



Universidad de las Ciencias Informáticas

Facultad 6

*Módulo Gestión de Infocintas para la Plataforma
de Televisión Informativa PRIMICIA v2.0*

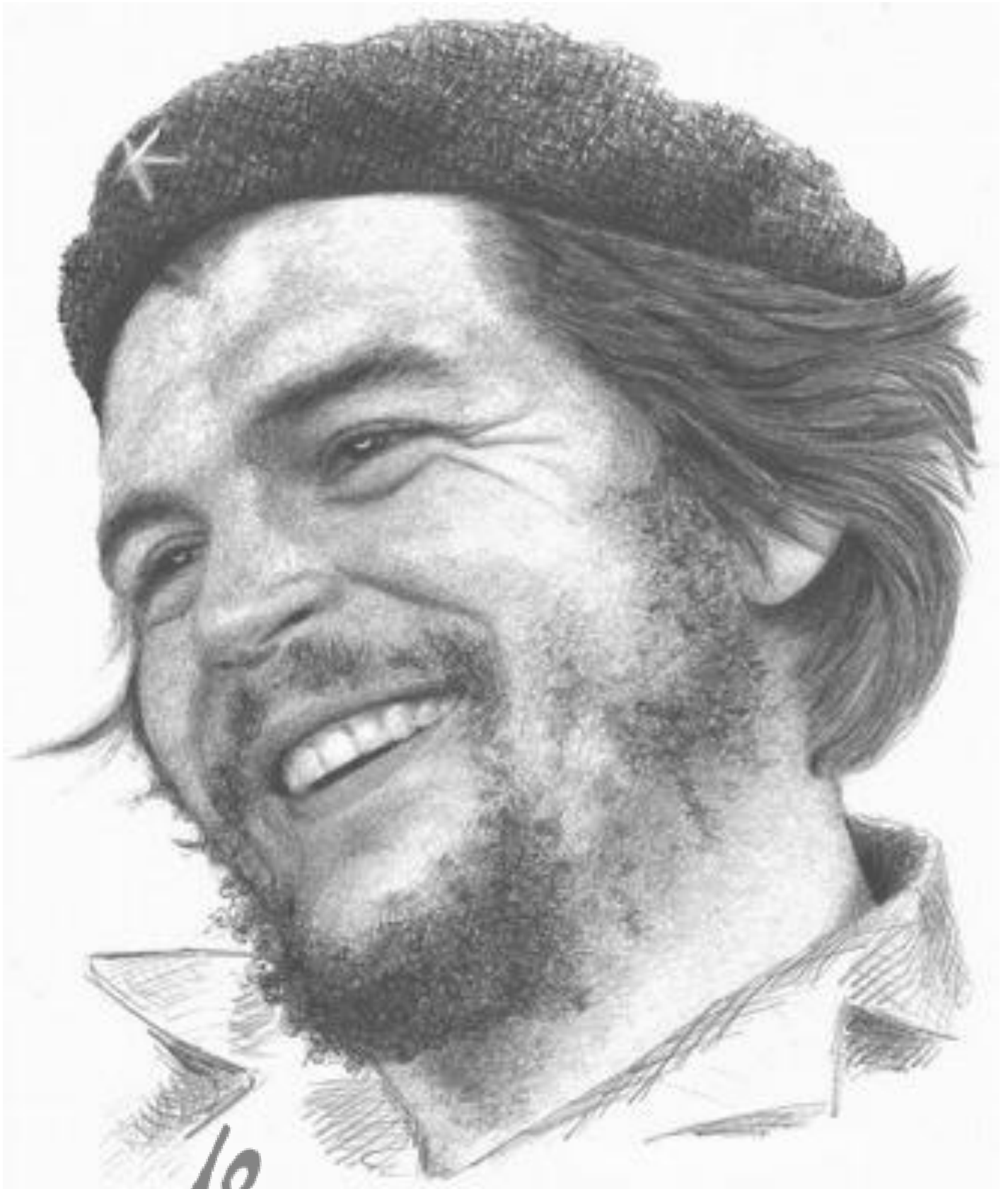
**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

Autor: Dayana Rose Carmona Darias

Tutor: Ing. Carlos de Jesús Andrés González

Ciudad de La Habana, junio 2012

“Año 54 de la Revolución”



o le

"Seamos realistas y hagamos lo imposible."

Dedicatoria

A mi madre por ser la fuerza y la inspiración que me guían...

A mi padre por apoyarme siempre...

A las personas especiales en mi vida...

A ustedes, todos mis triunfos...

Quiero agradecerles de todo corazón

A mis padres, por el amor, la confianza y el apoyo que me han dado toda la vida.

En especial a mi mamá por darme las ganas de seguir adelante, por ser mi inspiración en todo momento, por no dejarme caer nunca y estar siempre ahí para mí.

A mi abuela Edina por preocuparse siempre.

A mi tía Maricel por el apoyo en los primeros años de mi carrera.

A mi tutor por sus enseñanzas, por sus exigencias y por creer en mí.

A las amigas y amigos que han hecho de una forma u otra que estos años sean inolvidables, Gretel, Leyanis, Daimy, Laritza, Gabriel, gracias por estar ahí cuando los necesite y por sobrellevarme en los malos ratos.

A todos mis compañeros de proyecto por sus consejos y apoyo.

A todas esas personas que en su momento marcaron espacios en mi vida y que siempre les agradeceré por su total apoyo y confianza.

Muchas Gracias

Dayana Rose

Declaración de Autoría

Declaro que siendo la única autora del presente Trabajo de Diploma titulado: *Módulo Gestión de Infocintas para la Plataforma de Televisión Informativa PRIMICIA v2.0.*

Autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio teniendo en cuenta los derechos patrimoniales, con carácter exclusivo.

Para que así conste se firma el presente a los ____ días del mes de _____ del año _____.

Dayana Rose Carmona Darias

Autora

Ing. Carlos de Jesús Andrés González

Tutor

Datos de Contacto

Tutor:

Nombre y Apellidos: Carlos de Jesús Andrés González.

Sexo: M F

Institución: Universidad de las Ciencias Informáticas.

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 1/2, Reparto: Torrens,
Municipio: Boyeros, Provincia: Ciudad de La Habana.

Correo electrónico: candres@uci.cu.

Teléfono del trabajo: -

Teléfono particular: -

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2010.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Resumen

En la actualidad, la televisión constituye el medio de comunicación por excelencia. Su trascendencia aumenta día a día a pesar del surgimiento de nuevas tecnologías tales como Internet, que ha demostrado ser un potente impulsor de la comunicación. El hombre en su constante evolución ha vinculado la televisión a otras tecnologías como la informática y las telecomunicaciones, dando lugar al surgimiento y evolución de plataformas de televisión informativa en todo el mundo.

En Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), se desarrolla la Plataforma de Televisión Informativa PRIMICIA, una aplicación capaz de proveer un canal de televisión para la transmisión automática y constante de noticias. Además, se transmiten informaciones suplementarias o adicionales, para lo cual se utiliza un cintillo informativo denominado infocinta.

A partir de numerosos inconvenientes detectados en el proceso de gestión de infocintas de PRIMICIA, producto que se mantiene en constante evolución en cada ciclo de desarrollo, se determinó la ausencia de variedad, dinamismo y atractivo en la creación y transmisión de infocintas, además de la ausencia de funcionalidades que permitan mejorar la configuración de las mismas. En esta investigación se realiza análisis, diseño, implementación y prueba de un nuevo Módulo Gestión de Infocintas, que contribuye a erradicar los problemas existentes, los cuales quedan plasmados en el documento.

Palabras Claves:

Desarrollo, Gestión, Infocinta, Módulo, PRIMICIA.

Índice General

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación Teórica.....	5
1.1. Introducción.....	5
1.2. Conceptos asociados al dominio del problema.....	5
1.3. La gestión de infocintas en PRIMICIA	6
1.3.1. Descripción General.....	6
1.3.2. Situación Problemática	7
1.4. Análisis de otras soluciones existentes.....	8
1.5. Metodología, lenguaje y herramienta para el modelado de la solución	10
1.5.1. RUP: Proceso Unificado de Desarrollo de Software.....	10
1.5.2. UML: Lenguaje Unificado de Modelado.....	11
1.5.3. Visual Paradigm. Herramienta de modelado visual	12
1.6. Lenguaje y framework de programación.....	13
1.6.1. PHP y Symfony	13
1.6.2. JavaScript y JQuery	14
1.7. PostgreSQL. Sistema Gestor de Bases de Datos.....	15
1.8. Apache. Servidor Web.....	16
1.9. Conclusiones.....	17
CAPÍTULO 2: Análisis y Diseño de la Solución Propuesta.....	18
2.1. Introducción.....	18
2.2. Modelo de Dominio.....	18
2.2.1. Conceptos y principales eventos del entorno	18
2.2.2. Diagrama de clases del Modelo de Dominio	19
2.2.3. Conceptos principales del Modelo de Dominio.....	19
2.3. Especificación de los requisitos de software.....	20

2.3.1. Requisitos Funcionales	20
2.3.2. Requisitos No Funcionales.....	22
2.4. Descripción del Sistema Propuesto	23
2.4.1. Descripción de los actores del sistema	23
2.4.2. Casos de Uso del Sistema	24
2.4.3. Especificación de los Casos de Uso	24
2.5. Modelo de Análisis	34
2.5.1. Diagramas de Clases del Análisis	34
2.6. Arquitectura del software	35
2.7. Patrones de diseño.....	36
2.8. Modelo de Diseño.....	38
2.8.1. Diagramas de Clases del Diseño	39
2.9. Conclusiones.....	41
CAPÍTULO 3: Implementación y Validación de la Solución.....	42
3.1. Introducción.....	42
3.2. Modelo de Implementación.....	42
3.2.1. Diagrama de Componentes	42
3.3. Modelo de Despliegue.....	43
3.3.1. Diagrama de Despliegue.....	43
3.4. Pruebas al sistema propuesto	44
3.4.1. Casos de prueba y resultados.....	45
3.5. Conclusiones.....	54
CONCLUSIONES	56
RECOMENDACIONES	57
BIBLIOGRAFÍA REFERENCIADA	58
BIBLIOGRAFÍA CONSULTADA.....	60
GLOSARIO	62

Índice de Figuras

Figura 1: Ciclo de vida de RUP.....	11
Figura 2: Diagrama de clases del Modelo de Dominio.....	19
Figura 3: Diagrama de Casos de Uso del Sistema.....	24
Figura 4: Diagrama de Clases del Análisis del CU Visualizar Infocinta.....	34
Figura 5: Diagrama de Clases del Análisis del CU Gestionar Infocinta.....	35
Figura 6: Esquema simplificado del funcionamiento interno de Symfony2. (27).....	36
Figura 7: Diagrama de Clases del Diseño del CU Gestionar Infocinta.....	39
Figura 8: Diagrama de Clases del Diseño del CU Visualizar Infocinta.....	40
Figura 9: Framework JQuery.....	40
Figura 10: Diagrama de Componentes General.....	43
Figura 11: Diagrama de Despliegue.....	44

Índice de Tablas

Tabla 1: Propuesta de Actores del Sistema.	24
Tabla 2: Descripción del CU Gestionar Infocinta.	25
Tabla 3: Descripción del CU Visualizar Infocinta.	32
Tabla 4: Secciones a probar en el CU Gestionar Infocinta.	45
Tabla 5: Descripción de variables para el CU Gestionar Infocinta.	49
Tabla 6: Matriz de datos para la sección Adicionar infocinta.	49
Tabla 7: Matriz de datos para la sección Editar infocinta.	50
Tabla 8: Matriz de datos para la sección Eliminar infocinta.	52
Tabla 9: Matriz de datos para la sección Ordenar infocinta.	53
Tabla 10: Matriz de datos para la sección Filtrar infocinta.	53
Tabla 11: Secciones a probar en el CU Visualizar Infocinta.	54

INTRODUCCIÓN

El hombre desde sus inicios, ha tenido la necesidad de comunicarse y de mantenerse informado. Con el devenir de los años ha ido superándose y desarrollando nuevas vías y medios que facilitan el acceso a la información; así fue pasando de los pergaminos a la imprenta, a la radio y hacia una nueva era en el desarrollo de los medios de comunicación.

En la actualidad, donde la información ha llegado a ser indispensable, la televisión constituye el medio de comunicación por excelencia. Su trascendencia aumenta día a día a pesar del surgimiento de otras tecnologías tales como Internet, que ha demostrado ser un potente impulsor de la comunicación. El hombre en su constante evolución ha vinculado la televisión a otras tecnologías como la informática y las telecomunicaciones, con el objetivo de hacer llegar las informaciones con más rapidez, calidad y a un mayor número de personas. Esta unión dio lugar a grandes avances en el mundo de las noticias, como son aplicaciones informáticas que permiten el proceso de transmisión y gestión de informaciones textuales, imágenes fijas, sonido y video.

En Cuba, específicamente en el Centro Productivo de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 en la Universidad de las Ciencias Informáticas (UCI), se desarrolla la Plataforma de Televisión Informativa PRIMICIA, una aplicación capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos (texto, imagen, audio y video). Este proyecto tiene como antecedentes tres soluciones informáticas creadas para la transmisión de noticias. La primera es Señal 3, un sistema creado para mantener informada a toda la comunidad universitaria mediante la televisión interna de la UCI. Otra solución similar es la plataforma para la Agencia Cubana de Noticias (ACN), con el nombre de Señal ACN, cuyo objetivo principal es el de informar a los colaboradores cubanos que cumplen misión en el exterior y a los habitantes de las zonas de silencio de la geografía cubana. Por último TV Energía, desarrollada para el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET), con el objetivo de mantener informados a todos sus trabajadores y visitantes.

En PRIMICIA, una noticia se utiliza para mostrar la información referente a una temática determinada, la misma se compone por pantallas donde se combinan texto, imagen, video y audio, permitiendo mostrar una gran cantidad de información. De la misma manera es necesario mostrar información suplementaria o adicional, para lo cual se utiliza un cintillo informativo denominado infocinta.

En estos momentos las infocintas transmitidas en PRIMICIA solo muestran texto, el cual es mostrado siempre en la parte inferior de la pantalla, entrando de derecha a izquierda en una secuencia de

caracteres y sobre un fondo de un color o imagen invariable. Al ser creadas, el usuario especifica el texto que será mostrado y vincula esa información a una sección temática, además de especificar otros parámetros. No es posible configurar las infocintas para hacerlas más atractivas al televidente. Todo lo anterior evidencia falta de variedad, dinamismo y atractivo en la creación y transmisión de infocintas, además de la ausencia de nuevas funcionalidades que permitan mejorar su funcionamiento. A partir de lo anterior se ha determinado el siguiente **problema de la investigación**: ¿Cómo contribuir a que exista mayor funcionalidad y dinamismo en el Módulo Gestión de Infocintas de la Plataforma de Televisión Informativa PRIMICIA para su versión 2.0?

Partiendo del problema planteado se tiene como **objeto de estudio** los procesos relacionados con la gestión de infocintas en la Plataforma de Televisión Informativa PRIMICIA. El **campo de acción** es la gestión de infocintas de la Plataforma de Televisión Informativa PRIMICIA v2.0.

Se propone como **objetivo general**: Desarrollar un Módulo Gestión de Infocintas que permita mayor funcionalidad y dinamismo para la Plataforma de Televisión Informativa PRIMICIA v2.0.

Con la intención de darle solución al objetivo se trazaron las siguientes **tareas de investigación**:

1. Elaboración del marco teórico sobre la gestión de infocintas en la Plataforma de Televisión Informativa PRIMICIA.
2. Caracterización de las herramientas y tecnologías para el desarrollo de la Plataforma de Televisión Informativa PRIMICIA v2.0.
3. Definición de los requisitos funcionales y no funcionales de la nueva versión del sistema a desarrollar.
4. Realización del análisis y diseño de la nueva versión del sistema, haciendo uso de una herramienta de modelado.
5. Implementación del Módulo Gestión de Infocintas de la Plataforma de Televisión Informativa PRIMICIA v2.0.
6. Realización de pruebas de funcionalidad al módulo gestión de infocintas.

Se plantea como guía de la investigación la siguiente **idea a defender**: si se desarrolla un módulo gestión de infocintas para la Plataforma de Televisión Informativa PRIMICIA v2.0, se lograrán aportes que permitan mayor funcionalidad y dinamismo.

Los métodos científicos que se utilizaron en el desarrollo de la investigación son los siguientes:

Métodos Teóricos

Análítico-Sintético: Consiste en realizar un estudio de los fenómenos, examinar y detallar los elementos por separado, simplificando la forma de comprender su funcionamiento. Este método permitió analizar los elementos que influyen en el proceso de la gestión de infocintas, logrando una mejor comprensión del tema investigado.

Histórico-Lógico: Consiste en el estudio de la trayectoria de los fenómenos y acontecimientos en su devenir histórico, investigando las leyes generales de cómo funcionan. En la investigación, se empleó este método para realizar un estudio profundo y minucioso en las versiones 1.0, 1.5 y 1.6 de PRIMICIA, sobre el proceso de gestión de infocintas, permitiendo ver las principales deficiencias y analizar posibles soluciones.

