

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



## **Módulo de Gestión de Errores de la Plataforma de Televisión Informativa PRIMICIA**

---

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

**AUTOR:** Elaine Morales Alvarez

**TUTOR:** Ing. Yunior Montaner Hernández

Ciudad de la Habana, 2 de Julio del 2010

“Año 52 de la Revolución”

*Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.*

*Un esfuerzo total es una victoria completa.*

**Mahatma Gandhi**

## *DEDICATORIA*

*Dedico mi tesis en especial a mi mamá, por todo su apoyo y dedicación, a mis abuelitos Clara, Pepe, Yeya y a toda mi familia por su cariño diario. A mis amigos por compartir sus vidas conmigo y por estar siempre tan unidos.*

*Quisiera dedicarla también a mis profesores por todos sus consejos y enseñanzas y a todas las personas que me han apoyado y han confiado en mí.*

### *AGRADECIMIENTOS*

*A mi mamá por estar siempre a mi lado, por pensar siempre en mí y por ser además de madre, amiga. A mis abuelos por quererme tanto. A mi tía Elaine, por cuidarme mucho y por haber sido para mí como mi madre. A mi otra abuelita, Yeya, por todo su cariño y dedicación. A toda mi familia por tantos buenos momentos y a Clara, Magalís y Kike por ser parte de ella también.*

*A mi hermanito Elier, Randita y su familia por todo su cariño. A mis sobrinos hermosos, por quererme siempre y lograr que me sienta feliz cada vez que me dicen tía.*

*A mis amigos de toda la vida: Elizabeth, la China, Yaimara, Esgleyth, Iliana, Lorena, Modesto, Osmelí, Albertico, Blasito y a mi hermanito Alarcón por mantener siempre esta amistad unida a pesar de que algunos estábamos lejos y a sus padres por unirse a nosotros en nuestros planes y lograr que fuésemos siempre una gran familia.*

*A Leidiana, Yisel, Daneidis, Rafa, Yake y al Mello, por ayudarme y compartir conmigo no sólo los buenos momentos sino también los malos y por ser mis amigos incondicionales. A Lisandra y Albrecht, por apoyarme muchísimo en este año tan difícil y sobre todo por su gran amistad. A Karlen por todos los años que compartimos y a su familia por preocuparse siempre por mí.*

*A los tesistas de Primicia, por compartir conmigo estos meses de tanto estrés y por ayudarme a desarrollar esta investigación, no tuvimos mejores tutores que nosotros mismos. A mis compañeros de grupo, por hacer que recuerde estos 5 años con mucha alegría. A mis profesores, no sólo los que compartieron conmigo estos últimos años sino también aquellos que me educaron durante toda mi vida. A todos los miembros de mi tribunal, por su paciencia, por apoyarme y encaminar la investigación desde sus inicios.*

*A la Revolución Cubana por crear esta universidad que ha sido nuestra casa durante tanto tiempo y que nos ha forjado como verdaderos ingenieros.*

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Elaine Morales Alvarez

Yunior Montaner Hernández

---

---

### RESUMEN

El polo Video y Sonido Digital de la facultad 9 dedica sus esfuerzos al desarrollo de varias aplicaciones informáticas que tienen como objetivo ampliar la utilización de los medios de comunicación en Cuba y el mundo. Durante varios años se ha trabajado en el perfeccionamiento de la Plataforma de Televisión Informativa PRIMICIA, para que esta sea totalmente configurable y pueda adaptarse a las necesidades de los usuarios.

La presente investigación tiene como objetivo fundamental la implementación de un módulo que permita detectar y dar seguimiento a los errores que ocurren durante la utilización de la plataforma. Mediante el desarrollo de este sistema se podrán detectar los fallos ocurridos y se conocerá cómo darle solución y seguimiento a los mismos.

Para el desarrollo de la aplicación se utilizará el lenguaje de programación del lado del servidor PHP, apoyándose en el framework Symfony y como metodología de desarrollo RUP.

**Palabras claves:** *Plataforma de televisión informativa, PRIMICIA, Errores, Seguimiento, Módulo.*

## ÍNDICE DE TABLAS

Tabla 1. Ventajas y Desventajas del lenguaje HTML. ....	25
Tabla 2. Ventajas y Desventajas del lenguaje Javascript. ....	26
Tabla 3. Ventajas y Desventajas del lenguaje PHP .....	27
Tabla 4. Requerimiento de hardware.....	38
Tabla 5. Requerimiento de hardware.....	39
Tabla 6. Actores del Sistema. ....	40
Tabla 7. Descripción textual del CUS. Detectar error. ....	42
Tabla 8. Descripción textual del CUS. Gestionar error. ....	45
Tabla 9. Descripción textual del CUS. Generar reporte. ....	47
Tabla 10. Descripción textual del CUS. Exportar reporte.....	48
Tabla 11. Descripción textual del CUS. Buscar error.....	50
Tabla 12. Descripción textual del CUS. Generar alerta. ....	51

## ÍNDICE DE FIGURAS

Figura 1. El patrón Modelo- Vista- Controlador.....	32
Figura 2. Diagrama de clases del Modelo de Dominio. ....	34
Figura 3. Diagrama de Caso de Uso del Sistema.....	41
Figura 4. Flujo de trabajo de Symfony. ....	59
Figura 5. Relación de los componentes de Symfony con el paquete controlador y la vista.....	60
Figura 6. Diagrama de clases del diseño correspondiente al Caso de Uso Gestionar error.....	61
Figura 7. Diagrama de clases del diseño correspondiente al Caso de Uso Detectar error.....	62
Figura 8. Diagrama de clases del diseño correspondiente al Caso de Uso Generar reporte. ....	63
Figura 9. Diagrama de clases del diseño correspondiente al Caso de Uso Buscar error. ....	64
Figura 10. Modelo de datos genéricos en Symfony.....	65
Figura 11. Diagrama de clases persistentes.....	66
Figura 12. Modelo Entidad_Relación.....	66
Figura 13. Diagrama de despliegue.....	67

**ÍNDICE**

INTRODUCCIÓN.....5

CAPÍTULO 1. Fundamentación Teórica.....9

    1.1. Introducción.....9

    1.2. Conceptos asociados al dominio del problema.....9

    1.3. Objeto de Estudio.....10

    1.4. Análisis de otras soluciones existentes.....13

    1.5. Conclusiones.....15

CAPÍTULO 2. Tendencias y tecnologías actuales.....16

    2.1 Introducción.....16

    2.2 Metodologías de Desarrollo.....16

    2.3 Lenguaje Unificado de Modelado (UML).....19

    2.4 Herramientas CASE.....19

    2.5 Gestor de Base de Datos.....20

    2.6 Arquitectura Cliente- Servidor.....22

    2.7 Lenguaje de Programación.....23

    2.8 IDE de desarrollo.....28

    2.9 Framework.....30

    2.10 Conclusiones.....32

CAPÍTULO 3. Presentación de la solución propuesta.....33

    3.1 Introducción.....33

    3.2 Entorno donde trabajará el sistema.....33

    3.3 Requerimientos Funcionales.....36

3.4	Requerimientos No Funcionales. ....	37
3.5	Descripción del Sistema Propuesto. ....	40
3.6	Conclusiones. ....	51
CAPÍTULO 4. Construcción de la solución propuesta. ....		52
4.1	Introducción. ....	52
4.2	Patrones de Diseño. ....	52
4.3	Diagrama de Clases. ....	58
4.4	Diseño de la Base de Datos.....	65
4.5	Modelo de Despliegue. ....	67
4.6	Conclusiones. ....	68
CONCLUSIONES .....		69
RECOMENDACIONES.....		70
REFERENCIAS BIBLIOGRÁFICAS.....		71
BIBLIOGRAFÍA.....		74

## INTRODUCCIÓN

Los medios de comunicación son instrumentos en constante evolución. Constituyen una herramienta eficiente para mantener a las personas informadas sobre la situación social, política y económica que vive el mundo. Para bien o para mal, la televisión ha sido uno de los factores con mayor influencia en el desarrollo de la cultura actual y visión del mundo, se ha convertido en un medio de comunicación que influye en la vida de las personas, y tener el control de esta es una aspiración para muchos.

Actualmente la mayoría de las instituciones que poseen una red de televisión tienen la necesidad de que sus trabajadores o clientes conozcan una información en un tiempo determinado. La Universidad de las Ciencias Informáticas (UCI), no es una excepción, desde sus inicios estudiantes y trabajadores han estado trabajando en el desarrollo de una plataforma de televisión capaz de transmitir noticias, combinando texto, video e imagen.

Fue implantada en el 2005 la primera versión de la plataforma de televisión, conocida como Señal 3, transmitiendo diariamente noticias de interés para toda la comunidad universitaria. Un año más tarde a petición del Comandante en Jefe, Fidel Castro fue instalada una solución similar en la Agencia Cubana de Noticias, Señal ACN con el objetivo de que los cubanos que cumplen misión en Venezuela puedan conocer todo lo que ocurre en Cuba y el mundo sin necesidad de buscar esta información en los medios de comunicación donde se tergiversa en gran medida la realidad. Este canal no sólo benefició a los colaboradores sino también al pueblo cubano que vive en las zonas de silencio, permitiendo que ninguna persona esté desinformada por muy lejos que viva.

Muchas han sido las instituciones que se han interesado por transmitir información mediante un canal local. En el 2007 se comienza a comercializar este producto, el primer cliente, Venezuela, específicamente el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET), pero esta institución necesitaba de nuevos requerimientos y es implementada entonces la plataforma de televisión TV Energía.

Con el auge de esta solución informática que además fue la primera que el polo Video y Sonido Digital realizó completamente en software libre se abrieron nuevos mercados, lo que hace que surja la idea de

crear un producto adaptable a cualquier necesidad requerida por el cliente, desarrollándose así el producto PRIMICIA.

Desafortunadamente PRIMICIA ha ido arrastrando algunos de los errores de las versiones que le dieron origen, los cuales han traído consigo que en determinado momento no se pueda continuar con la transmisión del canal. Unos de los problemas que más afecta en estos momentos la confianza por parte de los clientes hacia PRIMICIA es que no se puede tener control sobre los fallos que se puedan presentar durante la utilización del canal. Es imposible conocer cuando ocurre un error a no ser que este sea visible, por ejemplo cuando el título y el pie de la foto de las noticias no salen con la cantidad de caracteres redactados. Actualmente por esta situación existe dependencia por parte del cliente pues este no conoce como darle solución a los errores ni puede hacer un seguimiento y tratamiento a los mismos. Continuamente necesitan de la asistencia de los desarrolladores de la plataforma de televisión informativa, lo cual resulta molesto para ambas partes e impide que más clientes se interesen en adquirir este producto.

Basándose en los argumentos descritos anteriormente, se ha planteado el siguiente **problema científico**: Inexistencia del Módulo de Gestión de Errores de la Plataforma de Televisión Informativa PRIMICIA, en el polo Video y Sonido Digital de la facultad 9.

Para darle solución al problema planteado se define como **objeto de estudio** el procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA y como **campo de acción** se precisa, la automatización de los procesos de seguimiento y tratamiento de los errores de PRIMICIA.

Se concibe como **objetivo general**, Desarrollar el Módulo de Gestión de Errores de la Plataforma de Televisión Informativa PRIMICIA.

A partir del objetivo general se obtiene la siguiente **idea a defender**: Con el desarrollo de una aplicación que permita el seguimiento y tratamiento de los errores en PRIMICIA se logrará conocer el origen y cómo darle solución a los posibles fallos producidos durante la utilización de la plataforma de televisión.

Las siguientes **tareas de investigación** permitirán darle cumplimiento al objetivo general:

- Identificar aplicaciones o soluciones existentes que puedan contribuir a la obtención del análisis y diseño del Módulo de Gestión de Errores.
- Describir las metodologías de desarrollo.
- Caracterizar el lenguaje de modelado y las herramientas CASE (Computer Aided Software Engineering).
- Identificar los requisitos funcionales y no funcionales del sistema.
- Desarrollar el prototipo de interfaz de usuario.
- Determinar los patrones de diseño a utilizar en el desarrollo del Módulo de Gestión de Errores.
- Generar los artefactos correspondientes al rol de Analista en el desarrollo del Módulo de Gestión de Errores de la Plataforma de Televisión informativa.
- Modelar los artefactos correspondientes al rol de Analista del sistema en el desarrollo del Módulo de Gestión de Errores de la Plataforma de Televisión informativa.
- Implementar un prototipo funcional del Módulo de Gestión de Errores.

