



**Universidad de las Ciencias Informáticas
Facultad 9.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Portal del Departamento de Señales Digitales de la
Facultad 9**

Autor:

Zuleira Figueredo Rodríguez

Tutor: José Luis Leyva Ramírez

Ciudad de la Habana, julio 2010

Declaración de Autoría

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Zuleira Figueredo Rodríguez.
Firma del Autor

José Luis Leyva Ramírez.
Firma del Tutor

Las Tecnologías de la Información y las Comunicaciones (TIC) forman parte de la cultura tecnológica que nos rodea debido a que posibilita el desarrollo social, además de que ha impuesto un nuevo orden mundial al proceso de gestión de la información. Por estas razones existen empresas actualmente que deciden escoger las alternativas automatizadas para obtener un buen desarrollo de los objetivos que se trazan, logrando un buen negocio dentro del mercado informático.

La investigación que se presenta consiste en la construcción de una aplicación Web, la cual pretende contribuir tanto con la divulgación de las actividades e informaciones del Departamento de Señales Digitales de la Facultad 9, como con la preparación continua de los trabajadores, teniendo una revista digital que mantenga a los mismos informados de las novedades respecto al video y el sonido digital. Para lograr este objetivo se realizó un estudio minucioso de las herramientas y tecnologías más utilizadas, así como las metodologías de desarrollo del software que guían los procesos de desarrollo. Se expone la propuesta y la construcción de la solución a través de los diferentes diagramas de análisis y diseño para la aplicación Web propuesta, dándole cumplimiento a los requisitos descritos por el cliente. Luego de culminar dicho producto se procedió a la prueba del software para demostrar que el mismo posee la calidad requerida para comenzar su uso de inmediato.

PALABRAS CLAVES

Aplicación Web, Departamento de Señales Digitales, Video y Sonido Digital, Análisis, Diseño, Prueba.

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción	5
1.2 Conceptos Asociados al dominio del problema	5
1.3 Portales.....	12
1.3.1 Características Generales	12
1.4 Conclusiones parciales	13
CAPÍTULO 2: Tecnologías y tendencias actuales a considerar	14
2.1 Introducción	14
2.2 Tendencias al desarrollo en software libre	14
2.3 Metodologías de desarrollo de software	15
2.3.1 Proceso Unificado de Desarrollo	16
2.3.2 Programación eXtrema.....	17
2.3.3 Comparación entre las metodologías RUP y XP	18
2.3.4 Metodología Escogida	20
2.4 CMS Drupal como Sistema de Gestión de Contenido	20
2.5 Gestor de Base de Datos	22
2.6 Lenguajes de Programación.....	23
2.6.1 PHP.....	24
2.7 IDEs	25
2.7.1 Eclipse.....	26
2.8 Herramientas Case de desarrollo de software	27
2.8.1 Visual Paradigm como herramienta CASE	27
2.8.2 Rational Rose.....	28
2.8.3 Herramienta de modelado escogida	29
2.9 Arquitectura de Software	29
2.9.1 Modelo-Vista-Controlador.....	31
2.10 Arquitectura de Información	33

2.11 Conclusiones parciales	36
CAPÍTULO 3: Presentación de la solución propuesta	37
3.1 Introducción	37
3.2 Modelo de dominio	37
3.2.1 Eventos principales del entorno.....	37
3.2.2 Conceptos principales del modelo de dominio.....	38
3.2.3 Diagrama de clases del modelo de dominio	40
3.3 Requerimientos del sistema.	40
3.4 Descripción de los actores del sistema.....	43
3.4.1 Casos de Uso del sistema	44
3.4.2 Diagrama de CU del sistema.....	45
3.4.3 Descripción del CU fundamental del sistema.....	46
3.4.4 Patrones de CU.....	46
3.4.5 Patrones de CU utilizados	47
3.5 Conclusiones parciales	47
CAPÍTULO 4: Construcción de la Solución Propuesta	48
4.1 Introducción	48
4.2 Modelo de análisis	48
4.2.1 Diagramas de clases del análisis.....	48
4.3 Diagramas de interacción.....	49
4.3.1 Diagrama de colaboración.....	50
4.4 Modelo de diseño.....	50
4.4.1 Diagrama de clases del diseño.....	51
4.4.2 Descripción del diseño	51
4.4.3 Patrones de Diseño	53
4.5 Diseño de la base de datos	54
4.6 Modelo de datos.....	55
4.7 Diagrama de Despliegue	55
4.8 Modelo de implementación.....	56
4.8.1 Diagrama de componente	56

4.9 Estándar de Codificación	57
4.10 Conclusiones parciales	62
CAPITULO 5: Validación de la Solución Propuesta	63
5.1 Introducción	63
5.2 Pruebas.....	63
5.2.1 Partición Equivalente.....	64
5.2.1.1 Secciones a probar en el Caso de Uso	64
5.2.1.2 Descripción de variables	65
5.2.1.3 Matriz de Datos	65
5.3 Conclusiones Parciales	68
Conclusiones	69
Recomendaciones	71
Referencias Bibliográficas.....	¡Error! Marcador no definido.
Bibliografía.....	¡Error! Marcador no definido.

INTRODUCCIÓN

En la actualidad, el mundo está sumergido en una sociedad donde la información y las comunicaciones juegan un papel importante en la vida diaria del ser humano. Han surgido nuevos medios para difundir y compartir la información, por lo que vale aclarar que estos medios son instrumentos que evolucionan constantemente. Estudios realizados por científicos demuestran que la primera forma de comunicación entre los humanos fue la de los signos y señales empleados en la prehistoria. La aparición de la escritura se toma como hito de inicio de la historia. A partir de ese momento gracias a los cambios económicos y sociales, al nacimiento y evolución de la escritura, a los medios audiovisuales unidos a la era de la electricidad, la revolución de la informática y las telecomunicaciones todos estos medios fueron impulsados. Como ejemplo de estos medios se encuentran el libro, el periódico, la historieta, el teléfono, la radio, la televisión, el cine y el más importante en estos momentos o el más usado, Internet.

Existen empresas dedicadas al desarrollo de estos medios logrando un avance a nivel mundial. Cuba no se excluye de tal desarrollo a pesar del riguroso bloqueo que tiene impuesto los Estados Unidos, por lo que el propósito fundamental es insertarse en el mercado de software a nivel mundial ya que esto trae consigo grandes perspectivas económicas. Hay empresas que se dedican a la creación de software para diferentes ramas de la sociedad como la educación, la salud y el comercio.

Para apoyar el desarrollo de la informatización del país se creó la Universidad de las Ciencias Informáticas (en lo adelante UCI), creada bajo el sello de Batalla de Ideas por su posterior objetivo a desempeñar. En la misma aparte del proceso docente educativo existen centros de producción por facultades, para la realización de los diferentes proyectos que más tarde la UCI desarrollará para la informatización del país. En el caso de la Facultad 9 se encuentra el Centro de Producción de Geoinformática y Señales Digitales.

El Departamento de Señales Digitales (en lo adelante DSD) fue estructurado en el 2010, trazándose como visión en primer lugar obtener mediante las líneas de investigación y las líneas de producción conocimientos sobre el Video y Sonido Digital. Otra visión trazada por dicho departamento es trabajar en el desarrollo de nuevos productos y servicios informáticos en diferentes ramas como son la televisión,

gestión de contenido audiovisual y procesamiento digital de video y sonido. El DSD se encuentra estructurado por diferentes proyectos que realizan tareas de acuerdo a las áreas temáticas correspondientes.

El DSD en el tiempo que lleva estructurado ha logrado avanzar, pero sin dejar de señalar que todavía presenta diversos problemas en cuanto a la comunicación interna del mismo, es decir, difusión de las actividades como Talleres Productivos, tanto para profesores como para el colectivo de trabajo en general (profesores y estudiantes), Ferias Productivas, Conferencias sobre algún tema específico, Cursos Optativos, Concursos, además de actividades recreativas y divulgación de informaciones importantes como reuniones, noticias y horario de trabajo.

Otro problema existente es la necesidad de un repositorio de herramientas, en el cual se guardarán todas aquellas herramientas que son utilizadas para el desarrollo de los productos. También se necesita una revista donde se plasme lo último acerca de video y sonido digital, obteniendo así que los integrantes del departamento tengan conocimientos frescos y novedosos sobre este tema.

La divulgación de todas las actividades que se realizan o planifican en el DSD es a través del correo electrónico. Para saber de las noticias actuales de temas del Departamento se recurre a Internet y estas se les hacen llegar de forma diferenciada a los individuos de los proyectos o de manera informal.

Debido a lo anteriormente planteado se identifica el siguiente **problema científico**: Necesidad de agilizar los procesos de gestión de la información del Departamento de Señales Digitales de la Facultad 9.

Como **objetivo general** se propone desarrollar un portal Web con herramientas libres que permita hacer eficientemente la gestión de la información del Departamento de Señales Digitales de la Facultad 9.

El **objeto de estudio** está centrado en el proceso de gestión de la información del Departamento de Señales Digitales de la Facultad 9 cuyo **campo de acción** se enmarca en la informatización del proceso de gestión de la información del Departamento de Señales Digitales de la Facultad 9.

Como **idea a defender**: Contar con un portal Web para la gestión de la información del Departamento de Señales Digitales, facilitará que los miembros del mismo tengan conocimiento sobre la planificación y publicación de las actividades.

Para asegurar el cumplimiento de los objetivos trazados se proponen las siguientes **Tareas de la investigación**:

1. Establecer la situación actual de las herramientas de gestión de contenido.
2. Establecer las tendencias actuales en la producción de portales Web.
3. Obtener los requisitos funcionales del Portal del DSD.
4. Diseñar la arquitectura de información.
5. Seleccionar las herramientas de implementación.
6. Realizar el análisis y diseño del Portal.
7. Implementar el Portal del DSD.

Como **posible resultado** se espera obtener una aplicación desarrollada en software libre que cumpla con todas las funcionalidades y requisitos descritos por el jefe del DSD que es el cliente.

Para darle cumplimiento a las tareas propuestas con anterioridad, de los métodos científicos existentes se utilizaron los métodos teóricos:

Métodos Teóricos

Analítico-sintético: Estudiando y analizando todo lo relacionado con el desarrollo de los portales Web, se pudo hacer una profundización de todas sus características, dígase objetivos, logros, o deficiencias que se derivan, así como también las mejoras que se pueden proponer para obtener resultados eficientes.

Histórico-Lógico: Se utilizó en la investigación para conocer las principales características de los portales tanto nacional como internacionalmente y así lograr un desarrollo eficiente de la propuesta del DSD.

Modelación: Una vez realizada la investigación y analizada la información, se procedió a la elaboración de los modelos correspondientes al ciclo de vida del sistema, lo que permitió facilidades a la hora de cumplir con las tareas de análisis y diseño de los procesos que intervinieron en la aplicación, así como para la implementación del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se comentarán los principales conceptos asociados al marco del problema que permitan un mejor entendimiento de la investigación. Estos conceptos serán descritos mediante sus características y ventajas, aspectos que son muy importantes a la hora de comenzar a desarrollar un proyecto. También se argumentará el objeto de estudio a través de sus características para tener más conocimientos sobre el tema a tratar.

1.2 Conceptos Asociados al dominio del problema

Aplicaciones Web

Para cumplir con el objetivo de este trabajo se debe desarrollar un sistema que brinde la posibilidad a múltiples usuarios de tener acceso al mismo tiempo a la aplicación, dándoles la oportunidad de obtener rápidamente la información. En este caso el uso de una aplicación Web es la manera más fácil y sencilla de acceder a la información que se registrará en dicha aplicación.

Una aplicación Web es un sistema informático que los usuarios pueden emplear accediendo a un servidor Web a través de Internet o de Intranet. Dentro de las mismas se encuentran webmails, wikis, weblogs, tiendas en líneas y Wikipedia.

Uno de los factores que se tiene en cuenta para la utilización de las aplicaciones Web son las ventajas que presentan, a continuación se muestran algunas:

- Funcionan en cualquier sistema operativo. (1)
- Brindan la posibilidad de acceso a varios usuarios simultáneamente al sistema.
- Proporcionan alta disponibilidad debido a que se puede realizar consultas en cualquier parte del mundo y a cualquier hora siempre y cuando se tenga acceso. (2)

- Permiten tener un avanzado sistema de consultas altas, bajas y modificaciones de datos procedentes de cualquier área de la institución, lo que proporciona que la información esté siempre actualizada. (2)
- Facilita el trabajo a distancia. (2)
- Desarrollo sencillo, económico y rápido. (1)

En conclusión, la utilización de las aplicaciones Web implica reducción de costos, siendo así una magnífica opción para que las instituciones automaticen sus procesos sin invertir mucho dinero en equipos, desarrollo y capacitación.

Las aplicaciones Web se reúnen en tres grupos: Informativas, De negocio y Portales.

Las aplicaciones Web informativas: En las aplicaciones Web informativas la página principal es la más importante debido a que es la que muestra de forma panorámica el contenido de la misma. En ellas se encuentra información de temas específicos, casi siempre estas aplicaciones no se actualizan de tal manera que los contenidos perduren a lo largo de su vida.

Las aplicaciones Web de Negocio: Estas aplicaciones Web las empresas tradicionales las utilizan para un mejor servicio a los usuarios, son procesos de negocios transfigurados a aplicaciones fundamentadas en Web o en línea para aportar una mejoría empresarial, de sistemas y resultados.

Los portales: Son el grupo más complejo de las aplicaciones Web. Son conocidos como “Puerta Grande”, es decir, es un sitio Web que será el punto de partida para la navegación en la red. Facilita el acceso a una serie de recursos y servicios muy comunes en estas aplicaciones, como son los motores de búsquedas, buzones de correo electrónico y foros de discusión.

Una de las tecnologías que se utiliza actualmente para realizar una aplicación Web con calidad y de la manera más eficiente y sencilla son los Sistemas de Gestión de Contenidos.

Sistema de Gestión de Contenido

Para la realización del portal Web del DSD de la Facultad 9 se decidió que utilizar un Sistema de Gestión de Contenido (en lo adelante CMS) era una muy buena opción, debido a que son herramientas que

brindan la posibilidad de crear y mantener un sitio Web de manera sencilla, asumiendo el trabajo más pesado.

