

Universidad de las Ciencias Informáticas

Facultad 6



**Título: “Implementación del Módulo de Personalización de la
Interfaz de Usuario de la
Plataforma de Televisión Informativa Primicia”**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Dagmar Fernández Cuesta

Tutor:

Ing. Lisandra Delgado Cabrera

La Habana, Junio 2011

“Año 53 de la Revolución”

Frase

“Cuando las puertas de la percepción sean depuradas todo se reflejará ante el hombre tal cual es - infinito.”

Jim Morrison

Declaración de Autoría

Yo Dagmar Fernández Cuesta declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Dagmar Fernández Cuesta

Ing. Lisandra Delgado Cabrera

Autor

Tutor

Datos de contacto

Tutor: Ing. Lisandra Delgado Cabrera

Graduado de Ingeniero en Ciencias Informáticas

e-mail: ldcabrera@uci.cu

Cargo: Profesor recién graduado en adiestramiento de la Universidad de la Ciencias Informáticas.

Dedicatoria

A mis padres por dejarme ver un mundo en un grano de arena,
a mis amigos por tener el infinito en la palma de la mano
y a ti por la eternidad en una hora.

Agradecimientos

A los primeros que quiero darles más que agradecimientos es a mis padres por ser tan geniales, a mi papá que me enseñó a valorar el conocimiento y a mi mamá por sus consejos de la vida...a mi hermano...a toda mi familia por preocuparse y ocuparse. Gracias a las amistades de toda una vida; aquellos que ya no están acá pero igual quedan en la memoria de lo que fue y no se olvidará, a los compañeros y compañeras de aula durante estos cinco años de estudio; a las que me soportaron más de cerca y a todas las otras amistades que conocí dentro y fuera de la UCI, un puño para ustedes; un grupo que no puedo dejar de mencionar por resistirme tanto, por todos los momentos y ayudas: inmensidad de gracias a May, Su, Naivy, Ana Marys, Osvaldo, Ariel, Boris, Orlando, Yasmany, Knito, Valdevila y Alexander.

Gracias a Lisandra, Estela, Felix, Rafael, a todos los tésistas de Primicia, al tribunal de tesis... gracias a Cuba y lo que lleva dentro, gracias a la música, a los libros y a google, en general a los que no fueron "another brick in the wall".

Resumen

La humanidad en su constante evolución desarrolló una nueva forma de comunicación, la televisión. Gracias al surgimiento de las redes informáticas y al uso de Internet en el mundo de las noticias se logra digitalizar la información y a partir de este y otros avances se crearon los llamados Sistemas de Producción de Noticias. En Cuba se han creado software con similares características, entre ellos "Primicia", desarrollado por un equipo de trabajo de la Universidad de las Ciencias Informáticas (UCI).

La plataforma Primicia se desarrolló siguiendo la idea de contar con un producto informático que permita la administración y transmisión de noticias para un canal de televisión informativa. Haciendo uso de las redes informáticas se puede utilizar una mayor cantidad de información que puede estar en varios formatos: texto, video, imagen y audio.

El presente trabajo surge a partir de la necesidad de obtener una interfaz de usuario completamente configurable y adaptable para la plataforma. Las funcionalidades que se desarrollen deben modificar con rapidez y facilidad todo el esquema del propio sistema. El nuevo cliente obtiene un producto acorde a sus necesidades, sin precisar la intervención directa de algún programador.

El objetivo general de la investigación es la implementación del módulo de personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. Con ello se obtendrán mejoras en el proceso de administración que van a permitir la creación de interfaces totalmente configurables. En el presente trabajo se plasma todo el proceso de personalización y de implementación de dicho módulo.

Palabras Clave: implementación, interfaz de usuario, Primicia, personalización, televisión informativa.

Tabla de Contenidos

Dedicatoria.....	I
Agradecimientos	II
Resumen	I
Introducción	2
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1 Aplicación Web.	6
1.2.2 Interfaz de usuario.	6
1.2.3 Personalizar.	7
1.2.4 Televisión informativa.....	7
1.3 Descripción general.....	7
1.3.1 Plataforma de Televisión Informativa Primicia.	7
1.4 Soluciones o sistemas similares existentes a nivel mundial.	9
1.4.1 Estructure.....	9
1.4.2 Sistemas de Teletexto en Europa.....	10
1.4.3 Plataformas de televisión sobre IP (IPTV).....	11
1.5 Situación nacional.	11
1.6 Análisis de la solución.	12
1.6.1 Descripción de términos para la personalización.....	13
1.6.2 Procesos relacionados con la personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. Características generales.....	14
1.6.3 Elementos configurables de la plantilla de transmisión.....	15
1.7 Conclusiones parciales.....	17
Capítulo 2. Tendencias y tecnologías actuales	18
2.1 Introducción.....	18

2.2	Metodologías de desarrollo.	18
2.2.1	Proceso Unificado de Desarrollo (RUP).	19
2.3	Lenguaje Unificado de Modelado (UML).	21
2.4	Herramienta de modelado.	21
2.4.1	Visual Paradigm 6.4.	22
2.5	Lenguajes de programación del lado del cliente.	23
2.5.1	HTML.	23
2.5.2	CCS.	23
2.5.3	JavaScript.	24
2.6	Lenguaje de programación del lado del servidor.	24
2.6.1	PHP.	24
2.7	Framework.	25
2.7.1	Framework Symfony 1.4.6.	25
2.7.2	Framework JavaScript.	26
2.7.3	ExtJS 3.0.	27
2.8	Entorno de Desarrollo Integrado.	28
2.8.1	NetBeans 6.9.	29
2.9	Firebug 1.7.0.	29
2.10	Conclusiones parciales.	30
Capítulo 3. Construcción de la solución		31
3.1	Introducción.	31
3.2	Estándar de codificación.	31
3.2.1	Criterios de calidad.	31
3.2.2	Nomenclatura para el nombramiento de elementos.	31
3.2.3	Nomenclatura de comentarios.	32

3.2.4	Nomenclatura de nombres de identificadores.....	32
3.2.5	Nomenclatura de identificadores de variables.	33
3.2.6	Otros aspectos del código.	33
3.3	Requerimientos funcionales.	35
3.4	Requerimientos no funcionales.	37
3.4.1	Apariencia o interfaz externa.....	37
3.4.2	Disponibilidad.....	37
3.4.3	Usabilidad.	37
3.4.4	Soporte.	37
3.4.5	Restricciones en el diseño y la implementación.	38
3.5	Modelo de implementación.....	38
3.6	Diagramas de componentes.....	39
3.7	Diagrama de despliegue.....	49
3.8	Conclusiones parciales	50
4.1	Introducción.....	51
4.2	Diseño de casos de prueba.....	51
4.2.1	CUS: Gestionar Plantilla de Transmisión.....	51
4.2.3	CUS: Gestionar Infocinta.....	58
4.2.4	CUS: Administrar Tema de Transmisión.	61
4.2.5	CUS: Administrar Formato de Fecha y Hora.	62
4.2.6	CUS: Seleccionar Plantilla de Administración.....	63
4.3	Conclusiones parciales.....	63
Conclusiones		64
Recomendaciones		65
Bibliografía.....		66

Referencias bibliográficas 68

Glosario de términos..... 70

Índice de ilustraciones

Ilustración 1 Salto de línea	33
Ilustración 2 Salto de línea después de punto y coma	34
Ilustración 3 Modelo de implementación.....	39
Ilustración 4 Diagrama de componentes-CU Gestionar infocinta	40
Ilustración 5 Diagrama de componentes-CU Gestionar plantilla de transmisión	41
Ilustración 6 Diagrama de componentes-CU Administrar iconografía	43
Ilustración 7 Diagrama de componentes-CU Administrar recursos visuales	44
Ilustración 8 Diagrama de componentes-CU Establecer formato fecha/hora	45
Ilustración 9 Diagrama de componentes-CU Gestionar panel informativo.....	46
Ilustración 10 Diagrama de componentes-CU Administrar elementos de identificación.....	47
Ilustración 11 Diagrama de componentes-CU Seleccionar plantilla de administración	48
Ilustración 12 Diagrama de despliegue	49

Índice de tablas

Tabla 1 Términos de la personalización	14
Tabla 2 Nombramiento de elementos	32
Tabla 3 CUS Gestionar Plantilla de Transmisión	55
Tabla 4 CUS Gestionar Elementos	58
Tabla 5 CUS Gestionar Infocinta.....	61
Tabla 6 CUS Administrar Tema de Transmisión	62
Tabla 7 CUS Administrar Formato de Fecha y Hora	63
Tabla 8 CUS Seleccionar Plantilla de Administración.	63

Introducción

Introducción

El hombre en su evolución siempre ha tenido la necesidad de informarse y comunicarse, a pesar de las grandes distancias que lo separen del resto de la humanidad. En la actualidad la comunicación es un punto muy importante en nuestras vidas para conocer sobre la situación política, económica y social que enfrenta nuestro alrededor y el resto del mundo.

La comunicación no es más que el intercambio de información de un lugar a otro, donde la información que se transmite es el llamado mensaje. Cualquier sistema que sea usado para establecer comunicación está diseñado para contar con un emisor, un mensaje con el canal a través del cual se envía y el receptor. (Revista RED, 2003)

En el año 1794 se transmite el primer telegrama de la Historia desde Lille a París, a lo largo de 230 kilómetros y 22 torres. El artefacto utilizado llamado telégrafo fue desde el punto de vista tecnológico el primer sistema de comunicación del ser humano. A partir de él, se llegó a crear desarrollados sistemas de telecomunicación con el uso de señales eléctricas, electromagnéticas u ópticas.

Estas transmisiones pueden ser utilizando dos formas distintas:

- un medio físico: cables para transmitir señales eléctricas o mediante fibra óptica para transmitir señales de luz (la televisión, el teléfono, redes de ordenadores por cable, etc.);(Revista RED, 2003)
- sin medio físico -inalámbricos-: Las señales se envían a través del aire mediante ondas electromagnéticas (la radio, redes wifi, bluetooth, etc.).(Revista RED, 2003)

El sistema de comunicación más expandido y utilizado, gracias a su fácil acceso y capacidad para la transmisión y recepción de imágenes en movimiento y sonido a distancia, es la televisión. Si las ondas enviadas son analógicas se le conoce como televisión analógica; si son digitales sería televisión digital.

Haciendo uso de las redes informáticas ocurre una nueva evolución y se crean aplicaciones informáticas que administran, conforman y transmiten las noticias, llamadas Sistemas de Producción de Noticias. Como ejemplo de estos sistemas están **SONAPS** (EEUU), **Estructure Enewsroom** (España) y **Primicia** (Cuba), este último desarrollado en la Universidad de las Ciencias Informáticas.

Introducción

Antes de crear la Plataforma de Televisión Informativa Primicia, en la UCI se desarrollan varias soluciones similares pero con objetivos particulares para cada una. La llamada **Señal 3** es utilizada para mantener informada a toda la comunidad universitaria dentro del centro de estudios. Con el objetivo principal de hacer llegar las noticias y otras informaciones de Cuba y el mundo a los colaboradores internacionalistas que cumplen misión en el extranjero, se creó **Señal ACN** como solución para la Agencia Cubana de Noticias. Otra de las aplicaciones es **TVEnergía**, plataforma creada para el Ministerio del Poder Popular para la Energía y el Petróleo de Venezuela.

Después de realizar las mencionadas soluciones se constituye una plataforma integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos (texto, imagen, audio y video). Dicho producto informático se crea por el Departamento de Señales Digitales del Centro de Desarrollo GEYSED¹ de la Facultad 6.

Este sistema es desarrollado y soportado completamente en software libre, contribuyendo de esta manera a la soberanía tecnológica del país. Surge por la necesidad de transmitir información en tiempo real, con el objetivo de mantener informados de manera rápida a trabajadores, usuarios o personal en general. Tal es el caso de universidades, centros de convenciones, hoteles, terminales de transportación y sedes ministeriales o empresariales, en las que existen circuitos cerrados de televisión.

En los sistemas de producción de noticias que se desarrollan y comercializan hoy en día es fundamental tratar de ajustar el diseño de la aplicación según la identidad gráfica² definida por la institución, para así lograr un producto final más cercano al medio donde se va a utilizar y a las exigencias propias del cliente. El presente trabajo de diploma consiste en el desarrollo de un informe de investigación práctico del rol de implementador dentro del proyecto Primicia para la personalización de los subsistemas de transmisión y administración.

Actualmente el producto Primicia no cuenta con un conjunto de funcionalidades que permitan la personalización de la interfaz de usuario de los distintos subsistemas que la componen. Cualquier cambio necesario a la hora de comercializar el producto con diferentes empresas o instituciones tiene

¹Geoinformática y Señales Digitales.

² *Concepto que responde visualmente ¿Quién soy? ¿Cómo soy? ¿Cuales son mis rasgos exclusivos?* (Agencia de diseño y comunicación gráfica, 2009)

Introducción

que ser implementado de forma manual por el equipo de desarrollo, o sea, no se le permite al cliente realizar los cambios en la interfaz de forma automática.

La preocupación de reunir al equipo de implementadores siempre que haya que adaptar el diseño de la interfaz según las exigencias del nuevo cliente con quien se vaya a comercializar, hace que el presente trabajo de diploma busque darle solución al siguiente **problema**: necesidad de personalizar la interfaz de usuario de los subsistemas de administración y transmisión de la Plataforma de Televisión Informativa Primicia.

Para el desarrollo de la investigación se define como **objeto de estudio**: los procesos relacionados con la personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. El **campo de acción** se define como la personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia.

Se propone como **objetivo general**: implementar el módulo de personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. Se plantea como guía de la investigación la siguiente **idea a defender**: con el desarrollo de un módulo que permita la personalización del sistema se logrará modificar la Plataforma de Televisión Informativa Primicia con una mayor adaptación al usuario final.

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

- 1) Caracterizar los procesos relacionados con la personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia.
- 2) Describir el estado del arte asociado a la solución planteada.
- 3) Caracterizar las tecnologías a utilizar para el desarrollo del módulo de personalización de la interfaz de usuario de Primicia.
- 4) Definir un estándar de codificación.
- 5) Modelar los diagramas correspondientes al rol implementador durante el desarrollo del módulo de personalización de la interfaz de usuario de Primicia.
- 6) Implementar los casos de uso propuestos por el analista en el diseño del módulo.

Introducción

7) Validar la solución propuesta realizando pruebas al módulo.

Este trabajo tiene como base el uso de diferentes métodos de investigación científica para garantizar la calidad de todo el proceso de estudio e investigación usando métodos tanto teóricos como empíricos.

Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observadas directamente. De ellos se utiliza el **Analítico – Sintético** para concretar los elementos más importantes relacionados con el proceso de personalización de la interfaz de usuario que necesita la plataforma Primicia. Se realizó el análisis de un gran volumen de documentación, que permitió sintetizar el contenido que da soporte a la propuesta que se realiza en el presente trabajo de diploma.

Otro método presente es el **Análisis Histórico – Lógico**, pues es necesario estudiar todos los procesos de personalización que presentan diversos sistemas informativos para encontrar las características más idóneas que pueden ser incorporadas en el módulo de personalización de Primicia y la valoración de las funcionalidades que presentan.

Los métodos empíricos permiten la obtención y elaboración de los datos empíricos y el conocimiento de los hechos fundamentales que caracterizan a los fenómenos y para ello se utiliza como principal procedimiento **la observación**.