Para el desarrollo de la investigación se utilizaron métodos científicos que permitieron identificar las características del objeto de estudio con el propósito de conocer su esencia, entre ellos están los métodos teóricos y empíricos.

El método **analítico- sintético** se utilizó para analizar el procedimiento de seguimiento y tratamiento de los errores de PRIMICIA, lo cual permitió sintetizar los elementos más importantes de dicho procedimiento.

El **inductivo- deductivo** se empleó para razonar sobre los errores que se producen durante la utilización del canal logrando hacer una generalización de su comportamiento, esto constituyó el punto de partida

para definir o confirmar formulaciones teóricas. Además sirvió para comparar soluciones informáticas con características similares a PRIMICIA, analizar cómo se le dieron tratamiento a los errores producidos y deducir la mejor solución para resolver el problema de la investigación.

La **modelación** se utilizó para la comprensión de los procesos y lineamientos de desarrollo a seguir en cada una de las fases, a partir de su utilización para el modelado de la aplicación, usando el Proceso Unificado de Desarrollo (RUP) y el Lenguaje Unificado de Modelado (UML).

Los métodos empíricos dieron la posibilidad de extraer de los elementos relacionados con el objeto de estudio la información que se necesitaba de ellos, en el caso de la presente investigación sólo mediante el experimento. El **experimento** fue empleado para verificar la idea a defender planteada en la investigación, es el más complejo y eficaz de los métodos empíricos. Mediante su utilización se logró estudiar los errores en varias condiciones dando la posibilidad de obtener la mejor solución pues se explotaron todas las posibles causas de su aparición.

### CAPÍTULO 1. Fundamentación Teórica.

#### ***1.1. Introducción.***

En este capítulo se abordan varios temas que conforman la fundamentación teórica del desarrollo del Módulo de Gestión de Errores. De manera general, está centrado en la definición de varios conceptos asociados al dominio del problema, la fundamentación del objeto de estudio y la situación problemática y se analizan algunas soluciones existentes a nivel internacional que se enmarcan en el campo del problema planteado.

#### ***1.2. Conceptos asociados al dominio del problema.***

**Video:** proviene del verbo latino videre, y significa "yo veo". Hace referencia a la captación, procesamiento, transmisión y reconstrucción de una secuencia de imágenes y sonidos que representan escenas en movimiento. [3]

El video surge como una tecnología íntimamente ligada a la televisión, pues nació como auxiliar de ésta para evitar que la programación fuera en directo, facilitando el trabajo de grabación, la planeación de horarios, el almacenaje de programas y la reproducción de los mismos. [3]

**Sonido:** El sonido es la sensación que experimenta el oído cuando está sometido a la acción de vibraciones de frecuencias comprendidas entre 20 y 20.000 hertzios. Físicamente el sonido es consecuencia de un movimiento vibratorio, transmitiéndose en un medio elástico, habitualmente el aire, hasta el oído, donde se produce la sensación fisiológica. [4]

**Televisión:** Es un sistema para la transmisión y recepción de imágenes en movimiento y sonido a distancia. Esta transmisión puede ser efectuada mediante ondas de radio o por redes especializadas de televisión por cable. El receptor de las señales es el televisor. [5]

Las primeras transmisiones de televisión no incluían el sonido. La idea de introducir en una misma emisión el vídeo y el audio se fue incorporando con posterioridad, planteándose dos posibilidades. En la primera, la televisión estaba destinada a complementar los programas de radio, transmitiendo imágenes de forma auxiliar. La otra entendía la transmisión de imágenes como aspecto principal junto con el sonido y la relacionaba con la transmisión de películas. [5]

**PRIMICIA:** es una solución integrada que provee de un canal de televisión capaz de unir varios formatos: texto, video y audio con el objetivo de mostrar la información lo más atractiva posible. [1]

Está estructurado en dos subsistemas:

- Subsistema de Administración.
- Subsistema de Transmisión.

El Subsistema de Administración permite la administración del canal y toda la gestión de las noticias y recursos multimedia. El Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados.

**Traza:** En sintaxis, la traza es una marca o huella dejada. Un registro de trazas es la conservación de los datos, en una estructura definida, de un evento sucedido, o también, de un elemento al haberse desplazado a una nueva posición.

### ***1.3. Objeto de Estudio.***

Para el desarrollo de la investigación se definió como objeto de estudio el procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.

#### **1.3.1. Descripción General.**

En el polo Video y Sonido Digital de la facultad 9 de la Universidad de las Ciencias Informáticas se trabaja durante varios años en el desarrollo de la Plataforma de Televisión Informativa PRIMICIA. Hasta el

momento la investigación se ha basado en el perfeccionamiento del proceso de redacción y transmisión de las noticias, sin embargo todo sería en vano si no se logra realizar un seguimiento y tratamiento de los errores que ocurren cuando se usa la plataforma de televisión. Para lograr esto se definió un procedimiento que garantiza la detección de los errores. Este permite conocer la estrategia a seguir para la generación y almacenamiento de las trazas de error, se definen responsabilidades, causas por las que aparece un error y las acciones a tomar ante la aparición de estos.

Todos estos elementos son imprescindibles para lograr un producto con calidad y que se ajuste a las necesidades actuales de PRIMICIA. Por tanto se hace necesario analizar el procedimiento, sirviendo este de base para la implementación del Módulo de Gestión de Errores.

A continuación se mencionan procesos y situaciones que generan errores:

- Parada del servicio de PostgreSQL, este error conlleva a la caída del sistema.
- Error cuando se muestra algún material audiovisual con problemas de códec.
- Parada del servicio de replicación de datos.
- Parada del servicio de actualización de ficheros, imágenes o videos.
- Los títulos y los pie de foto de las noticias cuando se transmiten no salen con la cantidad de caracteres que soporta.
- En algunas ocasiones el sistema no permite que se puedan corregir algunos textos de las noticias.
- La aplicación presenta problemas con ficheros grandes.

### **1.3.2. Situación Problemática.**

La televisión ha logrado llevar información a los lugares más apartados del planeta. Importantes compañías dedican sus esfuerzos hacia la mejora no solo de los formatos de video, audio e imagen

permitiendo que cada vez tengan mejor calidad y utilicen menos espacio para su almacenamiento sino también en desarrollar plataformas de televisión informativas cada vez más eficientes.

Las plataformas de televisión informativas pueden ser utilizadas en cualquier institución que desee emplear este medio para mantener informados a un grupo de personas. En los aeropuertos, terminales de ómnibus y trenes esta opción resulta muy atractiva, pues de esta forma contarían con un canal local por el cual se puede transmitir los horarios de salidas de los medios de transportes u otra información de interés. Estas instituciones no son los únicos clientes potenciales, sino que hoteles, hospitales, centros educacionales o cualquier organismo que tenga una red de televisión puede beneficiarse de las ventajas que ofrece la Plataforma de Televisión Informativa PRIMICIA.

Con el surgimiento de la Universidad de las Ciencias Informáticas surge también la idea de crear un medio de información que pudiese llegar a toda la comunidad universitaria. La amplia red de televisión por cable implementada en la universidad permitió hacer de esta idea una realidad. Muchas han sido las modificaciones realizadas a la primera versión de la plataforma de televisión informativa desarrollada por la UCI conocida como Señal 3. Una de las mejoras más notables es que pasó de ser un producto desarrollado con software propietario a ser una aplicación informática totalmente implementada en software libre. Las nuevas versiones han tomado varios nombres, pero se pretende realizar un producto integrado capaz de adaptarse a cualquier requerimiento solicitado por el cliente.

Aunque PRIMICIA ha sido un producto galardonado en varios eventos importantes aún presenta importantes deficiencias, no permite hacer un seguimiento y tratamiento a los errores que se puedan presentar durante la utilización del canal. Impide así conocer el origen del error y la forma de darle solución en un pequeño plazo. Las anteriores versiones del producto tampoco cuentan con este requerimiento, hasta el momento los errores producidos durante la transmisión son solucionados reiniciando el servidor o incluso programando soluciones provisionales, pero sólo son detectados cuando el canal deja de funcionar, sin que se pueda conocer por qué ocurrió esto. Este inconveniente resulta incómodo para los desarrolladores del software e incluso para los clientes pues estos últimos no tienen conocimientos de programación y necesitan constantemente que se le preste asesoramiento. Si se conociese la forma de resolver cada posible error sería innecesario la dependencia que existe hoy entre el cliente y los desarrolladores.

### ***1.4. Análisis de otras soluciones existentes.***

En Europa la compañía **GFI**, desarrollador internacional de software de seguridad de red, seguridad de contenido y mensajería, ha desarrollado el software GFI Events Manager, el cual permite el seguimiento del funcionamiento de los dispositivos de hardware. Cubre dos funciones principales: monitorización y administración y archivo de sucesos. La primera, ayuda a los administradores a monitorizar el estado y seguridad de toda la red, mejorando la disponibilidad, mientras que la función de administración y archivo permite al administrador centralizar los sucesos de múltiples orígenes en varios formatos de manera que sea más sencillo identificar deficiencias, proporcionar trazas detalladas de auditoría y cumplir diversas regulaciones. [1]

La más reciente compilación de GFI EventsManager mejora el nivel de alertas cuando se detectan en la red sucesos importantes o intrusiones.

GFI EventsManager permite activar acciones tales como: ejecución de secuencias de comandos o envío de alertas a una o más personas por correo electrónico, mensajes de red y notificaciones SMS.

**Centinela** es un software desarrollado por la empresa Convenco, esta es reconocida en el mercado tecnológico pues aporta soluciones integrales e innovadoras, con altos niveles de servicio.

Centinela actúa como un vigilante que se encuentra operativo las 24 horas del día, teniendo como función principal notificar oportunamente a través de mensajes al celular, vía e-mail o chat, cuando algo está funcionando de forma incorrecta.

Con esta capacidad, es posible actuar inmediatamente cuando determinada situación afecte a las máquinas, sistemas o procesos de negocio e, incluso, adelantarse a ciertos hechos que pudiesen provocar un caos a nivel organizacional. Junto con la posibilidad que el cliente efectúe el monitoreo en forma remota, a través de la web, Centinela permite que las notificaciones sean escaladas cuando el encargado no está disponible.

Existe la opción de generar reportes, los cuales se podrán visualizar gráficamente permitiendo realizar gestión sobre ellos, teniendo una perspectiva global y, a la vez, específica del comportamiento de los sistemas.

La **plataforma Communi.TV (CTV)** permite la publicación de contenidos y aplicaciones para web, móviles y televisión digital, mediante componentes reutilizables en diferentes tecnologías que facilitan el despliegue de servicios. Como plataforma de desarrollo multicanal, CTV está orientada a favorecer la productividad y la creación de nuevos formatos en canales digitales.

Cuenta con un sistema de registro de trazas y monitorización de errores. Registra los tiempos de respuesta totales y parciales (transiciones de estados) permitiendo detectar, controlar y corregir los cuellos de botella y la generación de alertas.

Una vez instalado el sistema se proporcionarán una serie de scripts de monitorización que interrogan al gestor de contenidos para determinar su disponibilidad y rendimiento. Los scripts están orientados a recoger métricas de funcionamiento de la plataforma. Con estas métricas y las proporcionadas por el cliente se determinará la necesidad de escalabilidad de la plataforma. [13]

Todos estos sistemas informáticos expuestos anteriormente fueron desarrollados con y para herramientas privativas, por lo tanto no pueden ser utilizados libremente. Analizando el costo que conlleva la utilización de alguna de estas aplicaciones a partir de la compra de su licencia y además de que estas no realizan una detección de todos los errores que ocurren en el canal, se decide realizar el Módulo de Gestión de Errores para la Plataforma de Televisión Informativa. Dicho módulo permitirá que PRIMICIA logre ser un producto atractivo para muchos clientes y que estos sientan confianza al utilizarlo.