Un CMS es una aplicación informática usada para crear, editar, gestionar y publicar contenido digital en diversos formatos, generando páginas dinámicas, interactuando con el servidor para desarrollar la página Web como el usuario la desee. (3)

Uno de los elementos fundamentales por lo que se rigió el DSD para utilizar un CMS en la elaboración del portal Web del departamento fue lo que a continuación se detalla.

Ventajas de usar un CMS:

- Generan, gestionan y publican la información con calidad, debido a que son herramientas que brindan la posibilidad al responsable de contenido del sitio Web o portal Web de publicar, editar, borrar, actualizar y administrar la información sin conocimientos de HTML.
- El propósito de un CMS es la creación de aplicaciones y sitios Web dinámicos de forma rápida y eficiente, integrando contenidos de numerosas fuentes y formatos como son textos, imágenes, audio y video.
- Sencilla administración de la Web o portal ya que la curva de aprendizaje es muy baja. La información se introduce rellenoando unos formularios a través de un navegador Web. La administración completa de la Web o portal se realiza desde una única aplicación Web.
- Solución adaptable mediante módulos: Los CMS presentan módulos independientes y componentes que pueden combinarse para conseguir la solución específica para unas necesidades concretas.

Otro elemento importante que no se puede dejar de mencionar se argumenta a continuación.

Necesidad de utilizar un CMS

- **Mantenimiento de gran cantidad de páginas:** En una Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a

las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos. (4)

- **Reutilización de objetos o componentes:** Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado. (4)
- **Páginas interactivas:** Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. (4)
- **Cambios del aspecto de la Web:** Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS¹ con lo que se consigue la independencia de presentación y contenido. (4)
- **Consistencia de la Web:** La consistencia en una Web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de la misma Web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas. (4)

Tipos de CMS

Existen diferentes variedades de CMS. Es posible agruparlos según el tipo de sitio que permiten gestionar. A continuación se muestran los fundamentales:

- **Genéricos:** Ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenidos, para soluciones de comercio electrónico, blogs y portales.
- **Específicos:** Para ONG². Nacen para cubrir las necesidades de las ONG, ofreciendo una plataforma de servicios de Internet que en ocasiones incluye además del CMS herramientas para la recaudación de fondos.

¹ Hojas de estilo en cascada

² Organizaciones No Gubernamentales

- **Foros:** Sitio que permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados. Ejemplos: phpBB, MyBB.
- **Blogs:** Publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión. Ejemplos: Wordpress, Typo.
- **Wikis:** Sitio Web dónde todos los usuarios pueden colaborar en los artículos, aportando información o reescribiéndola. También permite espacio para discusiones. Indicado para material que irá evolucionando con el tiempo. Ejemplos: Mediawiki, Tikiwiki.
- **e-Commerce:** Son sitios Web para comercio electrónico.
- **e-Learning:** Sirve para la enseñanza. Los usuarios son los profesores y estudiantes, se tienen aulas virtuales donde se ponen a disposición el material del curso. La publicación de un contenido por un profesor es la puesta de dicho contenido a disposición de los estudiantes en un aula virtual. Ejemplo: Moodle.
- **Publicaciones digitales:** Son plataformas diseñadas teniendo en cuenta las necesidades de las publicaciones digitales, tales como periódicos, revistas. Ejemplo: ePrints.
- **Portal:** Sitio Web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad. Ejemplos: PHPNuke, Postnuke, Drupal, Plone.

El software que se emplea para la utilización y/o actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios se le denomina Sistema Gestor de Base de Datos, el cual es necesario una vez decidido la utilización de un CMS.

Sistema Gestor de Base de Datos

Un sistema gestor de base de datos (en lo adelante SGBD) es un conjunto de programas que permiten crear y mantener una base de datos (en lo adelante BD) asegurando su confidencialidad, integridad y seguridad (5). El objetivo principal de un SGBD es servir de interfaz entre la BD, el usuario y la aplicación, de manera que se pueda almacenar y acceder rápida y estructuradamente a los datos, estos sistemas tienen diferentes características.

Características de un SGBD

- **Independencia:** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una BD sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima:** Un buen diseño de una BD logrará evitar la aparición de información repetida o redundante. (6)
- **Seguridad:** La información almacenada en una BD puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información. (6)
- **Integridad:** Se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. (6)
- **Respaldo y recuperación:** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder. (6)

Un SGBD debe incluir por lo menos las siguientes funciones:

- **Definición de los Datos:** El SGBD debe ser capaz de aceptar las definiciones de datos en versión fuente y convertirlas en la versión objeto. (7)
- **Manipulación de los Datos:** El SGBD debe atender las solicitudes de los usuarios para extraer, actualizar, adicionar o suprimir datos. (7)
- **Seguridad e Integridad de los Datos:** El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el Administrador de la Base de Datos DBA. (7)

El uso de un SGBD es muy conveniente debido a que proporciona numerosas ventajas en cuanto a la integración de los datos y a la interface común que brinda, otorgando seguridad y calidad al producto final. Ventajas que no se deben de pasar por alto a la hora de la elección de un buen SGBD, a continuación se mencionan las mismas.

Integración de los datos:

- **Control sobre la redundancia de datos:** Los sistemas de ficheros realizan varias copias de los datos en diferentes ficheros guardándolos, lo que trae consigo que se utilice más memoria de la que se necesita y que los datos pierdan consistencia.
- **Consistencia de datos:** Si existe un dato único almacenado, cada vez que se haga una actualización se hará una sola vez y estará disponible para los usuarios. Si este está duplicado y el sistema reconoce la redundancia se encargará de realizar las actualizaciones a todas las copias.
- **Compartición de datos:** En los sistemas de ficheros, estos pertenecen a las personas o a los departamentos del proyecto que los utilizan. En los sistemas de BD esta pertenece al proyecto y puede ser compartida por todos los usuarios que estén autorizados.

Interface común:

- **Mejora en la seguridad:** La seguridad de la BD se basa en la protección contra aquellos usuarios que no tienen autorización al acceso de la misma. Sin unas buenas medidas de seguridad, la integración de datos en las bases de datos hace que estos sean más vulnerables que en los sistemas de ficheros.
- **Mejora en la accesibilidad a los datos:** Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario realizar cualquier tipo de consulta sobre los datos, sin la necesidad de que un programador cree una aplicación para el cumplimiento de esta tarea.
- **Aumento de la concurrencia:** En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la BD y garantizan que no ocurran problemas de este tipo.

A partir de los conceptos asociados al dominio de la investigación mencionados anteriormente, es recomendable argumentar acerca del objeto de estudio de la misma, es decir, comentar sobre la actividad fundamental, los portales Web.

1.3 Portales

1.3.1 Características Generales

Un portal es el conjunto de páginas Web o sitios Web que sirve como punto de entrada único a la información de una institución o empresa y tiene como fuente de ingreso la publicidad de dichas empresas. Es un centro de contenido central donde se pone todo tipo de información a disposición de un público muy diverso.

Los portal Web deben de tener organizada la información del sitio en secciones y categorías para facilitar la navegación a los usuarios, permitiendo así una estructura sólida, ordenada y sencilla para los administradores. Dan la posibilidad de instalar, desinstalar y administrar componentes y módulos que agregarán servicios de valor a los visitantes del sitio Web, por ejemplo: galería de imágenes, foros, clasificados y envío de boletines. Los portales Web brindan la posibilidad de almacenar datos de usuarios registrados y también permiten enviar e-mails masivos a todos los usuarios.

A la hora de desarrollar un portal Web la empresa, centro o institución debe de tener claro dentro de las clasificaciones de los mismos cuál es la que tiene más similitud con el tema propuesto.

Tipos de portales

Existen diferentes tipos de portales Web en función de sus usuarios, éstos son:

- **Portal Horizontal:** Llamados también portales masivos o de propósito general, están enfocados a una audiencia amplia, estipulando llegar al usuario a través de muchas cosas, es decir, su objetivo son los usuarios. La mayoría ofrecen motores de búsquedas, correo electrónico, y otras prestaciones de comunicación.
- **Portal Vertical:** Son portales que se especializan en determinados temas con el fin de buscar público, es decir, atraer al usuario a través de sus intereses, como portales de música, empleo, inmobiliario, arte, deporte, de finanzas personales. Se pueden a su vez clasificar en función de sus objetivos:
 - **Portal Intranet:** Comunicación corporativa para los empleados.
 - **Portal Extranet:** Comunicación corporativa para los proveedores.

- **Portal Vertical:** Comunicación corporativa con clientes.
- **Portales Diagonales:** Se trata de la mezcla entre los dos portales anteriormente mencionados. Estos utilizan redes sociales o aplicaciones generalistas como Facebook, Linkd, Flickr o YouTube, completados con contenidos y/o utilidades dirigidas a un público muy concreto.

La diferencia entre los portales horizontales y verticales está en que los horizontales brindan sus servicios a todos los usuarios sin importar cuál sea el contenido del portal Web, entretanto, los verticales orientan sus servicios para alguna empresa o institución particular a través de la implementación de los diferentes tipos de portales verticales basándose en temas específicos. El tipo de portal Web a implementar para el DSD será un portal vertical debido a que se analizará, diseñará e implementará el mismo para un tema específico, video y sonido digital.

1.4 Conclusiones parciales

En este capítulo se realizó una descripción general del objeto de estudio. Se presentaron los conceptos asociados al tema para lograr un mayor entendimiento a través del camino a transitar. De ahí que surja la idea de desarrollar un sistema como el que se propone, con el objetivo de solucionar la situación problemática que genera el problema científico, razón de esta investigación.

CAPÍTULO 2: Tecnologías y tendencias actuales a considerar

2.1 Introducción

En el capítulo anterior se hizo referencia a los sistemas a utilizar para el desarrollo del producto deseado. En este capítulo se abordará de manera más específica en estos sistemas, cuáles son los que se utilizarán, describiéndolos. También se argumentará brevemente algunas de las principales tendencias actuales para el crecimiento de las aplicaciones Web en la UCI, metodología de desarrollo de software, herramienta CASE y lenguaje de programación.

2.2 Tendencias al desarrollo en software libre

Las aplicaciones Web hoy en día están entre las soluciones más codiciadas y usadas para la automatización de la gestión de la información. La mayoría de las instituciones y empresas aumentan su prestigio, economía y divulgación a través de éstas. En la UCI unas de las principales tendencias a la hora de desarrollar aplicaciones Web son: la utilización de Software Libre y CMS.

El software libre propone una serie de ventajas muy factibles para el trabajo. Es un software económico, con libertad de uso y de redistribución, además de ser un sistema de expansión. La libertad de uso y redistribución conlleva a que se puede realizar todas las modificaciones necesarias para adaptarlo a características específicas que pueda tener el producto a desarrollar. Una ventaja económica que se tiene muy en cuenta es que es un sistema en expansión que ha permitido un crecimiento constante del software libre a través de las aportaciones a las soluciones libres y de las aportaciones de las comunidades.

En cuanto a los CMS, se puede decir que las principales razones de su popularidad se deben a que es una forma sencilla de desarrollar aplicaciones Web, que administren grandes cantidades de información que necesitan constantes actualizaciones o para las que se pronostica un crecimiento futuro. También proporcionan alta interactividad a los visitantes y permite trabajar con el diseño y el contenido de manera independiente.

Cuando se comienza a desarrollar un proyecto de software es necesario decidir por cuál de las metodologías de desarrollo regirse y que sea adaptable a las necesidades reales.

2.3 Metodologías de desarrollo de software

Con el avance de las tecnologías se ha hecho necesario el desarrollo de métodos para poder guiar, controlar y documentar los proyectos de software que van surgiendo. De esta manera surgen las metodologías de desarrollo, con el objetivo de lograr que el desarrollo de software sea un proceso disciplinado, predecible y eficiente. Las metodologías se clasifican en:

- **Tradicional¹**: Estas imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que esté todo detallado, comienza el ciclo de desarrollo del producto.
- **Ágiles²**: Se caracterizan por tener un desarrollo incremental para producir tempranamente pequeñas entregas en ciclos rápidos, y predisposición para el cambio y la adaptación continúa. Estas metodologías por lo general se centran en desarrollar productos funcionales más que en conseguir una buena documentación.

Existen diferentes metodologías de desarrollo como son:

- **Tradicionales:**
 - RUP – Proceso Unificado de Desarrollo.
 - Métrica_3.
- **Ágiles:**
 - XP – eXtreme Programming.
 - Scrum.
 - FDD – Feature Driven Development.

¹ Metodologías Pesadas o Peso Pesado

² Metodologías Ligeras

Comparación de Ágiles - Tradicionales

Metodología Ágil	Metodología Tradicional
Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura	La arquitectura es esencial

Tabla 1: Comparación entre Metodologías Ágiles y Tradicionales

Dentro de los procesos de desarrollo de software más utilizados se encuentran el Proceso Unificado de Desarrollo (RUP) y Programación eXtrema (XP).

2.3.1 Proceso Unificado de Desarrollo

El Proceso Unificado de Desarrollo es una metodología pesada, está basada en una notación gráfica, la cual permite especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (8). La misma puede adaptarse a cualquier proyecto y está dividido en 4 fases de desarrollo como son inicio, elaboración, construcción y transición. La fase de inicio determina la visión del proyecto, la de elaboración es la que determina la arquitectura, es decir realiza la definición, análisis y diseño, la de construcción se encarga de la implementación y la de transición es cuando se llega al fin del proyecto y este se pone en marcha. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones y en cada ciclo de iteración se hace exigente el uso de artefactos. Los elementos que componen a RUP son las actividades que se determinan en cada iteración, los trabajadores, que son las personas involucradas en cada proceso y los artefactos que pueden ser un documento, un modelo o un elemento de modelo. En cuanto a sus características describe como diseñar una arquitectura flexible, que se acomode a los cambios, comprensible intuitivamente y promueve una más efectiva reutilización de

software. Está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. (8) RUP provee un enfoque sistemático para definir una arquitectura utilizando componentes nuevos y preexistentes. Muestra como modelar software visualmente para capturar la estructura y comportamiento de arquitecturas y componentes. Las abstracciones visuales ayudan a comunicar diferentes aspectos del software; comprender los requerimientos, ver como los elementos del sistema se relacionan entre sí, mantener la consistencia entre diseño e implementación y promover una comunicación precisa. El aseguramiento de la calidad se construye dentro del proceso, en todas las actividades, involucrando a todos los participantes, utilizando medidas y criterios objetivos, permitiendo así detectar e identificar los defectos en forma temprana. Es también una guía para establecer espacios de trabajo seguros para cada desarrollador, suministrando el aislamiento de los cambios hechos en otros espacios de trabajo y controlando los cambios de todos los elementos de software. Contiene abundante documentación y con él se trabaja muy organizado. Sus principales características son:

- Guiado por casos de uso: Los casos de uso son el instrumento para describir el comportamiento del software y extraer los casos de prueba con los que se valida el sistema. (8)
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño, describe la arquitectura del producto a desarrollar. (8)
- Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones superiores. (8)

RUP se basa en casos de uso para describir lo que se espera del software. Basándose en el Lenguaje Unificado de Modelado (UML). (9)

2.3.2 Programación eXtrema

XP es una metodología ágil que es utilizada en proyectos de corto plazo. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo y una buena comunicación con los usuarios, preocupándose por el aprendizaje de los

desarrolladores, y propiciando un buen clima de trabajo. Se basa en una serie de valores que se aplican una y otra vez en las distintas prácticas utilizadas en la programación extrema: realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Las etapas de esta metodología son planificación del proyecto, diseño, codificación y pruebas. Formalmente, XP comprende 12 prácticas centrales. En el juego del planeamiento, el cliente escribe escenarios simples en tarjetas sin un documento de especificación complejo y extenso, esta técnica de baja tecnología conduce a la comunicación cara a cara con el equipo de programación. El cliente permanece en el sitio de desarrollo e interactúa con los desarrolladores, así que los cambios en el diseño pueden suceder rápidamente.