Métodos Empíricos

Observación: Consiste en conocer la realidad mediante la percepción directa de los objetos y fenómenos. Este método permitió obtener información acerca del comportamiento de la gestión de infocintas en la práctica, permitiendo identificar parte de los requisitos funcionales y no funcionales que debe tener el módulo a implementar.

Para una mejor comprensión del trabajo de diploma se resume la estructura del mismo:

Capítulo 1: Fundamentación Teórica.

En este capítulo se realiza un estudio de los elementos teóricos afines a la gestión de infocintas; se describen los conceptos asociados al dominio del problema, se aborda con profundidad el objeto de estudio, la situación problemática y se analizan otras soluciones existentes. Igualmente se caracterizan las herramientas y tecnologías que se utilizarán para la construcción de la solución.

Capítulo 2: Análisis y Diseño de la Solución Propuesta.

En este capítulo se realiza el análisis y diseño de la solución propuesta. Se describe el modelo de dominio, permitiendo un mayor entendimiento de la solución. Se enumeran y detallan los requisitos funcionales y no funcionales que debe tener el sistema partiendo de conocer las condiciones y cualidades deseadas por los usuarios. Se realiza el diagrama de casos de usos del sistema y la

descripción textual. También se realizan los diagramas de clase de análisis y diseño que permitirán un mejor entendimiento de las funcionalidades del módulo.

Capítulo 3: Implementación y Validación de la Solución.

En el este capítulo se realiza el flujo de trabajo de implementación, donde se presentan los elementos fundamentales de la fase de implementación del módulo Gestión de Infocintas, se analizan los principales artefactos como el Modelo de Implementación y el Modelo de Despliegue. Además, se realizan una serie de pruebas, y un análisis de los resultados de las mismas, que permiten la validación de la solución propuesta.

CAPÍTULO 1: Fundamentación Teórica

1.1. Introducción

En este capítulo se proponen como objetivos realizar un estudio de los principales conceptos asociados al tema para que la investigación pueda ser comprendida en su totalidad; analizar el objeto de estudio profundizando desde los aspectos generales hasta los más específicos; definir claramente la situación problemática que dio origen al problema de la investigación y caracterizar las principales herramientas y tecnologías que guiarán el trabajo, así como las que permitirán la implementación del módulo.

1.2. Conceptos asociados al dominio del problema

A continuación, se definen conceptos esenciales para un mejor entendimiento de la investigación:

Módulo

Según el Diccionario de la Real Academia de la Lengua Española (DRAE):

Un módulo es una pieza o un conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica. (1)

En cuestión de programación el diccionario de informática ALEGSA plantea:

Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo. Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión. (2)

Analizando estos conceptos se puede entender que un módulo no es más que una parte de un software que funciona de manera independiente, que puede estar formado por otros módulos o subprogramas y que permite un mejor entendimiento de lo que se desea realizar.

Infocinta

Una infocinta no es un término propio del español ni de otro idioma, más bien es un término técnico, utilizado en PRIMICIA, por el cual se conoce ese elemento en la televisión, también conocido como cintillo informativo o marquesina. Técnicamente viene siendo el área de la pantalla dedicada a ofrecer

información suplementaria o adicional, también utilizada para publicidad o promociones. Dentro de una infocinta se pueden mezclar diferentes elementos como imágenes, segmentos de videos o textos, que se utilizan para que la información sea más interesante al televidente.

1.3. La gestión de infocintas en PRIMICIA

La Plataforma de Televisión Informativa PRIMICIA es una aplicación desarrollada con el uso de herramientas y de software libre que tiene como meta proporcionar un canal de transmisión automática y constante de informaciones. Como plataforma, PRIMICIA está implementada para que pueda ser aplicable a distintos clientes, por lo que constituye un producto informático del cual pueden beneficiarse televisoras, agencias de noticias, empresas, terminales de transporte, hoteles y cualquier otra entidad que tenga una red de televisión con necesidades de transmitir informaciones en distintos formatos como texto, imagen, texto-imagen y video. A continuación se describen las funcionalidades de la plataforma y el funcionamiento actual de la gestión de infocintas.

1.3.1. Descripción General

PRIMICIA se encuentra estructurada en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final a través de una red de televisión. El Subsistema de Administración es el responsable de que a través de él se realice la administración del canal y toda la gestión de las noticias y recursos multimedia. El Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados. (3)

Particularmente el Subsistema de Administración se encarga de gestionar los usuarios del sistema, las secciones temáticas, las infocintas, la redacción de noticias, el almacenamiento y reproducción de los recursos multimedia, también de los cambios entre las señales de canales televisivos. El Subsistema de Transmisión tiene como prestaciones generar una cartelera, transmitir las noticias y las infocintas, reproducir un fondo musical y transmitir las señales de canales televisivos en vivo.

El proceso actual de la gestión de infocintas se desarrolla en el Subsistema de Administración. La gestión de infocintas es un proceso que brinda pocas opciones de configuración y cuenta con funcionalidades que pueden ser refinadas. En el Subsistema de Transmisión, al ser transmitida una infocinta, el televidente observa un texto sobre un fondo de un color invariable que realmente no atrae la atención necesaria. A continuación se aborda con más profundidad el funcionamiento actual de la gestión de infocintas.

1.3.2. Situación Problemática

El módulo gestión de infocintas cuenta con varias funcionalidades como listar las infocintas existentes, permitiendo mostrar de cada infocinta el texto por el que están formadas, la sección temática a la que pertenecen, la hora de inicio y fin que especifica el tiempo que estarán siendo transmitidas, el orden en que serán mostradas, el cual puede ser modificado en la lista, además de las opciones de editar y eliminar una infocinta. También cuenta con la opción filtrar que permite realizar una búsqueda basada en varios parámetros, y un adicionar que permite la creación de las infocintas.

Para crear una infocinta se necesita obligatoriamente definir a que sección temática pertenece, una hora de inicio y fin que permite especificar el tiempo que estará vigente en el sistema, y el contenido, que en estos momentos es sólo texto. Se especifica de forma opcional si está activa o si es inmediata. Una infocinta creada como activa es transmitida, en caso contrario está almacenada en el sistema, pero no se muestra al televidente. Una infocinta creada como inmediata es transmitida en el momento que se crea, de lo contrario se transmite en el orden que tenga especificado.

En el Subsistema de Transmisión durante cada ciclo noticioso se identifican las infocintas activas y a medida que se trasmite una sección temática de noticias, se van mostrando las infocintas relacionadas con la misma. El televidente, durante la transmisión de una infocinta, ve en la parte inferior de la pantalla una cadena de texto. Esta se muestra de derecha a izquierda sobre un fondo rectangular de color azul. Para el caso específico de las infocintas inmediatas el color de fondo cambia, permitiendo al usuario ver la diferencia entre una y otra.

Para dar solución al problema se necesita que la gestión de infocintas sea mucho más atractiva, dinámica y que cuente además con otras funcionalidades. Para lograr esto las infocintas deben mostrar además de textos, otros elementos como imágenes o segmentos de videos. El contenido de las infocintas debe tener efectos de entrada y salida, y aparecer sobre un fondo con una imagen o color definido por el usuario que crea la información. La infocinta debe poder ser mostrada en diferentes lugares de la pantalla dependiendo de lo que se publique en ese momento, debe contar con un tamaño variable y una alineación que permita mostrarla a la izquierda, la derecha o en el centro de la pantalla. Las infocintas pueden contener información referente a cualquier tema, por lo que se considera que no deben estar vinculadas a una sección temática, teniendo en cuenta este aspectos, se recomienda al Subsistema de Transmisión que modifique el proceso de visualización actual de las infocintas ya que depende de las secciones temáticas. Las infocintas deben conservar la opción de inmediatas, puesto que existen informaciones de urgencia que no pueden esperar para ser mostradas

al televidente. Igualmente la opción de activa debe mantenerse permitiendo que existan infocintas almacenadas que no necesariamente deben ser transmitidas.

1.4. Análisis de otras soluciones existentes

En la actualidad existen un gran número de productos de software encaminados a la transmisión de información que facilitan el trabajo de periodistas y redactores, y permiten una mejor divulgación de las noticias. Estos sistemas gestionan infocintas, pero no se conoce con claridad como realizan el proceso. Los sistemas seleccionados serán analizados desde el punto de vista del televidente, o sea de lo que se puede observar al transmitir las informaciones.

Canal Cultural

Canal de televisión dirigido por la Dirección de Televisión Universitaria (DTU) de la UCI. Ofrece una programación balanceada encaminada a cultivar el gusto ético y estético, contribuyendo a su vez a la creación de paradigmas y modelos de Comunidad Universitaria. (4)

En el Canal Cultural se transmiten infocintas mediante hardware. Cuando se está transmitiendo una señal, está transmitiéndose por detrás un video con fondo negro, el cual tiene en la parte inferior una franja de un color diferente con un texto pasando de izquierda a derecha, que representa la infocinta. Cuando quiere mostrarse una infocinta, el trabajador encargado, mediante un switch con una palanca, mueve la señal principal hacia arriba, permitiendo que se vea el video que corre por detrás, o sea la infocinta. Cuando no quieren mostrarse más infocintas, se retorna la señal principal a su posición inicial. El video donde se crea la infocinta puede contener imágenes, además de texto.

Esta solución no se puede aplicar en PRIMICIA, ya que esta trabaja de forma automática, sin que exista un control directo de lo que se transmite, este tipo de solución a través de hardware requiere de una persona encargada de mostrar las infocintas. Además se deben tener conocimientos básicos de algún editor de video para crear los videos que se transmiten, lo que tomaría un tiempo de capacitación a los usuarios e iría en contra del principio de funcionamiento de la plataforma.

Canal Multivisión

El canal de televisión de más reciente creación en Cuba, el cual transmite desde La Habana, las 24 horas del día. La programación es casi en su totalidad de procedencia extranjera, previo acuerdo con compañías como Discovery Channel, Disney Channel, Telesur, entre otras. (5)

En este canal las infocintas también se transmiten mediante hardware, aunque algunas vienen grabadas en los programas que se transmiten de otras televisoras. Se muestran infocintas de varias formas, algunas están formadas por un cintillo de texto en la parte inferior o superior de la pantalla, el cual va encabezado o finaliza con una imagen que representa lo que se anuncia o el logo del canal en caso de ser una información relacionada con éste. Tienen fondos transparentes o con efectos de degradado, se muestran a lo largo de la pantalla o solo en una mitad. Se observan otras infocintas más sencillas, formadas solo por texto, el cual aparece desde un lado de la pantalla, con un efecto de desplazamiento, si la información es mayor al ancho de la pantalla, las letras desaparecen para dar lugar a las que siguen. Estas infocintas también pueden tener un fondo que permita diferenciarlas de lo que se está transmitiendo y otro texto en la parte superior, en forma de título. La mayoría de las infocintas mostradas cuentan con variados efectos de entrada o salida.

TeleSUR

TeleSUR, la nueva televisora del Sur, es una cadena de televisión latinoamericana que transmite en señal abierta y por satélite con sede en Caracas, Venezuela. TeleSUR fue creada con la misión de ofrecer Información para promover la integración de América Latina y como contrapeso a las grandes Cadenas internacionales de noticias como la CNN, Univisión y la BBC. (6)

Este canal muestra un tipo de infocintas diferente, ya que incluyen animaciones o pequeños videos dentro de las infocintas, dando lugar a una nueva variante para motivar a los televidentes.

En programas específicos, como los noticieros, muestran infocintas que permanecen estáticas en la parte inferior de la pantalla, las cuales van mostrando contenido descargado de servidores externos al canal, como por ejemplo resultados de juegos, cifras numéricas sobre temas económicos o el estado del tiempo. Estas infocintas cuentan con animaciones de entrada y salida de los textos, además de imágenes o videos que identifican el tema que están tratando. La mayoría de las infocintas contienen en la parte derecha un video del logo de Telesur, representando la televisora. También tiene efectos de brillo o iluminación cuando un contenido es relevante o para enfatizar algún tema específico.

Luego de analizar las características de las soluciones anteriores, se identificaron una serie de elementos en la visualización de las infocintas transmitidas en los canales Multivisión y Telesur que pueden incluirse en el módulo gestión de infocinta, como son la integración de diferentes elementos como textos, imágenes y videos, el uso de fondos y la aplicación de efectos que aportarían nuevas funcionalidades permitiendo un mayor dinamismo en las infocintas. A partir de este análisis es evidente

la necesidad de mejorar el módulo gestión de infocintas para PRIMICIA v2.0, con el fin de incorporar nuevas funcionalidades a las infocintas transmitidas actualmente.

1.5. Metodología, lenguaje y herramienta para el modelado de la solución

A continuación se describen las principales características de la metodología, el lenguaje y la herramienta definida en la arquitectura del sistema que se utilizarán para definir y consolidar el análisis y diseño correspondiente al problema.

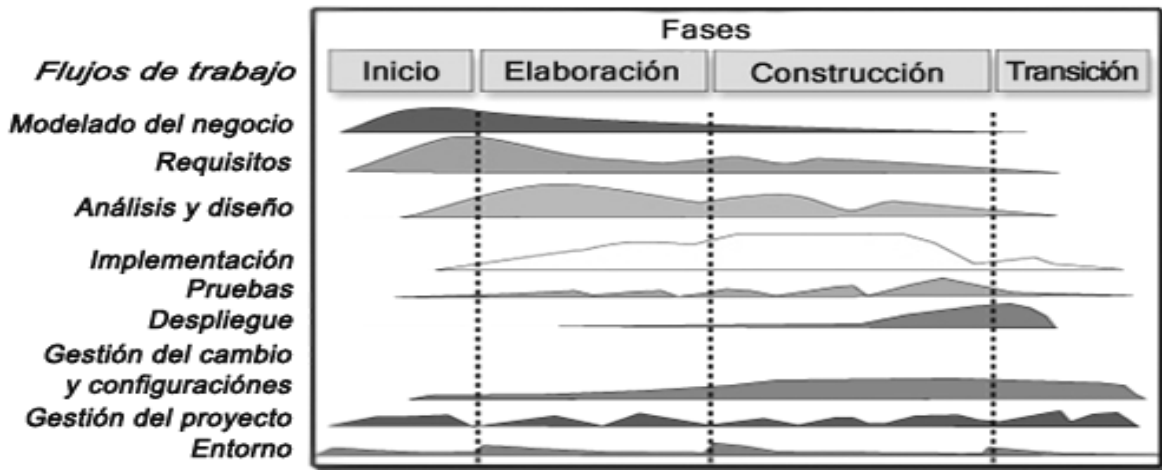
1.5.1. RUP: Proceso Unificado de Desarrollo de Software

La Metodología de desarrollo de software está dada por un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. (7) El Proceso Unificado de Desarrollo de Software (RUP), es una metodología robusta que cubre el ciclo de vida y desarrollo de software. Es un proceso planteado por Kruchten en 1996, con el objetivo producir software de alta calidad, es decir, que cumpla con los requisitos de los usuarios dentro del tiempo y presupuesto establecido.

RUP es un proceso de desarrollo de software, sin embargo, es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. (7)

RUP tiene como características principales que está guiado por casos de uso, centrado en la arquitectura, siendo iterativo e incremental. Basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para representar los esquemas de un sistema de software y describir todo el proceso. La metodología presenta 4 etapas muy importantes: Inicio, Elaboración, Construcción y Transición, donde cada una de ellas está compuesta por una o varias iteraciones. Además de las fases, contiene flujos de trabajos, los mismos se dividen en Flujos de trabajo de desarrollo y Flujos de trabajo de soporte. En la **Figura 1** se muestra el ciclo de vida de esta metodología.

Figura 1: Ciclo de vida de RUP.



RUP es la metodología que mejor se adapta a los proyectos a largo plazo como PRIMICIA, además permite utilizar Visual Paradigm, una herramienta multiplataforma que brinda excelentes resultados en el modelado, que ha sido utilizada en la modelación de los artefactos del proyecto. RUP genera una amplia documentación, bien detallada en cuanto a cómo la tarea será ejecutada y los resultados que se deben obtener.

1.5.2. UML: Lenguaje Unificado de Modelado

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. (8)

RUP utiliza el UML para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos. (7)

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (8)

Permite especificar las características que tendrá el sistema, incluso antes de su construcción. A partir de los modelos que se vayan generando se puede construir el sistema diseñado y estos propios

elementos gráficos sirven como documentación del sistema desarrollado para futuras revisiones y versiones. Es un lenguaje libre y totalmente independiente del lenguaje de programación que se utilice. Tiene como fin modelar sistemas en base al paradigma orientado a objetos, por tanto los diseños realizados usando UML se pueden implementar en cualquier lenguaje orientado a objetos.

1.5.3. Visual Paradigm. Herramienta de modelado visual

Las herramientas de modelado visual también son conocidas como herramientas CASE, como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. (9)

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, mediante la realización del diseño del proyecto, el cálculo de los costos, la implementación de parte del código automáticamente, la documentación y detección de errores.

