



Universidad de las Ciencias Informáticas

Facultad 9

**Diseño del Módulo de Personalización de la interfaz gráfica de la
Plataforma de Televisión Informativa PRIMICIA.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TITULO DE
INGENIERO EN CIENCIAS INFORMATICAS**

Autor:

Yanoisis Alonso González

Tutor:

Ing. José Andrés Hernández Bustio

Ciudad de La Habana, Junio 2009

“Año 51 de la Revolución”



Dedicatoria



A mis sobrinos queridos para que se esfuercen mucho y se empeñen en lograr todo lo que se propongan en la vida y a mis padres por todo el amor, apoyo y confianza que me brindaron siempre.





Agradecimientos



Les agradezco a mis padres pues gracias a ellos y a sus innumerables sacrificios es que he logrado llegar hasta aquí, a mi hermana Yamirka que siempre estaba preocupada por mí y mis estudios, a mi familia en general por interesarse por el estado de la tesis.

A mis amigos de todos los grupos por donde he pasado que ya perdí la cuenta, los que han compartido fiestas, locuras, problemas, madrugadas, dentro de los que tengo que destacar a Karelia que la pobre ha tenido que cargar conmigo desde primer año, aunque creo que en realidad fue una suerte para ella estar en todos los grupos en los que yo caía.

A mi grupo actual, la gente de la cueva: Jean, el Mello siempre enredado con su tesis pero dispuesto a ayudar a los demás, Daniel siempre tratando de que todos tengan las cosas como deben ser y no como las quieren hacer, El Chino, Olivier, Yumar de todos el que más provecho sacó de estudiar en La Habana, Gilberto, Eyonis, Jasón, a Ariadna la pobre siempre con su estrés y sus problemas, a las otras chicas que integraban el equipo de revisión ortográfica de este documento: Dayamí que la vengo arrastrando desde la vocacional y es la persona más peleona que he conocido pero yo sé que me quiere muchísimo, Yakelin y Zuleira.

Un eterno agradecimiento al equipo de PRIMICIA o mejor dicho a la familia de tutores Carlos, Lizandra y Elaine.

A todos los que en el barrio preguntaron: ¿Hey cuando te gradúas, cómo va la tesis, te mudaste pá La Habana?

En fin le agradezco de corazón a todos los que he conocido a lo largo de estos cinco magníficos años que no olvidaré y sé que es posible que a la mayoría no pueda volver a ver pero siempre estarán en mi





Agradecimientos

memoria a cada momento en cada acción siempre existirá algo que me recuerde uno de esos momentos que vivimos en la universidad de las colas interminables o cambios imprevistos, UCI.





Declaración de Autoría



Yo Yanoisis Alonso González declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la Facultad 9 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 _____.

Yanoisis Alonso González José A. Hernández Bustio Jean M. Suarez Pérez





Datos de Contactos



Tutor:

Ing. José Andrés Hernández Bustio

Graduado de Ciencias Informáticas

e-mail: jahernandez@uci.cu

Cargo: Responsable de Área Temática en el Departamento de Señales Digitales.

Profesor Instructor con dos años de experiencia docente en la Universidad de las Ciencias Informáticas.





Desde los inicios de la sociedad el hombre ha tenido la necesidad de comunicarse, y según este evolucionaba lo hacían también los medios de comunicación. Actualmente una de las formas de comunicación predilectas es la televisión la cual ha sufrido algunos cambios en la era de las computadoras pues pasó de la tradicional TV analógica a la digital. Hoy en día existen diversas empresas que mediante el uso de internet abastecen a sus usuarios de un sinnúmero de canales televisivos a través de la web.

La Universidad de las Ciencias Informáticas y más específicamente el Departamento de Señales Digitales ha alcanzado admirables resultados en el campo de la transmisión televisiva uno de los cuales es la plataforma de televisión informativa PRIMICIA. Esta permite la transmisión automática, inmediata y constante de un ciclo noticioso.

Debido al cambiante mundo de la web y la necesidad de lograr un producto con el cual el cliente se sintiera más a gusto y que pudiera ser adaptado de forma dinámica a sus necesidades se propone el diseño de un módulo que permita la personalización automática de la interfaz gráfica de la aplicación. En el documento se realiza un estudio de los elementos puntuales en los que se basan los clientes para ubicar un sitio web dentro de su preferencia, donde desempeña un papel fundamental la personalización y se generan una serie de artefactos definidos por la metodología de desarrollo seleccionada de acuerdo con el rol de analista.





Índice de Contenido.

Introducción.....	1
Capítulo 1	5
Fundamentación Teórica.....	5
1.1 Introducción.....	5
1.2 Plataforma de Televisión Informativa. PRIMICIA.....	5
1.3 Aplicación Web.....	5
1.4 Interfaz de usuario	6
1.5 Interfaz Gráfica.....	6
1.6 Procesos relacionados con la Plataforma de Televisión Informativa PRIMICIA. Descripción General.....	6
1.6.1 Procesos relacionados con la Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA. Características generales.....	8
1.6.2 Situación Problemática.....	10
1.7 Análisis de otras soluciones existentes.....	11
1.7.1 Sistemas de televisión informativa existentes.....	13
1.8 Caracterización del diseño y la arquitectura de software de la Plataforma.....	16
1.9 Conclusiones.....	18
Capítulo 2.....	19
Tendencias y Tecnologías Actuales	19





2.1	Introducción.....	19
2.2	Metodologías de Desarrollo.....	19
2.2.1	Metodologías ágiles.....	20
2.2.2	Metodologías pesadas.....	22
2.3	Lenguaje de modelado.....	26
2.4	Herramienta de Modelado.....	26
2.4.1	Rational Rose.....	26
2.4.2	Visual Paradigm.....	26
2.5	Framework JavaScript.....	28
2.6	Conclusiones.....	29
	Capítulo 3.....	31
	Presentación de la Solución Propuesta.....	31
3.1	Introducción.....	31
3.1.1	Modelo de Dominio.....	31
3.1.2	Glosario de Términos del Dominio.....	32
3.2	Requerimientos Funcionales.....	33
3.3	Requerimientos no Funcionales.....	36
3.3.1	Software.....	36
3.3.2	Hardware.....	36
3.3.3	Apariencia o interfaz externa.....	37
3.3.4	Confidencialidad.....	37
3.3.5	Disponibilidad.....	38
3.3.6	Usabilidad.....	38





3.3.7	Soporte.....	38
3.3.8	Restricciones en el diseño y la implementación.	38
3.4	Descripción del Sistema Propuesto.....	38
3.4.1	Descripción de los actores.....	38
3.4.2	Diagrama de Casos de Uso del Sistema.	39
3.4.3	Descripción de los Casos de uso del Sistema.....	40
3.5	Conclusiones.....	44
Capítulo 4	44
Construcción de la Solución Propuesta	44
4.1	Introducción.....	44
4.2	Modelo de análisis.	44
4.2.1	Diagramas de Clases del análisis.	44
4.3	Modelo de diseño.	47
4.3.1	Patrones de diseño.	47
4.3.2	Diagramas de Clase del Diseño.....	50
4.3.3	Diagrama de clases persistentes.	59
4.4	Base de Datos.....	59
4.4.1	Modelo de Datos.....	60
Conclusiones	60
Conclusiones Generales	61
Recomendaciones	62
Bibliografía Citada	63





Índice de Contenido



Bibliografía Consultada.....	63
Glosario de Términos.....	66





Introducción

El progresivo avance de las tecnologías de información y las comunicaciones (TICs) y la introducción de estas en ámbitos de la vida cotidiana han producido cambios en disímiles sectores de la sociedad, sin dudas los medios audiovisuales como la televisión constituyen un ejemplo de esto. La televisión desde sus orígenes se ha convertido en el medio de comunicación por excelencia y ha ido evolucionando de forma favorable desde la televisión analógica hasta la digital. Con la aparición de esta última, surgen nuevas formas de transmisión donde con la ayuda de las redes y otros medios informáticos se puede hacer llegar la señal televisiva de una forma más rápida y eficaz que con la transmisión tradicional.

La Universidad de las Ciencias Informáticas (UCI) ha avanzado considerablemente en el campo de la transmisión de señales de televisión desde sus orígenes. Partiendo de la necesidad real de mantener informada y actualizada a toda la comunidad universitaria en todo lo referente al ámbito nacional, internacional así como de otros temas de interés, surge la idea de desarrollar un producto que fuera capaz de transmitir noticias en distintos formatos Texto, Texto - Imagen, Imagen y Video, utilizando la red televisiva de docentes y áreas de la residencia.

Luego de un conjunto de investigaciones realizadas para materializar esta idea, se crea Señal 3, el canal informativo de la universidad, posteriormente se desarrolla Señal ACN, que surge a raíz de las misiones internacionalistas de los médicos cubanos. Señal ACN nace con el objetivo de mantenerlos informados hasta en los lugares más recónditos del planeta y llevar la información a las distintas zonas de silencio que existen en Cuba. Después de Señal ACN se crea TV Energía, esta última tuvo como novedad la migración completa a software libre a diferencia de sus predecesores creados en software propietario.

Realizando un estudio sobre la posibilidad de introducir un producto de este tipo en el mercado internacional, dando un aporte no solo económico y social al país sino también científico tecnológico, se llega a la conclusión de crear la Plataforma de Televisión Informativa PRIMICIA. Ésta estaba encargada de la transmisión inmediata y constante de información en distintos formatos logrando de esta manera que el público objetivo esté actualizado en lo que a información respecta. PRIMICIA, por la sencillez y la





rapidez de transmisión está dirigida principalmente a universidades, centros de convenciones, hoteles, terminales de transportación o sedes empresariales.

Como cada institución puede tener la necesidad de ajustar el diseño de la aplicación según la identidad gráfica definida por la misma, es de suma importancia que en esta se pueda personalizar su interfaz gráfica de modo que el cliente la adapte a sus necesidades.

Actualmente el producto PRIMICIA no cuenta con un conjunto de funcionalidades que permitan la personalización de la interfaz gráfica de los distintos subsistemas que la componen. Cualquier cambio necesario a la hora de comercializar el producto con diferentes empresas o instituciones tiene que ser implementado de forma manual por el equipo de desarrollo. Esto sin dudas constituye un gran inconveniente pues siempre que se vaya a comercializar el producto con un cliente nuevo se tiene que reunir a dicho equipo y adaptar el diseño de la interfaz gráfica de la plataforma según sus necesidades no permitiéndole al mismo que pueda realizar los cambios en la interfaz de forma automática.

Teniendo en cuenta la situación problemática antes planteada, ha sido identificado como **problema científico de la investigación**: ¿Cómo permitir la personalización automática de la interfaz gráfica en la Plataforma de televisión informativa PRIMICIA?

Debido al problema antes expuesto se pretende realizar el análisis y diseño de un módulo que contenga las funcionalidades necesarias para permitir la personalización de la interfaz gráfica en la aplicación permitiendo que esta se adapte a las necesidades del cliente de forma rápida y sencilla.

Dando seguimiento al problema planteado se tiene como **objeto de estudio** los Procesos relacionados con la Plataforma de Televisión Informativa PRIMICIA.

Se determinó como **campo de acción** la personalización de la interfaz gráfica de la Plataforma de televisión Informativa PRIMICIA y así demostrar la **hipótesis** de que si se logra una correcta caracterización de los procesos relacionados con la personalización de la interfaz gráfica de la Plataforma de Televisión Informativa, se podrá diseñar un módulo que permita la personalización del sistema.





En este estudio se tiene como **objetivo general**, diseñar un módulo para la Plataforma de Televisión Informativa PRIMICIA que permita la personalización de la interfaz gráfica de los subsistemas que la componen.

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

- ✓ Caracterizar los procesos relacionados con la personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA a partir de la conceptualización realizada.
- ✓ Caracterizar el diseño y la arquitectura de software de la Plataforma.
- ✓ Capturar los requisitos vinculados a los procesos de personalización del diseño gráfico de PRIMICIA.
- ✓ Identificar las metodologías y herramientas a utilizar en el desarrollo del diseño del módulo.
- ✓ Diseñar las funcionalidades identificadas para la plataforma.

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 cuales han sido especificados a continuación:

Métodos Teóricos

Analítico – Sintético: Permitirá hacer un estudio detallado de todos los elementos relacionados con la aplicación al dividir la investigación en distintas partes para llevarla a cabo, donde posteriormente, delimitando las relaciones que existen entre ellas, unir las como un todo.

Histórico – lógico: Para analizar la evolución de los sistemas de televisión informativa aprovechando los elementos teóricos que aportaran para el desarrollo de la aplicación.

Modelación: Para la realización de diagramas que permitan lograr un mejor entendimiento de lo que se va a implementar.

Métodos Empíricos





Introducción



Observación: Para percibir directamente el desarrollo de la investigación, entiéndase como el registro visual que garantizará lo que en realidad ocurre en el avance de la aplicación.

El documento presentado está compuesto por 4 capítulos de los cuales el primero tratará aspectos relacionados con la fundamentación teórica donde se plantean los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. En el Capítulo 2 se expondrán las tendencias tecnológicas actuales detallando algunos de los principales conceptos y tecnologías que se adecuan a la solución planteada. En el Capítulo 3 se presenta la solución propuesta mientras que en el 4 se describe la misma.





Capítulo 1

Fundamentación Teórica

1.1 Introducción.

En el siguiente capítulo se tratan una serie de conceptos relacionados con el objetivo general del trabajo investigativo realizado y con el problema planteado anteriormente. Además se describe de manera detallada la situación problemática expuesta. Aquí se describen los procesos de la plataforma PRIMICIA que tienen que ver con la personalización de la interfaz gráfica de la misma.

Para un mayor entendimiento del trabajo realizado se hace necesario el dominio de algunos conceptos tratados en el cuerpo del documento a los cuales se le hace referencia a continuación.