2.3.3 Comparación entre las metodologías RUP y XP

Las metodologías ágiles o ligeras reconocen las características inherentes de complejidad de software y tratan de mejorar su calidad por medio de una estrecha comunicación con las personas que intervienen en el proceso de desarrollo, mientras que las pesadas consiguen su objetivo final estableciendo un orden lógico y auxiliándose de abundante documentación. Por lo tanto se debe establecer una comparación entre ambos tipos de metodologías para decidir cuál usar en dependencia de las características y necesidades del proyecto a desarrollar.

RUP y XP no tienen muchas similitudes entre sí. RUP es uno de los procesos más generales en la actualidad, ya que se puede adaptar a cualquier proyecto y por medio de la preparación de tareas pendientes en función de los objetivos de la fase y actividad actual intenta reducir la complejidad del software.

Aspectos a comparar:

- **Obtención de requisitos:** RUP y XP describen los requerimientos de la aplicación desde el punto de vista del usuario y los definen sin adentrarse en detalles de implementación.
- **Carga de trabajo:** En el desarrollo de un proyecto para XP es más importante entregarle al cliente el software que necesita sin requerir tanta documentación externa y modelado, por lo que se reduce la carga de trabajo al no tener que elaborar la documentación de todas las funcionalidades que se van a implementar. Si se va a realizar algún cambio se acuerda directamente con el representante del cliente y no se tiene que mostrar por escrito. RUP por el contrario si requiere de

abundante documentación y todo cambio que se haga tiene que llevarse por escrito, define detalladamente cada fase del ciclo de vida del proyecto con sus artefactos y roles involucrados.

- **Relación con el cliente:** En RUP al finalizar cada fase se le presenta al cliente los artefactos obtenidos y no se procede a la próxima fase hasta que el cliente no los aprueba. La calidad de los mismos será probada durante la totalidad del ciclo de vida del proyecto y se presentará por escrito. Por el contrario XP asegura la calidad mediante controles y estableciendo una comunicación lo más fluida posible con el cliente, sin basarse tanto en la documentación. Al finalizar cada iteración el cliente recibe una parte funcional del programa, por lo que se mantiene informado constantemente sobre la situación del proyecto y puede intervenir inmediatamente si el desarrollo se aleja de sus necesidades.
- **Desarrollo:** XP está diseñado para facilitar el trabajo de los programadores. Es por ello que define todo el proceso de desarrollo al máximo, incluyendo la integración de código de forma permanente para que la detección de errores sea inmediata y la aplicación de pruebas automáticas para garantizar que el código este correcto. Mientras que RUP se centra más en la organización global y muchas de las actividades que realiza las asume como obligatorias sin haberlas definido completamente, como es el caso de la ejecución de pruebas. RUP genera versiones después de cada fase acompañado de notas de la versión, instrucciones de instalación, ayuda de uso, entre otras.
- **Código fuente:** XP presenta compartición de código al aplicar la programación en pareja, la cual evita que se pasen por alto errores comunes en el código. Al estar dos personas frente al ordenador el código sale más limpio y si se va alternando la pareja el conocimiento sobre cada parte del sistema se va extendiendo por todo el equipo. Cualquier miembro del equipo puede hacer cambios en el código, aportando mejor calidad a este. No siendo así en RUP, ya que opta por la propiedad del código, aunque define grupos y secciones de trabajo.
- **Conocimiento sobre la arquitectura:** En XP se conseguirá mediante la programación en pareja, pues en la creación del código se pueden evitar errores y malos diseños, controlando cada línea de código y decisión de diseño. Mientras que en RUP se intentará disminuir la complejidad del software a desarrollar mediante una planificación intensiva, para no perder el conocimiento de la aplicación.

2.3.4 Metodología Escogida

RUP es ajustable al tamaño del proceso en desarrollo, según las necesidades que tenga el proyecto en cuestión, se hace muy necesaria la documentación de todo el proceso en cada una de sus fases y flujo de trabajo, para que se le pueda dar seguimiento en un futuro para realizar actualizaciones de la aplicación planteada.

Como bien se mencionó en el capítulo anterior, para el desarrollo de Portal del Departamento de Señales Digitales se escogió un CMS, a continuación se describirá el CMS Drupal que se utilizará por todas las características que presenta.

2.4 CMS Drupal como Sistema de Gestión de Contenido

Drupal es un sistema de administración de contenidos¹ que permite tanto a un usuario, como a una comunidad de usuarios publicar, administrar, y organizar fácilmente una gran variedad de contenidos en un sitio Web. (10)

Drupal es el equivalente fonético en inglés a la palabra “druppel” que significa gota y fue originalmente escrito por DriesBuytaert. Es uno de los CMS más ampliamente usados, aproximadamente 10 mil personas y organizaciones usan Drupal en diferentes tipos de sitios Web, incluyendo portal Web de comunidades Web, sitios de discusión, sitios Web corporativos, redes sociales, blogs, sitios personales, aplicaciones de comercio electrónico. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

Es un CMS de código abierto, escrito en PHP y distribuido bajo la licencia GPL² y es desarrollado y mantenido por una comunidad de miles de usuarios y desarrolladores. (10)

Presenta características muy potentes y convincentes que hace recomendable su uso para el desarrollo de la aplicación en cuestión.

¹ CMS por sus siglas en inglés

² Licencia Pública General de GNU

Características Generales

- **Búsqueda:** Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento. (11)
- **Código abierto:** El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario de otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades. (11)
- **Módulos:** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc. (11)
- **Personalización:** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario. (11)
- **Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multiplataforma. 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.
- **Independencia de la base de datos:** Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una "capa de abstracción de base de datos" que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Gestión de usuarios

- **Autenticación de usuarios:** Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP. (11)
- **Permisos basados en roles:** Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles. (11)

Existen otras razones que hacen a Drupal el CMS elegido para el desarrollo de este portal Web como que es una buena base que permite integrar todo tipo de módulos, genera un código HTML extremadamente limpio y bien estructurado, el sistema base de gestión de contenido es limpio, flexible y potente, por lo que se puede resumir que es uno de los mejores en cuanto a diseño, flexibilidad y extensibilidad.

Una de las características fundamentales del CMS Drupal que se toma en cuenta para su utilización es la independencia de la base de datos (BD). Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, este CMS incorpora una capa de abstracción de BD que está implementada y mantenida para MySQL y PostgreSQL. Para almacenar toda la información que se gestionará en el portal Web es necesario utilizar uno de los dos SGBD mencioandos. A continuación se describe el SGBD escogido.

2.5 Gestor de Base de Datos

PostgreSQL, originalmente se llamó Postgres, fue creado en la Universidad de California en Berkeley, por un profesor de Ciencia de la Computación llamado Michael Stonebraker. PostgreSQL es un potente gestor de base de datos relacional libre (liberado bajo licencia BSD¹) (12). Es un SGBD multiplataforma y su desarrollo se lleva a cabo por un equipo de desarrolladores dispersos alrededor del mundo. Este es un proyecto de la comunidad PostgreSQL Global Development Group (PGDG) y que no está controlado por ninguna compañía.

Las características técnicas que tiene hacen que sea una de las bases de datos más potente y robusta del mercado. Lleva más de 15 años de desarrollo continuo y en todo ese tiempo estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la misma vez. (13) Además de permitir el control de clases ajenas, subconsultas y gestión de transacciones, características que no están disponibles para Access (Windows), ni en MySQL (Linux).

¹ Berkeley Software Distribution

Las ventajas de este gestor de base de datos son:

- **Gran escalabilidad:** Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, lo que posibilita atender un mayor número de peticiones concurrentes.
- **Extensible:** El código fuente está disponible para todos sin costo alguno, puede ser personalizado con un mínimo de esfuerzo y sin costos adicionales. (14)
- **Diseñado para ambientes de alto volumen:** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC¹ para conseguir una mejor respuesta en ambientes de grandes volúmenes. (14)
- **Soporte:** Existe una comunidad de profesionales que contribuyen con su desarrollo.
- **Multiplataforma:** Está disponible en casi cualquier UNIX (34 plataformas en la última versión estable) y con versión nativa para Windows. (14)

Por todo lo planteado se puede decir que este SGBD bajo la licencia de BSD tiene la libertad de usarse, modificarse y distribuirse sin costo alguno.

Para que las computadoras entiendan nuestras instrucciones debe usarse un lenguaje específico. A continuación se argumentará acerca del tema. Además se hace énfasis en el lenguaje PHP debido a que el CMS Drupal está escrito en dicho lenguaje de programación y para no crear incompatibilidades fue necesaria su utilización.

2.6 Lenguajes de Programación

Las máquinas en general, y las computadoras en particular, necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para poder controlar su comportamiento. Ese lenguaje que permite esta relación con las computadoras es el lenguaje de programación, algunos de ellos son Visual Basic, Java, ASP, Cobol, C++. (15)

Los lenguajes de programación pueden ser interpretados (que usan un intérprete) o compilados (que usan un compilador). Los lenguajes de programación interpretados utilizan un programa intérprete o traductor,

¹ Control de Concurrencia Multiversión

el cual analiza directamente la descripción simbólica del programa fuente y realiza las instrucciones dadas. La ventaja del proceso intérprete es que no necesita de dos fases para ejecutar el programa, sin embargo su inconveniente es que la velocidad de ejecución es más lenta ya que debe analizar e interpretar las instrucciones contenidas en el programa fuente.

2.6.1 PHP

PHP es un acrónimo recursivo¹ de PHP: HyperTextPreprocessor, aunque sus orígenes se remontan al nombre **P**ersonal **H**ome **P**age cuando fue creado por RasmusLerdof para conocer cuántas personas estaban leyendo su currículum vitae en su página Web. PHP es un lenguaje de programación interpretado que va embebido (incrustado) en páginas HTML. Su sintaxis es similar a la utilizada en otros lenguajes de programación de alto nivel como C, Java y Perl, sólo con algunas diferencias. El principal objetivo del lenguaje es permitir a los desarrolladores Web codificar rápidamente páginas que se generan dinámicamente. (16)

Es un lenguaje del lado del servidor, lo cual significa que el usuario abre la página HTML y hace una petición de interactuar, PHP, instalado en el servidor Web, interpreta la petición y envía una respuesta al usuario, lo cual marca la diferencia entre lenguajes como Javascript, que es del lado del cliente. Es un lenguaje código abierto, lo que significa que existe una comunidad que se dedica al continuo desarrollo y fortalecimiento del lenguaje, además de que no hay que pagar derechos de autor para utilizarlo. Una de sus principales características es su soporte para gran cantidad de bases de datos. Las ventajas de dicho lenguajes son:

- PHP es un lenguaje multiplataforma, el mismo código puede ser ejecutado en casi 25 plataformas, incluyendo diferentes versiones de UNIX, Windows y Macs.
- Es un lenguaje completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una gran variedad de extensiones de código.

¹ Recursivo porque en el acrónimo se encuentra el acrónimo.

- Presenta interfaces distintas para cada tipo de servidor. Actualmente PHP se puede ejecutar bajo Apache, Internet Information Server (IIS), AOLServer, Roxen. Otra alternativa es configurarlo como módulo Common Gateway Interface (CGI).
- Rapidez. Generalmente se utiliza como módulo de Apache lo que lo hace extremadamente veloz y al estar escrito en C, se ejecuta rápidamente usando poca memoria.

Para empezar con el desarrollo de la aplicación se deben tener bien definidas las herramientas a utilizarse. Se comenzará analizando algunos IDE de desarrollo que son compatibles con el lenguaje de programación anteriormente mencionado y posteriormente se describirá el escogido para el desarrollo del portal Web del DSD.

2.7 IDEs

Un entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. A continuación se describen algunos de ellos.

Zend Studio: Uno de los impulsores de PHP y orientado a desarrollar aplicaciones Web en este lenguaje de programación. Características como la refactorización, la generación de código, asistente de código y análisis semántico se combinan para permitir el desarrollo rápido de aplicaciones. Es uno de los mejores pero hay que pagar por su uso. (17)

PHPEdit: Ofrece un entorno de trabajo para todos aquellos programadores especializados en PHP, pero que también permite trabajar con HTML, XML y TXT; un entorno integrado para Windows y con una variada colección de herramientas. (18)

Php Designer: Es un completo entorno de desarrollo y programación especialmente diseñado para desarrolladores de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS, SQL y hay que pagar para poder usarlo. (19)

2.7.1 Eclipse

La plataforma Eclipse consiste en un IDE código abierto y extensible. Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad. (20)

El término Eclipse además identifica a la comunidad de software libre para el desarrollo de la plataforma Eclipse. Con este trabajo se quiere lograr proporcionar una plataforma robusta, escalable y con calidad para el desarrollo de software con el IDE Eclipse.

Características

Eclipse dispone de un editor de texto con resaltado de sintaxis y compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (*wizards*) para creación de proyectos, clases, tests y refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones Subversion e integración con Hibernate y con la herramienta de modelado Visual Paradigm.