Como parte de la estructura del documento se abordará en el primer capítulo sobre los principales conceptos, descripción general y estado del arte. En el segundo se expondrán las herramientas y lenguajes a utilizar para el desarrollo del módulo. Para lograr una correcta implementación se describirá en el capítulo 3 la nomenclatura a usar por el estándar de codificación y la conformación de los distintos diagramas a desarrollar como parte del flujo. En el capítulo 4 y final se describirán las pruebas hechas al módulo para así comprobar su correcto funcionamiento.

Capítulo 1: Fundamentación Teórica

Capítulo 1. Fundamentación Teórica

1.1 Introducción.

En este capítulo se explica un conjunto de conceptos relacionados con el objetivo general del trabajo de diploma y con el problema planteado. Además se describe de manera detallada la situación problemática expuesta y los procesos de la plataforma Primicia que tienen que ver con la personalización de la interfaz de usuario de la misma.

Para un mayor entendimiento del trabajo realizado se hace necesario el dominio de algunos conceptos tratados durante el desarrollo de la investigación a los cuales se hace referencia a continuación.

1.2 Conceptos asociados al dominio del problema.

1.2.1 Aplicación Web.

Un software donde los usuarios acceden a un servidor web a través de Internet o de una red local utilizando un navegador, es llamado aplicación web. Dicha aplicación es ejecutada totalmente por parte del propio navegador y para esto la web es codificada en un lenguaje soportado por ellos, como por ejemplo: HTML, JavaScript, Java, etc. (Hooping Publicidad S.L, 1995-2008)

Las aplicaciones web han demostrado su gran efectividad gracias al bajo consumo que hace de los recursos del ordenador, son multiplataforma, no ocupan espacio en nuestros discos duros, actualizaciones inmediatas al ser el propio desarrollador el que gestiona el software, además de lo ligero y práctico que son los navegadores.

Una Web no requiere ser instalada y permite que el cliente acceda a los datos de modo interactivo ya que puede contener elementos que permiten una comunicación activa entre el usuario y la información. La página gestiona cada petición realizada por éste sin que el mismo tenga que conocer elementos como su código o el lenguaje de programación usado para la confección de la misma.

1.2.2 Interfaz de usuario.

La interfaz de usuario (IU) representa la información y las acciones disponibles en la interfaz utilizando un conjunto de imágenes y objetos gráficos. Surge de la necesidad de hacer que los sistemas sean

Capítulo 1: Fundamentación Teórica

más accesibles para los usuarios comunes proporcionando un entorno visual sencillo para permitir una fácil comunicación. (UMSNH, 2000)

Es uno de los elementos más significativos de cualquier sistema computacional, pues actúa como el lazo entre el humano y la máquina. Su relevancia se basa en que la mayoría de las veces el éxito de un software se debe a qué tan rápido puede aprender el usuario a emplear el software, de igual importancia es el que el usuario alcance sus objetivos con el programa de la manera más sencilla posible.

La IU es la encargada de exigir comandos al usuario y de mostrar los resultados de la aplicación de una forma entendible, pero a la vez no es responsable de los cálculos de la aplicación, ni del almacenamiento, recuperación y transmisión de la información. Para tener el control de estas interfaces se tiene el uso de dispositivos de entrada a la computadora, como el teclado y el mouse mediante menús, ventanas, íconos y cajas de diálogo.

1.2.3 Personalizar.

Según la Real Academia Española personalizar es dar carácter personal a algo, incurrir en personalidades hablando o escribiendo. Aplicada a la informática en el ámbito de la web se entiende como el hecho de que los usuarios modifiquen o personalicen la interfaz de usuario y el comportamiento de determinadas aplicaciones web. La personalización permite guardar las propiedades o el estado de los controles de elementos web en almacenamiento de larga duración como en una base de datos.

1.2.4 Televisión informativa.

La televisión informativa es una señal de televisión basada en la actualidad informativa. Se trata de brindar información noticiosa de los acontecimientos más sobresalientes del día a los agentes sociales (individuos, grupos e instituciones). Estas noticias se dividen en bloques temáticos o secciones, asegurando el equilibrio en el interés de cada usuario y comenzando siempre con la noticia más importante del día en ese ámbito.

1.3 Descripción general.

1.3.1 Plataforma de Televisión Informativa Primicia.

Capítulo 1: Fundamentación Teórica

En el Centro GEYSED se desarrolla una aplicación web con el uso de herramientas de software libre que tiene como meta proporcionar un canal de televisión que transmita automática y constantemente informaciones en variados formatos acompañado de un fondo musical.

Como plataforma, Primicia está implementado para que pueda ser aplicable a distintos clientes, por lo que constituye un producto informático del cual pueden beneficiarse clientes que tengan una red de televisión con necesidades de transmitir informaciones en distintos formatos como texto, imagen, texto-imagen y video.

Está estructurado en dos subsistemas:

- **Subsistema de Administración:** administrar el canal y toda la gestión de las noticias;
- **Subsistema de Transmisión:** visualizar las noticias y materiales publicados.

El Subsistema de Administración tiene las siguientes prestaciones generales:

- ✓ Gestión de los usuarios del sistema, permite registrar y eliminar usuarios, así como establecer o modificar los permisos de acceso en el sistema.(Hernández García, y otros, 2008)
- ✓ Gestión de las sesiones temáticas del canal, establece el orden de las secciones, horario en que serán mostradas y permite habilitarlas o deshabilitarlas.(Hernández García, y otros, 2008)
- ✓ Funcionalidades para la redacción de noticias según los formatos definidos para las pantallas; la publicación de las noticias teniendo en cuenta fecha de inicio y fin de la circulación; la gestión de las noticias del canal para modificar, eliminar y archivar las noticias; la administración del archivo de noticias del canal, que permite reutilizar y eliminar las mismas.(Hernández García, y otros, 2008)
- ✓ Almacenamiento, administración y reproducción de recursos multimedia como imágenes, música y video.(Hernández García, y otros, 2008)
- ✓ Funcionalidades para la creación y administración de cintillos informativos o infocintas. La administración de los cintillos incluye establecer orden de prioridad de muestra y la habilitación o deshabilitación de los mismos.(Hernández García, y otros, 2008)

Capítulo 1: Fundamentación Teórica

- ✓ Generación de reportes sobre la actividad del sistema. Los reportes se realizan sobre la actividad de los trabajadores del sistema, realizar búsquedas de noticias publicadas atendiendo distintos criterios como fecha de publicación, temática, palabras claves y título. Ofrece facilidades para la impresión de los reportes y la exportación de los reportes a formato digital.(Hernández García, y otros, 2008)
- ✓ Administración de la señal del canal que permite cambiar entre la señal de televisión en vivo y la señal del canal.(Hernández García, y otros, 2008)

El Subsistema de Transmisión tiene las siguientes prestaciones generales:

- ✓ Generación de una cartelera del ciclo de transmisión, mostrando para cada noticia la sección temática y el titular, en el orden que se van a visualizar.(Hernández García, y otros, 2008)
- ✓ Visualización de noticias compuestas por pantallas de tipo texto, texto-imagen, imagen y video.(Hernández García, y otros, 2008)
- ✓ Reproducción de un fondo musical mientras se muestran las noticias, excepto cuando se muestre un video.(Hernández García, y otros, 2008)
- ✓ Presentación en las pantallas de tipo imagen un comentario que oriente al televidente acerca de lo que está observando.(Hernández García, y otros, 2008)

Para lograr tener una plataforma lo más genérica y amigable posible urge la necesidad de conocer a fondo los procesos de redacción de los principales Sistemas de Producción de Noticias. Teniendo en cuenta los avances actuales y las tendencias en este campo en el ámbito internacional y nacional, se logrará una mejor manera de crear un proceso de personalización lo mas extensible posible.

1.4 Soluciones o sistemas similares existentes a nivel mundial.

En el mundo, el desarrollo tecnológico ha logrado unir el uso de la televisión con las redes informáticas, logrando de esta forma la creación de sistemas de transmisión rápida de información sobre un sistema web automático. Ejemplos de estos sistemas de noticias son:

1.4.1 Estructure.

Capítulo 1: Fundamentación Teórica

Estructure es una empresa de desarrollo de sistemas para la producción y gestión audiovisual que proporciona soluciones tecnológicas avanzadas para la explotación de activos audiovisuales. Los usuarios comparten los archivos de media y el trabajo mediante procesos de gestión ágiles y sencillos, creando un entorno único, flexible y compartido para la producción audiovisual.

Dicha empresa tiene desarrollado un completo sistema de producción de noticias y contenidos audiovisuales diseñado para facilitar el trabajo de periodistas y usuarios llamado Enewsroom. En un entorno único de trabajo, el sistema integra todas las herramientas necesarias para explotación de noticias, en el que no existen transferencias ni tiempos de espera. Sin embargo este software no permite cambios en la interfaz, solamente cuenta con la funcionalidad básica del cambio de idioma.

1.4.2 Sistemas de Teletexto en Europa.

El primer canal en incorporar el teletexto fue la British Broadcasting Corporation (BBC³) en 1972. Más tarde, países como Austria, Alemania y Holanda también comenzaron a utilizar este singular sistema. El principal objetivo de este sistema, además de informar, consistía en prestar un servicio a las personas con discapacidad auditiva, de este modo comenzaron a crearse subtítulos para los principales programas de las emisoras. Gracias a este dispositivo se crearon parte de los actuales productos multimedia.

Consiste en un conjunto de páginas que se eligen a través del mando a distancia. Para transmitir información escrita y en gráficos se teclea el número de la página y después de un tiempo se visualizará en la pantalla del televisor.

En los sistemas de teletexto el usuario se comporta como un espectador ya que únicamente puede seleccionar las informaciones que desea a partir de las que se le ofrecen. El número de páginas de información está limitado porque el teletexto sólo utiliza ocho líneas de imágenes y por tanto la información queda limitada al espacio de la pantalla del televisor. Estos sistemas no brindan personalización alguna ya que no hay comunicación con la fuente de la información y por tanto no puede realizar una petición concreta.

³ Corporación Británica de Radiodifusión

Capítulo 1: Fundamentación Teórica

1.4.3 Plataformas de televisión sobre IP⁴ (IPTV).

Los sistemas de televisión sobre IP están basados en la recepción de canales digitales tanto terrestres como vía satélite, para luego distribuirlos a través de cualquier red IP. En un sistema IPTV un usuario es capaz de acceder y generar informaciones, simultáneamente y en tiempo real.

Pero en un sistema de este tipo la adaptabilidad de la interfaz de usuario tiene lugar solamente en la personalización de la programación que el usuario desea, permitiendo únicamente ordenar la lista de reproducción y configurar su perfil de chat y no alcanza la profundidad de lo deseado por el equipo de desarrollo del producto Primicia según el proceso de caracterización realizado por el mismo.

1.5 Situación nacional.

Producto de la Batalla de Ideas desarrollada en el país se creó la Universidad de las Ciencias Informáticas. Desde sus inicios contaba con una inmensa masa de estudiantes y profesores a los cuales era necesario hacerles llegar informaciones de ámbito nacional e internacional, así como noticias propias de la universidad. Por este motivo y gracias a las tareas investigativas y productivas de los integrantes del proyecto **UCITeVe**⁵, se crea un sistema informático para dar solución a la transmisión de canales de televisión utilizando computadoras llamado **Señal 3**, canal de televisión interno de la universidad que surge en el año 2005.

En el año 2006 se desarrolla con similares características el canal informativo de la Agencia Cubana de Noticias: **Señal ACN**. Dicho sistema automatizado de teletexto para la Plataforma de Televisión Satelital Cubana, facilita la transmisión de noticias de la prensa cubana a los colaboradores internacionalistas que cumplen misión en el extranjero y a los habitantes de las llamadas zonas de silencio de Cuba. Para la realización de estas soluciones se emplea software propietario disponible solo para sistemas operativos de la familia Windows, por ejemplo SQL Server como gestor de base de datos y Macromedia Director como entorno de desarrollo.

Gracias a una importante negociación con un cliente que deseaba un producto similar pero desarrollado con herramientas no propietarias, nace **TVEnergía**. Como resultado del trabajo y el estudio sobre la línea de investigación de sistemas informáticos para la transmisión de radio y

⁴ Internet Protocol

⁵ Proyecto de la televisión universitaria

Capítulo 1: Fundamentación Teórica

televisión sobre plataformas libres, se logra la integración de herramientas libres como PostgreSQL, Apache HTTP Server, ProFTPD, VideoLAN y Mozilla Firefox en el entorno de desarrollo Eclipse, utilizando como sistema operativo la distribución de Nova/Linux para estaciones de trabajo. Aún con estos resultados, el equipo de desarrollo continúa realizando investigaciones con el objetivo de incrementar funcionalidades y lograr optimizaciones del sistema (Bermúdez Valdés, 2010)

Después de conformada **TVenergía** y tomando las ventajas de la utilización de herramientas libres se da inicio a la conceptualización de la Plataforma de Televisión Informativa Primicia. Está diseñada y elaborada también sobre plataformas y herramientas totalmente libres como: Ubuntu o Nova, Visual Paradigm, Symfony, PostgreSQL, PHP, JavaScript y HTML. La creación de la misma se enfocó hacia la obtención de funcionalidades fácilmente escalables, que no dependan de un entorno dado y no se encuentren atadas a un diseño gráfico específico.

No obstante, una de las principales desventajas identificadas es que la interfaz de usuario no es personalizable. De manera que es necesario reunir al equipo de trabajo en caso de que se quiera comercializar el producto con un nuevo cliente y este desee como es de suponer una interfaz que lo represente.

1.6 Análisis de la solución.

En los sistemas web los aspectos como el diseño gráfico, la calidad cognoscitiva o la facilidad de uso, tienen tanta relevancia como el propio diseño técnico del sistema. La facilidad de migración de los usuarios a otras aplicaciones que ofrecen servicios en la web, que son fácilmente accesibles a través de la red y por otro lado a la audiencia heterogénea de este tipo de aplicaciones han provocado que la personalización se haya convertido en un elemento significativo del diseño gráfico y aporta un valor agregado a un contenido que debe además ser accesible y estar actualizado.

La creciente necesidad de analizar el comportamiento de los usuarios de un sitio web con el fin de aumentar su fidelidad, ha desencadenado la aparición del concepto de personalización que hace referencia a cómo una compañía reconoce, comprende y satisface las necesidades individuales de cada cliente sobre el conocimiento de históricos comportamientos en sesiones y preferencias, hábitos o gustos manifestados de forma expresada por el propio cliente como respuestas a sus necesidades.

La identificación temprana de las necesidades de personalización disminuye el coste de modificación de las aplicaciones y supone un factor importante para el éxito en la implantación de las mismas.

Capítulo 1: Fundamentación Teórica

1.6.1 Descripción de términos para la personalización.

<u>Término</u>	<u>Descripción</u>
Elementos de identificación	Conjunto de elementos propios de cada canal o empresa por los cuales la misma es reconocida ya sean logotipos, videos de presentación o video de fin de transmisión.
Iconografía	Conjunto de imágenes gráficas con las que se representan las secciones y el clima dentro de la aplicación.
Infocinta	Cintillo informativo que se muestra en las pantallas de tipo texto, imagen o texto-imagen, promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes.
Recursos visuales	Elementos que se enmarcan dentro de un panel informativo que ofrecen cierta información al usuario ya sea sobre el clima, la hora u otros elementos.
Panel informativo	Un panel informativo no es más que un área rectangular con una imagen de fondo que contiene recursos informativos de interés como pueden ser el clima y la hora.
Elementos configurables.	Conjunto de elementos a los que se les puede personalizar su diseño de interfaz de usuario.
Plantillas de transmisión	Estructura y orden que poseen los componentes a mostrar en una noticia. Puede poseer un banner, un panel informativo, áreas de texto, imágenes y videos.
Plantillas de administración	Forma en la que se muestra todo el entorno administrativo del sistema y sus componentes

Capítulo 1: Fundamentación Teórica

	visuales.
--	-----------

Tabla 1 Términos de la personalización

1.6.2 Procesos relacionados con la personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. Características generales.

