón de diseño para definir una única instancia para el acceso a la base de datos, o sea que al acceder a la base de datos queda implícito la utilización de este patrón.

Patrones Estructurales:

Bridge (Puente): Desacopla una abstracción de su implementación. (Tedeschi, 2011)

Los módulos en Drupal deben ser escritos de una forma que sea independiente del sistema de base de datos que se está utilizando, y proporciona la capa de abstracción para ello. En la implementación del módulo Editor, se utilizó este patrón para abstraer la capa de acceso a datos del resto.

Patrones de Comportamiento:

Chain of Responsibility (Cadena de responsabilidad): Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. (Tedeschi, 2011)

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

Particularmente en el módulo Editor se trabaja con estos menús para crear todas las Clases Interfaz de la aplicación. Dígase CI Gestionar Plantilla, CI Adicionar Plantilla, CI Eliminar Plantilla, etc.

Command (Orden): Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. (Tedeschi, 2011)

Muchos de los ganchos de Drupal utilizan el patrón de Command para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro, junto con los argumentos.

Este patrón es utilizado para la programación de los plugin del CKEditor y durante el desarrollo de esta investigación se trabajó con varios de estos plugin, ejemplo editor.js e image.js. De esta manera se evidencia la utilización de este patrón.

3.8.2. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales, no funcionales y algunas otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

3.8.3. Diagrama de Clases del Diseño

El conjunto de autores del libro “El proceso unificado de desarrollo de software” plantean: Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. (Ivar Jacobson, 2000)

A continuación se muestra el diagrama de clases del diseño correspondiente a la presente investigación.

Capítulo 3: “Construcción de la Solución Propuesta”

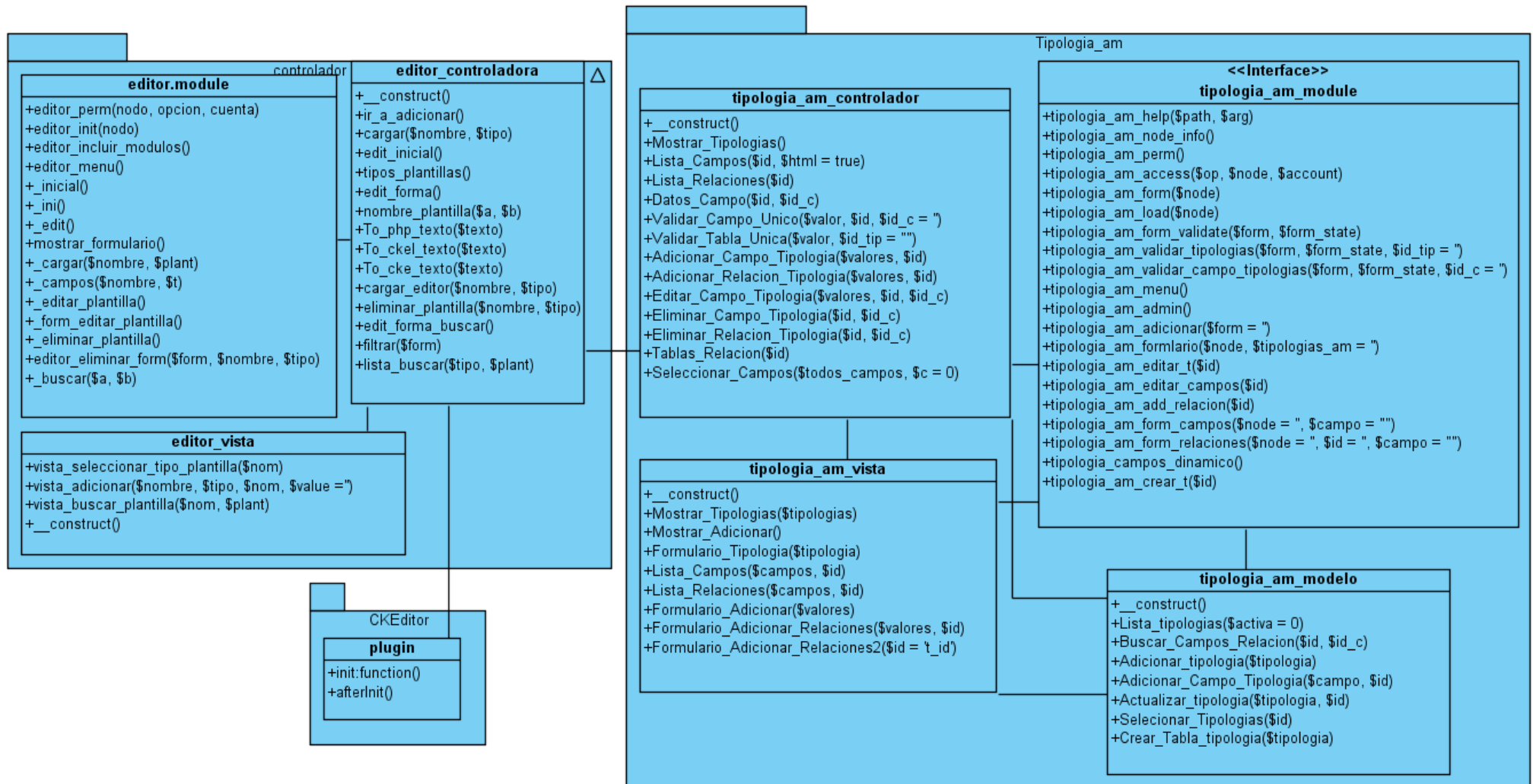


Figura 6: Diagrama de Clases del Diseño.

Capítulo 3: “Construcción de la Solución Propuesta”

