

Universidad de las Ciencias Informáticas

FACULTAD 6



Título: Desarrollo del módulo de gestión y publicación de señales para el subsistema de administración de la Plataforma de Televisión Informativa PRIMICIA.

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

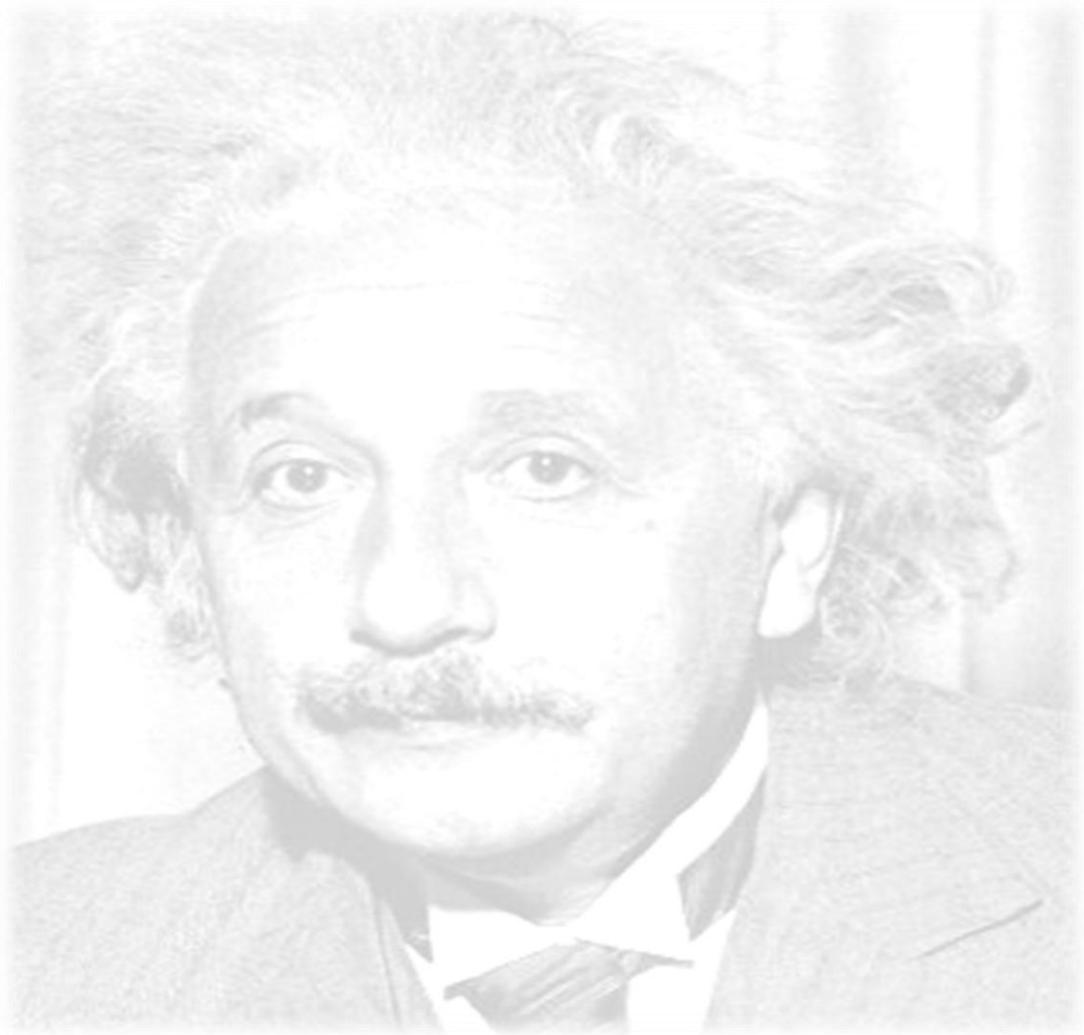
Autor: Yamila Tamayo Campaña

Tutor: Ing. Lisandra Delgado Cabrera

Co-tutor: Ing. Carlos De Jesús Andrés González

La Habana, Junio de 2012.

Año 54 de la Revolución.



“La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no”

Albert Einstein



Declaración de auditoría

Declaro que soy el único autor de este trabajo y autorizo al departamento de Señales Digitales del centro de Geoinformática y Señales Digitales (GEySED) 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 Junio del año 2012.

Yamila Tamayo Campaña

Ing. Lisandra Delgado Cabrera

Autor

Tutor



DATOS DE CONTACTO

Tutor: Ing. Lisandra Delgado Cabrera.

Categoría docente: Adiestrado.

Centro de trabajo: Universidad de las Ciencias Informáticas.

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

Año de graduación: 2010

Institución en la que se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: ldcabrera@uci.cu

Co-tutor: Ing. Carlos de Jesús Andrés González.

Categoría docente: Adiestrado.

Centro de trabajo: Universidad de las Ciencias Informáticas.

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

Año de graduación: 2010

Institución en la que se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: candres@uci.cu

Agradecimientos

A mis padres que me han dado todo el apoyo incondicional, el amor y la fuerza para poder llegar hasta aquí.

A mi hermana Yaimé y a su esposo Alexander por estar pendientes de mis problemas y brindarme su cariño.

A mi hermanita Mariolis por dar alegría a mi vida y por su cariño.

A mi compañero, amigo y novio Alejandro por ser mi sostén en los momentos difíciles, por tener siempre su apoyo y consideración, por ser la persona que es, te amo mucho.

A toda mi familia que de una forma u otra me han apoyado, en especial a mis abuelos.

A la familia de mi novio especialmente a mis suegros Flori y Pedro y a mi cuñado Leo.

A mi tutor y oponente, por el apoyo.

Al tribunal de tesis por sus críticas muy oportunas para mejorar el documento y la exposición.

A mi tribunal de tesis en Venezuela Rosy, Dysan y Jorge y a mi tribunal del curso pasado por todas sus sugerencias.

A todas mis amistades que siempre me han ayudado y apoyado en etapas de mi vida en especial a mis hermanas de estudio Lorena y Ana Lizandra.

A aquellos amigos que me ayudaron cuando estuve tan lejos de mi familia: Dysan, Oscar, Aylén, Mabel, Yolanda, Consuelo, Jorge y Rosy.

A los profes que han estado en el proyecto y a los que continúan en él: Geovannys, Félix, Lisandra y Carlos.

A Rolando por ayudarme en el momento en que lo necesité.

Agradezco a la Revolución y en especial a Fidel por darme la oportunidad de formarme como profesional.

En fin a todas aquellas personas que de una forma u otra me brindaron su ayuda...



Dedico este trabajo de forma muy especial a:

A mis padres por su ejemplo y amor infinito.

A mi hermana Yaimé y a su esposo Alexander.

A mi hermanita pequeña Mariolis.

A mi gran amor Alejandro Blanco.

A mis suegros y a mi cuñado.

A mis abuelos.

A mis tíos y primos.

A mis amistades.

RESUMEN

Los avances tecnológicos han logrado el desarrollo de sistemas informáticos que permiten la producción de noticias donde interactúan medios de comunicación como la televisión y la informática, constituyendo una forma de uso de las Tecnologías de la Información y las Comunicaciones (TIC). Ante estos avances el uso de la informática en la sociedad cubana continúa tomando auge. Ejemplo de esto es la plataforma de televisión informativa PRIMICIA creada en el departamento de Señales Digitales del centro de desarrollo Geoinformática y Señales Digitales (GEySED) de la facultad 6, la cual fue desarrollada con software libre garantizando una soberanía tecnológica para el país. Dicha plataforma permite la gestión y transmisión de noticias en varios formatos a través de una red televisiva. En la actualidad resulta imprescindible gestionar y publicar las señales pues no existe una manera de monitorearlas, no se publican noticias en bloques así como también resulta necesario gestionar bloques noticiosos y perfeccionar la gestión de las señales externas y de cambios programados hacia una señal. Este documento recoge resultados y conclusiones de la investigación realizada con el objetivo de caracterizar las tecnologías y herramientas necesarias para el desarrollo del módulo. También se describen las funcionalidades de la aplicación exponiéndose a través de las descripciones de casos de uso. Se reflejan los artefactos generados durante las fases de análisis, diseño e implementación. Una vez programada la aplicación se prueban sus principales funcionalidades y se hacen recomendaciones para la implementación de versiones posteriores.

Palabras Claves: bloques noticiosos, canal, señal.

ABSTRACT

Technological advances have achieved the development of computer systems that allow the production of news, where, communications media such as television and informatics interact, constituting a form of use of Information and Communications Technologies (ICT). Given these advances the use of informatics in the Cuban society continues gaining momentum. An example is the informative television platform PRIMICIA created in the department of Digital Signals of the Geoinformatics and Digital Signals development center (GEySED) of the faculty, which was developed with free software ensuring technological sovereignty for the country. This platform allows the management and transmission of news in various formats via a TV network. Today it is essential to manage and publish the signals because there is no way to monitor them, news are not published in blocks and it is also necessary to manage news blocks and improve the management of external signals and scheduled changes into a signal. This document presents findings and conclusions of research done with the aim of characterizing the technologies and tools for the system development. It also describes the application functionality exposed through the use case descriptions. Reflect artifacts generated during the phases of analysis, design and implementation. After programming the application, its main functionalities are tested and recommendations for the development of further versions are made.

Keywords: channel, news blocks, signal.

Figura 1: Diagrama del modelo de dominio.....	20
Figura 2: Diagrama de CU del sistema	26
Figura 3: Prototipo de listar señales.....	29
Figura 4: Prototipo de adicionar señal.....	29
Figura 5: Prototipo de editar señal	30
Figura 6: Prototipo de filtrar señal	30
Figura 7: Prototipo de eliminar señal.....	30
Figura 8: Prototipo de mostrar logo de la señal.....	30
Figura 9: Prototipo de cambiar señal.....	32
Figura 10: Prototipo de deshabilitar cambios programados.....	32
Figura 11: Prototipo de listar bloques.....	36
Figura 12: Prototipo de adicionar bloque.....	36
Figura 13: Prototipo de editar bloque	36
Figura 14: Prototipo de filtrar bloque	37
Figura 15: Prototipo de eliminar bloque.....	37
Figura 16: Prototipo de publicar noticia	40
Figura 17: Prototipo de listar noticias publicadas en bloque.....	40
Figura 18: Prototipo de eliminar noticia publicada.....	40
Figura 19: Diagrama de clases del análisis CU Cambiar señal del canal.....	42
Figura 20: Diagrama de clases del análisis CU Programar cambio de señal.....	42
Figura 21: Diagrama de clases del análisis CU Publicar noticia	43
Figura 22: Diagrama de clases del análisis CU Gestionar señal	43
Figura 23: Diagrama de clases del análisis CU Gestionar bloque	43
Figura 24: Diagrama de clases del diseño del CU Cambiar señal.....	47
Figura 25: Diagrama de clases del diseño del CU Gestionar cambios programados	48
Figura 26: Diagrama de clases del diseño del CU Publicar noticia.....	48
Figura 27: Diagrama de clases del diseño del CU Gestionar señal.....	49
Figura 28: Diagrama de clases del diseño del CU Gestionar bloque.....	49
Figura 29: Diagrama de clases persistentes	50
Figura 30: Diagrama Entidad-Relación	51
Figura 31: Diagrama de despliegue	52
Figura 32: Diagrama de componentes del CU Cambiar señal.....	54
Figura 33: Diagrama de componentes del CU Gestionar cambios programados	55

Figura 34: Diagrama de componentes del CU Publicar noticia	55
Figura 35: Diagrama de componentes del CU Gestionar señal.....	56
Figura 36: Diagrama de componentes del CU Gestionar bloque.....	56

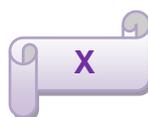


Tabla 1: Requerimiento de Hardware.....	23
Tabla 2: Actor del sistema.....	25
Tabla 3: Descripción del CU Gestionar señal.....	30
Tabla 4: Descripción del CU Cambiar señal del canal.....	32
Tabla 5: Descripción del CU Gestionar bloque.....	37
Tabla 6: Descripción del CU Publicar noticia.....	40
Tabla 7: Secciones a probar en el CU Gestionar señal.	60
Tabla 8: Descripción de variables del CU Gestionar señal.....	60
Tabla 9: Sección a revisar: Listar señales.....	60
Tabla 10: Sección a revisar: Adicionar señal.....	61
Tabla 11: Sección a revisar: Editar señal.	61
Tabla 12: Sección a revisar: Eliminar señal.....	62
Tabla 13: Sección a revisar: Filtrar señal.	62
Tabla 14: Sección a revisar: Mostrar logo de señal.....	62

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.	5
1.1 INTRODUCCIÓN	5
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	5
1.2.1 Plataforma de Televisión Informativa PRIMICIA.....	5
1.2.2 Módulo	5
1.2.3 Canal de televisión	6
1.2.4 Señal televisiva	6
1.2.5 Streaming.....	7
1.3 RESUMEN DE LOS CONCEPTOS ASOCIADOS.....	7
1.4 EL OBJETO DE ESTUDIO	7
1.5 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES	9
1.5.1 Ámbito internacional.....	9
1.5.2 Ámbito nacional.....	10
1.6 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	11
1.7 EL LENGUAJE UNIFICADO DE MODELADO (UML) COMO SOPORTE DE LA MODELACIÓN DE LA SOLUCIÓN PROPUESTA.....	12
1.8 TECNOLOGÍAS A UTILIZAR	12
1.8.1 Fundamentación de la herramienta de modelado.....	12
1.8.2 Fundamentación de los lenguajes	13
1.8.3 Fundamentación del framework	15
1.8.4 Fundamentación del entorno de desarrollo integrado (IDEs).....	16
1.8.5 Fundamentación del sistema de gestor de base de datos (SGBD).....	17
1.9 CONCLUSIONES DEL CAPÍTULO.....	17
CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES. .	19
2.1 INTRODUCCIÓN	19
2.2 MODELO DE DOMINIO.....	19
2.2.1 Conceptos o eventos fundamentales del dominio.....	19
2.2.2 Diagrama del modelo de dominio	20
2.3 ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.....	20
2.3.1 Requisitos funcionales.....	21
2.3.2 Requisitos no funcionales.....	22
2.4 REPRODUCTOR A UTILIZAR EN LA SOLUCIÓN	23

2.4.1 Plugin de VLC Media Player	23
2.4.2 Plugin de Media Player.....	24
2.4.3 Plugin gecko-mediaplayer	25
2.5 REPRODUCTOR SELECCIONADO	25
2.6 MODELO DE CASOS DE USO DEL SISTEMA	25
2.6.1 Definición de actores.....	25
2.6.2 Diagramas de casos de uso	26
2.6.3 Descripción textual de los casos de uso.....	26
2.7 CONCLUSIONES DEL CAPÍTULO.....	41
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES..	42
3.1 INTRODUCCIÓN	42
3.2 MODELO DE ANÁLISIS	42
3.2.1 Diagrama de clases del análisis	42
3.3 DESCRIPCIÓN DE LA ARQUITECTURA.....	44
3.3.1 Patrón arquitectónico.....	44
3.3.2 Patrones de diseño	45
3.4 MODELO DE DISEÑO	47
3.4.1 Diagrama de clases del diseño.....	47
3.5 DIAGRAMA DE CLASES PERSISTENTES	50
3.6 MODELO DE DATOS	50
3.6.1 Diagrama Entidad- Relación.....	51
3.7 MODELO DE DESPLIEGUE.....	51
3.7.1 Diagrama de despliegue.....	51
3.8 CONCLUSIONES DEL CAPÍTULO.....	52
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.	54
4.1 INTRODUCCIÓN	54
4.2 DIAGRAMAS DE COMPONENTES.....	54
4.3 PRUEBAS DEL SISTEMA.....	57
4.3.1 Casos de prueba	57
4.4 CONCLUSIONES DEL CAPÍTULO.....	63
CONCLUSIONES GENERALES.....	64
RECOMENDACIONES	65
REFERENCIAS BIBLIOGRÁFICAS.....	66

INTRODUCCIÓN

Las tecnologías de la información han transformado los medios de comunicación y suelen mezclarse con aspectos de la computación como la informática. Estos medios constituyen una herramienta persuasiva que permite mantener informada a la humanidad sobre sucesos de toda índole ocurridos en todo el planeta, ya sea político, social o económico. En las últimas décadas la televisión se ha convertido en el medio de comunicación más difundido y su importancia no ha dejado de crecer en todo el mundo. La televisión es el mayor medio de presión y de propaganda que se ha conocido hasta el momento estando presente en las casas de muchísimas personas. La televisión pudiera compararse con la internet ya que tienen como punto común que con su uso se puede vivir en directo algunos de los acontecimientos más importantes de los últimos años. La informática como ciencia surge a finales del siglo XIX y en pocos años experimentó un auge vertiginoso, lo que posibilitó que su uso fuera más allá de realizar simples operaciones de cómputo.

En el transcurso del siglo XXI la tecnología ha avanzado vertiginosamente, progresando en casi todos los campos de la ciencia. El país se encuentra en medio de un proceso de informatización que le ha permitido incursionar en muchas de las ramas del desarrollo de aplicaciones, destacándose algunas como: software educativo, software para la gestión de procesos, inteligencia artificial y televisión. Con el avance continuo de las Tecnologías de la Información y las Comunicaciones (TIC) y la necesidad de influir con la verdad cubana a la opinión pública internacional, se hace necesario realizar soluciones que transmitan noticias sobre Cuba y el mundo.