En la Plataforma de Televisión Informativa Primicia se identificaron diferentes procesos que tienen que ver con la personalización de la interfaz de usuario del sistema de forma dinámica. A continuación se expondrán por separado dichos procesos para ambos subsistemas ya que no presentan las mismas características y por ende las mismas funcionalidades.

Gestión del Diseño Gráfico, Subsistema de Administración:

1) Gestionar el logotipo principal de la plantilla:

En este momento el logotipo de la plantilla de administración no puede modificarse a menos que se cambie la imagen del mismo físicamente.

2) Personalizar el formato de la fecha y hora del sistema.

El sistema de administración no brinda la posibilidad de escoger otro formato diferente al implantado dentro del código.

3) Gestión de las plantillas de administración.

La plataforma sólo cuenta con una plantilla para el subsistema de administración, lo que impide que el cliente pueda hacer algún cambio en el estilo o disposición de la misma sin tener que adentrarse en el código fuente.

4) Gestión de íconos.

La iconografía es un elemento que debe tener una total relación con la plantilla que se esté utilizando, actualmente la aplicación cuenta con un paquete de íconos que se definen a partir del diseño de la plantilla que se utiliza al entregar el software.

5) Gestión de infocintas.

Capítulo 1: Fundamentación Teórica

Hoy en día la plataforma muestra las infocintas de una sola forma, estilo, posición y efecto, imposibilitando que las mismas sean genéricas.

Gestión del Diseño Gráfico, Subsistema de Transmisión:

1) Gestión de plantillas de transmisión.

El subsistema de transmisión sólo cuenta con una plantilla para la visualización de las noticias y demás informaciones que brinda. Esto le impide al cliente poder modificar a su conveniencia tanto el estilo de dicha plantilla como la disposición de todos los elementos que la conforman.

2) Personalización del formato de hora a mostrar.

Al igual que en el subsistema de administración, este subsistema carece de una funcionalidad capaz de establecer formato de hora para su correcta visualización acorde a la plantilla o estándares empleados por diversos clientes.

3) Personalización de los elementos de identificación del canal.

Los elementos de identificación, como son los logotipos, video de presentación y video de fin de transmisión, presentes en la plataforma actualmente no se pueden modificar según los gustos del cliente. Dicha tarea debe realizarse de forma manual cambiando físicamente los archivos correspondientes cuando surge algún cambio en la identidad de la empresa o institución en la que se encuentra instalada la aplicación.

1.6.3 Elementos configurables de la plantilla de transmisión.

Después de un análisis del proceso de redacción de la Plataforma de Televisión Informativa queda evidente la necesidad de identificar los elementos configurables de las noticias con el objetivo de utilizarlos en la gestión de las plantillas de transmisión en el módulo de personalización.

A continuación se presentan dichos elementos:

- Texto
- Imagen
- Video

Capítulo 1: Fundamentación Teórica

- Infocinta
- Fecha
- Hora
- Clima
- Tiempo restante
- Sección
- Próxima sección
- Próxima noticia
- Panel Informativo
- Plantilla

Definición de los elementos:

Texto: Es un elemento que contendrá un texto determinado.

Imagen: Es un elemento que estará referenciando a un archivo de imagen.

Video: Es un elemento que estará referenciando a un archivo de video.

Infocinta: Este elemento de la noticia tiene como objetivo mostrar informaciones alternativas durante la transmisión de la noticia como pudiera ser el parte del tiempo o las personas que están dentro de una imagen. Generalmente su aparición es de corta duración.

Fecha: Este elemento representa la fecha del día en que se está ejecutando la aplicación, puede tener un componente texto asociado.

Hora: Este elemento representa la hora en que se está ejecutando la aplicación, puede tener un componente texto asociado.

Clima: Este elemento brinda informaciones sobre el estado del tiempo de una región determinada.

Capítulo 1: Fundamentación Teórica

Tiempo restante: Este elemento indica en todo momento el tiempo que falta para terminar la noticia.

Sección: Este elemento indica la sección temática a la que pertenece la noticia mostrada.

Próxima sección: Este elemento indica la sección temática que le sigue a la sección en curso.

Próxima noticia: Indicará el título de la siguiente noticia.

Panel Informativo: Elemento que contendrá otros elementos informativos.

Plantilla: Este elemento es de los más importantes ya que brinda una distribución inicial de los componentes de la noticia, facilitando la edición de la misma.

1.7 Conclusiones parciales.

En este capítulo se realizó un análisis de los conceptos que sirven como base al problema científico y al objetivo planteado en el estudio realizado. Con el análisis de las soluciones existentes se puede concluir que no existe ninguna plataforma informativa que presente un nivel de personalización de la interfaz de usuario como la que se pretende alcanzar en la aplicación. Por tanto para la personalización de la plataforma de televisión informativa es necesario lograr la correcta implementación de las funcionalidades manifestadas en los elementos configurables del sistema.

Capítulo 2: Tendencias y tecnologías actuales

Capítulo 2. Tendencias y tecnologías actuales

2.1 Introducción.

En este capítulo se analizan las principales tecnologías utilizadas en el mundo para la construcción de aplicaciones web que tienen estrecha relación con la Plataforma de Televisión Informativa Primicia. Se argumenta sobre el uso de las metodologías de desarrollo y lenguajes de programación. El objetivo principal que tiene es dar a conocer qué tecnologías se aplican en la implementación del módulo de personalización y el por qué de su selección.

2.2 Metodologías de desarrollo.

La presencia de un proceso bien definido y bien gestionado es una diferencia esencial entre proyectos exitosos y otros que fracasan. Una metodología de desarrollo es la guía que contiene aquellos pasos y procedimientos que deben seguirse para desarrollar un software. (Ivar Jacobson, 2000)

Las metodologías de desarrollo se clasifican en **Tradicional** (Metodologías Pesadas, o Peso Pesado) y las **Ágiles**. Las **Metodologías Tradicionales** son aquellas que ponen mayor énfasis en la planificación y control del proyecto y en la especificación precisa de requisitos y modelado. Las **Metodologías Ágiles** están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

Para comenzar el análisis de las metodologías se comienza con el estudio de las ágiles. Las más conocidas son: **Extreme Programming (XP)**, **Scrum**, **Agile Modeling** **Adaptive Software Development (ASD)** y **Crystal Clear**.

Estas metodologías requieren que el cliente del software sea parte del equipo de desarrollo pero Primicia no cuenta con un cliente específico. Además, se tiene presente que la mayoría del equipo son estudiantes y que cambian de roles de trabajo, por tanto se hace necesario generar una documentación descriptiva y precisa que permita un buen entendimiento del problema a solucionar.

Dentro de las **Tradicional** las más conocidas son: **Microsoft Solution Framework (MSF)** y **Rational Unified Process (RUP)**. MSF no sería factible pues fue desarrollada por Microsoft y requiere el uso de Microsoft Visual Studio Team System para el modelado de la solución. Debido a que Primicia se desarrolla completamente sobre plataformas libres y ambas metodologías cuentan con

Capítulo 2: Tendencias y tecnologías actuales

características similares, se decide utilizar RUP ya que esta ofrece iguales beneficios y su filosofía de trabajo es conocida por todos los miembros del proyecto.

Por su carácter iterativo e incremental, dirigido por los casos de uso y centrado en la arquitectura; RUP permite que después de cada iteración se pueda analizar la calidad del producto acompañado de prototipos de la aplicación, lo que facilita una correcta comprobación de su funcionalidad. Por estas razones se escoge RUP como metodología para el desarrollo del módulo de personalización de la interfaz de usuario de Primicia.

2.2.1 Proceso Unificado de Desarrollo (RUP).

El proceso unificado de desarrollo:

- ✓ Proporciona una guía para ordenar las actividades de un equipo. (Ivar Jacobson, 2000)
- ✓ Especifica los artefactos que deben desarrollarse. (Ivar Jacobson, 2000)
- ✓ Dirige las tareas de cada desarrollador por separado y del equipo como un todo. (Ivar Jacobson, 2000)
- ✓ Ofrece criterios para el control y la medición de los productos y actividades del proyecto. (Ivar Jacobson, 2000)
- ✓ Unifica los mejores elementos de las metodologías anteriores. (Ivar Jacobson, 2000)
- ✓ Preparado para desarrollar grandes y complejos proyectos. (Ivar Jacobson, 2000)
- ✓ Orientado a Objetos. (Ivar Jacobson, 2000)
- ✓ Utiliza el UML como lenguaje de representación visual. (Ivar Jacobson, 2000)

Las principales características en su ciclo de vida son:

- ✓ **Guiado por casos de uso:** Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como la secuencia de acciones realizadas por un sistema que produce un resultado observable de valor para un actor particular. (Ivar Jacobson, 2000)

Capítulo 2: Tendencias y tecnologías actuales

- ✓ **Centrado en la arquitectura:** La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados, entre otros. (Ivar Jacobson, 2000)
- ✓ **Iterativo e incremental:** La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada mini-proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. (Ivar Jacobson, 2000)

El proceso de desarrollo según RUP se divide en cuatro fases las cuales son:

- **Inicio:** tiene como objetivo determinar la visión del proyecto.
- **Elaboración:** en esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** el objetivo es obtener la capacidad operacional inicial.
- **Transición:** debe llegar a obtenerse un entregable del producto.

En todas las iteraciones durante cada fase se llevan a cabo nueve flujos de trabajo, dentro de los cuales los seis primeros son de ingeniería y el resto de apoyo.

- **Modelado del negocio:** Se identifican los procesos de negocio, los que estarán sujetos a automatización y quiénes intervienen en los mismos.
- **Requisitos:** Se identifican las restricciones que se imponen y lo que el sistema debe hacer.
- **Análisis y Diseño:** Describe cómo el programa será realizado y define cómo será programado.
- **Implementación:** Define cómo estarán colocados los nodos y la ubicación en paquete de los objetos y clases.
- **Prueba:** Se localizan los defectos del software.

Capítulo 2: Tendencias y tecnologías actuales

- **Despliegue:** Se entrega una versión operacional.
- **Administración de configuración y cambio:** Describe el uso y la actualización concurrente de los elementos, el control de versiones entre otras actividades.
- **Administración de proyecto:** Encargado de organizar el trabajo y de que se termine el proyecto en el tiempo previsto.
- **Ambiente:** Describe los procesos y herramientas que soportarán el trabajo del proyecto.

2.3 Lenguaje Unificado de Modelado (UML).

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y cuenta con reglas para combinar tales elementos. (Booch, 2000)

Permite la modelación de sistemas con tecnología orientada a objetos, de tal forma que es independiente del lenguaje de programación. Permite especificar las características que tendrá el sistema incluso antes de su construcción y brinda la posibilidad de representar gráficamente los modelos de manera que puedan ser comprendidos por todos los desarrolladores de Primicia.

Los diagramas UML se clasifican en: de estructura estática, de comportamiento y de implementación. Como parte del rol se hacen los de implementación que son:

1. De implementación:
 - Diagrama de componentes
 - Diagrama de despliegue

A partir de los modelos que se vayan generando se van obteniendo elementos gráficos que sirven de documentación para futuras revisiones y versiones. Para este lenguaje se han desarrollado varias herramientas llamadas Computer Aided Software Engineering (CASE) que son de gran ayuda para los desarrolladores de software.

2.4 Herramienta de modelado.

Capítulo 2: Tendencias y tecnologías actuales

Las herramientas **CASE**⁶ son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, documentación o detección de errores entre otras. (Chamba, 2010)

Como ejemplos de herramientas CASE están Argo-Uml, Poseidon paraUml, Rational Rose y Visual Paradigm. La mayoría de las mencionadas necesitan de una licencia comercial para su uso y por ello se utiliza Visual Paradigm desde la primera versión de Primicia pues su uso no tiene trascendencia comercial ya que no se incorpora en el producto final.

2.4.1 Visual Paradigm 6.4.

Visual Paradigm es una herramienta visual de modelado que brinda una colección de menús, barras de herramientas y ventanas que forman el área de trabajo, lo cual permite crear diferentes tipos de diagramas en un ambiente completamente visual. Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software.

Provee una manera intuitiva para llevar a cabo sistemas de análisis y diseño orientado a objetos. Soporta los últimos estándares de anotaciones de Java y se integra con algunas herramientas de este lenguaje, como: Eclipse, NetBeans, JBuilder y Oracle.

Se elige la herramienta Visual Paradigm para el modelado de la personalización de la interfaz de usuario de Primicia pues es la escogida por el proyecto además de presentar las siguientes ventajas:

- ✓ Es una herramienta gratis.
- ✓ La Edición Comunidad (Visual Paradigm Community Edition) es libre.
- ✓ Multiplataforma.
- ✓ Incluye como idioma el castellano.

⁶En español Ingeniería de Software Asistida por Ordenador

Capítulo 2: Tendencias y tecnologías actuales

- ✓ Muy personalizable.

2.5 Lenguajes de programación del lado del cliente.

2.5.1 HTML.

HyperText Markup Language (HTML) o Lenguaje Marcado de Hipertexto es un estándar mundialmente utilizado para la elaboración de páginas web. Sus normas están definidas por la World Wide Web Consortium (W3C) colaborando a que toda página HTML sea vista de similar manera desde cualquier navegador indistinto del sistema operativo utilizado.

Este lenguaje puede ser creado y editado usando editores de texto básico (Bloc de notas o Gedit) aparte de otras aplicaciones que tienen como objetivo el diseño y la construcción de sitios web (Dreamweaver o Quanta). Como ventajas tiene que es sencillo, presentando el texto de una forma estructurada y agradable en pequeños archivos. Sin embargo tiene como inconveniente que es un lenguaje estático y hace que el diseño sea más lento.

2.5.2 CCS.

Con el aumento de la creación de páginas web HTML y el uso de Internet, surge la necesidad de un estándar para la definición de estilos con distinta apariencia. A ello se une la W3C apostando por su desarrollo para controlar la presentación de los documentos electrónicos con HTML o XHTML. De esta forma logran crear el lenguaje Cascading Style Sheets por sus siglas en inglés CSS definiendo las propiedades de cada elemento del contenido en una página web.

Ejemplo de propiedades CSS utilizadas:

- Color
- Tamaño
- Tipo de letra
- Separación horizontal y vertical
- Posición

Capítulo 2: Tendencias y tecnologías actuales

Ejemplo en el código:

```
#img-chooser-view .loading-indicator {  
  
font-size:11px;  
  
background-image:url('../resources/images/grid/loading.gif');  
  
background-repeat: no-repeat;  
  
background-position: left;  
  
padding-left:20px;  
  
margin:10px;  
  
}
```

2.5.3 JavaScript.

JavaScript es un lenguaje de programación interpretado y su código puede ser integrado dentro de las páginas web. Se utiliza principalmente para crear páginas web dinámicas, las cuales son aquellas que incorporan efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Este lenguaje enriquece la experiencia de los usuarios en los sitios web y en el caso de la personalización de la transmisión donde se requiere un alto dinamismo en la web para crear las nuevas plantillas, se hace imprescindible la utilización del mismo.