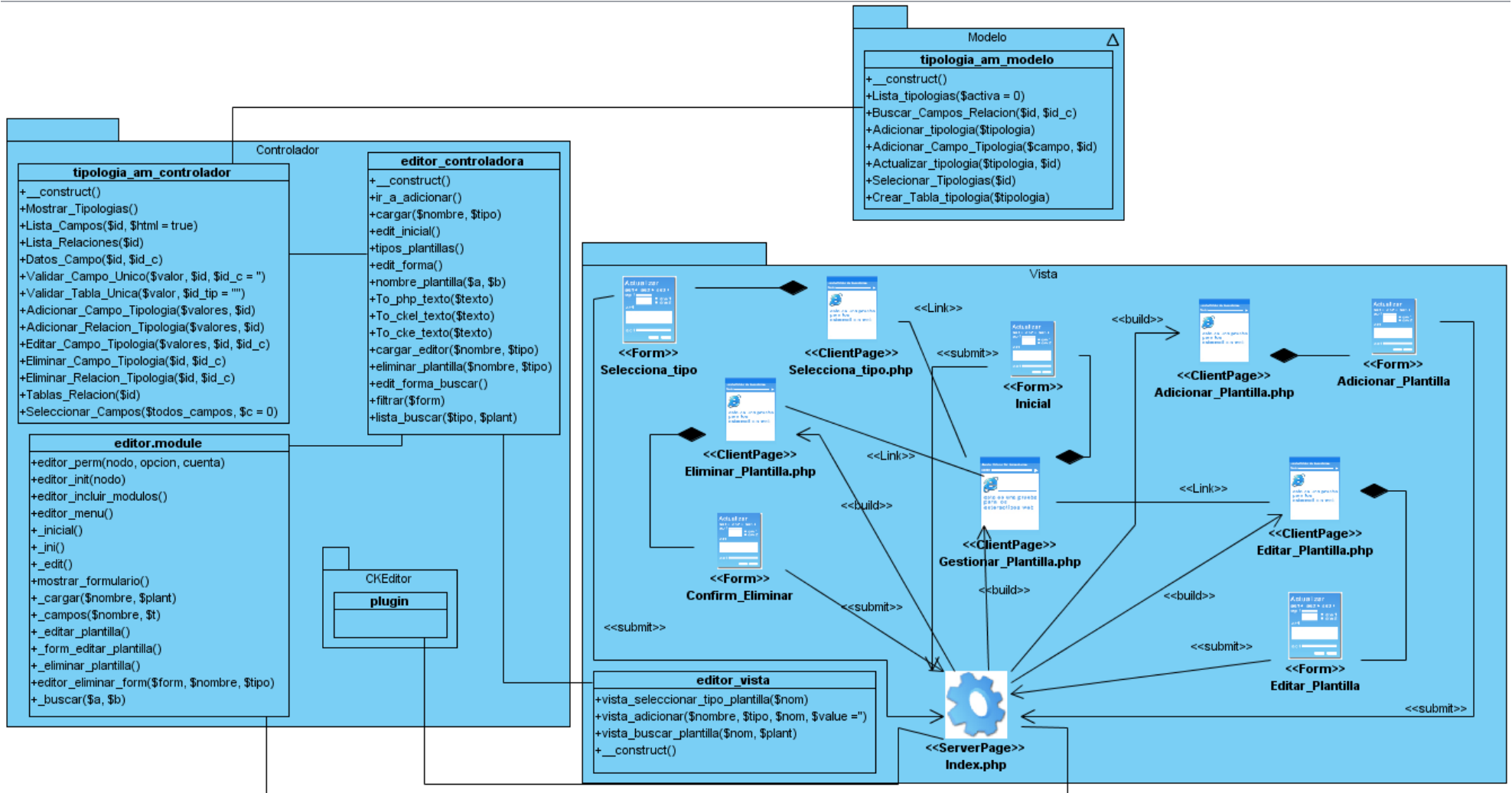


Figura 7: Diagrama de Clases del Diseño Web. CU Gestionar Plantilla

Descripción de las clases

En el diagrama de diseño se muestra la clase **index.php** del Drupal la cual se encarga de guiar la navegación en el sitio. Esta clase **index.php** busca el módulo que dé respuesta a las solicitudes hechas por los usuarios, en este caso al módulo Editor. Este módulo contiene la clase **editor.module** la cual se encarga de establecer la relación entre el módulo y el Drupal. La clase **editor.module** recibirá una solicitud y llamará a la clase **editor_controladora** la cual se encargará de gestionar la información necesaria para dar respuesta a dicha solicitud. Si fuera necesario crear un formulario para visualizarlo, la clase **editor_controladora** se encargará de solicitar este servicio a la clase **editor_vista**. La clase **editor_vista** a su vez creará todos los formularios necesarios para realizar la gestión de plantillas en la aplicación (véase Tabla 3 en esta página). Las páginas clientes que en esta misma tabla se muestran corresponderán a las páginas que creará el Drupal para visualizar las respuestas del módulo Editor. Cada una de estas clases página cliente contendrá su respectivo formulario. En el caso de que estos formularios incluyan un campo de texto se utilizará la clase **plugin**. La clase **plugin** se encargará de modificar la interfaz del CKEditor en dicho campo de texto e incluirá todas las funcionalidades necesarias para poder gestionar las plantillas. Cuando sea necesario buscar información de las tipologías en la base de datos, la clase **editor_controladora** solicitará este tipo de servicios a la clase **tipologia_am_controladora**, dicha clase llama a la clase **tipologia_am_modelo**. Esta clase **tipologia_am_modelo** se encarga de realizar todas las consultas a la tabla **tipologias_am** en la base de datos del sitio.

Tabla 3: Páginas Clientes y Formularios de la aplicación

Páginas Clientes	Formularios
Gestionar_Plantilla.php	Inicial
Adicionar_Plantilla.php	Adicionar_Plantilla
Selecciona_tipo.php	Selecciona_tipo
Editar_Plantilla.php	Editar_Plantilla
Eliminar_Plantilla.php	Confirm_Plantilla

3.8.4. Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el

modelado de una vista del negocio en cada escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos. (Ivar Jacobson, 2000)

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

A continuación se muestran los diagramas de secuencia del Caso de Uso Gestionar Plantilla el cual incluye: Adicionar, editar y eliminar plantilla. Estos diagramas reflejan el flujo normal de eventos en cada uno de los casos.

Capítulo 3: “Construcción de la Solución Propuesta”

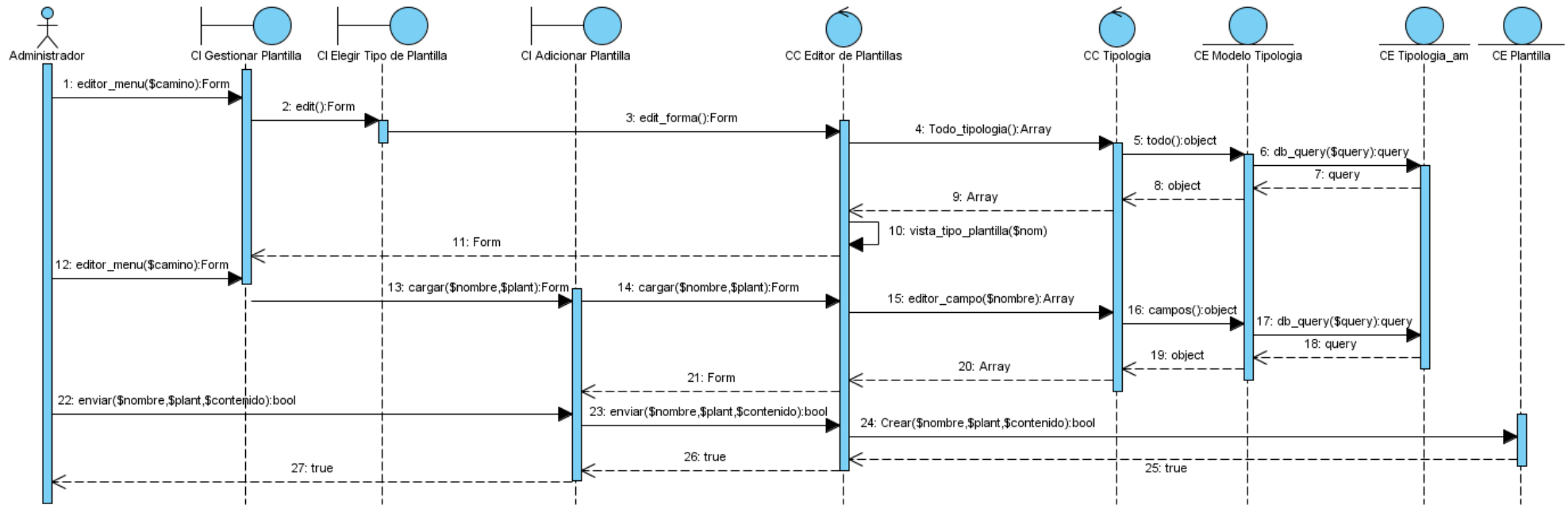


Figura 8: Diagrama de Secuencia. Adicionar Plantilla.

Capítulo 3: “Construcción de la Solución Propuesta”

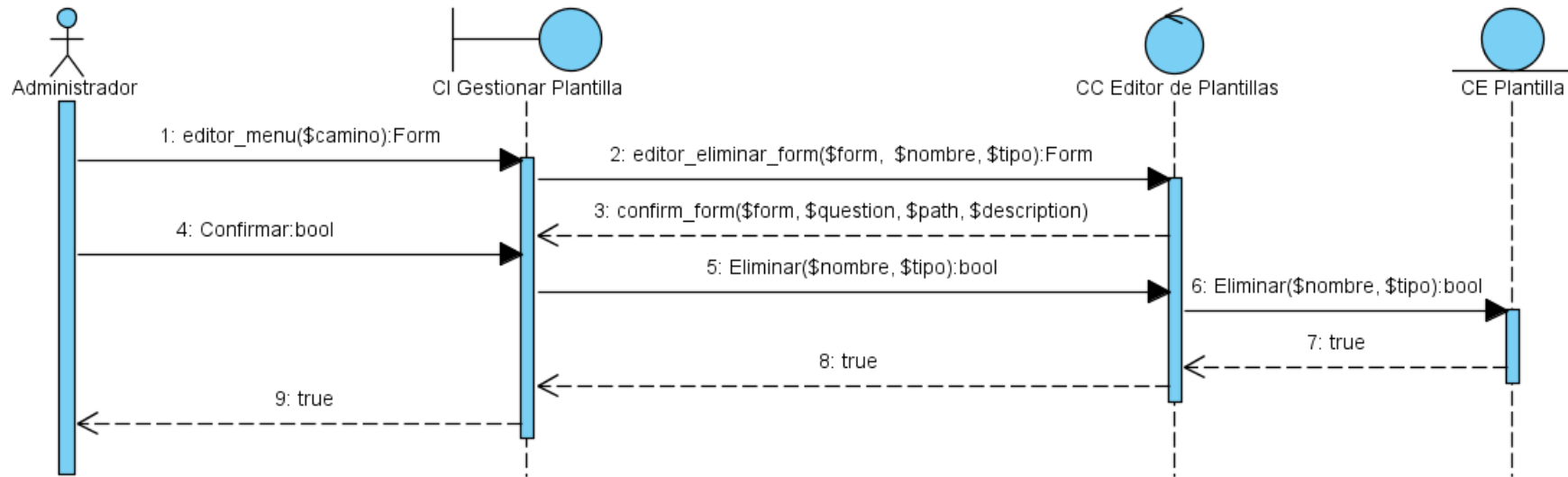


Figura 9: Diagrama de Secuencia. Eliminar Plantilla.