Licencias

Eclipse se distribuye bajo licencia EPL¹. Esta licencia es considerada como libre por la Fundación para el Software Libre y por la OSI². La licencia EPL permite usar, modificar, copiar y distribuir nuevas versiones del producto licenciado. El antecesor de EPL es CPL³. CPL fue escrita por IBM, mientras que EPL es obra del consorcio Eclipse.

Después de explicar de manera general la herramienta de Eclipse se proseguirá a realizar una breve comparación de algunas de las herramientas de modelado y así llegar a una conclusión concreta de la herramienta de modelado que se utilizará para modelar lo que el usuario desea.

¹ Eclipse Public License

² Organización dedicada a la promoción del código abierto

³ Common Public License

2.8 Herramientas Case de desarrollo de software

Las herramientas Case son programas y aplicaciones que pueden ser utilizadas en diversas funciones, además debe considerarse que están dentro de la línea del software y que son fáciles de utilizar.

2.8.1 Visual Paradigm como herramienta CASE

Visual Paradigm es una herramienta CASE¹ que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios. (21)

Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, importación de Rational Rose, exportación e importación XML. Soporta aplicaciones Web, exporta en formato HTML, está disponible en varios idiomas, es fácil de instalar y fácil de actualizar. Los desarrolladores pueden diseñar la documentación del sistema con plantillas de diseño. El analista de sistema puede estimar las consecuencias de los cambios con los diagramas de análisis de impacto, como la matriz y el diagrama de análisis.

Ventajas

- Navegación intuitiva entre el código y el modelo.
- Poderoso generador de documentación y reportes UML PDF/HTML/MS Word.
- Demanda en tiempo real, modelo incremental de viaje redondo y sincronización de código fuente.
- Superior entorno de modelado visual.
- Soporte completo de notaciones UML.
- Diagramas de diseño automático sofisticado.
- Análisis de texto y soporte de tarjeta CRC².

¹ Ingeniería de Software Asistida por Computador

² Clase, Responsabilidad, Colaboración

No se puede dejar de mencionar que esta herramienta es de alta calidad y prestaciones, además de presentar robustez, usabilidad y portabilidad. Actualmente en el DSD se utiliza la versión 6.4 de dicha herramienta. Las características que esta ofrece son las siguientes:

- Soporte de UML 2.0.
- Soporte para Notación BPMN¹ 1.1.
- Modelado colaborativo con CVS² y Subversion.
- Generador de informes para generación de documentación.
- Importación y exportación de ficheros XMI³.
- Soporte de diseño de patrones: Con este repositorio de patrones de diseño, no se necesitará tiempo para reproducir los patrones de diseño una y otra vez.
- Desarrollar progresivamente ERD⁴ conceptuales, lógicos y físicos: Visual Paradigm 6.4 permite estos tres niveles de modelado de datos para facilitar la creación de un modelo de base de datos independiente del sistema gestor de base de datos.
- Editor de Detalles de Casos de Uso: Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.

2.8.2 Rational Rose

Rational Rose es una herramienta CASE desarrollada por los creadores de UML que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. Es el producto más completo de la familia Rational Rose, siendo la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA. Proporciona un lenguaje común de modelado para el equipo de trabajo facilitando la creación del software con calidad y rápidamente.

¹ Notación para el Modelado de Procesos de Negocio

² Sistemas de Versiones Concurrentes

³ Intercambio de Metadatos usando XML

⁴ Diagrama de Entidad Relacional

El navegador de UML de Rational Rose permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el progreso de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. La misma proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

2.8.3 Herramienta de modelado escogida

Después de lo anteriormente mencionado vale aclarar que son herramientas muy valiosas y ambas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, pero aún así se determina que la ideal para el trabajo con la Ingeniería de Software es Visual Paradigm debido a que es una herramienta muy potente y con dos ventajas muy importantes, es una herramienta de software con licencia gratuita y comercial, además de poseer versiones multiplataforma. Es completamente compatible con la metodología que se decidió utilizar brindando muchas facilidades en la generación de la documentación del software que se está desarrollando. Todo esto favorece un buen desarrollo del producto por lo que se obtendrá una mayor calidad en el software.

Como bien se argumentó anteriormente la metodología que se escogió para el desarrollo del portal Web fue RUP y una de las características fundamentales que tiene es que es centrado en la arquitectura. Tener definida una arquitectura para el desarrollo de cualquier software es muy importante y necesario debido a que por esta es que se va guiando el desarrollo del software a través del conjunto de patrones y abstracciones.

2.9 Arquitectura de Software

En los años 1960 ya se acariciaba el concepto de arquitectura de software en los círculos de investigación. No obstante, toma popularidad en los años 1990 tras reconocerse la denominada crisis del software y como tema de interés de la incipiente disciplina de la ingeniería del software; por lo que uno de los conceptos que existen sobre esta disciplina es que la Arquitectura de Software es la organización

Capítulo 2: Tecnologías y Tendencias Actuales

fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (22)

¿Por qué se necesita una Arquitectura? Se requiere de una Arquitectura para que los desarrolladores puedan avanzar hasta tener una noción común de lo que se quiere hacer. Además de comprender el sistema, organizar el desarrollo, fomentar la reutilización y hacer evolucionar el sistema. (8)

Como el desarrollo de la aplicación se está desarrollando con el CMS Drupal, es importante destacar que la arquitectura que se debería utilizar es con la que éste trabaja internamente, una representación áspera del sistema Drupal podría ser la siguiente:

```
+-----+           +-----+           +-----+           +-----+
| Database | <==>   | Drupal modules | <==> | Drupal theme/template | <==> | html |
+-----+           +-----+           +-----+           +-----+
```

Drupal proporciona un marco muy flexible para la manipulación individual y personalizada de unidades de datos de contenido llamado nodos. Ha sido diseñado para tener una clara distinción entre los datos que maneja el sitio (absolutamente ‘ todos ‘ los que se encuentran en la base de datos), la " lógica de la aplicación", que toma estos datos y los organiza para su presentación (basándose en la URL a la que se ha hecho referencia, los datos que se enviaron, que usuario está solicitando la página), y la "capa de presentación" que no es más que el sistema de temas de Drupal.

Por lo tanto, para generar una página en concreto, los módulos de Drupal obtienen todos los datos necesarios de la base de datos, se les aplica un tema visual, y se envía de vuelta al navegador del cliente. El proceso de aplicarle un tema a los datos es esencialmente una cuestión de ejecutar una salida HTML sin estilo a través del fichero "XTemplate".

Los Módulos definen "hooks" que les permiten agregar funcionalidad en el flujo de ejecución de Drupal. Los "hooks" son funciones escritas en php, que utilizan una convención de nombres especial hook_* donde hook es remplazado por el nombre del módulo.

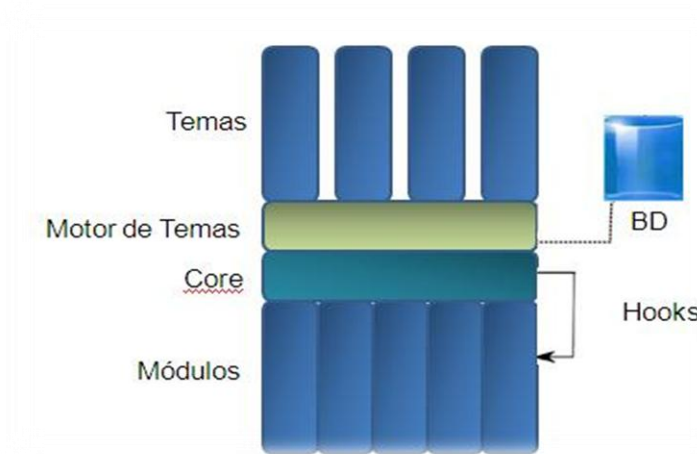


Figura 1: Arquitectura de Drupal

Drupal es perfectamente integrable con el patrón arquitectónico Modelo-Vista-Controlador. Para demostrar que esta afirmación es cierta hay muchas personas que desarrollan aplicaciones de tipo Web que tienen el criterio de que Drupal utiliza MVC adaptado y existen muchas soluciones ya probadas que muestran que es posible su integración. Por la afirmación anteriormente mencionada y teniendo en cuenta que el equipo de desarrollo presenta conocimientos sobre MVC, se ha tomado la decisión de utilizar este patrón arquitectónico.

2.9.1 Modelo-Vista-Controlador

Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio. (8)

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. (22)

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. (22)

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. (22)

Ciclo de vida del MVC

El ciclo de vida del MVC es normalmente representado por los tres componentes mencionados anteriormente y el cliente. A continuación se muestra a través de un diagrama el ciclo de vida de manera sencilla.

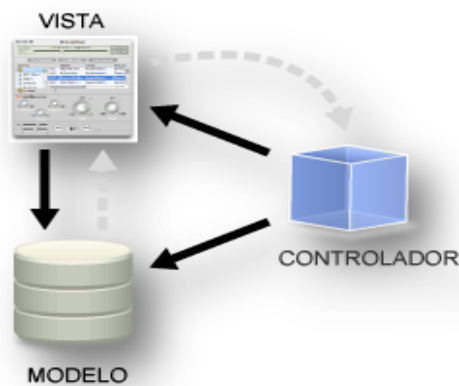


Figura 2: Ciclo de Vida del MVC

El primer paso en este ciclo comienza cuando el usuario realiza una solicitud al controlador con información sobre lo que él desea hacer. Entonces el controlador decide a quien debe delegar la tarea y es aquí donde el modelo empieza su trabajo. De aquí en adelante el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que solicita el controlador. Luego se determina su labor, le regresa al controlador la información resultante de sus operaciones, el cual a su vez redirige a la vista la cual es la encargada de transformar los datos en información visualmente entendible para el usuario.

Este modelo presenta una serie de ventajas que hacen más seguro su utilización:

- Se consiguen múltiples vistas del modelo. (22)
- Todas las vistas están sincronizadas.
- No acoplamiento, y facilidad de evolución, para cambiar las vistas y los controladores.

- La aplicación puede soportar un tipo de interfaz para cada rol en la aplicación.

A la hora de desarrollar un sitio Web o un portal Web se tiene que tener bien definido la estructura del mismo, es decir, de qué manera se le va a mostrar todo este contenido al usuario y a que usuarios. Para la ayuda de esta tarea existe la disciplina Arquitectura de Información la cual presenta una buena guía para que esta tarea se desarrolle de manera sencilla y que el producto final quede con la calidad y eficiencia requerida.

2.10 Arquitectura de Información

Con el desarrollo de Internet y por ende el de la cantidad de información que manejaba comenzó a aceptarse el término 'Arquitectura de Información' (en lo adelante AI) y ha tomado mucho auge con el desarrollo de los sitios Web en Internet e Intranets. Fue acuñado en 1976 por Richard Saul Wurman.

En el año 1998 dos profesionales de la información, devenidos en arquitectos de información, Louis Rosenfeld y Peter Morville, escribieron el libro Information Architecture for the World Wide Web donde definieron a la Arquitectura de Información como:

- La combinación de la organización, etiquetado y los esquemas de navegación dentro de un sistema de información. (23)
- El diseño estructural de un espacio de información para facilitar las tareas de acabado y acceso intuitivo a los contenidos. (23)

Para realizar una AI es necesario conocer las tareas que se deben de llevar a cabo para obtener un buen diseño de Interfaz del portal Web a desarrollar.

Tareas generales

- **La definición de los objetivos del producto:** Es la primera tarea que se realiza y debe llevarse a cabo a través de reuniones iniciales donde se debe recoger la mayor cantidad de información acerca del proyecto y del producto que se quiere.

- **El estudio del entorno, del mercado y los usuarios del producto:** Sobre el entorno se debe conocer los aspectos económicos, políticos, sociales y tecnológicos que pueden influir en la realización o no del producto. Para realizar esta parte de las preguntas sobre los usuarios al cliente en Cuba y en el mundo se hace a través de las metodologías existentes como son la entrevista, la encuesta, la observación.
- **El descubrimiento y organización de contenidos que llevará el producto:** Esta es la tarea de mayor importancia en la AI debido a que aquí es donde se aplicarán todas las herramientas de organización de manera lógica que permita estructurar el producto electrónico para que los usuarios queden satisfechos.
- **La representación de los contenidos ya organizados que contendrá el producto, a través de los diagramas:** La diagramación consiste en la realización de notaciones gráficas para representar la organización de la información y el funcionamiento de los procesos dentro de un producto electrónico (23). Se recomiendan 3 tipos de diagramas para esta tarea:
 - Diagrama de organización: Es el diagrama básico que ayuda a entender la estructura general del producto electrónico. En este se representan los grupos organizados y los elementos básicos que contienen.

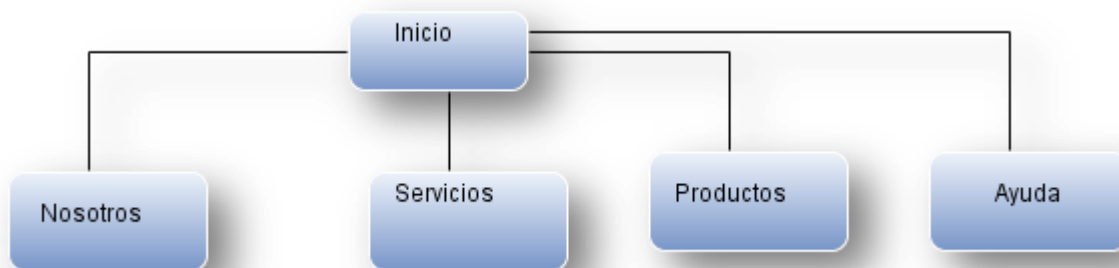


Figura 3: Diagrama de Organización

- Diagramas de funcionamiento: En este se representa la estructura con los flujos de navegación. Tiene un nivel superior al anterior complementándolo. Debe ser el que muestre los niveles de navegación así como los tipos de navegación en el producto.