1.2 Plataforma de Televisión Informativa. PRIMICIA.

PRIMICIA es un producto creado por la UCI en el Departamento de Señales Digitales de la Facultad 9. Se trata de una aplicación web realizada con herramientas de software libre capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos. El sistema está orientado a suministrar un canal de televisión básicamente con características informativas, donde se integran contenidos de texto, imagen fija y video. (1)

Las características de PRIMICIA hacen que pueda ser ajustable a diferentes clientes, por lo que constituye un producto informático del cual pueden beneficiarse televisoras, agencias de noticias, empresas, terminales de transporte, hoteles y cualquier otra entidad que tenga una red de televisión con necesidades de transmitir informaciones en distintos formatos. (2)

1.3 Aplicación Web.

Una aplicación web no es más que un software que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una red local mediante un navegador. En otras palabras, es un software que





se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener páginas web sin distribuir e instalar software a miles de usuarios. Una Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el cliente acceda a los datos de modo interactivo, gracias a que la página responderá a cada petición realizada por éste y sin que el mismo tenga que conocer elementos como su código o el lenguaje de programación usado para la confección de la misma.

1.4 Interfaz de usuario

Las interfaces de usuario son utilizadas en multitud de sistemas y hacen la vida más cómoda y segura. Los ingenieros y diseñadores industriales han venido trabajando en este campo desde hace mucho tiempo, creando sistemas de comunicación adecuadas entre los objetos que crean y los usuarios de los mismos. Interfaces de usuario de este tipo pueden ser, por ejemplo, el sistema de botones de un televisor, el mando a distancia que lo controla o los sistemas de puesta en hora de un reloj. (3)

1.5 Interfaz Gráfica.

Una interfaz gráfica es cualquier medio por el cual se puede interactuar con una computadora a través de algún tipo de software gráfico. Comúnmente, esto se consigue a través del control mediante dispositivos de entrada a la computadora como el teclado y el mouse mediante menús, ventanas, íconos y cajas de diálogo. (3)

1.6 Procesos relacionados con la Plataforma de Televisión Informativa PRIMICIA. Descripción General.

La Plataforma de Televisión Informativa PRIMICIA está diseñada para la transmisión cíclica de noticias en formato texto, texto-imágenes y video, acompañado siempre de un fondo musical. El sistema se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final a través de una red de televisión. El Subsistema de Administración es el responsable de





que a través de él se realice la administración del canal y toda la gestión de las noticias y recursos multimedia. El Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados. (1)

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.
- ✓ Gestión de las sesiones temáticas del canal, establece el orden de las secciones, horario en que serán mostradas y habilitarlas o deshabilitarlas.
- ✓ 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 publicación; 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.
- ✓ Almacenamiento, administración y reproducción de recursos multimedia como imágenes, música y video.
- ✓ 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.
- ✓ 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.
- ✓ Administración de la señal del canal que permite cambiar entre la señal de televisión en vivo y la señal del canal.





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.
- ✓ Visualización de noticias compuestas por pantallas de tipo Texto, Texto-Imagen, Imagen y Video.
- ✓ Reproducción de un fondo musical mientras se muestran las noticias, excepto cuando se muestre un Video.
- ✓ Presentación en las pantallas de tipo Imagen un comentario que oriente al televidente acerca de lo que está observando. (1)

1.6.1 Procesos relacionados con la Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA. Características generales.

En la Plataforma de Televisión Informativa PRIMICIA existen diferentes procesos que tienen que ver con la personalización de la interfaz gráfica del sistema. A continuación dichos procesos se exponen por separados para ambos subsistemas ya que no presentan las mismas características y por ende las mismas funcionalidades. Dichos procesos se encuentran vinculados a los elementos del sistema a los que se le pretende realizar la personalización de su interfaz gráfica de forma dinámica.

Gestión del Diseño Gráfico, Subsistema de Administración:

Gestionar el logotipo principal de la plantilla.

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

Personalizar el formato de la fecha y hora del sistema.

La fecha y hora del sistema puede presentarse en varios formatos. El sistema de administración no brinda la posibilidad de escoger otro formato diferente al establecido.





Gestión de las plantillas de administración.

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

Gestión de íconos.

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

Gestión de Infocintas.

Actualmente la plataforma presenta las infocintas de una sola forma, estilo, posición y efecto, imposibilitando que las mismas sean lo más genéricas posibles.

Gestión del Diseño Gráfico, Subsistema de Transmisión:

Gestión de plantillas de transmisión.

El subsistema de transmisión solo cuenta con una plantilla para la visualización de las noticias y demás informaciones que brinda dicho subsistema. 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.

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.

Personalización de los elementos de identificación del canal.

Los elementos de identificación, como son los logotipos, video de presentación, 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.

1.6.2 Situación Problemática.

La Universidad de las Ciencias Informáticas (UCI) surgida, producto a la batalla de ideas desarrollada en el país, en 2002 constaba desde sus inicios con una inmensa masa de estudiantes y profesores a los cuales era necesario hacerles llegar informaciones de ámbito nacional e internacional y noticias referentes a la propia universidad. Por este motivo y gracias a las tareas investigativas y productivas de los integrantes del proyecto UCITeVe, 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 (1)

Con similares características se desarrolla en el 2006 el canal informativo de la Agencia Cubana de Noticias, Señal ACN. Este sistema automatizado de teletexto para la Plataforma de Televisión Satelital Cubana facilita la transmisión de noticias de la prensa cubana a los cooperantes que cumplen misión en el extranjero y a los habitantes en lugares intrincados de la Isla, en las llamadas zonas de silencio. Para la realización de estas soluciones se emplea software propietario disponible solo para sistemas operativos de la familia Windows como SQL Server y Director, propiedad en aquel momento de Macromedia y actualmente de *Adobe Systems* (2)

Con la aparición de una importante negociación con un cliente que deseaba un producto similar pero desarrollado con herramientas no propietarias, nace TVEnergía. En ésta a diferencia de Señal 3 y Señal ACN, y como resultado del trabajo sobre la línea de investigación de sistemas informáticos para la transmisión de radio y 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 (2)





Con el surgimiento de TVEnergía y tomando las ventajas de la utilización de herramientas libres se da inicio a la conceptualización de PRIMICIA. La Plataforma de Televisión Informativa PRIMICIA está diseñada también con herramientas totalmente libres como: Nova/Linux, Visual Paradigm, Symfony, PostgreSQL, PHP, Java Script y HTML, donde 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 su interfaz gráfica no es personalizable de manera que es necesario reunir al equipo de trabajo en caso de que se quiera comercializar el producto con diferentes clientes y estos deseen como es de suponer una interfaz personalizada.

1.7 Análisis de otras soluciones existentes.

En la actualidad términos como web, hipertexto y multimedia son muy usados en el desarrollo de sistemas informáticos. Con el uso de las conexiones entre los ordenadores se obtienen valiosas ventajas cuando se trata del trabajo colaborativo y en este caso los aportes de las aplicaciones web son numerosos. No obstante el clima que envuelve dichas aplicaciones es muy cambiante.

En los sistemas web los aspectos como el diseño gráfico, la calidad cognitiva 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. (4)

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. (5)





En un estudio elaborado por Rubric (empresa comercializadora de diferentes productos) el 90% de los compradores indicaron que la personalización podría incentivarles a comprar de nuevo en un mismo web, y el 94% dijeron que serían más receptivos a recomendaciones basadas en sus perfiles e históricos de compras. (5)

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. Unido a esto se encuentran algunos elementos que no pueden faltar a la hora de crear un sistema informático web dentro de los cuales se encuentran:

- ❖ **Facilidad de uso**, el objetivo en ningún momento es realizar un software en el que solo pueda trabajar personal especializado en la rama de la informática o que las personas que interactúen con él tengan que leer grandes guías de usuario para poder obtener los beneficios que él brinda.
- ❖ **Respuesta mínima**, responder en el menor tiempo posible, teniendo en cuenta todos los requerimientos que hayan sido identificados con los usuarios.
- ❖ **Seguridad**, la información debe estar protegida en todo momento evitando que pueda ser accedida, manipulada, distribuida o cambiada por personas no autorizadas.
- ❖ **Actualizados**, cuando se habla de este tema se tienen en cuenta las herramientas y técnicas que se utilizan para la construcción de la aplicación.

Después de analizar cada uno de estos elementos no se debe pasar por alto los componentes que agilizan todo el proceso de navegación del usuario por la aplicación teniendo en cuenta que este puede no poseer ningún conocimiento informático, dentro de estos se destacan:

Página principal (portada, principal, homepage, home).

Menú principal.

Es el catálogo o relación de programas y procedimientos que aparece en pantalla con el fin de que, usando un teclado, un dispositivo táctil, un lápiz óptico o un ratón, el operador pueda elegir qué opción





desea ejecutar. Los denominados menues desplegables son el tipo más habitual en la actualidad: muestran un Menú Principal y, tras una sencilla operación, sus diferentes opciones. (6)

Banners, promociones.

Un banner es un anuncio normalmente rectangular colocado en la parte superior, inferior o en los laterales del contenido principal de un sitio web.

Animaciones multimedia.

Como su nombre indica, multimedia es la integración de múltiples formas de medios de comunicación. Incluyendo archivos de audio, video y elementos gráficos. (7)

1.7.1 Sistemas de televisión informativa existentes.

Para la transmisión rápida de información haciendo uso de la TV como soporte y teniendo como base un sistema web automático, existen a nivel internacional diversas soluciones como teletexto digital que es un servicio de teletexto basado en XHTML y CSS que se difunden principalmente en las cadenas de televisión digital de países europeos, existen además las plataformas de televisión sobre IP, las cuales divulgan una gran gama de canales de televisión en la red. En el marco nacional también existen algunas soluciones ya mencionadas en capítulos anteriores como Señal 3 de la (UCI) y Señal ACN de la Agencia Cubana de Noticias (ACN), los cuales están orientados a brindar informaciones las 24 horas a través de un sistema de teletexto adaptado.

Plataformas de televisión sobre IP (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, simultáneamente y en tiempo real, acceder y generar informaciones para:

- ✓ Videoconferencias.
- ✓ Realización de encuestas en tiempo real, con resultados en forma de gráficos y datos.





- ✓ Uso de cuadro digital, con posibilidad de interacción entre todos los usuarios en sesión.
- ✓ Visualización colectiva de aplicativos (con transmisión de imagen del conferenciante).
- ✓ Capacidad de transmisión de vídeos pregrabados a partir de equipamientos como reproductores de DVD y VHS, filmadoras.
- ✓ Creación de *Playlist* o listas de reproducción a partir de vídeos precodificados.
- ✓ Chat público y posibilidad de intercambios de mensajes privados.
- ✓ Capacidad de participación de usuarios con diferentes disponibilidades de velocidad de transmisión (kbps) en una sesión de videoconferencia.

Adoptando la solución IPTV, el usuario pasa a disponer de un o más canales exclusivos para transmitir cualquier programación para puntos determinados por él. El número de puntos por ordenadores que pueden tener acceso a esta programación es ilimitado. (8)

Pero en un sistema de este tipo la adaptabilidad de la interfaz gráfica 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.

Estructuretv.

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. Permite además el ajuste dinámico del idioma del sistema. (9)

Esta solución no permite una personalización profunda en lo que a interfaz gráfica de usuario se refiere, simplemente permite la funcionalidad básica del cambio de idioma de la aplicación.





Sistemas de Teletexto en Europa.

Concretamente, este sistema de teletexto europeo es el precursor de los actuales productos multimedia. Básicamente, consiste en un conjunto (más o menos extenso) de páginas, las cuáles se eligen a través del mando a distancia. Se escribe el número de la página (de tres cifras) y transcurrido un tiempo, se muestra en la pantalla del televisor. Sólo permite transmitir información escrita y gráficos. (10)

El teletexto es un sistema que permite el envío de información utilizando parte de la señal televisiva. En los sistemas de teletexto el usuario solo se comporta como un espectador ya que únicamente puede seleccionar las informaciones que desea a partir de las que se le ofrecen, no puede realizar una petición concreta ya que no hay comunicación con la fuente de la información. Sólo permite transmitir información escrita y gráficos como se muestra en la Fig. 1.



Figura 1 Imagen del Sistema de Teletexto de Televisión Española.





Es decir, el teletexto es un medio informativo que utiliza la televisión para hacer llegar los mensajes a los posibles receptores pero sin ofrecer la posibilidad de que se dé un *feedback*¹ entre emisor y receptor. El número de páginas de información está limitado ya que el teletexto solo utiliza ocho líneas de imágenes. Así, la mayoría de los sistemas de teletexto que funcionan en Europa proporcionan de cien a mil páginas. La información queda limitada al espacio de la pantalla del televisor. (10)

En el proceso de caracterización se detectaron los elementos configurables de la plataforma a los que se le puede realizar de forma dinámica la personalización de su interfaz, llevando al entorno web aspectos vistos más comúnmente en aplicaciones de escritorio como la selección de iconos, fuente, formato de fecha y hora, y demás, similar a los sistemas operativos como Windows. Por lo que se plantea que no existe una solución íntegra que abarque totalmente el objetivo del módulo a desarrollar.

1.8 Caracterización del diseño y la arquitectura de software de la Plataforma.

El diseño y la arquitectura de software consisten en un conjunto de patrones y abstracciones que proporcionan un marco de referencia útil para guiar el proceso de creación del software, estableciendo los fundamentos para que los diseñadores, programadores y todo el personal de desarrollo en general trabajen por una misma línea, para así poder alcanzar los objetivos del sistema y que este se desarrolle de manera organizada.

El diseño y la arquitectura de la plataforma están estructurados de manera que ésta sea de fácil adaptación a diversos entornos teniendo en cuenta aspectos importantes como son el presupuesto con el que cuente el cliente o la cantidad de información que este pretenda generar. Es por ello que su arquitectura física consta de dos variantes, una variante es utilizar un servidor para la gestión de los procesos de administración y transmisión. La otra alternativa posible es la utilización de dos servidores especializados para la gestión independiente de los procesos de administración y transmisión.

¹ Retroalimentación, respuesta del receptor al emisor de un mensaje. De esta forma la comunicación es bidireccional e interactiva.