### ***1.5. Conclusiones.***

En este capítulo se ha abordado de forma general la necesidad que existe de desarrollar un conjunto de funcionalidades incluidas en el Módulo de Gestión de Errores, que garantizarán el funcionamiento constante y eficiente de la Plataforma de Televisión Informativa PRIMICIA. Se describieron las características del sistema a desarrollar y se analizaron aplicaciones que en estos momentos permiten realizar control de errores. Además de relacionar un conjunto de conceptos que permitirán entender el marco del problema.

### **CAPÍTULO 2. Tendencias y tecnologías actuales.**

#### **2.1 *Introducción.***

En este capítulo se realizará un análisis de las principales tecnologías utilizadas en el mundo para la construcción de aplicaciones web que tienen estrecha relación con la Plataforma de Televisión Informativa PRIMICIA. Se abordan temas relacionados con el uso de las metodologías de desarrollo, lenguajes de programación y su conexión con gestores de bases de datos. El objetivo principal que tiene realizar un análisis de los elementos planteados es dar a conocer qué tecnologías se aplicarán en la implementación del Módulo de Gestión de Errores y el por qué de su selección.

#### **2.2 *Metodologías de Desarrollo.***

Un producto de software en sí es complejo. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de todas las posibles situaciones de ejecución que se puedan presentar (entradas, valores de variables, datos almacenados, software del sistema, otras aplicaciones que intervienen, el hardware sobre el cual se ejecuta, etc.). [2]

Un producto de software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tienen precedentes en productos de software similares. Esto hace que los requisitos sean difíciles de consolidar tempranamente y que los cambios sean inevitables, no sólo después de entregado el producto sino también durante el proceso de desarrollo. [2]

Es por ello que se hace necesario el uso de una metodología de desarrollo de software que sirva de guía durante todo el proceso.

Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Define con precisión los artefactos, roles y actividades, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto y guías para el uso de herramientas de apoyo.

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

Al realizar un análisis de las metodologías de desarrollo de software con el fin de seleccionar una que se ajuste a las características del Módulo de Gestión de Errores, se descartaron las que se clasifican como ágiles. PRIMICIA no cuenta con un cliente específico, y este tipo de metodologías requieren que el cliente del software sea parte del equipo de desarrollo. Es importante destacar que la mayoría de las personas encargadas de desarrollar el Módulo de Gestión de Errores son estudiantes, que cambian de rol constantemente, por lo tanto se necesita generar una amplia documentación que permita que los nuevos miembros del proyecto puedan tener abundante información sobre el problema al que le darán solución.

Entre las metodologías robustas las más conocidas son: Microsoft Solution Framework (MSF) y Rational Unified Process (RUP). Ambas cuentan con características similares, pero teniendo en cuenta que PRIMICIA se desarrolla completamente sobre plataformas libres, MSF no sería una metodología factible pues fue desarrollada por Microsoft y requiere el uso de Microsoft Visual Studio Team System para el modelado de la solución. Además el sistema requiere ser desarrollado en el menor tiempo posible, por lo que resulta innecesario que el equipo de desarrollado estudie el uso de esta metodología cuando RUP ofrece los mismos beneficios, y es una metodología conocida por todos los miembros del proyecto.

RUP al ser iterativo e incremental permitirá que tras cada iteración se pueda analizar la calidad del producto y que se vayan desarrollando prototipos de la aplicación con el fin de probarlos y detectar posibles fallos en su funcionamiento.

Basándose en estos argumentos se decide utilizar RUP como metodología para el desarrollo del Módulo de Gestión de Errores.

### 2.2.1. Proceso Unificado de Desarrollo (RUP).

El proceso unificado de desarrollo (RUP) es una metodología que va más allá del análisis y el diseño orientado a objetos, proporciona una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, iterativo e incremental y centrado en la arquitectura [7]

#### **Beneficios que aporta RUP.**

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- Permite que la producción de software cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.

### **2.3 Lenguaje Unificado de Modelado (UML).**

UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema. Representa un conjunto de especificaciones de notación orientadas a objeto, las cuales están compuesta por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. Su principal objetivo es simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que existían.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar. [8]

Otro de los beneficios que ofrece este modelado visual es que es independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se pueden implementar en cualquier lenguaje. De esta forma la modelación de la aplicación será totalmente reutilizable. Esta característica resulta muy importante pues en futuras versiones de PRIMICIA se puede cambiar de lenguaje de programación, sin afectar la mayoría de la documentación generada. Además no tiene propietario y está basado en el común acuerdo de la comunidad informática.

### **2.4 Herramientas CASE.**

Hoy en día, muchas empresas se han extendido a la adquisición de herramientas CASE, con el fin de automatizar los aspectos claves de todo el proceso de desarrollo de un sistema, desde el principio, hasta el final y así incrementar su posición en el mercado competitivo.

Principales ventajas que ofrecen:

- Ayuda a la reutilización del software.
- Permite la portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Permite un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Para desarrollar el Módulo de Gestión de Errores en el menor tiempo posible es necesario utilizar una herramienta case que permita aumentar la productividad, organizar y manejar la información del proyecto, permitiendo que el sistema se torne flexible y comprensible. Además que no requiera entrenamiento del personal y costos económicos. La herramienta CASE que cuenta con todas estas características es Visual Paradigm por tanto se selecciona para el modelado del sistema.

### **2.4.1. Visual Paradigm.**

Visual Paradigm es una herramienta profesional multiplataforma que soporta el ciclo de vida completo del desarrollo de software: negocio, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. La herramienta UML también proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos. [11].

Permite además la generación de bases de datos mediante la transformación de diagramas de Entidad-Relación en tablas de base de datos. Ofrece una serie de facilidades para la generación de informes que permiten la documentación del proyecto así como la exportación en diferentes formatos tales como: XMI y XML.

## **2.5 Gestor de Base de Datos.**

Un sistema gestor de base de datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos [12].

Ayuda a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad, manipulación y privacidad de los datos.

El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

### **Un SGBD está compuesto por varios elementos:**

- **El gestor de la base de datos.**

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. El gestor almacena una descripción de datos en lo que llamamos diccionario de datos, así como los usuarios permitidos y los permisos. Tiene que haber un usuario administrador encargado de centralizar todas estas tareas. [12]

- **Diccionario de datos.**

Es una base de datos donde se guardan todas las propiedades de la base de datos, descripción de la estructura, relaciones entre los datos, etc.

- **El administrador de la base de datos.**

Es una persona o grupo de personas responsables del control del sistema gestor de base de datos.

- **Los lenguajes.**

Un sistema gestor de base de datos debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos. Estos lenguajes son los siguientes:

- Lenguaje de definición de datos (DDL). Para definir los esquemas de la base de datos.
- Lenguaje de manipulación de datos (DML). Para manipular los datos de la base de datos.
- Lenguaje de control de datos (DCL). Para la administración de usuarios y seguridad en la base de datos. [12]

### 2.5.1. PostgreSQL.

Es un gestor de base de datos que se incluye entre los gestores objeto-relacionales porque aunque permite la herencia entre las tablas esta no se puede realizar entre objetos porque no existen. PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

PostgreSQL incorpora además la gestión de usuarios y los permisos que se le asignan a cada uno de ellos, además de funciones de diversa índole: manejo de fechas, operaciones con redes, etc. Permite la declaración de funciones propias, soporta el uso de índices, reglas y vistas.

La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. En el caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podrá ser restaurada. [23] Esto puede ser enormemente beneficioso pues uno de los errores que ocurren con mayor frecuencia en el canal es precisamente que deja de funcionar el servicio de la base de datos. De esta forma cualquier cambio que fue escrito en la base de datos pueden ser recuperado usando el dato que fue previamente registrado. Cuando el sistema haya sido habilitado nuevamente, el usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

### 2.6 *Arquitectura Cliente- Servidor.*

El Módulo de Gestión de Errores se basará en la arquitectura de red cliente- servidor, esta separa el cliente (a menudo una aplicación que utiliza interfaz gráfica) del servidor. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y el servidor, al proceso que responde a las solicitudes del cliente. [30]

Los principales componentes de este modelo son: cliente, servidor y la infraestructura de comunicación. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente. Los clientes frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el

servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad. [30]

Los servidores proporcionan un servicio al cliente y devuelven los resultados. La plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan computadoras eficaces, minicomputadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad. [30]

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones o protocolo de comunicación, la cual proporciona los mecanismos básicos de direccionamiento y transporte. [30]

Entre las principales características de la arquitectura Cliente-Servidor, se destacan las siguientes:

- El cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### ***2.7 Lenguaje de Programación.***

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué

acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. [14]

Los lenguajes de programación se clasifican en de: bajo nivel y alto nivel.

De la misma manera en que los lenguajes de alto nivel permiten que el programador escape de las complejidades del orden del código de la máquina, los sistemas de programación de niveles más bajos pueden proporcionar ayuda en la comprensión y en la manipulación de sistemas y componentes complejos. [14].

Basándose en que el Módulo de Gestión de Errores se desarrollará utilizando tecnologías web y sobre software libre, la selección de los lenguajes a utilizar para su desarrollo se rigió por estos aspectos. Además se tuvo en cuenta que fuesen conocidos por los implementadores del sistema o que presenten una sintaxis sencilla, para no emplear mucho tiempo en aprender a trabajar con ellos.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de internet, fueron existiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la web, que permitieran interactuar con los usuarios y que utilizaran sistemas de bases de datos. [15]

Los lenguajes que se emplearán para darle solución al problema científico planteado se dividieron según la arquitectura cliente- servidor explicada anteriormente para lograr una mejor comprensión de su relación con ella.

### **2.7.1. Lenguajes del lado del cliente.**

#### **HTML.**

Una página web se ve en el navegador, o cliente web, y parece una sola entidad, pero no es así, está compuesta por multitud de diferentes ficheros, como son las imágenes, los posibles vídeos y lo más

importante: el código fuente. El código de las páginas está escrito en un lenguaje llamado HTML, que indica básicamente donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. [16]

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de personas que lo utilizarían en un futuro.

A continuación se mencionan algunas de sus ventajas y desventajas.

Ventajas	Desventajas
Sencillo, permite describir hipertexto.	Lenguaje estático.
Texto presentado de forma estructurada y agradable.	La interpretación de cada navegador puede ser diferente.
Archivos pequeños.	El diseño es más lento.
Despliegue rápido.	Las etiquetas son muy limitadas.

**Tabla 1. Ventajas y Desventajas del lenguaje HTML.**

### **Javascript.**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias y es utilizado principalmente en páginas web. El principal problema de JavaScript es que su comportamiento depende del navegador. [18] Aunque el lenguaje soporta cuatro tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que dificulta el desarrollo de programas complejos, pero ayuda a programar con rapidez macros sencillas.

A continuación se mencionan algunas de sus ventajas y desventajas.

Ventajas	Desventajas
Los script tienen capacidades limitadas, por razones de seguridad.	El código debe descargarse completamente.
El código Javascript se ejecuta en el cliente.	Disponibilidad de objetos limitada.
Fácil de aprender.	Es necesario usarlo conjuntamente con otros lenguajes más evolucionados.

**Tabla 2. Ventajas y Desventajas del lenguaje Javascript.**

Este lenguaje de programación será utilizado para crear páginas web dinámicas, para mostrar ventanas con mensajes de aviso al usuario y para crear animaciones y otras funcionalidades.

### 2.7.2. Lenguajes del lado del servidor.

#### **PHP.**

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código PHP como su entrada y creando páginas web como salida. [17]. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente de este, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de

## CAPÍTULO 2. Tendencias y tecnologías actuales

datos [21] tales como: MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Esta característica permite que si se desea cambiar de gestor de base de datos en futuras versiones del módulo, se necesite cambiar pocas cosas en la implementación del mismo.

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (Linux o Mac OS X) y Windows, y puede interactuar con los servidores web más populares.

A continuación se mencionan algunas de sus ventajas y desventajas.

Ventajas	Desventajas
Soporta en cierta medida la orientación a objeto.	Se necesita instalar un servidor web.
Amplia documentación.	La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
Incluye gran cantidad de funciones.	La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
No requiere definición de tipos de variables.	