2.6 Lenguaje de programación del lado del servidor.

2.6.1 PHP.

Hypertext Preprocessor (PHP) es un lenguaje de programación interpretado en el lado del servidor que no necesita ser compilado para ejecutarse. Es especialmente utilizado para la creación de sitios dinámicos incluyendo el código fácilmente dentro del HTML. Está publicado bajo la PHP License, la cual es considerada por la Free Software Foundation como software libre.

Capítulo 2: Tendencias y tecnologías actuales

Una de las ventajas para su utilización es que su aprendizaje es fácil y rápido. Es un lenguaje multiplataforma, que tiene una gran capacidad de conexión con la mayoría de los gestores de base de datos. Incluye gran cantidad de funciones y no requiere definición de variables. Sus desventajas son la necesidad de tener instalado un servidor web, por tanto todo el trabajo es realizado por la máquina servidora y esto puede entorpecer el trabajo al aumentar la cantidad de peticiones.

2.7 Framework.

Es necesario entender que un framework es una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Entonces, un framework es concreto y también “incompleto”. Concreto porque es, desde un punto de vista simple, un conjunto de componentes; incompleto, porque por sí mismos no pueden ser utilizados, ya que guían a la solución de problemas de programación recurrentes, pero por lo general, no son la solución específica completa. (Van Der Henst S., 2011)

2.7.1 Framework Symfony 1.4.6.

Un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener y en eso Symfony ayuda mucho. Es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, del servidor y la presentación de la aplicación web basándose en el patrón arquitectónico MVC el cual está formado por:

- El **modelo**: Representa la información con la que trabaja la aplicación, es decir su lógica de negocio.
- La **vista**: Transforma el modelo en una página web para que el usuario pueda interactuar con la aplicación.
- El **controlador**: Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

Capítulo 2: Tendencias y tecnologías actuales

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- ✓ La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- ✓ Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).
- ✓ El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

Symfony es maduro, bien documentado y con una gran comunidad que la apoya. Permite ser personalizado para cumplir con los requisitos de Primicia y más específicamente del módulo de personalización de la interfaz de usuario que dispone de sus propias políticas y reglas para el desarrollo.

2.7.2 Framework JavaScript.

En su mayoría, los frameworks JavaScript proveen componentes para:

Capítulo 2: Tendencias y tecnologías actuales

- ✓ **Compatibilidad:** Agregan la posibilidad de ser totalmente compatible con todos los navegadores más utilizados, de forma tal que aumentan su portabilidad.(Van Der Henst S., 2011)
- ✓ **Comunicación asíncrona (Ajax):** Facilita utilizar XMLHttpRequest para manejar y manipular los datos en los elementos de un sitio, aumentando la interactividad y experiencia del usuario. (Van Der Henst S., 2011)
- ✓ **DOM:** Maximiza la capacidad de agregar, editar o eliminar elementos de manera dinámica agregando librerías que facilitan usar DOM. (Van Der Henst S., 2011)
- ✓ **Validación de Formularios:** Permiten de una manera relativamente fácil validar campos dentro de uno o varios formularios. Esto, desde el punto de vista del desarrollador, simplifica y reduce el código para procesar dichos formularios, ya que los datos llegan previamente validados, reduciendo los errores de tipos de datos. (Van Der Henst S., 2011)
- ✓ **Efectos visuales:** Utilizando la manipulación de los elementos, se pueden crear efectos visuales y animaciones. Entre los efectos se encuentran: Aparecer y Desaparecer, Redimensionamiento y Mover. (Van Der Henst S., 2011)
- ✓ **Manejo JSON:** Incrementa al máximo el manejo de datos, que pueden ser utilizados para presentar informaciones de manera dinámica y en tiempo de ejecución. (Van Der Henst S., 2011)
- ✓ **Manejo de Eventos:** Permite reaccionar de una manera u otra dependiendo de las acciones del usuario. (Van Der Henst S., 2011)
- ✓ **Recibidores de Datos:** Permiten utilizar diferentes formatos de datos como XML, HTML, Texto, JSON, entre otros. (Van Der Henst S., 2011)

2.7.3 ExtJS 3.0.

Originalmente fue construido como una extensión de **YUI**⁷ y luego le incluyeron adaptación con **jQuery** y **Prototype**. Posee controles para campos de textos, selectores de fecha, campos numéricos,

⁷ Yahoo User Interface

Capítulo 2: Tendencias y tecnologías actuales

caja de radio (radiobox) y casilla de verificación (checkbox). También componentes para crear y manipular celda de datos (DataGrids) donde goza de cierta ventaja sobre otros frameworks. Es posible crear “ventanas” con barras de herramientas y menús con estilo de aplicaciones de escritorio, diálogos modales y eventos.

Ext JS puede ser adquirido bajo licencias libres y comerciales. La compañía detrás de este framework (Sencha) ofrece cursos de capacitación y un extenso soporte. La versión 3.0 tuvo su liberación final el 10 de Agosto de 2009 con mejoras en consistencia y manejo de memoria. Es la primera versión que aparece con el nombre Ext JS.

Sus principales características son:

- ✓ Alto rendimiento en ejecución debido a la optimización de código JavaScript.
- ✓ Controles de usuario personalizables.
- ✓ Modelo orientado a componentes, bien diseñado y extensible.
- ✓ Posee una API intuitiva y fácil de utilizar.
- ✓ Crea aplicaciones web con interfaces similares a escritorio.
- ✓ Permite la integración con el framework de PHP Symfony.

Alguno de los componentes y funcionalidades que trae consigo son los necesarios para el desarrollo del módulo y como ayuda se tiene que es usado en otros proyectos productivos de la universidad pudiendo contar con esa experiencia. Tiene un buen acoplamiento con Symfony, por tanto es el escogido para la implementación de la interfaz y manejo de datos del módulo.

2.8 Entorno de Desarrollo Integrado.

Un Entorno Integrado de Desarrollo (Integrated Development Environment por sus siglas en ingles IDE) es una aplicación compuesta de herramientas útiles para un programador que puede ser usado tanto para un solo lenguaje de programación como para varios. Están compuestos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Capítulo 2: Tendencias y tecnologías actuales

Dentro de los IDEs que soportan múltiples lenguajes se encuentran NetBeans y Eclipse los cuales son proporcionados por plugins que se instalan en el mismo. Su construcción está basada en lenguajes multiplataforma como Java o MonoDevelop lo cual ofrece como característica que se ejecuten en múltiples plataformas incluyendo Windows, GNU/Linux, y Mac OS.

2.8.1 NetBeans 6.9.

NetBeans es un reconocido entorno de desarrollo integrado disponible para los Sistemas Operativos Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto, libre, gratuito y sin restricciones de uso. Ha logrado un gran éxito con una inmensa base de usuarios, una comunidad en constante crecimiento y con cerca de 100 socios en todo el mundo.

Respecto a otros IDE tiene un mejor aprovechamiento del tiempo de desarrollo, alta organización de la aplicación basada en estándares y patrones estructurales y de diseño. Posee una arquitectura consistente y robusta y un mejor rendimiento en cuanto a tiempo de ejecución y optimización de recursos.

Es una plataforma que permite a los desarrolladores crear aplicaciones de escritorio con un enfoque modular como la de Primicia y pensando en características como la extensibilidad y la escalabilidad. Se integra fácilmente con el framework Ext JS anteriormente elegido para el desarrollo del módulo de personalización de la interfaz de usuario. La versión usada NetBeans 6.9 cuenta con soporte para PHP 5.3 y para Symfony con las siguientes prestaciones:

- ✓ Ejecución de comandos Symfony.
- ✓ Limpiar caché.
- ✓ Ayuda para encontrar comandos y para conocer los parámetros.
- ✓ Muestra toda la ejecución de cada comando (creación de ficheros, directorios, etc.).
- ✓ Configuración de atajos de teclado.

2.9 Firebug 1.7.0.

Es una extensión para el navegador Mozilla Firefox que permite depurar el código fuente de cualquier web sin tener que abandonar el navegador. Está pensada para programadores y desarrolladores web

Capítulo 2: Tendencias y tecnologías actuales

de tal forma que es posible inspeccionar al detalle el código fuente y el funcionamiento de la aplicación o sitio web, localizando e incluso depurando cualquier error al instante. Dicha extensión abre una ventana a modo de consola en el navegador mostrándote distintas opciones como inspeccionar los elementos de la página, el código HTML, CSS, scripts y objetos DOM.

2.10 Conclusiones parciales.

En este capítulo se explican las herramientas y lenguajes a utilizar para la implementación del módulo de personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia definidas en el documento de arquitectura del proyecto. Se define RUP como metodología de desarrollo, pues facilita toda la documentación necesaria para el desarrollo del software; para el modelado se utilizará UML y como herramienta CASE Visual Paradigm, que es la utilizada desde los inicios del proyecto. Los framework Symfony y Ext JS junto con el NetBeans, logran una gran unificación para la implementación de todo el código necesario que es la base para la terminación del módulo.

Capítulo 3: Construcción de la solución

Capítulo 3. Construcción de la solución

3.1 Introducción.

En este capítulo se hace una breve descripción del estándar de codificación a emplear. Se modela el diseño del sistema a implementar haciendo referencia a los principales artefactos que constituyen la base para el flujo de implementación.

3.2 Estándar de codificación.

El estándar en el código de programación es muy importante en cualquier proyecto, principalmente si este involucra muchos desarrolladores. Esto asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener. Los criterios y nomenclaturas planteadas son las definidas por el proyecto y el departamento, teniendo como referencia el estándar de codificación de Zend, empresa del creador de PHP Rasmus Lerdorf.

3.2.1 Criterios de calidad.

En la codificación se consideran los siguientes criterios de calidad:

- ✓ Los nombres de cada uno de los elementos del programa son significativos; su nombre explica en lo posible el uso del elemento.
- ✓ No se maneja en los programas más de una instrucción por línea.
- ✓ Las variables son declaradas en líneas separadas.
- ✓ Se añaden comentarios descriptivos junto a cada declaración de variables, si es necesario.
- ✓ La mayoría de los elementos se nombran usando sustantivos.
- ✓ Los atributos comienzan con letra minúscula y los métodos deben comenzar con letra mayúscula.

3.2.2 Nomenclatura para el nombramiento de elementos.

<u>Elemento</u>	<u>Reglas de nombramiento</u>
-----------------	-------------------------------

Capítulo 3: Construcción de la solución

Clases, interfaces y archivos fuente.	Nombre sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas. Las clases e interfaces comienzan con un prefijo alegórico al componente que correspondan, evitando nombres duplicados de los mismos. Si el nombre de la clase está comprimido en más de una palabra la primera letra será mayúscula. Letras mayúsculas seguidas no son permitidas.
Constantes.	Nombre sustantivo en mayúsculas. Para separar palabras se usará el guión bajo: _.
Paquetes y directorios.	Nombre sustantivo singular en minúsculas.

Tabla 2 Nombramiento de elementos

3.2.3 Nomenclatura de comentarios.

Cada función tiene un encabezado con:

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

Se utiliza dos tipos de comentarios:

- ✓ Se puede utilizar en el caso de un atributo, de un método o algún segmento de código (`/* */`).
- ✓ Los otros comentarios necesarios deben figurar a partir de la columna 35. (`//.....`)

3.2.4 Nomenclatura de nombres de identificadores.

Los identificadores siguen las siguientes normas:

Capítulo 3: Construcción de la solución

- ✓ Un nombre significativo para que con su simple lectura se conozca la función.
- ✓ Para nombres que se usen con frecuencia o para términos largos, se usa abreviaturas estándar para que éstos tengan una longitud razonable.
- ✓ Se evitan identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.

3.2.5 Nomenclatura de identificadores de variables.

Se comienza siempre con la primera letra minúscula correspondiente a su tipo de dato. Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (_).

3.2.6 Otros aspectos del código.

- ✓ **Indentación.**
 - Usar los sangrados para indentar el código y nunca espacios.
- ✓ **Saltos de línea.**
 - Añadir un salto de línea después del cierre de los paréntesis de los parámetros.

Ejemplo en el código:

```
function Formulario_guardar()
{
    if(ArrComponentes.length > 0)
    {
```

Ilustración 1 Salto de línea

- Añadir un salto de línea después de un punto y coma, cuando termina la sentencia.

Ejemplo en el código:

Capítulo 3: Construcción de la solución

```
$nombres = array();
foreach($temastransmision as $ttrans)
{
    $nombre = $ttrans -> getNombre();
    $nombres[] = $nombre;
}
$output = $nombres;
$this -> salida = json_encode($output);
```

Ilustración 2 Salto de línea después de punto y coma

✓ **Espacios y líneas en blanco.**

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

✓ **Longitud de la línea.**

- Evitar las líneas de más de 80 caracteres cuando se supera esta cifra se reorganiza el código usando algún principio descrito anteriormente.

✓ **String literales.**

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

```
$file = 'Almacen/Xml/';
```

✓ **String literales con apóstrofes.**

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

✓ **Sustitución por variables.**

- Cuando un texto contiene porciones que serán sustituidas por el valor de alguna variable, se toma como correcto únicamente estas dos formas:

Capítulo 3: Construcción de la solución

```
$reloj = "La hora es: $time";
```

```
$reloj = "La hora es: {$time}.";
```

✓ **Concatenación.**

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

```
$framework = 'Ext.' " 'JS';
```

✓ **Arrays.**

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

3.3 Requerimientos funcionales.

Los requisitos funcionales definen las acciones que debe realizar el sistema. Son capacidades o condiciones que la aplicación debe cumplir, éstos se mantienen invariables sin importar con que propiedades o cualidades se relacionen. A continuación se explican los requerimientos planteados en el documento de diseño y análisis de la personalización de la interfaz de usuario para un mayor entendimiento del módulo.

R1. Gestionar Infocinta: Es necesario crearlas, modificarlas, eliminarlas. Las infocintas que se proponen tiene una serie de propiedades como son: tipo, estilo, recursos, efectos, duración.

✓ **R1.1. Crear Infocinta.**

✓ **R1.2. Modificar Infocinta.**

✓ **R1.3. Eliminar Infocinta.**

✓ **R1.4. Visualizar Infocinta.**

R2. Gestionar plantillas de transmisión: La plataforma debe brindar la posibilidad de adicionar diversos esquemas de plantillas para la transmisión. Debe poder modificar dichas plantillas tanto global

Capítulo 3: Construcción de la solución

como parcialmente en aras de modificar el contenido y eliminar esquemas de plantillas no deseados o parte de los elementos que conforman una en dependencia de las necesidades del cliente.

- ✓ **R2.1. Crear plantillas.**
- ✓ **R2.2. Modificar plantillas.**
- ✓ **R2.3. Eliminar plantillas.**
- ✓ **R2.4. Visualizar plantillas.**

R3. Administrar iconografía: Se cargará un comprimido de iconos a escoger por el usuario. Esta nueva funcionalidad permitirá que la plataforma tenga un mayor acople entre las diferentes necesidades del cliente.

R4. Administrar recursos visuales: Se propone definir en cada plantilla una serie de paneles cuya función sería la de enmarcar los diferentes recursos tales como: el tiempo, la fecha, el clima, etc. Dichos recursos solo aparecerían dentro del límite de cada panel habilitado en cada plantilla y se deberá poder adicionar, eliminar, modificar y ordenar cada uno de estos recursos dentro del panel.

- ✓ **R4.1. Seleccionar recursos visuales.**
- ✓ **R4.2. Ordenar recursos visuales.**

R5. Establecer formato de Fecha/Hora: Se podrá establecer el formato de visualización de la fecha y hora del sistema para garantizar que estas se puedan visualizar de diferentes modos según lo necesitado por el cliente.