El desarrollo del sistema se realizó completamente utilizando software libre, siendo el resultado del empleo e integración de las siguientes herramientas y lenguajes de programación: el dotProject como herramienta de gestión de proyecto, Subversion para el control de versiones, Visual Paradigm para el modelado, como Entorno de Desarrollo Integrado (IDE) se utiliza Eclipse PDT, como framework de desarrollo se tiene a Symfony, sistema gestor de base de datos PostgreSQL y los lenguajes de programación son PHP, JavaScript, HTML. El patrón de arquitectura usado es el Modelo - Vista - Controlador (MVC) ya que este es el implementado por el framework sobre el que se trabaja.

El sistema cuenta con dos servidores, uno para la gestión de recursos, datos y administración del canal; y el segundo encargado de la transmisión. Se hará uso de un servidor FTP para subir los archivos multimedia tales como: músicas, imágenes y videos. El sistema solo manejará archivos de video en formato MPG, AVI, y MOV.

A continuación se muestra la distribución lógica del sistema:



Figura 2. Distribución lógica del sistema.

La arquitectura física del sistema puede ser acoplada en dependencia del tipo de red de televisión. Seguidamente se muestran los diagramas de instalación del sistema para una red satelital y cableada:



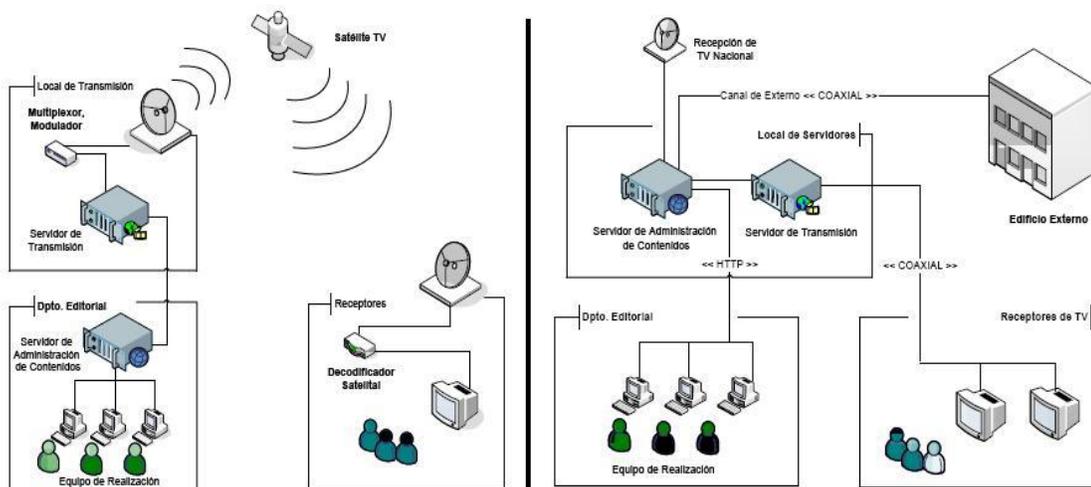


Figura 3. Distribuciones físicas de la Plataforma.

1.9 Conclusiones.

En este capítulo se realizó un análisis de los elementos teóricos que sirven como base al problema científico y a los objetivos planteados en el estudio realizado. Se puede sintetizar que para la optimización de la plataforma de televisión informativa es necesario lograr la correcta implementación de las funcionalidades detectadas en los elementos configurables del sistema, teniendo en cuenta la puntualización sobre los elementos técnicos imprescindibles para lograr una interfaz gráfica personalizable de forma óptima, ya sean menús, banners o animaciones. Con el análisis de las soluciones existentes se puede concluir además que no existe ninguna plataforma informativa que presente un nivel de personalización de la interfaz gráfica tal como el que se pretende alcanzar en la aplicación.





Capítulo 2

Tendencias y Tecnologías Actuales

2.1 Introducción.

En este capítulo se analizan detalladamente las tendencias y tecnologías actuales relacionadas con el uso de aplicaciones web. Además se define la metodología más apropiada para el análisis y diseño del módulo teniendo en cuenta las ventajas que podría aportar al desempeño del trabajo.

La creación de un software no solo se basa en la implementación a ciegas, también requiere de organización y es en dicha organización donde juega un papel fundamental la aplicación de metodologías de desarrollo que permitan la confección de un producto exitoso que cumpla con las expectativas del cliente y del equipo de desarrollo en general.

2.2 Metodologías de Desarrollo.

El proceso de desarrollo del software es una tarea que requiere de una rigurosa organización y planificación. Con el objetivo de lograr esto surgen las metodologías de desarrollo. Estas imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente, para lograrlo, realizan un proceso detallado con un fuerte énfasis en la planificación inspirado por otras disciplinas de la ingeniería.

Todo desarrollo de un proyecto es riesgoso y difícil de controlar, pero si no se tiene una metodología de por medio, solo se obtiene la insatisfacción de los clientes y de los propios desarrolladores con el resultado.

Las metodologías de desarrollo del software se dividen en dos grandes grupos, las metodologías pesadas y las ágiles, la aplicación de las primeras obliga a forzar al cliente a que este tome la mayoría de las decisiones al inicio del desarrollo del producto por lo que el coste de cambio de una decisión tomada





puede llegar a ser muy elevado, las segundas cambian significativamente algunos de los énfasis de los métodos ingenieriles.

La diferencia inmediata entre estas es que las metodologías ágiles son menos orientadas al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientadas al código pues estas se basan en la idea de que la parte importante de la documentación es el código fuente.

2.2.1 Metodologías ágiles.

Microsoft Solution Framework (MSF).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



Figura 4 Metodología MSF





MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación. (11)

MSF tiene las siguientes características:

- Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. (10)

XP

La metodología XP (*Extreme Programming*) se enfoca en el proceso de desarrollo de software y dirige las fases de análisis, desarrollo y pruebas para lograr la calidad en el producto final. La Programación Extrema surge ideada por Kent Beck², como proceso de creación de software diferente al convencional. Ésta es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y para desarrollar software. El principal objetivo de esta metodología se encuentra en fortalecer las relaciones en el equipo de desarrolladores como clave fundamental para el éxito del proyecto, promoviendo el trabajo en equipo, aumentando el nivel de aprendizaje de los desarrolladores, y proporcionado un buen clima de trabajo.

² **Kent Beck** fundador y director de *Three Rivers Institute (TRI)*. Algunas de sus contribuciones al desarrollo de software son: los patrones de software, la familia xUnit de herramientas de prueba para desarrolladores, y *Extreme Programming*.





Objetivos de XP:

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final del ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Bases de XP

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Lo que buscan en definitiva es la reducción de costos.

Sin embargo, XP presenta varios inconvenientes, en comparación con otras metodologías basadas en el documento, incluidos los problemas con requisitos inestables, inexistencia de la documentación y la falta de un diseño global o un documento de especificaciones. Lo que es muy desventajoso en XP es el hecho de evitar cualquier tipo de documentación fuera del código fuente. Esto hace difícil utilizar la experiencia ganada en proyectos anteriormente desarrollados, ya que no se han generado documentos y se tiene que comenzar a generar todo desde cero. (12)

2.2.2 Metodologías pesadas.

Métrica

Existen muchas metodologías encaminadas a desarrollar sistemas de información, entre estas se encuentra Métrica. Esta es una metodología española, elaborada por el Ministerio para las Administraciones Públicas. Métrica es una guía formal, aunque flexible en su utilización, para el diseño y construcción de sistemas de información, empleando conceptos y técnicas de ingeniería de sistemas de información y tecnología de la información. (13)

Esta metodología ofrece un marco de trabajo en el que se define:





- ✓ Una estructura de proyecto que sirve de guía al equipo de trabajo e involucra a los usuarios en su desarrollo y en sus puntos decisivos.
- ✓ Un conjunto de productos finales a desarrollar.
- ✓ Las diferentes responsabilidades y funciones de los miembros del equipo de proyecto y de los usuarios. (13)

En ella se describe en detalle la sucesión de pasos, estructurados en fases, módulos, actividades y tareas, así como los productos que se obtienen en cada uno de dichos pasos. Estos productos pueden ser productos finales o bien productos intermedios que servirán para la realización de algún paso posterior y, por último, describe la estructura final de la documentación obtenida. Métrica está dividida en cinco fases que se descomponen en siete módulos. Los módulos, a su vez, se descomponen en actividades y estas en tareas. (13)

Las fases en las que se divide Métrica son:

- Fase 0: Plan de sistemas de información.
- Fase 1: Análisis de sistemas.
- Fase 2: Diseño de sistemas.
- Fase 3: Construcción de sistemas.
- Fase 4: Implantación de sistemas.

RUP

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (14)



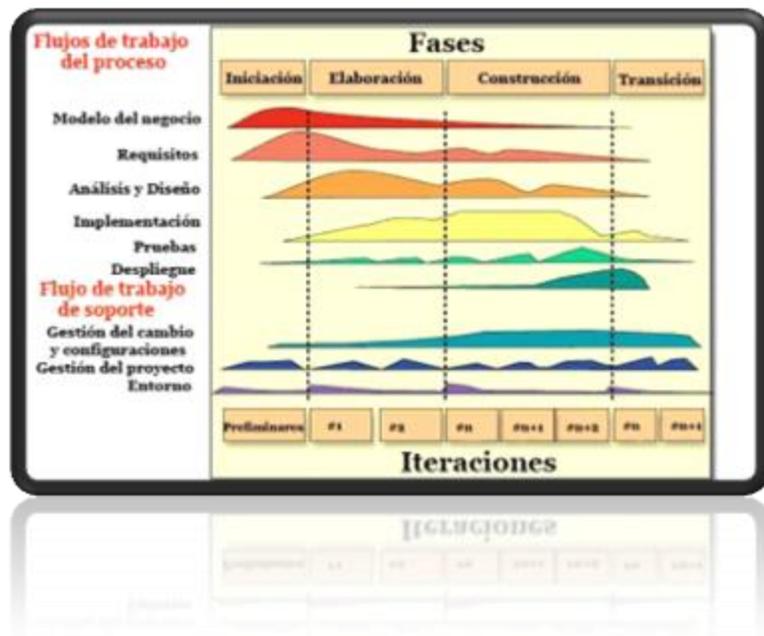


Figura 5 Faces de RUP.

Características principales de RUP

- ✓ **Centrado en los modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema. (15)
- ✓ **Guiado por los Casos de Uso:** Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba. (15)
- ✓ **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar. (15)
- ✓ **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo. (15)

Beneficios que aporta RUP.



- ✓ Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- ✓ Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- ✓ Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- ✓ Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- ✓ Proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web, pruebas y calidad. También proporciona guías para desarrollar en plataformas *IBM WebSphere* y *Microsoft Web Solution* para acelerar el desarrollo de los proyectos.
- ✓ Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado³ y un punto de vista de cómo desarrollar software.
- ✓ Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.
- ✓ No solo garantiza que los proyectos abordados serán ejecutados íntegramente sino que además evita desviaciones importantes respecto a los plazos.
- ✓ Permite una definición acertada del sistema en un inicio para hacer innecesarias las reconstrucciones parciales posteriores. (16)

³ **Lenguaje de modelado** es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un diseño de software orientado a objetos.





Estos aspectos son los que hicieron que RUP fuera la metodología seleccionada para el desarrollo del proyecto ya que para la confección del mismo se hace necesario soporte ingenieril fuerte con una documentación numerosa y organizada.

2.3 Lenguaje de modelado.

EI UML

El UML, por sus siglas en inglés, (*Unified Modeling Language*) es el lenguaje de modelado de sistemas de software para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, detallar los artefactos en la aplicación, documentar y construir de forma organizada los elementos en el desarrollo del proyecto.

2.4 Herramienta de Modelado.

2.4.1 Rational Rose.

Permite a los arquitectos y diseñadores producir modelos independientes de lenguaje en Unified Modeling Language (UML) de arquitectura de software, necesidades de negocios, activos re-utilizables y comunicaciones en nivel de gerencia, posee una experiencia de diseño única, su desarrollo es dirigido por modelo con soporte de UML, permite el soporte a múltiples modelos para Model-Driven Architecture (MDA), se ejecuta tanto stand-alone o integrado a Microsoft® Visual Studio.NET. Crea arquitecturas independientes de plataforma que se pueden implementar en plataformas Java y .NET. Sus estándares son definibles para usuarios para crear, personalizar y apalancar estándares de diseño arquitectónico. Las referencias de modelo cruzado y de realización de versiones hasta el nivel de diagrama y clase permiten estructuración para adecuarse a cualquier proyecto. Mantiene la posibilidad de rastreo entre modelos de análisis, diseño e implementación. La modelación es de forma libre y permite publicación e informes en Web. (17)

2.4.2 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software es decir las etapas de análisis, diseño, construcción, pruebas y despliegue. El software de





modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite la elaboración de todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. (18)



Figura 6 Visual Paradigm.

La herramienta Visual Paradigm está desarrollada sobre el lenguaje Java por lo que posee versiones capaces de trabajar en diferentes sistemas operativos. Es un producto colaborativo, por lo que permite que múltiples usuarios trabajen sobre el mismo proyecto a la vez lo cual es muy provechoso para el buen desempeño del módulo a realizar. Es capaz de generar la documentación del proyecto automáticamente en diversos formatos.

Luego de haber realizado un análisis de 2 de las herramientas CASE más potentes, es fácil notar que ambas poseen características que, aunque útiles para el desarrollo de cualquier software, las diferencian en gran medida, como es el caso de Visual Paradigm el cual es multiplataforma mientras el Rational Rose no lo es y además este último es software propietario. Es por ello que se eligió para el modelado del proyecto Visual Paradigm.





2.5 Framework JavaScript.

Actualmente existen diversas herramientas que mediante el uso de frameworks y librerías del lenguaje JavaScript permiten realizar el diseño de interfaces de forma fácil rápida de de muy buena calidad visual. Tal es el caso de Dojo, Prototype, JQuery, y EXT JS.