**Tabla 3. Ventajas y Desventajas del lenguaje PHP**

Entre las características que se tuvieron en cuenta para la selección de PHP como lenguaje del lado del servidor se encuentran:

- Es un lenguaje de programación creado especialmente para el desarrollo de aplicaciones web dinámicas, esta característica le da la posibilidad de tener predefinidas un gran número de funciones que agilizan el desarrollo de la aplicación.

- La experiencia que tiene el equipo de desarrollo en la utilización de este lenguaje de programación, lo que permite que se exploten al máximo todas sus funcionalidades para lograr así una solución eficiente y novedosa.

### **2.8 IDE de desarrollo.**

Un entorno de desarrollo integrado o IDE es un programa informático compuesto por un conjunto de herramientas de programación.

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 (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

#### **Desventajas de los IDEs:**

- Consumen más recursos.
- Son más lentos.
- Para manejarlos es necesario cierto aprendizaje.
- Ausencia de un IDE universal.

El IDE de desarrollo a utilizar debe soportar trabajar con varios lenguajes de programación, específicamente con los seleccionados para la implementación del Módulo de Gestión de Errores y permitir la integración con el framework de desarrollo escogido para la implementación de la aplicación, además de ser un programa libre. Existen un gran número de entornos de desarrollo que cumplen con estas restricciones, a continuación se mencionan otras características que se tuvieron en cuenta para la selección de Eclipse como el IDE que se utilizará para desarrollar el sistema.

### 2.8.1. Eclipse

Es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins. Además, da soporte a todo tipo de proyectos que abarcan el ciclo de vida del desarrollo de aplicaciones, incluyendo soporte para modelado. [25]

La arquitectura de plug-ins permite integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones accesorias. Conservan el registro de las versiones, generan y mantienen la documentación de cada etapa del proyecto. [25]

Eclipse es soportado por los principales sistemas operativos: Linux, Windows, Solaris 8, Mac OSX – Mac/Carbon. [25]

Entre las partes del programa están: un editor de código PHP con diversas ayudas que aparecen mientras se va escribiendo y autocompletado de código, un editor de proyectos o archivos, que deben estar colgados en algún servidor web con soporte para PHP, y un sistema para hacer debug en PHP y depurar las aplicaciones web de una manera sencilla.

Es importante destacar el navegador de proyectos que lleva incorporado, el cual permite tener los archivos de la página web a mano. El navegador de funciones, muestra un listado de todas las funciones existentes en el fichero en el cual se encuentra el programador, lo cual resulta altamente útil cuando se tiene un archivo con un gran número de operaciones. Este navegador da la posibilidad de moverse con total libertad dentro del proyecto, pudiendo duplicar ficheros, eliminarlos, etc.

Incluye también un controlador de versiones integrado en el entorno permitiendo controlar las versiones de los ficheros con total comodidad desde el mismo navegador del proyecto.

Mientras se va escribiendo el código, el IDE va señalando dónde hay fallos, pero no sólo la línea, si no el punto exacto. También incluye un manual completo de PHP, la opción de buscar en manuales externos y mostrar los resultados directamente en el editor.

En conclusión, eclipse es un entorno de programación que puede competir con cualquier paquete de software comercial.

### **2.9 Framework.**

Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

El framework seleccionado para el desarrollar el Módulo de Gestión de Errores es Symfony principalmente porque proporciona varias herramientas y clases que permiten reducir el tiempo de desarrollo de una aplicación web. Además, automatiza todo el proceso de acceso a datos, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de la aplicación.

#### **2.9.1. Symfony.**

Es un framework PHP pensado y diseñado para optimizar el desarrollo de aplicaciones web. Ha incorporado las ventajas de varios frameworks existentes e incluido ideas propias. El resultado de estas combinaciones es un framework elegante, estable, productivo y muy bien documentado. [20]

#### **Características de Symfony.**

Symfony automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo:

- Uso de plantillas en la capa de presentación.
- Validación automatizada y relleno automático de datos en los formularios.
- Sistema de caché eficiente.
- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.

- Paginación automatizada. Symfony incluye una clase especial para paginar resultados llamada `sfPropelPager`, que realiza la paginación de forma automática.
- Las interacciones con Ajax son muy fáciles de implementar.

### **Modelo Vista Controlador (MVC).**

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles [28]:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. [28]

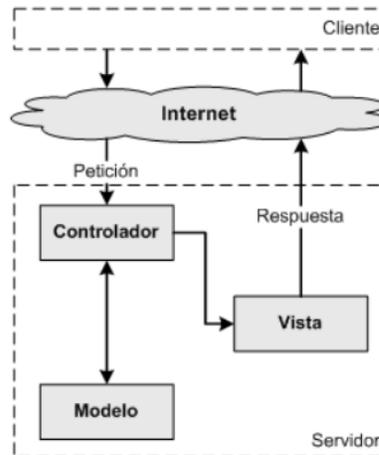


Figura 1. El patrón Modelo- Vista- Controlador.

### **2.10 Conclusiones.**

En este capítulo se expusieron las ventajas de la herramienta CASE a utilizar para el modelado de los artefactos que servirán de guía para el desarrollo del Módulo de Gestión de Errores. Además se definieron los lenguajes de programación web a utilizar, el gestor de base de datos así como el framework que dará soporte a la aplicación. De esta forma quedan definidas las tecnologías a utilizar para el desarrollo del producto.

### **CAPÍTULO 3. Presentación de la solución propuesta.**

#### ***3.1 Introducción.***

En este capítulo, partiendo del análisis realizado sobre el entorno que engloba el problema se dará a conocer la solución propuesta para resolver el mismo. Además se identificarán los requerimientos funcionales y no funcionales del sistema, así como se realizará una descripción de los actores y los casos de usos para lograr un mejor entendimiento del producto a desarrollar.

#### ***3.2 Entorno donde trabajará el sistema.***

La Plataforma de Televisión Informativa PRIMICIA, está concebida como un producto que puede ser configurado y comercializado para varios clientes. Esto impide que se pueda realizar un modelo de negocio, pues resulta difícil identificar los procesos que intervienen en este. Por este motivo se decide realizar sólo un modelo de dominio además del glosario de términos que servirá para identificar y definir los vocablos presentes en dicho modelo.

El modelado de dominio tiene como objetivo describir y comprender las clases y objetos más significativos dentro del contexto del problema, lo cual permite definir los procesos y roles más relevantes para el sistema a desarrollar. Esto ayuda a los usuarios, desarrolladores y clientes, manteniendo un lenguaje único y consistente que evite confusiones y posibilite el entendimiento de todas las partes interesadas.

[29]

##### **3.2.1. Eventos principales del entorno.**

El proceso de detección de errores comienza cuando el **proceso de monitorización** encuentra que están ocurriendo fallos en el funcionamiento de la plataforma de televisión informativa. Cuando esto ocurre inmediatamente se envía un mensaje de error al **usuario** para que conozca que no puede continuar haciendo uso de los servicios de la aplicación hasta que sea solucionado el problema.

Los administradores del sistema reciben un correo electrónico informándoles de la dificultad presentada, para que esta sea solucionada lo antes posibles y el resto de los usuarios del sistema puedan continuar haciendo uso de las funcionalidades que brinda la plataforma.

El proceso de monitorización genera una **traza** con algunos elementos importantes correspondiente al error ocurrido la cual es guardada en la **base de datos** para luego ser utilizada de ser necesario.

### 3.2.2. Diagrama de clases del Modelo de Dominio.

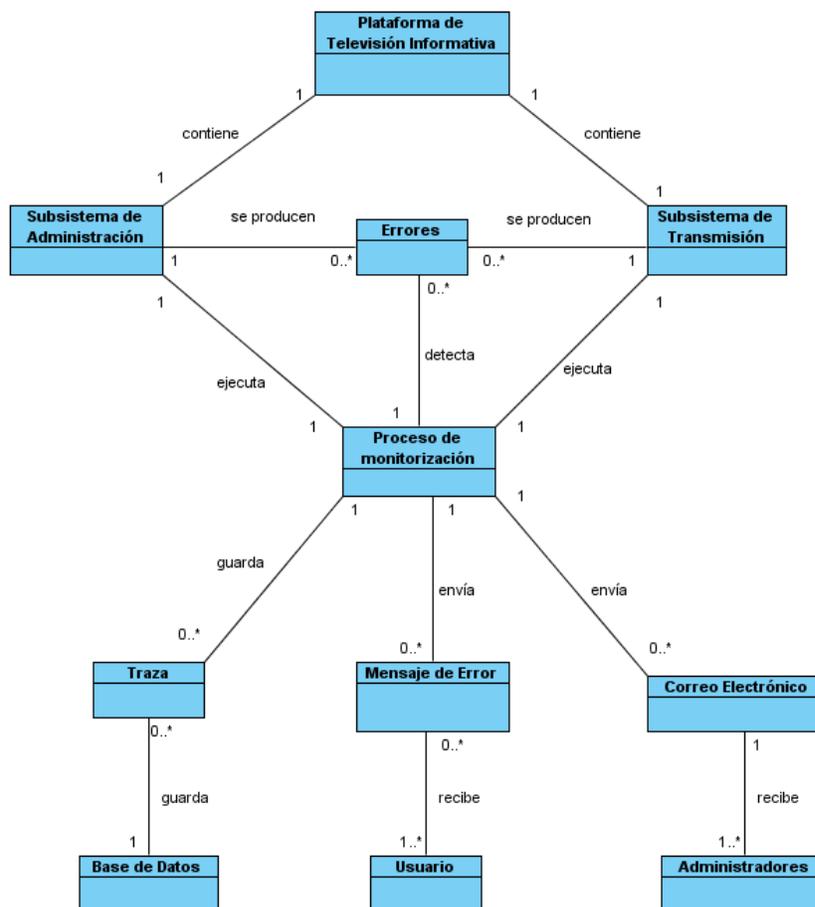


Figura 2. Diagrama de clases del Modelo de Dominio.

### 3.2.3. Glosario de Términos del Dominio.

**Error:** En la Plataforma de Televisión Informativa PRIMICIA, ocurren varios errores que impiden el funcionamiento eficiente del canal. A continuación se explican los que afectan en mayor medida la utilización de la plataforma.

- Parada de la aplicación de transmisión.

Los textos, imágenes y videos al combinarse forman también parte de la noticia y ésta es transmitida a través de la aplicación de transmisión. En ella se encuentran ejecutándose un grupo de herramientas que son las responsables de que esto suceda. Un problema surgido aquí significa que la transmisión de las noticias se ve afectada completamente, provocando además que se detenga la aplicación y la plataforma estaría prácticamente fuera de servicio.

- Parada del servicio de replicación.

El servicio de replicación es el que se encarga de enviar todos los datos al servidor de transmisión para que éstos sean utilizados y transmitidos.

Si el volumen de los datos crece rápidamente y existe una gran concurrencia de transacciones en el sistema lo mejor sería hacer uso de la replicación. Mediante esta acción se mantiene actualizado el servidor de transmisión en cuanto a las noticias nuevas que se adicionan, así como también de videos u otros materiales.

Un fallo en esta operación trae graves afectaciones en el sistema, no se actualiza el servidor de transmisión, el cual estará mostrando las mismas noticias continuamente, y en el peor de los casos se cae la transmisión.

- Parada del servicio de PostgreSQL.

El servicio del PostgreSQL es uno de los motores principales en el funcionamiento de la plataforma. La interrupción de éste trae como consecuencia la caída de la aplicación, ya que alrededor del 90% de las funcionalidades que brinda la plataforma dependen de este servicio.

- Caída de la red entre servidores.

Este problema es típico en arquitecturas donde los servidores están separados. El fallo en las conexiones entre servidores, provoca la caída de todos los servicios que presta la plataforma, ya que la transferencia de datos e informaciones entre los servidores se ve afectada seriamente. En el caso que se esté usando el servicio de replicación entre los dos servidores, las afectaciones podrían ser de menor grado, pero no se actualizaría la base de datos que se encuentra en la parte de transmisión y se estarían transmitiendo las mismas noticias una y otra vez.

**Traza:** cuando ocurre un error el sistema debe crear una traza sobre el fallo ocurrido, guardando un conjunto de datos que caracterizan al error detectado: servidor, descripción, atendido, fecha y hora.