Capítulo 3: “Construcción de la Solución Propuesta”

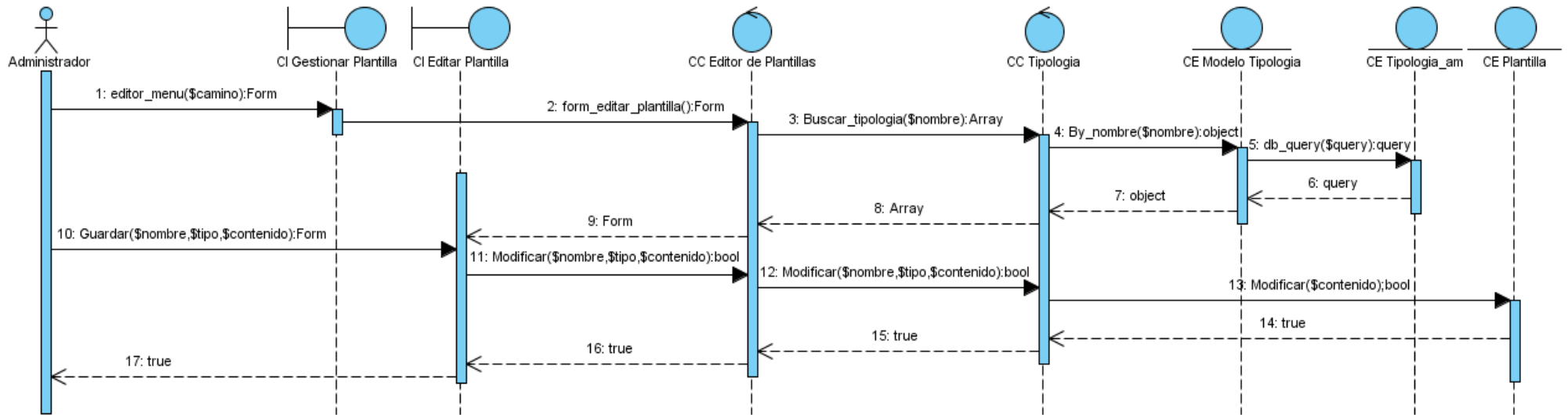


Figura 10: Diagrama de Secuencia. Editar Plantilla.

3.9. Modelo de implementación

EL modelo de implementación describe cómo los elementos del modelo de diseño y las clases se implementan en términos de componentes, como los ficheros de código fuente, ejecutables etc. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructura y modulación disponibles en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (Jacobson, 2000)

3.9.1. Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes) (Chile, 2011)

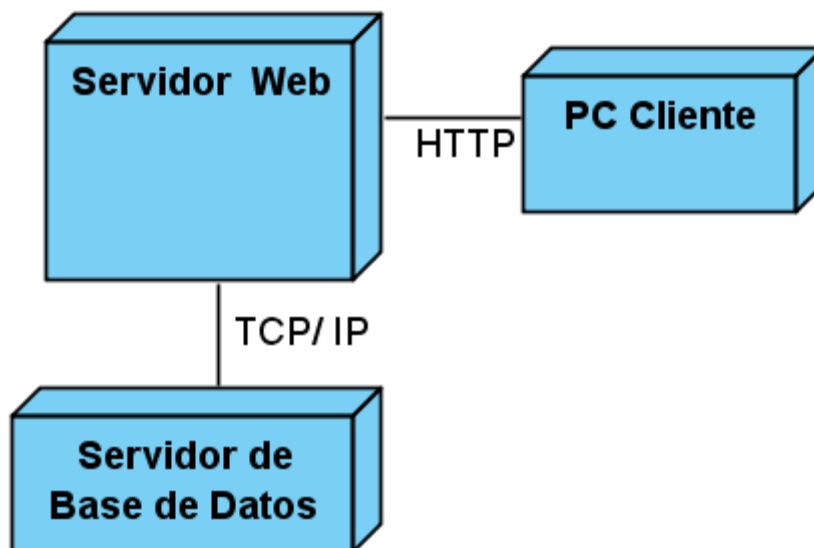


Figura 11: Diagrama de Despliegue

A continuación la descripción de los componentes y protocolos presentes en el diagrama de despliegue.

Nodos:

1. PC Cliente: Este nodo representa una PC cliente, la cual puede ser usada por un usuario para acceder a la aplicación y consumir de los servicios que se brindan en ella mediante un navegador web.
2. Servidor Aplicación Web: Este nodo constituye el servidor donde se encuentra hospedada la aplicación.
3. Servidor BD PostgreSQL: Este nodo estará ejecutando el servidor PostgreSQL con la base de datos del sistema.

Protocolos:

1. HTTP: HTTP por sus siglas en inglés HyperText Transfer Protocol las cuales traducidas al español significan Protocolo de Transferencia de Hipertexto. Este protocolo se utiliza para la web y el acceso a sitios web, cada vez que se activa cumple con un proceso de cuatro etapas entre el navegador y el servidor que consiste en lo siguiente:
 - Conexión: el navegador busca el nombre de dominio o el número IP de la dirección indicada intentando hacer contacto con esa computadora.
 - Solicitud: el navegador envía una petición al servidor (generalmente un documento), incluyendo información sobre el método a utilizar, la versión del protocolo y algunas otras especificaciones.
 - Respuesta: el servidor envía un mensaje de respuesta acerca de su petición mediante códigos de estado de tres dígitos.
 - Desconexión: se puede iniciar por parte del usuario o por parte del servidor una vez transferido un archivo.
- TCP/IP: TCP/IP por sus siglas en inglés las cuales traducidas al español significan Protocolo de Control de Transmisión / Protocolo de Internet. El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes.

3.9.2. Diagrama de Componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño.

Capítulo 3: “Construcción de la Solución Propuesta”

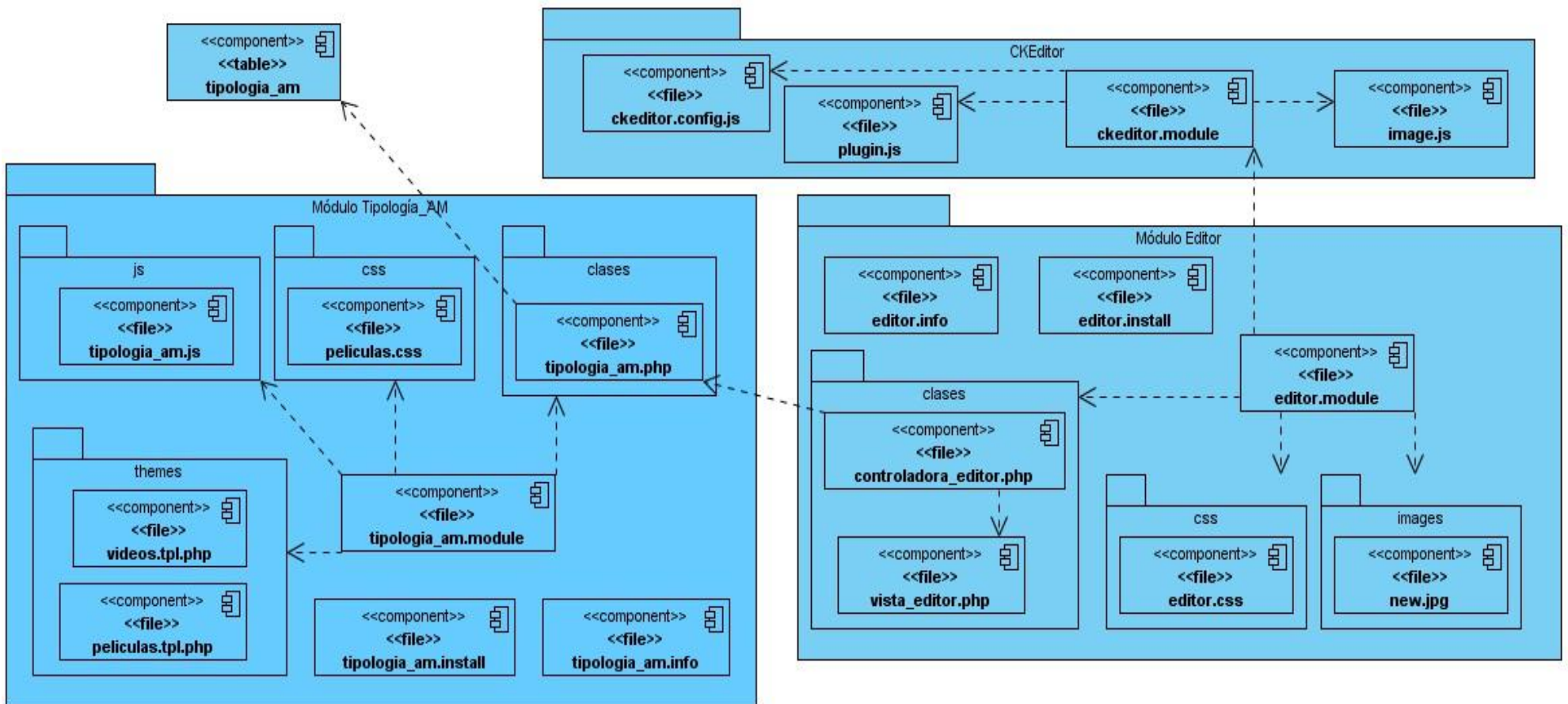


Figura 12: Diagrama de Componentes

Descripción de los componentes:

El diagrama anterior muestra que la estructura física del sistema está compuesta por estructuras estereotipadas con <<file>> que representan ficheros que contienen código fuente o datos y <<table>> esto hace referencia a una tabla de base de datos.