Dojo

Es un framework muy potente y muy completo, que otorga al usuario desarrollador las armas necesarias para crear una aplicación muy vistosa y eficaz en poco tiempo. Incluye consigo la librería Dijit, que contiene componentes gráficos tales como campos de sugerencias (llamados *combobox* o *filteringselects*). Sin embargo, al generar una página web expuesta al público, Dojo puede ser demasiado pesado, y es por eso que se hace necesario revisar otros frameworks. En este caso Prototype, JQuery y EXT JS.

Prototype

Prototype es una librería JavaScript diseñada para extender las propiedades básicas del lenguaje y regularizar las diferencias de implementación entre navegadores. Prototype aprovecha la maleabilidad de JavaScript para enriquecer la funcionalidad de sus objetos y métodos base (Objetos, Arreglos, Numéricos). Incluye nuevas clases utilitarias para trabajar con Ajax y un sencillo sistema de herencia para acercar a JavaScript a las nociones OOP⁴ familiares en Java, u otros lenguajes orientados a objetos. Prototype no es en sí una aplicación, sino una base para escribir aplicaciones más sólidas. De hecho, es usado para desarrollar librerías especializadas como Scriptaculous (para Ajax avanzado). (19)

JQuery

JQuery es un framework JavaScript que ofrece una infraestructura con la que se tiene mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos y aplicaciones que hacen uso de

⁴ **OOP** (*Object Oriented Programming*) es un paradigma de programación que utiliza objetos y sus interacciones para diseñar softwares.





Ajax. Simplemente se debe tener conocimiento de las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la obtención de los objetivos. (20)

EXT JS

EXT JS es un framework usado para crear aplicaciones web con interfaces muy similares a la de una aplicación de escritorio, algo muy cercano a lo que se pretende lograr en la plataforma. Se utiliza para el desarrollo de aplicaciones enriquecidas para la web haciendo un uso intensivo de las tecnologías AJAX, XHTML/ DHTML y DOM. Originalmente fue creado como una extensión de Yahoo User Interface (YUI) otro framework similar, Ext incluye interoperabilidad con JQuery, Prototype y Scriptaculous.

Sus principales características son:

- Alto rendimiento en ejecución debido a la optimización de código Java Script.
- Controles de usuario personalizables.
- Modelo orientado a componentes, bien diseñado y extensible.
- Posee una API intuitivo y fácil de utilizar.
- Distribuido bajo licencias Open Source y comerciales.

Este permite crear aplicaciones complejas utilizando componentes predefinidos. También trae por defectos muchos de los componentes y funcionalidades que se necesitan usar en la plataforma, además es muy usado en otros proyectos de la universidad lo cual otorga la facilidad de contar con una fuente de experiencia en el desarrollo de la aplicación. (21)

Este framework permite la integración además de otras librerías JavaScript y se acopla muy bien con el framework PHP Symfony usado por el equipo de desarrollo de PRIMICIA en la implementación del sistema.

2.6 Conclusiones.

En este capítulo se trató todo lo referente a las herramientas y las tecnologías actuales usadas en el mundo para poder finalmente seleccionar la más acorde al producto justificando la elección. Teniendo en cuenta la importancia de una correcta selección de la metodología de desarrollo para el desempeño fluido del proceso de desarrollo cabe destacar que RUP cumple con lo requerido para la creación del módulo de





Capítulo 2



personalización pues abarca todas las fases de desarrollo de un software dentro de las que se encuentra el flujo de análisis y diseño. Además en el capítulo se describen herramientas de modelado y se elige Visual Paradigm pues posee facilidades para la confección de diagramas y otros artefactos del flujo de trabajo de análisis y diseño.





Capítulo 3

Presentación de la Solución Propuesta

3.1 Introducción.

Para un nivel de comprensión profundo de todo lo referente al funcionamiento del sistema y de todo el proceso que se lleva a cabo en la institución se debe tener un completo dominio de los conceptos que se manejan en el mismo. Este capítulo se refiere al funcionamiento del sistema y las acciones de los actores en el mismo. Además se generan un conjunto de artefactos que permiten una fluida comunicación entre los desarrolladores del producto.

3.1.1 Modelo de Dominio.

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelo conceptual, modelo de objetos del dominio y modelos de objetos de análisis. Por tanto, el modelo del dominio podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio. (22)

El proyecto en el cual se enmarca la realización del módulo de personalización está sujeto a una gran cantidad de cambios, tener un modelo de dominio reducirá el costo total de esos cambios. Encapsular todo el comportamiento de la empresa que es probable que cambie en una sola parte del software se reduce la cantidad de tiempo que se necesita realizar un cambio porque todo se realizará en un lugar. Al aislar ese código tanto como sea posible, reducimos la posibilidad de cambios en otros lugares que hacen que se interrumpa, lo que reduce el tiempo que tarda estabilizar el sistema.



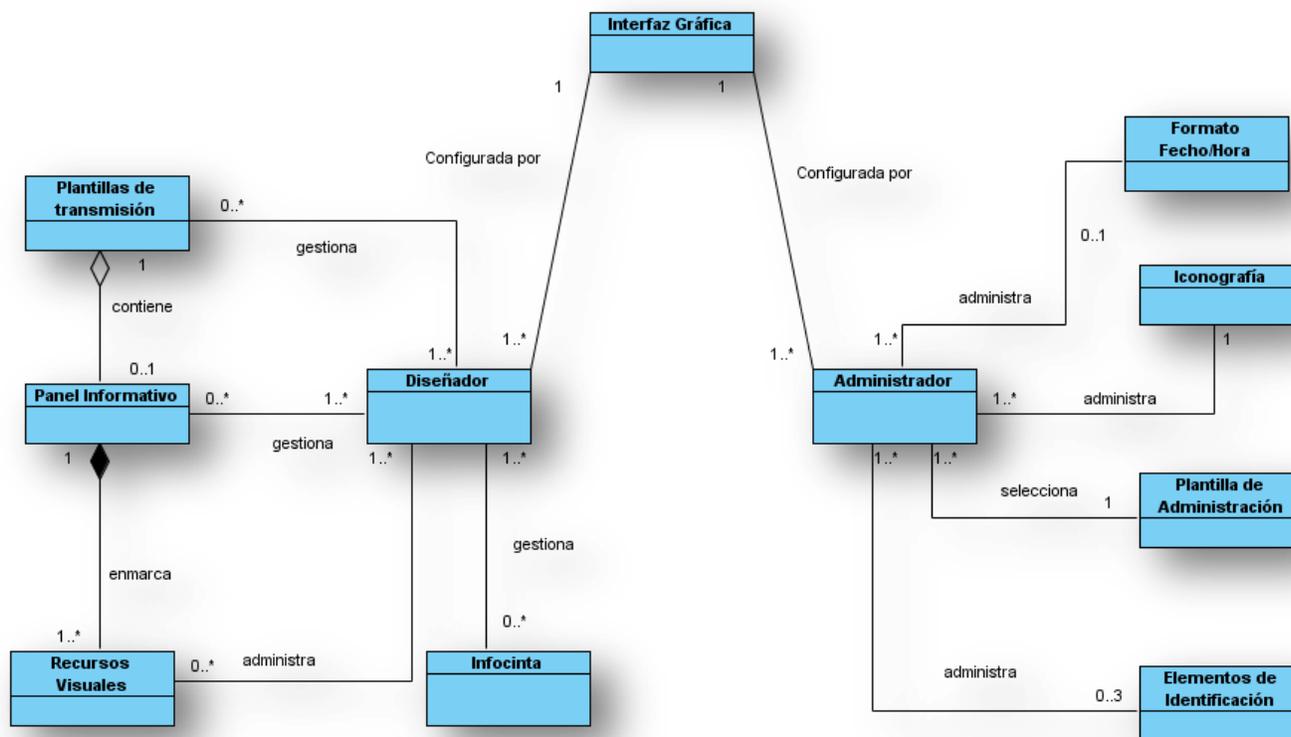


Figura 7 Diagrama de clases Modelo de Dominio.

3.1.2 Glosario de Términos del Dominio.

Elementos configurables.

Conjunto de elementos a los que se les puede personalizar el diseño de su interfaz gráfica.

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.

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 lo pueden ser el clima y la hora.

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.

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.

Plantillas de administración.

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

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 o texto-imagen, promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes.

3.2 Requerimientos Funcionales.

Los requisitos funcionales definen las acciones que debe realizar el sistema. Son capacidades o condiciones que el sistema debe cumplir, éstos se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

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 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.

Serán propuestos un conjunto de iconos a escoger por el cliente de los cuales solo se habilitaran los seleccionados para su posterior uso. 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.

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.





3.3 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, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

3.3.1 Software.

- Se debe utilizar como servidor web el “Apache”.
- Sistema operativo para los servidores “Linux”.

3.3.2 Hardware.

En la tabla que se muestra a continuación se encuentran los requerimientos mínimos (filas azul oscuro) y óptimos (filas azul claras) recomendados que debe cumplir el equipamiento tecnológico de la plataforma:

Variante de un (1) servidor:

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Administración Transmisión	Pentium IV 2.8 GHz	1 Gb	120 Gb	Ethernet 10/100 Mbs	Capturadora Hauppauge PVR 350 WinTV
	Dual-Core Xeon 1.60 GHz	4 Gb	500 Gb	Ethernet 10/100 Mbs	Exportadora Radeon X300 ATI

Tabla 1 Requerimiento de hardware.





Variante de dos (2) servidores:

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Administración	Pentium IV 2.8 GHz	1 Gb	120 Gb	2 puertos Ethernet 10/100 Mbs	Capturadora Hauppauge WinTV PVR 350
	Dual-Core Xeon 2.33 GHz	4 Gb	500 Gb	2 puertos Ethernet Gigabit	
Transmisión	Pentium IV 2.8 GHz	512 Mb	40 Gb	Ethernet 10/100 Mbs	Exportadora ATI Radeon X300
	Dual- Core Xeon 1.60 Ghz	1 Gb	80 Gb	Ethernet Gigabit	

Tabla 2 Requerimiento de hardware.

3.3.3 Apariencia o interfaz externa.

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

3.3.4 Confidencialidad.

Acceder a la información dependiendo de los permisos de cada rol del usuario.





3.3.5 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.

3.3.6 Usabilidad.

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

3.3.7 Soporte.

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

3.3.8 Restricciones en el diseño y la implementación.

- El sistema estará implementado en lenguaje PHP, utilizando como IDE de desarrollo Eclipse. 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.

3.4 Descripción del Sistema Propuesto.

3.4.1 Descripción de los actores.

Un actor no es más que un agente externo que no forma parte del sistema, sino que interactúa con éste. El mismo puede intercambiar información o ser un recipiente pasivo de información y representando a un





ser humano, a un software o a una máquina que intercambia acciones con el sistema. En este caso interactúan 2 actores los cuales se definen a continuación:

Actores	Descripción
Diseñador	Encargado de diseñar todos los elementos que se visualizarán en el subsistema de transmisión.
Administrador	Usuario registrado con capacidad total de manipular configuraciones, registros y permisos.

Tabla 3 Actores del sistema.

3.4.2 Diagrama de Casos de Uso del Sistema.

Un caso de uso es un documento narrativo que describe la secuencia de un actor agente externo que utiliza un sistema para completar un proceso. Seguidamente se presentan las interacciones entre los actores y los casos de uso del sistema.



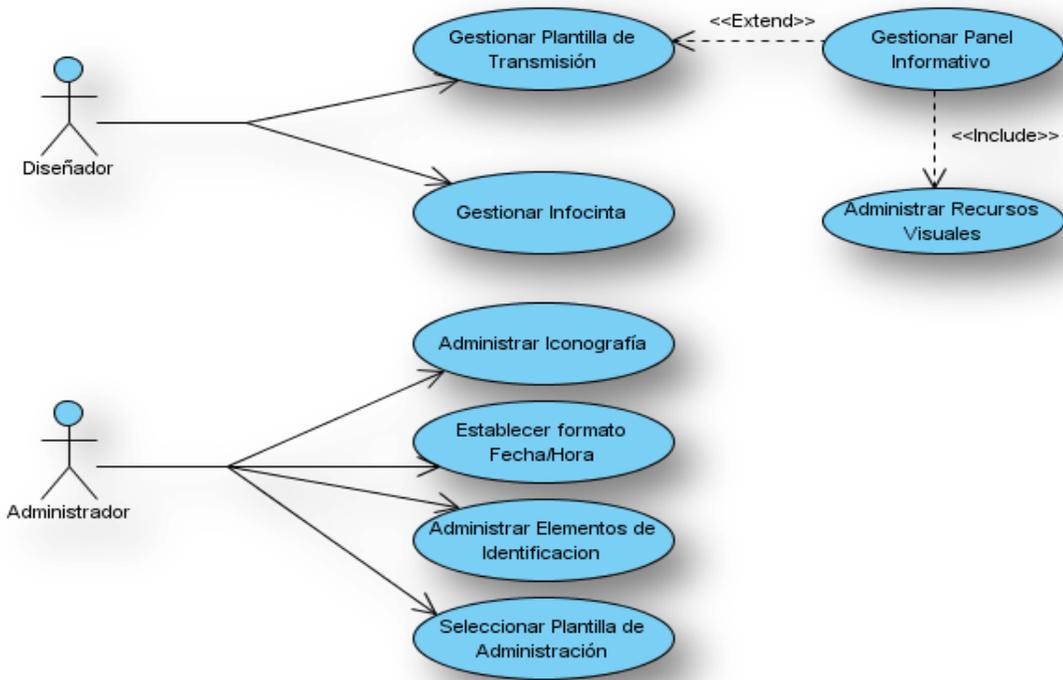


Figura 8 Diagrama de Casos de Uso del Sistema.

3.4.3 Descripción de los Casos de uso del Sistema

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos funcionales del sistema:

Caso de Uso:	Gestionar infocinta
Actores:	Diseñador.
Resumen:	El caso de uso comienza cuando el diseñador selecciona la opción gestionar infocinta para crear, modificar o eliminar infocintas que no son más que cintillos informativos que brindan algún tipo de información.