**Proceso de monitorización:** Es el encargado de iniciar el proceso de detección de errores, generando la traza correspondiente e informando de la falla ocurrida en la aplicación.

**Usuario:** Persona que interactúa con la plataforma para la administración y/o transmisión de noticias.

**Base de datos:** Entidad en la cual se almacena de manera estructurada la traza generada durante la detección de un error.

### ***3.3 Requerimientos Funcionales.***

**R1: Detectar error.** El sistema debe ser capaz de detectar los errores que ocurren.

**R2: Gestionar error.** El sistema debe permitir gestionar los errores.

R2.1: Eliminar error.

R2.2: Atender error.

R2.3: Visualizar descripción.

**R3: Buscar error.** El sistema debe permitir la búsqueda de errores atendiendo a diferentes criterios.

**R4: Generar reporte.** El sistema debe permitir la generación de reportes como parte del proceso de seguimiento de errores.

**R5: Exportar reporte.** El sistema debe permitir exportar un reporte para que este sea guardado en formato PDF.

**R6: Generar alerta:** Cuando ocurre un error el sistema debe mandar una alerta de forma automática a los administradores del sistema informando que ocurrió una falla.

### ***3.4 Requerimientos No Funcionales.***

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. A continuación se detallan dichos requerimientos basándose en diferentes clasificaciones.

#### **3.4.1. Software.**

- Se debe utilizar como servidor web el Apache.
- Se debe utilizar como sistema operativo en los servidores una personalización del sistema operativo “Nova-Linux”.

#### **3.4.2. Hardware.**

A continuación se muestran los requerimientos mínimos (filas rojo claras) y óptimos (filas blancas) recomendados que debe cumplir el equipamiento tecnológico de la plataforma:

## CAPÍTULO 3. Presentación de la solución propuesta

Variante de un (1) servidor:

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

**Tabla 4. Requerimiento de hardware.**

Variante de dos (2) servidores:

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

	Dual- Core	1 Gb	80 Gb	Ethernet	Exportadora ATI
	Xeon 1.60 Ghz			Gigabit	Radeon X300

**Tabla 5. Requerimiento de hardware.**

### **3.4.3. Apariencia o interfaz externa.**

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

### **3.4.4. Confidencialidad.**

Acceder a la información según el rol del usuario.

### **3.4.5. Disponibilidad.**

- El sistema debe estar disponible los 7 días de la semana y las 24 horas del día, garantizando el buen funcionamiento de la plataforma.
- Los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

### **3.4.6. Usabilidad.**

El sistema será usado por usuarios que no tienen necesariamente conocimientos informáticos por tanto las funcionalidades deben ser claras y se debe mostrar la información de forma lógica y correctamente estructurada.

### **3.4.7. Soporte.**

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

### 3.4.8. Restricciones en el diseño y la implementación.

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

### 3.5 Descripción del Sistema Propuesto.

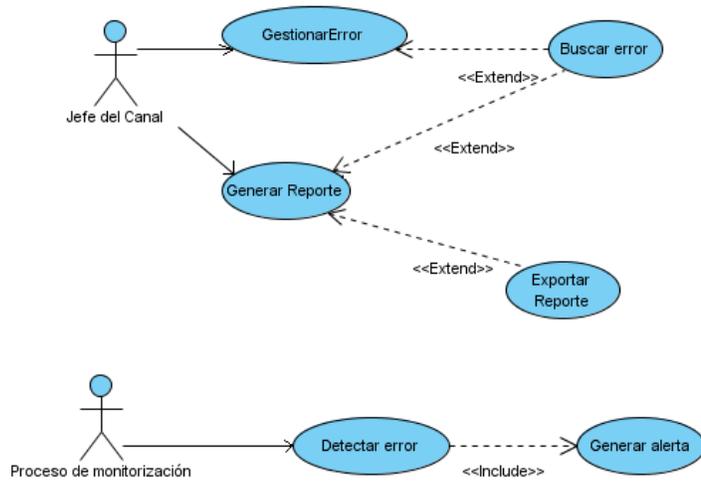
#### 3.5.1. Descripción de los actores.

Actor	Descripción
Jefe del Canal	Persona encargada de controlar toda la actividad de seguimiento y tratamiento de los errores.
Proceso de monitorización	El proceso de monitorización representa al sistema el cual inicia la detección de los errores, generando la traza correspondiente e informando de la falla ocurrida.

Tabla 6. Actores del Sistema.

#### 3.5.2. Casos de Uso del Sistema.

En este epígrafe se muestra la realización de los Casos de Uso del Sistema.



**Figura 3. Diagrama de Caso de Uso del Sistema.**

### Descripción del Caso de Uso Detectar error.

<b>Caso de Uso:</b>	Detectar error.	
<b>Actores:</b>	Proceso de monitorización.	
<b>Resumen:</b>	El caso de uso se inicializa cuando se comienza a realizar una monitorización del funcionamiento del canal.	
<b>Referencias</b>	RF1	
<b>Prioridad</b>	Alta	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El actor determina que es necesario realizar una monitorización del funcionamiento del canal.		2. El sistema realiza una monitorización en busca de errores.

	<p>3. El sistema genera una traza con los datos de los errores encontrados (Fecha, Hora, Servidor, Descripción, Atendido), y esta es guardada en la base de datos.</p>
	<p>4. El sistema genera una alerta, ver Caso de Uso “Generar alerta”.</p>
<b>Poscondiciones</b>	Se detectan los errores y se informa a los usuarios de la ocurrencia de los mismos.

**Tabla 7. Descripción textual del CUS. Detectar error.**

### Descripción del Caso de Uso Gestionar error.

<b>Caso de Uso:</b>	Gestionar error.
<b>Actores:</b>	Jefe del Canal.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea eliminar, atender o visualizar la descripción de algún error.
<b>Precondiciones:</b>	El actor se ha autenticado y tiene los permisos necesarios para acceder a la funcionalidad
<b>Referencias</b>	RF 2
<b>Prioridad</b>	Alta
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>1. El sistema muestra un listado con los errores y la información de ellos: fecha,</p>

	hora, servidor. Además de las opciones: eliminar (A), modificar estado (B), visualizar descripción (C) y buscar (D).
<p>2. El usuario selecciona una opción:</p> <p>a) Eliminar error, Sección “Eliminar error”.</p> <p>b) Atender error, Sección “Atender error”.</p> <p>c) Visualizar Descripción, Ver Sección “Visualizar descripción”.</p> <p>d) Buscar, Ver Caso de Uso “Buscar error”.</p>	
<b>Sección Eliminar error.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona el error que desea eliminar y presiona el botón de Eliminar.	2. Muestra un mensaje de confirmación para eliminar el error, dando la posibilidad de aceptar o cancelar la operación.
3. Confirma la eliminación.	4. Elimina el error del sistema.
<b>Sección Atender error</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona el error al cual ya se le ha dado solución y presiona el botón Atender.	2. Muestra un mensaje confirmando la operación.
3. Confirma la acción.	4. Marca el error como atendido.

Sección Visualizar descripción	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el error del que desea ver la descripción y selecciona la opción Visualizar Descripción.	2. Muestra en una ventana la descripción del error.
3. Cierra la ventana de descripción.	4. Cierra la ventana.

Prototipo de Interfaz

Identificador	Atendido	Fecha	Hora	Servidor	
45		26/1/2010	3:00pm	Transmisión	  
69		5/2/2010	9:00am	Administración	  



**Visualizar descripción.**

Descripción [Close]

El canal no se encuentra funcionando en estos momentos, puede que el servicio de Postgres no se haya iniciado o se ha detenido repentinamente.

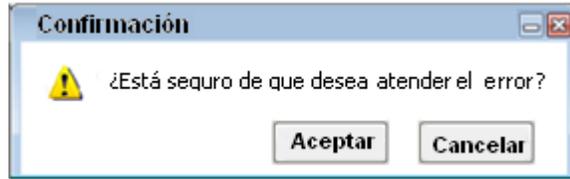
**Mensaje para confirmar la eliminación de un error.**

Confirmación de eliminación. [Close]


¿Está seguro de que desea eliminar el error?

Aceptar
Cancelar

**Mensaje para confirmar que se desea atender el error.**



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. Sección <b>Eliminar error</b> . Cancela la eliminación del error.	4. Cancela la acción.
3. Sección <b>Atender error</b> . Cancela la acción.	4. El error queda como no atendido.
	2. Sección <b>Atender error</b> . Muestra un mensaje informando al usuario que el error ya ha sido atendido.
<b>Poscondiciones</b>	Se obtiene la gestión de los errores.

Tabla 8. Descripción textual del CUS. Gestionar error.

**Descripción del Caso de Uso Generar Reporte.**

<b>Caso de Uso</b>	Generar reporte.
<b>Actores:</b>	Jefe del Canal.
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea conocer el comportamiento de los errores realizando una búsqueda atendiendo a diferentes criterios. El sistema le muestra la información que desea según el tipo de búsqueda realizada y le da

	la posibilidad de exportar e imprimir el reporte generado quedando de esta forma finalizado el caso de uso.															
<b>Precondiciones:</b>	El actor se ha autenticado y tiene los permisos necesarios para <i>Generar reporte</i> .															
<b>Referencias</b>	RF 4															
<b>Prioridad</b>	Baja															
<b>Flujo Normal de Eventos</b>																
<b>Sección “Generar reporte”</b>																
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>															
1. El caso de uso comienza cuando el actor accede a la funcionalidad Gestión de reportes y realiza una búsqueda de los errores (A) que desea que aparezcan en el reporte, (Ver CU. Buscar error).	2. El sistema muestra el listado resultante de la búsqueda realizada y muestra la opción:  a) <i>Exportar reporte (B)</i> .															
3. Selecciona una opción:  a) <i>Exportar reporte</i> , ver caso de uso “Exportar reporte”.																
<b>Prototipo de Interfaz</b>																
<b>Mostrar reporte de errores.</b>																
<table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th style="width: 15%;">Identificador</th> <th style="width: 20%;">Atendido</th> <th style="width: 15%;">Fecha</th> <th style="width: 15%;">Hora</th> <th style="width: 35%;">Servidor</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">45</td> <td></td> <td style="text-align: center;">26/1/2010</td> <td style="text-align: center;">3:00pm</td> <td style="text-align: center;">Transmisión</td> </tr> <tr> <td style="text-align: center;">69</td> <td></td> <td style="text-align: center;">5/2/2010</td> <td style="text-align: center;">9:00am</td> <td style="text-align: center;">Administración</td> </tr> </tbody> </table> <div style="display: flex; justify-content: center; align-items: center; gap: 20px;"> <div style="text-align: center;"> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">A</span>  </div> <div style="text-align: center;"> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">B</span>  </div> </div>		Identificador	Atendido	Fecha	Hora	Servidor	45		26/1/2010	3:00pm	Transmisión	69		5/2/2010	9:00am	Administración
Identificador	Atendido	Fecha	Hora	Servidor												
45		26/1/2010	3:00pm	Transmisión												
69		5/2/2010	9:00am	Administración												

<b>Pos condiciones</b>	Se generó el reporte de la información solicitada por el actor.
------------------------	---

**Tabla 9. Descripción textual del CUS. Generar reporte.**

### Descripción del Caso de Uso Exportar reporte.

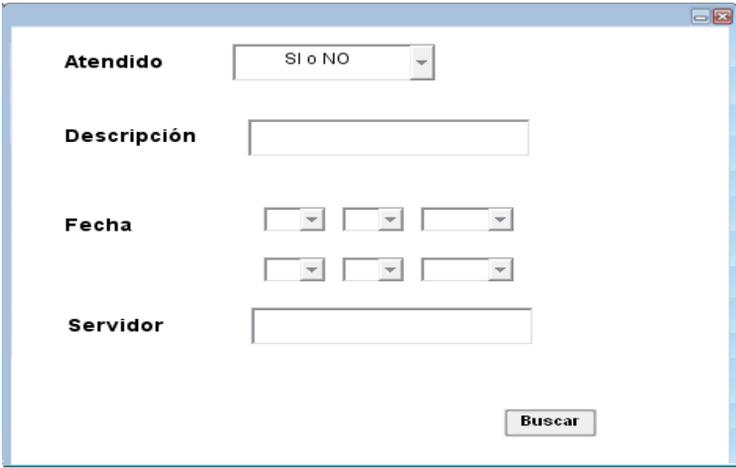
<b>Caso de Uso</b>	Exportar reporte.
<b>Actores:</b>	Jefe del Canal.
<b>Resumen:</b>	Se inicia como parte del CU Generar reporte cuando el actor decide exportar la información a un formato digital, se crea un documento PDF en el que aparece el reporte generado por el sistema y se le da la posibilidad al usuario de guardar físicamente este archivo.
<b>Precondiciones:</b>	Se ha generado el reporte que se desea exportar.
<b>Referencias</b>	RF 5
<b>Prioridad</b>	Baja
<b>Flujo Normal de Eventos</b>	
<b>Sección “Exportar reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

	<ol style="list-style-type: none"> <li>1. Genera un documento en formato digital PDF que incluye:             <ol style="list-style-type: none"> <li>a. Encabezado y pie de página.</li> <li>b. Reporte generado.</li> </ol> </li> </ol>
<ol style="list-style-type: none"> <li>2. Escoge el lugar donde desea guardarlo y presiona el botón Aceptar.</li> </ol>	<ol style="list-style-type: none"> <li>3. Guarda el reporte en el lugar especificado por el usuario.</li> </ol>
<b>Flujo Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección Exportar reporte</b> , Acción 2. Cancela la operación.	Cancela la operación.
<b>Pos condiciones</b>	Se exporta el reporte en formato digital PDF y se guarda en el lugar especificado por el usuario.