Visual Paradigm 8.0 Enterprise Edition es una herramienta profesional para el modelado con UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. (10)

Ha sido diseñada para un gran número de usuarios, incluyendo ingenieros de sistemas, analistas de sistemas, analistas de negocio y arquitectos de sistemas. Permite representar todos los tipos de diagramas de clases, código inverso, generar documentación y código desde diagramas. Visual Paradigm es una herramienta multiplataforma pues posee la capacidad de ejecutarse sobre diferentes sistemas operativos. Es muy fácil de usar pues presenta un ambiente gráfico agradable e intuitivo para el usuario. Es muy poderosa pues permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requerimientos, de base de datos y la generación de código PHP.

Admite la integración con varios entornos de desarrollo (IDE). Es una herramienta colaborativa pues soporta múltiples usuarios trabajando sobre el mismo proyecto y a la vez permite realizar el control de versiones, característica que tiene un impacto positivo en proyectos que cuentan con varios entregables como PRIMICIA. El uso de esta herramienta también implica un ahorro de tiempo en capacitación, pues ha sido la herramienta CASE utilizada por el equipo de trabajo en el desarrollo de las versiones anteriores de la plataforma.

1.6. Lenguaje y framework de programación

Para poder interpretar las instrucciones que se le dan a una computadora es necesario un lenguaje propio que permita controlar el comportamiento de la comunicación del hombre con la máquina en general, para ello se utilizan los lenguajes de programación.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. (11)

Después de seleccionar un lenguaje de programación, se hace importante seleccionar un framework que proporcione una estructura al código fuente. El mismo debe contar con varias funcionalidades que hagan que el proceso de implementación sea rápido, flexible, eficiente y que la solución que se obtenga sea compatible con lo establecido en el proyecto.

A continuación se describen las principales características de los lenguajes de programación y los frameworks correspondientes, que serán utilizados en la implementación de la solución.

1.6.1. PHP y Symfony

PHP, acrónimo de PHP: Hypertext Preprocessor, es un lenguaje Open Source interpretado de alto nivel, especialmente pensado para el desarrollo web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (12)

Este es un lenguaje de programación interpretado que se ejecuta por medio de un intérprete. Fue creado con el objetivo de construir páginas web dinámicas en las que exista procesamiento del lado del servidor. PHP puede ser interpretado por la mayoría de las aplicaciones servidores web existentes, entre ellos Apache, utilizado por PRIMICIA. PHP presenta un soporte para la mayoría de los sistemas gestores de bases de datos como, PostgreSQL, característica muy importante para el desarrollo de aplicaciones web que gestionan bases de datos. Se puede ejecutar en la mayoría de los sistemas operativos, otra característica favorable, debido a que PRIMICIA es un proyecto multiplataforma.

El Subsistema de Administración de la Plataforma de Televisión Informativa PRIMICIA, está desarrollado sobre el framework Symfony. Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web desarrolladas en PHP.

Fue concebido originalmente por Fabien Potencier de la empresa Sensio Labs para el desarrollo de aplicaciones web de sus propios clientes y dado a conocer en el 2005. En la actualidad es uno de los principales marcos para el desarrollo con PHP. Con el soporte de Sensio Labs y de una gran comunidad, Symfony cuenta con importantes recursos como son una abundante documentación y el apoyo de la comunidad de desarrollo. (13)

Symfony se diseñó para facilitar el trabajo en las aplicaciones web y tiene como principales características que es fácil de instalar y configurar en la mayoría de plataformas, independiente del sistema gestor de bases de datos, basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional, lo que permite ahorrar tiempo de desarrollo. También sigue la mayoría de las mejores prácticas y patrones de diseño para la web, su código es fácil de leer y permite un mantenimiento muy sencillo, lo que en un futuro será factible para otros desarrolladores que trabajen con el módulo. (14)

Es un framework fácil de extender, lo que permite su integración con librerías desarrolladas por terceros, como JQuery, la cual será utilizada en desarrollo del módulo. La última versión conocida es Symfony 2, cuenta con las características anteriores y además incluye mejoras como son la rapidez y flexibilidad, una nueva estructura de directorios de los proyectos que organiza mejor el código, llamada bundels, los archivos de configuración se pueden escribir en PHP, XML, YAML o INI, a conveniencia del programador y permite el autocompletado en los IDE, como el caso del NetBeans.

Symfony está basado en el patrón arquitectónico Modelo Vista Controlador (MVC), donde el modelo se encarga de representar la información, la vista de generar las páginas con las que interactúa el usuario y el controlador se encarga de interactuar entre el modelo y la vista.

Por las características presentadas, además de la experiencia obtenida en iteraciones anteriores del desarrollo del producto PRIMICIA, se empleará PHP 5.3 como lenguaje de programación y su integración con el framework Symfony 2.0 para la implementación del módulo gestión de infocintas.

1.6.2. JavaScript y JQuery

JavaScript es un lenguaje de programación que puede usarse para añadir interactividad a las páginas web. (15) Es un lenguaje interpretado, o sea que no requiere compilación, el navegador se encarga de interpretar las secuencias de código y ejecutarlas adecuadamente.

Es un lenguaje script multiplataforma orientado a eventos, lo que permite amplias oportunidades de respuesta ante las acciones de los usuarios. Es orientado a objetos de forma reducida, pero incluye los

elementos necesarios para que los scripts puedan tener acceso a la información mostrada y actuar sobre la interfaz del navegador. JavaScript se utiliza principalmente para crear páginas web dinámicas, o sea que incorporen efectos visuales y animaciones permitiendo la interacción con el usuario. (15)

Para facilitar el trabajo con JavaScript surge JQuery, un eficiente y ligero framework JavaScript que facilita el desarrollo de aplicaciones web interactivas. JQuery ofrece una excelente API que facilita el uso de las funciones del DOM así como otras no soportadas por éste, manipulación de CCS, incluye funcionalidades para el trabajo con AJAX evitando la carga del servidor, manipulación de eventos, animaciones, diseño y efectos del contenido web. Es de fácil aprendizaje y posee una excelente documentación en su sitio oficial. (16)

Algunas de sus características principales es que cuenta con una amplia variedad de plugins desarrollados sobre la base de la propia librería. JQuery permite obtener efectos sorprendentes y facilidades en un sitio web y además se complementa con interfaces de usuario prediseñadas (JQueryUI), lo que permite aumentar la familiarización del usuario con la aplicación web. Es completamente funcional en los navegadores web como Firefox, Internet Explorer, Opera y Chrome. Emplea selectores CSS para referirse a uno o a un grupo de elementos sobre los que actúa. (16)

JavaScript ha sido utilizado en versiones anteriores de PRIMICIA por lo que se tiene experiencia trabajando con él, además al ser un lenguaje interpretado, facilita la generación de páginas dinámicas. A través de JQuery las sintaxis de código son muy fáciles de aprender y sencillas de utilizar, además permite el intercambio de datos entre cliente y servidor de manera viable, ventajas que favorecen la utilización de JavaScript 1.2 y JQuery 1.7 para la implementación del módulo gestión de infocintas.

1.7. PostgreSQL. Sistema Gestor de Bases de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor. (17)

PostgreSQL 9.0 es un servidor de base de datos relacional de software libre. Es una alternativa de otros sistemas de bases de datos de código abierto, como MySQL, Firebird y MaxDB, así como de sistemas como Oracle o DB2. Es ampliamente considerado como una de las alternativas más avanzadas de SGBD de código abierto pues utiliza un modelo cliente/servidor y usa multiprocesos en

vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (18)

Es un sistema objeto-relacional, incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. PostgreSQL está disponible bajo una Licencia BSD que permite tanto el uso comercial como el no comercial. El desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores con la que se puede contar para obtener soporte ante cualquier problema presentado con el uso del SGBD.

Como principales características de este gestor se encuentran: que corre en casi todos los sistemas operativos: Linux, Windows, entre otros; cuenta con una documentación muy bien organizada, pública, libre y con comentarios de los propios usuarios. También presenta soporte nativo para los lenguajes de programación más populares del medio: PHP, C, C++, Perl y Python, así como soporte de protocolo de comunicación encriptado por SSL, además cuenta con numerosos tipos de datos y con la posibilidad de definir nuevos tipos.

Por las características anteriores, y por ser el gestor definido en la arquitectura del proyecto, se considera que PostgreSQL es el SGBD idóneo para dar soporte a la persistencia de los datos del módulo a desarrollar.

1.8. Apache. Servidor Web

Los Servidores Web o servidores HTTP son programas informáticos que permiten a los clientes compartir datos, documentos y multimedia en formato Web.

El Servidor Web Apache 2.0 permite interpretar PHP, el lenguaje de programación en el que se va a implementar el módulo, es un servidor flexible, rápido y eficiente. Está desarrollado bajo licencia BSD permitiendo modificar el código siempre que se reconozca el trabajo de la comunidad. (19)

Es el servidor web más utilizado en Internet por las características que presenta. Corre en la mayoría de los Sistemas Operativos. Apache es un servidor altamente configurable de diseño modular. Trabaja con Perl, PHP y con lenguajes de script, teniendo todo el soporte que se necesita para tener páginas dinámicas. Apache permite personalizar la respuesta ante los posibles errores que puedan aparecer en el servidor. Este servidor web también ha sido utilizado en las versiones anteriores de PRIMICIA por lo que se selecciona como el servidor web a utilizar.

1.9. Conclusiones

En el capítulo se realizó un estudio de los principales conceptos asociados al tema, logrando que la investigación sea más fácil de comprender. Se caracterizó el objeto de estudio a través del cual se pudieron definir mejor las necesidades de la investigación. Se especificó la situación problemática profundizando en los aspectos de mayor importancia. Se analizó la visualización de infocintas en otros canales de televisión concluyendo que se hace necesario mejorar el módulo gestión de infocinta, aplicando las nuevas características observadas.

Se hizo en el desarrollo del capítulo una caracterización de las principales herramientas y tecnologías para la implementación del módulo, definiendo como metodología de desarrollo RUP, UML como lenguaje de modelado visual con Visual Paradigm 8.0 como herramienta. Se identificaron como lenguajes de programación, PHP 5.3 con el framework Symfony 2.0 y JavaScript 1.2 con JQuery 1.7. También se define como servidor web a utilizar, Apache 2.0 y como sistema gestor de base de datos PostgreSQL 9.0.

CAPÍTULO 2: Análisis y Diseño de la Solución Propuesta

2.1. Introducción

En el presente capítulo se realiza el análisis y diseño de la solución propuesta. Se describe el modelo de dominio, permitiendo un mejor entendimiento de la solución. Se enumeran los requisitos funcionales y no funcionales que debe tener el sistema partiendo de conocer las condiciones y cualidades deseadas por los usuarios. Luego se realiza el diagrama de casos de usos del sistema y la descripción detallada de cada caso de uso. También se realizan los diagramas de clase de análisis y diseño que permitirán un mejor entendimiento de las funcionalidades del módulo.

2.2. Modelo de Dominio

El Modelo de Dominio o Modelo Conceptual permite representar los conceptos más importantes de los objetos que se muestran en el dominio del problema. Su objetivo es lograr la representación de las clases conceptuales del mundo real, significativas en un dominio del problema, no de componentes de software, aunque algunos objetos del modelo pueden terminar siéndolo. (20)

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. (7)

RUP como metodología de desarrollo de software propone realizar un Modelo de Dominio para cuando los procesos de negocio no están claramente definidos. Las nuevas funcionalidades del Módulo Gestión de Infocintas hacen que el proceso sea muy variable. Cada usuario puede confeccionar las infocintas de la forma que estime conveniente, por esto no existe un proceso de negocio visible y bien definido para estas funcionalidades. Al no tener bien definidos los procesos del negocio se realiza el Modelo de Dominio y se explica cada uno de los conceptos que forman parte del mismo, además se expone un análisis detallado de los eventos más significativos del módulo.

2.2.1. Conceptos y principales eventos del entorno

Cuando un **usuario** con los permisos necesarios para realizar cualquiera de las funcionalidades presentes en el módulo, accede al Subsistema de **Administración** y al módulo en sí, puede **gestionar** las **infocintas**. El usuario puede editar, eliminar, ordenar, buscar dentro de las infocintas existentes o adicionar una nueva. Una infocinta está formada por un **fondo** y un **contenido**, este último puede ser una imagen, un texto, un segmento de video o la combinación de todos ellos, además el contenido

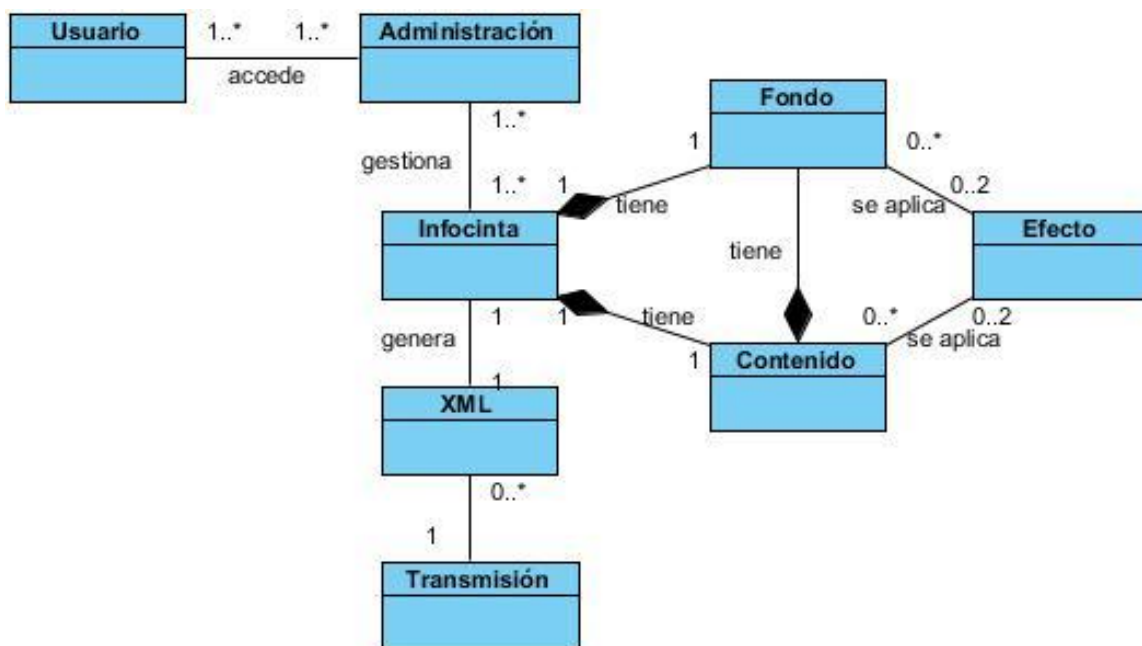
también cuenta con un fondo. Tanto al contenido como al fondo de la infocinta se le pueden aplicar **efectos** de entrada y salida que permiten lograr mayor dinamismo y atractivo.

Después que una infocinta es creada o actualizada, genera un fichero **XML**, que almacena todas las configuraciones que el usuario especificó en la infocinta. El Subsistema de **Transmisión** a la hora de publicar las infocintas, busca los XML que le permiten obtener todas las informaciones necesarias para la visualización de las mismas, manteniendo todas las características con que fueron creadas.

2.2.2. Diagrama de clases del Modelo de Dominio

En la Figura 2 se muestran los principales conceptos asociados al dominio del problema y las relaciones que se establecen entre éstos.

Figura 2: Diagrama de clases del Modelo de Dominio.



2.2.3. Conceptos principales del Modelo de Dominio

Usuario: Persona con los permisos necesarios para acceder al módulo y realizar cualquiera de las funcionalidades presentes en el mismo.

Administración: Se refiere al Subsistema de Administración en el cual se encuentra el módulo gestión de infocintas.

Infocinta: Se refiere a las infocintas que se gestionan en el módulo.

Contenido: Es la parte que va a ser mostrada en una infocinta, puede ser texto, imagen, video o una relación entre estos.

Fondo: Puede ser una imagen un color definido por el usuario para fondo de la infocinta o del contenido.

Efecto: Son los efectos que se aplican al fondo de la infocinta y al contenido para lograr más dinamismo y atractivo en las infocintas.

XML: Es un metalenguaje extensible de etiquetas que se propone como un estándar para el intercambio de información estructurada entre los dos Subsistemas de PRIMICIA.

Transmisión: Se refiere al Subsistema de Transmisión donde van a ser transmitidas las infocintas.

2.3. Especificación de los requisitos de software

Para llevar a cabo el desarrollo de un software, es necesario saber qué debe hacer dicho sistema y cómo debe hacerlo, tener bien claro cuáles son las expectativas del cliente con respecto al sistema que se va a desarrollar. Según la IEEE un requisito es una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo que debe cumplirse o que posea un sistema o componente del sistema para satisfacer un contrato, norma, especificación u otro documento formalmente impuesto. (21)

2.3.1. Requisitos Funcionales

Los requisitos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema, éstas se corresponden con operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema. (22)

Para el módulo a desarrollar se definen los siguientes requisitos funcionales (RF):

- RF 1.** Listar infocintas: El sistema debe permitir que se muestre un listado de todas las infocintas que estén creadas en la plataforma.
- RF 2.** Crear infocinta: El sistema debe permitir que se creen las infocintas.
 - RF 2.1.** Crear el contenido de la infocinta: El sistema debe permitir que se establezca el contenido de la infocinta (texto, imagen, video o la combinación de estos).