Referencias	R1, R1.1, R1.2, R1.3, R1.4
--------------------	----------------------------

Tabla 4 Descripción del CU Gestionar Infocintas.

Caso de Uso:	Gestionar plantillas de transmisión.
Actores:	Diseñador.
Resumen:	El caso de uso comienza cuando el diseñador selecciona la opción gestionar plantillas de transmisión para crear, modificar o eliminar dichas plantillas las cuales están encargadas de mostrar la noticia de diferentes formas de acuerdo a las necesidades del cliente.
Referencias	R2, R2.1, R2.2, R2.3, R2.4

Tabla 5 Descripción del CU Gestionar Plantillas de Transmisión.

Caso de Uso:	Administrar Iconografía
Actores:	Administrador
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción administrar iconografía el sistema muestra un conjunto de opciones para cambiar la iconografía del sistema.
Referencias	R3

Tabla 6 Descripción del CU Administrar Iconografía.





Caso de Uso:	Administrar recursos visuales
Actores:	Diseñador.
Resumen:	El caso de uso comienza cuando el usuario selecciona la opción administrar recursos visuales donde se dan las opciones de seleccionar y ordenar dichos recursos que pueden ser el elemento clima, tiempo restante y título de la noticia.
Referencias	R4, R4.1, R4.2

Tabla 7 Descripción del CU Administrar Recursos Visuales.

Caso de Uso:	Establecer formato de Fecha\Hora.
Actores:	Administrador.
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción establecer formato de Fecha\Hora, el sistema muestra un conjunto de opciones para cambiar el formato de fecha y hora para la parte administrativa del sistema.
Precondiciones:	El usuario debe estar autenticado en el sistema, poseer privilegios administrativos.
Referencias	R5

Tabla 8 Descripción del CU Establecer formato Fecha/Hora

Caso de Uso:	Gestionar panel informativo.
---------------------	------------------------------





Actores:	Diseñador.
Resumen:	El caso de uso comienza cuando el diseñador selecciona la opción gestionar panel informativo para crear, modificar o eliminar dichos paneles los cuales enmarcan los recursos visuales.
Referencias	R6, R6.1, R6.2, R6.3

Tabla 9 Descripción del CU Gestionar Panel Informativo.

Caso de Uso:	Administrar elementos de identificación.
Actores:	Administrador.
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción administrar elementos de identificación, el sistema muestra un conjunto de opciones para cambiar los elementos de identificación del canal que pueden ser el logotipo de la empresa, el video de presentación y el video de fin de transmisión.
Referencias	R8.

Tabla 10 Descripción del CU Administrar Elementos de Identificación.

Caso de Uso:	Seleccionar plantilla de administración.
Actores:	Administrador.
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción seleccionar





	plantilla de administración el sistema muestra un conjunto de plantillas para seleccionar la deseada lo que modificara el aspecto de la aplicación de la parte administrativa.
Referencias	R2.

Tabla 11 Descripción del CU Seleccionar Plantilla de Administración.

3.5 Conclusiones.

En este capítulo se le dio inicio al desarrollo de la solución propuesta obteniéndose un listado con las funciones que debe tener la aplicación, representadas mediante un diagrama de casos de uso del sistema a partir del análisis de los conceptos que intervienen en el dominio del problema. De cada Caso de Uso se confeccionó una descripción textual representando las acciones de los actores de la aplicación y las respuestas que se debería dar. Luego de haber concluido todo esto se puede comenzar a construir la aplicación, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias en este capítulo.





Capítulo 4

Construcción de la Solución Propuesta

4.1 Introducción.

En este capítulo se realizará el análisis y diseño de la solución propuesta. A través de los flujos de trabajo de Análisis y Diseño, se muestran los diagramas de clases del análisis y del diseño de los diferentes casos de usos definidos en el capítulo anterior, y lo referente al modelo de datos.

4.2 Modelo de análisis.

En la construcción del modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones que existen entre ellas. Con esta información se construye el diagrama de clases del análisis.

4.2.1 Diagramas de Clases del análisis.

En el diagrama de clases del análisis se identifican las clases que utiliza la aplicación y sus relaciones, así como los atributos y sus funcionalidades, las clases normalmente se convierten en las tablas de la base de datos. A continuación se muestran los diagramas de clases del análisis (DCA) de todos los casos de uso del sistema.

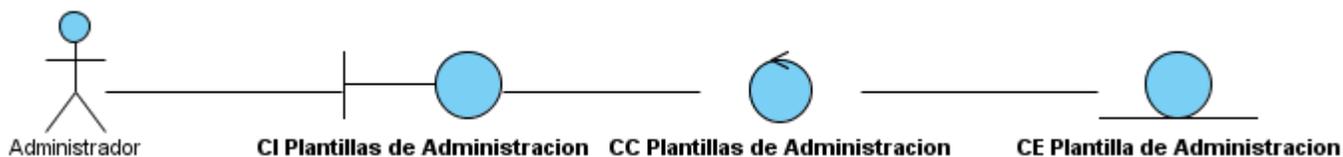


Figura 9 DCA Seleccionar Plantilla de administración.



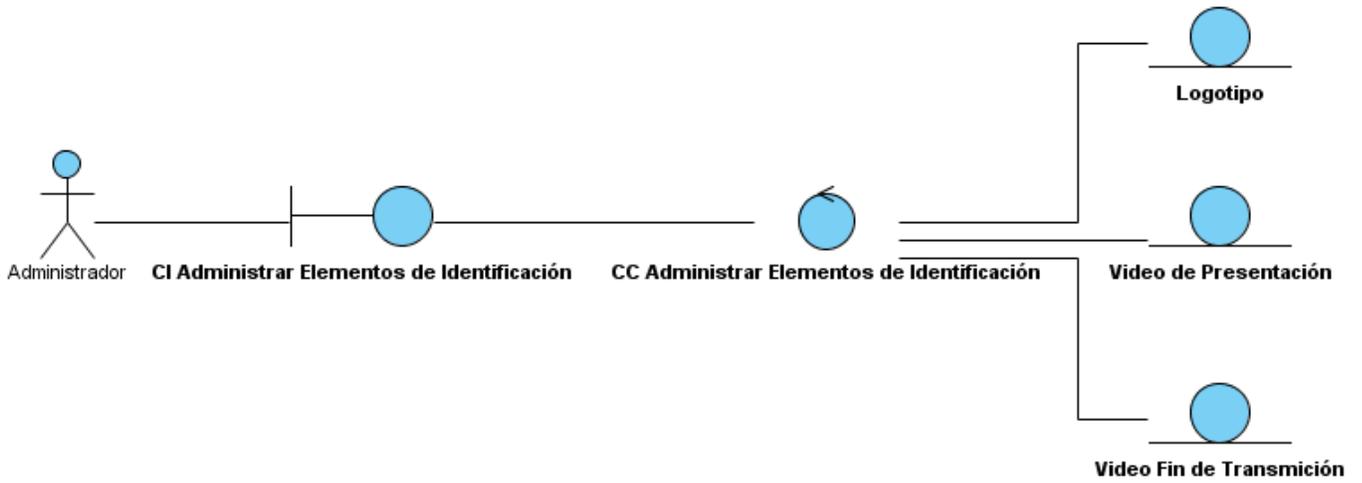


Figura 10 DCA Administrar Elementos de Identificación.



Figura 11 DCA Administrar Iconografía.



Figura 12 Administrar Recursos Visuales.

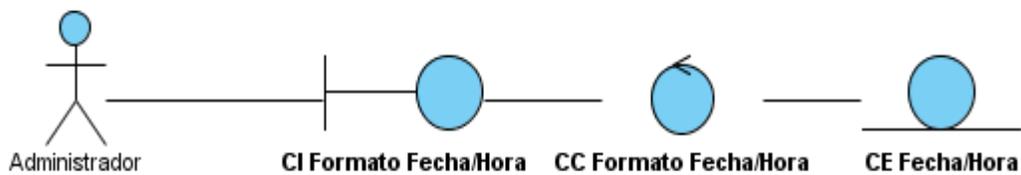


Figura 13 DCA Establecer Formato Fecha/Hora.



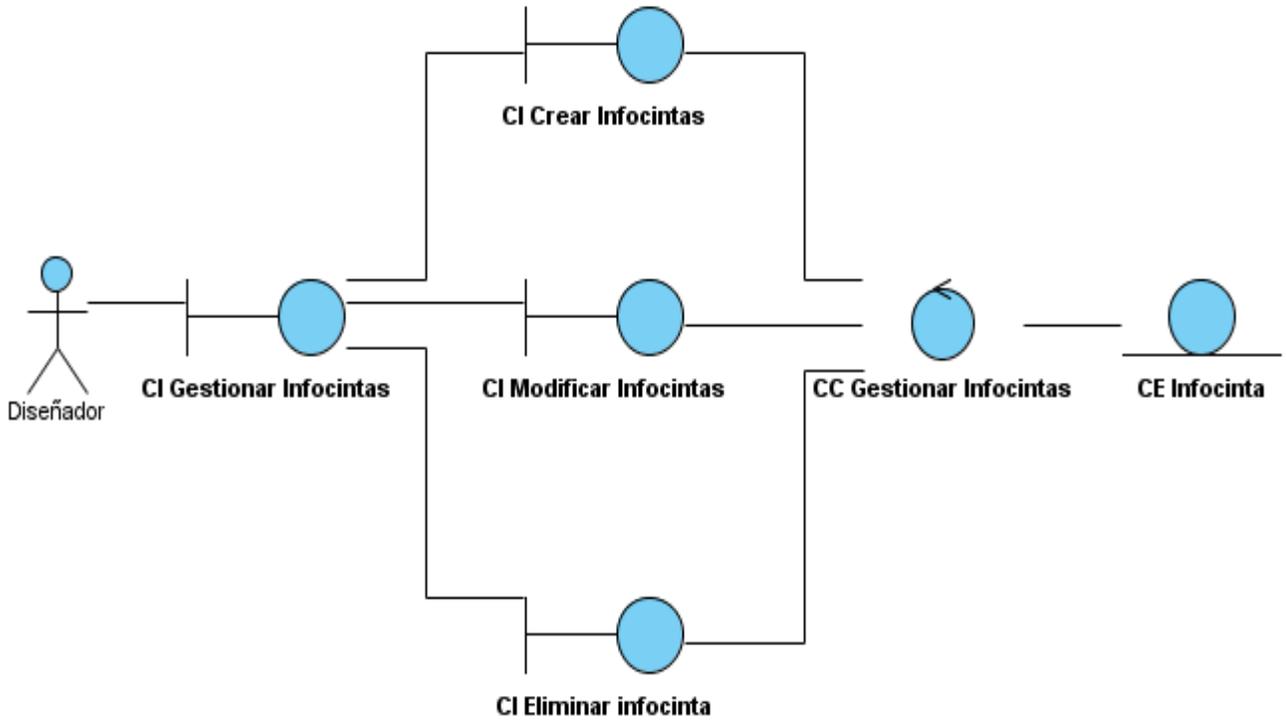


Figura 14 DCA Gestionar Infocintas.

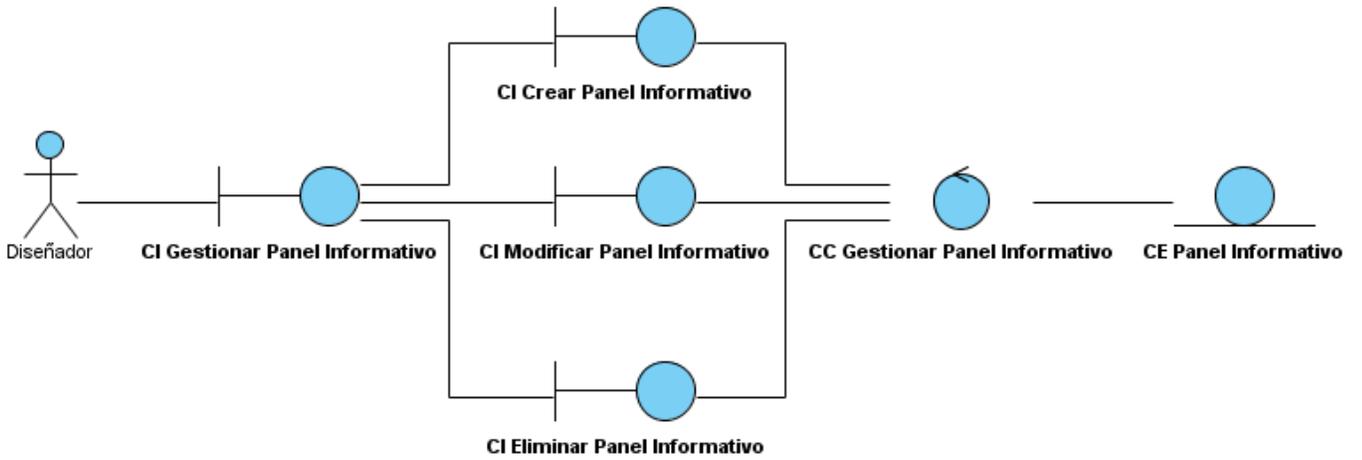


Figura 15 DCA Gestionar Panel Informativo.