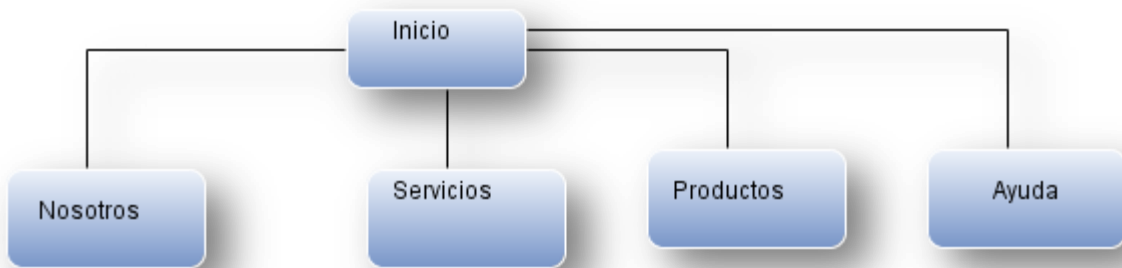


Figura 4: Diagrama de Funcionamiento

- Diagramas de presentación: Es el que debe mostrar las formas de organización visual de los contenidos en las páginas principales, por ejemplo: la página inicial, las páginas interiores, páginas de productos, etc. Este diagrama no pretende representar el diseño gráfico o diseño visual en detalle, sino especificar el esqueleto organizativo de la interfaz.



Figura 5: Diagrama de Presentación

2.11 Conclusiones parciales

En este capítulo se definieron de manera general las principales tecnologías que se utilizarán durante el desarrollo de esta aplicación, además de caracterizar y justificar la elección de la metodología a utilizar. No se puede dejar de mencionar que la elección de estas tecnologías fue la correspondencia de los requisitos establecidos por el cliente. De esta manera se arribó a la conclusión de que el uso de las mismas, es de gran importancia ya que contribuyen al desarrollo eficiente de la solución propuesta con un bajo costo, lo cual garantiza un mayor rendimiento.

CAPÍTULO 3: Presentación de la solución propuesta

3.1 Introducción

En este capítulo se dará a conocer la solución propuesta después de haber realizado un estudio inicial del problema a resolver. Además brindará información sobre los actores, trabajadores del sistema, los requerimientos funcionales, los casos de uso, con el objetivo de lograr un entendimiento del producto que se desea obtener.

3.2 Modelo de dominio

Durante el análisis de los procesos surgieron varios conceptos relacionados entre sí, estos definen el modelo del dominio, que permite a los usuarios, clientes, desarrolladores e interesados, utilizar un vocabulario común para poder entender el contexto en que se ubica el mismo, para capturar correctamente los requisitos y lograr una solución adecuada.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (8)

El objetivo del modelo del dominio es especificar y conocer las clases y objetos más significativos dentro del entorno del problema, lo que ayuda a determinar los procesos y roles más relevantes para el sistema a desarrollar. Esto ayuda a los usuarios, desarrolladores y clientes a la manipulación de un vocabulario común para así compartir el conocimiento manteniendo un lenguaje único y consistente que evite confusiones y posibilite el entendimiento de todas las partes interesadas.

3.2.1 Eventos principales del entorno

El **Portal** a realizar desarrolla diferentes **contenidos** como son **Noticia, Información, Investigación, Horario, Herramientas**. Para que un **usuario** (**autor, revisor, editor, administrador, planificador**) pueda acceder a estos **contenidos** para realizar cualquier acción (**registrar, editar, publicar, eliminar**) y

Capítulo 3: Presentación de la Solución Propuesta

luego almacenarla en la **base de datos** debe primeramente acceder al **Portal** y **registrarse**. Una vez dentro del mismo con el rol asignado podrán hacer cualquiera de las acciones mencionadas. Además de estos **usuarios registrados** también pueden acceder al sistema cualquier **usuario** sin tener rol, solamente para informarse y obtener conocimientos.

De los **usuarios registrados** el único que tiene acceso al **contenido** del **horario** es el **planificador** el cual es el encargado de gestionar (**crearlo, modificarlo, eliminarlo**) el mismo y almacenar todos los cambios en la **base de datos** según sea necesario. Los otros **usuarios** tengan un rol o no, sólo se informarán con él; y el **usuario administrador** es el máximo responsable de gestionar (**registrarla, publicarla, descargarla, eliminarla**) las **herramientas** que se almacenarán en un **repositorio** en el **Portal** para estar disponible para cualquier **usuario**.

Luego de haber visto los eventos principales del entorno a desarrollar, se realizará una breve descripción de las palabras más importantes y que pudieran traer problemas de entendimiento a la hora de su implementación o de ser leídos por cualquier persona ajena al equipo de desarrollo.

3.2.2 Conceptos principales del modelo de dominio

- **Portal:** Conjunto de páginas Web o sitios Web que sirve como punto de entrada único a la información de una institución o empresa.
- **Contenidos:** Es cualquier texto, imagen o sonido que los usuarios se encuentran en una interfaz del software.
- **Noticia:** Redacción o recorte de la realidad de un texto informativo divulgando temas referentes a elementos del DSD que se publicará posteriormente a todos los usuarios.
- **Información:** Contenido fundamental y necesario en todo proceso comunicativo para transmitir mensajes a los usuarios.
- **Investigación:** Contenido redactado por un actor, editor o revisor referente a algún tema específico del DSD.
- **Usuario:** Persona que interactúa con la aplicación Web para la navegación.
 - **Administrador:** Usuario registrado con capacidad total de manipular usuarios, registros y permisos. Puede editar los datos de los usuarios, bloquearlos, eliminarlos, cambiar su contraseña y asignarle otro nivel de acceso.

Capítulo 3: Presentación de la Solución Propuesta

- **Autor:** Usuario registrado con permisos para enviar contenidos, pero no puede editar los contenidos de los demás, ni publicar los que escribe él mismo. Estos contenidos quedan pendientes hasta que un revisor los autorice y publique.
 - **Editor:** Usuario registrado con permisos para enviar contenidos y editar los enviados por los demás, pero no puede publicar. Los contenidos quedan pendientes, al igual que el caso del Autor.
 - **Revisor:** Usuario registrado con permisos para enviar, modificar y publicar sus contenidos o los de otros usuarios.
 - **Planificador:** Usuario registrado con permisos para modificar y publicar el horario de tiempo de máquina.
-
- **Base de Datos:** Entidad en la cual se almacenan de manera estructurada y con la menor redundancia posible los datos referentes a las noticias, informaciones, investigaciones y los usuarios de la aplicación.
 - **Horario:** Proceso de planificación de trabajo para cada miembro del Departamento.
 - **Herramientas:** Aplicaciones informáticas que permiten la automatización de diferentes trabajos.
 - **Repositorio:** Sitio centralizado donde se almacenan y mantienen archivos digitales.
 - **Editar:** Realizarle cualquier modificación o cambio a un elemento determinado.
 - **Registrar:** Acción que se debe realizar para poder acceder a la parte administrativa del portal Web.
 - **Publicar:** Hacer algo público o del conocimiento de todos.
 - **Descargar:** Forma de obtener lo que se encuentra almacenado en un servidor.
 - **Eliminar:** Borrar cualquier acción que se realice.
 - **Modificar:** Arreglar, cambiar o transformar cualquier dato, contenido.

Después de haber explicado y especificado los conceptos del modelo de dominio se mostrará representativamente para una mejor visión y un mejor entendimiento del mismo.

3.2.3 Diagrama de clases del modelo de dominio

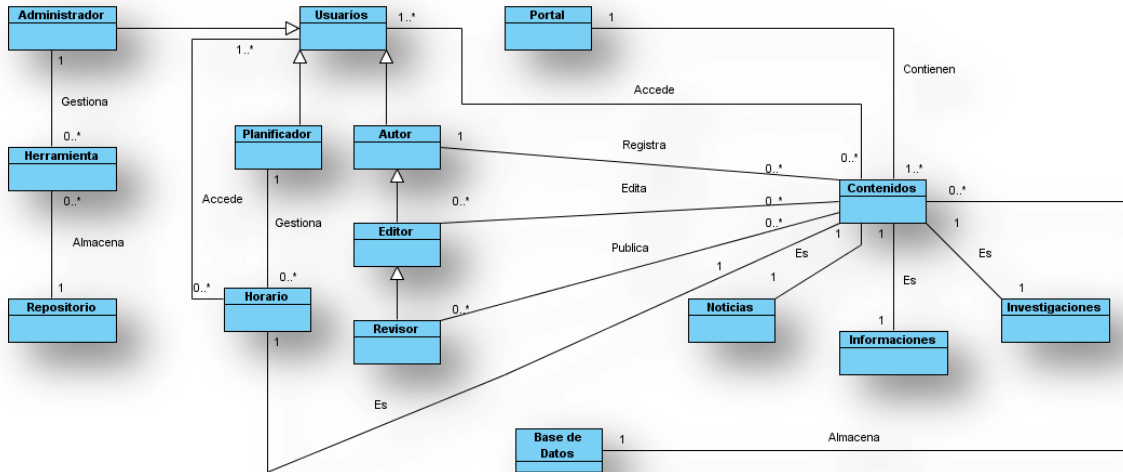


Figura 6: Diagrama de clases del Modelo de Dominio

3.3 Requerimientos del sistema.

Las empresas dedicadas al desarrollo de software se rigen por ciertas características y cualidades que hacen al producto final eficiente y confiable. Para el desarrollo de la aplicación propuesta se ha realizado una discusión minuciosa con el cliente acerca de sus necesidades. A través de los requerimientos es que el producto va a tomar forma, ya que es la vía más segura de que este quede según lo planteado por el cliente.

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. (8)

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

Capítulo 3: Presentación de la Solución Propuesta

Requerimientos Funcionales

La funcionalidad del sistema va a estar guiada por los requisitos funcionales, que no son más que las condiciones o capacidades que el sistema debe cumplir. La aplicación debe ser capaz de:

R1. Autenticar Usuario.		R11. Publicar Investigación.	
R2. Gestionar Usuario.	R2.1 Agregar Usuario.	R12.Gestionar Herramientas.	R12.1 Registrar Herramientas.
	R2.2 Editar Usuario.		R12.2 Publicar Herramientas.
	R2.3 Eliminar Usuario.		R12.3 Descargar Herramientas.
	R12.4 Eliminar Herramientas.		
R3. Registrar Noticia.		R13.Gestionar Horario.	R13.1 Crear Horario.
			R13.2 Modificar Horario.
			R13.3 Buscar Horario.
			R13.4 Eliminar Horario.
R4. Registrar Información.		R14.Gestionar Recursos.	R14.1 Crear Recursos.
			R14.2 Buscar Recursos.
			R14.3 Eliminar Recursos.
R5. Registrar Investigación.		R15.Gestionar Personal.	R15.1 Crear Personal.
			R15.2 Modificar Personal.
			R15.3 Buscar Personal.
			R15.4 Eliminar Personal.
R6. Modificar Noticia.		R16.Gestionar Proyecto.	R16.1 Crear Proyecto.
			R16.2 Modificar Proyecto.

Capítulo 3: Presentación de la Solución Propuesta

		R16.3 Buscar Proyecto.
		R16.4 Eliminar Proyecto.
R7. Modificar Información.	R17. Buscar Noticia.	
R8. Modificar Investigación.	R18. Buscar información.	
R9. Publicar Noticia.	R19. Buscar Investigación.	
R10. Publicar Información.		

Tabla 2: Tabla de los Requerimientos Funcionales

Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, así sean restricciones del entorno o de la implementación.

R.20 Requisitos de Software.

Cliente

Sistema operativo Linux (Ubuntu 7.10, Debían o superior en ambos casos), Windows (2000 o superior).

Servidor

Sistema operativo Linux (Ubuntu 7.10, Debían o superior en ambos casos), Windows (2000 o superior).

R.21 Requisitos de Hardware.

Cliente

Las PC donde se usará el software deben tener al menos 128 MB de memoria RAM como mínimo (recomendable 256 MB). Procesador Pentium de 2.0 GHz como mínimo.

Servidor

Las PC donde se usará el software deben tener al menos 256 MB de memoria RAM como mínimo (recomendable 512 MB o superior). Procesador Pentium de 2.8 GHz como mínimo. (Recomendable 3.0GHz). De disco duro se recomienda 80 GB.

R.22 Requerimientos de Usabilidad.

El Sistema podrá ser usado por todos los integrantes del DSD.

Capítulo 3: Presentación de la Solución Propuesta

R.23 Requerimientos de Seguridad.

La aplicación podrá ser utilizada solamente por las personas que autorice el departamento de dirección del DSD. Protección contra acciones que puedan afectar la integridad de los datos.

R.24 Restricciones en el Diseño y la Implementación.

La aplicación será desarrollada utilizando CMS Drupal. Como herramienta de modelado CASE se utilizará Visual Paradigm.

R.25 Requerimientos de apariencia o interfaz externa.

La aplicación debe tener una interfaz que brinde alta usabilidad para el usuario que la vaya a utilizar.

3.4 Descripción de los actores del sistema

Un actor es un rol que cumple un usuario, puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. (8)

Por las características peculiares del sistema se tienen los siguientes actores:

- **Usuario:** Persona que interactúa con la aplicación Web para la publicación y gestión de contenidos.
- **Administrador:** Usuario registrado con capacidad total de manipular usuarios, registros y permisos. Puede editar los datos de los usuarios, bloquearlos, eliminarlos, cambiar su contraseña y asignarle otro nivel de acceso.
- **Autor:** Usuario registrado con permisos para enviar contenidos, pero no puede editar los contenidos de los demás, ni publicar los que escribe él mismo. Estos contenidos quedan pendientes hasta que un revisor los autorice y publique.
- **Editor:** Usuario registrado con permisos para enviar contenidos y editar los enviados por los demás, pero no puede publicar. Los contenidos quedan pendientes al igual que el caso del Autor.
- **Revisor:** Usuario registrado con permisos para enviar, editar y publicar sus contenidos o los de otros usuarios.

Capítulo 3: Presentación de la Solución Propuesta

- **Planificador:** Usuario registrado con permisos para crear, modificar y publicar el horario de tiempo de máquina.

3.4.1 Casos de Uso del sistema

Un caso de uso no es más que la entrada fundamental al comportamiento del sistema precisando la secuencia de acciones que arroja valores para un actor que utiliza un sistema para concluir un proceso.

Casos de Uso	Clasificación
Autenticar Usuario.	Secundario
Gestionar Usuario.	Secundario
Registrar Noticia.	Secundario
Registrar Información.	Secundario
Registrar Investigación.	Secundario
Editar Noticia.	Secundario
Editar Información.	Secundario
Editar Investigación.	Secundario
Publicar Noticia.	Secundario
Publicar Información.	Secundario
Publicar Investigación.	Secundario
Gestionar Herramienta.	Secundario
Gestionar Horario.	Crítico
Gestionar Recurso.	Crítico
Gestionar Personal.	Crítico
Gestionar Proyecto	Crítico

Capítulo 3: Presentación de la Solución Propuesta

Buscar Noticia	Secundario
Buscar Información	Secundario
Buscar Investigación	Secundario

Tabla 3: Listado de CU del sistema

3.4.2 Diagrama de CU del sistema

Los diagramas de casos de uso sirven para determinar la comunicación y el comportamiento de un sistema a través de la interacción con los usuarios y/u otros sistemas. Los diagramas de casos de uso se utilizan para dibujar los requerimientos del sistema mostrando cómo reacciona una respuesta a eventos que se producen en el mismo.