R6. Gestionar paneles informativos: La plataforma deberá permitir la ubicación de los paneles informativos de acuerdo a las posiciones definidas a la hora de crear la plantilla en caso de no haber seleccionado una por defecto así crear modificar o eliminar los mismos y administrar los recursos enmarcados por éste.

- ✓ **R6.1. Crear panel.**
- ✓ **R6.2. Modificar panel.**
- ✓ **R6.3. Eliminar panel.**

Capítulo 3: Construcción de la solución

R7. Administrar los elementos de identificación de la plataforma: Cada empresa posee ciertos elementos que la identifican tales como logotipo, videos de presentación de su canal y el sistema debe lograr la inserción de estos elementos la aplicación.

R8. Seleccionar plantillas de Administración: El sistema debe brindar la posibilidad de cambiar la plantilla de la parte administrativa de la plataforma permitiendo que el cliente escoja la plantilla más adecuada según su gusto.

3.4 Requerimientos no funcionales.

Los requerimientos no funcionales forman una parte significativa de la especificación. No son más que propiedades o cualidades que el producto debe tener. Dichos requerimientos son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

3.4.1 Apariencia o interfaz externa.

Interfaz amigable, interactiva, de fácil comprensión para el usuario, facilitando en todo momento la interacción de este con el sistema.

3.4.2 Disponibilidad.

- El sistema debe ser accesible los 7 días de la semana y las 24 horas del día, garantizando el buen funcionamiento de la aplicación.
- Los dispositivos o mecanismos de seguridad usados no deben retrasar a los usuarios para obtener los datos deseados en un momento que se requiera.

3.4.3 Usabilidad.

El sistema será utilizado por usuarios que no tienen necesariamente que poseer conocimientos informáticos, debido a esto las funcionalidades deben ser claras y mostrar la información de forma lógica y correctamente estructurada.

3.4.4 Soporte.

Capítulo 3: Construcción de la solución

El soporte y/o mantenimiento del sistema no debe necesariamente detener el servicio.

3.4.5 Restricciones en el diseño y la implementación.

- Para la implementación se debe utilizar como sistema operativo: Ubuntu 10.10.
- Se debe utilizar como servidor web: Apache.
- El sistema estará implementado en lenguaje PHP, utilizando como IDE de desarrollo NetBeans. Para la modelación se utilizará el Visual Paradigm.
- Diseño e implementación de una arquitectura flexible, que permita la fácil integración o desintegración de componentes.
- La arquitectura debe soportar migrar la interfaz de usuario de forma rápida.

3.5 Modelo de implementación.

El modelo de implementación proporciona una descripción de las principales capas y subsistemas de componentes de la aplicación. Describe cómo se organizan los elementos 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. (Ivar Jacobson, 2000) Entre los componentes se puede encontrar datos, archivos, ejecutables, código fuente y los directorios.

La Plataforma de Televisión Informativa Primicia esta conformada por dos subsistemas: Administración y Transmisión. Integrado en el Subsistema de Administración se encuentra el módulo de Personalización, el cual utiliza para sus interfaces código JavaScript brindado por el framework ExtJS. El framework Symfony de PHP se incluye para el manejo de datos del lado del servidor con la utilización del ORM⁸ (Object-Relational Mapping) Doctrine que facilita la interacción con la base de datos. (ver ilustración 3)

⁸ORM, en español Mapeo Relacional de Objetos, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

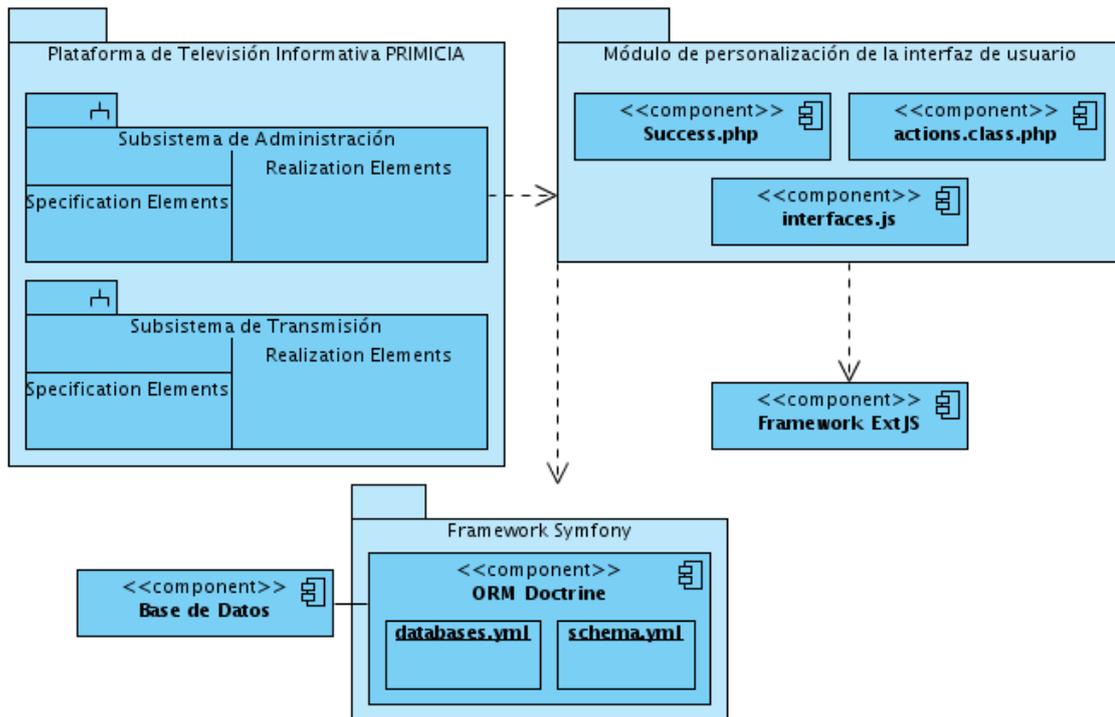


Ilustración 3 Modelo de implementación

3.6 Diagramas de componentes.

Los diagramas de componentes modelan la vista estática de un sistema. Se representan como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

Capítulo 3: Construcción de la solución

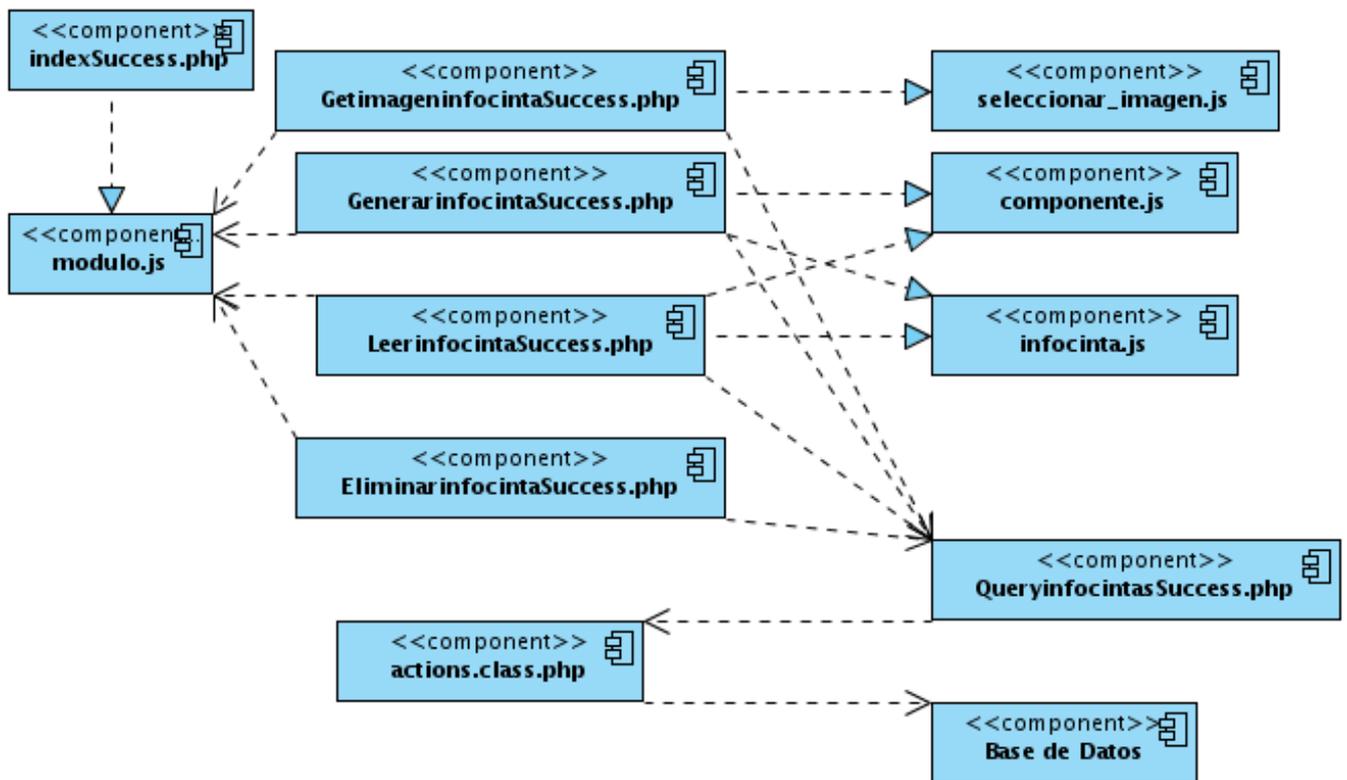


Ilustración 4 Diagrama de componentes-CU Gestionar infocinta

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso gestionar infocinta (ver ilustración 4):

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `GetimageninfocintaSuccess.php`: vista llamada al crear una nueva infocinta para obtener la colección de imágenes.
- `GenerarinfocintaSuccess.php`: vista que crea la plantilla de la infocinta con todos sus atributos.
- `LeerinfocintaSuccess.php`: vista que carga la información de la plantilla de la infocinta tanto para modificarla como para visualizarla.

Capítulo 3: Construcción de la solución

- EliminarinfocintaSuccess.php: vista que brinda los nombres de las plantillas de infocintas creadas para eliminarlas.
- seleccionar_imagen.js: interfaz JavaScript que muestra todas las imágenes físicas añadidas en la base de datos.
- componente.js: elemento creado como extensión de un objeto Ext.Panel del framework ExtJS para crear las distintas áreas de la plantilla.
- infocinta.js: elemento creado como extensión de un objeto Ext.BoxComponent del framework ExtJS para crear la infocinta con imagen de fondo.
- QueryinfocintasSuccess.php: vista que llama a la consulta con la base de datos para obtener las imágenes y plantillas existentes.
- actions.class.php: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

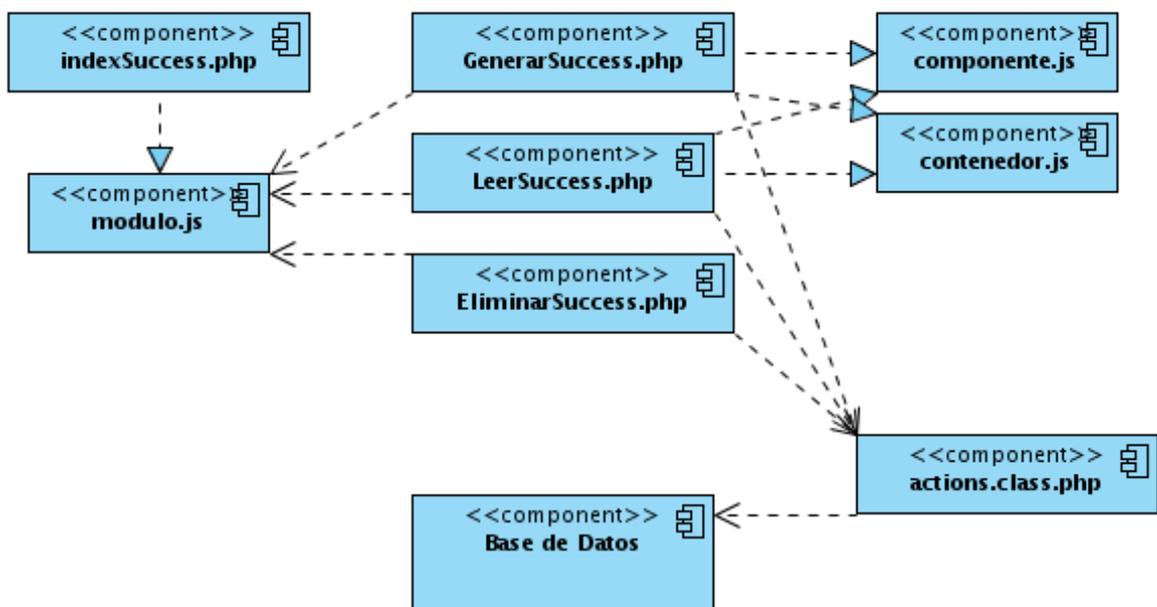


Ilustración 5 Diagrama de componentes-CU Gestionar plantilla de transmisión

Capítulo 3: Construcción de la solución

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso gestionar plantilla de transmisión (ver ilustración 5):

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `GenerarSuccess.php`: vista para crear la plantilla de transmisión con todos sus atributos.
- `LeerSuccess.php`: vista que captura todas las propiedades de la plantilla de transmisión para mostrarla tanto para el modificar como para el visualizar.
- `EliminarSuccess.php`: vista que devuelve todas las plantillas existentes en la base de datos para eliminar la seleccionada.
- `componente.js`: elemento creado como extensión de un objeto `Ext.Panel` del framework ExtJS para crear las distintas áreas de la plantilla.
- `contenedor.js`: elemento creado como extensión de un objeto `Ext.Panel` del framework ExtJS para crear los contenedores de los elementos insertados en la plantilla.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

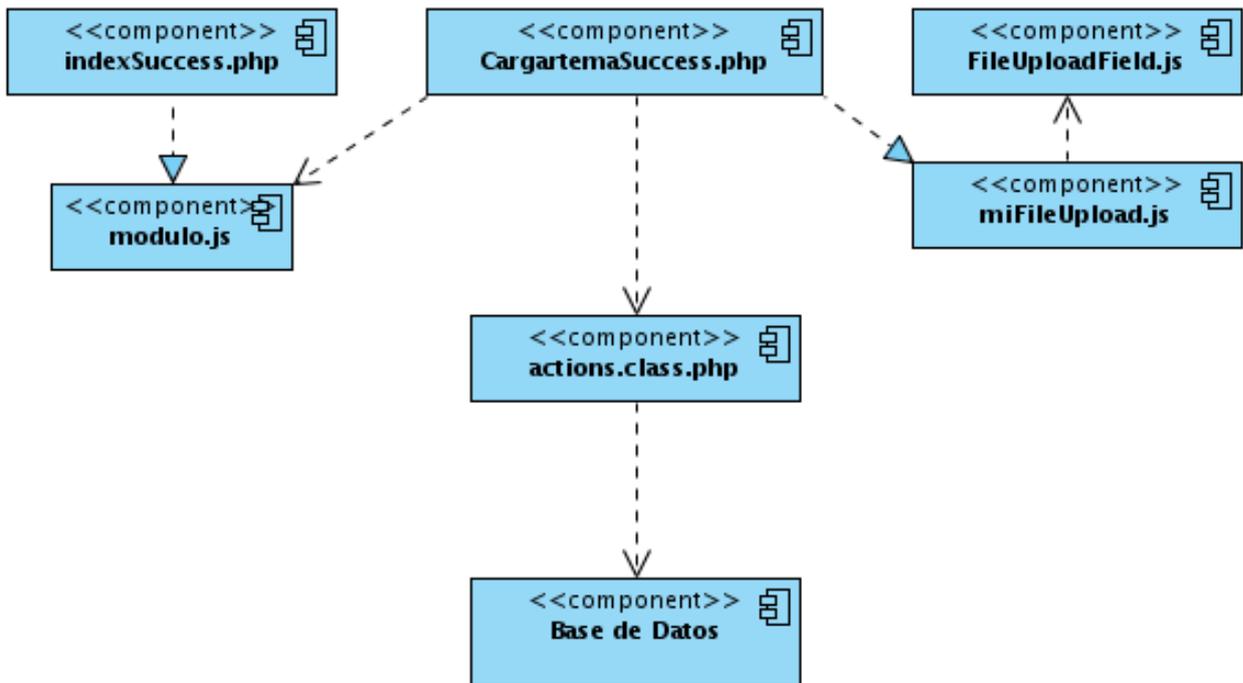


Ilustración 6 Diagrama de componentes-CU Administrar iconografía

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso administrar iconografía (ver ilustración 6):

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `CargartemaSuccess.php`: vista encargada de subir al servidor el tema de la transmisión con los nuevos iconos del clima y de las secciones temáticas.
- `miFileUpload.js`: clase JavaScript que brinda como funcionalidad la subida de archivos al servidor.
- `FileUploadField.js`: extensión del framework ExtJS de crear la interfaz para examinar el sistema de archivos y seleccionar.