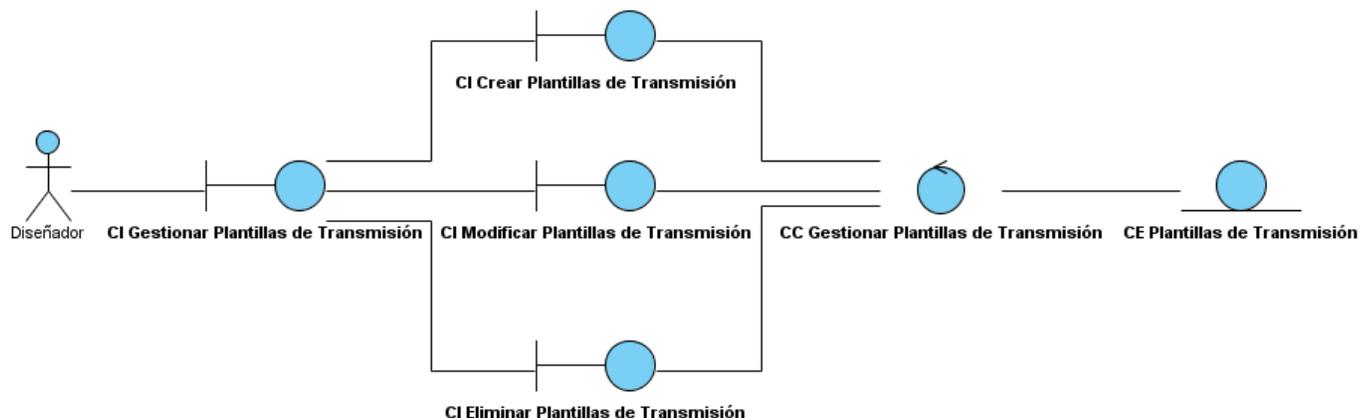


Figura 16 DCA Gestionar Plantillas de Transmisión.

4.3 Modelo de diseño.

Para el modelado del diseño se siguieron varios patrones de diseño, los cuales se mencionan a continuación:

4.3.1 Patrones de diseño.

Los patrones de diseño no son más que soluciones simples y elegantes a problemas específicos y comunes del diseño. Estos aportan soluciones basadas en la experiencia y cuyo funcionamiento ha sido demostrado. Los patrones de diseño se dividen en dos grandes grupos: los básicos donde se encuentran los patrones GRASP (*General Responsibility Assignment Software Patterns*) y los avanzados que enmarcan los GOF (*Gang of Four*) estos últimos a su vez se dividen en creacionales, estructurales y de comportamiento. Seguidamente se darán a conocer los patrones de diseño usados y una breve explicación de cómo el framework seleccionado los implementa.

Patrones GRASP:

✓ **Creador:**

Se encarga de darle a la clase A la responsabilidad de crear objetos de la clase B. En este caso A es creador de los objetos B. En Symfony la clase Actions contiene las acciones definidas y se ejecutan cada





una de ellas. En estas se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

✓ **Controlador:**

Propone asignarle la responsabilidad de manejar todos los eventos del sistema a una clase. Todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

✓ **Experto:**

Se encarga de asignar una responsabilidad al experto en información, o sea, la clase con la información suficiente para cubrir la necesidad. Este es uno de los más utilizados en el framework, puesto que Doctrine es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

✓ **Bajo Acoplamiento:**

Garantiza que una clase no dependa de muchas otras ya que se hace difícil de reutilizar. La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

✓ **Alta Cohesión:**

Se encarga de asignar responsabilidades a las clases de modo que las funcionalidades estén estrechamente relacionadas y que cada elemento del diseño tenga una labor única. Una clase con baja cohesión se hace difícil de entender, de conservar y de reutilizar. Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.





Patrones GOF:

Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a `sfContext::getInstance()`. En una acción, el método `getContext()`, un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Modelo Vista Controlador (MVC): Este patrón plantea la separación del problema en tres capas: la capa modelo, que representa la realidad; la capa controlador, que conoce los métodos y atributos del modelo, recibe y realiza lo que el usuario quiere hacer; y la capa vista, que muestra un aspecto del modelo y es utilizada por la capa anterior para interactuar con el usuario.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. El patrón usado para la realización del trabajo es una especificación del MVC implementada por el framework Symfony. Este toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

Estructurales:





Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño mencionado.

4.3.2 Diagramas de Clase del Diseño.

Los diagramas de clases del diseño son de utilidad a la hora de la toma de decisiones, en el diseño y la implementación. Los diagramas brindan una vista a sencilla de la solución final y resulta mucho más fácil realizar modificaciones a los mismos, una vez detectados los problemas, que tener que efectuar cambios a la solución ya implementada. En este trabajo se hace necesario dividir el diseño en tres paquetes lógicos que representan la filosofía del framework dejando claramente evidenciada la funcionalidad de cada uno de los elementos que se encuentran agrupados en dichos paquetes. A continuación se hace mención a los componentes principales de cada paquete:

La capa del Modelo

- Abstracción de la base de datos
- Acceso a los datos

Las clases de la capa del modelo se generan automáticamente, en función de la estructura de datos de la aplicación. Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración.

La capa de la Vista

- Vista
- Plantilla
- Layout





La vista se encarga de producir las páginas que se muestran como resultado de las acciones. La vista en Symfony está compuesta por diversos componentes, los success, las páginas clientes y los formularios, cada success responde a una acción siendo esto una facilidad pues de esta forma estas partes pueden ser modificables por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

La capa del Controlador

- Controlador frontal
- Acción

Este paquete contiene dos elementos fundamentales, el controlador frontal, que es el único punto de entrada a la aplicación, además maneja todas las peticiones Web, y las acciones, que contienen la lógica de las páginas, por cada acción es una clase que se crea y todas a su vez heredan de sfAction la cual contiene métodos comunes para las demás clases facilitando la usabilidad y en caso de cambio no se afectaría nada más que aquellas clases implicadas en él. Seguidamente se muestran los diagramas de clase del diseño (DCD) de cada caso de uso.

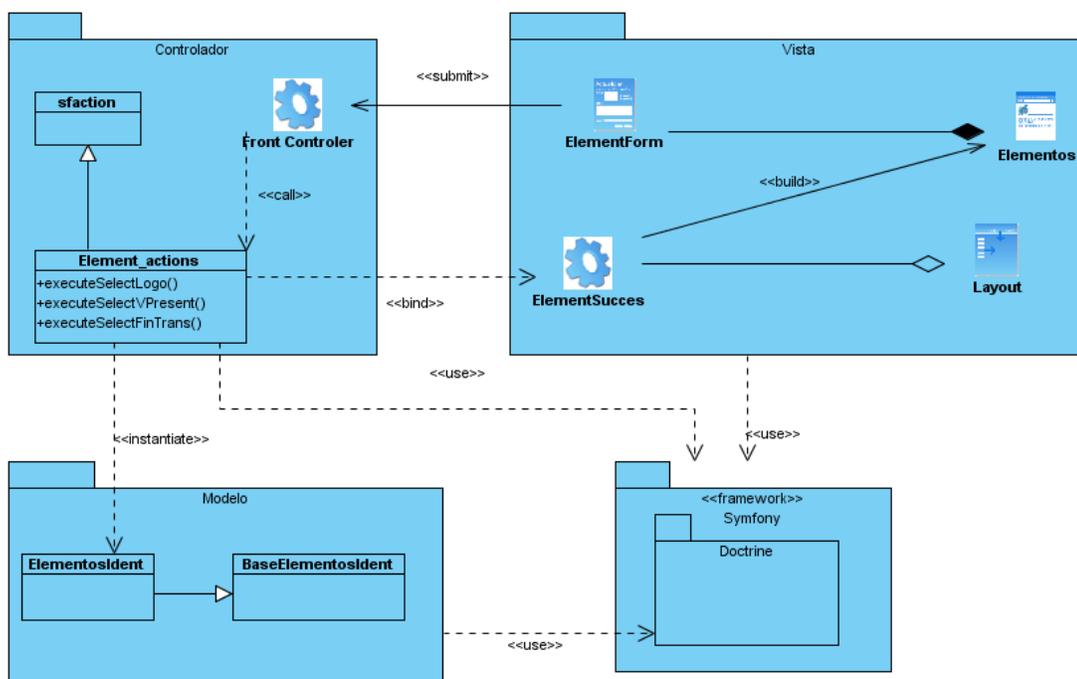


Figura 17 DCD Administrar Elementos de Identificación.

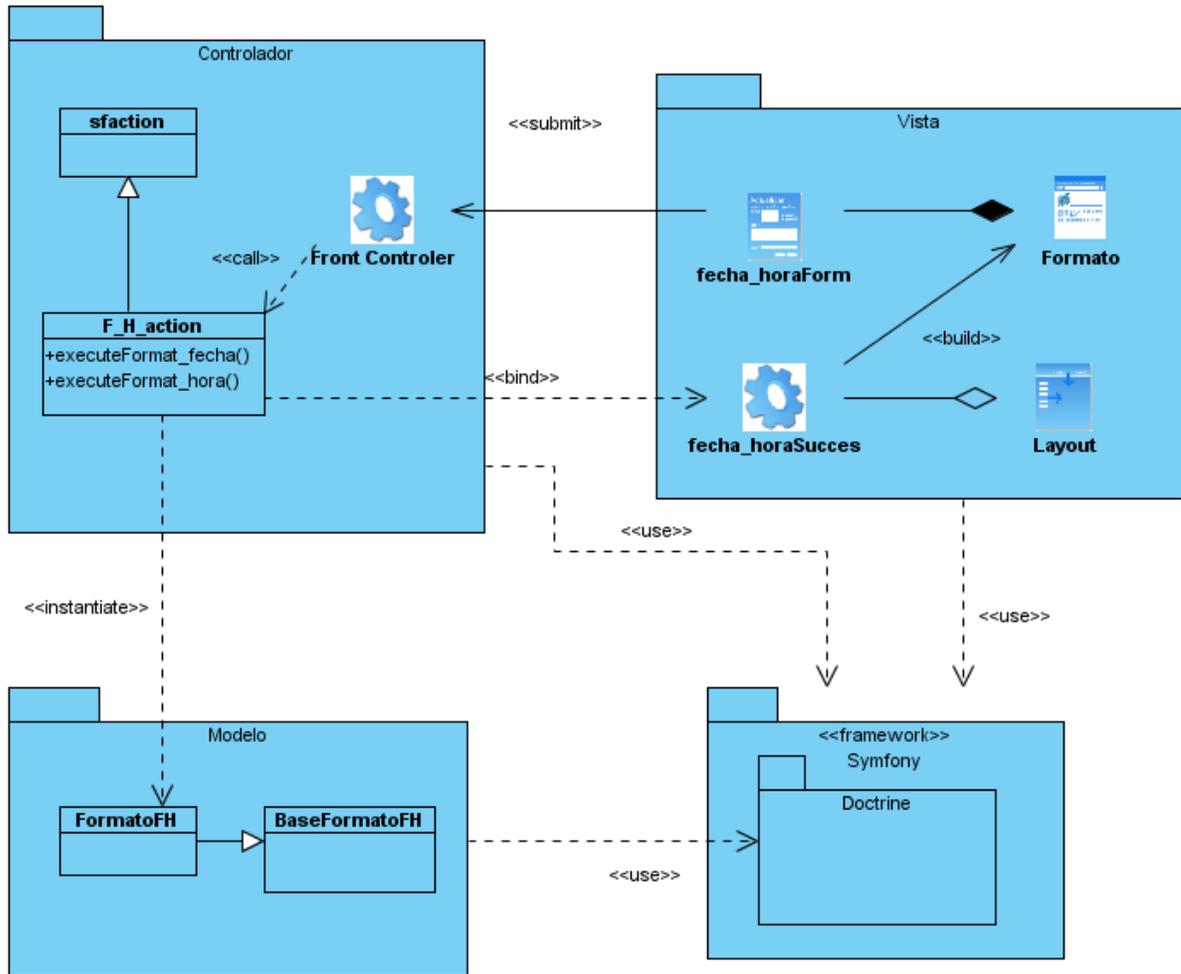


Figura 18 DCD Establecer Formato Fecha/Hora.

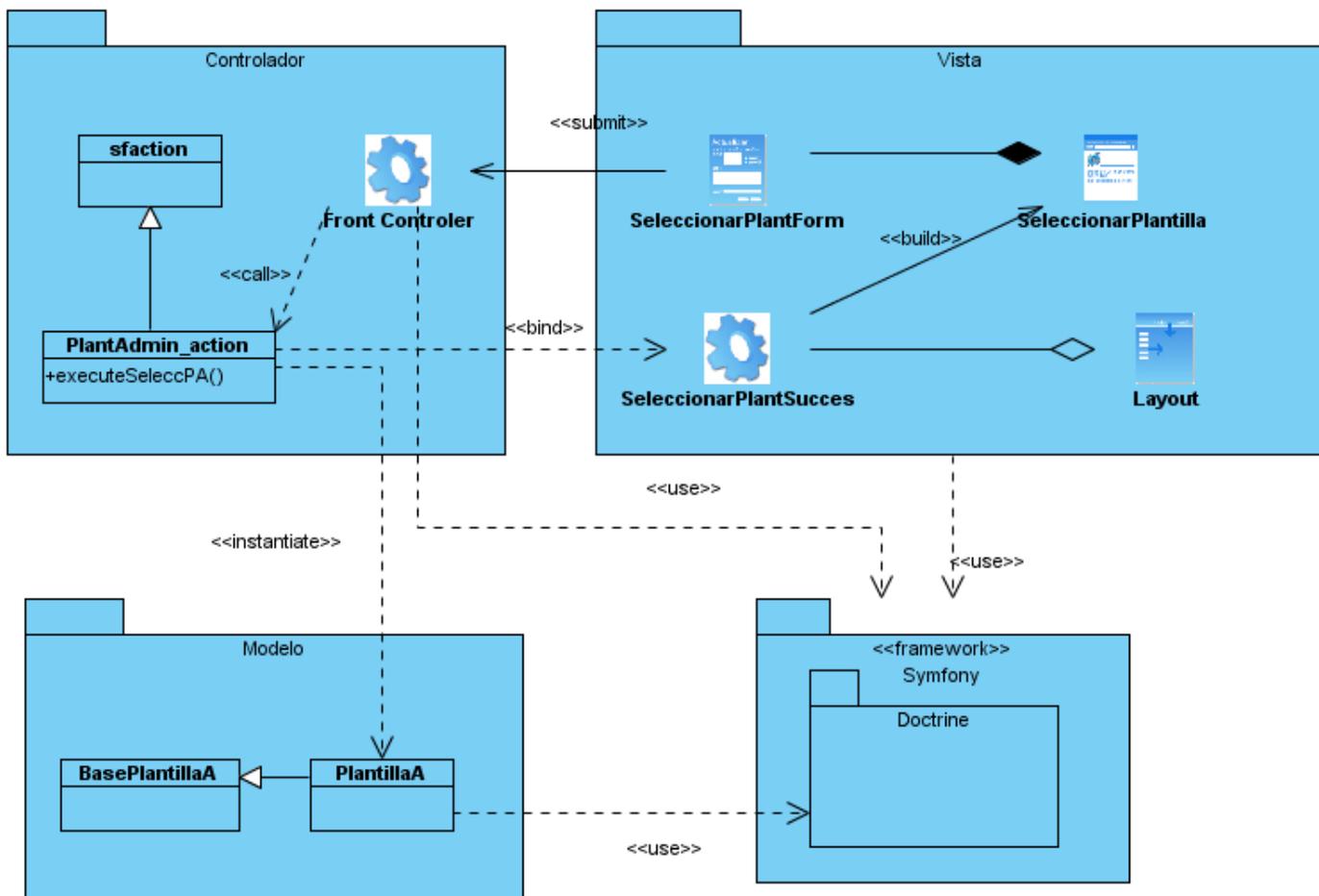


Figura 20 DCD Seleccionar Plantilla de Administración.