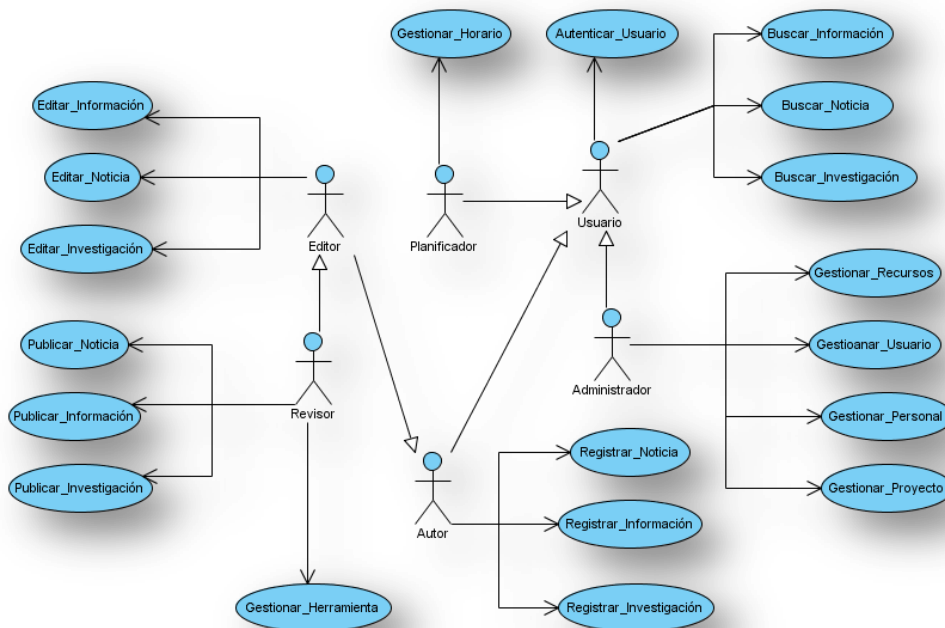


Figura 7: Diagrama de CU del Sistema

3.4.3 Descripción del CU fundamental del sistema

Después de haber realizado el diagrama de caso de uso del sistema es necesario hacer la descripción de cada uno de los casos de uso debido a que es importante que se sepa cómo es que va a trabajar cada uno de ellos según las solicitudes que realice el usuario. A continuación se muestra la descripción del caso de uso Gestionar Horario.

Caso de Uso:	Gestionar Horario.
Actores:	Planificador.
Resumen:	Comienza cuando el planificador accede al sistema para crear, modificar o eliminar un horario. Luego el sistema ejecuta la opción que se escogió dándole paso al planificador a terminarla. Con la culminación de las mismas el sistema muestra un mensaje indicando que las opciones fueron realizadas satisfactoriamente.
Precondiciones:	Solo puede realizar esta operación el planificador.
Referencias:	R13 R13.1 R13.2 R13.3 R13.4

Tabla 4: Descripción CU Gestionar Horario

Para el desarrollo de los diagramas de casos de uso es recomendable utilizar patrones de casos de uso debido a que estos garantizan una correcta definición y descripción de los casos de uso, así como el desarrollo de diagramas más funcionales, compactos, claros y entendibles por los miembros del equipo de trabajo, permitiendo ganar en tiempo y organización.

3.4.4 Patrones de CU

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Expresado lo

Capítulo 3: Presentación de la Solución Propuesta

anterior con palabras más simples, el patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. (22)

3.4.5 Patrones de CU utilizados

Los patrones aplicados fueron los siguientes:

- **Múltiples actores: Rol común:** Este patrón permite que varios actores inicien uno o varios casos de uso mediante un mismo rol. Se aplicó este patrón al existir varios actores del sistema desempeñando el mismo papel hacia un caso de uso por lo que se hizo necesario representarlos mediante otro actor, del cual heredan los actores que comparten este rol. (24)
- **CRUD:** Este patrón es aplicado en algunos casos de uso como **Gestionar Herramienta**, **Gestionar Horario y Gestionar Usuario** entre otros, ya que en estos se agrupan las funciones básicas (Adicionar, Editar y Borrar) en una sola haciendo el modelo más claro. (24)

3.5 Conclusiones parciales

Durante el desarrollo de todo el capítulo se llegan a resultados sustanciales para la puesta en marcha de la futura aplicación tales como: Descripción detallada de la propuesta de solución a través de la modelación del dominio propuesto; identificando los actores y trabajadores que intervienen, así como los Casos de Uso correspondientes. Se identifican los requisitos funcionales y no funcionales con los cuales debe cumplir la aplicación, llegando de esta forma a una Modelación detallada del sistema a desarrollar.

CAPÍTULO 4: Construcción de la Solución Propuesta

4.1 Introducción

El objetivo fundamental de la investigación es el desarrollo de un portal Web a través de un CMS. Este implementa casi todos los requerimientos funcionales descritos en el capítulo anterior, menos Gestionar Horario, Gestionar Personal, Gestionar Proyecto y Gestionar Recursos, los cuales son funcionalidades nuevas que se le incluyen al portal Web a través de módulos. El presente capítulo describe como se realizó el análisis y diseño de la aplicación Web para estos módulos presentando los diagramas correspondientes a estos flujos de trabajo a través del proceso Gestionar Horario.

4.2 Modelo de análisis

El modelo de análisis es una parte fundamental del flujo de trabajo de Análisis y Diseño debido a que ayuda a refinar y estructurar los requisitos funcionales. Es uno de los artefactos que se genera en este flujo de trabajo, proporcionando un mayor entendimiento para los desarrolladores pues se logra obtener una visión más clara de qué debe hacer el sistema. Se desarrolla utilizando el lenguaje de los desarrolladores y es la primera aproximación al modelo de diseño es por ello su vital importancia, siendo un paso opcional a desarrollar. En este modelo los casos de uso del sistema se describen a través de las clases del análisis y sus objetos.

4.2.1 Diagramas de clases del análisis

El diagrama de clases del análisis es uno de los artefactos que se generan en el modelo del análisis. Se utilizan para el modelado de las realizaciones de los casos de uso, mostrando las clases y sus relaciones. Las clases que se encuentran en estos diagramas se clasifican en **Interfaz**, **Control** y **Entidad**.

Las clases interfaz se encargan de modelar la interacción del actor con el sistema. Representan abstracciones de ventanas, formularios, así como la comunicación con otros sistemas o dispositivos. Las clases controladoras modelan los aspectos dinámicos del sistema, de forma tal que puedan coordinar las

Capítulo 4: Construcción de la Solución Propuesta

acciones y los flujos de control. Las clases entidad modelan información que poseen una larga vida y que a menudo son persistentes. Suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema. A continuación se muestra un ejemplo con el Diagrama de Clases del Análisis para el caso de uso Gestionar Horario.

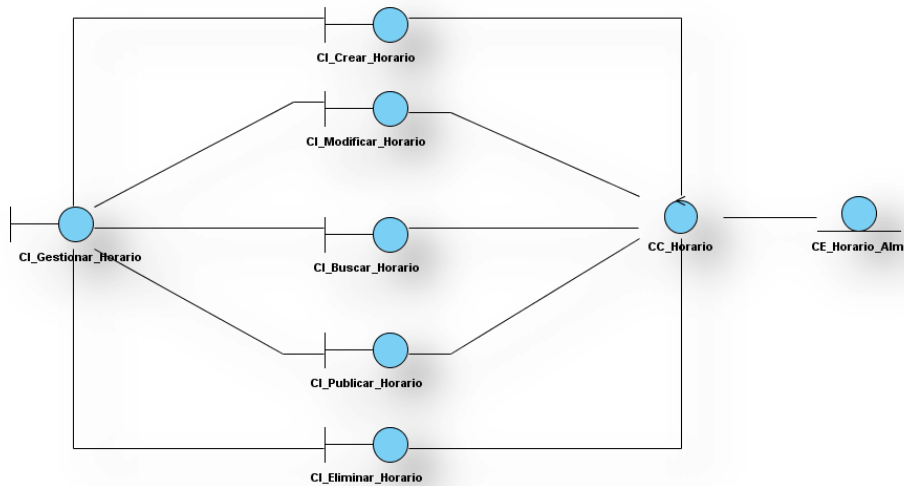


Figura 8: Diagrama de Clases del Análisis CU Gestionar Horario

4.3 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos del sistema, son la realización de los casos de uso. Modelan instancias concretas o prototípicas de clases interfaces, componentes y nodos, así como los mensajes enviados entre ellos. Los diagramas de interacción se denotan en dos tipos diferentes, los diagramas de secuencia y diagramas de colaboración, que en su generalidad los dos responden a realizaciones de los casos de usos correspondientes, en el desarrollo de este trabajo se realizó específicamente los diagramas de colaboración.

4.3.1 Diagrama de colaboración

El diagrama de colaboración muestra las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. Este presenta elementos como: objetos y enlaces. Los objetos representan las clases con sus nombres y los enlaces que constituyen una instancia de una asociación en un diagrama de clases. Los enlaces están acompañados por un número que indica el orden dentro de la interacción, simbolizando el orden en que se van a efectuar las llamadas y respuestas del sistema. A continuación un ejemplo del caso de uso Gestionar Horario, la sección de Crear Horario.

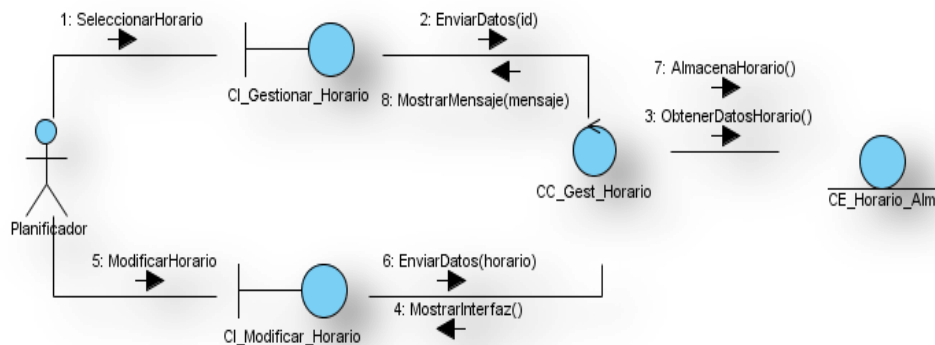


Figura 9: Diagrama de Colaboración CU Gestionar Horario Sección Modificar Horario

4.4 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con 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 utilizado como entrada fundamental de las actividades de implementación.

4.4.1 Diagrama de clases del diseño

Los diagramas o modelos de clases del diseño sirven para la toma de decisiones en el diseño y la implementación. Los modelos brindan una vista a menor y menos costosa escala de la solución final y resulta mucho más fácil efectuar cambios a los mismos una vez detectados los problemas, que tener que efectuar cambios a la solución ya implementada. A continuación se muestra el diagrama de clases del diseño del CU Gestionar Horario.

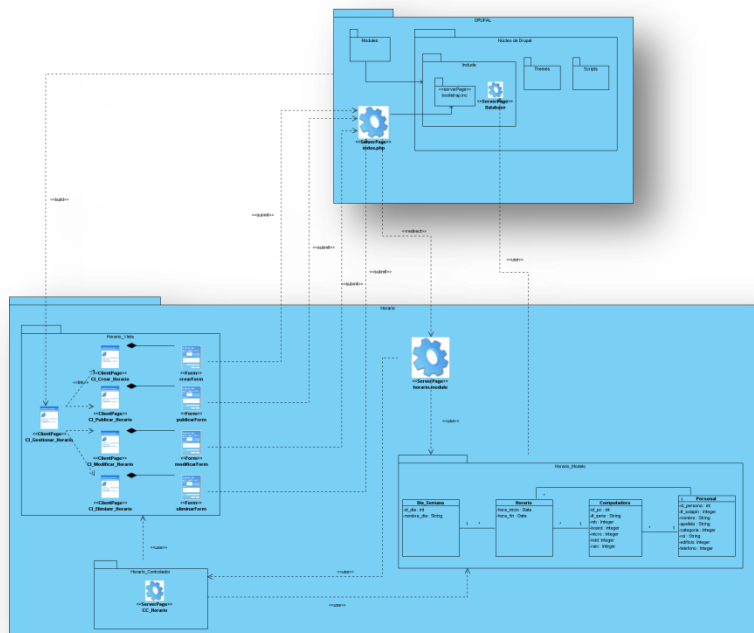


Figura 10: Diagrama de Diseño CU Gestionar Horario

4.4.2 Descripción del diseño

Para entender de manera clara el diagrama de clases del diseño de la aplicación en conjunto con las clases del CMS Drupal, se explicará a continuación el mismo.

El Sistema de Administración de Contenidos Drupal es una plataforma flexible que permite muchas opciones como: cambiar el diseño del sitio, la interacción de los usuarios con él, los tipos de contenido que se pueden mostrar. Provee un tipo de contenido genérico llamado "node", y permite la creación, edición y publicación de este tipo de contenido. Los documentos creados se depositan en una base de datos

Capítulo 4: Construcción de la Solución Propuesta

central, donde también se guardan el resto de datos. Posee un robusto sistema de seguridad basado en la creación de usuarios y roles, en dependencia de los permisos que se necesiten para administrar el sitio y acceder a los módulos. El diagrama de clases del diseño está dividido en 2 paquetes, uno es del módulo Horario y el otro el que utiliza Drupal.

El paquete de Drupal está contenido fundamentalmente por:

Modules: Paquete que almacena todos los módulos que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.

Index.php: Página que recibe todas las peticiones que realizan los usuarios, enviándolas posteriormente al bootstrap.inc para que se siga el flujo de actividades.

Núcleo de Drupal: El mismo está compuesto por el paquete **includes** que contiene un conjunto de ficheros imprescindibles para el funcionamiento del Drupal como es el caso del **bootstrap.inc** que es el encargado de activar los módulos según las peticiones que estén solicitando los usuarios, el fichero **database.inc** que proporciona las funcionalidades de acceso a la base de datos. Otro paquete es el **themes**, en el cual se almacenan todas las plantillas de los diseños que se le quieran aplicar a la aplicación Web; otro paquete es el **scripts** que contiene un conjunto de ficheros indispensables para el funcionamiento del Drupal, fundamentalmente orientados al aspecto visual, CSS y Javascript.