Capítulo 3: Construcción de la solución

- actions.class.php: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

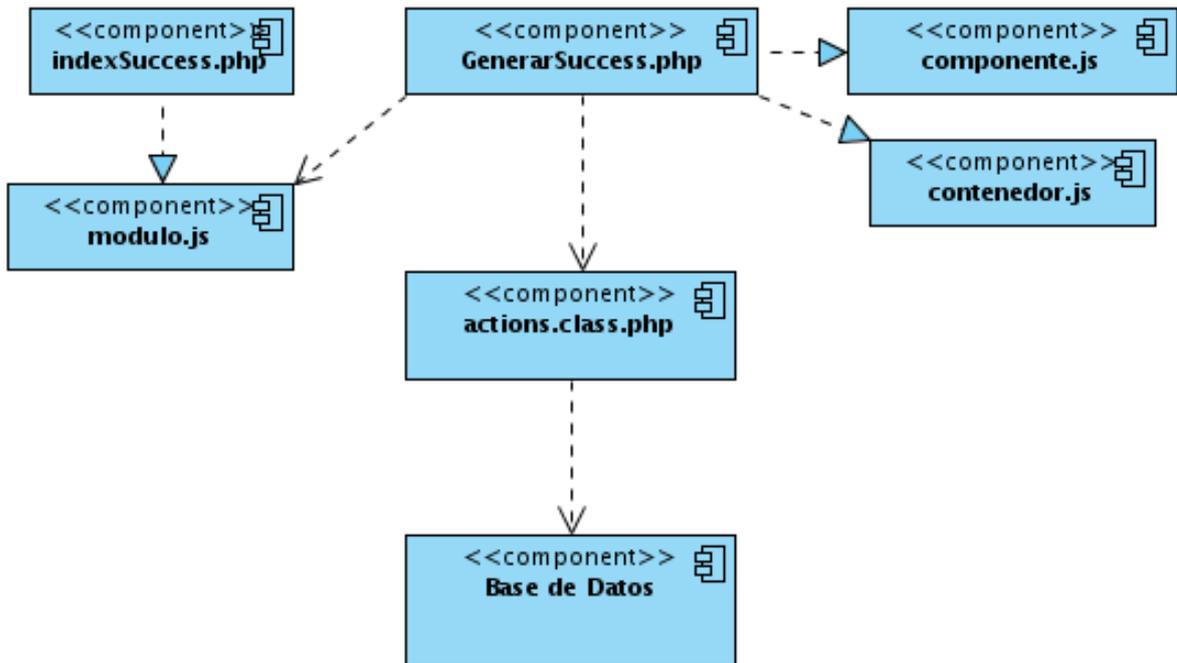


Ilustración 7 Diagrama de componentes-CU Administrar recursos visuales

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso administrar recursos visuales (ver ilustración 7):

- indexSuccess.php: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- modulo.js: clase JavaScript que construye la interfaz del módulo.
- GenerarSuccess.php: vista para crear la plantilla de transmisión con todos sus atributos, en este caso incluyendo el panel informativo que contiene los recursos visuales.
- componente.js: elemento creado como extensión de un objeto Ext.Panel del framework ExtJS para crear las distintas áreas de la plantilla.

Capítulo 3: Construcción de la solución

- `contenedor.js`: elemento creado como extensión de un objeto `Ext.Panel` del framework `ExtJS` para crear los contenedores de los elementos insertados en la plantilla.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

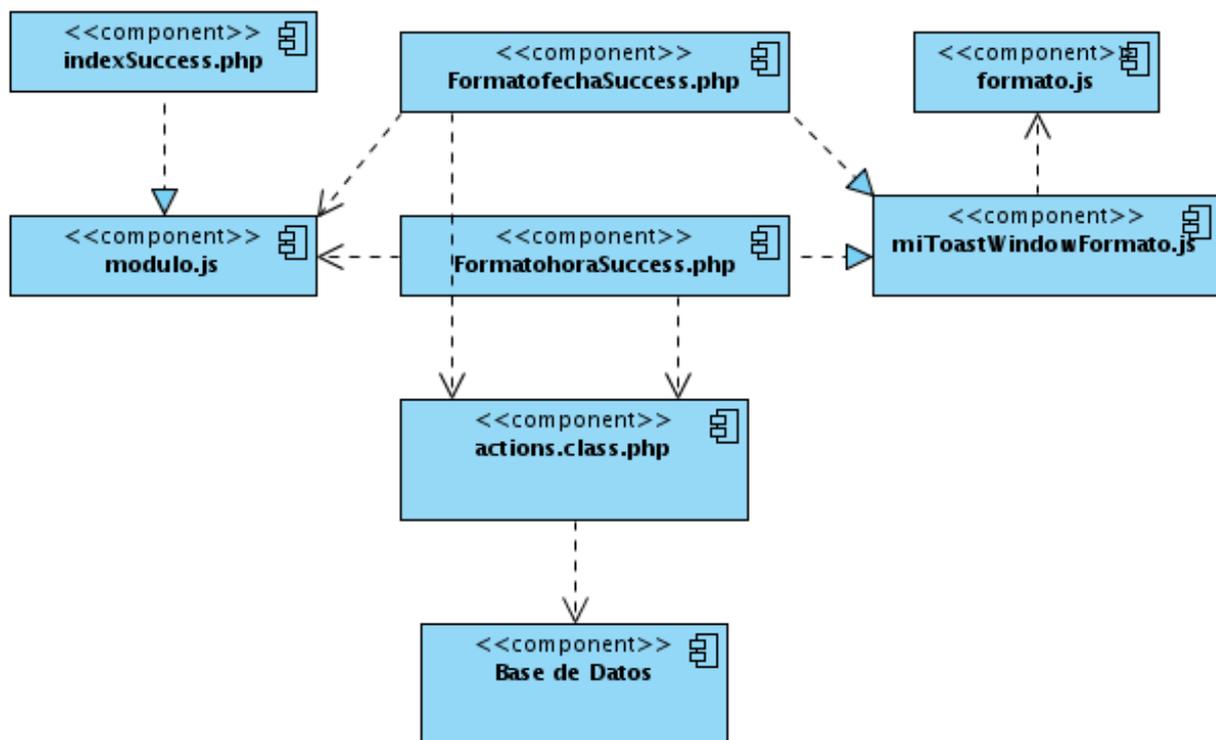


Ilustración 8 Diagrama de componentes-CU Establecer formato fecha/hora

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso establecer formato fecha/hora (ver ilustración 8):

- `indexSuccess.php`: clase perteneciente a la vista del framework `symfony` que construye la interfaz `js`.
- `modulo.js`: clase `JavaScript` que construye la interfaz del módulo.
- `FormatofechaSuccess.php`: vista que muestra la interfaz `miToastWindowFormato.js` con el formato de la fecha

Capítulo 3: Construcción de la solución

- `FormatohoraSuccess.php`: vista que muestra la interfaz `miToastWindowFormato.js` con el formato de la hora.
- `miToastWindowFormato.js`: interfaz que brinda la opción de poder modificar el formato actual de la fecha y la hora
- `formato.js`: manejo auxiliar de datos para la selección del formato de la fecha y hora de la aplicación.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

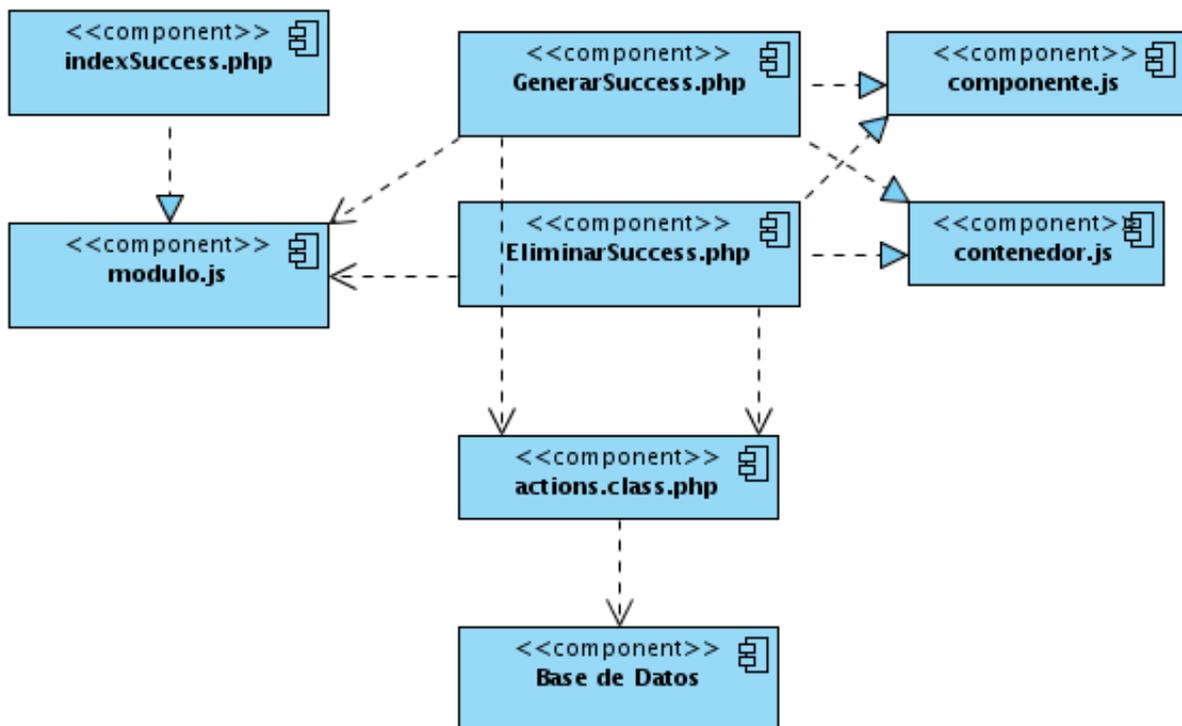


Ilustración 9 Diagrama de componentes-CU Gestionar panel informativo

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso gestionar panel informativo (ver ilustración 9):

Capítulo 3: Construcción de la solución

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `GenerarSuccess.php`: vista para crear la plantilla de transmisión con todos sus atributos incluyendo panel informativo.
- `LeerSuccess.php`: vista que captura todas las propiedades del panel informativo para mostrarla tanto para el modificar como para el visualizar plantilla de transmisión.
- `componente.js`: elemento creado como extensión de un objeto `Ext.Panel` del framework `ExtJS` para crear las distintas áreas del panel informativo.
- `contenedor.js`: elemento creado como extensión de un objeto `Ext.Panel` del framework `ExtJS` para crear el panel informativo de la plantilla.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

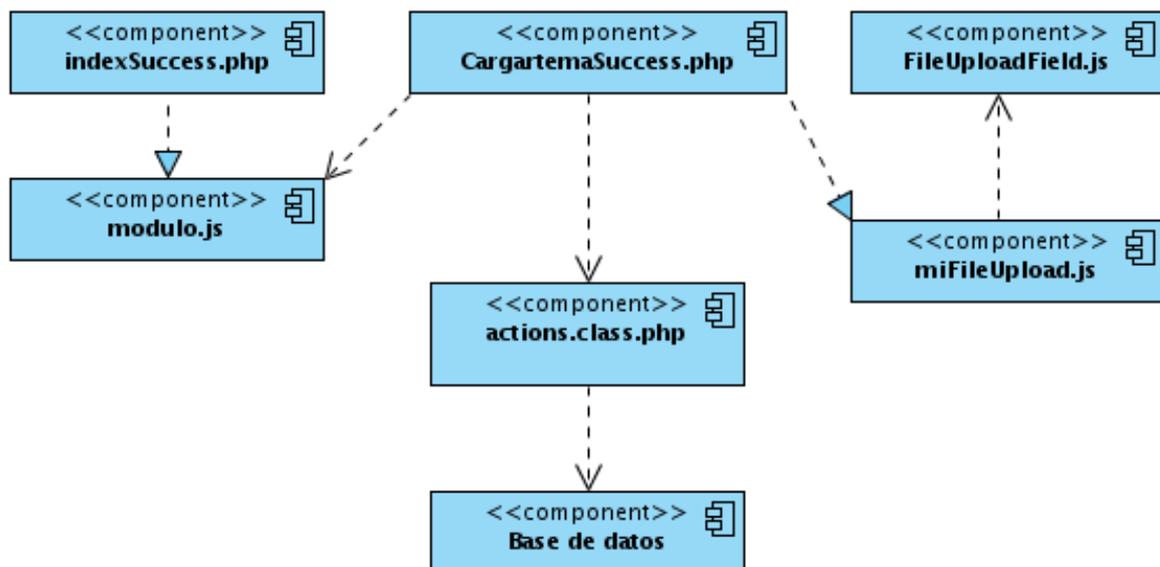


Ilustración 10 Diagrama de componentes-CU Administrar elementos de identificación

Capítulo 3: Construcción de la solución

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso administrar elementos de identificación (ver ilustración 10):

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `CargartemaSuccess.php`: vista encargada de subir al servidor el tema de la transmisión con los nuevos iconos del clima y de las secciones temáticas.
- `miFileUpload.js`: clase JavaScript que brinda como funcionalidad la subida de archivos al servidor.
- `FileUploadField.js`: extensión del framework ExtJS de crear la interfaz para examinar el sistema de archivos y seleccionar.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

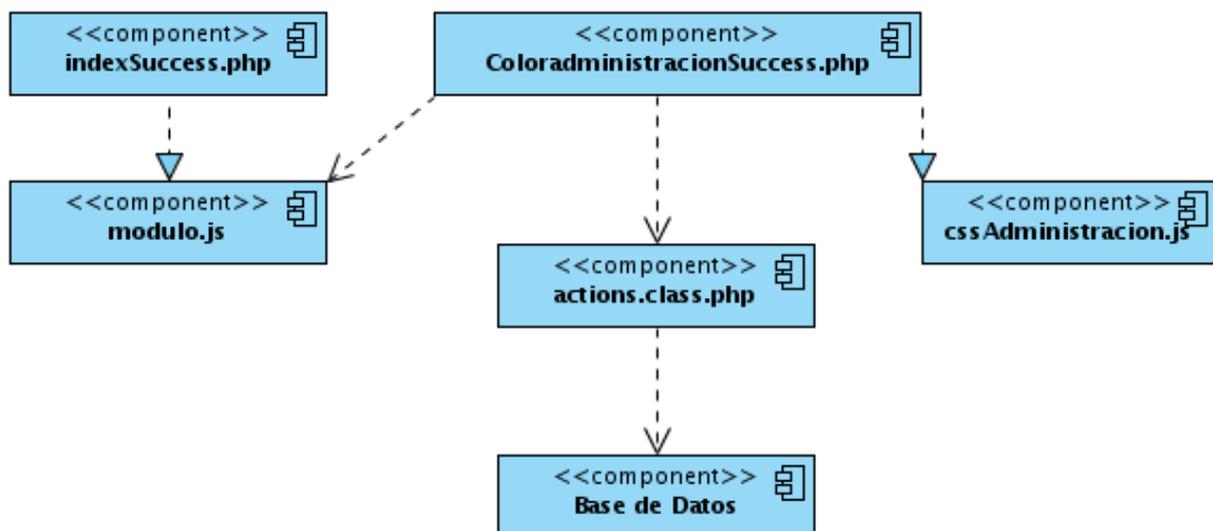


Ilustración 11 Diagrama de componentes-CU Seleccionar plantilla de administración

Capítulo 3: Construcción de la solución

A continuación se explicarán los elementos mostrados en el diagrama de componentes del caso de uso seleccionar plantilla de administración (ver ilustración 11):

- `indexSuccess.php`: clase perteneciente a la vista del framework symfony que construye la interfaz js.
- `modulo.js`: clase JavaScript que construye la interfaz del módulo.
- `ColoradministracionSuccess.php`: vista que nos muestra la interfaz `cssAdministracion.js`
- `cssAdministracion.js`: interfaz para seleccionar la vista del subsistema de administración.
- `actions.class.php`: controlador de todo el manejo de datos con la vista y la base de datos.
- Base de Datos: representación física de la base de datos.