- RF 2.2.** Seleccionar efecto del contenido: El sistema debe permitir que se establezcan los efectos de entrada y salida al contenido de la infocinta.
- RF 2.3.** Seleccionar el fondo: El sistema debe permitir que se establezca un fondo a la infocinta y al contenido de la infocinta (imagen o color).
- RF 2.4.** Seleccionar efecto del fondo de la infocinta: El sistema debe permitir que se establezcan los efectos de entrada y salida al fondo de la infocinta.
- RF 2.5.** Seleccionar la posición de la infocinta: El sistema debe permitir que se establezca la posición de la infocinta en la pantalla (arriba o abajo).
- RF 2.6.** Seleccionar el tamaño de la infocinta: El sistema debe permitir que se le establezca un tamaño a la infocinta.
- RF 2.7.** Seleccionar la alineación de la infocinta: El sistema debe permitir que se establezca la alineación de la infocinta (centro, izquierda o derecha).
- RF 3.** Editar infocinta: El sistema debe permitir que se modifiquen los datos de la infocinta.
 - RF 3.1.** Editar el contenido de la infocinta: El sistema debe permitir que se modifique el contenido de la infocinta (texto, imagen, video o la combinación de estos).
 - RF 3.2.** Editar efecto del contenido: El sistema debe permitir que se modifiquen los efectos de entrada y salida al contenido de la infocinta.
 - RF 3.3.** Editar el fondo: El sistema debe permitir que se modifique el fondo de la infocinta y del contenido.
 - RF 3.4.** Editar efecto del fondo: El sistema debe permitir que se modifiquen los efectos de entrada y salida al fondo de la infocinta.
 - RF 3.5.** Editar la posición de la infocinta: El sistema debe permitir que se modifique la posición de la infocinta en la pantalla (arriba o abajo).
 - RF 3.6.** Editar el tamaño de la infocinta: El sistema debe permitir que se modifique el tamaño de la infocinta.
 - RF 3.7.** Editar la alineación de la infocinta: El sistema debe permitir que se modifique la alineación de la infocinta (centro, izquierda o derecha).
- RF 4.** Ajustar Propiedades: El sistema debe permitir que se modifiquen las propiedades de los componentes de la infocinta (alto, ancho, posición, profundidad, opacidad, entrada y tiempo de vida).
- RF 5.** Eliminar infocinta: El sistema debe permitir que se eliminen las infocintas existentes.
- RF 6.** Visualizar infocinta: El sistema debe mostrar cómo se verá al ser transmitida una infocinta de las listadas en el sistema.

RF 7. Ordenar infocintas: El sistema debe permitir que se modifique el orden en que las infocintas serán visualizadas.

RF 8. Filtrar infocinta: El sistema debe facilitar hacer un filtrado de las infocintas dados ciertos criterios de búsqueda (nombre, activa, inmediata, comienza y termina).

RF 9. Generar XML: El sistema debe permitir que se cree un XML de la infocinta, permitiendo almacenar todos los datos de presentación de la misma.

2.3.2. Requisitos No Funcionales

Los requerimientos o requisitos no funcionales son propiedades o cualidades que el producto debe tener, posibilitando que el mismo sea más atractivo, usable, confiable, entre otras. Normalmente están vinculados a RF, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. También puede ser alguna restricción que el sistema debe tener para cumplir una determinada funcionalidad.

Requisitos de apariencia o interfaz externa

La apariencia del módulo debe estar limitada por las mismas condiciones que PRIMICIA, haciendo uso de colores apropiados que denoten un producto profesional. La información debe presentarse de forma clara y organizada, permitiendo una correcta interpretación de los usuarios.

Requisitos de usabilidad

El sistema debe poseer funcionalidades consistentes, debe mostrar mensajes al usuario, que le ayuden a llevar a cabo la tarea que realiza. Se debe hacer uso de botones, con imágenes que indiquen de modo intuitivo la función que realizan, así mismo los controles visuales deben mostrar mensajes aclarando sus propósitos.

Requisitos de portabilidad

El sistema estará realizado para funcionar en los sistemas operativos GNU/Linux o Windows, garantizando de esta manera un producto multiplataforma.

Requisitos de software

Se debe utilizar Apache 2.0 como servidor web y PostgreSQL 9.0 o superior como sistema gestor de base de datos. Se deberá acceder al módulo a través de los navegadores más populares, como

Mozilla Firefox 8.0 o superior, Chrome 16.0 o superior y cualquier versión de Internet Explorer igual o superior a la 8.0.

Requisitos de hardware

El servidor de Administración debe tener un procesador Pentium IV a 2.8 GHz o superior, con una memoria RAM de 1Gb como mínimo, un disco duro de 80Gb o mayor y una tarjeta de red Ethernet 10/100 Mbps o superior.

Restricciones en el diseño y la implementación

Para la modelación del sistema se utilizará UML, a través de la herramienta Visual Paradigm. Se requiere el uso de la arquitectura MVC. Se deben usar como lenguajes de programación PHP 5.3 y JavaScript 1.2, y como framework de presentación JQuery 1.7.

2.4. Descripción del Sistema Propuesto

Después de haber realizado el levantamiento de requisitos del sistema, se propone el desarrollo del nuevo módulo Gestión de Infocintas para PRIMICIA, que permita solucionar los problemas existentes en el módulo actual, aportando un mayor dinamismo, atractivo y nuevas funcionalidades al sistema. Se han identificado las principales funcionalidades que ofrecerá el módulo, permitiendo añadir un fondo a las infocintas y al contenido de las mismas formado por imágenes o video además del texto, así como la aplicación de efectos de entrada y salida al fondo de la infocinta y al contenido. Brindará además, la opción de previsualizar las infocintas, dando al usuario una idea de cómo se verán al ser transmitidas. Contendrá una interfaz agradable y de fácil entendimiento para los usuarios.

2.4.1. Descripción de los actores del sistema

Los actores representan a los seres humanos, a un software, hardware externo u otros sujetos que interactúen con el sistema que se esté especificando. (23)

Un actor del sistema no es parte del sistema en desarrollo, es un agente externo que intercambia información con el mismo en pos de obtener un resultado esperado. Durante el desarrollo del módulo se definió un único actor, él mismo se describe a continuación.

Tabla 1: Propuesta de Actores del Sistema.

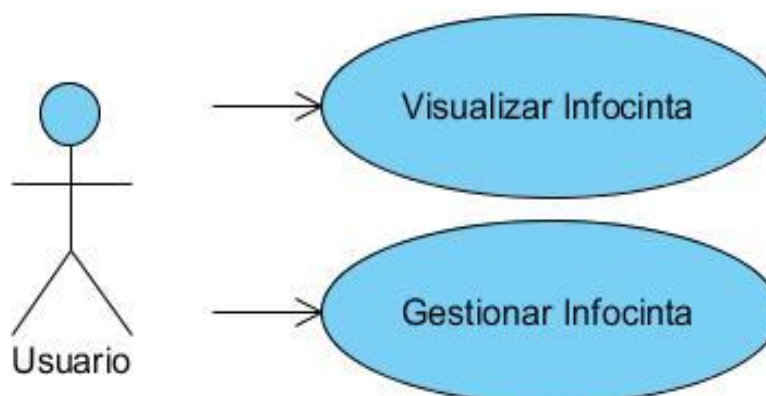
Actores	Descripción
Usuario	Persona con los permisos necesarios para acceder al módulo y realizar cualquiera de las funcionalidades presentes en el mismo.

2.4.2. Casos de Uso del Sistema

Un Diagrama de Casos de Uso presenta la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. (23)

A partir de la identificación del actor que interactúa con la aplicación, así como la recopilación del conjunto de funcionalidades escritas en forma de requerimientos, que a su vez han sido agrupados en casos de uso según sus peculiaridades, se conforma el Modelo de Casos de Uso que se representa a continuación.

Figura 3: Diagrama de Casos de Uso del Sistema.



2.4.3. Especificación de los Casos de Uso

Cada uno de los Casos de Uso (CU) identificados presenta características particulares que para su mayor entendimiento se hace necesario describirlas textualmente. Estos describen qué hace el sistema, y no cómo lo hace. Los casos de uso se obtienen de los servicios requeridos por los usuarios finales, lo que significa que representan un hilo conductor como guía en el desarrollo de software. (7) En la especificación de los CU se describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. A continuación se expone una descripción de cada uno de los casos de uso que conforman el módulo.

Capítulo 2: Análisis y Diseño de la Solución Propuesta

Tabla 2: Descripción del CU Gestionar Infocinta.

Caso de Uso:	Gestionar Infocinta.	
Actores:	Usuario.	
Resumen:	El caso de uso comienza cuando el actor decide adicionar, editar, eliminar, listar, ordenar o filtrar las infocintas. El actor realiza alguna de las acciones que brinda el sistema y termina así el caso de uso.	
Precondiciones:	El actor se ha autenticado y tiene los permisos necesarios para gestionar las infocintas.	
Pos condiciones:	Se actualizan las infocintas gestionadas.	
Referencias:	RF1, RF2, RF2.1, RF2.2, RF2.3, RF2.4, RF2.5, RF2.6, RF2.7, RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5, RF3.6, RF3.7, RF4, RF5, RF7, RF8, RF9.	
Prioridad:	Crítico.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso inicia cuando el actor accede a la opción "Infocintas" del menú principal.	2. Muestra un listado con las infocintas que se encuentran en el sistema. Además se brinda la posibilidad de mostrar, adicionar, editar, eliminar, ordenar, filtrar y restablecer las infocintas. (ver interfaz 1)
	3. Selecciona una opción de las que brinda el sistema.	4. El sistema realiza la operación según la opción que escoja el usuario: a) Si selecciona Mostrar (A), ver CU Visualizar Infocinta. b) Si selecciona Adicionar (B), ver sección Adicionar infocinta. c) Si selecciona Editar (C), ver sección Editar infocinta. d) Si selecciona Eliminar (D), ver sección Eliminar infocinta. e) Si selecciona Ordenar (E), ver sección Ordenar infocinta. f) Si selecciona Filtrar (F), ver sección Filtrar infocinta. g) Si selecciona Restablecer (G) se recarga el listado con todas las infocintas existentes.
Prototipo de Interfaz		

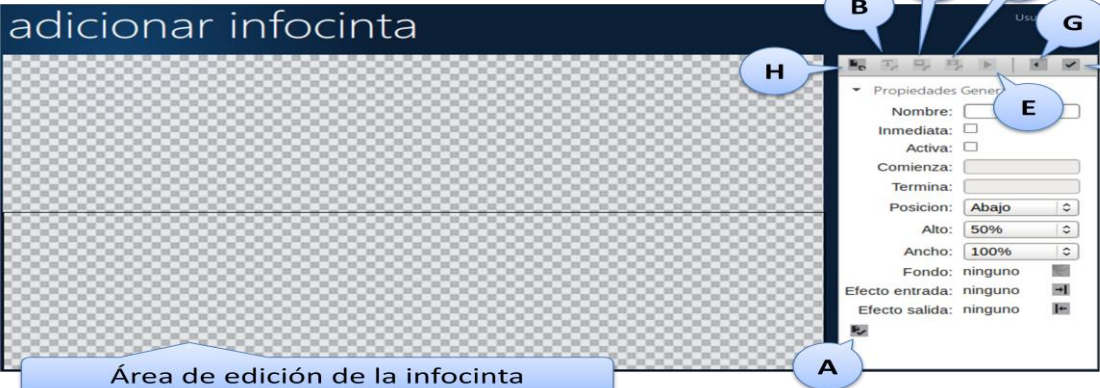




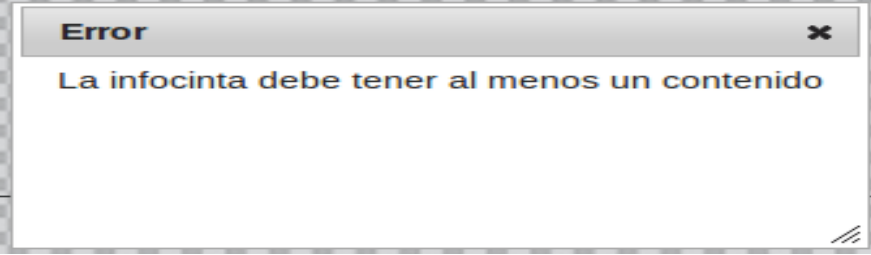
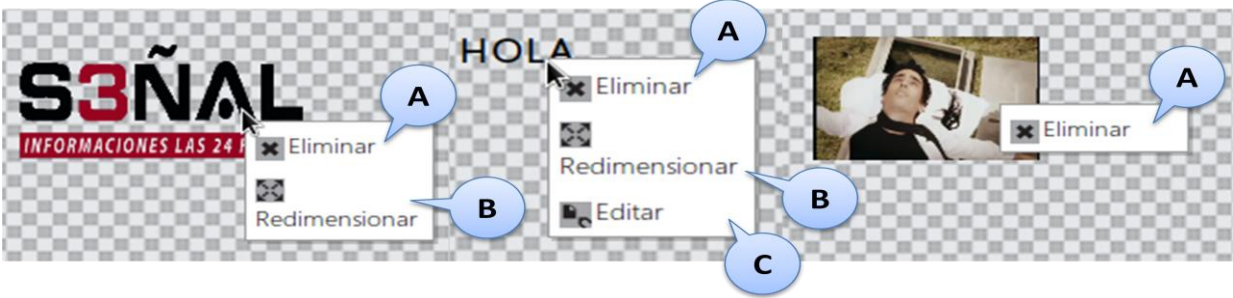
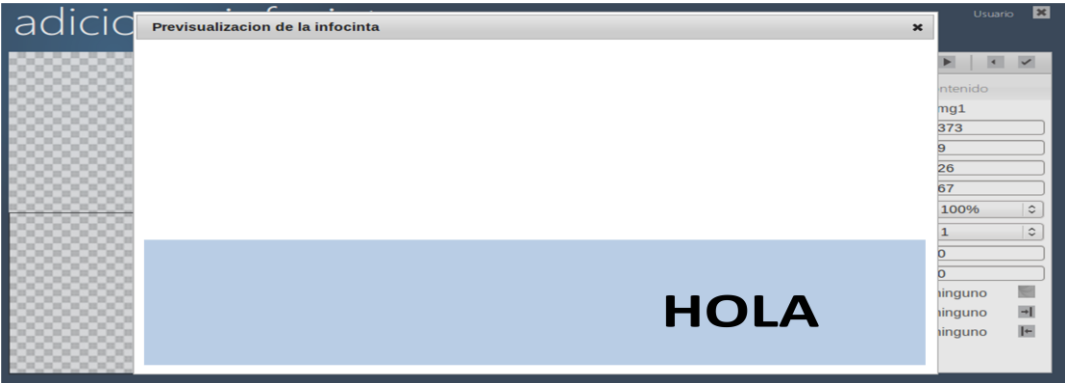
Sección "Adicionar infocinta"

	<ol style="list-style-type: none"> 1. Muestra una interfaz con un panel propiedades generales para especificar un nombre, seleccionar si estará activa, si será inmediata, una posición en la pantalla, un tamaño y una alineación, seleccionar un fondo y un efecto de entrada o salida. Además muestra el área de edición de la infocinta (ver interfaz 2)
<ol style="list-style-type: none"> 2. El usuario llena los campos de propiedades generales y presiona el botón Aplicar Propiedades (A). 	<ol style="list-style-type: none"> 3. Actualiza los valores generales en el área de edición, oculta el panel de propiedades generales, muestra el panel de propiedades de contenido y habilita las opciones para insertar y previsualizar los mismos. (Insertar texto (B), Insertar imagen (C), Insertar Video (D) y Previsualizar (E)). (ver interfaz 3)
<ol style="list-style-type: none"> 4. Presiona cualquier botón para insertar un contenido. 	<ol style="list-style-type: none"> 5. Muestra una ventana emergente para la selección de los contenidos a insertar (para escribir el texto, para seleccionar la imagen y para seleccionar el video). Inserta el contenido en el área de edición. (ver interfaz 3 y 4)
<ol style="list-style-type: none"> 6. Selecciona uno con un click, modifica las propiedades de contenido y presiona el botón Aplicar propiedades (A). 	<ol style="list-style-type: none"> 7. Aplica las propiedades del contenido seleccionado y actualiza las mismas en el área de edición. (ver interfaz 3)
<ol style="list-style-type: none"> 8. Presiona el botón Guardar (F) 	<ol style="list-style-type: none"> 9. Almacena en la base de datos una nueva infocinta con los datos proporcionados por el