El paquete del módulo del Horario está dividido según el patrón arquitectónico Modelo Vista Controlador:

Horario_Vista: En este paquete se van a encontrar todas las páginas clientes y los formularios que interactuarán con el usuario directamente.

Horario_Modelo: En este paquete se encontrarán todas las tablas que se relacionen con la petición recibida, encargándose de gestionar toda la información y buscar las respuestas para los usuarios.

Horario_Controlador: En este paquete se encuentra la página servidora CC_Horario que es la clase controladora del módulo, la cual distribuirá las acciones pedidas por los usuario.

horario.module: Contiene diferentes funcionalidades que permiten al módulo interactuar con el núcleo del Drupal.

El diseño de Drupal implementa una serie de Patrones de Diseño GoF que a continuación se mencionarán algunos de ellos.

4.4.3 Patrones de Diseño

Gran parte de la estructura interna de Drupal es más complicada que una simple herencia e interacción a través de mensaje, sin embargo, las características más interesantes del sistema establecido es el uso de patrones de diseño de software. Drupal, la mayoría de los patrones que utiliza pertenecen a la banda de los cuatro, GoF¹, por ejemplo:

Drupal tiene una única página servidora que atiende todas las peticiones en la aplicación, a este patrón se le denomina **Singleton (Instancia única)** que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Se puede nombrar también el patrón **Bridge (Puente)**, este permite añadir soportes para más sistemas de base de datos sin la necesidad de modificar el código de los módulos debido a que éstos deben ser escritos de manera independiente del sistema de base de datos que se esté utilizando.

El **Chain of Responsibility (Cadena de responsabilidad)** permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada, es decir, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y qué función se llama para hacer el trabajo, siguiendo así el flujo de trabajo de este patrón hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario o la cadena se ha agotado.

Decorator (Envoltorio), patrón que añade funcionalidades a una clase dinámicamente, ejemplo, **Node** es la unidad mínima de contenido que se puede almacenar, todos los tipos de contenidos incorporan la información de **Node** y definen los propios. El uso de `hook_nodeapi()` permite a los módulos extender arbitrariamente el comportamiento de todos los Nodos. Ejemplo: Una página, que es un tipo de contenido del Dupal tiene las atributos básicos (Nombre, fecha de creación, cuerpo entre otras.), si se deseara subir ficheros y adjuntarlo a este contenido implicaría modificar sus atributos. El módulo de Drupal **upload** satisface esta necesidad de forma mucho más modular, mediante **nodeAPI** interrelaciona cada **Nodo** que lo solicite posibilitando tener archivos adjuntos de forma sencilla.

Otro patrón que se puede mencionar es **Observer (Observador)**, el cual es el encargado de definir una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Cuando se hace una

¹ Gang of Four: Grupo de autores compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides que escribieron el libro Design Patterns donde se recogen 23 patrones de diseño comunes.

Capítulo 4: Construcción de la Solución Propuesta

modificación a un vocabulario de la **taxonomía** del sistema de Drupal, el gancho de **taxonomía** se llama en todos los módulos que lo implementan y los cambios se podrán tratar de forma apropiada.

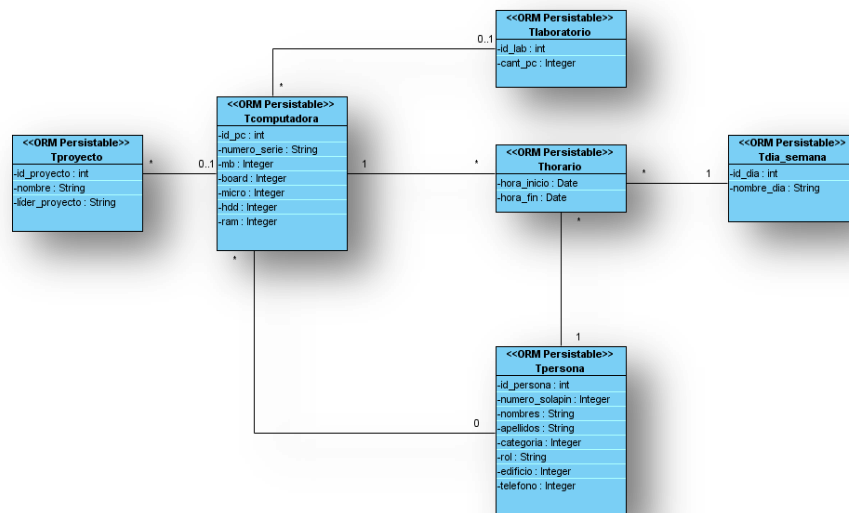
Command (Orden) encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. Muchos de los ganchos de Drupal utilizan el **Command** para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro junto con los argumentos. De hecho, el propio sistema de gancho utiliza este modelo, a fin de que los módulos no tengan que definir cada gancho, sino sólo las que necesiten implementar.

Los sistemas de base de datos almacenan información y permiten a los usuarios recuperarla y actualizarla según sus solicitudes, por lo que es muy importante que se realice un diseño de la base de datos.

4.5 Diseño de la base de datos

El modelo de datos de la aplicación está enfocado en el caso de uso Gestionar Horario y los que se relacionan con él, debido a que como la aplicación se está desarrollando con el CMS Drupal, todos los demás son manejados por el CMS por lo que no se presentarán en el modelo.

Para comenzar a diseñar la base de datos se parte del diagrama de clases persistentes. La persistencia de dichas clases se debe a que los valores que esta posea perdurarán independientemente del tiempo y el espacio.



Capítulo 4: Construcción de la Solución Propuesta

Figura 11: Diagrama de Clases Persistentes

4.6 Modelo de datos

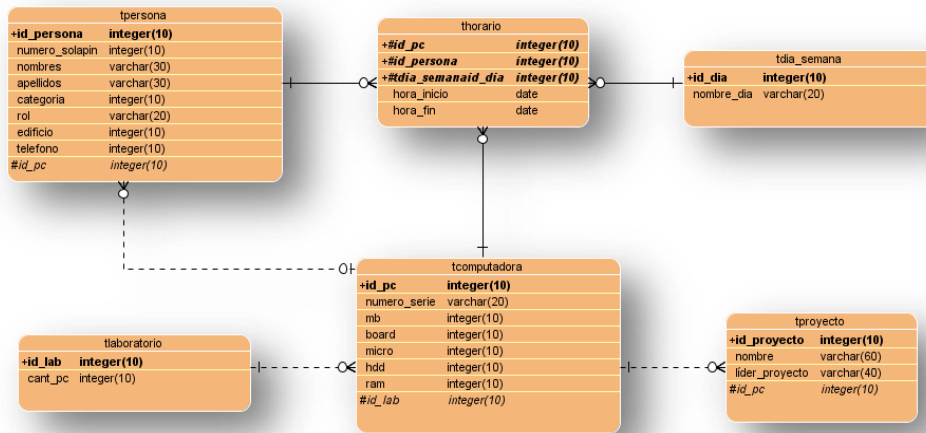


Figura 12: Modelo de Datos

4.7 Diagrama de Despliegue

Un diagrama de despliegue es un modelo de objeto donde se describe la distribución física del sistema, es decir, la relación física entre los componentes hardware y software en el sistema final. A continuación se muestra el Diagrama de Despliegue de la aplicación en ejecución.

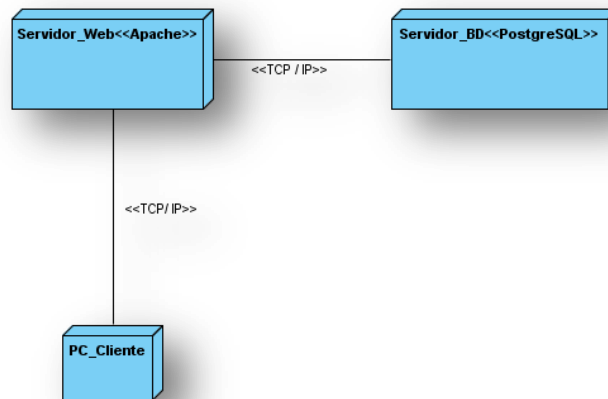


Figura 13: Diagrama de Despliegue

4.8 Modelo de implementación

El modelo de implementación describe los elementos del modelo de diseño y las clases que se implementan en términos de componentes. Además de cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponible en el entorno de implementación y en el lenguaje utilizado, y cómo dependen los componentes unos de otros.

4.8.1 Diagrama de componente

El diagrama de componentes describe cómo interactúan los componentes físicos, o sea, muestran las dependencias lógicas que existe entre componentes software, ya sean ejecutables, de código fuente o binario, se utiliza para modelar la vista de implementación estática del sistema. A continuación se muestra el diagrama de componente referente al caso de uso "Gestionar Horario".

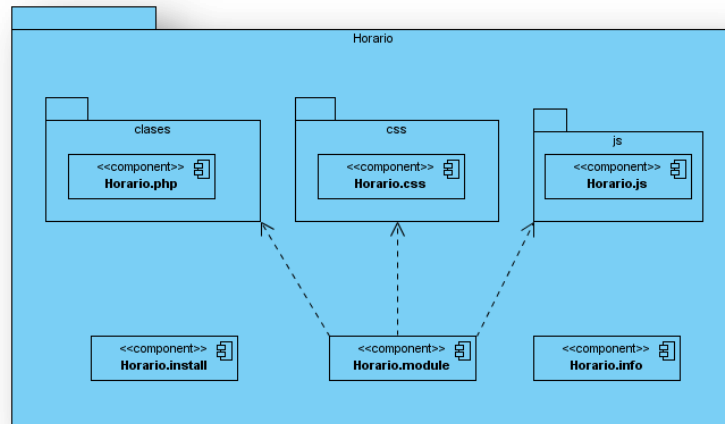


Figura 14: Diagrama de Componentes

4.9 Estándar de Codificación

El estándar de codificación en la implementación es muy importante en cualquier producto, principalmente si este involucra muchos desarrolladores. Esto asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener.

Principios Generales

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.

Elemento	Reglas de nombramiento
	Nombre sustantivo singular, con la primera letra en

Capítulo 4: Construcción de la Solución Propuesta

Clases, interfaces y archivos fuente	mayúscula y las demás en minúsculas. Si el nombre de la clase está comprimido en más de una palabra la primera letra debe de ser mayúscula. Letras mayúsculas seguidas no son permitidas. Ejemplo: “Zend_PDF” no es permitido mientras que “Zend_Pdf” sí lo es.
Constantes	Nombre sustantivo en mayúsculas. Para separar palabras se usará el guión bajo: _
Paquetes y directorios	Nombre sustantivo singular en minúsculas.

Tabla 5: Nomenclatura de los elementos.

Comentarios

Cada clase deberá comenzar con un comentario que incluya:

- Objetivo, o problema que resuelve el la clase
- Algoritmo.

Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.
- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

Se utilizará dos tipos de comentarios.

```
/* Se puede utilizar en el caso de un atributo, de un método o algún segmento de código */
```

Otros Comentarios Internos

```
// .....  
// .....
```


Capítulo 4: Construcción de la Solución Propuesta

Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.
- Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.

Identificadores de variables

Los identificadores de variables comenzarán siempre con la primera letra minúscula.

Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (_).

Ejemplo:

nombre_del_usuario

Indentación

- Define que deben usarse los sangrados para indentar el código y nunca espacios.

Saltos de Línea

- Añadir un salto de línea después del cierre de los paréntesis de los parámetros.
- Añadir un salto de línea después un punto y coma, cuando termina la sentencia.

Espacios y líneas en blanco

- Usar espacios en blanco para mejorar la legibilidad del código.
- Usar espacios en blanco en ambos lados del operador de símbolos, después de comas y después de las declaraciones.
- Usar líneas en blanco para separar trozos de código de alta complejidad.

Longitud de la línea

Capítulo 4: Construcción de la Solución Propuesta

- Evitar las líneas de más de 80 caracteres, cuando se supere esta cifra se debe reorganizar el código usando algún principio descrito anteriormente.

String Literales

- Cuando se le asigna un texto literal (sin contenido de variables) se utilizarán comillas simples.

```
$a = 'Texto de ejemplo';
```

String Literales con apóstrofes

- Cuando se le asigna un texto literal con apóstrofes se utilizarán comillas dobles, este caso es principalmente en las sentencias SQL.

```
$sql = "SELECT `id`, `name` from `some_people` WHERE `name`='Fred' OR `name`='Susan'";
```

Concatenación

- Para concatenar Strings se utilizará el operador "." (punto), con un espacio entre medio para mejorar la lectura.

```
$company = 'Zend' . ' ' . 'Technologies';
```

Arrays

En aquellos arrays de índices numéricos estos deberán ser números positivos, en la declaración de los valores del array se dejará un espacio en blanco luego de la coma para mejorar la lectura.

Clases

- En la declaración de la clase el nombre deberá seguir la convención ya expresada anteriormente, debe estar precedida por una documentación, luego la declaración y desarrollo.

```
/**
```

Capítulo 4: Construcción de la Solución Propuesta

```
* Documentación del bloque aquí.  
*/  
class SampleClass {  
// Todo el contenido de la clase aquí  
// Identada por cuatro espacios vacíos  
}
```

Control de flujo

- En las declaraciones `if/then/else` deberá tener un espacio antes y después del paréntesis condicional, en la misma línea se abre llave y se cierra en una línea diferente, lo mismo se aplica al `elseif`, a continuación un ejemplo que lo ilustra:

```
if (condition1 || condition2) {  
    action1;  
}  
elseif (condition3 && condition4) {  
    action2;  
}  
else {  
    defaultaction;  
}
```

- En el caso de un **Switch** la convención es la misma, adicionando que el `break` tiene una indentación mayor que el resto a continuación un ejemplo:

```
switch (condition) {  
    case 1:  
        action1;  
        break;  
  
    case 2:  
        action2;  
        break;  
  
    default:
```

Capítulo 4: Construcción de la Solución Propuesta

```
defaultaction;  
}
```

La opción default jamás deberá ser omitida en un Switch.