Paquete Módulo Editor:

- editor.info: Los ficheros “.info” de los módulos de Drupal son utilizados para brindar cierta información acerca del módulo. Este fichero solo va a ser utilizado para visualizar esta información en la interfaz donde se instalan los módulos con el fin de que los usuarios conozcan datos referentes a las utilidades del módulo. En este caso particular el fichero edito.info muestra una pequeña información acerca del módulo editor.
- editor.install: Los ficheros “.install” de los módulos de Drupal guardan la información necesaria para que el módulo sea instalado en la raíz de este CMS.
- editor.module: Los ficheros “.module” de los módulos de Drupal son los más importantes. Estos ficheros sirven para “conectar” las funcionalidades del módulo y el Drupal. En este caso el fichero editor.module es utilizado para enlazar el resto de ficheros ubicados en los otros paquetes del módulo con el Drupal.

Paquete Clases:

- vista_editor.php: Este fichero contiene el código de la clase de su mismo nombre. Esta clase construye los formularios que se muestran en las páginas del editor.
- controladora_editor.php: Este fichero contiene el código de la clase de su mismo nombre. Con esta clase se consigue cargar formularios con llamadas a la clase vista_editor. Además realiza consultas a la base de datos y otras funcionalidades que permiten que el módulo cumpla con los requerimientos definidos.

Paquete CSS:

- editor.css: contiene el código CSS utilizado en la visualización.

Paquete Images:

- new.jpg: Esta es una imagen que se utiliza en el editor para visualizarlo como botón en la interfaz del CKEditor.

Paquete CKEditor:

- `plugin.js`: Este fichero contiene el código que realiza las nuevas funcionalidades en el CKEditor para poder dar solución a los requerimientos de la investigación.
- `ckeditor.config.js`: Este fichero contiene código referente a la configuración de CKEditor. Este código fue modificado para poder agregar el nuevo plugin con el que interactúa el módulo de editor.
- `image.js`: Este es el fichero del plugin image del CKEditor. Este plugin se encarga, entre otras cosas, de insertar imagen en el cuerpo de texto que es modificado con este editor WYSIWYG. En la presente investigación con el fin de lograr su objetivo se modificó el código de este plugin.
- `ckeditor.module`: Este fichero contiene el código que permite realizar la conexión entre el módulo del CKEditor y el Drupal.

3.9.3. Conclusiones del Capítulo

Luego de realizar un estudio del entorno en el que se desenvuelve el problema se pudieron definir los actores y casos de uso del dominio y los requerimientos con que debe contar la aplicación a desarrollar. Este estudio del sistema permitió desarrollar posteriormente el modelado del análisis para poder darle forma a la solución propuesta y crear su diseño.

Con el desarrollo del modelo de implementación se pudo definir la estructura de los componentes físicos que se utilizarán. Estas actividades permiten comenzar con la construcción de la aplicación

Capítulo 4: Pruebas al Sistema

4.1. Introducción

El presente capítulo está dedicado a la realización de los pasos que define la metodología RUP para el desarrollo de software en la fase de Pruebas.

4.2. Pruebas de Software

Como se planteo anteriormente en el presente trabajo de diploma, para el desarrollo de esta investigación se utilizará como metodología de desarrollo de software RUP. Esta metodología define dentro del ciclo de vida del software la llamada “*Fase de Pruebas*” con el fin de corregir errores que podrían comenzar desde el primer momento del proceso en que los objetivos estén planteados incorrectamente hasta fases más avanzadas como son Diseño e Implementación.

Para entender mejor el significado de prueba de software se toma en la presente investigación el siguiente concepto de “*prueba*”.

Prueba, es la acción y efecto de probar (hace un examen o experimento de las cualidades de alguien o algo). Las pruebas, por lo tanto, son los ensayos que se hacen para saber cómo resultará algo en su forma definitiva, o los argumentos y medios que pretenden demostrar la verdad o falsedad de algo. (definicion.de, 2009)

Entonces para cuando se habla de pruebas de software se pueden tomar las definiciones hechas por la ingeniera Maria Clara Choucair Cárdenas en uno de sus artículos que aborda esta temática y por el conjunto de autores del artículo “Estudios de la Ingeniería de Software y Base de Datos”.

La Ing. Maria Clara Choucair plantea referente al proceso de pruebas de software: “Proceso realizado concurrentemente a través de las diferentes etapas de desarrollo de software que utiliza y mantiene el testware y cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación.” (Cárdenas, 2007)

Por su parte el conjunto de autores del artículo “Estudios de la Ingeniería de Software y Base de Datos” plantean que la prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Una prueba tiene éxito si descubre un error no detectado hasta entonces. (Beatriz Pérez Lamancha, 2005)

Realizando un análisis de los conceptos planteados anteriormente se puede concluir entonces diciendo que las pruebas de software definitivamente son imprescindibles para una mejor calidad en la terminación de un producto. Para acercarse a un software sin errores se debe ante todo tener un conocimiento sobre las pruebas del software y lograr escoger cuales técnicas deben aplicarse según se corresponda. Las pruebas del software deben sostenerse en las diferentes etapas de desarrollo del software para controlar las deficiencias y detectarlas en el tiempo requerido, así luego corregir, ahorrando tiempo y costo.

La etapa de Pruebas del Software implica:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

4.2.1. Objetivos de las pruebas de software

Pressman para definir los objetivos de las pruebas hace referencia a Glem Myers quien establece algunos atributos que pueden servir para definir los objetivos de las mismas:

1. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Como parte que es de un proceso industrial, la fase de pruebas añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre; ahí reside su valor. El objetivo específico de la fase de pruebas es encontrar la mayor cantidad posible de errores, con un mínimo de costo y tiempo. (Luis Alberto Mañas Arzuaga, 2006)

Un objetivo de prueba es un conjunto de interacciones entre el sistema y un caso de prueba (que reemplaza a un actor) para verificar que el comportamiento del sistema es el comportamiento definido en sus casos de uso. Los objetivos de pruebas se definen como caminos a través del modelo de

comportamiento o como diagramas de actividades dónde sólo existe un único camino (Javier J. Gutiérrez, 2006)

Luego de analizar los planteamientos anteriores se puede decir que los objetivos principales de las pruebas son: tener un buen caso de prueba y descubrir un error no descubierto antes. Los métodos de prueba del software tienen el objetivo de diseñar las pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Siempre que se cumplan los objetivos de las “pruebas de software” se puede ver la calidad del producto una vez terminado.

4.2.2. Estrategia de pruebas para aplicaciones web

La estrategia para realizarle pruebas a una aplicación web recoge un grupo de principios básicos que están implícitos dentro de las pruebas de software en sentido general. No obstante este tipo de aplicación reúne un grupo de pruebas que son típicas de este tipo de software. Dentro de las principales actividades que se deben tener en cuenta en esta tarea los siguientes pasos proponen el enfoque a llevar.

1. Se revisa el modelo de contenido de la aplicación web para descubrir errores.
2. Se revisa el modelo de la interfaz para asegurarse de que todos los casos de uso pueden acomodarse.
3. Se revisa el modelo de diseño de la aplicación web para descubrir errores de navegación.
4. Se prueba la interfaz del usuario para descubrir errores en la presentación o los mecanismos de navegación.
5. Componentes funcionales seleccionados se prueban en forma individual.
6. Se prueba la navegación a través de toda la arquitectura.
7. La aplicación web se implementa en diversas configuraciones ambientales y se prueba su compatibilidad con cada configuración.
8. Se realizan pruebas de seguridad con el objetivo de explotar vulnerabilidades en la aplicación web o dentro de su ambiente. (Pressman, 2005)

Todos estos pasos constituyen operaciones, tipos de pruebas y elementos a considerar a la hora de realizarle pruebas a la aplicación. Si se logran cubrir todos estos pasos con las pruebas que se realicen a la aplicación se podrá garantizar la calidad del software.