Capítulo 2: Análisis y Diseño de la Solución Propuesta

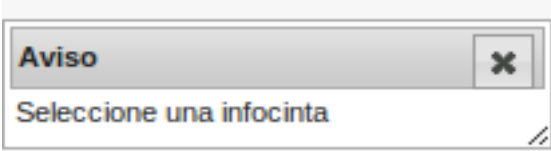
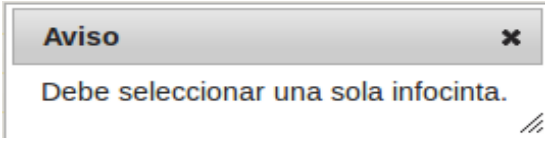
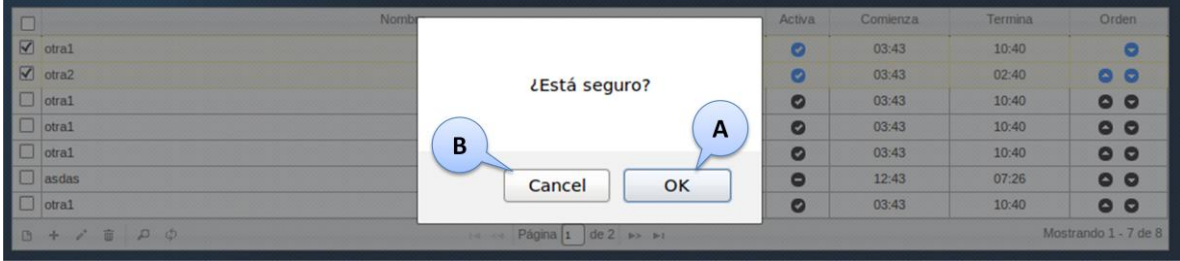
	actor y regresa al listado de infocintas.
Flujos Alternos	
2. El usuario presiona el botón Aplicar Propiedades (A) sin cambiar ningún valor.	3. Mantiene los valores generales por defecto en el área de edición, oculta el panel de propiedades generales, muestra el panel de propiedades de contenido y habilita las opciones para insertar y previsualizar los mismos. (Insertar texto (B), Insertar imagen (C), Insertar Video (D) y Previsualizar (E)). (ver interfaz 3)
2. El usuario presiona el botón Guardar (F) sin llenar los campos.	3. Muestra un cartel de error: Debe llenar todos los campos requeridos. (ver interfaz 5)
2. El usuario presiona el botón Guardar (F) con los valores de los campos, pero sin insertar ningún contenido.	3. Muestra un cartel de error: Debe insertar al menos un contenido. (ver interfaz 6)
2. El usuario presiona el botón Volver a la lista (G)	3. Regresa al listado de infocintas sin almacenar ningún valor.
6. El usuario presiona click derecho sobre un contenido y selecciona la opción Eliminar (A), Redimensionar (B) o Editar (C).	7. Se muestra un menú contextual con las opciones de eliminar, redimensionar en el caso del texto y la imagen, y editar en el caso del texto. Si se presiona eliminar, el sistema elimina el contenido del área de edición; si se presiona redimensionar el usuario puede modificar el tamaño; si se presiona editar, se muestra una ventana que permita la edición del texto. (ver interfaz 7)
8. El usuario presiona el botón Previsualización (E)	9. Muestra una ventana emergente donde se puede observar cómo se comportara la infocinta y sus contenidos al ser transmitida. (ver interfaz 8)
Prototipos de Interfaz	

Interfaz 2	 <p>adicionar infocinta</p> <p>Área de edición de la infocinta</p> <p>Propiedades Generales</p> <ul style="list-style-type: none">Nombre: <input type="text"/>Inmediata: <input type="checkbox"/>Activa: <input type="checkbox"/>Comienza: <input type="text"/>Termina: <input type="text"/>Posición: AbajoAlto: 50%Ancho: 100%Fondo: ningunoEfecto entrada: ningunoEfecto salida: ninguno
Interfaz 3	 <p>adicionar infocinta</p> <p>Área de edición de la infocinta</p> <p>Propiedades Contenido</p> <ul style="list-style-type: none">Identificador: img1Posición X: 228Posición Y: 51Alto: 83Ancho: 294Opacidad: 100%Profundidad: 1Entrada: 0Tiempo de vida: 0Fondo: ningunoEfecto entrada: ningunoEfecto salida: ninguno
Interfaz 4	 <p>Insertar Texto</p> <p>Insertar Imagen</p> <p>Insertar Video</p>
Interfaz 5	<p>Debe llenar todos los campos requeridos.</p> <p>Aceptar</p>

Interfaz 6	
Interfaz 7	
Interfaz 8	
Sección "Editar infocinta"	
<p>1. El usuario selecciona una infocinta del listado y presiona el botón Editar (C). (ver interfaz 1)</p>	<p>2. Muestra una interfaz con un panel propiedades generales con los valores que trae la infocinta seleccionada. Además muestra el área de edición de la infocinta (ver interfaz 2)</p>
<p>3. El usuario modifica los campos de propiedades generales y presiona el botón Aplicar Propiedades (A).</p>	<p>4. Actualiza los valores generales en el área de edición, oculta el panel de propiedades generales, muestra el panel de propiedades de contenido y habilita las opciones para insertar y previsualizar los mismos. (Insertar texto (B), Insertar imagen (C), Insertar Video (D) y Previsualizar (E)). Además muestra los</p>

Capítulo 2: Análisis y Diseño de la Solución Propuesta

	contenidos de la infocinta seleccionada con sus propiedades correspondientes. (ver interfaz 3)
5. Presiona cualquier botón para insertar un contenido.	6. Muestra una ventana emergente para la selección de los contenidos a insertar (para escribir el texto, para seleccionar la imagen y para seleccionar el video). Inserta el contenido en el área de edición. (ver interfaz 3 y 4)
7. Selecciona uno con un click, modifica las propiedades de contenido y presiona el botón Aplicar propiedades (A).	8. Aplica las propiedades del contenido seleccionado y actualiza las mismas en el área de edición. (ver interfaz 3)
9. Presiona el botón Guardar (F)	10. Almacena en la base de datos los nuevos valores y contenidos de la infocinta y regresa al listado de infocintas.
Flujos Alternos	
1. El usuario no selecciona ninguna infocinta del listado y presiona el botón Editar (C).	2. Muestra un mensaje de error: Seleccione una infocinta. (ver interfaz 9)
1. El usuario selecciona más de una infocinta del listado y presiona el botón Editar (C).	2. Muestra un mensaje de error: Seleccione una sola infocinta. (ver interfaz 10)
3. El usuario presiona el botón Aplicar Propiedades (A) sin cambiar ningún valor.	4. Mantiene los valores generales por defecto en el área de edición, oculta el panel de propiedades generales, muestra el panel de propiedades de contenido y habilita las opciones para insertar y previsualizar los mismos. (Insertar texto (B), Insertar imagen (C), Insertar Video (D) y Previsualizar (E)). (ver interfaz 3)
3. El usuario presiona el botón Guardar (F) dejando campos vacíos.	4. Muestra un cartel de error: Debe llenar todos los campos requeridos. (ver interfaz 5)
3. El usuario presiona el botón Guardar (F) con los valores de los campos, pero sin insertar ningún contenido.	4. Muestra un cartel de error: Debe insertar al menos un contenido. (ver interfaz 6)
3. El usuario presiona el botón Volver a la lista (G)	4. Regresa al listado de infocintas sin cambiar los valores de la infocinta.
7. El usuario presiona click derecho sobre un contenido y selecciona la opción Eliminar (A), Redimensionar (B) o Editar (C).	8. Se muestra un menú contextual con las opciones de eliminar, redimensionar en el caso del texto y la imagen, y editar en el caso del texto. Si se presiona eliminar, el sistema elimina el contenido del área de edición; si se presiona redimensionar el usuario puede modificar el tamaño; si se presiona editar, se muestra una ventana que permita la edición del texto. (ver interfaz 7)

7. El usuario presiona el botón Previsualización (E)	8. Muestra una ventana emergente donde se puede observar cómo se comportara la infocinta y sus contenidos al ser transmitida. (ver interfaz 8)
Prototipos de Interfaz	
Interfaz 9	
Interfaz 10	
Sección "Eliminar infocinta"	
1. Selecciona una o varias infocintas y presiona el botón Eliminar (D). (ver interfaz 1)	2. Muestra un mensaje de confirmación sobre la eliminación de las infocinta. (ver interfaz 11)
3. El usuario confirma que desea eliminar las infocintas y presiona el botón OK (A).	4. Se eliminan las infocintas del sistema y se actualiza el listado.
Flujos Alternos	
1. No selecciona ninguna infocinta y presiona el botón Eliminar (D).	2. Muestra un mensaje de error: Seleccione una infocinta. (ver interfaz 9)
3. Presiona el botón Cancelar (B).	4. No se elimina la infocinta del sistema. Cierra el mensaje de confirmación.
Prototipos de Interfaz	
Interfaz 11	
Sección "Ordenar Infocinta"	
1. Presiona el botón Ascender (A) o Descender (B) y mueve la infocinta hacia la posición deseada. (ver interfaz 12)	2. Desplaza las infocintas, insertando la que el usuario mueve al lugar escogido. Guarda los cambios de forma automática en la base de datos y actualiza el listado.
Prototipos de Interfaz	

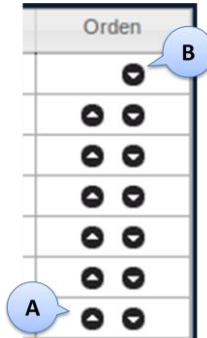
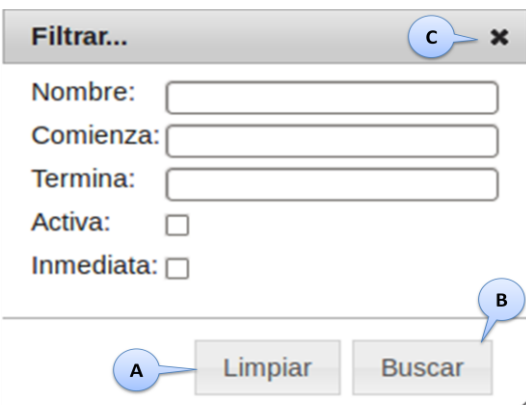

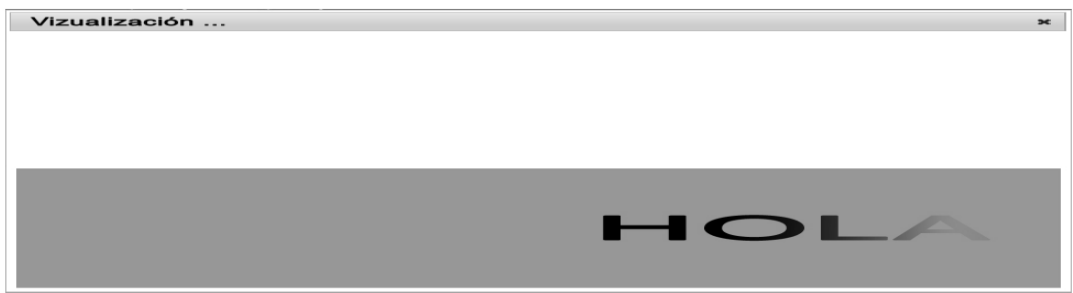
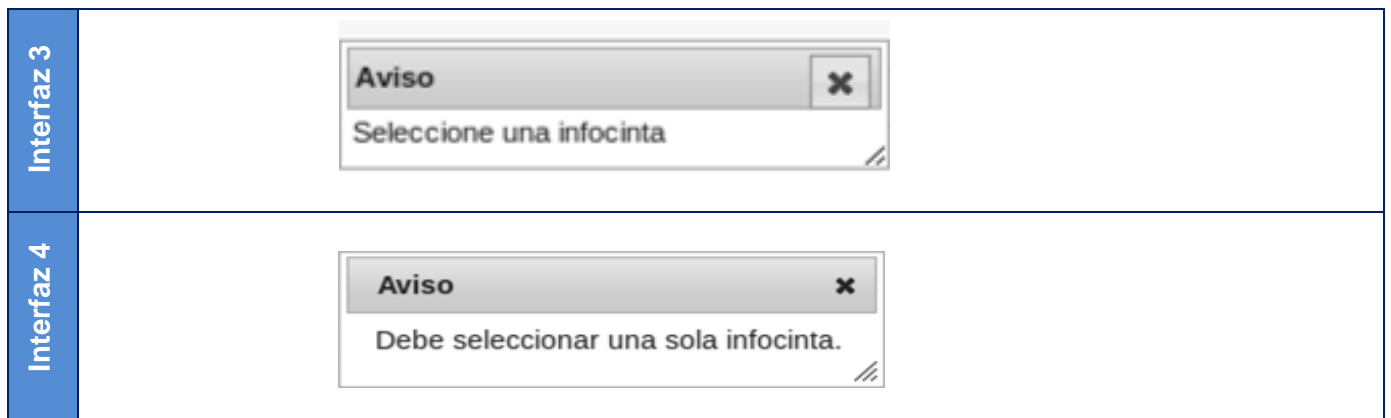
Interfaz 12	
Sección “Filtrar Infocintas”	
	<ol style="list-style-type: none"> 1. Muestra una ventana con los criterios de búsqueda: nombre, activa, inmediata, comienza o termina. (ver interfaz 13)
<ol style="list-style-type: none"> 2. Introduce los criterios de búsqueda (nombre, activa, inmediata, comienza o termina) y presiona el botón Buscar (A). 	<ol style="list-style-type: none"> 3. Muestra en el listado las infocintas que cumplan con los criterios de búsqueda.
Flujos Alternos	
<ol style="list-style-type: none"> 2. Presiona el botón Limpiar (B). 	<ol style="list-style-type: none"> 3. Muestra en el listado todas las infocintas y reinicia los criterios de búsqueda.
<ol style="list-style-type: none"> 2. Presiona el botón Cerrar (B). 	<ol style="list-style-type: none"> 3. Muestra el listado con el filtrado de las infocintas, o con todas las infocintas si no se ha realizado la búsqueda.
Prototipos de Interfaz	
Interfaz 13	

Tabla 3: Descripción del CU Visualizar Infocinta.

Caso de Uso:	Visualizar Infocinta.
Actores:	Usuario.

Resumen:	El caso de uso comienza cuando el actor decide mostrar una de las infocintas listadas. El actor espera que se visualice la infocinta o la cierra, terminando así el caso de uso.	
Precondiciones:	El actor se ha autenticado y tiene los permisos necesarios para visualizar las infocintas.	
Pos condiciones:	Se visualiza la infocinta seleccionada por el actor.	
Referencias:	RF6.	
Prioridad:	Alta.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona una infocinta del listado y presiona el botón Mostrar (A). (ver interfaz 1)	2. Muestra una simulación de cómo se verá la infocinta al ser transmitida. (ver interfaz 2)	
3. Selecciona la opción Cerrar (A).	4. Regresa a la lista de infocintas.	
Flujos Alternos		
1. El usuario no selecciona ninguna infocinta del listado y presiona el botón Mostrar (A).	2. Muestra un mensaje de error: Seleccione una infocinta. (ver interfaz 3)	
1. El usuario selecciona más de una infocinta del listado y presiona el botón Mostrar (A).	2. Muestra un mensaje de error: Seleccione una sola infocinta. (ver interfaz 4)	
Prototipos de Interfaz		
Interfaz 1		
Interfaz 2		



2.5. Modelo de Análisis

El modelo de análisis es una representación abstracta del sistema que se está modelando, suficiente para capturar la lógica esencial del sistema sin entrar en muchos detalles. El modelo de análisis, es la primera representación técnica de un sistema. Utiliza una combinación de formatos en texto y diagramas para representar los requisitos de los datos, las funciones y el comportamiento de una manera que es relativamente fácil de entender y, aún más importante, conduce a una revisión para lograr la corrección, la integralidad y la consistencia. (22)

Los diagramas que se muestran son el resultado del análisis de los requisitos capturados en el flujo de trabajo anterior y el modelo de casos de uso del sistema, que servirán como primera entrada al diseño.

2.5.1. Diagramas de Clases del Análisis

En los siguientes diagramas se representan las clases del análisis y las relaciones que se establecen entre ellas. Las clases del análisis son clases asociadas al contexto del dominio del problema y que describen la realización de los casos de uso.

Figura 4: Diagrama de Clases del Análisis del CU Visualizar Infocinta.