Con la creación en Cuba de la Universidad de Ciencias Informáticas (UCI) y el incremento de la matrícula estudiantil en esta institución surge la necesidad de crear un medio que permita mantener informada y actualizada a la gran comunidad universitaria de un modo rápido y eficiente donde se vinculen recursos multimedia como texto, imagen, video y sonido. Para dar solución a lo planteado anteriormente en el año 2005 se obtiene Señal 3, como primera versión de la plataforma, concebida para la divulgación de noticias importantes a los estudiantes de la UCI. Señal 3 está orientado a informar a la comunidad universitaria las 24 horas a través de un sistema de teletexto adaptado a las características del centro; el cual es transmitido por la red de televisión por cable que llega a cada sitio de la universidad. Con similares características surgió en el 2006 para la Agencia Cubana de Noticias el canal informativo Señal ACN, un sistema automatizado de teletexto para la Plataforma de Televisión Satelital Cubana que facilita la transmisión de noticias de los distintos medios de prensa a los cooperantes cubanos en el exterior y a los habitantes de lugares intrincados del país, llamados zonas de silencio. En el año 2007 se comienza la comercialización del producto en el extranjero, particularmente con el Ministerio de Energía y Petróleo de Venezuela (MENPET) donde se le

adicionaron otras funcionalidades debido a la necesidad del cliente de que fuera desarrollada con software libre, dando como resultado una nueva versión del producto al cual se le denominó TV Energía.

El desarrollo de estas soluciones ha posibilitado que el sistema se conceptualice como producto informático, decidiéndose desarrollar la Plataforma de Televisión Informativa, PRIMICIA, como una respuesta a la automatización de la transmisión de televisión informativa. Los beneficiados potenciales de esta plataforma son televisoras, agencias de noticias, terminales de transporte, universidades, hoteles, centros convencionales, sedes ministeriales o empresariales. La plataforma se encuentra en un proceso de mejoras y búsqueda de nuevas funcionalidades, mediante el cual se espera llegar a la versión 2.0 del producto.

En las versiones anteriores a PRIMICIA se reflejaron problemas como por ejemplo que en los canales informativos Señal 3 y Señal ACN no se gestionan las señales pues no están diseñados para interactuar con canales externos. En TVEnergía se transmiten señales externas de los canales Venezolana de Televisión y del canal interno de PDVSA, pero el producto no le brinda la posibilidad al usuario de ver uno de estos canales externos por el que se esté transmitiendo alguna noticia de importancia para el ministerio.

En PRIMICIA existe un módulo de gestión de la señal que presenta problemas al adicionar cambios programados hacia una señal pues en el mismo rango de fecha se pueden insertar varios cambios. Otro problema es que al cambiar hacia una nueva señal de manera inmediata, esta no se transmite hasta que la aplicación de transmisión realiza un chequeo de la base de datos para ver los cambios ocurridos en la misma. Actualmente la plataforma transmite por secciones temáticas agrupando noticias de un mismo tema y en caso de querer hacerlo por bloques noticiosos agrupando todas las noticias que serán mostradas, la misma no cuenta con este tipo de transmisión.

Partiendo de la situación anterior se define como problema de investigación: ¿Cómo garantizar la gestión, monitoreo y publicación de señales para el subsistema de administración de la Plataforma de Televisión Informativa PRIMICIA?

Tomando como punto de partida el problema de investigación se define como objeto de estudio, el proceso de gestión y publicación de señales en la Plataforma de Televisión Informativa PRIMICIA.

Se define como campo de acción, la automatización del proceso de gestión y publicación de señales.

El objetivo general de esta investigación está dirigido a: Desarrollar un módulo que garantice la gestión y publicación de la señal para el subsistema de administración de la Plataforma de Televisión Informativa.

Como guía de la investigación se plantea la siguiente idea a defender: Con el desarrollo del módulo de gestión y publicación de señales, se garantizará en la Plataforma de Televisión Informativa PRIMICIA

la gestión de bloques noticiosos, la publicación de noticias en bloques, el monitoreo de señales y se perfeccionará la gestión de señales externas y cambios programados hacia una señal.

Para dar cumplimiento a los objetivos propuestos se trazaron las siguientes tareas de la investigación:

1. Caracterización de soluciones existentes en el ámbito nacional e internacional que puedan contribuir al desarrollo del módulo de gestión y publicación de señales.
2. Descripción de los conceptos asociados al dominio del problema.
3. Caracterización de las tecnologías y herramientas a utilizar.
4. Identificación de los requisitos funcionales y no funcionales vinculados al proceso de gestión y publicación de señales.
5. Diseño del módulo de gestión y publicación de señales para el subsistema de administración de la Plataforma de Televisión Informativa PRIMICIA.
6. Implementación del módulo de gestión y publicación de señales para el subsistema de administración de la Plataforma de Televisión Informativa PRIMICIA.
7. Validación del sistema desarrollado.

Al finalizar el trabajo se espera obtener como posibles resultados: Artefactos vinculados con la metodología de desarrollo de software utilizada y el código fuente de la aplicación.

En la realización de esta investigación se utilizaron para dar cumplimiento a las tareas de investigación varios de los métodos científicos existentes, tanto teóricos como empíricos.

Se empleó el análisis histórico-lógico para establecer un estudio bibliográfico y una base sólida de fundamentos teóricos que permitan indagar sobre la evolución de los procesos de gestión y publicación de señales aprovechando los elementos teóricos que contribuyeran con el desarrollo de la aplicación.

Resultó indispensable sintetizar los documentos bibliográficos permitiendo extraer lo fundamental para el desarrollo de la investigación, evidenciándose el método analítico sintético.

La observación, que durante mucho tiempo constituyó la forma básica de obtener información científica de forma directa e inmediata, se utilizó para la formulación del problema a investigar y fue de gran utilidad en el diseño de la investigación durante toda la etapa de identificación, análisis y recopilación de datos, permitió obtener conocimiento acerca del proceso de gestión y publicación de señales y como éste se da en la realidad.

Capítulo 1: En este capítulo se brinda una visión general de los aspectos teóricos relacionados con el proceso de gestión y publicación de la señal interna y las externas. Se analizan soluciones similares existentes en el mundo. Se muestran las principales características de las herramientas, metodologías y tecnologías que serán empleadas en el desarrollo del sistema.

Capítulo 2: En este capítulo se presenta el modelo de dominio y se describen los principales conceptos reflejados en él. Se identifican las funcionalidades que presenta el módulo y las cualidades que debe cumplir. Además se realiza el diagrama de casos de uso del sistema.

Capítulo 3: En este capítulo se realiza el análisis del sistema a desarrollar con el objetivo de cumplir con todos los requerimientos definidos en el capítulo anterior. Se describe la arquitectura del sistema. Se desarrolla el modelo de diseño que contiene los diagramas de clases del diseño. Se genera el diagrama de entidad relación de la base de datos y el diagrama de despliegue.

Capítulo 4: En este capítulo se realiza el diagrama de componentes donde se muestra los principales componentes y sus relaciones. Además se realizan pruebas de caja negra con el objetivo de comprobar el correcto cumplimiento de los requisitos funcionales propuestos para el módulo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.

1.1 Introducción

El objetivo fundamental de este capítulo es abordar aspectos relacionados con el proceso de gestión y publicación de la señal interna y las externas. Por tal motivo resultó necesario hacer un análisis de los conceptos que giran alrededor del problema y del objetivo general de la investigación. Incluye además un análisis de las soluciones existentes a nivel internacional, nacional y en la universidad. Se fundamentan las metodologías de desarrollo y tecnologías usadas para dar solución al problema planteado.

1.2 Conceptos asociados al dominio del problema

1.2.1 Plataforma de Televisión Informativa PRIMICIA

El producto PRIMICIA provee un canal de teletexto para una red de televisión que tiene como objetivo la transmisión de noticias realizando una labor informativa y formativa en cualquier entidad donde este tipo de sistema sea una necesidad. La aplicación está orientada a proveer un canal de televisión básicamente con características informativas, donde se integran contenidos de texto, imagen, audio y video. **(1)**

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. En el subsistema de administración se realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten y el subsistema de transmisión es el encargado de la visualización de las noticias y materiales publicados.

1.2.2 Módulo

Un módulo es una pieza o un conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica. El módulo, por lo tanto, forma parte de un sistema y mantiene algún tipo de relación o vínculo con el resto de los componentes. **(2)**

Según el Diccionario de la Real Academia de la Lengua Española (DRAE) el término módulo, en cuestiones de programación se define como:

“*Software* que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión”. **(3)**

En programación, un módulo es una parte de un programa, que realizará una o varias tareas del total que debe realizar el programa para cumplir con sus objetivos. Un módulo está compuesto por al menos una función o procedimiento; o viceversa un procedimiento o función es un módulo. Una ventaja de dividir un programa en módulos es que hace el código más entendible y como cada módulo es una unidad independiente de otro, estos pueden ser ejecutados y compilados por separado.

1.2.3 Canal de televisión

Un canal de televisión es un tipo de estación emisora que transmite audio y video a receptores de televisión en un área concreta. Tradicionalmente, los canales de televisión hacían sus transmisiones enviando señales de radio especialmente codificadas por el aire, llamada televisión terrestre. Normalmente se conceden licencias por parte de una agencia gubernamental a cada canal de televisión para usar una parte determinada del espectro de radio (un canal) a través del cual ellos envían sus señales. **(4)**

1.2.4 Señal televisiva

Se puede asumir que señal es un valor que proporciona cierta información, que puede ser interpretada o utilizada en función de un objetivo determinado. La señal televisiva es una señal que es producida y transmitida. Una señal vía streaming es una señal generada por un flujo de datos de audio o video proveniente de un servidor de streaming.

La transmisión de señal se puede realizar a través del aire, cable o por satélite. La transmisión por aire es aquella señal que llega al usuario procedente de un emisor o de un repetidor terrestre. Por satélite la señal que llega al usuario procedente de un satélite artificial. Esta trae como ventaja la posibilidad de llegar a toda la superficie de un territorio concreto, facilitando el acceso a zonas muy remotas y aisladas haciendo que los programas de televisión lleguen a todas partes. Si la comunicación se realiza mediante cable, se permite incorporar un canal de retorno que permite al usuario enviar información. La televisión por cable surge por la necesidad de llevar señales de televisión y radio, de índole diversa, hasta el domicilio de los abonados, sin necesidad de que éstos deban disponer de diferentes equipos receptores, reproductores y sobre todo de antenas. **(5)**

1.2.5 Streaming

Consiste en la distribución de audio o video por Internet. La tecnología de *streaming*, se utiliza para aligerar la descarga y ejecución de audio y video, ya que permite escuchar y visualizar los archivos mientras se están descargando.

El *streaming* se diferencia de una simple transferencia de ficheros en que el cliente reproduce la información mientras la está recibiendo, sin necesidad de esperar la descarga completa de un archivo.

(6)

1.2.6 Bloque noticioso

Un bloque noticioso es un elemento donde se agrupan noticias que serán mostradas en un horario determinado. Cada bloque está formado por una determinada cantidad de noticias que el total de sus duraciones no sobrepase a la del bloque.

1.3 Resumen de los conceptos asociados

Al analizar los conceptos anteriormente citados se resume que:

- Primicia es una plataforma de televisión informativa que permite la gestión, administración y transmisión de noticias combinando recursos multimedia.
- Un módulo es un complemento de un software, normalmente representa a una página web o a un grupo de páginas web con un propósito relacionado de acuerdo a su función en la aplicación.
- Un canal de televisión es un emisor por el cual se envían diferentes señales.
- Una señal es la programación que se ve en los televisores y puede captarse por aire, cable o satélite. Una señal streaming es generada por un flujo de datos proveniente de un servidor de streaming.
- Streaming es una forma de reproducción de audio y video en la web sin necesidad de descargarlos.
- Bloque noticioso es un elemento donde se agrupan noticias que serán mostradas.

1.4 El objeto de estudio

La Plataforma de Televisión Informativa PRIMICIA constituye una solución integral que permanece en constante evolución y perfeccionamiento capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos. La aplicación está orientada a proveer un canal de televisión donde se integren contenidos de texto, imagen y video. No está atada a un diseño gráfico y no depende de un entorno dado. Es una solución integral que puede ser ajustable a diferentes clientes, obteniendo de esta manera un producto informático del cual pueden beneficiarse

Capítulo 1: Fundamentación teórica.

cualquier otra entidad que tenga una red de televisión con necesidad de mantener informados a visitantes, clientes y trabajadores. **(1)**

Actualmente el tipo de transmisión con el que cuenta la plataforma es por sección temática donde se encuentran agrupadas noticias por un interés común, permitiendo establecer el orden en que serán mostradas las secciones, se comienza con la primera sección temática y las noticias asociadas a la misma, una vez que termine con estas se pasa a la otra y así sucesivamente, creando un ciclo de transmisión mientras exista vigencia en las noticias publicadas. En caso que exista una sección temática que no tenga noticias asociadas esta no es mostrada. Resulta necesario que además se transmita por bloques noticiosos. Para dar solución a esta problemática estos deben ser gestionados y se deben publicar noticias dentro de ellos. Las noticias agrupadas dentro de cada bloque son aquellas que fueron revisadas y están listas para transmitir. A cada bloque se le establece el horario en que saldrá al aire y durante ese tiempo se muestran todas las noticias asociadas al mismo donde la duración de las mismas no debe exceder a la del bloque.

PRIMICIA permite que se transmitan señales externas la problemática aquí es que la aplicación no brinda la posibilidad de monitorear dichas señales y la idea es enlazarlas al canal interno cuando por estas se está transmitiendo alguna información de interés. El monitoreo de señales es visualizar en la web las direcciones streaming de la señal interna y externas que se encuentran en base de datos. Hasta el momento cuando el usuario desea estar al tanto de la transmisión de los canales externos tiene que auxiliarse de televisores haciendo llegar los canales externos a través de la red de televisión cableada del centro. Lo mismo sucede en caso de que se quiera dar seguimiento a la señal del canal interno. Por lo planteado anteriormente es necesaria la visualización del canal interno y de los externos.

Cuando el subsistema de transmisión efectúa el cambio hacia un nuevo canal realiza peticiones periódicas a la base de datos para detectar que han surgido nuevos cambios. Esto trae consigo la demora en los cambios de canales, la sobrecarga en la red y en el acceso a la base de datos. Tiene un gran significado que la transmisión capture dichos cambios de una forma rápida y directa desde el subsistema de administración, esto podría solucionarse mediante el mecanismo de comunicación entre procesos.

Cuando se desea cambiar hacia una señal en un momento determinado es necesario insertar un cambio programado hacia la señal deseada, para ello se debe tener en cuenta si las fechas de inicio y de fin del que se va a insertar se encuentra en el rango de la de otros cambios de señal almacenados en la base de datos, es decir que exista solapamiento entre ellos y en caso de existir no debe permitirse la inserción del mismo. También se debe tener presente que un cambio manual tiene más

prioridad y si en ese tiempo hay un cambio programado, se dejará de transmitir la señal a la que se cambió manualmente y comenzará a transmitirse la relacionada con el cambio que se encuentra programado, por ende es importante deshabilitar los cambios programados en el momento en que se realice uno manual.

En PRIMICIA hasta el momento se ha centrado la investigación en el perfeccionamiento del proceso de redacción y transmisión de las noticias, sin embargo resulta imprescindible lograr garantizar el monitoreo y publicación de la señal interna y de las externas y perfeccionar la gestión de las mismas.

1.5 Análisis de otras soluciones existentes

Actualmente existen pocas soluciones informáticas que pudieran aportar al desarrollo del Módulo gestión y publicación de señales.

1.5.1 Ámbito internacional

Los sistemas que realizan estas funciones a nivel internacional están desarrollados con fines privativos por lo que invertir en la utilización de estos implicaría gastos significativos debido a la gran cantidad de funcionalidades con las que cuentan. El producto Multiscreen de Estructure Media Systems es la solución informática que más puede servir en dicha investigación.

Estructure es básicamente una avanzada plataforma sobre la que se construyen y configuran sistemas informáticos de gestión y explotación de contenidos audiovisuales, integrando todas las funcionalidades y tareas necesarias de forma flexible y modular, permitiendo una correcta y ágil organización, explotación y flujo del trabajo. **(7)**

Estructure presenta el producto Multiscreen, un sistema de visionado multipantalla de última generación. Multiscreen ofrece una excelente calidad de imagen debido a su resolución de hasta 3800x2040 píxeles. Permite configurar y personalizar diferentes espacios de trabajo a los que el usuario puede acceder rápidamente mediante teclas predefinidas. Incorpora además, ajustes de brillo, contraste y color independientes para cada canal. Es la opción más rentable para aquellos entornos que trabajen con múltiples señales de vídeo, como pueden ser salas de control y supervisión, ya que el sistema permite optimizar el espacio y reducir los costes pues permite aprovechar al máximo el espacio de trabajo. Entre las principales características técnicas de Multiscreen se encuentran: **(8)**

- Hasta 16 canales de entrada de video
- Entradas en SDI, Compuesto, HDMI y DVI.

- 2 salidas de video HDMI y DVI.
- Resolución de salida hasta 3800x2040 píxeles.
- Formatos 4:3 y 16:9.
- Vúmetros independientes para cada canal.
- Códigos de tiempo insertados en vídeo.
- Creación rápida de sets personalizables por el usuario.
- Asignación de teclas de acceso rápido a sets predefinidos.
- Múltiples relojes configurables por el usuario.

El problema de utilizar este producto es que necesita de requerimientos de hardware muy avanzados y resulta difícil la adquisición de estos debido a que tienen elevados precios por concepto de equipamiento tecnológico. Además que tiene como desventaja que es una herramienta privativa lo cual implica grandes gastos en compra de licencias.

1.5.2 Ámbito nacional

Como principal antecedente de PRIMICIA en el ámbito nacional, se encuentra TV Energía, la cual sirvió para valorar el proceso de gestión de la señal. TV Energía es un sistema informático que utiliza el Ministerio de Energía y Petróleo de Venezuela (MENPET) para mantener informados a los trabajadores y visitantes del ministerio sobre la actividad ministerial y presidencial en el ámbito nacional e internacional. A continuación se explicará detalladamente este proceso.

El proceso de gestión de la señal comienza una vez que el usuario accede a la funcionalidad de gestionar la señal del canal contando con los privilegios necesarios para esto. Después se le brinda la oportunidad de cambiar la señal que se está transmitiendo o programar cambios de señal que se realizarán en un momento determinado. Cuando se listan los cambios que se encuentran programados y las señales externas en el caso de decidir cambiar o eliminar alguno, el sistema le brinda esta posibilidad.