4.2.3. Técnicas de Pruebas de Software. Pruebas de Caja Negra

Las técnicas de pruebas permitirán examinar los detalles procedimentales del código y la lógica del programa así como la interfaz y su funcionalidad. Por medio de las diferentes técnicas se brindan criterios variados para generar casos de pruebas que provoquen fallos en los programas. Es por ello que es de vital importancia conocer dichas técnicas.

Cualquier producto de ingeniería se puede probar de dos formas:

1. Pruebas de caja negra: Realizar pruebas de forma que se compruebe que cada función es operativa.
2. Pruebas de caja blanca: Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

Pruebas de Caja Negra

Para la presente investigación se ha decidido realizar pruebas de Caja Negra debido a que lo más importante para este proyecto es que las funcionalidades trabajen correctamente.

El método de caja negra se enfoca en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa. No considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. (Velthuis, 2006)

Con este equipo de pruebas se intenta encontrar:

19. Funciones incorrectas o ausentes.
20. Errores de interfaz.
21. Errores en estructuras de datos o en accesos a la bases de datos externas
22. Errores de rendimiento.
23. Errores de inicialización y terminación.

Los casos de prueba de la caja negra pretende demostrar que:

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

4.2.4. Casos de pruebas

Como se planteó en el epígrafe anterior en la presente investigación se realizarán pruebas de Caja Negra con el fin de medir la fortaleza y robustez de la aplicación, rectificar posibles errores que sean detectados y así aumentar la calidad de la aplicación. Pero como se materializa la integración entre los casos de pruebas y las pruebas de caja negra.

El grupo de autores del libro “El proceso unificado de desarrollo” plantean al respecto que: “Un caso de prueba que especifica como probar un caso de uso o un escenario específico de un caso de uso incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y postcondiciones especificadas por el caso de uso y que se sigue la secuencia de pasos especificadas por el caso de uso.”. Un caso de prueba basado en un caso de uso especifica típicamente una prueba del sistema como caja negra. (Ivar Jacobson, 2000)

Se puede plantear entonces que durante las pruebas al sistema se analizará el flujo de eventos definidos para los casos de uso de la aplicación según se muestran en la descripción de los casos de uso. Así se podrá corroborar que las funcionalidades cumplen con lo establecido y satisface los requisitos de la investigación.

4.2.5. Pruebas de Caja Negra. Caso de uso Gestionar Plantilla

Este caso de uso se inicia cuando el usuario desde la página gestionar plantilla desea adicionar, eliminar o editar plantillas en el sistema. La única condición de ejecución de este caso de uso está que el usuario está autenticado como administrador.

Sesiones a probar en el caso de uso

Tabla 4: Sesiones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Adicionar nueva plantilla.	EC 1.1: Seleccionar tipo de plantilla con éxito.	El sistema muestra el formulario seleccionar tipo de plantilla con una lista con todos los tipos de plantillas y todas las tipologías de archivo multimedia para conocer que plantilla desea crear el administrador. El administrador selecciona la tipología y el tipo de plantilla y oprime el botón “Enviar”. El sistema muestra la interfaz adicionar plantilla.

	EC 1.2: Seleccionar tipo de plantilla. Cancelar	El administrador presiona el vínculo “Cancelar”, de la interfaz seleccionar tipo de plantilla. El sistema cancela la operación y vuelve a la interfaz gestionar plantillas.
	EC 1.3: Adicionar plantilla con éxito.	El sistema muestra un formulario con los datos de la tipología seleccionada, el editor de texto CKEditor para realizar el diseño de la plantilla y un botón “Guardar”. Luego de realizar el diseño de la plantilla el administrador presiona el botón “Guardar” y el sistema crea un fichero físico el cual constituye la plantilla PHP.
	EC 1.4: Adicionar plantilla. Cancelar.	El administrador presiona el vínculo “Cancelar”, de la interfaz adicionar plantilla, el sistema cancela la operación y vuelve a la interfaz gestionar plantillas.
SC 2 Eliminar plantilla.	EC 2.1 Eliminar plantilla con éxito.	El administrador selecciona el vínculo “Eliminar” de alguna de las plantillas que se muestran en la página gestiona plantilla. El sistema muestra la interfaz eliminar plantilla. El administrador presiona el botón “Confirmar” y el sistema elimina la plantilla y muestra el mensaje “Se eliminó correctamente”.
	EC 2.2 Eliminar plantilla. Cancelar.	El administrador selecciona la opción “Cancelar” de la interfaz eliminar plantilla. El sistema cancela la operación y vuelve a la página gestionar plantilla.
SC 3 Editar plantilla.	EC 3.1 Editar plantilla con éxito.	El administrador selecciona el vínculo “Editar” de alguna de las plantillas que se muestran en la página gestiona plantilla. El sistema muestra la interfaz editar plantilla donde está un listado de los campos de la tipología a la que pertenece la plantilla, un cuadro de texto con el CKEditor donde se cargan los valores que tenía esta plantilla y un botón “Guardar”. El administrador modifica el contenido del campo de texto y presiona el botón “Guardar”, el sistema guarda la plantilla con los cambios realizados.

	EC 3.1 Editar plantilla. Cancelar.	El administrador selecciona la opción “Cancelar” de la interfaz eliminar plantilla. El sistema cancela la operación y vuelve a la página gestionar plantilla.
--	---------------------------------------	---

Descripción de las variables

Tabla 5: Descripción de las variables

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Seleccionar tipología	Campo de selección	No	El campo no puede quedar vacío, es necesario seleccionar una tipología.
2-	Tipo de plantilla	Campo de selección	No	El campo no puede quedar vacío, debe seleccionarse un tipo de plantilla
3-	Área de texto con editor CKEditor	Campo de texto	No	El campo puede quedar vacío, el diseño realizado por el administrador puede ser como él desee según las posibilidades que le permita el CKEditor con sus funciones.

Matriz de datos

a) SC 1: Adicionar nueva plantilla

Tabla 6: Matriz de datos. SC 1 Adicionar nueva plantilla

Escenario	Selección tipología	Seleccionar tipo de plantilla	Vínculo “Cancelar”	Respuesta del Sistema	Resultado de la Prueba	Flujo Central

EC 1.1: Seleccionar tipo de plantilla con éxito.	V	V	I	Se procede a seleccionar tipo de plantilla y tipología de la plantilla	El escenario se realiza satisfactoriamente	El administrador elige una tipología y un tipo de plantilla oprime el botón “Enviar” y muestra la interfaz adicionar plantilla.
EC 1.2: Seleccionar tipo de plantilla. Cancelar	No es necesario	No es necesario	V	El sistema cancela la operación y vuelve a la interfaz gestionar	El escenario se realiza satisfactoriamente	El administrador selecciona la opción “Cancelar”.

Escenario	Datos de la tipología	Vínculo “Cancelar”	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 1.3: Adicionar plantilla con éxito.	No es obligatorio	I	Se procede a adicionar la plantilla permitiendo que el administrador opere con las funcionalidades del CKEditor correctamente	El escenario se realiza satisfactoriamente	El administrador desde la interfaz adicionar plantilla realiza un diseño de la plantilla, adiciona a su consideración si desea introducir “Datos de la tipología” (columnas de la tabla) y oprime el botón “Guardar”. La plantilla se crea y es almacenada satisfactoriamente.
EC 1.4: Adicionar plantilla. Cancelar	No es obligatorio	V	El sistema cancela la operación y vuelve a la interfaz gestionar	El escenario se realiza satisfactoriamente	El administrador selecciona la opción “Cancelar”.