4.10 Conclusiones parciales

Al arribar al término de este capítulo se ha llegado a la conclusión de que son muy importantes todos los diagramas realizados en esta fase de construcción para hacer más entendible lo que el usuario quiere y lo que se debe de implementar. Además hay que tener en cuenta que para el usuario es fundamental obtener una rápida y eficiente respuesta de sus solicitudes y esto se logra a través de un buen diseño de la base de datos.

CAPITULO 5: Validación de la Solución Propuesta

5.1 Introducción

Una prueba es el proceso mediante el cual se establece la existencia de errores. Con la culminación de un producto de software siempre se lleva a cabo la realización de las pruebas para comprobar la calidad del mismo. En el presente capítulo se argumentará brevemente este tema y se comentará el tipo de prueba escogido para medir el producto final.

5.2 Pruebas

La prueba de un sistema se define como el proceso de ejercitar o evaluar el sistema, por medios manuales o automáticos, para verificar que satisface los requerimientos o para identificar diferencias entre los resultados esperados y los que producen el sistema (IEEE).

El objetivo de las pruebas del software es convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su uso operacional, es decir, las pruebas es un proceso con el que se intenta proporcionar confianza en el software.

Luego de haber estudiado los diferentes tipos de pruebas que propone la ayuda del Rational, se decidió que la prueba de Dimensión de calidad, dentro de esta la de Función es la que se le realizará al sistema ya que se centra en la validación de funciones del destino de la prueba. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas. (25)

Dentro de esta prueba se encuentran las técnicas de caja blanca y de caja negra, en este caso se utilizará la de caja negra. A esta técnica se le aplican métodos para poder desempeñar dicha prueba y obtener un resultado, en este caso el método aplicado será el de partición equivalente. Este método divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores (por ejemplo, proceso incorrecto de todos los datos de carácter). La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. (26)

A continuación se muestra un ejemplo de las pruebas realizadas a través del caso de uso Gestionar Horario.

5.2.1 Partición Equivalente

5.2.1.1 Secciones a probar en el Caso de Uso

Crear Horario.

Modificar Horario.

Buscar Horario.

Eliminar Horario.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Crear Horario	EC 1.1: Se crea un horario correctamente.	El sistema muestra el horario vacío para insertar los datos. Primero se selecciona la hora inicio, luego la hora fin y después al personal.
SC2: Modificar Horario	EC 2.1: Se modifica el horario deseado.	El sistema muestra el horario a modificar con los siguientes datos: personal, hora inicio, hora fin. El sistema modifica los datos.
	EC 2.2: Se cancela la modificación.	Luego de haber modificado los datos, se puede cancelar dicha modificación, y el sistema no modifica.
	EC 3.1: Se marca la opción buscar.	El sistema muestra una lista con todos los horarios y sus datos.

SC3: Buscar Horario	EC 3.2: Se introduce el horario que se desea buscar.	El sistema muestra el horario y sus datos.
SC4: Eliminar Horario	EC 4.1: Se elimina horario.	El sistema da la opción de eliminar, se selecciona en la lista el horario que se desea eliminar y se elimina.
	EC 4.2: No se elimina.	Si no se desea eliminar luego de haber apretado el botón eliminar, este tiene la opción de cancelar y no se realiza dicha eliminación.

Tabla 6: Prueba al CU Gestionar Horario

5.2.1.2 Descripción de variables

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Sesión	Campo de selección	No	Solo letras
2	Personal	Campo de selección	No	Solo letras
3	Guardar y Salir	Botón	Si	Botón para guardar
4	Registrar	Botón	Si	Botón para registrar

Tabla 7: Descripción de las variables del CU Gestionar Horario

5.2.1.3 Matriz de Datos

SC1: Crear Horario

Escenario	1	2	3	4	Respuestas del Sistema	Resultado de la	Flujo Central

Capítulo 5: Estudio de Factibilidad

						Prueba	
EC 1.1: Se crea un horario correctamente.	V	V	V	NA	El sistema crea el horario.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Horario de producción • Crear nuevo horario
EC 1.2: Se cancela la creación del nuevo horario.	V	V	V	V	Se cancela la creación del nuevo horario.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Horario de producción • Crear nuevo horario • Cancelar

Tabla 8: Prueba a la SC1: Crear Horario

SC2: Modificar Horario

Escenario	1	2	3	4	Respuestas del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Se modifica el horario deseado.	V	V	V	NA	El sistema modifica los datos del horario.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Horario de producción • Modificar
EC 2.2: Se cancela la modificación.	V	V	V	NA	Se cancela la modificación de los datos.	Satisfactorio.	<ul style="list-style-type: none"> • Principal • Horario de producción • Aceptar

Capítulo 5: Estudio de Factibilidad

							• Cancelar
--	--	--	--	--	--	--	------------

Tabla 9: Prueba a la SC2: Modificar Horario

SC3: Buscar Horario

Escenario	Buscar	Respuestas del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1: Se marca la opción buscar.	V	El sistema muestra una interfaz para que se llenen los campos de proyecto y pc para buscar el horario deseado.	Satisfactorio.	<ul style="list-style-type: none"> Principal Horario de producción Buscar
EC3.2: Se introduce el recurso que se desea buscar.	V	El sistema muestra el recurso y sus datos.	Satisfactorio.	<ul style="list-style-type: none"> Principal Horario de producción Buscar

Tabla 10: Prueba a la SC4: Buscar Horario

SC4: Eliminar Horario

Escenario	1	2	3	4	Respuestas del Sistema	Resultado de la Prueba	Flujo Central
EC4.1: Se elimina un horario.	V	V	V	V	El sistema elimina los datos del horario.	Satisfactorio.	<ul style="list-style-type: none"> Principal Horario de producción Modificar
EC 4.2: No se elimina.	V	V	V	V	Se cancela la eliminación de los datos.	Satisfactorio.	<ul style="list-style-type: none"> Principal Horario de producción

							<ul style="list-style-type: none">• Aceptar• Cancelar
--	--	--	--	--	--	--	--

Tabla 11: Prueba a la SC5: Eliminar Horario

5.3 Conclusiones Parciales

Luego de haber aplicado las pruebas al sistema se llegó a la conclusión de que el producto obtenido no presenta ningún problema en cuanto a validaciones, ni de interfaz, por lo que se puede decir que el portal Web cumple con las expectativas del usuario.

Conclusiones

En el mundo existe hoy una tendencia creciente de llevar al formato electrónico toda la información de las empresas o entidades. Esto facilita el comercio electrónico de las mismas, garantizando de esta manera la difusión de sus productos, ya sean de software, de hardware, inmobiliarios de empresas, de oficinas, culturales, deportivos, tiendas virtuales y muchos otros productos comerciales. Al llegar al final de esta investigación se arribaron a una serie de conclusiones que a continuación se muestran:

- Para darle cumplimiento a la situación problemática mencionada al inicio de la investigación se realizó un análisis exhaustivo de las diferentes tendencias, herramientas y tecnologías actuales, para desarrollar el sistema concebido satisfaciendo las necesidades del cliente y garantizando la construcción de una aplicación estable que se pudiera mantener.
- El Sistema de Gestión de Contenido Drupal es una tendencia actual en el desarrollo de portal Web debido a que es un CMS de código abierto, es multiusuario, multiplataforma, multilenguaje, extensible y modular.
- Con el uso de este CMS se hizo necesario utilizar un Sistema Gestor de Base de Datos eficiente para la gestión de la información del portal Web como es PostgreSQL.
- Buscar un IDE de desarrollo que utilizara el lenguaje de programación que utiliza Drupal (PHP) como es el caso de Eclipse.
- Para tener una visión más clara y entendible de que es lo que quería el usuario se hizo necesario utilizar una herramienta de modelado integrable con dicho IDE como es el caso de Visual Paradigm.
- Escoger una metodología de desarrollo que en este caso fue RUP y luego de analizar varias metodologías de desarrollo se llegó a una estandarización de un procedimiento de trabajo para el desarrollo de portales Web basados en el CMS Drupal que incluye un ciclo de trabajo completo, un equipo de desarrollo dividido en roles, tareas establecidas en cada fase del ciclo de trabajo y documentación en cada fase.
- Con la puesta en marcha del sistema se logró cumplir con el objetivo general trazado al inicio del proceso de desarrollo y con la implementación del módulo Gestionar Horario, esperando que se le

dé solución a los problema que existente dentro del Departamento de Señales Digitales, cumpliendo además con las tareas trazadas para el buen desarrollo de portal Web.

Recomendaciones

Con la culminación del proceso de investigación y realización de la aplicación se considera necesario hacer las siguientes recomendaciones:

- Agregar nuevas funcionalidades de acuerdo a las expectativas de los usuarios finales por parte de los desarrolladores vinculados al proyecto.
- Que este documento sea consultado por los futuros desarrolladores de la aplicación a la hora de realizar cualquier actualización futura.
- Enriquecer constantemente los contenidos y servicios brindados por el sistema con el objetivo de que sea un pilar de referencia en el ámbito productivo del Departamento de Señales Digitales.

Bibliografía

1. WopSolutions. *WopSolutions*. [En línea] [Citado el: 10 de Noviembre de 2009.] <http://www.wopsolutions.com/ventajas-de-las-aplicaciones-web.html>.
2. Esencia Humana. [En línea]
<http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html>.
3. SIIMIT S.A. [En línea]
http://despertarydesarrollodeconciencia.com/siimt/index.php?option=com_content&view=article&id=23&Itemid=23.
4. Mosaic. [En línea] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>.
5. Ortíz, Antonio. Error500. [En línea]
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
6. CAVSI. [En línea] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
7. Instituto Geográfico Agustín Codazzi. [En línea]
http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistemasdegestiondebasededatos.htm.
8. Jacobson, Ivar. *Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000. ISBN:9788478290369.
9. Pressman, R. *Software Engineering. A Practitioner's Approach*. . USA : s.n., 1999.
10. Drupal. *Drupal*. [En línea] [Citado el: 16 de Noviembre de 2009.] <http://drupal.org/about>.
11. Drupal. [En línea] <http://drupal.org.es/drupal>.
12. PostgreSQL. *PostgreSQL*. [En línea] [Citado el: 7 de Octubre de 2009.] <http://www.postgresql.org/about/> .
13. PostgreSQL . [En línea] http://www.postgresql-es.org/sobre_postgresql.
14. Links Global Services. [En línea] http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
15. Lanzillotta, Analía. Mastermagazine. *Mastermagazine*. [En línea] [Citado el: 12 de Octubre de 2009.]
<http://www.mastermagazine.info/termino/5560.php> .
16. PHP Oficial Home Page. *PHP Oficial Home Page*. [En línea] [Citado el: 7 de Octubre de 2009.]
<http://www.php.net/manual/en/faq.general.php>.
17. Zend. [En línea] [Citado el: 18 de Mayo de 2010.] <http://www.zend.com/products/studio/>.
18. phpedit. [En línea] [Citado el: 18 de Mayo de 2010.] <http://www.phpedit.com/>.
19. Desarrollo Web. [En línea] [Citado el: 18 de Mayo de 2010.] <http://www.desarrolloweb.com/articulos/php-designer.html>.
20. Atenas. [En línea] http://www.atenas.cult.cu/rl/informatica/manuales/sl/introduccion_al_SL/eclipse.html.
21. John Worsley, J.D. *PostgreSQL Practico(Practical PostgreSQL)*. 2001.
22. Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN: 970-17-0261-1.
23. León, Lic. Rodrigo Ronda. *Productos electrónicos: principios y pautas*. 2005.
24. Zuñiga, Yurisleidys Leiva. *Análisis del Módulo de Soporte del Proyecto Informatización del Convenio Cuba Venezuela*. Ciudad de La Habana : s.n., 2008.

25. Corporation, IBM. *Ayuda Rational Unified Process*. 2006.
26. Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*.
27. Sommerville, Ian. *Ingeniería del Software. Séptima Edición*. Madrid : Pearson Educación. S.A, 2005. ISBM: 84-7829-074-5.
28. *Ventajas del CMS (Sistema de Manejo de Contenido)*. Miranda, Rubén.
29. Larman. "UML y patrones" Tomo I Capítulos 18.
30. Universidad del Valle, Cali, Colombia. *Universidad del Valle, Cali, Colombia*. [En línea]
http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/PRUEBASoftware.pdf.
31. Universidad Carlos III de Madrid. *Universidad Carlos III de Madrid*. [En línea] [Citado el: 25 de Noviembre de 2009.]
http://www.uc3m.es/portaHelp2/ohw/state/content/locale.es/vtTopicFile.welchelp_hs_es%7Cwelcport~htm/navId.3/navSetId._/.
32. Tecnologías de Información: Sistemas de Gestión y Planificación. *Tecnologías de Información: Sistemas de Gestión y Planificación*. [En línea] [Citado el: 16 de Noviembre de 2009.] <http://www.tecnologias-informacion.com/aplicacionesweb.html>.
33. Soluciones y propuestas RATIONAL. [En línea] <http://www.rational.com.ar/herramientas/rup.html>.
34. Soluciones Racionales. [En línea] 2008.
35. MSDN Home Page. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28VS.71%29.aspx>.
36. Milenium Home Page. *Milenium Home Page*. [En línea] [Citado el: 3 de Diciembre de 2009.]
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>.
37. MicroMedia2000. [En línea] <http://www.micromedia2000.com/Documentos/doc.pdf>.
38. Macros Excel VBA. *Macros Excel VBA*. [En línea] [Citado el: 5 de Noviembre de 2009.]
<http://www.webandmacros.com/webestatica.htm>.
39. Lenguajes de Programación. *Lenguajes de Programación*. [En línea] [Citado el: 20 de Octubre de 2009.]
<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
40. Departamento de Ingeniería y Sistemas Telemáticos, Universidad Politécnica de Madrid. *Departamento de Ingeniería y Sistemas Telemáticos, Universidad Politécnica de Madrid*. [En línea]
<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s1>.
41. Departamento de Informática de la Universidad de Valladolid. *Departamento de Informática de la Universidad de Valladolid*. [En línea] [Citado el: 3 de Noviembre de 2009.]
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>.
42. Coders. [En línea] <http://www.coders.me/php/codigo-mas-bonito>.
43. Aplicaciones Web. *Aplicaciones Web*. [En línea] [Citado el: 25 de Octubre de 2009.] <http://www.aplicacionesweb.net/>.
44. Abadía Digital. *Abadía Digital*. [En línea] [Citado el: 6 de Diciembre de 2009.] <http://www.abadiadigital.com/articulo/diez-ventajas-del-software-libre-y-propietario/>.