**Tabla 10. Descripción textual del CUS. Exportar reporte.**

### Descripción del Caso de Uso Buscar error.

<b>Caso de Uso</b>	Buscar error.
<b>Actores:</b>	Jefe del Canal.
<b>Resumen:</b>	Es iniciado en distintos momentos como parte de varios casos de uso cuando es necesario buscar un error de acuerdo a diferentes criterios. El caso de uso devuelve un listado de los errores encontrados brindando todos los datos del mismo.
<b>Referencias</b>	RF 3

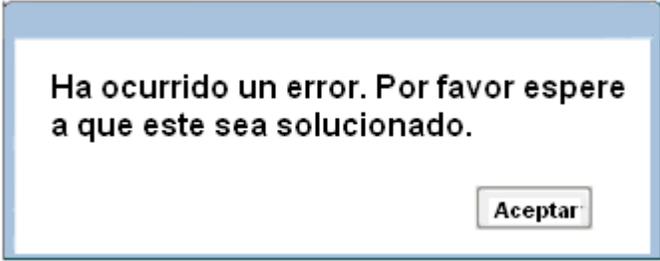
<b>Prioridad</b>	Baja		
<b>Flujo Normal de Eventos</b>			
<b>Sección "Buscar error"</b>			
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>	
		<ol style="list-style-type: none"> <li>1. El sistema muestra un conjunto de criterios: fecha de inicio, fecha final, descripción, servidor, y si desea buscar errores que ya han sido atendidos y/o los que no.</li> </ol>	
<ol style="list-style-type: none"> <li>2. El usuario llena los campos según la búsqueda que desea realizar.</li> </ol>		<ol style="list-style-type: none"> <li>3. Muestra el listado de los errores encontrados.</li> </ol>	
<b>Prototipo de Interfaz</b>			
			
<b>Flujos Alternos</b>			
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>	

	<p><b>Sección Buscar error</b>, Acción 4. Muestra un mensaje informando que no se encontraron errores que tuviesen las características especificadas.</p>
<p><b>Prototipo de Interfaz</b></p> <p><b>No se encontraron resultados</b></p>	
Pos condiciones	Se obtiene un listado de los errores encontrados según los parámetros especificados por el usuario.

Tabla 11. Descripción textual del CUS. Buscar error.

**Descripción del Caso de Uso Generar alerta.**

Caso de Uso	Generar alerta.	
Actores:	Proceso de monitorización.	
Resumen:	Es iniciado en distintos momentos como parte del caso de uso Detectar error. El caso de uso muestra un mensaje de error a los usuarios cuando ocurre algún problema en el sistema y envía un correo a los administradores informando sobre la situación presentada.	
Referencias	RF 6	
Prioridad	Media	
Flujo Normal de Eventos		
Sección “Generar alerta”		
	Acción del Actor	Respuesta del Sistema

	<p>1. El sistema muestra un mensaje de error al usuario.</p> <p>Envía un correo automático a los administradores del sistema.</p>
<p>2. El usuario presiona el botón Aceptar.</p>	<p>3. Cierra la ventana emergente.</p>
<p><b>Prototipo de Interfaz</b></p> 	
<p><b>Pos condiciones</b></p>	<p>Se muestra un mensaje de alerta al usuario y se envía un correo automático a los administradores del sistema.</p>

**Tabla 12. Descripción textual del CUS. Generar alerta.**

### **3.6 Conclusiones.**

En este capítulo se definió mediante el Modelo de Dominio el entorno del negocio, describiendo los errores más comunes que afectan el funcionamiento de la plataforma de televisión informativa en estos momentos. Se determinaron los requerimientos funcionales y no funcionales, y se desglosó el sistema en casos de usos, lo cual permite que se entienda con mayor facilidad cómo funcionará el sistema y la forma en la que se le dará solución al problema planteado.

### **CAPÍTULO 4. Construcción de la solución propuesta.**

#### ***4.1 Introducción.***

En el presente capítulo se describen un conjunto de elementos relacionados con el framework Symfony, lo cual permite que se entienda con mayor facilidad los diagramas de diseño propuestos para cada caso de uso. Estos diagramas constituyen el punto de partida para la implementación del Módulo de Gestión de Errores.

#### ***4.2 Patrones de Diseño.***

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. [26]

Al aplicar un patrón, el código resultante no tiene por qué delatar el patrón o patrones que lo inspiró. No obstante, últimamente hay múltiples esfuerzos enfocados a la construcción de herramientas de desarrollo basados en los patrones y frecuentemente se incluye en los nombres de las clases el nombre del patrón en que se basan, facilitando así la comunicación entre desarrolladores. [26]

Es difícil reutilizar la implementación de un patrón. Al aplicar un patrón aparecen clases concretas que solucionan un problema determinado y que no será aplicable a otros problemas que requieran el mismo patrón.

#### **Ventajas:**

Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente. Por tanto están basados en la recopilación del conocimiento de los expertos en desarrollo de software. Es una experiencia real, probada y que funciona.

### 4.2.1. Patrones para asignar responsabilidades (GRASP).

La calidad de diseño de la interacción de los objetos y la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender. Una implementación hábil se funda en los principios cardinales que rigen un buen diseño orientado a objetos. En los patrones GRASP se codifican algunos de los principios, que se aplican al preparar los diagramas de interacción. [26]

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

#### **Experto**

**Problema:** ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?

Un modelo de clase puede definir docenas y hasta cientos de clases de software, y una aplicación tal vez requiera el cumplimiento de cientos o miles de responsabilidades. [26]

Si estas se asignan en forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se puede presentar la oportunidad de reutilizar los componentes en futuras aplicaciones. [26]

**Solución:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

**Aplicación:** Este es uno de los más utilizados por Symfony, puesto que Propel es la librería externa que este utiliza para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

**Beneficios de su uso:** Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.

El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clase “sencillas” que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión.

### **Creador**

**Problema:** ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?

La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilización. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. [26]

**Solución:** Asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos: [26]

- B agrega los objetos A.
- B contiene los objetos A.
- B registra las instancias de los objetos A.
- B utiliza especialmente los objetos A.
- B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A). B es un creador de los objetos A.

Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A.

**Aplicación:** En la clase Actions se definen acciones y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

### Bajo Acoplamiento

**Problema:** ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. Acoplamiento alto significa que una clase recurre a muchas otras clases. Esto presenta los siguientes problemas: [26]

- Los cambios de las clases afines ocasionan cambios locales.
- Difíciles de entender cuando están aisladas.
- Difíciles de reutilizar puesto que dependen de otras clases.

**Solución:** Asignar una responsabilidad para mantener bajo acoplamiento.

El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión. Sin embargo, es un factor a considerar cuando se intente mejorar el diseño. [26]

**Aplicación:** La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

### Alta Cohesión

**Problema:** ¿Cómo mantener la complejidad dentro de límites manejables?

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. [26]

Una baja cohesión hace muchas cosas no afines o realiza trabajo excesivo. Esto presenta los siguientes problemas:

- Son difíciles de comprender.
- Dificiles de reutilizar.
- Dificiles de conservar.
- Las afectan constantemente los cambios.

**Solución:** Asignar una responsabilidad de modo que la cohesión siga siendo alta.

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

### Controlador

**Problema:** ¿Quién debería encargarse de atender un evento del sistema?

Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema. [26]

Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. [26]

**Solución:** Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

- El “sistema” global (controlador de fachada).
- La empresa u organización global (controlador de fachada).

- Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
- Un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados “Manejador<NombreCasodeUso>” (controlador de casos de uso).

**Aplicación:** Todas las peticiones web son manejadas por un solo controlador frontal 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.

### 4.2.2. Patrones GOF.

#### Patrones Creacionales.

**Singleton** (Instancia única): El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una. En este caso, si más de un objeto necesita utilizar una instancia de la clase Singleton, esos objetos comparten la misma instancia de la clase Singleton. En un uso más avanzado, este patrón puede ser utilizado también para administrar  $n$  instancias de una clase. [25]

El patrón singleton también es muy útil cuando existen datos en una aplicación que van a ser compartidos por muchas clases y no se quiere cargar en memoria todas las veces que se va a utilizar porque ocupa mucho tiempo. [25]

En el controlador frontal de Symfony hay una llamada a `sfContext::getInstance()`. El método `getContext()`, es un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

**Abstract Factory** (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando.

Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. [27]

### Patrones Estructurales.

**Decorator** (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. [27]

**Composite** (Objeto compuesto): Permite tratar objetos compuestos como si se tratase de uno simple. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera. [27]

### 4.3 Diagrama de Clases.

Los diagramas de clases del diseños se realizaron basándose en la implementación que realiza Symfony de la arquitectura MVC, quedando estructurada de la siguiente manera:

**Controlador:** Se divide en un controlador frontal y en las acciones. [29]

- **Controlador frontal:** Ofrece un punto de entrada único para toda la aplicación, controla todas las peticiones de la misma, carga la configuración y determina la acción a ejecutarse. [29]
- **Acciones:** Incluyen el código específico del controlador de cada página y pueden incluir la lógica de la aplicación cuando esta no es muy compleja. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación. Utilizan el modelo y definen variables para la vista. Las acciones son métodos con el nombre **executeNombre** de una clase llamada **nombreActions** que hereda de la clase **sfActions**. [29]

**Vista:** Se separa en un layout y en una plantilla.

## CAPÍTULO 4. Construcción de la solución propuesta

- **Layout:** Es global en toda la aplicación o al menos en un grupo de páginas y contiene el código HTML común para estas.[29]
- **Plantilla:** Presentan los datos de la acción que se está ejecutando y se encarga de visualizar las variables definidas en el controlador. [29]

**Modelo:** Se divide en la capa de acceso a datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. [29]

En la figura que se muestra a continuación se describe el flujo de trabajo de Symfony, basándose en lo anteriormente explicado:

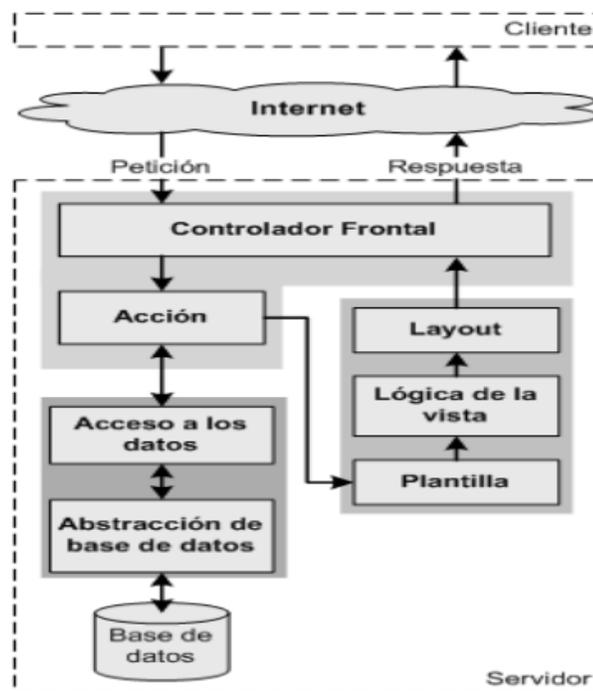


Figura 4. Flujo de trabajo de Symfony.