Seleccionar Plantilla.			plantillas.	mente	
------------------------	--	--	-------------	-------	--

b) SC 2: Eliminar plantilla

Tabla 7: Matriz de datos. SC 1 Eliminar plantilla

Escenario	Botón: “Confirmar”	Vínculo: “Cancelar”	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Eliminar Plantilla	V	I	El sistema elimina la plantilla y muestra un mensaje de confirmación: “Se eliminó correctamente”	El escenario se realiza satisfactoriamente	El administrador selecciona la opción “Confirmar”.
EC 2.2: Eliminar plantilla. Cancelar.	I	V	El sistema cancela la operación y vuelve a la interfaz gestionar plantillas.	El escenario se realiza satisfactoriamente	El administrador selecciona la opción “Cancelar”..

c) SC 3: Editar plantilla

Tabla 8: Matriz de datos. SC 1 Editar plantilla

Escenario	Datos de la tipología	Vínculo: “Cancelar”	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
-----------	-----------------------	---------------------	-----------------------	------------------------	---------------

EC Adicionar plantilla con éxito.	1.1:	No es obligatorio	I	Se procede a editar los valores de la plantilla que hay en el campo de texto permitiendo que el administrador opere con las funcionalidades del CKEditor correctamente	El escenario se realiza satisfactoriamente	El administrador desde la interfaz adicionar plantilla realiza un diseño de la plantilla, adiciona a su consideración si desea introducir “Datos de la tipología” (columna dos de la tabla) y oprime el botón “Guardar”. La plantilla se crea y es almacenada satisfactoriamente.
EC Adicionar plantilla. Cancelar Seleccionar Plantilla.	1.2:	No es obligatorio	V	El sistema cancela la operación y vuelve a la interfaz gestionar plantillas.	El escenario se realiza satisfactoriamente	El administrador selecciona la opción “Cancelar”.

Análisis de los resultados obtenidos

A partir del análisis de los resultados obtenidos de los casos de prueba se pudo comprobar que la aplicación realiza correctamente las funcionalidades y logra dar respuesta a los requisitos definidos en el capítulo 3 del documento de la presente investigación. En todos los escenarios se puede operar con la aplicación sin problemas, siguiendo los pasos que especifican en la descripción de los caso de uso.

4.2.6. Conclusiones del capítulo

Con la realización de este capítulo se pudo conocer sobre la importancia que tienen las pruebas durante el ciclo de vida del software y se abordaron algunos tipos de pruebas que se pueden realizar a la aplicación. Con la realización de esta tarea se pudo dar paso a realizarles pruebas al software, estas arrojaron resultados positivos.

Conclusiones Generales

El empleo de métodos científicos facilitó el conocimiento del estado del objeto de estudio y así se pudo abordar algunos conceptos asociados al tema investigativo. Con el análisis de estos conceptos se pudo ir de lo general a lo particular desglosando la situación actual que rodea la problemática planteada. Así se conoció de las tecnologías existentes cuales de ellas eran convenientes para utilizar en el presente trabajo. De esta manera se estableció el CKEditor como módulo de Drupal a utilizar para realizar la aplicación de la investigación, además se pudo definir la metodología y otras tecnologías de desarrollo.

Con el estudio del dominio del problema se pudo conocer como estaban estructuradas las clases conceptuales que interactuaban en él y así elaborar el modelo de dominio. Con este modelo se conoció mejor el funcionamiento del negocio en torno a la problemática y se pudo establecer los cambios necesarios para dar solución a dicha problemática. Una vez definido el modelo de dominio y los requisitos se pudo dar un enfoque más práctico y de esta manera se pudo conformar el modelado del sistema. Este modelo de sistema permitió refinar los requisitos para llevarlos a casos de uso.

Con la elaboración del modelo de análisis y diseño se pudo dar forma a la aplicación necesaria para dar solución a la problemática existente. Esta actividad se desarrolló a partir del diseño de casos de uso realizada en el modelado del sistema y haciendo uso de patrones arquitectónicos y de diseño. Estos patrones sirvieron para que la aplicación diseñada cumpliera las normativas más avanzadas que se proponen para desarrollar aplicaciones a nivel mundial por parte de las grandes empresas del software.

Con los modelos de análisis y de diseño se pudo dar paso a la implementación de la aplicación y así construir el modelo de implementación. Con este modelo se recogieron los diagramas que describen la estructura de los nodos y ficheros físicos con que se trabajaron durante dicha implementación. Además se pudieron automatizar las funcionalidades recogidas en el modelado del sistema haciendo uso del diseño definido en el modelado del diseño. Con la implementación terminada se pudo probar el sistema. Estas pruebas permitieron corregir los problemas encontrados a la aplicación y así conseguir que se obtuviera un producto robusto y con buena calidad.

Con la implementación de este componente se logró crear un editor a partir del módulo del CKEditor. Este nuevo módulo mantiene todas las funcionalidades típicas del CKEditor. Además incluye una nueva funcionalidad la cual permite introducir las variables PHP correspondientes a los metadatos de los archivos multimedia que se gestionan en el proyecto Plataforma VideoWeb. De esta manera se podrá

diseñar de una manera más dinámica las plantillas necesarias para publicar estos archivos multimedia. Con esto se podrá librar a los administradores del sitio de realizar la codificación de las plantillas.

Recomendaciones

Después de realizar la siguiente investigación se plantean como recomendaciones:

Vincular el editor con la sección de publicación de las medias. Esto se refiere a una funcionalidad que permita editar plantillas desde una publicación específica. En este caso el editor debe cargar los metadatos de esa media, y guardar el diseño realizado para el tipo de media en cada caso.

Se propone además mejorar las potencialidades del editor con funcionalidades que permitan un posicionamiento más libre de los componentes que se inserten en el editor. También se podrían agregar funcionalidades que permitan incluir archivos CCS para introducir estilos en las plantillas.

Bibliografía Citada

C.J.Date Introducción a los sistemas de Base de Datos [Libro]. - México : PEARSON EDUCACIÓN, 2001.

Definición.org Definición.org [En línea] // Definición.org. - 2010. - 13 de 11 de 2010. - <http://www.definicion.org/sitio web>.

Ivar Jacobson Grady Booch y James Rumbaugh El Proceso Unificado de Desarrollo de Software [Libro]. - Madrid : Addison Wesley, 2000.

Pérez Javier Eguíluz eva.uci.cu [En línea] // Introducción a XHTML. - 2008. - 10 de enero de 2011. - http://eva.uci.cu/file.php/452/Bibliografia_Basica/introduccion_xhtml_2caras.pdf.

Pérez Javier Eguíluz eva.uci.cu [En línea] // Introducción a CSS. - 2009. - 10 de enero de 2011. - http://eva.uci.cu/file.php/452/Bibliografia_Basica/introduccion_css.pdf.

Pressman Roger Ingeniería de Software, un enfoque práctico [Libro]. - New York : MacGraw-Hill, 2003.

R.M.M.García Lic. Diseño de Base de datos [Libro]. - 1999.

Bibliografía Consultada

Alvarez Miguel Angel desarrolloweb.com [En línea] // desarrolloweb.com. - DesarrolloWeb.com, 2001. - 29 de 11 de 2010. - <http://www.desarrolloweb.com/articulos/que-es-html.html>.

Alvarez Miguel Angel desarrolloweb.com [En línea] // desarrolloweb.com. - DesarrolloWeb.com, 25 de marzo de 2009. - 29 de 11 de 2010. - <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.

Alvarez Miguel Angel Historia del XML [En línea]. - desarrolloweb.com, 14 de 8 de 2001. - 10 de 5 de 2011. - <http://www.desarrolloweb.com/articulos/450.php>.

Alvarez Miguel Angel Qué es XHTML? [En línea]. - desarrolloweb.com, 10 de enero de 2011. - 10 de mayo de 2011. - <http://www.desarrolloweb.com/articulos/que-es-xhtml.html>.

Aportela Rodríguez Intranets, I. M. Las tecnologías de información y comunicación en función de la organización [Informe]. - 2007.

Austerberry David The Technology of Video and Audio Streaming Second Edition [Informe]. - 2003.

Barry M. Leiner Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Lawrence G. Roberts, Stephen Wolff Una breve historia de internet [Informe]. - 2003.

Beatriz Pérez Lamancha Macario Polo Usaola Estudios de la Ingeniería de Software y Base de Datos. [Conferencia] // Estudios de la Ingeniería de Software y Base de Datos.. - Buenos Aires : Editorial Universitaria, 2005.

C.J.Date Introducción a los sistemas de Base de Datos [Libro]. - México : PEARSON EDUCACIÓN, 2001.

Caraballo Pérez Yeter La gestión de contenidos en portales Web [En línea]. - 2007. - 11 de enero de 2011. - ftp://bvs.sld.cu/revistas/aci/vol15_3_07/aci07307.htm..

Cárdenas Maria Clara Choucair Pruebas de software ¿la salvación, un proceso sin utilidad, trivial, simplemente una moda, o? [Informe]. - Bogota. Colombia : Biblioteca Luis Angel Arango, 2007.

Chile Universidad de Centro de Estudios Universidad de Chile [En línea]. - 2011. - 13 de marzo de 2011. - <http://www.dcc.uchile.cl/~luguerre/cc61j/re>.

definicion.de Definicion.de [En línea]. - Definicion.de, 24 de mayo de 2009. - 5 de mayo de 2011. - <http://definicion.de/prueba/>.