Los siguientes DCD presentan una estructura distinta ya que son el resultado de la integración de Ext JS y el framework Symfony. Cabe destacar que las peticiones a las acciones se realizan mediante Ajax Request y las respuestas a dichas peticiones son a través de prototipos de objetos JSON y las mismas son enviadas por las clases JavaScript.

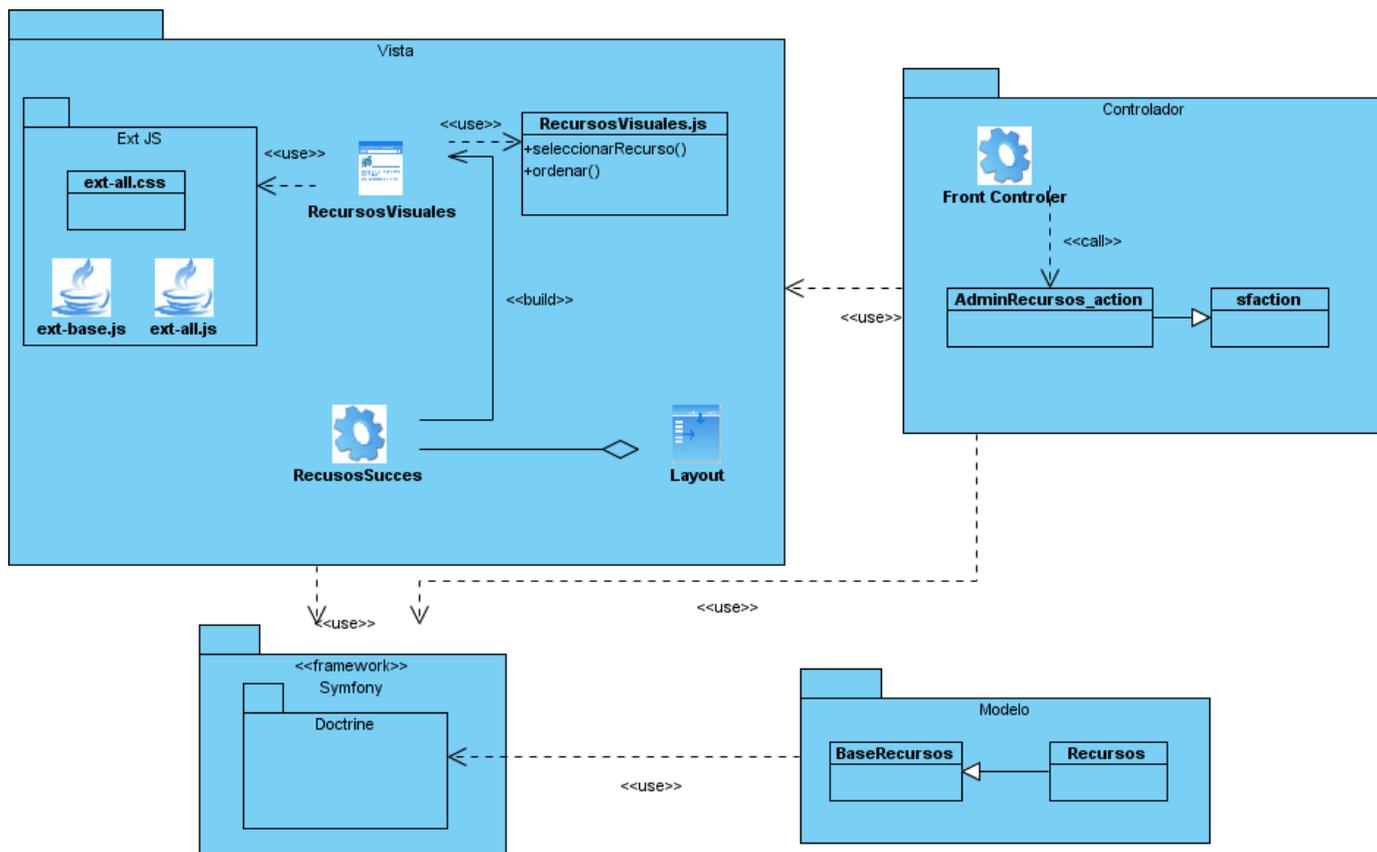


Figura 21 DCD Administrar Recursos Visuales.

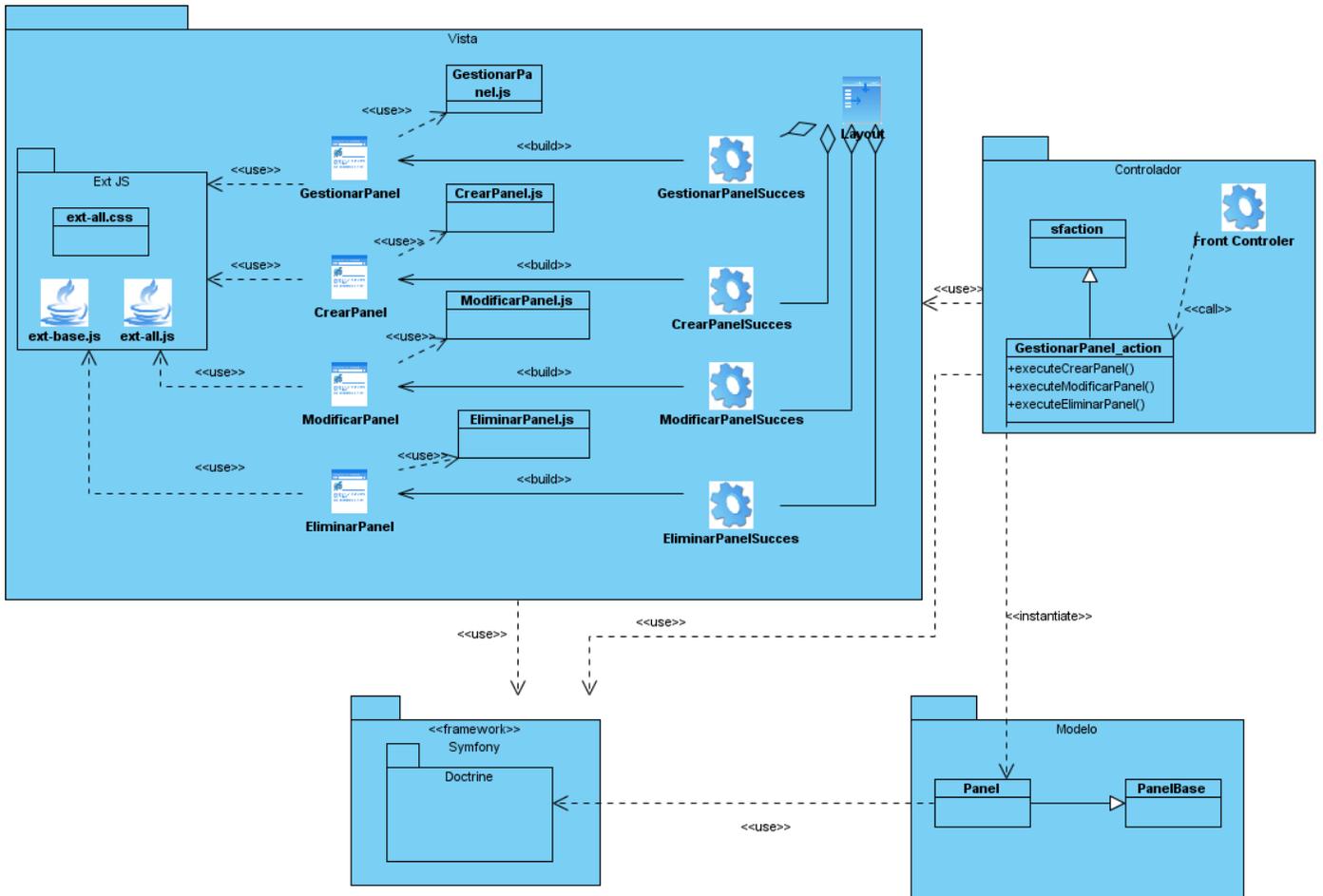


Figura 23 DCD Gestionar Panel Informativo.

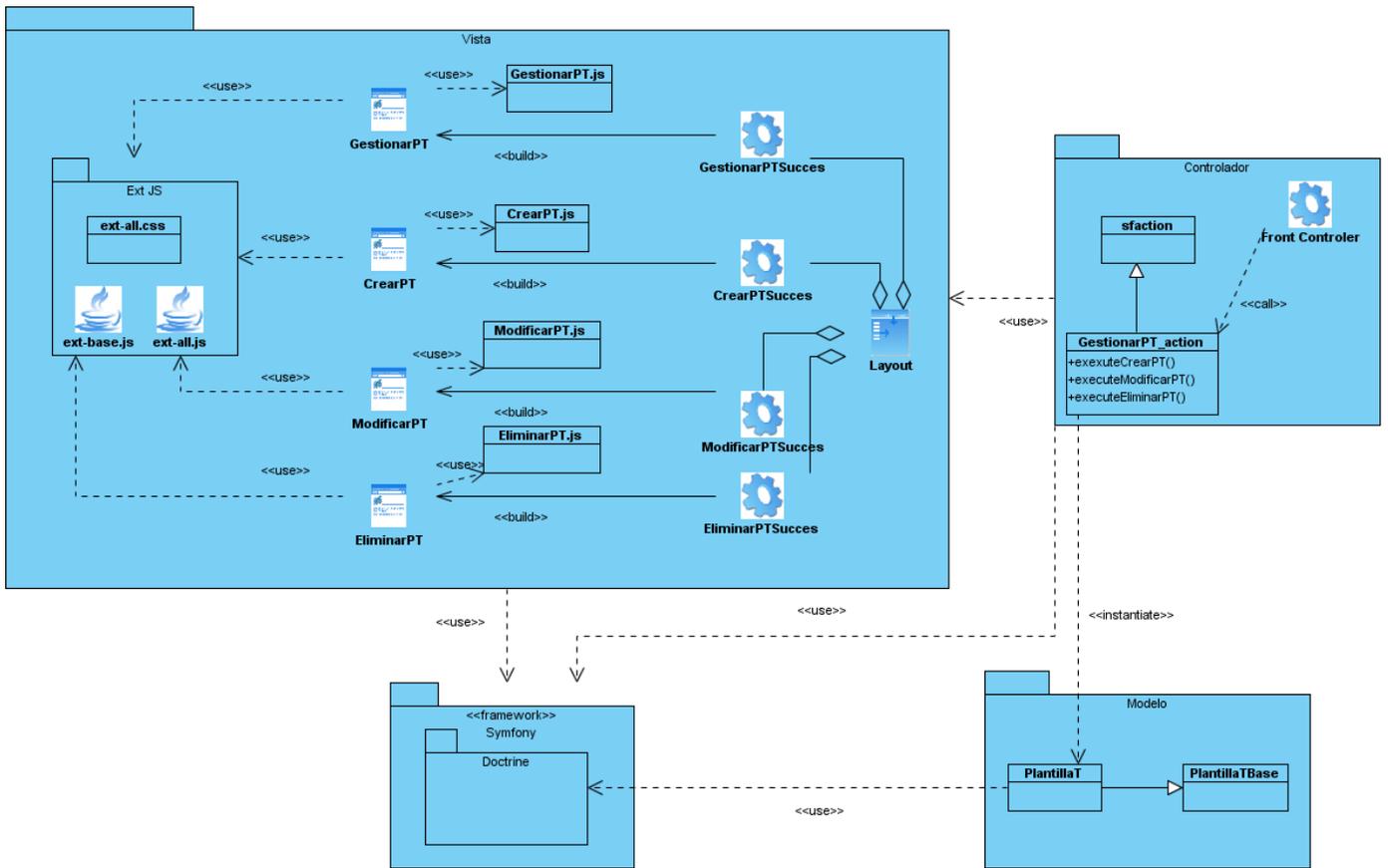


Figura 24 DCD Gestionar Plantillas de Transmisión.