## CAPÍTULO 4. Construcción de la solución propuesta

Cuando un cliente realiza una solicitud, esta es recibida por el controlador frontal (representado por `index.php` en los diagramas de clases del diseño), que se encarga de decodificarla y mediante los mecanismos de enrutamiento del framework seleccionar la acción que se ejecutará para dar respuesta a la petición. La acción hace uso del modelo, si es necesario, para iniciar las variables que se mostrarán. [29]

Los componentes internos del framework identifican la plantilla que corresponde a la acción que se ejecuta, esta le da formato a las variables modificadas en la acción y luego es decorada con el layout. Finalmente el controlador frontal construye la respuesta a la petición y la envía al cliente.

En el diagrama que se muestra a continuación se representan las clases más relevantes que describen el comportamiento interno del framework y sus relaciones con las clases del paquete controlador y la vista.

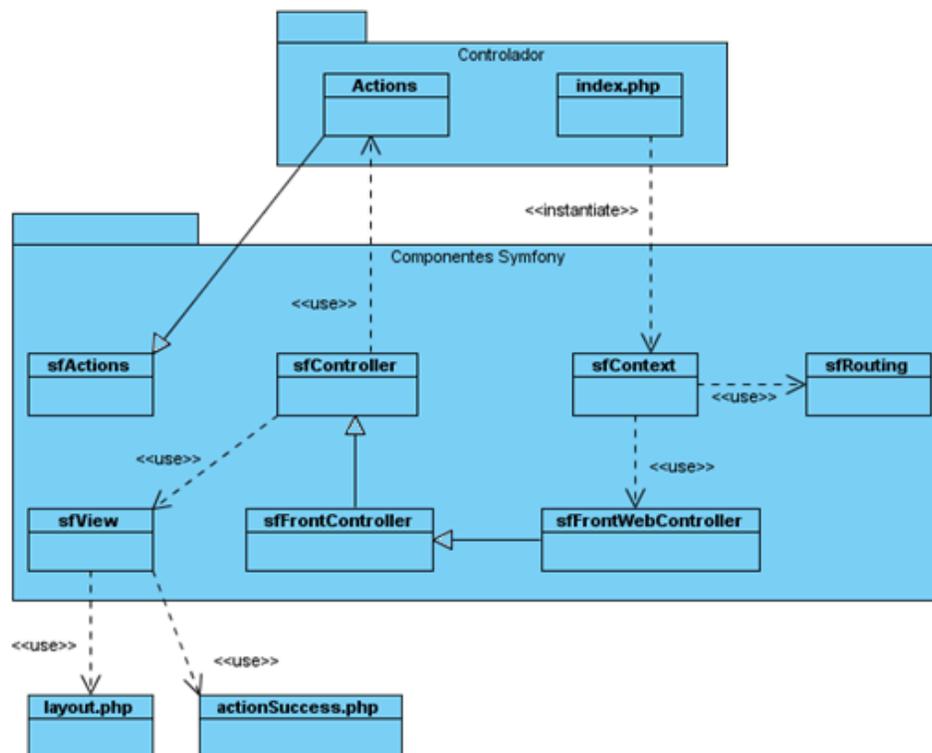


Figura 5. Relación de los componentes de Symfony con el paquete controlador y la vista.

Luego de analizado el flujo de trabajo de symfony se presentan los diagramas de clases del diseño, divididos por casos de uso para proporcionar un mejor entendimiento del diseño propuesto.

## 4.3.1. Caso de Uso. Gestionar error

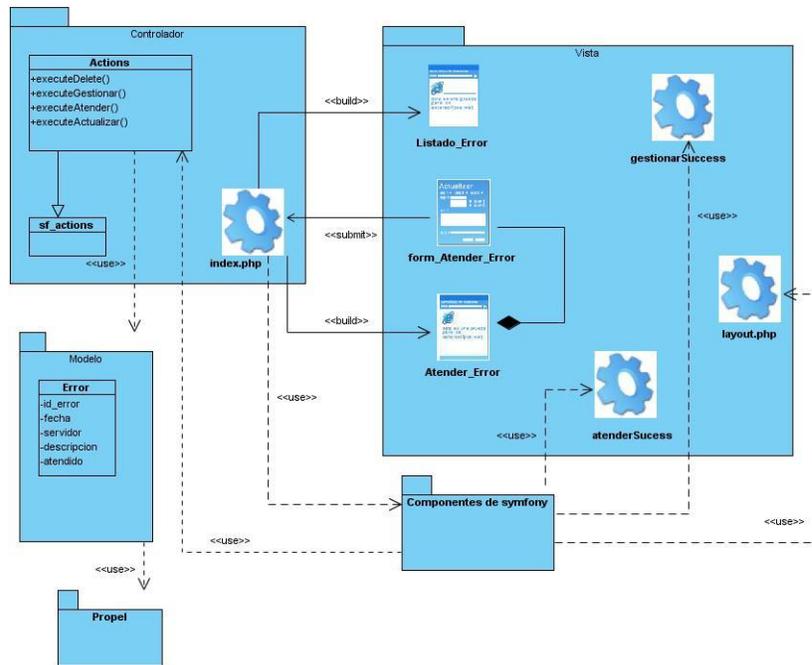


Figura 6. Diagrama de clases del diseño correspondiente al Caso de Uso Gestionar error.

4.3.2. Caso de Uso. Detectar error.

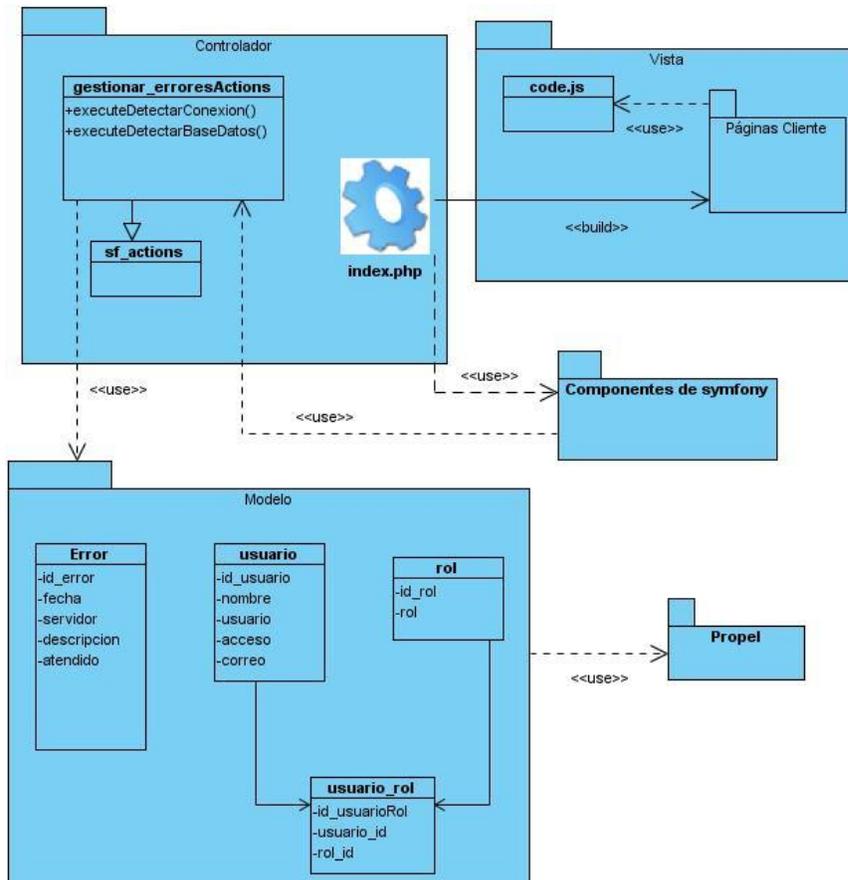


Figura 7. Diagrama de clases del diseño correspondiente al Caso de Uso Detectar error.

4.3.3. Caso de Uso. Generar reporte.

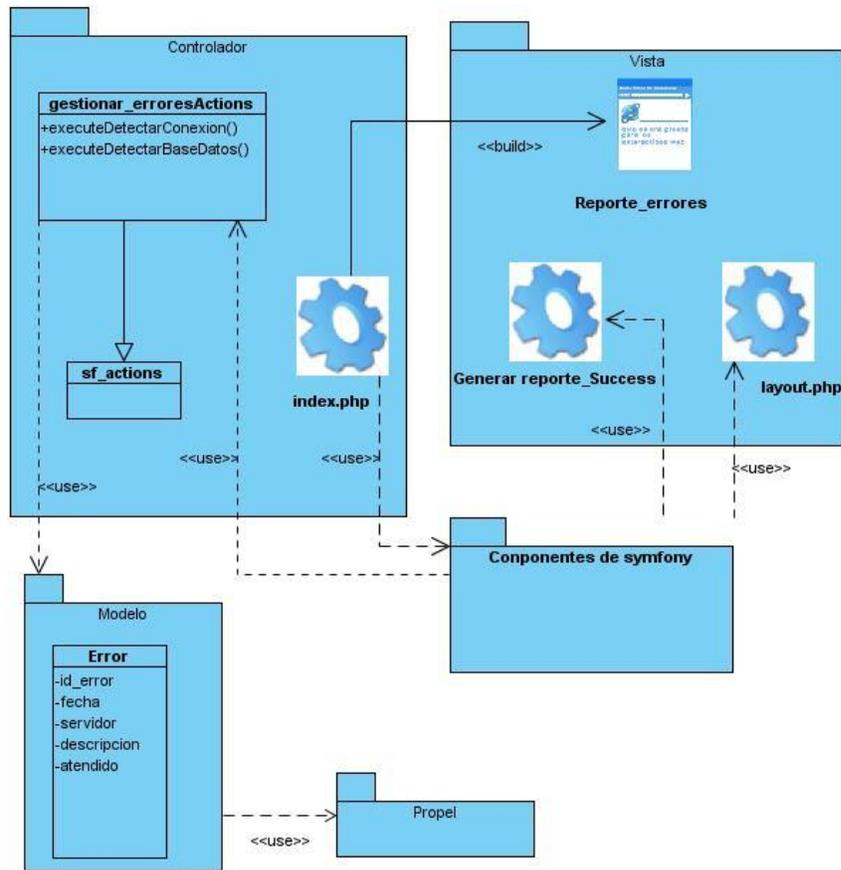


Figura 8. Diagrama de clases del diseño correspondiente al Caso de Uso Generar reporte.



### 4.4 Diseño de la Base de Datos.

En el proceso de construcción de un software resulta necesario el diseño de la base de datos. El buen desarrollo de este permite que el acceso a la información sea sencillo y fácil de realizar. En el presente epígrafe se muestra el diagrama de clases persistentes y el modelo entidad\_relación.

#### 4.4.1. Diagrama de clases persistentes.

El modelo de datos de un proyecto Symfony es generado automáticamente por el framework y está constituido por cuatro clases por cada tabla de la base de datos, estas son:

**baseNombreTablaPeer:** Son clases que tienen métodos estáticos para trabajar con las tablas de la base de datos. Proporcionan los medios necesarios para obtener los registros de las tablas y sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada. Representan la parte del modelo que se encarga del acceso a los datos. [28]

**baseNombreTabla:** Son clases objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados. Representan la parte del modelo que se encarga de la abstracción de los datos. [28]

**nombreTabla, nombreTablaPeer:** Sirven para extender las funcionalidades de sus clases bases correspondientes. [28]

La representación genérica del modelo quedaría como se muestra en la siguiente figura:

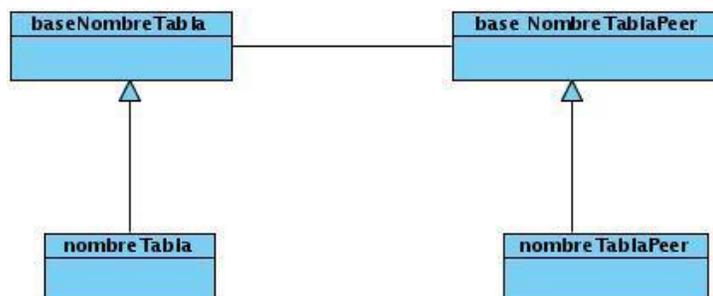


Figura 10. Modelo de datos genéricos en Symfony.