Definición.de Definición.de [En línea] // Definición de Metodología. - 2011. - 14 de enero de 2011. - <http://definicion.de/metodologia/>.

Definición.org Definición.org [En línea] // Definición.org. - 2010. - 13 de 11 de 2010. - <http://www.definicion.org/sitio web>.

Drupal Sitio oficial de Drupal.org [En línea] // Drupal.org. - Drupal.org, 2010. - 12 de 10 de 2010. - <http://drupal.org.es/caracteristicas>.

ECURED La enciclopedia de Cuba [En línea]. - 2005. - 10 de junio de 2011. - http://www.ecured.cu/index.php/Patrones_de_Casos_de_Uso#Patr.C3.B3n_CRUD.

Gracia Joaquin Patrones de diseño.Análisis y Diseño. Ingeniería de Software. [En línea]. - 27 de 5 de 2005. - <http://www.ingenierosoftware.com/analisisydiseño/patrones-diseno.php>.

guia-ubuntu.org guia-ubuntu.org [En línea]. - 2009. - 20 de enero de 2011. - http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

International Visual Paradigm Visual Paradigm International [En línea]. - Visual Paradigm, 2007. - 15 de enero de 2011. - <http://www.visual-paradigm.com/product/vpuml/communityedition.jsp..>

Ivar Jacobson Grady Booch y James Rumbaugh El Proceso Unificado de Desarrollo de Software [Libro]. - Madrid : Addison Wesley, 2000.

Jacobson Ivar, Booch, Grady y Rumbaugh, James El proceso unificado de desarrollo de [Libro]. - Madrid : Pearson Education S.A, 2000.

Javier J. Gutiérrez María J. Escalona, Manuel Mejías y Jesús Torres Modelos y algoritmos para la generación de objetivos de prueba [Informe]. - Universidad de Sevilla : Escuela Técnica Superior de Ingeniería Informática, 2006.

Lazaro Juliana Monteiro desarrolloweb.com [En línea] // desarrolloweb.com. - DesarrolloWeb.com, 2009. - 29 de 11 de 2010. - <http://www.desarrolloweb.com/articulos/26.php>.

Luis Alberto Mañas Arzuaga Enrique Alonso Ruseiro, Lic Marsela Capiña Santos Mitenconologico.com [En línea]. - 12 de enero de 2006. - 2 de mayo de 2011. - <http://www.mitecnologico.com/Main/GenerarUnPlanDePruebas>.

Malvaez Alejandro NetBeans un Entorno de Desarrollo Integrado [En línea]. - 2008. - 20 de enero de 2011. - http://blogs.sun.com/AlejandroMalvaez/entry/netbeans_un_entorno_de_desarrollo..

netbeans.org netbeans.org [En línea]. - 2011. - 20 de enero de 2011. - http://netbeans.org/community/releases/61/index_es.html.

Pérez Javier Eguíluz eva.uci.cu [En línea] // Introducción a XHTML. - 2008. - 10 de enero de 2011. - http://eva.uci.cu/file.php/452/Bibliografia_Basica/introduccion_xhtml_2caras.pdf.

- Pérez Javier Eguíluz** eva.uci.cu [En línea] // Introducción a CSS. - 2009. - 10 de enero de 2011. - http://eva.uci.cu/file.php/452/Bibliografia_Basica/introduccion_css.pdf.
- Perissé Marcelo Claudio.** Proyecto Informático Una Metodologías Simplificada [En línea] // Ciencia y Técnica Administrativa. - 2005. - 16 de enero de 2011. - <http://cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/>. 987-43-2947-5..
- Pressman Roger** Ingeniería de Software, un enfoque práctico [Libro]. - New York : MacGraw-Hill, 2003.
- Pressman Roger S.** Ingeniería de Software [Sección del libro] // Ingeniería de Software. Un enfoque práctico.. - [s.l.] : Mc Graw Hill, 2001.
- Pressman ROger S.** Ingeniería de Software. Un enfoque práctico 6ta Edición [Libro]. - 2005.
- R.M.M.García Lic.** Diseño de Base de datos [Libro]. - 1999.
- Sarduy Domínguez Yanetsys y Urra González, Pedro** Sistemas de gestión de contenidos: En busca de una plataforma ideal [En línea]. - 2006.. - 12 de enero de 2011. - http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm..
- Schmuller Joseph** Aprendiendo UML en 24 Horas [Libro]. - México : PEARSON EDUCATION, 2000. - Vols. 968-444-463-X.
- Sitio Oficial de Javascript** About.com Javascript [En línea] // About.com Javascript. - Corporación de Javascript, 2007. - 29 de 11 de 2010. - <http://javascript.about.com>.
- Studios Java** Scribd.com [En línea]. - 2004. - 10 de junio de 2011. - <http://es.scribd.com/doc/52009826/18/Modelo-Vista-Controlador>.
- Tedeschi Ing. Nicolás** Microsoft.com [En línea]. - Microsoft.com, 12 de abril de 2011. - 16 de mayo de 2011. - <http://msdn.microsoft.com/es-es/library/bb972240.aspx#XSLTsection123121120120>.
- Together Borland** SlideShare Inc. [En línea]. - Borland Together - Administración Bases de Datos, 2005. - 15 de enero de 2011. - <http://www.slideshare.net/unimauro/borland-together-administracion-bases-de-datos..>
- Velthuis Macario Polo Usaola y Mario Piattini** AUTOMATIZACIÓN DEL PROCESO DE PRUEBAS UNITARIAS [Informe]. - Escuela Superior de Informática Universidad de Castilla-La Mancha : PRIS 2006, 2006.
- www.postgresql.net** postgresql.net [En línea]. - 2011. - 20 de enero de 2011. - [www.potgresql.net](http://www.postgresql.net).

Anexos

Anexo 1. Glosario de termino:

--C--

CASE: Por sus siglas en ingles Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering).

CMS: Por sus siglas en ingles Sistema Gestor de Contenido (Content Management System).

--F--

Framework: Marco de trabajo, generalmente brinda funcionalidades para agilizar el desarrollo de programas informáticos.

--G--

GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GPL (General Public License).

--H--

Hardware: parte física de un computador y más ampliamente de cualquier dispositivo electrónico

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto).

--I--

Interfaz: Zona de contacto o conexión entre dos componentes de "hardware"; entre dos aplicaciones, o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

Internet: Conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

--M—

Modular: La modularidad es la capacidad que tiene un sistema de funcionar con la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común. Cada una de esas partes en que se encuentre dividido el sistema recibe el nombre de módulo. Idealmente un módulo debe poder cumplir las condiciones de caja negra, es decir, ser independiente del resto de los módulos y comunicarse con ellos (con todos o sólo con una parte) a través de unas entradas y salidas bien definidas.

MySQL: El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo

masivo de software. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. MySQL es una marca registrada de MySQL AB.

--N--

Navegabilidad: La navegabilidad o navegabilidad web es la facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sitio web.

--S--

Software: Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.

SQL: El Lenguaje de consulta estructurado o SQL (Structured Query Language), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

--T--

Tipología: Término utilizado en el proyecto Plataforma VideoWeb para definir grupos de archivos multimedia que son de un mismo tipo.

--U--

URL: Es el acrónimo de Uniform Resource Locator y especifica cómo encontrar un recurso en internet. En realidad, un localizador de recursos uniforme está compuesto por cuatro partes: el servicio o protocolo, el nombre del servidor, el recorrido y el nombre del documento o file. También se puede encontrar en algunas bibliografías como Universal Resource Locator (Localizador Universal de Recursos).

--W--

Web: El término se utiliza para definir el universo del World Wide Web, los sitios, la información y los servicios de la telaraña.

WYSIWYG: What You See Is What You Get por sus siglas en inglés el cual traducido al español significa lo que ves es lo que obtienes.

--X--

XML: Siglas en inglés de Extensible Markup Language (Lenguaje de Etiquetado Extensible).