TV Energía sirve como base del desarrollo del módulo de gestión y publicación de señales pero hace falta optimizar el proceso y agregar nuevas funcionalidades debido a los problemas que traen consigo su uso y que fueron mencionados en el análisis del objeto de estudio.

1.6 Metodología de desarrollo de software

Un proceso de desarrollo de software puede hacerse tan complejo como el producto lo requiera y muchas veces el control de dicho proceso se va de las manos de los desarrolladores debido a la no utilización de alguna metodología que guíe de manera coherente y eficiente todas sus actividades.

Para evitar insatisfacciones de los clientes, aumento de presupuesto del producto y mayor carga de trabajo por la parte de los desarrolladores, es que se aconseja basar todo el trabajo en una metodología de desarrollo de software, principalmente en una que se adecúe a las características del proyecto, del ambiente de los desarrolladores y del producto final como tal.

Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto y guías para uso de herramientas de apoyo. Utiliza el lenguaje unificado de modelado UML.

El Proceso Unificado de Desarrollo (RUP) es una metodología de desarrollo de software que está basado en componentes e interfaces bien definidas y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Como RUP es un proceso, en su modelación define como sus principales elementos: **(9)**

RUP tiene 3 características esenciales: **(9)**

- Dirigido por los casos de uso: Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño, constituye la arquitectura del producto que se desea desarrollar.
- Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

RUP unifica todo el equipo de desarrollo de software y optimiza su comunicación. Unificando al equipo se simplifica la comunicación, asegurando la asignación de recursos en forma eficiente, la entrega de los artefactos correctos y el cumplimiento de los tiempos límite. Esta metodología se centra en obtener al final del ciclo de vida del proyecto un producto documentado e implementado con calidad.

Teniendo en cuenta la metodología que se adapta más al medio en que se desarrolla el software y a los objetivos finales que se quieren cumplir se seleccionó el Proceso Unificado de Desarrollo de Software (RUP). Para mejorar la plataforma se incluyen constantemente nuevas funcionalidades perfeccionadas y personalizadas, trayendo esto consigo que su desarrollo sea a largo plazo y esta metodología es la más adaptable para proyectos de este tipo. Otra de las ventajas que trae consigo su

uso es que genera abundante información imprescindible para un equipo de trabajo que se renueva constantemente como PRIMICIA.

1.7 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.

El Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software: UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso; es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. **(9)**

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real.

Se escogió UML como lenguaje de modelado debido a que el proceso unificado de desarrollo (RUP) lo utiliza para la modelación de los artefactos generados durante sus flujos de trabajo.

1.8 Tecnologías a utilizar

Todo sistema automatizado necesita para su desarrollo un conjunto de tecnologías. A continuación se presenta un análisis de estas, las cuales fueron agrupadas en: herramienta de modelado, lenguajes de programación, plataformas o *frameworks*, entornos de desarrollo integrados (IDE) y sistema gestor de base de datos (SGBD).

Por políticas del proyecto PRIMICIA del departamento de Señales Digitales del centro de desarrollo GEySED se definió que en el desarrollo del mismo se utilizará la herramienta de modelado Visual Paradigm, el lenguaje de programación PHP y JavaScript, el *framework* Symfony, el entorno de desarrollo integrado NetBeans y el sistema gestor de base de datos PostgreSQL, por tanto no fue necesario realizar una comparación con otras herramientas, sólo se fundamentan las que se encuentran seleccionadas.

1.8.1 Fundamentación de la herramienta de modelado

En la actualidad muchas empresas se han extendido a la adquisición de herramientas CASE¹. Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a

¹ Siglas que corresponden a: Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación

los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Las principales ventajas que proporcionan las herramientas CASE consisten en el aumento de la productividad y de la calidad de las aplicaciones desarrolladas. **(10)**

Las herramientas CASE ayudan a los desarrolladores a expresar gráficamente los procesos de negocio y sus correspondientes modelos de datos, validando la integridad y consistencia de los mismos y generando en muchos casos el código necesario para convertir dichos modelos en nuevas aplicaciones. Así mismo, las herramientas CASE pueden producir rápidamente prototipos de las aplicaciones a desarrollar, los cuales se pueden probar, corregir e incorporar a la aplicación final.

Visual Paradigm para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Entre los beneficios que aporta para la creación del módulo, es que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño permitiendo representar los tipos de diagramas de clases y la documentación desde estos diagramas.

1.8.2 Fundamentación de los lenguajes

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Aunque muchas veces se usa lenguaje de programación y lenguaje informático como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el HTML. **(11)**

Preprocesador de hipertexto (PHP):

Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una librería de funciones y mucha documentación. Se encuentra embebido dentro del HTML y utilizado mayormente para desarrollar servicios Web. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web. **(12)**

Generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas

que no funcionan, además no se está forzado a pagar actualizaciones anuales para tener una versión que funcione. **(13)**

Entre los beneficios que aporta para la creación del módulo es que es un lenguaje de código libre y tiene abundante documentación. Una ventaja de utilizarlo es que es muy fácil integrar el acceso a bases de datos en páginas HTML. Además porque no requiere demasiados recursos de sistema, utiliza su propio sistema de administración de recursos y dispone de sofisticado método de manejo de variables, conformando un sistema robusto y estable. Como está embebido en HTML es muy fácil ejecutarlo en el servidor. También porque es un potente lenguaje de programación orientado a objeto para el desarrollo de aplicaciones web.

Lenguaje de Marcas Hipertextuales (HTML):

Es un lenguaje estático para el desarrollo de sitios Web. Los documentos HTML no son documentos de texto normal, sino documentos de hipertexto, es decir, texto presentado de forma estructurada y agradable, ya que en el propio documento aparecen enlaces a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). Coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. **(14)**

Entre los beneficios que aporta para la creación del módulo es que es un lenguaje de fácil aprendizaje, las páginas escritas con él son archivos pequeños y son desplegadas con rapidez. Además porque lo admiten la mayoría de los navegadores y hay bastante documentación sobre este lenguaje. Es independiente ya que puede ser escrito en cualquier editor de texto.

JavaScript:

Es un lenguaje interpretado, no requiere compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Utilizado principalmente en páginas Web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript. El código JavaScript puede ser integrado dentro de las páginas Web. Es un lenguaje de scripting seguro y fiable, los script tienen capacidades limitadas, por razones de seguridad. El código JavaScript se ejecuta en el cliente, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. **(14)**

Entre los beneficios que aporta para la creación del módulo es que es simple. Es dinámico, responde a eventos en tiempo real. Con el JavaScript se consigue la validación de la información como datos,

controla el movimiento de imágenes, menús interactivos y animaciones de elementos. Además, es soportado por todos los navegadores, es sencillo de aprender y está contemplado por los distintos organismos de normalización.

Hojas de estilo en cascada (CSS):

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, y por extensión en XHTML. Su acrónimo en inglés de HyperText Markup Language. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. **(15)**

Entre los beneficios que aporta para la creación del módulo es que por un lado, se evita hacer a los archivos demasiado pesados, y se define el estilo visual de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta, para cada una de las páginas. Por otro, se trabaja con estándares, y se separa hasta cierto punto la estructura de la presentación, logrando una manera más nítida de trabajar, y lo que es más: en un sencillo documento CSS, se define una plantilla para todo un sitio.

1.8.3 Fundamentación del framework

El término framework se utiliza constantemente en el desarrollo de software. Un framework es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Es necesaria la utilización de un framework debido a que el cambio en robustez y rendimiento es muy grande respecto a la programación PHP sin *frameworks*.

Es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proveyendo una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación. **(16)**

Symfony

Es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación 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. **(17)**

Symfony simplifica al máximo el desarrollo de aplicaciones web profesionales con PHP, utilizando las mejores prácticas y los patrones de diseño más importantes. Además ha incorporado ideas propias y el resultado es un framework elegante, estable, productivo y muy bien documentado. Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de

mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony es fácil de instalar y configurar en la mayoría de plataformas, es independiente del sistema gestor de bases de datos, es sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos, sigue la mayoría de mejores prácticas y patrones de diseño para la Web, es lo suficientemente estable como para desarrollar aplicaciones a largo plazo, código fácil de leer y que permite un mantenimiento muy sencillo y además es fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Entre los beneficios que aporta para la creación del módulo es que emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación Web. Desde su primera versión ha sido creado sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características. El uso de Symfony, reducirá considerablemente los costos de desarrollo y el tiempo.

JQuery

Es una biblioteca o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas Web. Posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. **(18)**

JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Este framework ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Las librerías del framework permiten realizar la programación mucho más rápida y libre de errores.

1.8.4 Fundamentación del entorno de desarrollo integrado (IDEs)

Un entorno de desarrollo integrado o en inglés Integrated Development Enviroment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. **(19)**

El uso de un IDE posibilita ahorrar esfuerzo y tiempo en escribir un programa. Netbeans es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Es un producto libre y gratuito sin restricciones de uso.

NetBeans elimina la necesidad de los equipos de desarrollo que tienen que invertir demasiado tiempo manteniendo los modelos actualizados con revisiones exhaustivas que garanticen la actualización de indicadores y códigos de anotación. Esta característica puede ahorrar al desarrollador muchas horas de trabajo y acelerar la disponibilidad de un nuevo programa. **(20)**

Las características de NetBeans de flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio.

Entre los beneficios que aporta es que con el uso de NetBeans como entorno de desarrollo, se simplifica la tarea de programación enormemente, permitiendo centrarse en las particularidades de la aplicación a desarrollar y evitando la complejidad inherente a cualquier desarrollo sobre una plataforma gráfica.

1.8.5 Fundamentación del sistema de gestor de base de datos (SGBD)

Un Sistema gestor de bases de datos (SGBD) es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. Gestiona la estructura física de los datos y su almacenamiento. **(21)**

Consiste en una colección de datos interrelacionados y una colección de programas para acceder a los datos. Proporciona un entorno conveniente y eficiente para los usuarios que lo usan para la recuperación y almacenamiento de la información.

PostgreSQL es un sistema de base de datos denso, lo cual a menudo se describe como la versión de código abierto de Oracle. Puede comprimir y descomprimir sus datos sobre la marcha con un rápido sistema de compresión para encajar más datos en un espacio de disco asignado. La ventaja de los datos comprimidos, además de ahorrar espacio en disco, es que la lectura de datos demora menos, por lo que lee datos con mayor rapidez. Ofrece una garantía de integridad en los datos muy fuerte. Presenta una buena escalabilidad y rendimiento bajo grandes cargas de trabajo. **(22)**

Entre los beneficios que aporta PostgreSQL para la creación del módulo es que es libre, publicado bajo la licencia BSD, no hay costo asociado a la licencia del software ni al código fuente. Además es uno de los sistemas gestores de base de datos más potentes, robustos y seguros.

1.9 Conclusiones del capítulo

El análisis de los conceptos asociados permitió una mejor comprensión del dominio del problema y servirá de apoyo para el estudio, reflexión y entendimiento de la investigación. El estudio de soluciones similares permitió acumular información para la implementación del proceso de gestión y publicación

Capítulo 1: Fundamentación teórica.

de señales. Se utilizarán las siguientes herramientas en el desarrollo del sistema, PHP para la programación del lado del servidor, JavaScript y JQuery para hacer la interfaz de usuario más amigable, CSS para definir el estilo visual de las páginas web, HTML para mostrar contenido estático, Symfony para ahorrar tiempo de desarrollo, el entorno integrado de desarrollo Netbeans para simplificar la programación, el gestor de base de datos PostgreSQL para almacenar información y la herramienta CASE Visual Paradigm para generar los artefactos durante el desarrollo del software.

CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.

2.1 Introducción

En este capítulo se presenta el modelo de dominio y se describen los conceptos reflejados en él. Se hace un análisis de reproductores que sean capaces de visualizar las señales externas, las cuales tienen direcciones streaming. Se plantean además los requerimientos funcionales y no funcionales que guían el desarrollo del trabajo y se realiza el diagrama de casos de uso del sistema.

2.2 Modelo de dominio

El modelo del dominio es una representación de los conceptos u objetos del mundo real, significativos para un problema. Tiene como objetivo fundamental la descripción de las clases más importantes en el sistema y representa conceptos del mundo real. Define un modelo de clases común para todos los implicados en el desarrollo, sirve como interlocutor entre clientes y desarrolladores. **(23)**

El propósito fundamental de este modelo es generar una terminología común y sentar las bases del entendimiento del desarrollo y no para definir el sistema completo. Los objetos del dominio representan los eventos que suceden en el entorno en que trabaja el sistema.

Debido a la no existencia de una definición clara de los procesos de negocio, se decide desarrollar un modelo de dominio, el cual abarca las definiciones asociadas a los conceptos encontrados en el entorno donde está enmarcado el sistema, así como las relaciones existentes entre ellos.

2.2.1 Conceptos o eventos fundamentales del dominio.

Para brindar una mejor comprensión del diagrama del modelo de dominio a continuación se realiza una breve descripción de los conceptos o eventos encontrados en el problema.

Primicia: Es una plataforma que permite la administración y transmisión de noticias en diferentes formatos mediante un canal de televisión.

Transmisión: Transmite un canal que recibe las señales que pueden ser la interna o las externas. Es el envío y recepción de la información en forma de noticias a través de un equipo informático por medio de señales encaminadas en un canal.

Canal: Es el emisor por el cual se van a enviar las diferentes señales. Por él se transmite audio y video a receptores de televisión en un área concreta.

Señal: Es la señal que es transmitida por el Subsistema de Transmisión.

Externa: Son aquellas con las que enlazará la plataforma para una transmisión en vivo. Pertenece a una entidad en particular y no fue generado por las noticias de PRIMICIA.

Interna: Es la del propio canal, es decir aquella generada por las noticias y bloques.

Cambios de señal: Son cambios programados a transmitirse en una fecha y hora indicada y que tienen una señal asociados.

Noticia Publicada: Son aquellas noticias que se encuentran listas para publicar, estas puede que se hayan obtenido por fuentes web o por otra vía.

Bloque: Es un elemento utilizado para agrupar las noticias que serán mostradas.

2.2.2 Diagrama del modelo de dominio

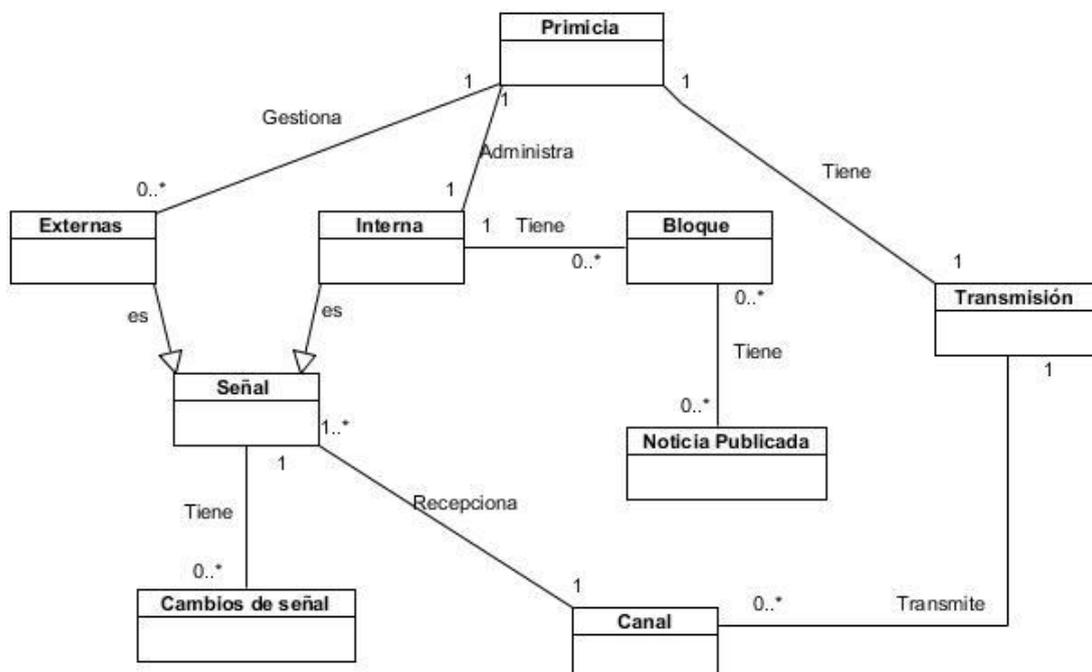


Figura 1: Diagrama del modelo de dominio

2.3 Especificación de los requisitos del software

Los requisitos se pueden clasificar en: funcionales y no funcionales. Los requerimientos deben ser descritos como una característica del sistema a entregar.

Los requisitos se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Un gran número de proyectos de software naufragan debido a una mala definición, especificación o administración de requisitos. La calidad con que se realice la captura de los requisitos influye en todo el proceso de desarrollo del software, repercutiendo en el resto de las fases de desarrollo del mismo.

(24)

2.3.1 Requisitos funcionales

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

RF 1: Insertar señal: El sistema debe permitir que se inserten señales.

RF 2: Modificar señal: El sistema debe permitir que se modifique una señal seleccionada.

RF 3: Eliminar señal: El sistema debe permitir que se elimine una señal o varias señales.

RF 4: Listar señales: El sistema debe permitir que se muestre un listado de las señales.

RF 5: Filtrar señal: El sistema debe permitir la búsqueda de señales según el nombre y si se encuentra activa o no.

RF 6: Mostrar logo de señal: El sistema debe permitir que se le muestre al usuario el logo de la señal seleccionada.

RF 7: Visualizar señales: El sistema debe permitir que se visualicen la señal interna y las externas.

RF 8: Cambiar señal: El sistema debe permitir que se active la señal a la que el usuario desea cambiar y se le envíe un mensaje a la transmisión informando el cambio.

RF 9: Insertar cambio programado: El sistema debe permitir que se inserten cambios de señal. La fecha de inicio y fecha de fin del cambio no puede estar en el rango de la de otros cambios.

RF 10: Eliminar cambio programado: El sistema debe permitir que se elimine un cambio de señal. Debe permitir que se le envíe un mensaje a la transmisión informando lo ocurrido.