3.7 Diagrama de despliegue.

El diagrama de despliegue brinda información sobre la distribución física del sistema y cada uno de sus componentes, además de dar solución a parte de los requisitos no funcionales de hardware y software.

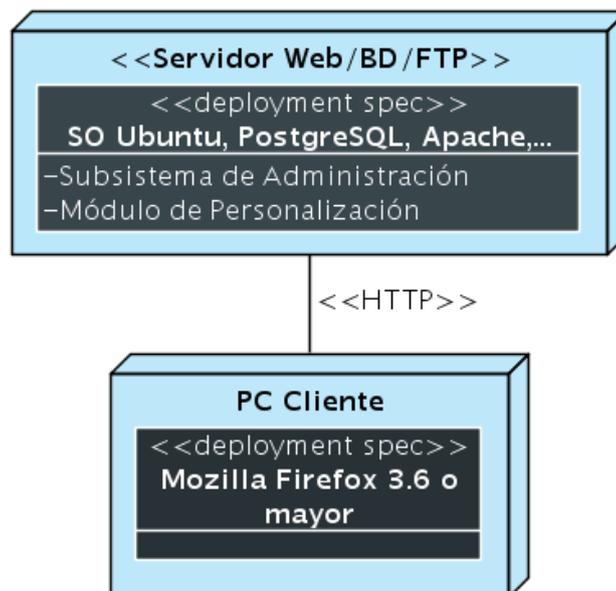


Ilustración 12 Diagrama de despliegue

Capítulo 3: Construcción de la solución

A continuación se describirán los nodos físicos representados en el diagrama de despliegue (ver ilustración 12):

- **Nodo Subsistema de Administración:** nodo para administrar el canal que contendrá el módulo de personalización de la interfaz de usuario de Primicia.
- **Nodo PC Cliente:** redactores o diseñadores que personalizan la transmisión.
- **HTTP:** Protocolo de comunicación entre la PC cliente y el Subsistema de Administración.

3.8 Conclusiones parciales

En este capítulo se aclara la nomenclatura a utilizar en el código para lograr un mejor entendimiento entre los desarrolladores del proyecto. Además se realizaron los artefactos del flujo de implementación establecidos por RUP y necesarios para la construcción del módulo. En ellos se muestra como se encuentran conformados en componentes los distintos casos de usos a implementar para lograr la personalización de Primicia.

Capítulo 4: Pruebas

Capítulo 4. Pruebas.

4.1 Introducción.

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requisitos específicos, los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente. (Ivar Jacobson, 2000)

El objetivo fundamental de la aplicación de pruebas a un software es detectar todo posible error antes de que se produzca. Un error que se encuentre durante la fase de pruebas es costoso de reparar, pero siempre es peor que sea el usuario final quien lo detecte. La efectividad de las pruebas está determinada por la cantidad de errores que queden por descubrir luego de haberlas realizado.

4.2 Diseño de casos de prueba.

Al módulo se le realizan tres iteraciones de pruebas de caja negra para verificar si cada función es plenamente operacional y buscar los posibles errores que tenga.

4.2.1 CUS: Gestionar Plantilla de Transmisión.

Secciones.

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
----------------------	-----------	------------------	-----------------------	------------------------

Capítulo 4: Pruebas

SC1: Crear Plantilla.	EC 1.1: Crear plantilla correctamente .	El usuario oprime el botón Nueva Plantilla y el sistema muestra el área de trabajo con las opciones necesarias para crear la plantilla. Luego inserta todos los elementos que va a ubicar en la plantilla que está creando. (Ver CU: Gestionar Elementos.)	El sistema muestra en el área de trabajo los elementos en la ubicación que desee el usuario.	Satisfactoria
	EC 1.2: Crear plantilla. (Flujo alterno)	El usuario oprime el botón Nueva Plantilla, con un área de trabajo abierta de otro flujo de trabajo.	El sistema muestra un mensaje de alerta para que el usuario cierre el área de trabajo abierta y comience a crear la plantilla en una nueva.	Satisfactoria
SC2: Modificar Plantilla.	EC 2.1: Modificar plantilla correctamente .	El usuario oprime el botón Modificar Plantilla y se muestra una ventana para que escoja la plantilla que va a modificar, de las existentes en el sistema. El usuario selecciona la plantilla y se muestra el área de trabajo con los elementos de la plantilla escogida. Luego realiza las modificaciones a los	El sistema muestra en el área de trabajo los elementos en la ubicación que desee el usuario.	Satisfactoria

Capítulo 4: Pruebas

		elementos que desea, puede moverlos por el área de trabajo, así como insertarlos y eliminarlos. (Ver CU: Gestionar Elementos.)		
	EC 2.2: Modificar plantilla. (Flujo alternativo)	El usuario oprime el botón Modificar Plantilla y no existen plantillas creadas.	El sistema muestra un mensaje notificando al usuario que no hay plantillas creadas en el sistema.	Satisfactoria
	EC 2.3: Modificar plantilla. (Flujo alternativo)	El usuario oprime el botón Modificar Plantilla, con un área de trabajo abierta de otro flujo de trabajo.	El sistema muestra un mensaje de alerta para que el usuario cierre el área de trabajo abierta antes de comenzar a trabajar con otra.	Satisfactoria
SC3: Eliminar Plantilla.	EC 3.1: Eliminar plantilla correctamente .	El usuario oprime el botón Eliminar Plantilla y se muestra una ventana para que escoja la plantilla que va a eliminar de las existentes en el sistema. El usuario escoge una plantilla y esta queda eliminada del sistema.	El sistema elimina la plantilla seleccionada y muestra un mensaje notificando que la plantilla fue eliminada correctamente.	Satisfactoria
	EC 3.2: Eliminar	El usuario oprime el botón Eliminar Plantilla y	El sistema muestra un mensaje notificando al	Satisfactoria

Capítulo 4: Pruebas

	plantilla. (Flujo alternativo)	no existen plantillas creadas.	usuario que no hay plantillas creadas en el sistema.	
SC4: Visualizar Plantilla.	EC 4.1: Visualizar plantilla correctamente .	El usuario oprime el botón Visualizar Plantilla y se muestra una ventana para que escoja la plantilla que va a visualizar de las existentes en el sistema. El usuario escoge una plantilla.	El sistema muestra los elementos de la plantilla seleccionada en el área de trabajo.	Satisfactoria
	EC 4.2: Visualizar plantilla. (Flujo alternativo)	El usuario oprime el botón Visualizar Plantilla y no existen plantillas creadas.	El sistema muestra un mensaje notificando al usuario que no hay plantillas creadas en el sistema.	Satisfactoria
	EC 4.3: Visualizar plantilla. (Flujo alternativo)	El usuario oprime el botón Visualizar Plantilla, con un área de trabajo abierta de otro flujo de trabajo.	El sistema muestra un mensaje de alerta para que el usuario cierre el área de trabajo abierta antes de comenzar a trabajar con otra.	Satisfactoria
SC5: Guardar Plantilla.	EC 5.1: Guardar plantilla correctamente .	El usuario oprime el botón Guardar y se muestra una ventana para que introduzca el nombre de la plantilla	El sistema muestra un mensaje informando que la plantilla fue guardada correctamente.	Satisfactoria

Capítulo 4: Pruebas

		que va a guardar.		
	EC 5.2: Guardar plantilla. (flujo alterno)	El usuario desea guardar una plantilla que no tenga componentes.	El sistema muestra una alerta notificando que debe insertar componentes en la plantilla.	Satisfactoria

Tabla 3 CUS Gestionar Plantilla de Transmisión

4.2.2 CUS: Gestionar Elementos.

Secciones.

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Adicionar elemento.	EC 1.1: Insertar Componente correctamente.	El usuario oprime el botón Insertar y selecciona la opción Componente. El sistema muestra una ventana para que escoja el tipo de componente y en qué contenedor lo va a insertar.	El sistema muestra en el área de trabajo el componente en la ubicación que desee el usuario.	Satisfactoria
	EC 1.2: Insertar Componente. (Flujo alterno)	Luego de seleccionada la opción insertar componente, el usuario no escoge el tipo de componente y el contenedor donde lo va a	El sistema muestra una ventana notificando al usuario que debe llenar todos los campos.	Satisfactoria

Capítulo 4: Pruebas

		insertar.		
	EC 1.3: Insertar Contenedor correctamente.	El usuario oprime el botón Insertar y selecciona la opción Contenedor. El sistema muestra una ventana para que escoja en qué contenedor lo va a insertar y si será editable o no.	El sistema muestra en el área de trabajo el contenedor ubicado dentro del elegido por el usuario.	Satisfactoria
	EC 1.4: Insertar Contenedor. (Flujo alternativo)	El usuario desea insertar contenedor y no selecciona el contenedor en que lo quiere insertar.	El sistema muestra un mensaje notificando al usuario que debe seleccionar un contenedor destino.	Satisfactoria
	EC 1.5: Insertar Panel Informativo correctamente.	El usuario oprime el botón Insertar y selecciona la opción Panel Informativo. El sistema muestra una ventana para que seleccione los recursos visuales que estarán en el panel informativo.	El sistema muestra en el área de trabajo el panel informativo agrupando los recursos visuales seleccionados.	Satisfactoria
	EC 1.6: Insertar Panel Informativo. (Flujo alternativo)	El usuario desea insertar el panel y no selecciona los recursos visuales que quiere insertar.	El sistema muestra una alerta indicando que el usuario debe escoger al menos un recurso visual.	Satisfactoria

Capítulo 4: Pruebas

SC2: Eliminar elemento.	EC 2.1: Eliminar Componente correctamente.	El usuario oprime el botón Eliminar y el sistema muestra una ventana para que escoja el componente que desea eliminar.	El sistema elimina del área de trabajo el componente seleccionado.	Satisfactoria
	EC 2.2: Eliminar Componente. (Flujo alterno)	Luego de seleccionada la opción eliminar componente, el usuario no escoge el componente que desea eliminar el sistema muestra una alerta notificando que debe escoger el componente.	El sistema muestra una alerta indicando que el usuario debe escoger el componente que desea eliminar.	Satisfactoria
	EC 2.3: Eliminar Contenedor correctamente.	El usuario oprime el botón Eliminar y el sistema muestra una ventana para que escoja el contenedor que desea eliminar.	El sistema elimina el contenedor seleccionado con todos los contenedores y componentes que tenga asociados.	Satisfactoria
	EC 2.4: Eliminar Contenedor. (Flujo alterno)	Luego de seleccionada la opción eliminar componente el usuario no escoge el componente que desea eliminar.	El sistema muestra una ventana notificando al usuario que debe seleccionar el contenedor que va a eliminar.	Satisfactoria
	EC 2.5: Eliminar	El usuario oprime el	Se elimina el panel con	Satisfactoria

Capítulo 4: Pruebas

	Panel Informativo correctamente.	botón Eliminar y selecciona la opción Panel Informativo.	sus recursos visuales del área de trabajo.	
	EC 2.6: Eliminar Panel Informativo. (Flujo alterno)	Selecciona la opción Panel Informativo del menú Eliminar.	Si no hay paneles en el área de trabajo se muestra una alerta indicando que no existe panel informativo.	Satisfactoria

Tabla 4 CUS Gestionar Elementos

4.2.3 CUS: Gestionar Infocinta.

Secciones.