4.3.3 Diagrama de clases persistentes.

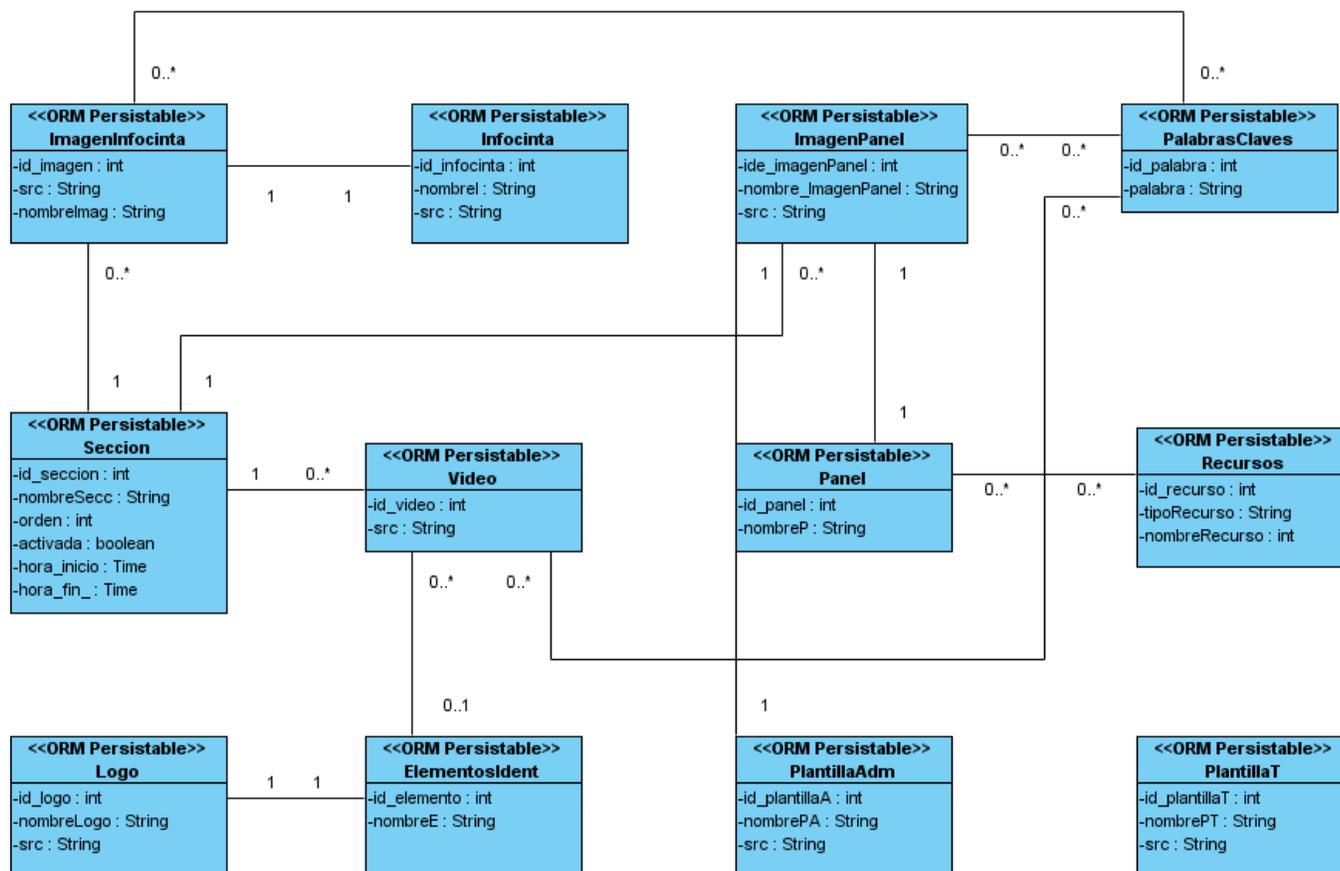


Figura 25 Diagrama de Clases Persistentes.

4.4 Base de Datos.

La base de datos de la plataforma está concebida sin tener en cuenta como es lógico los cambios que provoca la agregación del módulo de personalización. Es por ello que se hace necesaria la creación de nuevas tablas mostradas en amarillo en el modelo de datos mientras que las tablas que permanecen sin cambios se muestran de color naranja.





4.4.1 Modelo de Datos.

El modelo de datos describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación.

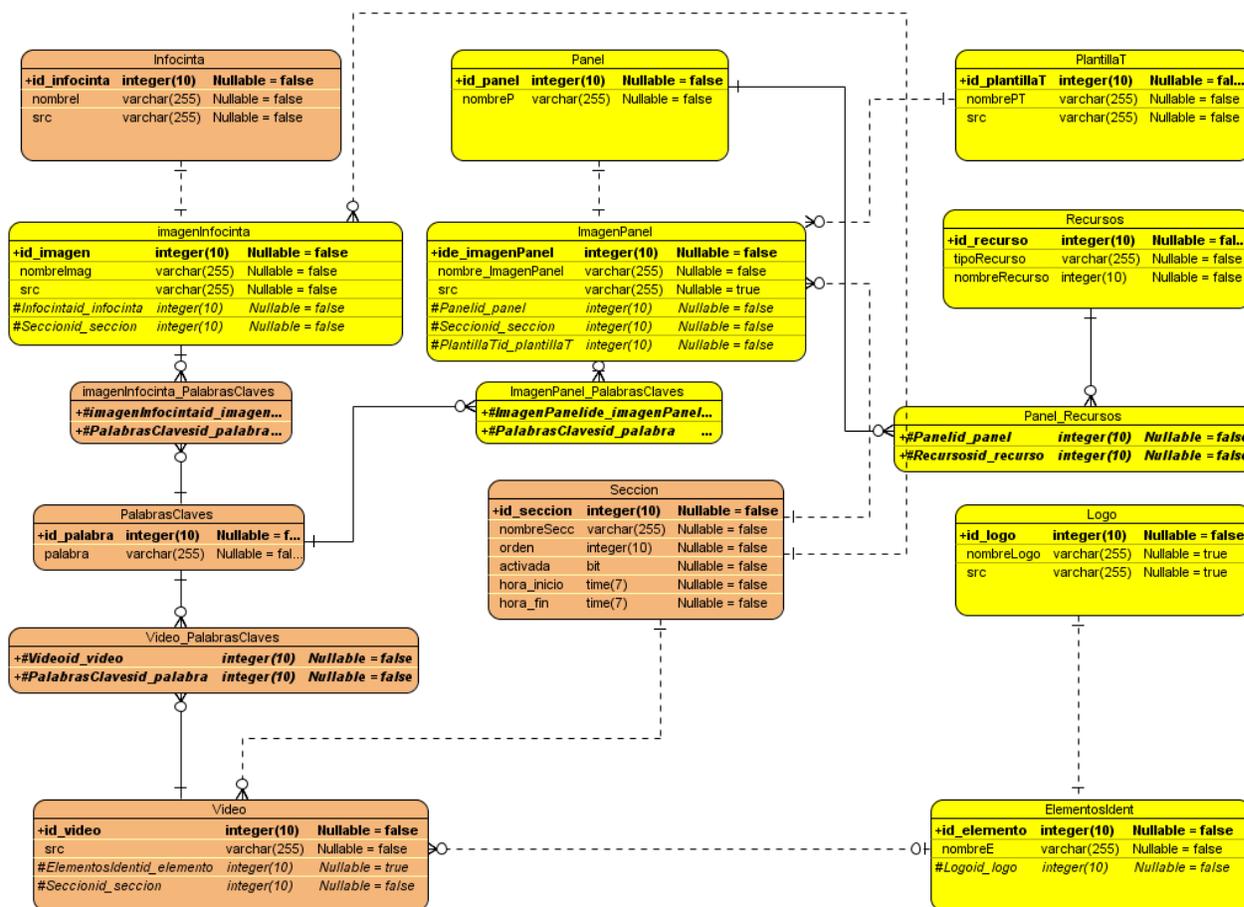


Figura 26 Modelo de Datos.

Conclusiones.

En el capítulo se finaliza el diseño de las funcionalidades detectadas. Se generaron los artefactos correspondientes al flujo de trabajo análisis y diseño, específicamente los diagramas de clase del análisis y del diseño así como los diagramas de colaboración, teniendo en cuenta los patrones de diseño utilizados por el framework a utilizar en la implementación de dichas funcionalidades. Se concluye que es necesario





Capítulo 4



la inclusión de nuevas tablas en la actual base de datos de la plataforma debido al surgimiento de nuevas entidades necesarias para lograr los objetivos del módulo de personalización de interfaz gráfica.





Conclusiones Generales.

Dando por terminado el diseño del módulo de personalización de la interfaz gráfica de la plataforma de Televisión Informativa PRIMICIA se concluye:

- ✓ La personalización es un elemento de gran peso en el desarrollo de sitios web por lo que se considera que la decisión de permitir que los clientes puedan adaptar la interfaz de la plataforma primicia a sus necesidades traerá cuantiosos beneficios a la hora de comercializar el producto.
- ✓ El análisis y diseño de las funcionalidades para la personalización dinámica de la interfaz gráfica de la plataforma aporta a los desarrolladores la documentación necesaria para la implementación de las mejoras pertinentes en el producto.
- ✓ El trabajo provocó cambios en la estructura actual del producto tales como: el surgimiento de nuevas tablas en la base de datos y las bases ingenieriles para la implementación de nuevas funcionalidades para crear una interfaz gráfica personalizable.
- ✓ El problema al que se le da solución exigió un estudio profundo de los elementos configurables presentes en los subsistemas y los efectos positivos que traería para el cliente una interfaz configurable en la aplicación capaz de facilitarle a los mismos la interacción con la misma.

Los objetivos trazados para el presente trabajo de diploma han sido cumplidos de forma satisfactoria incluyéndose una serie de recomendaciones para el desarrollo de futuros cambios en el sistema.





Recomendaciones.

A partir de los resultados o beneficios que proporciona el estudio realizado, y con vista a trabajos futuros se recomienda:

- ✓ Realizar los restantes flujos de trabajo que propone la metodología utilizada, RUP, llegando a implementar las funcionalidades que se han propuesto para el desarrollo de un módulo que permita la personalización dinámica de la interfaz gráfica en la plataforma PRIMICIA.
- ✓ Publicar este documento para que sirva de material de estudio a investigaciones futuras y como fuente de experiencia para desarrolladores de software con características similares al sistema en cuestión.





Bibliografía Citada

1. **Hernández García, Ruber y Montaner Hernandez, Yunior.** *Primicia: Plataforma de televisión Informativa.* Ciudad de la Habana : UCI, 2008.
2. **Bermúdez Valdés, Geovanys.** *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA.* Ciudad de la Habana : s.n., 2008.
3. **Abadal Falgueras, Ernest.** *El web el nuevo estandar de distribución.* Barcelona : Ediciones Universidad de Barcelona, 2001.
4. **Cachero Castro, Cristina.** *OO-H: UNA EXTENSION A LOS MÉTODOS OO PARA EL MODELADO Y GENERACIÓN AUTOMÁTICA DE INTERFACES HIPERMEDIALES.* Alicante : s.n., 2003.
5. *Técnicas y modelos de personalización de sitios web. iWorld, La revista de la tecnología y estrategia de negocio en Internet.* 2001.
6. Mastermagazine, Revista digital lider en informática. [En línea] [Citado el: 25 de Enero de 2010.] <http://www.mastermagazine.info/termino/5857.php>.
7. Techterms.com. [En línea] [Citado el: 28 de Noviembre de 2009.] <http://www.techterms.com/definition/multimedia>.
8. IP.TV. [En línea] http://www.ip.tv/iptv_site/esp/htm/plataforma.html.
9. Estructuretv. [En línea] <http://www.estructuretv.com>.
10. **García Duardo, Ivanni.** *PRIMICIA, Plataforma de Televisión Informativa. Rol Analista.* 2009.
11. **Sanchez, María A Mendoza.** *Metodologías De Desarrollo De Software.* Perú : s.n.
12. **Molpeceres, Alberto.** *Procesos de desarrollo: RUP, XP y FDD.*





13. **Informática, Consejo Superior de.** *Metodología de Planificación y Desarrollo de Sistemas de Información. Métrica Versión 2.1. Guía de Técnicas.* Madrid : s.n.
14. **Jacobson, Ivar, Rumbaugh, James y Booch, Grady.** *El Proceso Unificado de Desarrollo de Software.*
15. **Barrientos Enríquez, Aleida Mirian.** *El desarrollo de sistemas de información empleando el lenguaje de modelado unificado UML.*
16. Proceso Unificado de Rational para el desarrollo de software. [En línea] www.dybox.cl/metodologia/rup.html.
17. **HERNÁNDEZ ORALLO, E.** El Lenguaje Unificado de Modelado (UML), Soluciones racionales. [En línea] 2010. http://www.solucionesracionales.com/data_mod.html.
18. Visual-Paradigm. [En línea] [Citado el: 5 de febrero de 2010.] <http://www.visual-paradigm.com>.
19. Prototype. JavaScript Framework. [En línea] <http://prototypejs.org/>.
20. JQuery, Write less do more. [En línea] <http://docs.jquery.com>.
21. **García, Jesus D.** EXT JS in Action. s.l. : MEAP.
22. **Larman, Craig.** *UML y Patrones.* 2da Edición.





Bibliografía Consultada.

23. **Eguíluz Pérez, Javier.** *Introducción a CSS.* 2009.

24. —. *Introducción a JavaScript.* 2009.

25. **Alvarez, Rubén.** Desarrolloweb.com. *Introducción al HTML.* [En línea] [Citado el: 2 de febrero de 2010.] www.desarrolloweb.com/articulos/534.php.

26. —. Desarrolloweb.com. *Componentes de una interfaz web. Interfaces.* [En línea] [Citado el: 23 de enero de 2010.] www.desarrolloweb.com/articulos/2171.php.

27. **Alonso Guerrero, Zorilin y Leyva González, Genry.** *Sistema de Captura y Transcripción de Audio.* Ciudad Habana : s.n., 2009.

28. **Jack, Keith y Tsatsulin, Vladimir.** *Dictionary of Video and Television Technology.* Estados Unidos de América : Elsevier Science, 2002. ISBN 1-878707-99-X.

29. DiCom. [En línea] [Citado el: 22 de noviembre de 2010.] <http://maestriadicom.org/trabajos-dicom/la-personificacion-en-interfaces-maleables-analisis-de-la-propuesta-de-interaccion-del-cms-joomla-y-sus-implicancias-para-el-usuario/>.

30. **Eguíluz Pérez, Javier.** *Introducción a AJAX.*





Glosario de Términos

CSS.

Lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

JavaScript.

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Página web dinámica.

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

Symfony.

Symfony es un completo *framework* diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. (...). Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