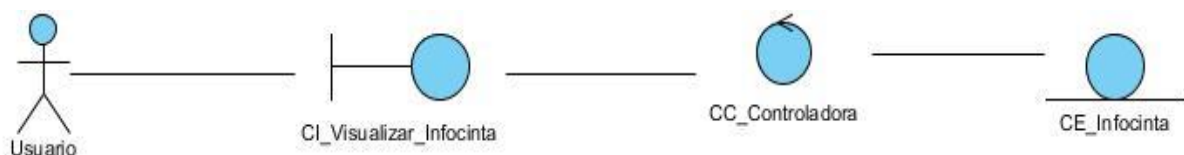
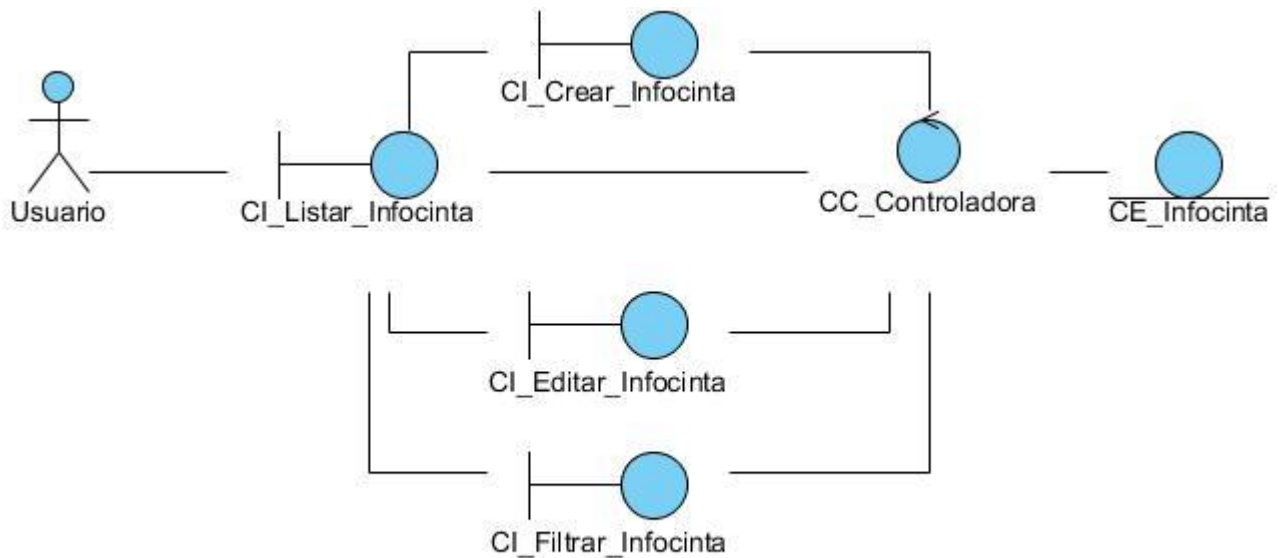


Figura 5: Diagrama de Clases del Análisis del CU Gestionar Infocinta.



2.6. Arquitectura del software

La arquitectura del software alude a la estructura general del software y las formas en que la estructura proporciona una integridad conceptual para un sistema. En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa, la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes. En un sentido más amplio, los componentes pueden generalizarse para representar elementos importantes del sistema y sus interacciones. (22)

La selección de un patrón arquitectónico es una decisión elemental del diseño, define la estructura básica de un sistema. El patrón Modelo Vista Controlador (MVC) es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones web, facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite no mezclar lenguajes de programación en el mismo código (24), en otras palabras, el patrón de diseño MVC organiza el código en base a su función. Este patrón ayuda a darle cierta estructura lógica a la aplicación.

MVC divide las aplicaciones en tres niveles de abstracción:

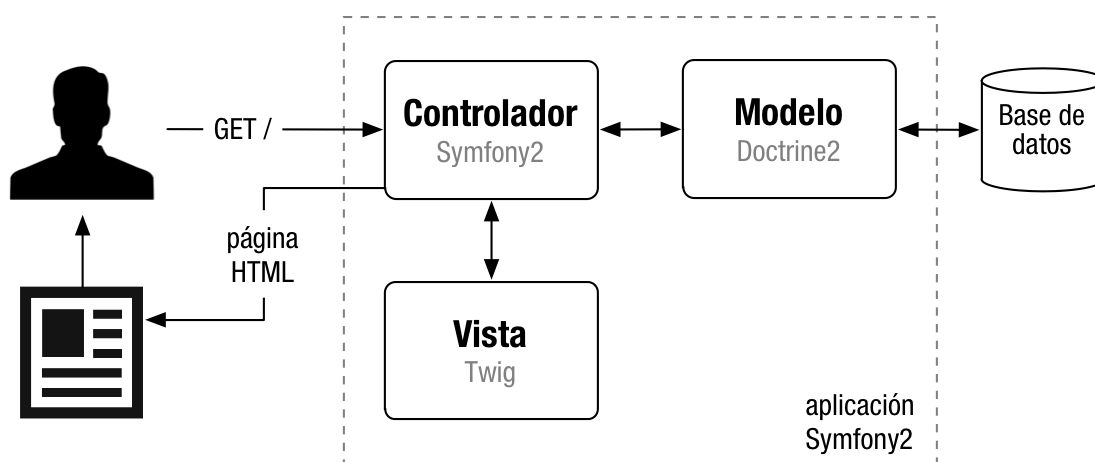
Modelo: Representa la lógica de negocios. Es el encargado del acceso a los datos actuando como intermediario con la base de datos.

Vista: Es la encargada de mostrar la información al usuario de forma gráfica.

Controlador: Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario.

Symfony2 utiliza el patrón arquitectónico MVC tomando lo mejor del mismo para garantizar un modelo, una vista y una controladora de forma independiente. El mismo accede a realizar modificaciones en cada una de las partes sin afectar las demás, permitiendo que el desarrollo de aplicaciones sea rápido y sencillo.

Figura 6: Esquema simplificado del funcionamiento interno de Symfony2. (27)



2.7. Patrones de diseño

Los patrones de diseño se aplican a un elemento específico de diseño como un agregado de componentes para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar las comunicaciones de componente a componente. (22)

Debido a la amplia variedad y cantidad de patrones de diseño que existen, es casi imposible analizarlos. Por lo anterior se recomienda el uso de los patrones que estén predeterminados en cada uno de los estilos que se seleccionen para la arquitectura.

A continuación se describen los patrones de principios generales para asignar responsabilidades (GRASP) y los patrones del diseño orientado a objetos (GoF) que implementa Symfony.

Patrones GRASP

Patrón Experto:

Este patrón se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa simplemente la “intuición” de que los objetos hacen cosas relacionadas con la información que poseen. (20)

Es uno de los más utilizados en Symfony, ya que Doctrine2 es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Patrón Creador:

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. (20)

En la clase DefaultController se encuentran las acciones definidas para el módulo gestión de infocintas y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase DefaultController es la creadora de dichas entidades.

Patrón Bajo Acoplamiento:

El Bajo Acoplamiento estimula la asignación de una responsabilidad de modo que no incremente el acoplamiento al punto de producir resultados negativos propios de un alto acoplamiento.

El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. (20)

En Symfony2, la clase DefaultController hereda solamente de Controller, que es una clase estable en cuanto a su implementación, garantizando un bajo acoplamiento entre las clases.

Patrón Alta Cohesión:

La Alta Cohesión es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Se da una alta cohesión funcional cuando los elementos de un componente (clase, por ejemplo) colaboran para producir algún comportamiento bien definido. (20)

Symfony2 permite asignar responsabilidades a una clase con una alta cohesión pues plantea que debe existir una acción para cada una de las plantillas que producen la vista. Además la clase Controller colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, proporcionando que el software sea flexible frente a grandes cambios.

Patrones GoF

Patrones Creacionales:

Abstract Factory: También conocido como Fábrica abstracta, permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí haciendo transparente el tipo de familia concreta que se esté usando. (20)

Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Como la definición por defecto de la factoría para las peticiones es `getRequest()`, Symfony2 crea un objeto de esta clase para tratar con las peticiones.

Patrones Estructurales:

Decorator: También conocido como Decorador añade funcionalidad a una clase de una forma dinámica y transparente. (25)

En Symfony2 los archivos `base.html.twig` y `layout.html.twig` almacenan el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de las plantillas se integra en estos archivos, o si se mira desde el otro punto de vista, el archivo base y el layout decoran el contenido de las plantillas.

2.8. Modelo de Diseño

La peculiaridad del diseño es modelar el sistema, encontrar su forma (incluida la arquitectura) para que soporte todos sus requisitos. Una entrada esencial en el diseño es el resultado del análisis, es decir, el Modelo del Análisis.

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de usos centrándose en cómo los requisitos funcionales y requisitos no funcionales, junto con otras restricciones relacionadas con el entorno de la implementación, tienen impacto en el sistema a desarrollar. Sirve de abstracción de la implementación del sistema y es de ese modo utilizada como una entrada fundamental de las actividades de implementación. (7)

2.8.1. Diagramas de Clases del Diseño

Los diagramas de clases del diseño servirán de apoyo para la implementación de las funcionalidades. Constituyen la mayor aproximación al módulo que se está proponiendo, por lo que una buena modelación de las clases del diseño contribuye a que el software se implemente con buena calidad. El diseño propuesto se estructura en consecuencia con el patrón arquitectónico MVC.

Figura 7: Diagrama de Clases del Diseño del CU Gestionar Infocinta.

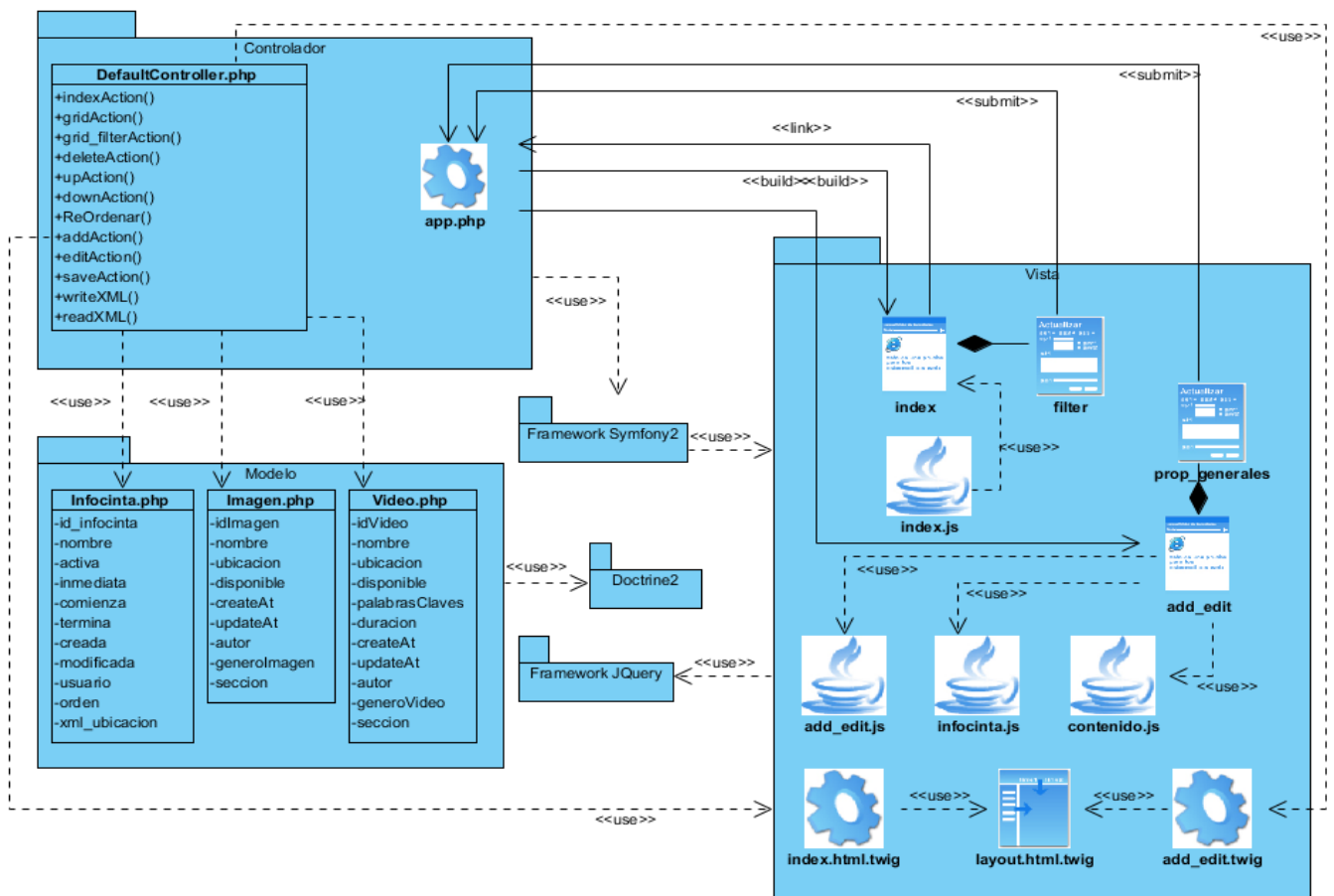
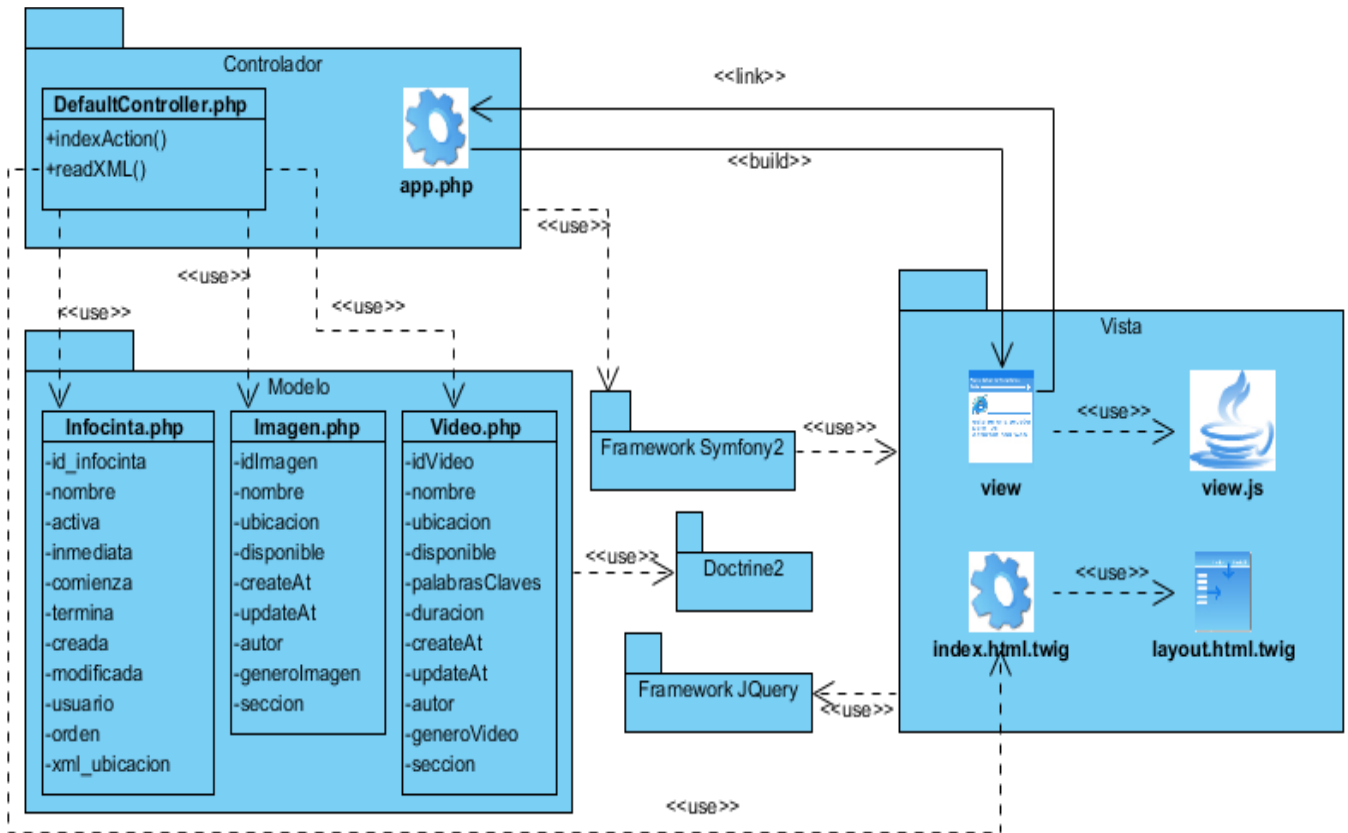
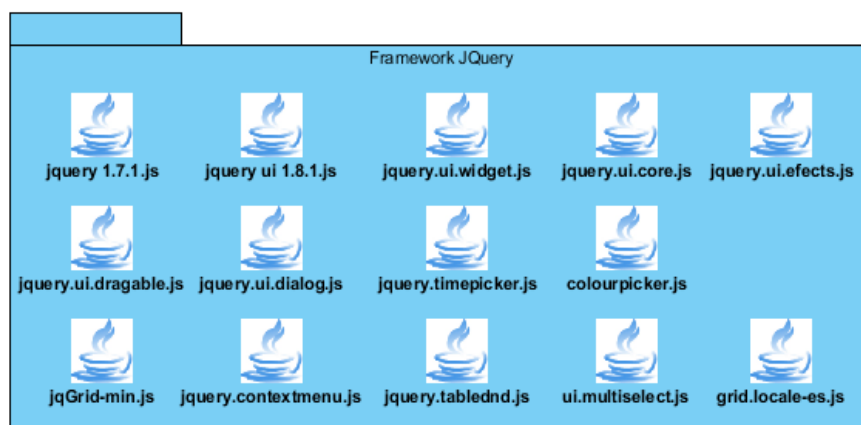


Figura 8: Diagrama de Clases del Diseño del CU Visualizar Infocinta.



El diagrama siguiente describe detalladamente el framework JQuery, utilizado para lograr un mejor diseño de la interfaz y una rápida respuesta a los usuarios.