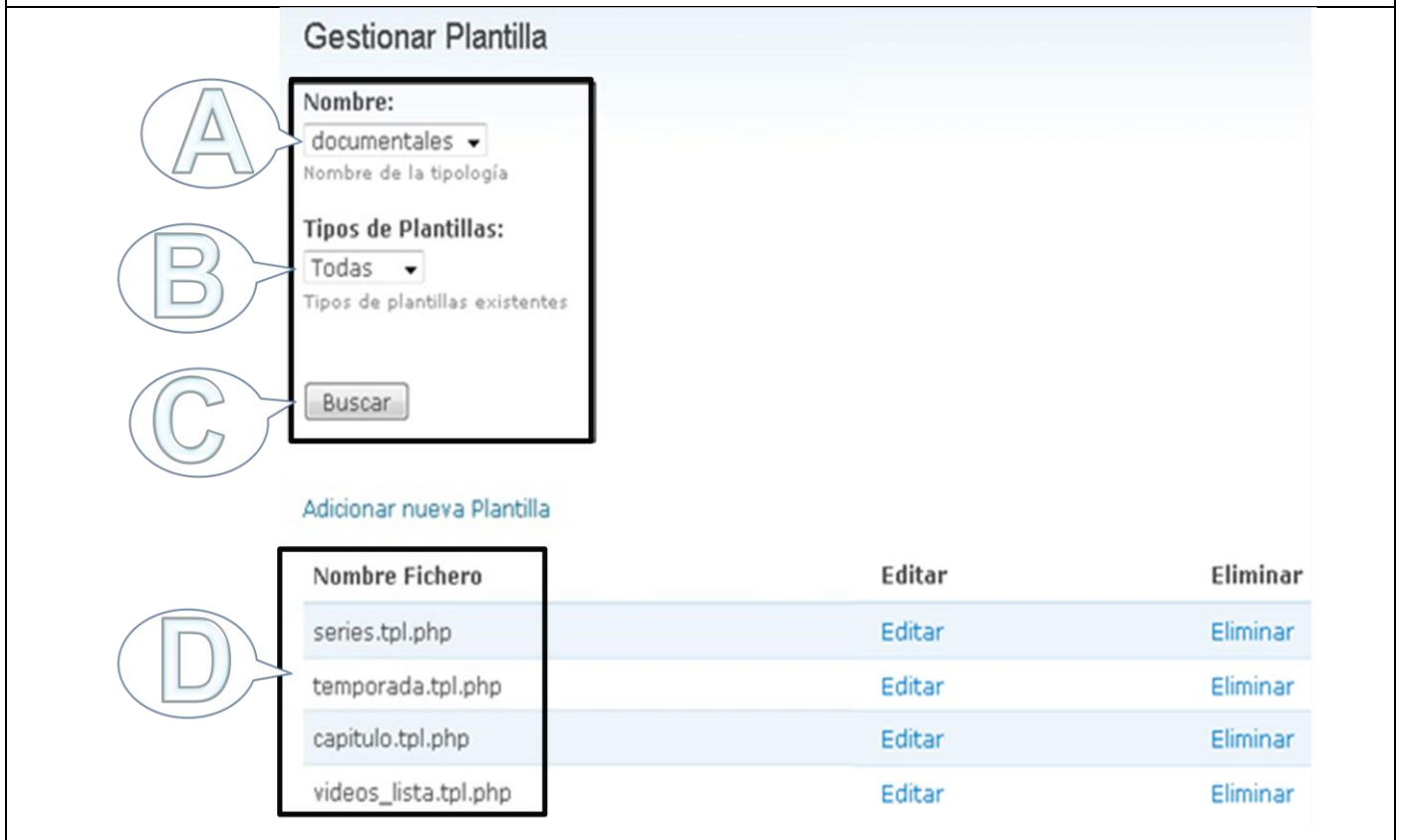
Anexo 2. Descripción del Caso de Uso Buscar Plantilla

Tabla 9: Descripción del Caso de Uso Buscar Plantilla.

Caso de uso	Buscar Plantillas	
Actores	Administrador	
Resumen	Este caso se inicia cuando el Administrador desde la opción de Gestionar Plantilla, en el menú de trabajo, podrá buscar una o un grupo de plantillas. El caso de uso termina cuando se muestre en una lista la, o las plantillas o cuando se muestre un mensaje de que no se encontraron plantillas asociadas a la búsqueda.	
Precondiciones	Se ha autenticado como administrador.	
Referencia	RF 2	
Prioridad	Crítico	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador presiona el vínculo “Gestionar plantilla” de su interfaz de trabajo.	2. El sistema muestra la interfaz Gestionar Plantilla la cual cuenta con un formulario con una lista de las tipologías (A) y de los tipos de plantillas (B) existentes, además tiene un botón “Buscar” (C). Además muestra un listado de todas las plantillas existentes (D) hasta ese momento y componentes corresponden al CU Gestionar Plantilla (ver en la página 39)	
3. El administrador realiza una selección de la tipología que desea buscar (A) y del tipo de plantilla (B) y selecciona la opción “Buscar” (C).	4. El sistema muestra la interfaz Gestionar Plantilla con el único cambio de que la lista de plantillas (D) va a mostrar las plantillas resultantes de la búsqueda.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	4. El sistema muestra la interfaz Gestionar	

Plantilla con lista de plantillas (D) vacía. Además muestra un mensaje de error en caso de que la búsqueda no haya tenido resultado.

Prototipo de Interfaz Gestiona Plantilla (porción correspondiente al CU Buscar Plantilla)



Anexo 3. Diagrama de Clases del Análisis. CU Buscar Plantilla

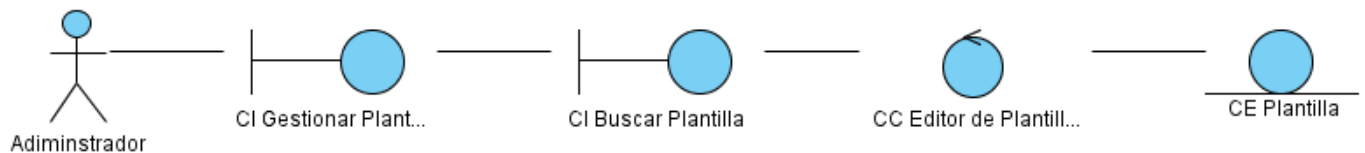
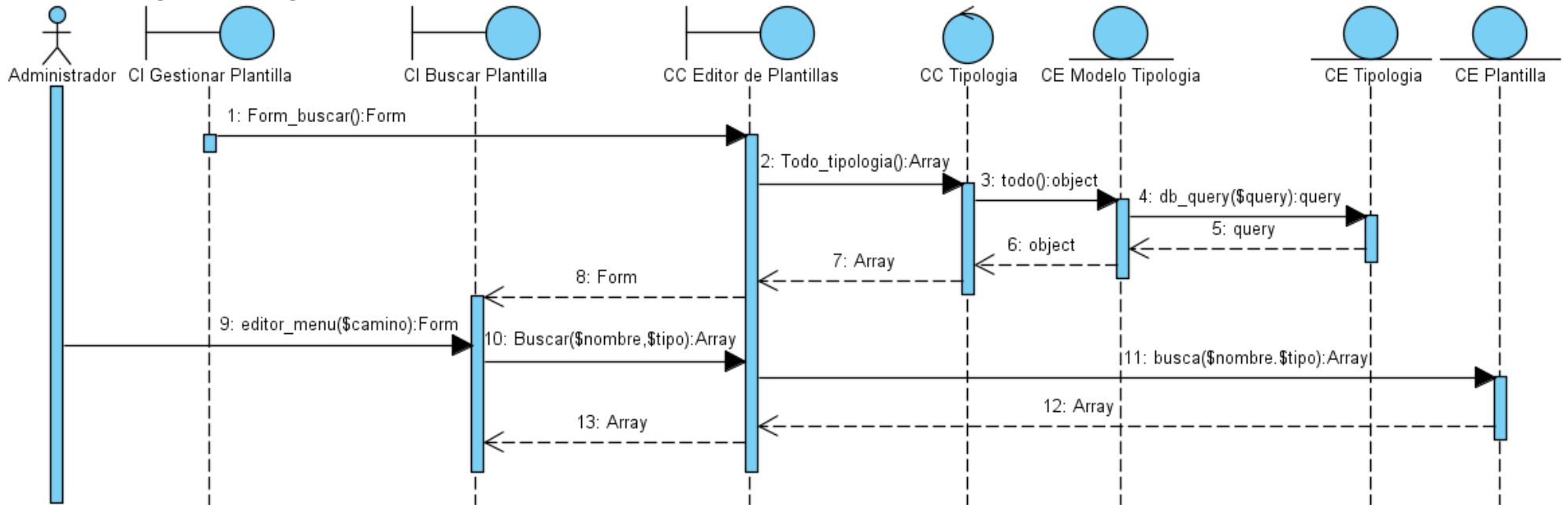


Figura 13: Diagrama de Clases del Análisis. CU Buscar Plantilla.

Anexo 4. Diagrama de Secuencia. CU Buscar Plantilla

Figura 14: Diagrama de Secuencia. CU Buscar Plantilla



Anexo 5. Diagrama de Clases del Diseño. CU Buscar Plantilla

Figura 15: Diagrama de Clases del Diseño. CU Buscar Plantilla