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Crear Infocinta.	EC 1.1: Crear Infocinta correctamente.	El usuario selecciona la imagen que va a asignar a la infocinta, luego escoge el estilo que tendrá la infocinta (texto, climática o mixta).	El sistema muestra en el área de trabajo la imagen seleccionada con el estilo escogido.	Satisfactoria
	EC 1.2: Guardar infocinta correctamente.	El usuario oprime el botón Guardar y se muestra una ventana para que introduzca el nombre de la infocinta que va a guardar.	El sistema muestra un mensaje informando que la infocinta fue guardada correctamente.	Satisfactoria

Capítulo 4: Pruebas

	EC 1.3: Guardar plantilla. (flujo alterno)	El usuario desea guardar una infocinta que no tenga componentes.	El sistema muestra una alerta notificando que debe insertar un componente en la infocinta.	Satisfactoria
	EC 1.4: Eliminar Componente.	El usuario oprime el botón Eliminar. Ver CU Gestionar Elementos, Sección Eliminar Elementos.	El sistema elimina el componente de la infocinta.	Satisfactoria
SC2: Modificar Infocinta.	EC 2.1: Modificar Infocinta correctamente.	El usuario selecciona la infocinta que va a modificar y se muestra en el área de trabajo el componente. Realiza los cambios que desee y presiona el botón Guardar.	El sistema muestra en el área de trabajo los cambios realizados y los guarda.	Satisfactoria
	EC 2.2: Modificar Infocinta (Flujo Alterno).	No existen infocintas creadas.	El sistema muestra un mensaje notificando que no existen infocintas en el sistema.	Satisfactoria
	EC 2.3: Modificar Infocinta (Flujo Alterno).	El usuario selecciona modificar infocinta y hay otra área de trabajo abierta.	El sistema muestra un mensaje notificando que debe cerrar el área de trabajo que se está usando.	Satisfactoria

Capítulo 4: Pruebas

SC3: Eliminar Infocinta.	EC 3.1: Eliminar Infocinta correctamente.	El usuario selecciona la infocinta que va a eliminar.	El sistema elimina la infocinta seleccionada y muestra un mensaje notificando que la infocinta fue eliminada correctamente.	Satisfactoria
	EC 3.2: Eliminar Infocinta (flujo alterno).	El usuario no selecciona la infocinta que va a eliminar.	El sistema muestra un mensaje notificando que no se seleccionó ninguna infocinta.	Satisfactoria
	EC 3.3 Eliminar Infocinta (flujo alterno).	El usuario selecciona la opción Eliminar Infocinta y no hay infocintas en el sistema.	El sistema muestra un mensaje notificando que no hay infocintas creadas.	Satisfactoria
SC4: Visualizar Infocinta.	EC 4.1: Visualizar Infocinta correctamente.	El usuario selecciona la infocinta que va a visualizar.	El sistema muestra la infocinta seleccionada en el área de trabajo.	Satisfactoria
	EC 4.2: Visualizar Infocinta (flujo alterno).	El usuario selecciona la opción Visualizar Infocinta y no hay infocintas en el sistema.	El sistema muestra un mensaje notificando que no hay infocintas creadas.	Satisfactoria
SC5: Cargar Nueva Imagen.	EC 5.1: Cargar Nueva Imagen correctamente.	Se muestra un formulario para buscar la imagen que se va a subir al servidor, el usuario selecciona la	El sistema muestra un mensaje notificando que se subió la imagen para el servidor.	Satisfactoria

Capítulo 4: Pruebas

		imagen.	
--	--	---------	--

Tabla 5 CUS Gestionar Infocinta

4.2.4 CUS: Administrar Tema de Transmisión.

Secciones

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Seleccionar Tema de Transmisión.	EC 1.1: Seleccionar Tema de Transmisión correctamente.	El usuario selecciona el tema de transmisión que desea.	El sistema muestra en el área de trabajo el tema seleccionado y un mensaje notificando que el tema fue cargado correctamente.	Satisfactoria
SC2: Adicionar Tema de Transmisión.	EC 2.1: Adicionar Tema de Transmisión correctamente.	Se muestra un formulario para introducir el nombre del tema, escoger un video de presentación, una imagen de fondo del canal, otra de logo, un video para el cambio de secciones, un archivo comprimido con los logos de las secciones temáticas y otro con los iconos del clima y finalmente una imagen para transmitir cuando esté el canal fuera de	El sistema muestra un mensaje confirmando que los elementos del tema fueron subidos correctamente y guarda el nuevo tema.	Satisfactoria

Capítulo 4: Pruebas

		servicio.		
	EC 2.2: Adicionar Tema de Transmisión (Flujo Alternativo).	El usuario no selecciona todos los elementos que hay que importar.	El sistema muestra una alerta notificando al usuario que tiene que seleccionar todos los elementos que se piden.	Satisfactoria
	EC 2.3: Adicionar Tema de Transmisión (Flujo Alternativo).	El usuario selecciona archivos que no tienen el formato que se solicita.	El sistema muestra una alerta notificando al usuario que tiene que seleccionar los elementos que se piden con el formato requerido.	Satisfactoria

Tabla 6 CUS Administrar Tema de Transmisión

4.2.5 CUS: Administrar Formato de Fecha y Hora.

Secciones

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Seleccionar Idioma de la Fecha.	EC 1.1: Seleccionar Idioma de la Fecha correctamente	El usuario selecciona el idioma en que se va a transmitir la fecha.	El sistema guarda el idioma seleccionado y muestra un mensaje confirmando el idioma seleccionado.	Satisfactoria
SC2: Seleccionar Formato de la Fecha.	EC 2.1: Seleccionar Formato de la Fecha correctamente	El usuario selecciona el formato en que se va a transmitir la fecha.	El sistema muestra ejemplos del formato seleccionado en ventanas emergentes y guarda el seleccionado y muestra un mensaje	Satisfactoria

Capítulo 4: Pruebas

	.		confirmando el formato seleccionado.	
SC3: Seleccionar Formato de la Hora.	EC 3.1: Seleccionar Formato de la Hora correctamente	El usuario selecciona el formato en que se va a transmitir la hora.	El sistema muestra ejemplos del formato seleccionado en ventanas emergentes y guarda el seleccionado y muestra un mensaje confirmando el formato seleccionado.	Satisfactoria

Tabla 7 CUS Administrar Formato de Fecha y Hora

4.2.6 CUS: Seleccionar Plantilla de Administración.

Secciones

Nombre de la sección	Escenario	Acción realizada	Respuesta del sistema	Resultado de la prueba
SC1: Seleccionar Plantilla de Administración.	EC 1.1: Seleccionar Plantilla de Administración correctamente.	El usuario selecciona la plantilla de administración que desea utilizar.	El sistema muestra un mensaje confirmando la plantilla seleccionada.	Satisfactoria

Tabla 8 CUS Seleccionar Plantilla de Administración.

Luego de realizar varias iteraciones de las pruebas que se describen anteriormente, se observa la disminución gradual de errores que afectaban el flujo de trabajo del módulo para la personalización. En las tablas se reflejan los resultados finales luego de ser corregido el código fuente, demostrando un satisfactorio resultado obteniéndose un producto de mayor calidad.

4.3 Conclusiones parciales.

En el presente capítulo se analizaron diferentes aspectos como el significado de las pruebas de software, sus objetivos y alcance. Se diseñaron casos de pruebas específicos para los principales algoritmos, comprobándose que el flujo de trabajo de los mismos estuvo correcto ya que cumplieron con las condiciones necesarias que se habían planteado siendo positiva la validación de la solución propuesta.

Conclusiones

Conclusiones

Al concluir el presente trabajo se confirma la necesidad y la importancia de lograr el cumplimiento del objetivo general de implementar el módulo de personalización de la interfaz de usuario de la Plataforma de Televisión Informativa Primicia. Ayudando a resolver de modo eficiente los problemas referentes al proceso de personalización para cualquier canal cliente se plantea como conclusión que:

- ✓ Se analizaron soluciones existentes descubriendo deficiencias en cuanto a que sus niveles de personalización no llegan a la profundidad requerida por la interfaz de usuario de Primicia.
- ✓ Gracias a la utilización de RUP como metodología de desarrollo de software se obtuvo una amplia documentación para posteriores mejoras al módulo.
- ✓ El uso de las tecnologías y herramientas propuestas por el documento de Arquitectura del proyecto Primicia posibilitó dar solución al problema planteado y estar acorde con las políticas de migración hacia el software libre.
- ✓ Se mostraron los aspectos más significativos de la solución propuesta como la descripción de la implementación a través de diagramas que muestran la conformación del módulo en distintos componentes y el estándar de codificación utilizado evidenciando la obtención de un producto funcional a partir de los requisitos propuestos por los analistas.
- ✓ Con la validación de la solución propuesta mediante el diseño y aplicación de las pruebas de caja negra se pudo comprobar la calidad de la solución propuesta arrojando resultados positivos.
- ✓ Se logró modificar la interfaz de usuario de la Plataforma de Televisión Informativa Primicia con una mayor adaptación al cliente final.

Por todo lo antes mencionado se evidencia el cumplimiento de los objetivos propuestos en el presente trabajo de diploma lo cual conlleva al cumplimiento del objetivo general.

Recomendaciones

Recomendaciones.

Las recomendaciones propuestas para la continuidad del presente trabajo son:

- Ampliar las funcionalidades del módulo con los nuevos requerimientos que surjan por necesidades del cliente.
- Poner este documento al alcance de todos como material de estudio, guía y apoyo para su posterior continuación.

Bibliografía

Bibliografía

Agencia de diseño y comunicación gráfica. 2009. Almas de papel. [En línea] 2009.
<http://www.almasdepapel.com/conceptos/identidad-e-identidad-visual.html>.

Alonso García, Yanoisis. 2010. *Diseño del Módulo de Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA*. Ciudad de la Habana : UCI, 2010.

Bermúdez Valdés, Geovanys. 2010. *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de la Habana : UCI, 2010.

Booch, Rumbaugh y Jacobson. 2000. *El Lenguaje Unificado de Modelado*. s.l. : Addison Wesley, 2000.

Cachero Castro, Cristina. 2001. *Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales*. 2001.

Chamba, Julio. 2010. Herramientas CASE. [En línea] 2010. [Citado el: 22 de Octubre de 2010.]
<http://www.scribd.com/doc/40173039/Herramientas-Case>.

Eguíluz Pérez, Javier. 2008. *Introducción a CSS*. 2008.

—. 2008. *Introducción a Javascript*. 2008.

—. 2008. *Introducción a XHTML*. 2008.

Estructure Media Systems. 2001-2011. Estructure. [En línea] 2001-2011.
<http://www.estructuretv.com/enewsroom.htm>.

Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'. 2008. *Learning Ext JS*. Birmingham - Mumbai : Packt Publishing Ltd., 2008.

Garcia, Jesus. 2010. *EXT JS in Action*. 2010.

Hernández García, Ruber y Montaner Hernandez, Yunior. 2008. *Primicia: Plataforma de Televisión Informativa*. Ciudad de la Habana : UCI, 2008.

Hooping Publicidad S.L. 1995-2008. Hooping.net. [En línea] 1995-2008.
<http://www.hooping.net/glossary/aplicaciones-web-146.aspx>.

IDG Communications. 2001. *Técnicas y modelos de personalización de sitios web*. 2001. iWorld, La revista de la tecnología y estrategia de negocio en Internet.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley, 2000.

Molpeceres, Alberto. 2002. *Procesos de desarrollo: RUP, XP y FDD*. 2002.

Bibliografía

NetBeans Team. 2011. NetBeans. [En línea] Oracle Corporation, 2011.

http://netbeans.org/community/releases/69/index_es.html.

Potencier, Fabien y Zaninotto, François. 2008. *Symfony la guía definitiva*. 2008.

Prototype Core Team. 2006-2007. Prototype. [En línea] 2006-2007. <http://prototypejs.org/>.

Revista RED. 2003. CiberHábitat. [En línea] Febrero de 2003.

<http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/museo/cerquita/redes/medios/intro.htm>.

The jQuery Project. 2010. JQuery. [En línea] 2010. http://docs.jquery.com/Main_Page.

UMSNH. 2000. Facultad de Ciencias Físicos-Matemáticas. [En línea] 2000. Universidad Michoacana de San Nicolás de Hidalgo, 2010. <http://www.fismat.umich.mx/~crivera/tesis/node11.html>.

Van Der Henst S., Christian. 2011. Maestros del Web. [En línea] Maestros del Web Corp., 2011.

<http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>.

Visual Paradigm. 1999-2011. Visual Paradigm. [En línea] 1999-2011. <http://www.visual-paradigm.com/>.

Zend Technologies Ltd. 2006-2011. Zend Framework. [En línea] 2006-2011. [Citado el: 1 de Marzo de 2011.] <http://framework.zend.com/manual/en/coding-standard.html>.

Referencias bibliográficas

Referencias bibliográficas

Agencia de diseño y comunicación gráfica. 2009. Almas de papel. [En línea] 2009.
<http://www.almasdepapel.com/conceptos/identidad-e-identidad-visual.html>.

Alonso García, Yanoisis. 2010. *Diseño del Módulo de Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA*. Ciudad de la Habana : UCI, 2010.

Bermúdez Valdés, Geovanys. 2010. *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de la Habana : UCI, 2010.

Booch, Rumbaugh y Jacobson. 2000. *El Lenguaje Unificado de Modelado*. s.l. : Addison Wesley, 2000.

Cachero Castro, Cristina. 2001. *Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales*. 2001.

Chamba, Julio. 2010. Herramientas CASE. [En línea] 2010. [Citado el: 22 de Octubre de 2010.]
<http://www.scribd.com/doc/40173039/Herramientas-Case>.

Eguíluz Pérez, Javier. 2008. *Introducción a CSS*. 2008.

—. 2008. *Introducción a Javascript*. 2008.

—. 2008. *Introducción a XHTML*. 2008.

Estructure Media Systems. 2001-2011. Estructure. [En línea] 2001-2011.
<http://www.estructuretv.com/enewsroom.htm>.

Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'. 2008. *Learning Ext JS*. Birmingham - Mumbai : Packt Publishing Ltd., 2008.

Garcia, Jesus. 2010. *EXT JS in Action*. 2010.

Hernández García, Ruber y Montaner Hernandez, Yunior. 2008. *Primicia: Plataforma de Televisión Informativa*. Ciudad de la Habana : UCI, 2008.

Hooping Publicidad S.L. 1995-2008. Hooping.net. [En línea] 1995-2008.
<http://www.hooping.net/glossary/aplicaciones-web-146.aspx>.

IDG Communications. 2001. *Técnicas y modelos de personalización de sitios web*. 2001. iWorld, La revista de la tecnología y estrategia de negocio en Internet.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley, 2000.

Molpeceres, Alberto. 2002. *Procesos de desarrollo: RUP, XP y FDD*. 2002.

Referencias bibliográficas

NetBeans Team. 2011. NetBeans. [En línea] Oracle Corporation, 2011.

http://netbeans.org/community/releases/69/index_es.html.

Potencier, Fabien y Zaninotto, François. 2008. *Symfony la guía definitiva*. 2008.

Prototype Core Team. 2006-2007. Prototype. [En línea] 2006-2007. <http://prototypejs.org/>.

Revista RED. 2003. CiberHábitat. [En línea] Febrero de 2003.

<http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/museo/cerquita/redes/medios/intro.htm>.

The jQuery Project. 2010. JQuery. [En línea] 2010. http://docs.jquery.com/Main_Page.

UMSNH. 2000. Facultad de Ciencias Físicos-Matemáticas. [En línea] 2000. Universidad Michoacana de San Nicolás de Hidalgo, 2010. <http://www.fismat.umich.mx/~crivera/tesis/node11.html>.

Van Der Henst S., Christian. 2011. Maestros del Web. [En línea] Maestros del Web Corp., 2011.

<http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>.

Visual Paradigm. 1999-2011. Visual Paradigm. [En línea] 1999-2011. <http://www.visual-paradigm.com/>.

Zend Technologies Ltd. 2006-2011. Zend Framework. [En línea] 2006-2011. [Citado el: 1 de Marzo de 2011.] <http://framework.zend.com/manual/en/coding-standard.html>.

Glosario de términos

Glosario de términos

Ajax: (Asynchronous JavaScript And XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

DOM: (Document Object Model), es el Modelo de Objetos del Documento para acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML.

Elementos configurables: Conjunto de elementos a los que se les puede personalizar el diseño de su interfaz.

Elementos de identificación: Conjunto de elementos propios de cada canal o empresa por los cuales la misma es reconocida ya sean logotipos, videos de presentación o video de fin de transmisión.

Iconografía: Conjunto de imágenes gráficas con las que se representan las secciones dentro de la aplicación.

Infocinta: Cintillo informativo que se muestra en las pantallas de tipo texto, imagen o texto-imagen, promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes.

JavaScript: Lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

JQuery: Biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Panel informativo: Un panel informativo no es más que un área rectangular con una imagen de fondo en el cual contiene recursos informativos de interés como pueden ser el clima y la hora.

Plantillas de administración: Forma en la que se muestra todo el entorno administrativo del sistema y sus componentes visuales.

Plantillas de transmisión: Estructura y orden que poseen los componentes a mostrar en una noticia. Puede poseer un banner, un panel informativo, áreas de texto, imágenes y videos.

Glosario de términos

Prototype: Framework escrito en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones web. Es una herramienta que implementa las técnicas AJAX.

Recursos visuales: Elementos que se enmarcan dentro de un panel informativo que ofrecen cierta información al usuario ya sea sobre el clima, la hora u otros elementos.

XMLHttpRequest (XHR): (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

YUI: El framework o librería YUI (Yahoo User Interface) es un conjunto de utilidades y controles escritos en JavaScript que se utilizan para crear aplicaciones web dinámicas complejas.