RF 11: Modificar cambio programado: El sistema debe permitir que se modifique un cambio de señal. Debe permitir que se le envíe un mensaje a la transmisión informando lo ocurrido.

RF 12: Listar cambio programado: El sistema debe permitir que se muestre un listado de los cambios programados.

RF 13: Filtrar cambio programado: El sistema debe permitir que el usuario pueda buscar un cambio según la señal asociada al mismo.

RF 14: Insertar bloque: El sistema debe permitir que se inserten bloques. La fecha de inicio y fecha de fin del bloque no puede estar en el rango de la de otros bloques.

RF 15: Modificar bloque: El sistema debe permitir que se modifique un bloque seleccionado. Debe permitir que se le envíe un mensaje a la transmisión informando lo ocurrido.

RF 16: Eliminar bloque: El sistema debe permitir que se elimine un bloque o varios bloques. Debe permitir que se le envíe un mensaje a la transmisión informando lo ocurrido.

RF 17: Listar bloques: El sistema debe permitir que se muestre un listado de los bloques.

RF 18: Filtrar bloque: El sistema debe permitir la búsqueda de bloques según el nombre y si se encuentra activo o no.

RF 19: Asignar noticias a bloque: El sistema debe permitir que se asignen noticias a los bloques.

RF 20: Ordenar noticias del bloque: El sistema debe permitir ordenar las noticias dentro de un bloque siguiendo un orden seleccionado por el usuario.

RF 21: Eliminar noticias del bloque: El sistema debe permitir que se eliminen noticias que se encuentran publicadas en un bloque.

2.3.2 Requisitos 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. Las propiedades no funcionales pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Usabilidad

1. El sistema de forma general debe brindar gran facilidad de uso para personas con conocimientos básicos de computación.
2. Las funcionalidades deben ser claras y se debe mostrar la información de forma lógica y correctamente estructurada.

Soporte

1. El soporte y/o mantenimiento del sistema no debe necesariamente detener la transmisión del canal.

Confidencialidad

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

Disponibilidad

1. El sistema debe ser accesible los 7 días de la semana y las 24 horas del día.

Restricciones de diseño e implementación

1. Para la modelación del sistema se utilizará como lenguaje de modelado UML y herramienta Visual Paradigm 8.0.
2. Se requiere el uso de la arquitectura Modelo-Vista-Controlador implementada por el framework Symfony 2.
3. El sistema estará implementado en lenguaje PHP 5.3, utilizando como IDE de desarrollo el

NetBeans 6.8 o superior.

Interfaz.

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

Software

1. Se debe utilizar como servidor web el “Apache 2” y como sistema gestor de base de datos PostgreSQL 8.4 o superior.
2. Se debe utilizar como navegador web Firefox 3.6 o superior.

Hardware

A continuación se muestran los requerimientos recomendados que debe cumplir el equipamiento tecnológico de la plataforma:

	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red
Servidor de administración	Pentium IV 2.8 GHz	1 GB	120 GB	Ethernet 10/100 Mbps
Servidor de transmisión	Pentium IV 2.8 GHz	512 MB	40 GB	Ethernet 10/100 Mbps
PC Cliente	Intel Celeron 2.6 GHz	256 MB	40 GB	Ethernet 10/100 Mbps

Tabla 1: Requerimiento de Hardware

2.4 Reproductor a utilizar en la solución

Para dar solución al requisito funcional mostrar vistas de monitoreo, donde se visualizarán en la web la señal interna y externas es necesario utilizar un reproductor, por tanto se realizó un análisis para seleccionar uno que brinde tal posibilidad y sea capaz de reproducir las señales, las cuales son direcciones *streaming*. Después de un minucioso análisis se decidió usar un *plugin* para el navegador. Se descartaron para el análisis aquellos *plugin* que son privativos, como el *plugin* de RealPlayer, el *plugin* Microsoft Silverlight, el complemento del QuickTime publicado por Apple Computer Inc y disponible para Mac y Windows y el *plugin* de Windows Media Player para Firefox.

A continuación se analizarán los *plugin* libres VLC Media Player, Media Player y *gecko-mediaplayer* para definir cuál es el idóneo para reproducir *streaming* en la web.

2.4.1 Plugin de VLC Media Player

Uno de los programas más utilizados por los usuarios de GNU/Linux, más allá de la distribución con la que trabajen, es el denominado VLC Media Player. Es un reproductor multimedia de código abierto que

reproduce la mayoría de los formatos de audio y video. Incluye un plugin para reproducir contenido multimedia dentro del navegador Firefox.

Para utilizar las funciones avanzadas del *plugin* se pueden usar funciones de JavaScript para controlar la reproducción o extraer información del *plugin*. Por el diseño de su interfaz gráfica resulta a simple vista una aplicación sencilla, y es capaz de reproducir archivos contenidos en la PC, como también videos online, sin necesidad de descargarlos.

Por otra parte, el VLC Media Player permite reproducir archivos incompletos e incluso dañados, e incluye una función para reparar los daños que pueden llegar a tener los videos, y la posibilidad de utilizar filtros de post-producción, como ajustes de brillo y contraste de las imágenes. VLC incluye de forma nativa un gran número bibliotecas de códec, reduciendo la necesidad de instalar o calibrar códec propietarios. Muchos de los códec incluidos en VLC son proporcionados por la biblioteca libavcodec del proyecto FFmpeg, aunque principalmente utiliza sus propios filtros de multiplexación. **(25)**

Soporta varios códec de audio y video, así como diferentes formatos de archivos, además de DVD, VCD y varios protocolos de streaming; también tiene la capacidad de transmitir datos streaming a través de redes y convertir archivos multimedia en formatos distintos al original. **(25)**

VLC Media Player es totalmente gratuito y se distribuye bajo los términos de la licencia GPL. Es uno de los reproductores más independientes en cuanto a plataforma se refiere, con versiones para GNU/Linux, Microsoft Windows, Mac OS X, BeOS y BSD.

2.4.2 Plugin de Media Player

Este plugin es otro de los programas más elegidos por los usuarios de GNU/Linux, ya que es considerado uno de los mejores de su tipo, por una serie de detalles que lo caracterizan.

En principio, cabe destacar que Mplayer utiliza códec nativos incluidos en su librería libavcodec, la cual está basada en el proyecto FFmpeg, y por otra parte incorpora códec binarios, que permiten la reproducción de una enorme cantidad de formatos de video. **(25)**

Con Mplayer es posible reproducir archivos multimedia contenidos en la PC, además es capaz de reproducir streaming y diversos tipos de videos online, por intermedio de los protocolos HTTP, FTP, MMS y RTSP/RTP.

Al igual que el VLC Media Player, el Mplayer es capaz de soportar diferentes formatos de archivos de subtítulos, reparar archivos dañados, y utilizar diferentes filtros para mejorar la reproducción de la imagen de los archivos multimedia.

Asimismo, la reproducción de DVD ya no es un inconveniente, ya que el Mplayer incluye los códec necesarios que encuentra en las librerías libdvdread y libdvdcss, mediante lo cual es posible navegar por los menús de los DVDs. **(25)**

2.4.3 Plugin gecko-mediaplayer

El plugin gecko-mediaplayer es un plugin de código abierto que permite reproducir la mayoría de los formatos de audio y video utilizando GNOME MPlayer. Utiliza a GNOME MPlayer, que a su vez utiliza a Mplayer, y que permite visualizar vídeos en una gran cantidad de formatos, casi tantos como VideoLan Client. Funciona con todos los navegadores compatibles con el API NS4 (Firefox, Opera, entre otros). **(26)**

Se trata de un complemento para Firefox que permite ver vídeos incrustados y además se integra perfectamente con el escritorio. Tiene como desventaja que no se encuentra disponible para las plataformas Mac y Windows.

2.5 Reproductor seleccionado

Con el análisis de estos plugin para Mozilla Firefox se decide utilizar el plugin de VLC Media Player, el cual además de ser totalmente gratuito permite la reproducción de videos sin necesidad de descargarlos por intermedio de los protocolos HTTP, FTP, MMS, UDP y RTSP/RTP. Permite visualizar videos en una gran cantidad de formatos y se puede utilizar en varios sistemas operativos.

2.6 Modelo de casos de uso del sistema

El modelo de caso de uso debe ser un medio de comunicación entre el cliente, los usuarios y los desarrolladores del sistema. Este modelo consta de casos de uso y actores. Cada caso de uso del modelo se describe detalladamente, mostrando paso a paso el modo en que el sistema interactúa con los actores y lo que el sistema hace en el caso de uso. Los casos de uso funcionan como hebra unificadora en todo el ciclo vital del software; el mismo modelo de caso de uso se utiliza en el análisis, diseño, implementación y prueba del sistema. **(27)**

2.6.1 Definición de actores

Actor	Descripción
Usuario	Es la persona o sistema que utiliza, según el nivel de acceso que tenga, las funcionalidades.

Tabla 2: Actor del sistema

2.6.2 Diagramas de casos de uso

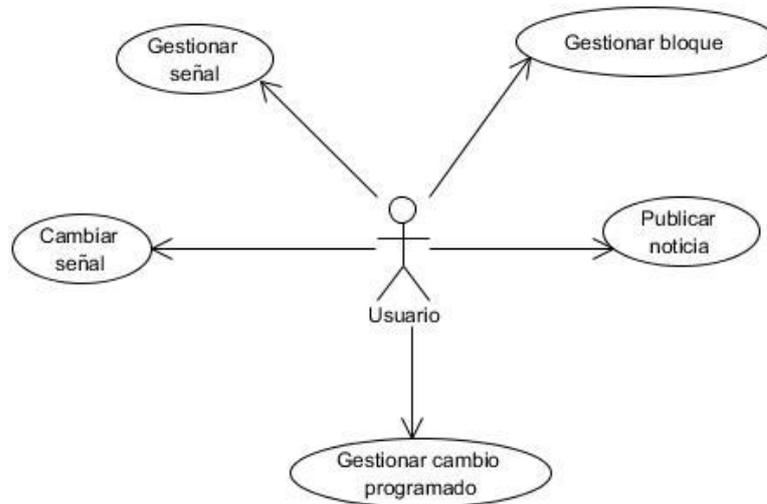


Figura 2: Diagrama de CU del sistema

2.6.3 Descripción textual de los casos de uso

Caso de Uso del Negocio: Gestionar señal

Caso de Uso: CU – 1	Gestionar señal
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el actor accede a la funcionalidad de Gestionar señal, dándole la posibilidad de insertar, listar, modificar, eliminar, filtrar y mostrar logo de la señal.
Precondiciones:	El usuario se autenticó en el sistema y tiene los permisos para Gestionar señal.
Referencias	RF 1, RF 2, RF 3, RF 4, RF 5, RF 6
CU asociados:	
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Listar señales”	
Acción del Actor	Respuesta del Sistema
1. El actor accede a la opción Señales del menú principal.	1. Muestra un listado con todas las señales, especificando: el identificador de la señal, nombre, dirección streaming, el logo y si

Capítulo 2: Características del sistema.

	está activa. Mostrará además la posibilidad de Insertar, Modificar, Eliminar, Filtrar y Mostrar logo de la señal.
2. Selecciona la opción a realizar.	3. Si elige: <ul style="list-style-type: none"> ➤ Insertar. Ver sección Insertar señal. ➤ Modificar. Ver sección Modificar señal. ➤ Eliminar. Ver sección Eliminar señal. ➤ Filtrar. Ver sección Filtrar señal. ➤ Mostrar logo. Ver sección Mostrar logo.
Sección “Insertar señal”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un formulario con un campo para el nombre de la señal, otro para la dirección streaming y otro para el logo.
2. Llena los campos y presiona el botón Guardar.	3. El sistema verifica que se hayan introducido todos los datos y que sean correctos. Si son correctos guarda la señal insertada.
Sección “Modificar señal”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un formulario donde aparecen los datos de la señal que fue seleccionada para modificar, dando la posibilidad de cambiar los contenidos de ella.
2. Realiza las modificaciones deseadas y presiona el botón Guardar.	3. Muestra un mensaje de confirmación para las modificaciones realizadas, dando la posibilidad de Aceptar o Cancelar la operación.
4. Confirma la realización de los cambios.	5. El sistema verifica que todos los datos introducidos sean correctos, si son correctos guarda las modificaciones realizadas.
Sección “Eliminar señal”	

Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje de confirmación para eliminar la señal, dando la posibilidad de Aceptar o Cancelar la operación.
2. Confirma la eliminación de la señal.	3. Si la señal no se encuentra activa la elimina.
Sección “Filtrar señal”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un formulario con la posibilidad de seleccionar el nombre de la señal y si se encuentra activa o no para filtrar el listado.
2. Escoge la opción por la que desea buscar y selecciona aceptar.	3. Deja en el listado solo las que cumplan con los criterios de búsqueda.
4. Selecciona el botón Restablecer.	5. El sistema muestra el listado de todas las señales.
Sección “Mostrar logo”	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una ventana con la imagen del logo de la señal y la opción de cerrar el diálogo.
2. El usuario selecciona Cerrar.	3. El sistema cierra la ventana.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Sección Insertar señal acción 3. Llena los campos y presiona el botón Guardar.	4. En caso de que no haya insertado todos los datos o no sean correctos, muestra el formulario para que el usuario vuelva a llenar los campos.
Sección Modificar señal acción 4 cancela las modificaciones.	5. No guarda las modificaciones de la señal.
Sección Modificar señal acción 4 confirma las modificaciones.	5. Si los datos no son correctos no modifica la señal.
Sección Eliminar cambio, acción 2. Cancela la	3. El sistema no elimina la señal.

eliminación de la señal.	
Sección Eliminar cambio, acción 2. Confirma la eliminación de la señal.	3. Si la señal se encuentra activa el sistema no la elimina.
Sección Buscar señal acción 2. Introduce los datos de la opción escogida. Presiona el botón Buscar.	3. En caso de no existir señales que cumplan con el criterio de búsqueda, muestra un listado en blanco.
Sección Buscar señal acción 4. Selecciona el botón Restablecer.	5. El sistema muestra el listado de todas las señales.

Prototipos de Interfaz de Usuario

Listado de señales

Figura 3: Prototipo de listar señales

Figura 4: Prototipo de adicionar señal



Figura 5: Prototipo de editar señal



Figura 6: Prototipo de filtrar señal



Figura 7: Prototipo de eliminar señal



Figura 8: Prototipo de mostrar logo de la señal

Pos condiciones

Tabla 3: Descripción del CU Gestionar señal

Caso de Uso del Negocio: Cambiar señal.

Caso de Uso: CU-2	Cambiar señal	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el actor accede a la funcionalidad de Cambiar la señal del canal, mostrándole la visualización de canales externos e internos y los cambios de señal asociados a cada una de ellas, además le da la posibilidad de cambiar la señal que se está transmitiendo en ese momento.	
Precondiciones:	El usuario se autenticó en el sistema y tiene los permisos para Cambiar la señal del canal.	
Referencias	RF-7, RF-8	
CU asociados:	CUS Gestionar cambios programados	
Prioridad	Crítico	
Flujo Normal de Eventos		
“Sección Mostrar señales a cambiar”		
Acción del Actor	Respuesta del Negocio	
1. El actor accede a la opción Cambio inmediato del menú principal.	2. El sistema muestra un formulario con las señales posibles a transmitir por el canal y un campo que podrá ser seleccionado, indicando que es la señal que se va a transmitir. Inicialmente estará seleccionado la que está saliendo al aire en ese momento y además se estará visualizando. Se muestra además las opciones cambiar, ver señal y habilitar cambios programados. Si elige habilitar cambios programados va al CU Gestionar cambios programados Sección Habilitar cambios programados.	
“Sección Cambiar señal”		
Acción del Actor	Respuesta del Negocio	
1. El actor selecciona la señal a la que desea	2. El sistema muestra un mensaje de si	

cambiar y aprieta el botón de aceptar.	desea deshabilitar los cambios programados.
3. El actor confirma que se deshabiliten o no los cambios programados.	4. El sistema deshabilita o no los cambios programados y cambia la señal que se va a transmitir, además envía un mensaje al Subsistema de Transmisión informándole el cambio de señal.

“Sección Ver señal seleccionada”

Acción del Actor	Respuesta del Negocio
1. El usuario selecciona la señal que desea visualizar dando clic en el botón Ver.	2. El sistema visualiza la señal seleccionada.

Prototipos de Interfaz de Usuario



Figura 9: Prototipo de cambiar señal

¿Desea deshabilitar los cambios programados?

Si No

Figura 10: Prototipo de deshabilitar cambios programados

Pos condiciones	
------------------------	--

Tabla 4: Descripción del CU Cambiar señal del canal

Caso de Uso del Negocio: Gestionar bloque

Caso de Uso: CU-4	Gestionar bloque	
Actores:	Usuario	
Resumen:	El caso de uso inicia cuando el actor accede a la funcionalidad de Gestionar bloque, dándole la posibilidad de insertar, listar, modificar, eliminar y buscar bloque.	
Precondiciones:	El usuario se autenticó en el sistema y tiene los permisos para Gestionar bloque.	
Referencias	RF 14, RF-15, RF 16, RF 17, RF 18	
CU asociados:		
Prioridad	Crítico	
Flujo Normal de Eventos		
“Sección Listar bloques”		
Acción del Actor	Respuesta del Negocio	
1. El actor accede a la opción Bloques del menú principal.	1. Muestra un listado con todos los bloques, especificando: la fecha y hora de inicio, fecha y hora de fin, el nombre. Mostrará además la posibilidad de Insertar, Modificar, Eliminar y Filtrar bloque.	
2. Selecciona la opción a realizar.	3. Si elige: <ol style="list-style-type: none"> 1. Insertar. Ver sección Insertar bloque 2. Modificar. Ver sección Modificar bloque. 3. Eliminar. Ver sección Eliminar bloque. 4. Filtrar. Ver sección Filtrar bloque. 	
Sección “Insertar bloque”		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra un formulario con un campo para la hora de inicio del bloque, otro para la hora de fin y otro para el nombre.	
2. Llena los campos y presiona el botón Guardar.	3. El sistema verifica que se hayan introducido todos los datos y que sean	

Capítulo 2: Características del sistema.

	correctos. Si son correctos guarda el bloque insertado.
Sección “Modificar bloque”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un formulario donde aparecen los datos del bloque que fue seleccionado para modificar, dando la posibilidad de cambiar los contenidos de él.
2. Realiza las modificaciones deseadas y presiona el botón Guardar.	3. Muestra un mensaje de confirmación para las modificaciones realizadas, dando la posibilidad de Aceptar o Cancelar la operación.
4. Confirma la realización de los cambios.	5. El sistema verifica que todos los datos introducidos sean correctos, si son correctos guarda las modificaciones realizadas. Se envía un mensaje al Subsistema de Transmisión informándole lo ocurrido.
Sección “Eliminar bloque”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje de confirmación para eliminar el bloque, dando la posibilidad de Aceptar o Cancelar la operación.
2. Confirma la eliminación del bloque.	3. El usuario verifica que el bloque no se encuentre activo y de ser así elimina el bloque. Se envía un mensaje al Subsistema de Transmisión informándole lo ocurrido.
Sección “Filtrar bloque”	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características del sistema.