# CAPÍTULO 4. Construcción de la solución propuesta

Basándose en lo expuesto anteriormente y teniendo en cuenta que son las clases baseNombreTabla las que mapean las tablas de la base de datos a objetos, se decide sean estas consideradas como las clases persistentes.

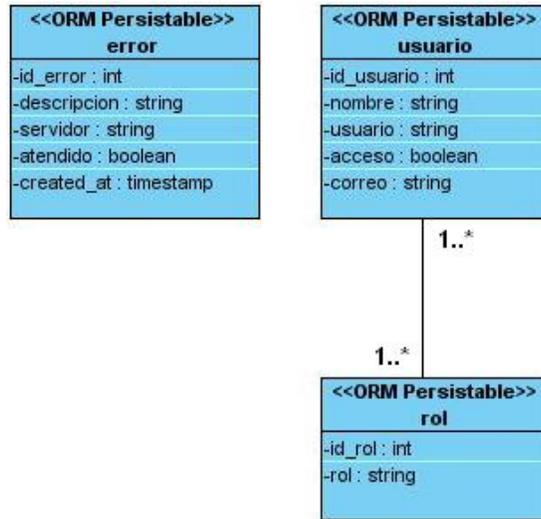


Figura 11. Diagrama de clases persistentes.

## 4.4.2. Modelo Entidad\_Relación.

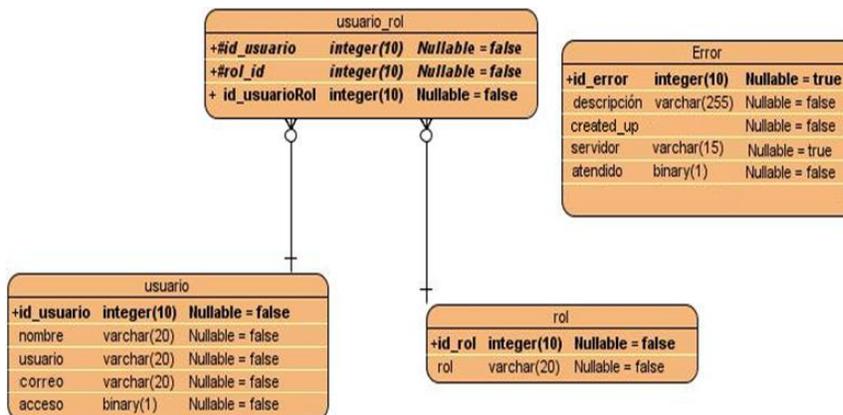


Figura 12. Modelo Entidad\_Relación.

### 4.5 Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo [7], dando una medida de la tecnología necesaria para el correcto funcionamiento del mismo y de cómo se satisfacen los requerimientos no funcionales de hardware y software.

A continuación se muestra el diagrama de despliegue que describe la aplicación que se desarrolla y se describen cada uno de sus componentes.

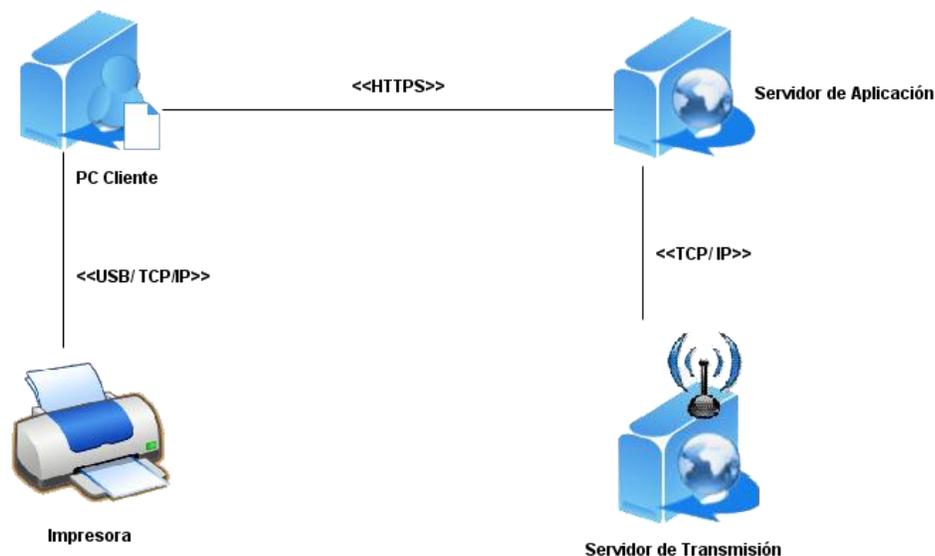


Figura 13. Diagrama de despliegue.

**PC Cliente:** Su función es acceder al sistema e interactuar con el mismo según sus necesidades. Su conexión es a través del Protocolo Seguro de Transferencia de Hipertexto (HTTPS por sus siglas en inglés).

**Servidor de Aplicaciones:** 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.

**Servidor de Transmisión:** Es el encargado de transmitir las noticias.

**Impresora:** Dispositivo encargado de imprimir los reportes generados por los clientes.

### ***4.6 Conclusiones.***

Con el desarrollo de este capítulo se diseñaron los diagramas de diseño, los cuales servirán de inicio para la implementación del Módulo de Gestión de Errores. Además se modelaron otros artefactos que pertenecen al flujo de trabajo Análisis y Diseño y que contribuyen al diseño de la base de datos, logrando que de esta manera quede modelada la construcción de la solución propuesta.

### CONCLUSIONES

El Módulo de Gestión de Errores permitirá que PRIMICIA resulte un producto más atractivo y confiable pues los clientes tendrán control de los errores que se puedan producir durante la utilización de la plataforma. Por todo lo antes expuesto se puede concluir que:

- Se profundizó sobre los conocimientos relacionados con el procedimiento de seguimiento y tratamiento de los errores en la Plataforma de Televisión Informativa PRIMICIA.
- Se modeló adecuadamente el sistema que posibilitó la automatización de las funcionalidades requeridas para el desarrollo del Módulo de Gestión de Errores.
- Se seleccionaron las tecnologías que sirvieron de apoyo para implementar el Módulo de Gestión de Errores.
- Se desarrolló el prototipo funcional de un sistema libre para gestionar los errores que se producen durante la utilización de PRIMICIA.

Después de lo visto anteriormente se puede afirmar que el objetivo de la investigación fue cumplido pues se logró desarrollar el Módulo de Gestión de Errores ampliando de esta forma las funcionalidades que brinda hoy la Plataforma de Televisión Informativa PRIMICIA.

### **RECOMENDACIONES**

Para lograr una integración completa del Módulo de Gestión de Errores con la Plataforma de Televisión Informativa se recomienda que la interfaz de usuario sea modificada y adaptada al diseño actual de PRIMICIA.

Se recomienda además ampliar la cantidad de errores que detecta en estos momentos el Módulo de Gestión de Errores, pues sólo fueron implementados los que resultan críticos para el funcionamiento del canal, pero existen otros que influyen en la utilización de la plataforma.

### REFERENCIAS BIBLIOGRÁFICAS

1. **Calzado, Frank Benitez.** *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.* Ciudad de la Habana : s.n., 2009.
2. Introducción a la Ingeniería de software. [En línea] [http://eva.uci.cu/file.php/102/Curso\\_2009-2010/Semana\\_1/Conferencia\\_1/Materiales\\_Basicos/C1\\_guia\\_estudiante.doc](http://eva.uci.cu/file.php/102/Curso_2009-2010/Semana_1/Conferencia_1/Materiales_Basicos/C1_guia_estudiante.doc).
3. **Jack, Keith.** *Dictionary of Video and Television Technology.* United State of America. s.l. : Elsevier Sciencie, 2002. 1-878707-99-X.
4. **Morrilla, Jose A. Poó.** *Curso práctico, sonido.* 2006.
5. **Sanjúan, Olga Perez.** *Recordando la Historia: El concepto de "televisión" en sus orígenes.* 2007. pág. 160. ISSN 0210-3923.
6. **Beck, K.** *Extreme Programming. Embrace Change.* s.l. : Pearson Education, 1999.
7. **Jacobson Ivar, Rumbaugh James , Booch Grady.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Education, 200. 84 7829 066 2.31.
8. **Orallo, Enrique Hernández.** eva. [En línea] [Citado el: 15 de 11 de 2009.] [http://eva.uci.cu/file.php/102/Curso\\_2008-2009/Materiales\\_Complementarios/Materiales\\_Complementarios\\_Conf\\_1/UML.pdf](http://eva.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf).
9. **Perisse, Marcelo Claudio.** [En línea] [Citado el: 19 de 10 de 2009.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm..>
10. **Mestras, Juan Pavón.** *Rational Rose.* Madrid : Sistemas Informáticos y Programación, Universidad de Madrid.

11. Visual Parading. [En línea] [Citado el: 3 de 11 de 2009.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(Iglesia\\_Anglicana\)\\_%5BMac\\_OS\\_X\\_cuenta\\_14717\\_p/..](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/)
12. Sistemas Gestores de Base de Datos. [En línea] 31 de 7 de 2007. [Citado el: 5 de 10 de 2009.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html...>
13. *Plataforma Communi. TV. Gestión de contenidos multicanal y servicios interactivos.* 2006.
14. **Louden, Kenneth C.** *Lenguajes de programación: principios y prácticas.* Mexico : s.n., 2004.
15. **Valdes, Damián Perez.** [En línea] 2007. [Citado el: 7 de 10 de 2009.] [http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/..](http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/)
16. **Alvarez, Miguel Angel.** Lenguaje HTML. [En línea] 14 de 3 de 2002. [Citado el: 25 de 11 de 2009.] <http://www.desarrolloweb.com/articulos/711.php..>
17. **González, Carlos D.** *Lenguaje PHP.* Buenos Aires : s.n., 2009.
18. **M, G. Figueroa.** *Introducción a JavaScript.* Costa Rica : s.n., 2008.
19. **Alvarez, Miguel Angel.** Lenguaje de programación de propósito general, orientado a objeto, que también puede utilizarse para el desarrollo web. [En línea] [Citado el: 7 de 12 de 2009.] <http://www.desarrolloweb.com/articulos/1325.php..>
20. **Mata, Manel Perez.** *Que es Symfony.* 2009.
21. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid.** *Manual de PHP.* s.l.: Grupo de documentación de PHP, 2001.
22. **A, Ernesto Quiñones.** *Introducción a PosrgreSQL.* s.l. : Asociación peruana de software libre.
23. **John.** *Características de PostgreSQL.* 2006.

24. **Laura Bermejo Sanz, Enrique Gómez Monreal.** *Eclipse como IDE de desarrollo.* s.l. : Universidad Rey Juan Carlos.
25. **García, Pedro.** *Patrón Singleton.* España : s.n., 2005.
26. **UCI.** [En línea] [Citado el: 25 de 1 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=14077..>
27. **Rabaix, Thomas.** *Patrones de diseño que implementa symfony.* 2008.
28. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.*
29. **Marín, Angel Dayán.** *Desarrollo de la Plataforma VideoWeb. Rol Analista.* Ciudad de la Habana : s.n., 2009.

## BIBLIOGRAFÍA

1. Alvarez, Carlos. **METODOLOGIA DE LA INVESTIGACION CIENTIFICA. Santiago de Cuba : s.n., 1995.**
2. Hernández, Rolando. **EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA.** Ciudad de la Habana: s.n., 2002.
3. Day, Robert. **Cómo escribir y publicar trabajos científicos.** Washington: s.n., 2005.
4. Mendoza Sanchez, María A. **Metodologías De Desarrollo De Software. Perú: S.A.C., 2007.**
5. Bayarre, H y Hersford, R. **Metodología de la Investigación. Ciudad de La Habana: Ciencias Médicas, 2004.**