Figura 9: Framework JQuery



2.9. Conclusiones

En el capítulo se trataron los temas necesarios para definir las características con las que contará el Módulo de Infocinta. Se desarrolló el Modelo de Dominio, que permitió comprender los conceptos principales asociados al sistema. Se definieron los requisitos funcionales y no funcionales que debe tener el sistema, sirviendo de guía para la definición de las funcionalidades del sistema, plasmadas en los Casos de Uso. Se construyen los Diagramas de Clases del Análisis como primer paso para llegar a la construcción de los Diagramas de Clases del Diseño, los cuales reflejan las clases a tener en cuenta para implementar las nuevas funcionalidades. Se realizó un análisis de la arquitectura Modelo-Vista-Controlador, y de los patrones de diseño utilizados por Symfony2, permitiendo una mejor comprensión del funcionamiento del framework.

CAPÍTULO 3: Implementación y Validación de la Solución

3.1. Introducción

En el presente capítulo se describe la implementación del sistema, después de haber realizado en el capítulo anterior el análisis y el diseño. Se realizan los diagramas de componentes del Módulo Gestión de Infocintas los cuales han sido desarrollados, así como el diagrama de despliegue. También se presenta todo lo relacionado con las pruebas realizadas al sistema, encaminadas a validar la solución.

3.2. Modelo de Implementación

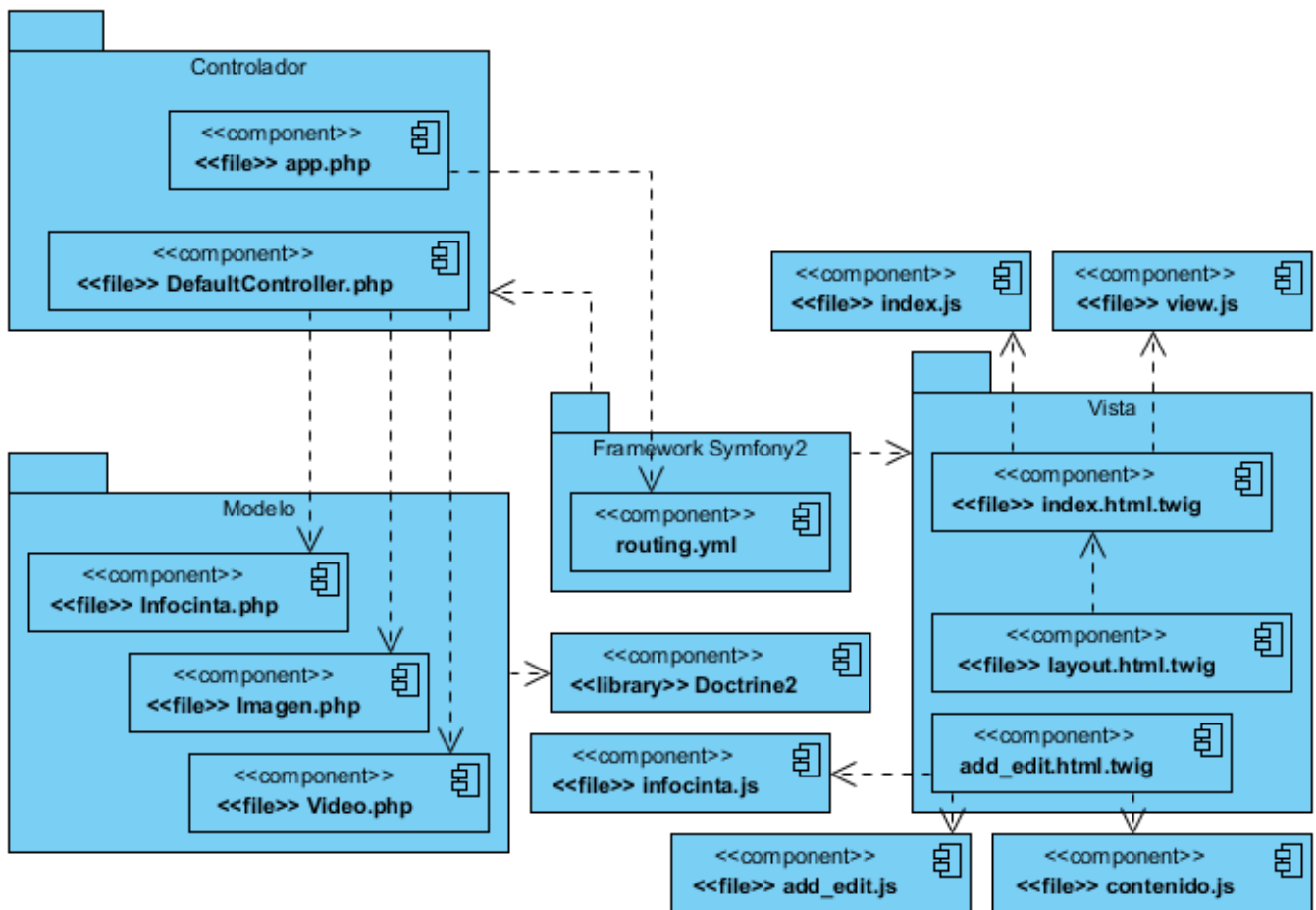
El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (7)

3.2.1. Diagrama de Componentes

El Diagrama de Componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. En el Diagrama de Componentes se modelan componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes y paquetes de componentes. (26)

Este diagrama ofrece una vista de la estructura de los componentes desarrollados o por desarrollar. Los componentes pueden ser ficheros, ejecutables, librerías, tablas de la base de datos o documentos. A continuación se muestra la vista general del diagrama de componentes.

Figura 10: Diagrama de Componentes General.



3.3. Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Puede describir diferentes configuraciones de red, incluidas las configuraciones para prueba y para simulación. (7)

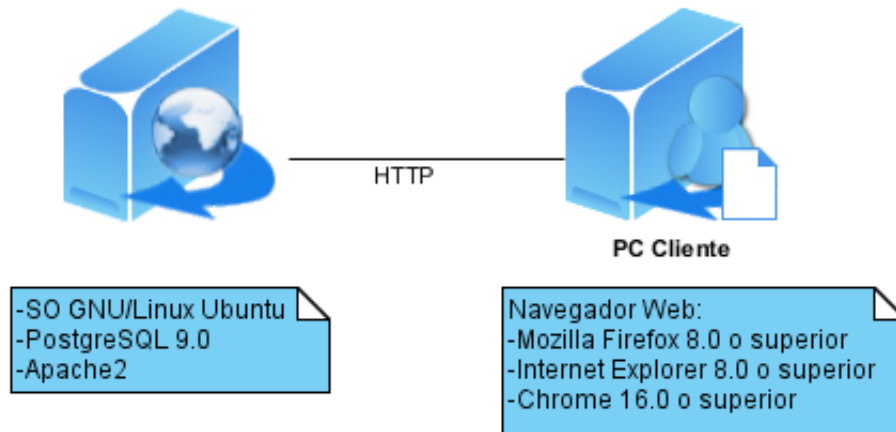
3.3.1. Diagrama de Despliegue

El diagrama de despliegue muestra la configuración de los nodos de proceso y las instancias de componentes y objetos que residen en ellos. Los diagramas de despliegue son, en gran medida, semejantes a los diagramas de objetos. Generalmente, muestran las instancias individuales del nodo implicadas en un sistema. (8)

Capítulo 3: Implementación y Validación de la Solución

Un diagrama de despliegue se representa como un conjunto de nodos unidos por conexiones de comunicación, objetos y procesos que representan manifestaciones del código en tiempo de ejecución. Muestra las relaciones físicas entre los componentes hardware y software en el sistema final.

Figura 11: Diagrama de Despliegue.



Nodo PC Cliente: Representa una PC que puede ser usada por un usuario para acceder a la aplicación y consumir los servicios que se brindan en ella mediante un navegador web.

Nodo Servidor de Administración: Este nodo representa el servidor donde estará alojado el Subsistema de Administración, el cual tendrá como sistema gestor de base de datos, PostgreSQL 9.0 y como servidor web, Apache2.

Protocolo HTTP: Este protocolo se utiliza para el acceso a sitios web. Consta de cuatro etapas entre el navegador y el servidor, conexión, solicitud, respuesta y desconexión y por lo general trabaja sobre el puerto 80.

3.4. Pruebas al sistema propuesto

El desarrollo de un software implica la realización de una serie de actividades encaminadas a encontrar errores, por lo que se deben incorporar acciones que evalúen la calidad del producto que se está desarrollando. Dentro del proceso de desarrollo de un software, el flujo de trabajo de Prueba, es mediante el que se puede validar que el diseño y los requisitos se estén cumpliendo satisfactoriamente.

Capítulo 3: Implementación y Validación de la Solución

Las pruebas de software son el proceso de revisar el funcionamiento de un programa con intención de encontrarle defectos. Se dice que una prueba exitosa es aquella que logra descubrir problemas no detectados.

Las pruebas se pueden realizar basándose en dos técnicas diferentes: pruebas de caja blanca y pruebas de caja negra. Las de caja blanca comprueban los caminos lógicos del software mediante la elaboración de un examen minucioso de los detalles procedimentales. Por tanto, requieren del conocimiento de la estructura interna del programa y son realizadas a partir de las especificaciones internas de diseño o el código. Las de caja negra se llevan a cabo sobre la interfaz del software, sin tener en cuenta la estructura interna del mismo. Tratan de demostrar que todo funcione correctamente, se centra principalmente en los requisitos funcionales del software. Permiten encontrar errores de interfaz, funciones incorrectas o ausentes, o errores de inicialización y terminación.

Analizando lo anterior, se ha decidido realizar pruebas de caja negra al Módulo Gestión de Infocintas, con el fin de medir la fortaleza y robustez del sistema, rectificar posibles errores que sean detectados y de esta forma aumentar su calidad.

3.4.1. Casos de prueba y resultados

Para aplicar las pruebas se necesita diseñar los casos de prueba correspondientes. Un caso de prueba que especifica como probar un caso de uso o un escenario específico de un caso de uso incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y postcondiciones especificadas por el caso de uso y que se sigue la secuencia de pasos especificadas por el caso de uso. (7) Un caso de prueba basado en un caso de uso especifica típicamente una prueba del sistema como caja negra.

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

Tabla 4: Secciones a probar en el CU Gestionar Infocinta.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Adicionar infocinta.	EC 1.1: Adicionar infocinta satisfactoriamente.	El usuario llena todos los campos, inserta al menos un contenido y presiona el botón guardar. El sistema

Capítulo 3: Implementación y Validación de la Solución

		valida los datos y almacena la infocinta en la base de datos.
	EC 1.2: Adicionar infocinta con campos vacíos.	El usuario deja campos vacíos y presiona el botón guardar. El sistema muestra un cartel de error indicando que debe llenar todos los campos requeridos.
	EC 1.3: Adicionar infocinta sin contenidos.	El usuario no inserta ningún contenido y presiona el botón guardar. El sistema muestra un cartel de error indicando que debe insertar al menos un contenido.
	EC 1.4: Adicionar infocinta con fechas incorrectas.	El usuario llena los campos insertando una fecha de inicio mayor a la fecha de fin y presiona el botón guardar. El sistema muestra un cartel de error: La fecha de inicio debe ser menor que la fecha de fin.
	EC 1.5: Propiedades generales.	El usuario presiona el botón propiedades generales. El sistema oculta los contenidos y las propiedades de estos y muestra el panel de propiedades generales.
	EC 1.6: Previsualizar infocinta.	El usuario, después de haber insertado algún contenido presiona el botón previsualizar infocinta. El sistema muestra una ventana donde se puede observar cómo se comportará la infocinta y sus contenidos al ser transmitida.
	EC 1.7: Presiona el botón volver a la lista.	El usuario presiona el botón volver a la lista. El sistema regresa a la lista y no almacena la infocinta.

Capítulo 3: Implementación y Validación de la Solución

SC 2: Editar infocinta.	EC 2.1: Editar infocinta satisfactoriamente.	El usuario modifica los campos, los contenidos y presiona el botón guardar. El sistema valida los datos y almacena la infocinta en la base de datos.
	EC 2.2: Editar infocinta con campos vacíos.	El usuario deja campos vacíos y presiona el botón guardar. El sistema muestra un cartel de error indicando que debe llenar todos los campos requeridos.
	EC 2.3: Editar infocinta sin contenidos.	El usuario no elimina los contenidos y presiona el botón guardar. El sistema muestra un cartel de error indicando que debe insertar al menos un contenido.
	EC 2.4: Editar infocinta con fechas incorrectas.	El usuario cambia los campos insertando una fecha de inicio mayor a la fecha de fin y presiona el botón guardar. El sistema muestra un cartel de error: La fecha de inicio debe ser menor que la fecha de fin.
	EC 2.5: Propiedades generales.	El usuario presiona el botón propiedades generales. El sistema oculta los contenidos y las propiedades de estos y muestra el panel de propiedades generales.
	EC 2.6: Previsualizar infocinta.	El usuario, después de haber modificado algún contenido presiona el botón previsualizar infocinta. El sistema muestra una ventana donde se puede observar cómo se comportará la infocinta y sus contenidos al ser transmitida.

Capítulo 3: Implementación y Validación de la Solución

	EC 2.7: Presiona el botón volver a la lista.	El usuario presiona el botón volver a la lista. El sistema regresa a la lista y no guarda los cambios realizados.
SC 3: Eliminar infocinta.	EC 3.1: Eliminar infocinta satisfactoriamente.	El usuario selecciona una o varias infocintas y presiona el botón Eliminar infocinta. El sistema muestra un mensaje de confirmación. El usuario acepta. El sistema elimina la infocinta del listado y la borra de la base de datos.
	EC 3.2: Cancelar la opción Eliminar infocinta.	El usuario selecciona una o varias infocintas y presiona el botón Eliminar infocinta. El sistema muestra un mensaje de confirmación. El usuario cancela. El sistema regresa al listado de infocintas.
	EC 3.3: Eliminar infocinta sin que haya ninguna seleccionada del listado.	El usuario no selecciona ninguna infocinta y presiona el botón eliminar infocinta. El sistema muestra un aviso: Debe seleccionar una infocinta.
SC 4: Ordenar Infocinta	EC 4.1: Ordenar infocinta satisfactoriamente.	El usuario selecciona la opción Subir o Bajar para modificar el orden. El sistema actualiza el listado y el orden en la base de datos.
SC 5: Filtrar Infocintas	EC 5.1: Filtrar infocinta satisfactoriamente	El usuario llena los campos por los que desea realizar la búsqueda y presiona el botón buscar. El sistema realiza la búsqueda de los datos y lo muestra en el listado.
	EC 5.2: Limpiar la búsqueda	El usuario presionar el botón limpiar. El sistema recarga el listado, vacía los campos y mantiene la ventana de búsqueda.

Capítulo 3: Implementación y Validación de la Solución

	EC 5.3: Cerrar la búsqueda	El usuario selecciona el botón cerrar. El sistema cierra la ventana y mantiene los datos del listado.
--	----------------------------	--

Tabla 5: Descripción de variables para el CU Gestionar Infocinta.

Nº	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	nombre	Campo de texto	No	En este campo el usuario puede escribir cualquier cadena de caracteres.
2	activa	checkbox	Si	El usuario puede o no seleccionar la opción.
3	inmediata	checkbox	Si	El usuario puede o no seleccionar la opción.
4	comienza	Campo de tiempo	No	En este campo el usuario debe seleccionar un tiempo.
5	termina	Campo de tiempo	No	En este campo el usuario debe seleccionar un tiempo.

Tabla 6: Matriz de datos para la sección Adicionar infocinta.

Escenario	Nombre	Activa	Inmediata	Comienza	Termina	Respuesta del sistema	Resultado de la prueba
Adicionar infocinta satisfactoriamente.	V	N/A	N/A	V	V	El sistema valida los datos y almacena la infocinta en la base de datos.	Satisfactorio
Adicionar infocinta con campos vacíos.	I	N/A	N/A	I	V	El sistema muestra un cartel de error: Debe llenar los campos requeridos.	No satisfactorio
Adicionar infocinta	N/A	N/A	N/A	N/A	N/A	El sistema	No

Capítulo 3: Implementación y Validación de la Solución

sin contenidos.						muestra un cartel de error: debe insertar al menos un contenido.	satisfactorio
Adicionar infocinta con fechas incorrectas.	V	N/A	N/A	I	I	El sistema muestra un cartel de error: La fecha de inicio debe ser menor a la fecha de fin.	Satisfactorio
Propiedades generales.	N/A	N/A	N/A	N/A	N/A	El sistema oculta los contenidos y las propiedades de estos y muestra el panel de propiedades generales.	Satisfactorio
Previsualizar infocinta.	N/A	N/A	N/A	N/A	N/A	El sistema muestra una ventana donde se observa cómo se comportará la infocinta al ser transmitida.	Satisfactorio
Presiona el botón volver a la lista.	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la lista y no almacena la infocinta.	Satisfactorio

Tabla 7: Matriz de datos para la sección Editar infocinta.