	1. Muestra un formulario con la posibilidad de seleccionar los datos del bloque para filtrar el listado y las opciones Filtrar y Cancelar.
2. Escoge la opción por la que desea buscar y selecciona Filtrar.	3. Deja en el listado solo los que cumplan con los criterios de búsqueda.
4. Selecciona el botón Restablecer.	5. El sistema muestra el listado de todos los bloques.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Sección Insertar bloque acción 3. Llena los campos y presiona el botón Guardar.	4. En caso de que no haya insertado todos los datos o no sean correctos, muestra mensaje de error y el formulario para que el usuario vuelva a llenar los campos.
Sección Modificar bloque acción 2 cancela las modificaciones.	3. No guarda las modificaciones del cambio de bloque.
Sección Modificar bloque acción 4	5. El sistema verifica que la hora de inicio y de fin no se encuentren en el rango de la de otros bloques. Si se encuentra en el rango, muestra un mensaje informándole que debe cambiar la hora de inicio o la de fin o ambas y muestra el formulario para que el usuario vuelva a llenar los campos.
Sección Eliminar bloque , acción 2. Cancela la eliminación del bloque.	3. El sistema no elimina el bloque.
Sección Eliminar bloque , acción 2. Confirma la eliminación del bloque.	3. Si el bloque se encuentra activo no debe permitir que se elimine.
Sección Filtrar cambio acción 2. Escoge la opción por la que desea buscar y selecciona Filtrar.	3. El sistema muestra mensaje de error informando que debe introducir datos para realizar la búsqueda.

Prototipos de Interfaz de Usuario

Listado de bloques

<input type="checkbox"/>	Id	Fecha de Inicio	Hora de Inicio	Fecha de Fin	Hora de Fin	Nombre	Activo	Opciones
<input type="checkbox"/>	92	2012-05-28	12:05:00	2012-05-29	18:05:00	Noche		
<input type="checkbox"/>	114	2012-05-26	18:05:00	2012-05-27	23:05:00	Tarde		
<input type="checkbox"/>	118	2012-05-24	08:05:00	2012-05-25	09:05:00	Mañana		
<input type="checkbox"/>	119	2012-05-30	07:05:00	2012-05-30	13:05:00	Tarde		
<input type="checkbox"/>	120	2012-05-30	15:05:00	2012-05-31	04:05:00	Mañana		
<input type="checkbox"/>	121	2012-05-28	02:05:00	2012-05-28	11:05:00	Día		

Restablecer
 Filtrar
 Eliminar varios

Eliminar
 Editar

Adicionar

Figura 11: Prototipo de listar bloques

adicionar bloque

Fecha de Inicio

Hora de Inicio

Fecha de Fin

Hora de Fin

Nombre

Guardar

Figura 12: Prototipo de adicionar bloque

editar bloque

Fecha de Inicio

Hora de Inicio

Fecha de Fin

Hora de Fin

Nombre

Guardar

Figura 13: Prototipo de editar bloque

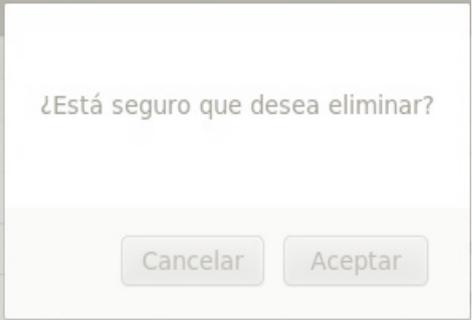


Filtrar Bloque ✕

Nombre:

Activo:

Figura 14: Prototipo de filtrar bloque



¿Está seguro que desea eliminar?

Figura 15: Prototipo de eliminar bloque

Pos condiciones	
------------------------	--

Tabla 5: Descripción del CU Gestionar bloque

Caso de Uso del Negocio: Publicar noticia

Caso de Uso: 5	Publicar noticia
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el actor accede a la funcionalidad de Publicar noticia, dándole la posibilidad de asignarle noticias listas para publicar, listar las noticias publicadas, ordenarlas ó eliminarlas.
Precondiciones:	El usuario se autenticó en el sistema y tiene los permisos para Publicar noticia.
Referencias	RF-19, RF-20, RF-21
CU asociados:	
Prioridad	Crítico
Flujo Normal de Eventos	

“Sección Listar bloques y noticias a publicar”	
Acción del Actor	Respuesta del Negocio
1. El actor accede a la opción Publicar noticia en el menú principal.	2. Muestra un listado con todos los bloques, especificando: la fecha y hora de inicio, fecha y hora de fin y otro con las noticias a publicar especificando el título y la duración. Mostrará además la posibilidad de Publicar noticia y Listar noticias publicadas.
“Sección Asignar noticias”	
Acción del Actor	Respuesta del Negocio
1. El usuario selecciona de los listados la(s) noticia(s) que desea publicar y bloque(s) donde desea hacerlo. Luego aprieta el botón Publicar.	2. Si se seleccionaron noticia(s) y bloque(s) el sistema guarda la publicación de las noticias.
“Sección Listar noticias publicadas”	
Acción del Actor	Respuesta del Negocio
1. El actor selecciona un bloque y da clic en el botón Ver noticias del bloque.	2. El sistema muestra un listado con las noticias publicadas en el bloque seleccionado. Mostrará además la posibilidad de Ordenar y Eliminar las noticias publicadas.
2. Selecciona la opción a realizar.	3. Si elige: <ul style="list-style-type: none"> ➤ Ordenar. Ver sección Ordenar noticias publicadas. ➤ Eliminar. Ver sección Eliminar noticias publicadas.
“Sección Ordenar noticia publicada”	
Acción del Actor	Respuesta del Negocio
	1. El sistema muestra por cada noticia publicada la opción de cambiarle el orden por la de arriba (Subir) o por la de abajo

Capítulo 2: Características del sistema.

	(Bajar).
2. Selecciona la opción a realizar.	3. El sistema cambia el orden de la noticia publicada según la opción seleccionada y muestra el listado con las noticias ordenadas.
“Sección Eliminar noticia publicada”	
Acción del Actor	Respuesta del Negocio
	1. Muestra un mensaje de confirmación para eliminar la noticia publicada, dando la posibilidad de Aceptar o Cancelar la operación.
2. Confirma la eliminación de la noticia publicada.	3. Elimina la noticia publicada. Si la noticia eliminada se encuentra dentro del bloque que está activo debe enviársele un mensaje al Subsistema de Transmisión informándole lo ocurrido.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Sección Asignar noticia acción 3. Selecciona la(s) noticia(s) que desea publicar y bloque(s) donde desea hacerlo. Luego aprieta el botón Publicar.	4. En caso de que no haya seleccionado noticia(s) y bloque(s) no se guarda la publicación de la noticia.
Sección Eliminar noticia publicada , acción 2. Cancela la eliminación de la noticia publicada.	3. El sistema no elimina la noticia publicada.
Sección Ordenar noticia publicada acción 2. Selecciona la opción a realizar.	3. En caso de que el orden de la noticia publicada sea la primera y el usuario seleccionó subir o sea la última y el usuario seleccionó bajar el sistema muestra un mensaje de error.
Prototipos de Interfaz de Usuario	

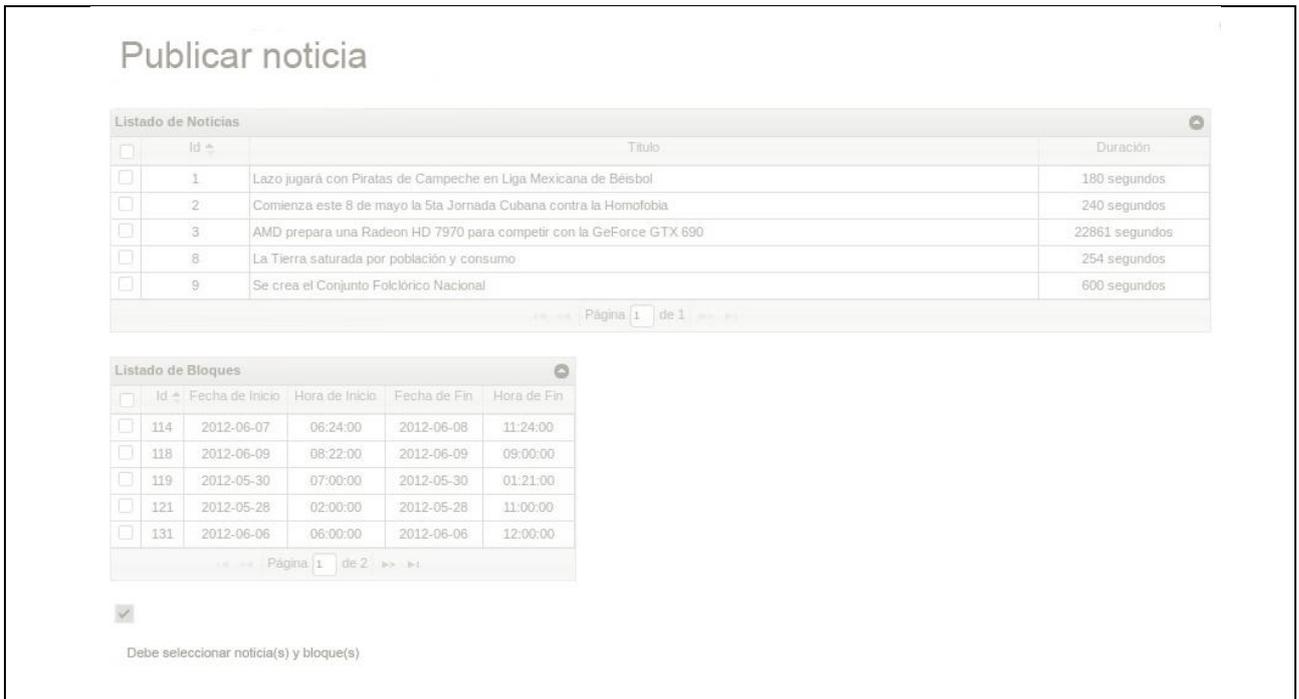


Figura 16: Prototipo de publicar noticia



Figura 17: Prototipo de listar noticias publicadas en bloque



Figura 18: Prototipo de eliminar noticia publicada.

Pos condiciones

Tabla 6: Descripción del CU Publicar noticia

En el anexo 1 se encuentra la descripción textual del caso de uso Gestionar cambio programado.

2.7 Conclusiones del capítulo

Para la representación de los conceptos significativos en el problema se realizó el modelo del dominio debido a la no existencia de una definición clara de los procesos de negocio permitiendo comprender mejor el funcionamiento del sistema. Los requerimientos funcionales y no funcionales guiarán el proceso de desarrollo del sistema pues se definió lo que debe realizar el mismo y las características que debe tener. Utilizando el plugin de VLC Media Player para reproducir las señales se garantizará visualizar su monitorización. Se elaboró el diagrama de casos de uso del sistema y se hicieron las especificaciones de los mismos para garantizar una mejor comprensión de los requisitos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.

3.1 Introducción

En este capítulo se describe la arquitectura del sistema, haciendo énfasis en los patrones de arquitectura y diseño usados por el *framework* Symfony para de esta forma alcanzar un mejor entendimiento del diseño que se realice. Además, se desarrolla el análisis y diseño del sistema, donde se construyen los diagramas de clases del análisis y del diseño.

3.2 Modelo de análisis

El modelo de análisis contiene las clases de análisis y cualquier producto de trabajo asociados. El modelo de análisis puede ser un producto de trabajo temporal, como cuando evoluciona hacia un modelo de diseño, o puede permanecer durante algunos o todos los proyectos, y quizás más, sirviendo como visión general conceptual del sistema. (27)

3.2.1 Diagrama de clases del análisis



Figura 19: Diagrama de clases del análisis CU Cambiar señal del canal

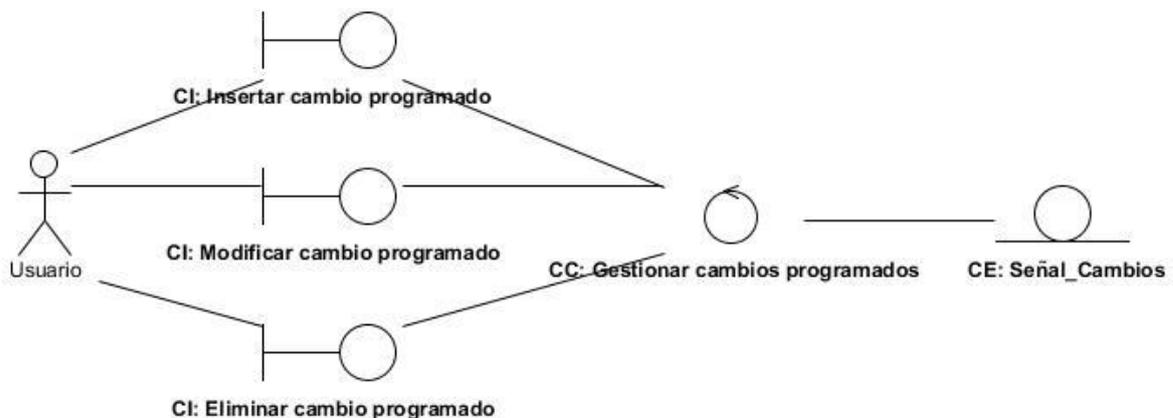


Figura 20: Diagrama de clases del análisis CU Programar cambio de señal

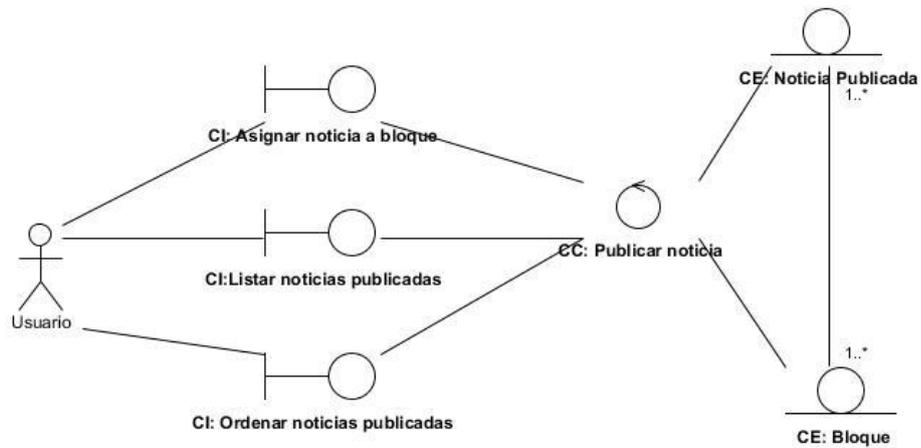


Figura 21: Diagrama de clases del análisis CU Publicar noticia

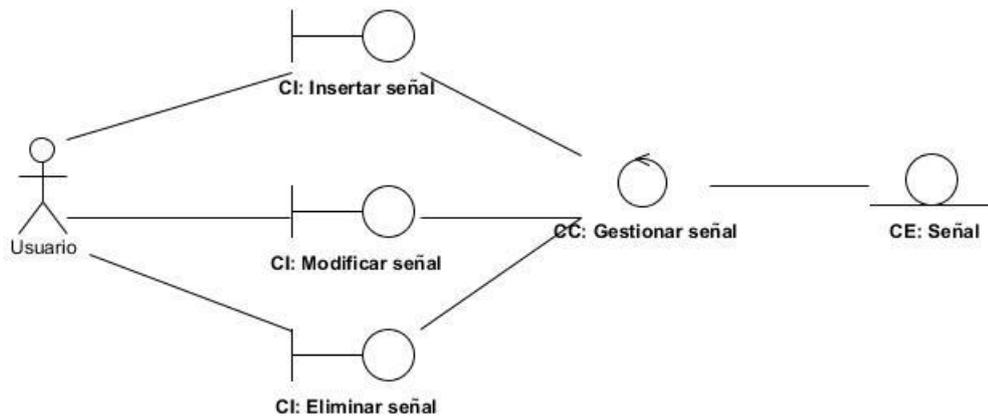


Figura 22: Diagrama de clases del análisis CU Gestionar señal

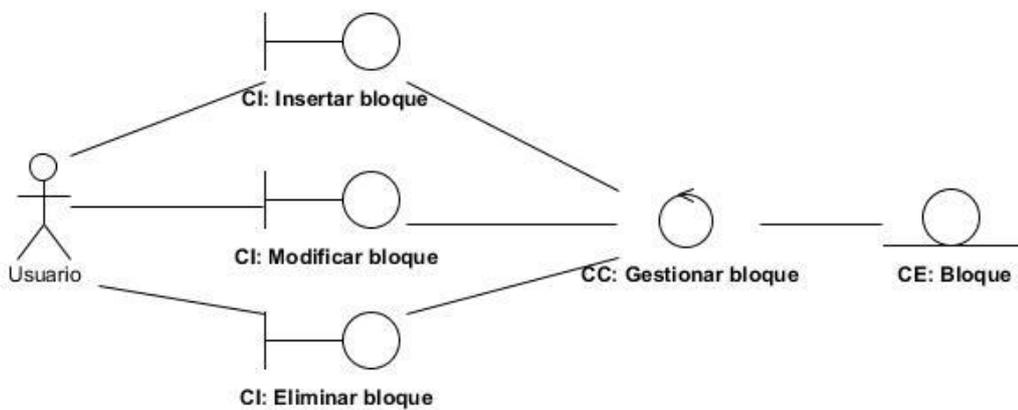


Figura 23: Diagrama de clases del análisis CU Gestionar bloque

3.3 Descripción de la arquitectura

La arquitectura del software desempeña un papel fundamental durante el desarrollo de un sistema ya que permite representar su estructura, además permite organizar sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución.

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura lógica. Se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiarse en el desarrollo de software dentro de un sistema informático. Dichos patrones serán capaces de aportar elementos idóneos para tomar buenas decisiones. **(28)**

3.3.1 Patrón arquitectónico

Los patrones arquitectónicos expresan un esquema organizativo estructural fundamental para sistemas de software y definen las reglas generales de organización, las restricciones en la forma y la estructura de un grupo numeroso de sistemas de software. La selección de un patrón arquitectónico es una decisión fundamental de diseño en el desarrollo de un sistema de software. **(28)**

Symfony emplea el tradicional patrón arquitectónico Modelo Vista Controlador para separar las distintas partes que forman una aplicación web. Este patrón es muy usado en aplicaciones web.

Con el Modelo Vista Controlador la aplicación se puede desarrollar rápidamente. Separar las funciones de la aplicación en modelos, vistas y controladores hace que la aplicación sea muy ligera. Esta separación también permite hacer cambios en una parte de la aplicación sin que las demás se vean afectadas.

El patrón Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios.
- El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista.
- El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos.

Implementación del Modelo Vista Controlador en Symfony:

- La capa **Modelo** define la lógica de negocio (la base de datos pertenece a esta capa). Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio `SenalBundle/Entity/`.

- La **Vista** es con lo que el usuario interactúa (un motor de plantillas es parte de esta capa). En Symfony, la vista es principalmente la capa de plantillas PHP. Estas son guardadas en varios directorios *views/*.
- El **Controlador** es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Cuando se instala Symfony todas las solicitudes son gestionadas por un controlador frontal (*app.php* y *app_dev.php*). Estos controladores frontales delegan la verdadera labor a las acciones. Estas acciones son, lógicamente, agrupadas en paquetes (*bundles*).

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno Modelo Vista Controlador, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

3.3.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. **(29)**

Dentro de los patrones de producto de software se encuentran los de análisis, arquitectura, diseño y lenguaje de programación. Para el desarrollo de la solución se aplicaron diferentes patrones de diseño, fundamentalmente los patrones de diseño: General Responsibility Assignment Software Patterns (GRASP) y Gang of Four (GoF), con el objetivo de facilitar el mantenimiento del software. A continuación se abordarán los patrones de diseños utilizados en el sistema desarrollado para asegurar una solución mucho más confiable y contribuir a la realización de un producto reutilizable y escalable.

Patrones GRASP

Los patrones de asignación de responsabilidades GRASP, permiten asignar correctamente las responsabilidades a cada una de las clases que intervienen en el modelo; de este grupo de patrones fueron tomados en cuenta para una correcta asignación de las relaciones entre las clases y un correcto diseño de las mismas, los siguientes: **(30)**

Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la realidad que representa, por lo que ofrece una analogía con el mundo real. Con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.

El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener. **(30)**

Este patrón se pone de manifiesto con la utilización de Doctrine2, que es la librería externa que se utiliza para realizar la 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.

Creador: Se refiere a asignar responsabilidades a las clases de crear instancias de otras conociendo que las primeras son las que contienen la información para ello. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. **(30)**

Este patrón se pone de manifiesto en la clase DefaultController donde se definen acciones y se ejecutan cada una de ellas. En esta clase se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase DefaultController es "creador" de dichas entidades.

Alta cohesión: Este patrón se utiliza para asignar responsabilidades a las clases de manera que todos sus métodos tuvieran un comportamiento bien definido. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. El patrón Alta Cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. Con el uso de este patrón mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifican el mantenimiento y las mejoras en funcionalidad. **(30)**

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase DefaultController tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades. Está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Patrón Controlador: La primera categoría de controlador es un controlador de fachada que representa al "sistema" global. Es una clase que, para el diseñador representa de alguna manera al sistema entero. La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. La misma clase controlador debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso. **(30)**

Todas las peticiones Web son manejadas por un solo controlador frontal (app.php), 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. Este patrón se encuentra ejemplificado en los archivos DefaultController.php, InmediatoController.php, BloqueController.php, PublicarNoticiaController.php y CambiosProgramadosController.php.

Patrones GoF

Decorator (Envoltorio): Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia.

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.

3.4 Modelo de diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos. (27)

3.4.1 Diagrama de clases del diseño

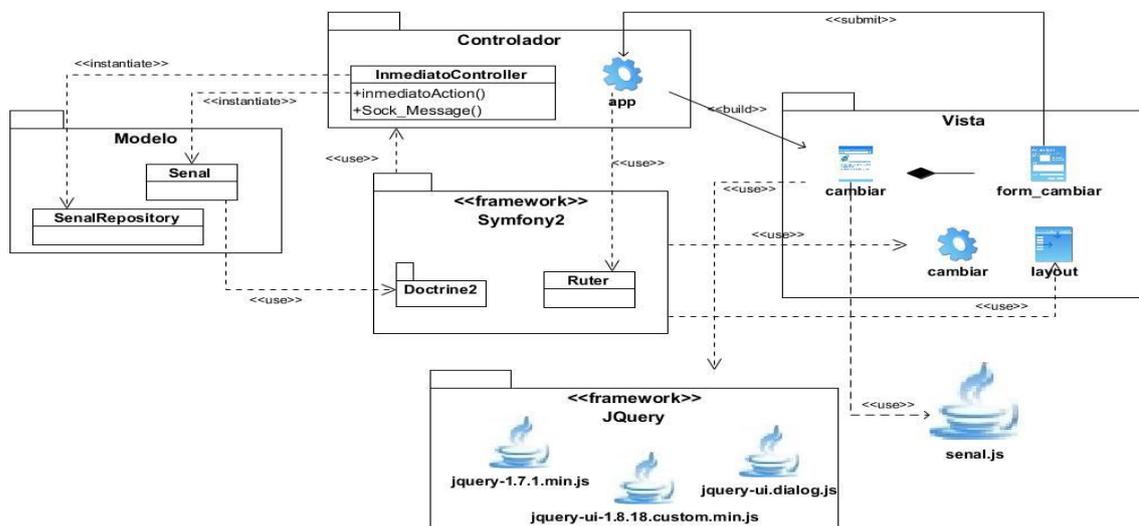


Figura 24: Diagrama de clases del diseño del CU Cambiar señal

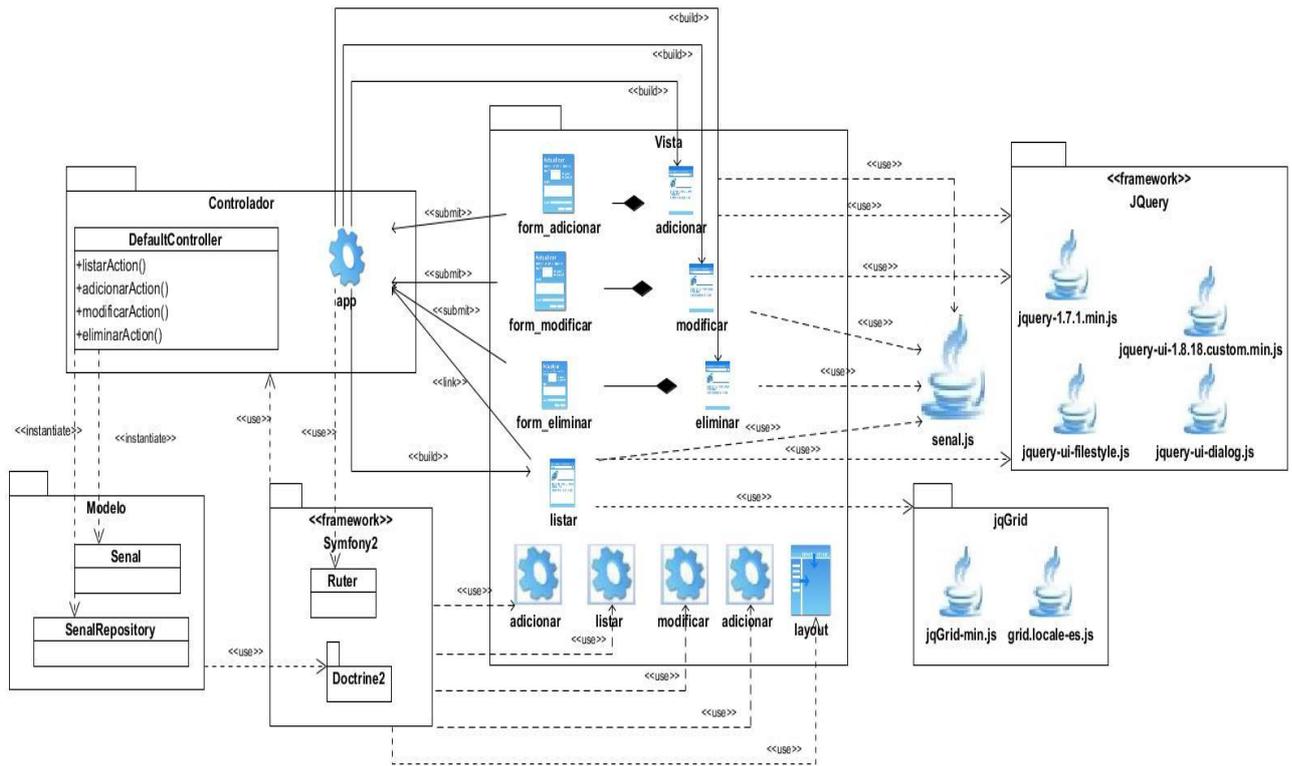


Figura 27: Diagrama de clases del diseño del CU Gestionar señal

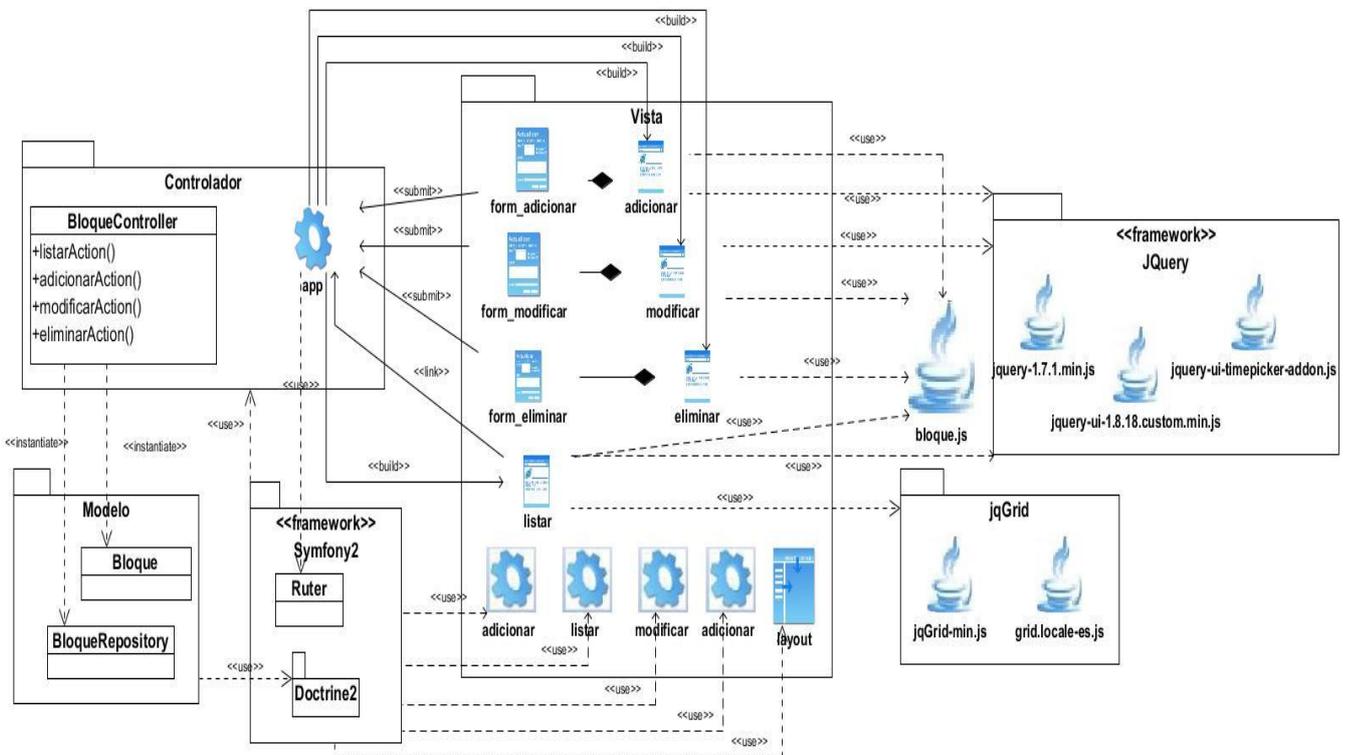


Figura 28: Diagrama de clases del diseño del CU Gestionar bloque

3.5 Diagrama de clases persistentes

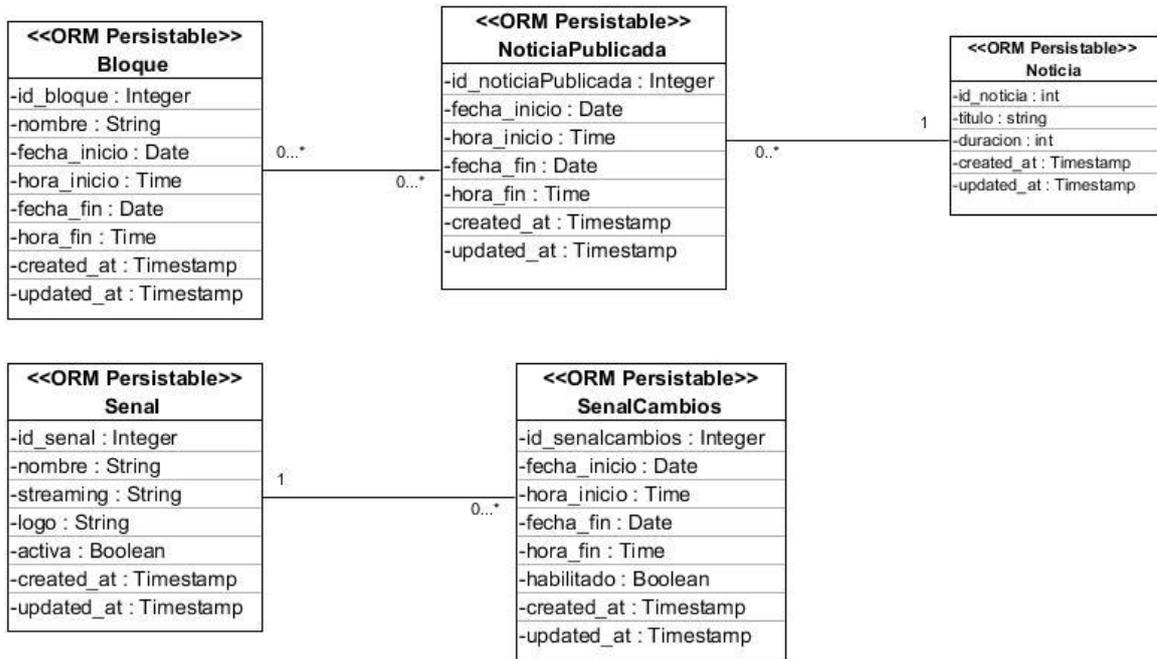


Figura 29: Diagrama de clases persistentes

3.6 Modelo de datos

Este modelo representa a la realidad a través de un esquema gráfico empleando los terminologías de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo. Describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. (27)

3.6.1 Diagrama Entidad- Relación

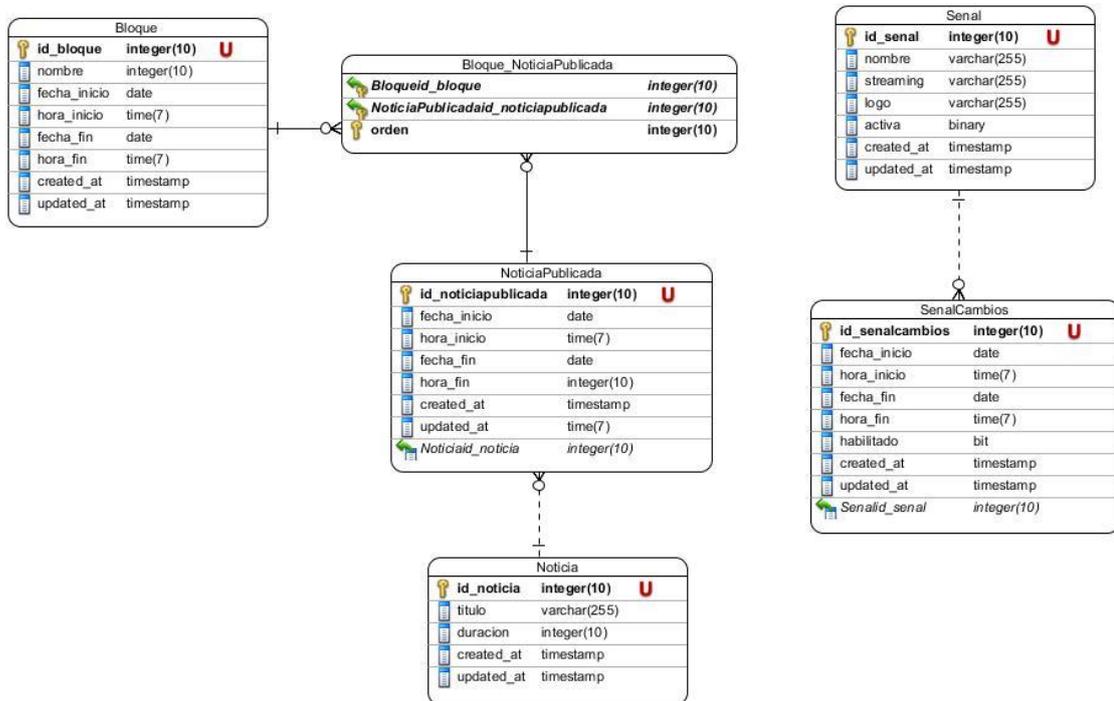


Figura 30: Diagrama Entidad-Relación

3.7 Modelo de despliegue

Este modelo tiene como objetivo capturar la configuración de los elementos de proceso y las conexiones entre elementos de proceso, en el sistema. (27)

3.7.1 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo.

Nodos: Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.

Dispositivos: Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

Conectores: Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

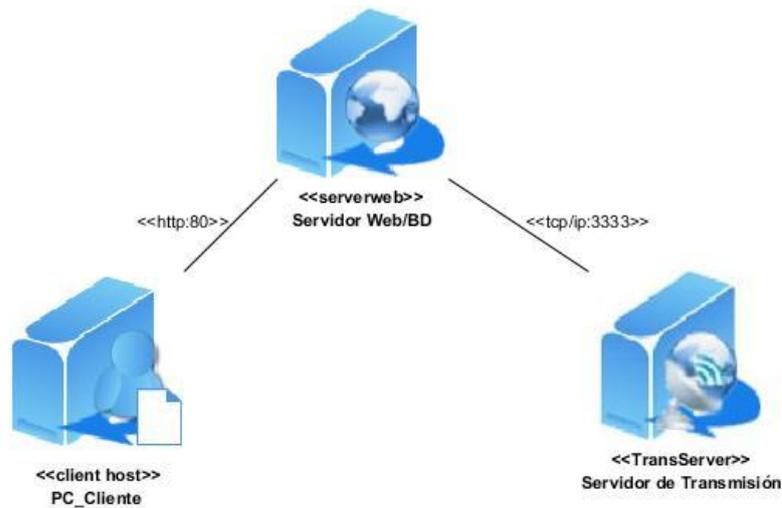


Figura 31: Diagrama de despliegue

Descripción de los nodos:

Servidor Web/BD: Es el nodo que albergará el servidor web y la base de datos. Es el encargado de responder a las peticiones de las PCs Clientes. El servidor tendrá instalado como sistema operativo una distribución de Linux específicamente Ubuntu 11.10, el cual sólo constará con las funcionalidades básicas para que la Plataforma brinde sus servicios, estos software son el gestor de base de datos PostgreSQL, Apache y un servidor FTP.

Servidor de Transmisión: Es el encargado de transmitir las noticias. El servidor tendrá instalado como sistema operativo una distribución de Linux específicamente Ubuntu 11.10 y VLC para transmitir los videos por streaming.

PC Clientes: Estas computadoras serán las que interactuarán con la plataforma, las cuales pueden tener cualquier sistema operativo siempre y cuando su navegador sea Firefox con plugin de vlc instalado.

Descripción de los conectores:

TCP/IP: Se establecerá una conexión entre los servidores a 1 Gigabit, para así lograr una máxima eficiencia en la transmisión de la información.

HTTP: Se utilizará HTTP ya que el sistema está implementado en tecnología Web.

3.8 Conclusiones del capítulo

Los diagramas de clases de análisis constituyeron el primer paso para la construcción de los diagramas de clases del diseño y sirvieron para reflejar una visión general de las clases a tener en cuenta para implementar las nuevas funcionalidades. El análisis de la arquitectura y diseño de

Symfony permitió obtener una mejor representación de los diagramas de clases del diseño. Las clases persistentes dieron paso a la construcción del modelo de datos para describir la estructura lógica y física de la información persistente gestionada por el sistema. Con la interpretación del diagrama entidad relación surgió la necesidad de realizar cambios en la base de datos de PRIMICIA para desarrollar satisfactoriamente las funcionalidades identificadas.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE GESTIÓN Y PUBLICACIÓN DE SEÑALES.

4.1 Introducción

En la implementación se parte de los resultados obtenidos en el diseño. Se desarrolla el diagrama de componentes, donde se describen los elementos físicos del sistema y sus relaciones. Se aprecia además la realización de pruebas encaminadas a encontrar errores en el sistema.

4.2 Diagramas de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Representa como un sistema de software es dividido en componentes y muestra dependencias entre estos componentes. Prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Se puede decir que un componente es la materialización de una o más clases.

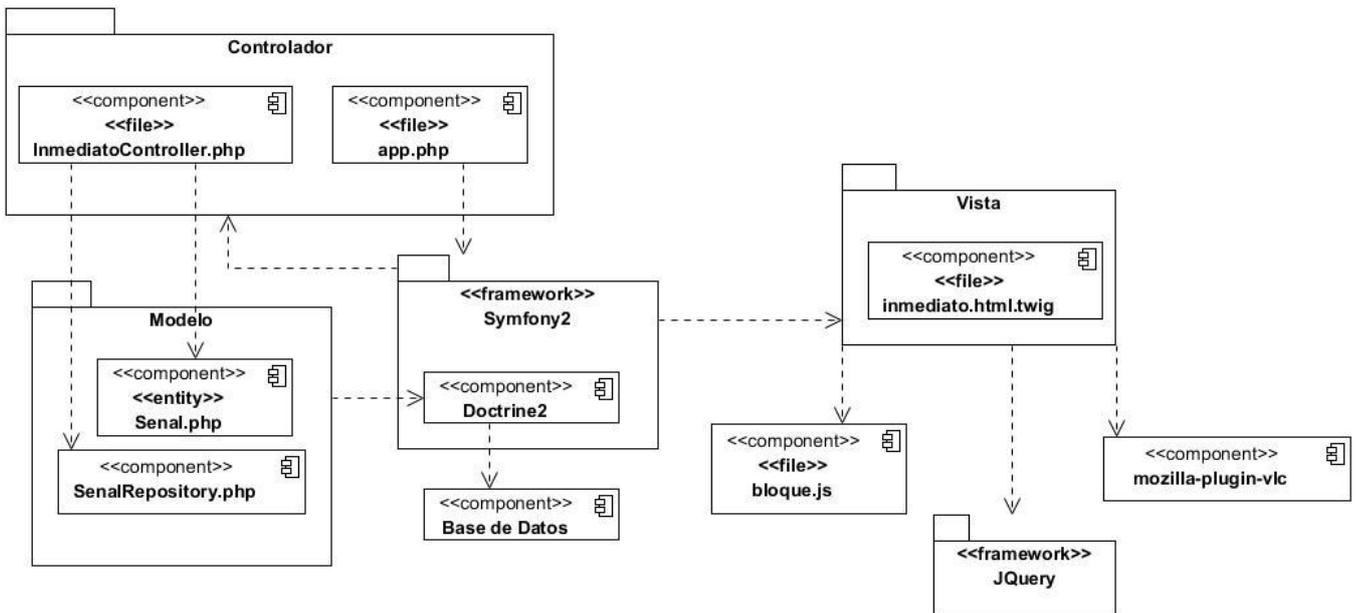


Figura 32: Diagrama de componentes del CU Cambiar señal

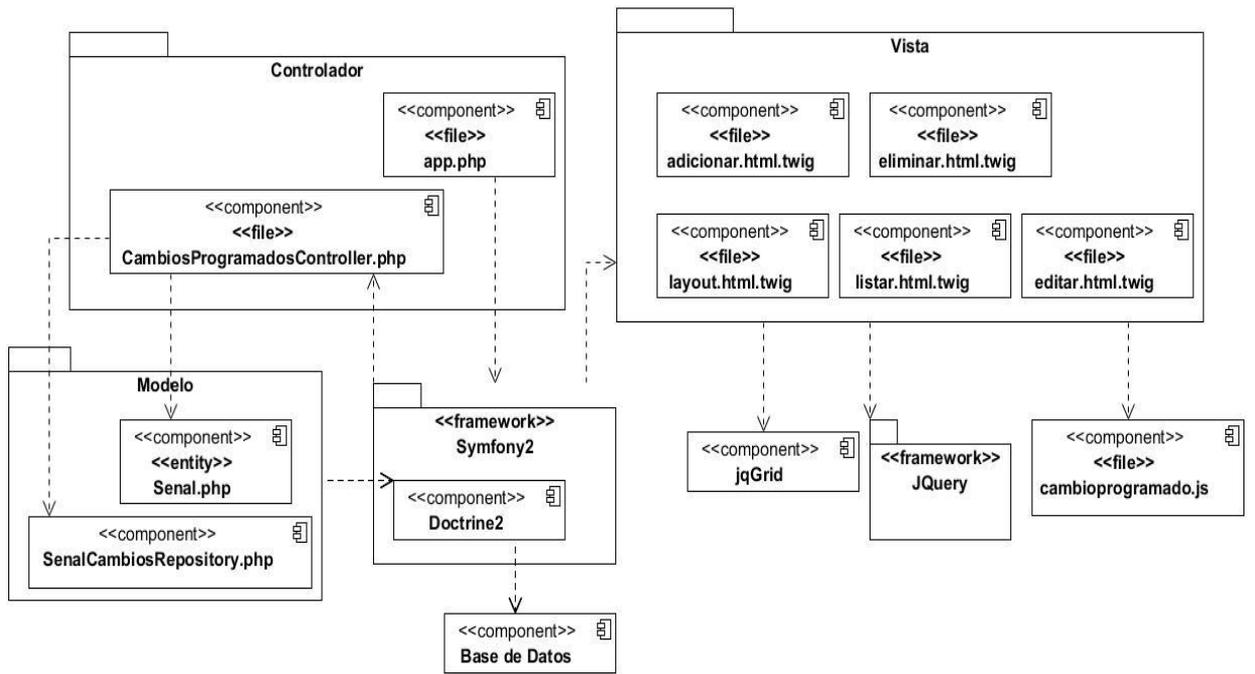


Figura 33: Diagrama de componentes del CU Gestionar cambios programados

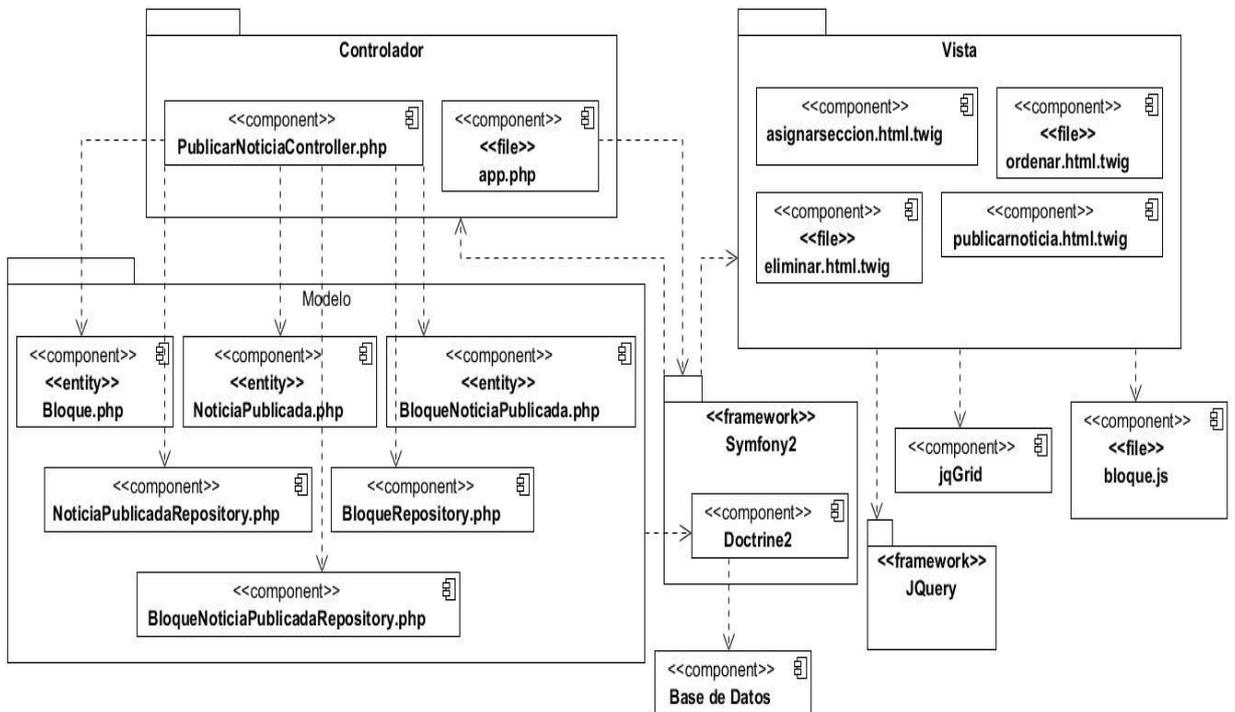


Figura 34: Diagrama de componentes del CU Publicar noticia

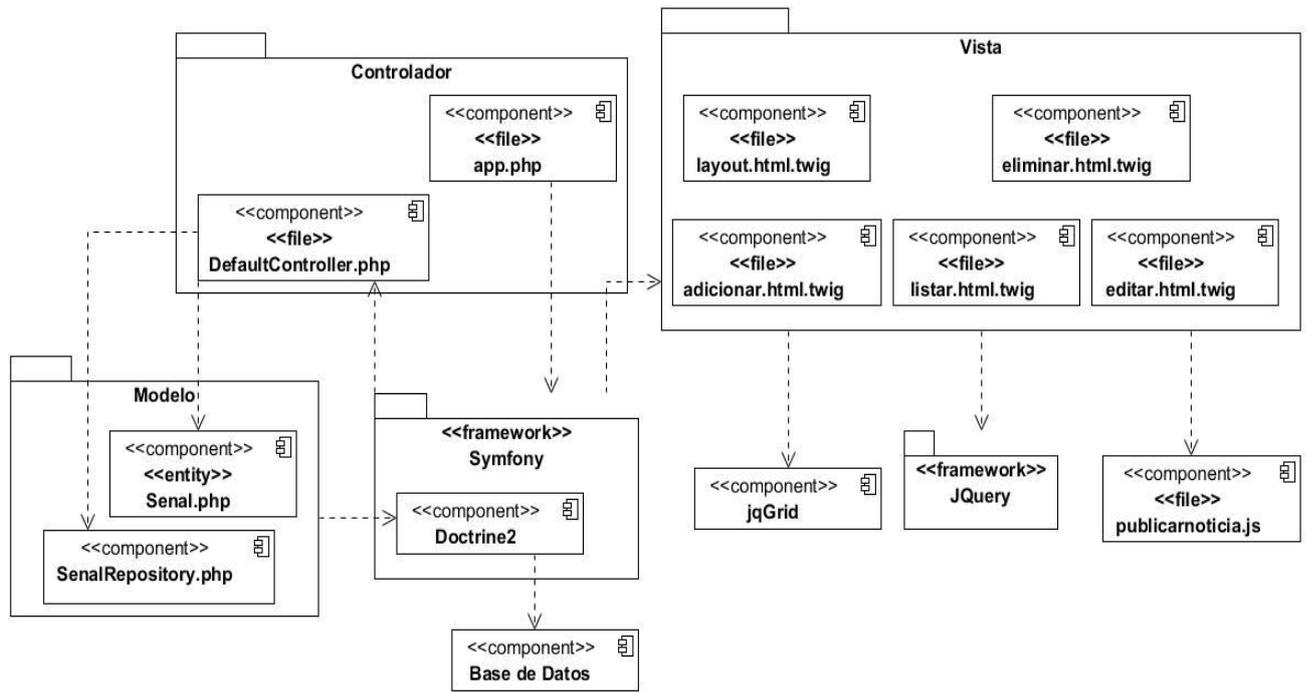


Figura 35: Diagrama de componentes del CU Gestionar señal

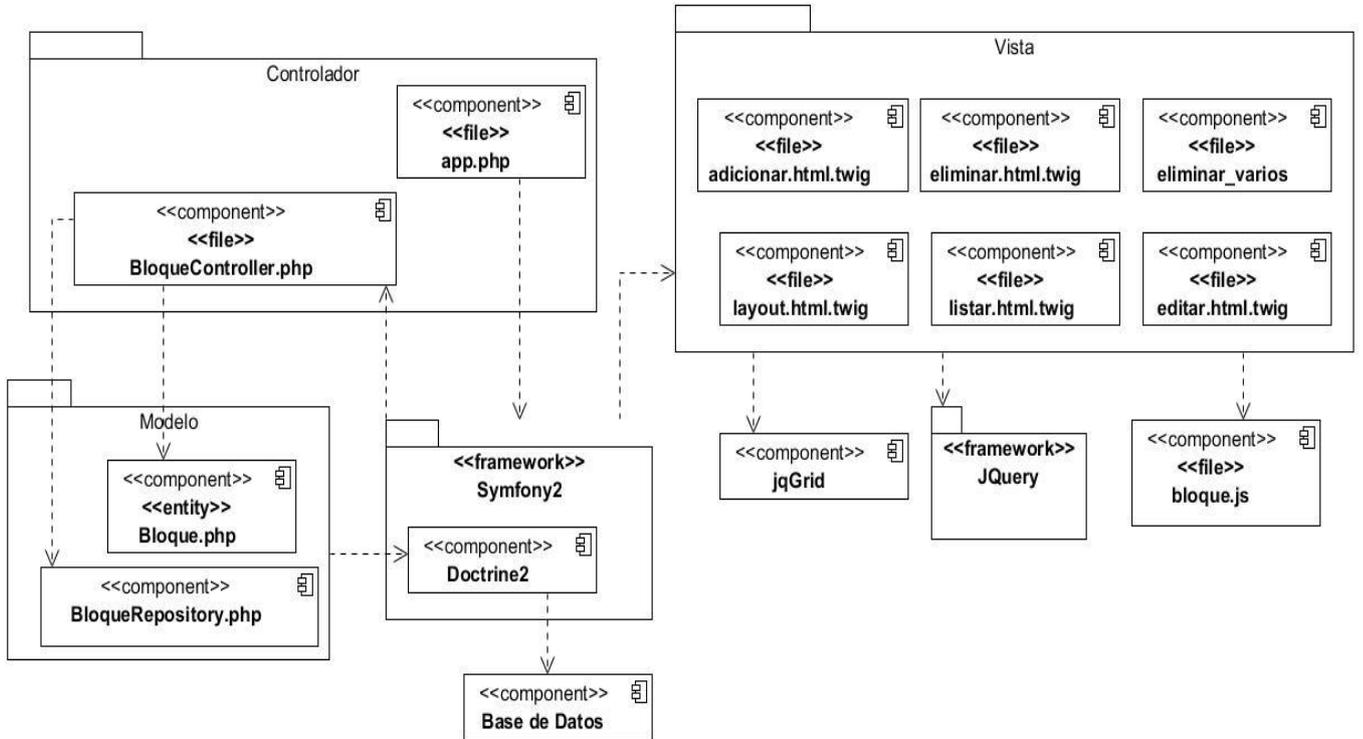


Figura 36: Diagrama de componentes del CU Gestionar bloque

4.3 Pruebas del sistema

Cuando aparecieron los primeros grandes sistemas informáticos se incluyó a nivel metódico e imprescindible un hasta entonces nuevo proceso en la confección de los mismos: el proceso de prueba.

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. **(31)**

El método de prueba realizado al sistema son las pruebas de caja negra, las mismas se centran en los requisitos funcionales del software y se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa.

Dentro del método de caja negra la técnica de la partición equivalente es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software.

La técnica de partición equivalente divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba a desarrollar. **(32)**

4.3.1 Casos de prueba

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de usos del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema.

En la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

Para el desarrollo de las pruebas al componente se desarrollaron cinco casos de pruebas, uno por cada caso de uso del sistema. Los casos de pruebas desarrollados se muestran a continuación:

Caso de prueba: Gestionar señal

Descripción del CU Gestionar señal: El caso de uso inicia cuando el usuario accede a la opción de Señal, mostrándose el listado de las señales almacenadas en la base de datos. Termina al crear, modificar, eliminar, filtrar, mostrar logo de la señal o con un mensaje de proceso fallido.

Secciones a probar en el caso de uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Listar señal.	EC 1.1: Mostrar listado de señales.	El usuario selecciona la opción Señal del menú superior y el sistema muestra un listado con las señales almacenadas en la base de datos. Muestra además la posibilidad de insertar, modificar, eliminar, filtrar o mostrar logo de señal.
SC2: Adicionar señal.	EC 2.1: Adicionar señal satisfactoriamente.	El usuario selecciona la opción adicionar. El sistema muestra un formulario para entrar los datos de la señal. El usuario entra los datos correspondientes y presiona el botón guardar. El sistema guarda los datos de la señal.
	EC 2.2: Adicionar señal con campos vacíos.	El usuario selecciona la opción adicionar. El sistema muestra un formulario para entrar los datos de la señal a adicionar. El usuario deja algún campo de los obligatorios en blanco y presiona el botón guardar. El sistema muestra un mensaje indicando que los campos no deben estar en blanco.
	EC 2.3 Adicionar señal con datos no válidos.	El usuario selecciona la opción adicionar. El sistema muestra un formulario para entrar los datos de la señal a adicionar. El usuario introduce datos no válidos y presiona el botón guardar. El sistema muestra un mensaje informando que estos no son correctos.
SC 3: Editar señal.	EC 3.1: Editar señal satisfactoriamente.	El usuario selecciona la opción editar. El sistema muestra los datos de la señal seleccionada por el usuario. El usuario edita los datos y presiona el botón guardar. Los datos se editan correctamente y se muestra un mensaje al usuario indicando que la modificación fue satisfactoria.

	EC 3.2: Editar señal con campos vacíos.	El usuario selecciona la opción editar. El sistema muestra los datos de la señal seleccionada por el usuario, el usuario edita los datos dejando campos en blanco y presiona el botón guardar. El sistema muestra un mensaje indicando que los campos no deben estar en blanco.
	EC 3.3: Editar señal con campos no válidos.	El usuario selecciona la opción editar. El sistema muestra los datos de la señal seleccionada por el usuario, el usuario los cambia por datos no válidos y presiona el botón guardar. El sistema muestra un mensaje indicando que los datos no son correctos.
SC 4: Eliminar señal.	EC 4.1: Eliminar señal satisfactoriamente	El usuario selecciona la señal a eliminar y el sistema lo elimina satisfactoriamente.
	EC 4.2: Eliminar señal falla.	El usuario selecciona la señal a eliminar, el sistema muestra un mensaje indicando que ocurrió un error durante el proceso de eliminación.
SC5: Filtrar señales.	EC 5.1: Filtrar señales introduciendo criterios de búsqueda.	El usuario selecciona la opción filtrar. El sistema muestra un formulario para entrar los datos para filtrar, el usuario entra los datos correspondientes y presiona el botón filtrar. El sistema muestra un listado con las señales que cumplen los criterios de búsqueda.
	EC 5.2: Filtrar señales sin introducir criterios de búsqueda.	El usuario selecciona la opción filtrar y presiona el botón filtrar. El sistema muestra un mensaje indicando que los criterios de búsqueda están vacíos.

SC6: Mostrar logo de señal.	EC 6.1: Visualizar logo de la señal.	El sistema muestra en una ventana emergente el logo de la señal seleccionada.
-----------------------------	--------------------------------------	-------------------------------------------------------------------------------

Tabla 7: Secciones a probar en el CU Gestionar señal.

Descripción de variables.

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Cualquier combinación de letras, números, guiones bajo y espacios.
2	Enlace	Campo de texto	No	Cualquier combinación de letras, números y guiones bajo y que comience con mms, rstp o http.
3	Logo	Campo tipo <i>file</i>	No	Una imagen seleccionada que sea de tipo jpg, gif o png y que su tamaño sea menor o igual a 1024k.
4	Id	Campo oculto	No	Número entero que debe estar en correspondencia con el identificador de una señal existente en la base de datos.
5	Activa	Campo de selección	No	Puede tomar los valores Sí (verdadero) y No (falso).

Tabla 8: Descripción de variables del CU Gestionar señal.

Secciones a revisar.

SC 1: Listar señales

Id del escenario	Respuesta del Sistema	Resultado de la Prueba
EC 1: Mostrar listado de señales	El sistema muestra un listado de las señales almacenadas en base de datos. Muestra además la posibilidad de insertar, modificar, eliminar, filtrar y mostrar logo de la señal.	Satisfactoria

Tabla 9: Sección a revisar: Listar señales

SC 2: Adicionar señal

Id del escenario	Nombre	Enlace	Logo	Respuesta del Sistema	Resultado de la Prueba
EC 2.1 Adicionar señal satisfactoriamente.	V	V	V	Añade la señal y se muestra un mensaje informando que se añadió satisfactoriamente.	Satisfactoria
EC 2.2 Adicionar señal con campos vacíos.	V	I	I	Muestra un mensaje indicando que los campos no deben estar en blanco.	Satisfactoria
	V	V	I		
	V	I	V		
	I	I	I		
EC 2.3: Adicionar señal con datos no válidos.	I	V	V	Muestra un mensaje indicando que la señal no se adicionó pues los datos no son correctos.	Satisfactoria
	V	I	V		
	V	V	I		
	I	I	I		

Tabla 10: Sección a revisar: Adicionar señal.

SC 3: Editar señal

Id del escenario	Nombre	Enlace	Logo	Id	Respuesta del Sistema	Resultado de la Prueba
EC 3.1 Editar señal satisfactoriamente.	V	V	V	V	Edita la señal correctamente y se muestra un mensaje informando que la operación fue satisfactoria.	Satisfactoria
EC 3.2 Editar datos de señal con campos vacíos.	I	V	V	V	Muestra un mensaje indicando que los campos no deben estar en blanco.	Satisfactoria
	V	I	V	V		
	V	V	I	V		
	V	V	V	I		
	I	I	I	I		
EC 3.3: Editar señal con datos no válidos.	I	V	V	V	Muestra un mensaje indicando que la señal no se editó pues los datos no son correctos.	Satisfactoria
	V	I	V	V		
	V	V	I	V		
	V	V	V	I		
	I	I	I	I		

Tabla 11: Sección a revisar: Editar señal.

SC 4: Eliminar señal

Id del escenario	Id de la señal	Respuesta del sistema	Resultado de la prueba
EC 4.1 Eliminar señal satisfactoriamente.	V	Elimina la señal correctamente y muestra un mensaje informando que la operación fue satisfactoria.	Satisfactoria
EC 4.2 Eliminar señal falla.	I	Muestra un mensaje indicando que la señal no se pudo eliminar de forma satisfactoria.	Satisfactoria

Tabla 12: Sección a revisar: Eliminar señal.

SC 5: Filtrar señal

Id del escenario	Nombre	Activa	Respuesta del Sistema	Resultado de la Prueba
EC 5.1: Filtrar señales introduciendo criterios de búsqueda.	V	V	El sistema muestra un listado con las señales que cumplen los criterios de búsqueda.	Satisfactoria
	V	I		
	I	V		
EC 5.2: Filtrar señales sin introducir criterios de búsqueda.	I	I	El sistema muestra un mensaje indicando que los criterios de búsqueda están vacíos.	Satisfactoria

Tabla 13: Sección a revisar: Filtrar señal.

SC6: Mostrar logo de señal.

Id del escenario	Id de la señal	Respuesta del Sistema	Resultado de la Prueba
EC 6.1: Visualizar logo de la señal.	V	El sistema muestra en una ventana emergente el logo de la señal seleccionada.	Satisfactoria

Tabla 14: Sección a revisar: Mostrar logo de señal.

En el anexo 2 se encuentran los restantes cuatro casos de prueba desarrollados.

Se pudo apreciar anteriormente el diseño de los casos de prueba, así como el resultado de la tercera iteración de la aplicación de la prueba de caja negra, el mismo se puede concluir debido a la respuesta del sistema como satisfactorio. En la primera iteración se detectaron 13 no conformidades, y en la segunda iteración se detectaron 6, estas fueron corregidas y ya para una tercera iteración no fueron detectadas.

4.4 Conclusiones del capítulo

Se obtuvo el diagrama de componentes para mostrar la estructura física que tienen los mismos y sus relaciones permitiendo visualizar la estructura general del sistema. También se realizaron las pruebas al sistema validando los requisitos necesarios para el correcto funcionamiento del mismo, dándole la calidad requerida al producto. Se elaboró el modelo de pruebas, conformado por cinco casos de prueba que se corresponden con los cinco casos de uso identificados anteriormente, reflejándose que las no conformidades más vistas fueron las de errores ortográficos.

CONCLUSIONES GENERALES

Una vez culminado el desarrollo de la investigación del presente trabajo de diploma, la implementación del sistema y validación de la aplicación a través de las pruebas, es posible arribar a las siguientes conclusiones:

- Se logró conocer el estado del objeto de estudio de la investigación mediante el empleo de los métodos de investigación teóricos y empíricos.
- Un momento clave de la investigación fue la búsqueda y descripción de aplicaciones similares a la solución propuesta. Lo que permitió tomar sus elementos positivos e implantarlos en el módulo.
- Se caracterizaron las herramientas y tecnologías de desarrollo establecidas en la arquitectura del producto, permitiendo reconocer los beneficios que las mismas brindan y de qué forma utilizarlas para el desarrollo.
- El proceso de desarrollo del software permitió generar todos los artefactos y documentación correspondiente al mismo.
- Para examinar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron los casos de prueba, estos arrojaron resultados satisfactorios, demostrando el cumplimiento de los requerimientos funcionales establecidos en la fase inicial del proceso de desarrollo del producto.
- Como resultado de la investigación se logró solucionar el problema a resolver que da origen a la misma, logrando el desarrollo del módulo de gestión y publicación de señales.

RECOMENDACIONES

- Dar seguimiento a las actualizaciones de HTML 5 con el objetivo de conocer si la etiqueta video puede reproducir direcciones streaming que utilicen los protocolos mms y rstp.
- Realizar un manual de usuario para un mejor entendimiento del sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. Plataforma de Televisión Informativa basada en tecnologías libres. Ruber Hernández, Yuniór Montaner. No 2, La Habana : Revista de Ingeniería Electrónica, Automática y Comunicaciones, 2009, Vol. Vol. 1.
2. Gutiérrez. Definicion.de. [En línea] 2008. [Citado el: 25 de 11 de 2010.] <http://definicion.de/modulo>.
3. Real Academia Española. [En línea] [Citado el: 12 de 11 de 2010.] http://buscon.rae.es/draef/SrvltConsulta?TIPO_BUS=3&LEMA=modulo.
4. Televisio y Guía. [En línea] [Citado el: 12 de 11 de 2010.] http://televisioyguia.com/ciudad_Antofagasta_pais_Chile_portaf_Canales+de+television
5. ACADEMIC. [En línea] [Citado el: 12 de 11 de 2010.] http://www.esacademic.com/dic.nsf/eswiki/203629#Televisi.C3.B3n_por_sat.C3.A9lite.
6. Integración de los Servidores de Media en las Redes Empresariales. Limonta, I. R. La Habana : La Revista del Empresario Cubano.
7. EstructureTV. [En línea] [Citado el: 12 de 11 de 2010.] <http://www.estructuretv.com>.
8. EstructureTV. [En línea] [Citado el: 12 de 11 de 2010.] <http://www.estructuretv.com/multiscreen.htm>.
9. Entorno Virtual de Aprendizaje. [En línea] UCI. [Citado el: 12 de 11 de 2010.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_2/Conferencia_3/Materiales_complementarios/Introduccion_a_RUP_y_UML.pdf.
10. Scribd. [En línea] [Citado el: 12 de 11 de 2010.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
11. Adark. SlideShare Inc. [En línea] [Citado el: 20 de 11 de 2010.] <http://www.slideshare.net/Adark/conocimiento-de-programacin-y-un-lenguaje-estructurado-2553596>.
12. Alvarez, M. A. Desarrollo Web. [En línea] 09 de 05 de 2001. [Citado el: 20 de 11 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.
13. El Portal del Webmaster. [En línea] [Citado el: 25 de 11 de 2010.] <http://www.webtaller.com/maletin/articulos/por-que-elegir-php.php>.
14. Maestros del Web. [En línea] [Citado el: 25 de 11 de 2010.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.
15. Cortez, J. Infored. [En línea] 12 de 11 de 2010. [Citado el: 25 de 11 de 2010.] <http://www.infored.com.mx/articulos/hojas-de-estilo-en-cascada-css-.html>.
16. UPLA. [En línea] 13 de 10 de 2009. [Citado el: 25 de 11 de 2010.] <http://sitios.upla.cl/contenidos/2009/10/13/difunden-ventajas-del-framework-symfony>.

17. TreeWeb. [En línea] 10 de 11 de 2010. [Citado el: 25 de 11 de 2010.]
<http://www.treeweb.es/TreeWeb/Articulos/Manual/Symfony-1.2-La-guia-definitiva>.
18. Scribd Inc. [En línea] 2012. [Citado el: 18 de 05 de 2012.]
<http://es.scribd.com/doc/93966104/JQuery>
19. Babylon. [En línea] [Citado el: 2010 de 11 de 25.]
http://www.babylon.com/definicion/entorno_de_desarrollo/
20. NetBeans. [En línea] [Citado el: 25 de 11 de 2010.]
http://netbeans.org/community/releases/68/index_es.html.
21. Andrés, María Mercedes Marqués. Wordpress. [En línea] 12 de 02 de 2001. [Citado el: 25 de 11 de 2010.] <http://uvfdatabases.wordpress.com/2009/02/07/terminos-de-repaso-introductorios-a-bds>.
22. Fher. guatewireless. [En línea] 01 de 08 de 2009. [Citado el: 25 de 11 de 2010.]
<http://www.guatewireless.org/articulos/mysql-vs-postgresql>.
23. Garcerant, Iván. Synergix. [En línea] 10 de 07 de 2008. [Citado el: 18 de 02 de 2011.]
<http://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.
24. Abian Ruíz, Yeray Santana, David Frigolet y Arístides Suárez. Tourism Revolution Blog. [En línea] 13 de 07 de 2011. [Citado el: 10 de 05 de 2012.] <http://www.blogtrw.com/2011/07/la-importancia-de-los-requisitos-en-un-proyecto-de-software/>
25. Marker, Graciela. Informatica-hoy. [En línea] [Citado el: 15 de 02 de 2011.] <http://www.informatica-hoy.com.ar/software-linux/En-busca-del-mejor-reproductor-de-video-para-Linux.php>.
26. Caullieres, Blanca. Alcance Libre. [En línea] 18 de 08 de 2008. [Citado el: 15 de 02 de 2011.]
<http://www.alcance Libre.org/article.php/conoce-gecko-mediaplayer>.
27. Rational Unified Process. s.l. : Corporation, IBM, 2006.
28. EcuRed. [En línea] [Citado el: 20 de 02 de 2011.]
http://www.ecured.cu/index.php/Patr%C3%B3n_Modelo_Vista_Controlador.
29. Buenas Tareas. [En línea] [Citado el: 22 de 02 de 2011.]
<http://www.buenastareas.com/ensayos/Patrones-De-Dise%C3%B1o/1048954.html>.
30. Saavedra, Jorge. El Mundo Informático. [En línea] [Citado el: 22 de 02 de 2011.]
<http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman>.
31. PRUEBASDESFTWARE. [En línea] 2005. [Citado el: 25 de 05 de 2012.]
<http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
32. Natalia Juristo, Ana M. Moreno, Sira Vegas. 2006. Técnicas de Evaluación de software.