Escenario	Nombre	Activa	Inmediata	Comienza	Termina	Respuesta del sistema	Resultado de la prueba
Editar infocinta satisfactoriamente.	V	N/A	N/A	V	V	El sistema valida los datos y almacena la	Satisfactorio

Capítulo 3: Implementación y Validación de la Solución

						infocinta en la base de datos.	
Editar infocinta con campos vacíos.	I	N/A	N/A	V	V	El sistema muestra un cartel de error: Debe llenar los campos requeridos.	No satisfactorio
Editar infocinta sin contenidos.	N/A	N/A	N/A	N/A	N/A	El sistema muestra un cartel de error: debe insertar al menos un contenido.	No satisfactorio
Editar infocinta con fechas incorrectas.	V	N/A	N/A	I	I	El sistema muestra un cartel de error: La fecha de inicio debe ser menor a la fecha de fin.	Satisfactorio
Propiedades generales.	N/A	N/A	N/A	N/A	N/A	El sistema oculta los contenidos y las propiedades de estos y muestra el panel de propiedades generales.	Satisfactorio
Previsualizar infocinta.	N/A	N/A	N/A	N/A	N/A	El sistema muestra una ventana donde se observa cómo se comportará la infocinta al ser transmitida.	Satisfactorio
Presiona el botón volver a la lista.	N/A	N/A	N/A	N/A	N/A	El sistema regresa a la lista y no guarda los cambios	Satisfactorio

Capítulo 3: Implementación y Validación de la Solución

						realizados.	
--	--	--	--	--	--	-------------	--

Tabla 8: Matriz de datos para la sección Eliminar infocinta.

Escenario	Nombre	Activa	Inmediata	Comienza	Termina	Respuesta del sistema	Resultado de la prueba
Eliminar infocinta satisfactoriamente.	N/A	N/A	N/A	N/A	N/A	El sistema muestra un mensaje de confirmación, el usuario acepta y el sistema elimina la infocinta del listado y la borra de la base de datos.	Satisfactorio
Cancelar la opción Eliminar infocinta.	N/A	N/A	N/A	N/A	N/A	El sistema muestra un mensaje de confirmación, el usuario cancela y el sistema regresa al listado.	Satisfactorio
Eliminar infocinta sin que haya ninguna seleccionada del listado.	N/A	N/A	N/A	N/A	N/A	El sistema muestra el aviso: debe seleccionar una infocinta.	No satisfactorio

Capítulo 3: Implementación y Validación de la Solución

Tabla 9: Matriz de datos para la sección Ordenar infocinta.

Escenario	Nombre	Activa	Inmediata	Comienza	Termina	Respuesta del sistema	Resultado de la prueba
Ordenar infocinta satisfactoriamente.	N/A	N/A	N/A	N/A	N/A	El sistema actualiza el listado y el orden en la base de datos.	Satisfactorio

Tabla 10: Matriz de datos para la sección Filtrar infocinta.

Escenario	Nombre	Activa	Inmediata	Comienza	Termina	Respuesta del sistema	Resultado de la prueba
Filtrar infocinta satisfactoriamente.	V	Marcado	Marcado	N/A	N/A	El sistema realiza la búsqueda de los datos y los muestra en el listado.	Satisfactorio
	N/A	No Marcado	Marcado	V	V		
Limpiar la búsqueda	N/A	N/A	N/A	N/A	N/A	El sistema recarga el listado, vacía los campos y mantiene la ventana de búsqueda.	Satisfactorio
Cerrar la búsqueda	N/A	N/A	N/A	N/A	N/A	El sistema cierra la ventana y mantiene los datos del listado.	Satisfactorio

Capítulo 3: Implementación y Validación de la Solución

Tabla 11: Secciones a probar en el CU Visualizar Infocinta.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Visualizar Infocinta	EC 1.1: Visualizar infocinta satisfactoriamente	El usuario presiona el botón visualizar infocinta, después de haber seleccionado la infocinta que desea mostrar. El sistema muestra una simulación dinámica de cómo se verá la infocinta al ser transmitida. El usuario puede cerrar la visualización antes de que esta finalice, o esperar y el sistema muestra el listado cuando termina la visualización.
	EC 1.2: Visualizar infocinta sin que haya ninguna seleccionada del listado	El usuario no selecciona una infocinta antes de presionar el botón visualizar infocinta. El sistema muestran un aviso: Debe seleccionar una infocinta.

Resultado de las pruebas

Fueron probados dos CU, los cuales representan el cien por ciento de los requisitos funcionales. Para cada CU se tuvo en cuenta los diferentes escenarios que pudieran existir, los cuales implican cada una de las posibles iteraciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. Las pruebas arrojaron cinco no conformidades, las mismas fueron solucionadas inmediatamente, garantizando de esta manera que la aplicación cumpla con todas las especificaciones.

3.5. Conclusiones

En el capítulo se realizó el flujo de implementación y prueba propuesto por RUP. Se representó el Diagrama de Componentes, permitiendo describir los elementos del diseño en términos de componentes y modelar la estructura del software. Se realizó el Diagrama de Despliegue, el cual permite conocer cómo quedará la distribución física del módulo en el momento de realizar el despliegue de la aplicación. También se realizaron pruebas de Caja Negra al software mediante los casos de pruebas que cubrieron la totalidad de sus funcionalidades, permitiendo encontrar errores que

Capítulo 3: Implementación y Validación de la Solución

afectaban el funcionamiento. Los resultados obtenidos durante la fase de pruebas representaron un elemento fundamental para la corrección de los errores encontrados, contribuyendo a la culminación del módulo de forma satisfactoria.

CONCLUSIONES

Los resultados obtenidos con la realización del presente trabajo han dado cumplimiento al objetivo propuesto, comprobándose la idea a defender como respuesta del problema de la investigación.

Con la utilización de los métodos de investigación científica, se completó el estudio de los procesos relacionados con la gestión de infocintas en PRIMICIA, permitiendo la comprensión de la situación problemática de la investigación y sentando las bases teóricas para el desarrollo del software.

Se obtuvo una visión de características positivas para el desarrollo del módulo, a partir del análisis de otros canales que transmiten infocintas.

Se desarrolló el sistema propuesto y se generó la documentación necesaria para la posterior comprensión y reutilización del mismo, a través de la realización del análisis, diseño e implementación de los requisitos definidos.

La solución fue implementada con tecnologías libres cumpliendo con la política de soberanía tecnológica por la que aboga el país. Cuenta con nuevas funcionalidades que aportan dinamismo a las infocintas y permiten a los usuarios la configuración de las mismas.

RECOMENDACIONES

Después de concluir con un estudio profundo y de haber realizado el módulo Gestión de Infocintas, se recomienda seguir investigando con el propósito de identificar aún nuevas funcionalidades que se incluyan en la gestión de infocintas de PRIMICIA. Incluir las infocintas basadas en plantillas, con el objetivo de agilizar el proceso de creación de las mismas. Utilizar fuentes web con el propósito de perfeccionar el proceso de construcción de las infocintas.

BIBLIOGRAFÍA REFERENCIADA

1. Diccionario de la Real Academia Española. *DRAE*. [En línea] Vigésima Edición, Real Academia Española, 2011. http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=m%C3%B3dulo.
2. Alegsá. *Portal de informática, internet, tecnologías y web*. [En línea] ALEGSA, 2011. www.alegsa.com.ar/Dic/modulo.php.
3. **Hernández García, Ruber, y otros**. *Primicia, una Plataforma de Televisión sobre Tecnologías Libres*.
4. **Dirección de Televisión Universitaria**. *Inter-Nos*. [En línea] <http://inter-nos.uci.cu/canalcultural/cc.asp>.
5. Enciclopedia Cubana en la Red (EcuRed). *Canal Multivisión*. [En línea] 2010. <http://www.ecured.cu/index.php/Multivisi%C3%B3n>.
6. Enciclopedia Cubana en la Red (EcuRed). *TeleSUR*. [En línea] 2010. <http://www.ecured.cu/index.php/Telesur>.
7. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education, 2000.
8. —. *El Lenguaje unificado de modelado. Manual de Referencia*. Madrid : Pearson Educación, 2000.
9. Scribd. *Capítulo I HERRAMIENTAS CASE*. [En línea] 2011. [Citado el: 09 de diciembre de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
10. Sitio de descargas de software. *Paradigma Visual para UML*. [En línea] [Citado el: 10 de diciembre de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
11. Definición.org. *Definición de Lenguaje de Programación*. [En línea] [Citado el: 10 de diciembre de 2011.] <http://www.definicion.org/lenguaje-de-programacion>.
12. PHP. *Manual de PHP*. [En línea] <http://docs.php.net/manual/es/preface.php>.
13. **Potencier, Fabien**. *Symfony*. [En línea] Sensio Labs. <http://symfony.com/symfony-at-a-glance>.

14. **Potencier, Fabien y Zaninotto, François.** *The Definitive Guide to Symfony*. s.l. : Apress, 2007.
15. **Negrino, Tom y Smith, Doris.** *Guía de aprendizaje JavaScript*. 3ra. Madrid : Pearson Education, 2000.
16. JQuery. *Write less, do more*. [En línea] <http://jquery.com/>.
17. **Alvarez, Sara.** DesarrolloWeb.com. *Introducción a este concepto y características especiales*. [En línea] 31 de julio de 2007. [Citado el: 27 de 04 de 2012.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
18. **Martinez, Rafael.** Portal en español sobre PostgreSQL. *Sobre PostgreSQL*. [En línea] 02 de 02 de 2010. [Citado el: 28 de 04 de 2012.] http://www.postgresql.org.es/sobre_postgresql.
19. Ciberaula. *Una Introducción a APACHE*. [En línea] 2010. http://linux.ciberaula.com/articulo/linux_apache_intro/.
20. **Craig, Larman.** *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. Mexico : Pearson, 1999.
21. Thiyagarajan Veluchamy. *IEEE Standard Glossary of Software Engineering Terminology 610*. [En línea] [Citado el: 25 de enero de 2012.] <http://thiyagarajan.wordpress.com/glossary/>.
22. **Pressman, Roger. S.** *Ingeniería del Software. Un enfoque práctico*. 6ta. s.l. : McGraw-Hill, 2007.
23. **Sommerville, Ian.** *Ingeniería del Software*. Madrid : Pearson Educación, 2002.
24. **Bahit, Eugenia.** *Programación Orientada a Objetos y Modelo Vista Controlador en PHP*.
25. Geek the planet. [En línea] . <http://geektheplanet.net/5462/patrones-gof.xhtml>.
26. **Fraser, Dustin Edward.** Scribd. [En línea] 2012. [Citado el: 5 de mayo de 2012.] <http://es.scribd.com/doc/74894700/44/CAPITULO-4-IMPLEMENTACION-Y-PRUEBA-DEL-SISTEMA>.
27. **Eguiluz, Javier.** *Desarrollo Web Ágil con Symfony2*. 2011.
28. **Meléndrez Hernández, Edelsys.** *Cómo escribir una tesis*. 2006.

BIBLIOGRAFÍA CONSULTADA

1. Diccionario de la Real Academia Española. *DRAE*. [En línea] Vigésima Edición, Real Academia Española, 2011. http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=m%C3%B3dulo.
2. Alegsá. *Portal de informática, internet, tecnologías y web*. [En línea] ALEGSA, 2011. www.alegsa.com.ar/Dic/modulo.php.
3. **Hernández García, Ruber, y otros**. *Primicia, una Plataforma de Televisión sobre Tecnologías Libres*.
4. **Dirección de Televisión Universitaria**. *Inter-Nos*. [En línea] <http://inter-nos.uci.cu/canalcultural/cc.asp>.
5. Enciclopedia Cubana en la Red (EcuRed). *Canal Multivisión*. [En línea] 2010. <http://www.ecured.cu/index.php/Multivisi%C3%B3n>.
6. Enciclopedia Cubana en la Red (EcuRed). *TeleSUR*. [En línea] 2010. <http://www.ecured.cu/index.php/Telesur>.
7. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education, 2000.
8. —. *El Lenguaje unificado de modelado. Manual de Referencia*. Madrid : Pearson Educación, 2000.
9. Scribd. *Capítulo I HERRAMIENTAS CASE*. [En línea] 2011. [Citado el: 09 de diciembre de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
10. Sitio de descargas de software. *Paradigma Visual para UML*. [En línea] [Citado el: 10 de diciembre de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
11. Definición.org. *Definición de Lenguaje de Programación*. [En línea] [Citado el: 10 de diciembre de 2011.] <http://www.definicion.org/lenguaje-de-programacion>.
12. PHP. *Manual de PHP*. [En línea] <http://docs.php.net/manual/es/preface.php>.
13. **Potencier, Fabien**. *Symfony*. [En línea] Sensio Labs. <http://symfony.com/symfony-at-a-glance>.

14. **Potencier, Fabien y Zaninotto, François.** *The Definitive Guide to Symfony*. s.l. : Apress, 2007.
15. **Negrino, Tom y Smith, Doris.** *Guía de aprendizaje JavaScript*. 3ra. Madrid : Pearson Education, 2000.
16. JQuery. *Write less, do more*. [En línea] <http://jquery.com/>.
17. **Alvarez, Sara.** DesarrolloWeb.com. *Introducción a este concepto y características especiales*. [En línea] 31 de julio de 2007. [Citado el: 27 de 04 de 2012.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
18. **Martinez, Rafael.** Portal en español sobre PostgreSQL. *Sobre PostgreSQL*. [En línea] 02 de 02 de 2010. [Citado el: 28 de 04 de 2012.] http://www.postgresql.org.es/sobre_postgresql.
19. Ciberaula. *Una Introducción a APACHE*. [En línea] 2010. http://linux.ciberaula.com/articulo/linux_apache_intro/.
20. **Craig, Larman.** *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. Mexico : Pearson, 1999.
21. Thiyagarajan Veluchamy. *IEEE Standard Glossary of Software Engineering Terminology 610*. [En línea] [Citado el: 25 de enero de 2012.] <http://thiyagarajan.wordpress.com/glossary/>.
22. **Pressman, Roger. S.** *Ingeniería del Software. Un enfoque práctico*. 6ta. s.l. : McGraw-Hill, 2007.
23. **Sommerville, Ian.** *Ingeniería del Software*. Madrid : Pearson Educación, 2002.
24. **Bahit, Eugenia.** *Programación Orientada a Objetos y Modelo Vista Controlador en PHP*.
25. Geek the planet. [En línea] . <http://geektheplanet.net/5462/patrones-gof.xhtml>.
26. **Fraser, Dustin Edward.** Scribd. [En línea] 2012. [Citado el: 5 de mayo de 2012.] <http://es.scribd.com/doc/74894700/44/CAPITULO-4-IMPLEMENTACION-Y-PRUEBA-DEL-SISTEMA>.
27. **Eguiluz, Javier.** *Desarrollo Web Ágil con Symfony2*. 2011.
28. **Meléndrez Hernández, Edelsys.** *Cómo escribir una tesis*. 2006.

GLOSARIO

AJAX: Proviene de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

API: Proviene de Application Programming Interface (Interfaz de Programación de Aplicaciones), es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Arquitectura: Conjunto de patrones y abstracciones coherentes que guían la construcción del software informático.

BBC: Proviene de British Broadcasting Corporation (Corporación Británica de Radiodifusión), es el servicio público de comunicación del Reino Unido. Sus funciones son las de transmitir televisión y radio de forma pública.

CASE: Proviene de Computer-Aided Software Engineering (Ingeniería de Software Asistida por Ordenador). Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CNN: Proviene de Cable News Network (Cadena de Noticias por Cable), es un canal de televisión de noticias operado por Time Warner que transmite las 24 horas del día.

CSS: Proviene de Cascading Style Sheets (Lenguaje de Hojas de Estilos), creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

Discovery Channel: Es un canal de televisión por cable propiedad de Discovery Communications, una señal de entretenimiento, cultura y educación distribuida en todo el mundo. El canal es conocido principalmente por su variedad de programación científica, particularmente documentales y programas sobre la naturaleza.

Disney Channel: Es un canal internacional de The Walt Disney Company que emite en gran parte del mundo series originales. La programación del canal está destinada al público juvenil.

DOM: Proviene de Document Object Model (Modelo de Objeto de Documento), es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GOF: Proviene de Gang of Four (Grupo de los Cuatro o Banda de los Cuatro), se refiere a los patrones de diseño en el campo del diseño orientado a objetos.

GRASP: Proviene de General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignación de Responsabilidades), son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

Hardware: Conjunto de los componentes que integran la parte material de una computadora. Los elementos tangibles.

IDE: Proviene de Integrated Development Environment (Entorno de Desarrollo Integrado), 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.

Lenguaje interpretado: Lenguaje de programación diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

Página web dinámica: Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Plugin: Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

RUP: Proviene de Rational Unified Process (Proceso Unificado de Desarrollo de Software), es una metodología que busca mejorar las practicas que se implementan en el desarrollo de software.

Software: Equipamiento lógico o soporte lógico de una computadora digital, o sea el conjunto intangible de datos y programas.

Subsistema: Agrupación de componentes que hace parte de un sistema.

Switch: Una caja o interruptor con una o varias palancas que manipulan la posición de las señales transmitidas por un mismo canal.

UML: Proviene de Unified Modeling Language (Lenguaje Unificado de Modelado), es un lenguaje, que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, se centra en la representación gráfica de un sistema.

Univisión: Es la cadena de televisión en español más grande de los Estados Unidos. Univisión ofrece una variedad de novelas, películas, shows, deportes y televisión general en español.

XML: Proviene de Extensible Markup Language (Lenguaje de Marcas Extensible). Se trata de un metalenguaje extensible de etiquetas, no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades.