

**Universidad de las Ciencias Informáticas**  
**FACULTAD 6**



**Título: Módulo de reportes y notificaciones de eventos  
para PRIMICIAv2.0.**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autores:** Maralys Domenech Hidalgo

Elieser Corona Silot

**Tutor:** Ing. Félix Iván Romero Rodríguez

La Habana, Junio 2013

“Año 55 de la Revolución”

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Maralys Domenceh Hidalgo (autor)

Elieser Corona Silot (autor)

---

---

Ing. Félix Iván Romero Rodríguez (tutor)

---

## DATOS DE CONTACTO

**Tutor:** Ing. Félix Iván Romero Rodríguez.

**Formación Académica:** Ingeniero en Ciencias Informáticas (Julio/2011).

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** [firomero@uci.cu](mailto:firomero@uci.cu)

## AGRADECIMIENTOS

*Maralys Domenech Hidalgo*

*Le agradezco a mi eterno papa: Frank Fonseca Viera, por ser el primero en depositar su confianza en mí y estar presente en mis momentos de desespero ante cualquier revés. Hoy recuerdo cuando me decía: -Ten calma, esto es una carrera de resistencia no de velocidad. Y tenía mucha razón, hoy estoy aquí haciendo realidad su sueño.*

*A mi mamá Magnamara, por toda su entrega y dedicación en estos cinco años los cuales no han sido fáciles para ella ni para mí. No se encuentra presente pero se que está ansiosa esperando la noticia de que todo salió bien. Gracias por tu amor, dedicación, entrega, por estar hay siempre que lo necesito, en las buenas y malas, por darme la vida. Te quiero muchísimo mamita.*

*A mi papá René, por todos sus consejos, su apoyo, su amor incondicional y su entrega, que aunque un poco distante en ocasiones nunca dejo de estar al tanto de todo lo que ocurría con su pucha. Gracias mi papito, te quiero muchísimo.*

*A mis hermanas: Daniela, Dianela (nene), por estar siempre presentes y a la pequeña Marialis quien llegó en el momento indicado para llenar mi vida de alegrías, las quiero muchísimo,*

*A mi hermano Renesito, insoportable pero igual lo quiero mucho y a las jimaquas que viene en camino.*

*A mi abuela Nereida, por preocuparse por mí y cuidarme como si fuera mi mamá durante estos cinco años lejos de mi casa.*

*A mis tías:*

*Nati por estar siempre al tanto de cómo me van las cosas y por recargarme la tarjeta para hablar con mi mamá, gracias.*

*Damaris, que aún no logra que le diga tía, pero ella sabe que la quiero y que le agradezco que siempre se preocupe por mí.*

*A la mellizas por ser las encargadas de los envíos de dinero y estar al tanto de mi.*

*En fin a todas, Sírta, Ever, Mirta, La Negra las quiero mucho.*

*A mis primas por todo el apoyo que me han dado, Yanetsis, Marian, Mirtica, Latirza, Maliuska en fin a todas.*

*A resto de mi familia por estar siempre pendientes del progreso de mi carrera.*

*A Deymis por soportarme durante toda la carrera, en mis altas y baja, con mis virtudes y defectos, mi carácter y mi forma de ser, muchas gracias amiga y espero que esta amistad perdure.*

*A mi querido profesor y amigo, que se encuentra en estos momentos en Venezuela, Yudiel (cariñosamente la figura) a quien le debo casi toda mi carrera, por toda su preocupación y por estar hay siempre dispuesto para ayudarme en todo lo que necesité, te quiero muchísimo aunque no sea de la manera que tu quieras, gracias por todo.*

*A mi tutor Félix Iván Romero Rodríguez, el mejor de los tutores, gracias por toda su entrega, dedicación, por el apoyo que nos dio para que todo saliera bien.*

*A mi compañero de tesis, con quien compartí los dolores de cabeza cuando algo no salía como esperábamos pero pese a todas las dificultades supimos salir adelante y lograr el objetivo en común que siempre fue graduarnos.*

*A mis amistades: Andris, por joderlo tanto ante problemas con la PC y que luego de gruñir un poco al final siempre estuvo hay cuando lo necesité, Vivi por explicarme las cosas para las pruebas, Yunet por la espuma del detergente, las papitas y los plátanos fritos, Leydis, Tahimi, Dreyssi, Tamara, Jose, Jorge, Raul, Yovenys, Adrian, Darinca en fin a todos los que me han acompañado durante toda la carrera.*

*A todos lo presentes (...) por estar hoy aquí acompañándome en este momento tan importante de mi vida. Gracias por asistir.*

*Elieser Corona Silot*

DEDICATORIA

*Maralys Domenech Hidalgo*

*Dedico mi tesis a:*

*Mis padres, que al darme la vida me dieron la oportunidad de estar hoy aquí, por todo su amor incondicional.*

*A mi eterno papá: Frank Fonseca Viera, quien por cosas del destino no llegó a ver uno de sus sueños hecho realidad el cual fue verme convertida en una profesional, sueño que hoy se concreta, por todo su amor y dedicación.*

*A mis hermanas: Daniela, Dianela (nene), la pequeña Marialis y a mi hermano Renesito.*

*Elieser Corona Silot*

## RESUMEN

La Plataforma de Televisión Informativa PRIMICIA se desarrolla con el objetivo de proveer un canal de televisión informativo para la transmisión automática y continua de noticias en distintos formatos. Esta plataforma no cuenta con un mecanismo que permita generar reportes sobre los procesos y recursos que se administran en la misma, así como notificar a los responsables de su administración de algunas irregularidades que ocurran en la misma. El presente trabajo de diploma persigue como objetivo desarrollar un módulo que permita generar reportes sobre los recursos y procesos que se llevan a cabo en la Plataforma de Televisión Informativa PRIMICIA, además de notificar por diferentes vías de comunicación, correo electrónico o dispositivos para mensajes de texto, el estado de dichos procesos. Aquí se recogen los resultados de la investigación teórica que se llevó a cabo para dar solución a la problemática en cuestión. Se realizó un análisis de las herramientas y tecnologías a utilizar para su desarrollo, entre ellas RUP<sup>1</sup> como metodología de desarrollo, Visual Paradigm como herramienta CASE<sup>2</sup>, UML<sup>3</sup> como lenguaje de modelado, Symfony2 como marco de trabajo, NetBeans como entorno integrado de desarrollo, PHP<sup>4</sup> y JavaScript como lenguajes de programación y JQuery como biblioteca JavaScript para la presentación del módulo. Se describieron las funcionalidades del mismo a través de los diferentes diagramas realizados según la metodología seleccionada, destacando además que su desarrollo fue sobre tecnologías y herramientas libres. Para la validación del módulo se realizaron pruebas de caja negra y aceptación, arrojando resultados favorables para el producto.

### Palabras claves:

PRIMICIA, Administración, Reporte, Notificaciones.

---

<sup>1</sup>RUP: Proceso Unificado de Desarrollo.

<sup>2</sup>CASE: Ingeniería de Software Asistida por Computadora.

<sup>3</sup>UML: Lenguaje Unificado de Modelado.

<sup>4</sup>PHP: Procesador de Páginas de Hipertextos.



## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SISTEMA DE REPORTES Y NOTIFICACIÓN DE EVENTOS.....</b>	<b>5</b>
1.1 Introducción.....	5
1.2 Conceptos Fundamentales. Sistema de reportes y notificación de eventos .....	5
1.3 Los procesos de monitoreo a través de la generación de reportes y sistemas de notificación de eventos .....	8
1.3.1 Descripción General del objeto de estudio.....	8
1.4 Análisis de otras soluciones existentes .....	9
1.4.1 Ámbito internacional.....	9
1.4.2 Ámbito nacional .....	12
1.5 Descripción actual del dominio del problema .....	13
1.6 Características de la Plataforma de Televisión Informativa, PRIMICIA.....	14
1.7 Conclusiones parciales .....	15
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>16</b>
2.1 Introducción.....	16
2.2 Metodología, lenguajes de programación, herramientas y tecnologías .....	16
2.2.1 Metodología de Desarrollo de Software.....	16
2.2.2 Lenguaje Unificado de Modelado.....	16
2.2.3 Lenguaje PHP .....	17
2.2.4 Lenguaje JavaScript.....	17
2.2.5 Visual Paradigm como herramienta CASE.....	18
2.2.6 Symfony 2.0 como marco de trabajo de desarrollo.....	18
2.2.7 JQuery como biblioteca Javascript.....	19
2.2.8 Netbeans como IDE .....	19
2.2.9 PostgreSQL.....	20
2.3 Proceso de generación de reportes y notificación de eventos en Primicia-v2.0.....	20
2.4 Modelo del dominio.....	21
2.4.1 Diagrama de Clases del Modelo del Dominio .....	22
2.4.2 Descripción de las clases del dominio .....	22
2.5 Requisitos del sistema .....	22
2.5.1 Requisitos funcionales.....	23
2.5.2 Requisitos no funcionales: .....	24
2.6 Descripción de la solución propuesta .....	25

2.6.1 Generación de reportes en la Plataforma de Televisión Informativa, PRIMICIA .	25
2.6.2 Notificaciones en la Plataforma de Televisión Informativa, PRIMICIA.....	26
2.7 Casos de uso del sistema.....	27
2.7.1 Diagrama de Casos de Uso del Sistema.....	27
2.7.2 Caso de uso Generar Reporte .....	27
2.7.2.1 Diagrama de clases del análisis para el caso uso Generar Reporte .....	36
2.7.3 Caso de Uso Notificar Eventos .....	37
2.7.3.1 Diagrama de clases del análisis para el caso de uso Notificar Eventos .....	40
2.8 Conclusiones parciales .....	41
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE REPORTES Y NOTIFICACIONES.....</b>	<b>42</b>
3.1 Introducción.....	42
3.2 Descripción de la Arquitectura de software.....	42
3.2.1 Patrón arquitectónico .....	42
3.2.2 Patrones de diseño .....	43
3.2.2.1 Patrones GRASP .....	44
3.2.2.2 Patrones GOF.....	45
3.3 Diagrama de Clases del Diseño .....	46
3.4 Modelo de Implementación.....	46
3.4.1 Modelo de datos .....	46
3.4.1 Diagrama de componentes.....	47
3.4.2 Diagrama de despliegue.....	47
3.5 Pruebas de software .....	48
3.5.1 Método de prueba de caja negra .....	49
3.5.1.2 Diseños de casos de prueba .....	50
3.5.2 Pruebas de aceptación.....	57
3.6 Conclusiones parciales .....	59
<b>CONCLUSIONES.....</b>	<b>60</b>
<b>RECOMENDACIONES .....</b>	<b>61</b>
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>ANEXOS.....</b>	<b>66</b>
<b>GLOSARIO.....</b>	<b>67</b>

## INTRODUCCIÓN

Enmarcados en la importancia de las nuevas tecnologías, desde un principio, en la Universidad de las Ciencias Informáticas (UCI) fueron creados varios canales televisivos con fines educativos, culturales e informativos. Todo este proceso controlado por la Dirección de Televisión Universitaria (DTU) y apoyada por el Departamento de Señales Digitales del centro Geo-Informática y Señales Digitales (GEYSED) perteneciente a la Facultad 6, en el cual se desarrolla la Plataforma de Televisión Informativa, PRIMICIA la cual constituye una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos.

En los sistemas que llevan a cabo el proceso de gestión de recursos se da la necesidad de tener un control y monitoreo de la administración de los mismos. También se necesita conocer estadísticamente el empleo de estos, de manera que se puedan determinar indicadores como el porcentaje de uso, el porcentaje de errores y de su aprovechamiento. Además es necesario tener controlada la información sobre el rendimiento del servidor en el cual se encuentra desplegada la aplicación, para saber a qué nivel están sus prestaciones con respecto a la misma. En estos sistemas también se hace necesario mantener a los responsables de su administración informados de los eventos que ocurran en el mismo, estas eventualidades pueden ser enviadas a través de notificaciones ya sea por correo electrónico o cualquier otro dispositivo para mensajes de texto.

En la Plataforma de Televisión Informativa, PRIMICIA se llevan a cabo una serie de procesos y se trabaja con distintos tipos de recursos, por lo que se hace compleja la gestión de información referente a los mismos. A los responsables de la administración de dicha plataforma, se les hace difícil obtener información específica de los procesos y recursos que se manejan en la misma, así como tener en cuenta cuales de estos están obsoletos o se encuentran al límite de uso. También se les dificulta estar informados de algunas irregularidades que ocurren en la aplicación y dar solución a estas. Por lo que surge la necesidad de tener un mejor control del sistema, además de mantener a los usuarios responsables de su administración informados sobre el estado de los recursos, procesos y eventos que suceden, a través de notificaciones por diferentes vías de comunicación.

De la anterior situación problemática se identifica como **problema a resolver**: ¿Cómo monitorear el estado de los recursos y procesos que se manejan en la Plataforma de Televisión Informativa, PRIMICIA?

Se plantea como **objeto de estudio** de la investigación: El proceso de monitoreo a través de la generación de reportes y sistemas de notificación de eventos; y se enmarca como **campo de acción**: El proceso de monitoreo a través de la generación de reportes y sistemas de notificación de eventos en la Plataforma de Televisión Informativa, PRIMICIA.

Se propone como **objetivo general** de la investigación: Desarrollar un módulo que permita generar reportes sobre los recursos y procesos que se llevan a cabo en la Plataforma de Televisión Informativa, PRIMICIA y además permita notificar por diferentes vías de comunicación el estado de dichos procesos.

Por lo planteado anteriormente se asume como **idea a defender** que: El desarrollo de un módulo que genere reportes y notificaciones, permitirá monitorear el estado de los recursos y procesos que se manejan en la Plataforma de Televisión Informativa, PRIMICIA.

Como guía para la realización de esta investigación se han propuesto las siguientes **tareas de la investigación**:

- Describir los sistemas de reportes a nivel internacional y nacional.
- Describir los sistemas de reportes en aplicaciones web integrados a framework<sup>5</sup> de desarrollo.
- Describir los sistemas de notificación haciendo uso de los medios más conocidos.
- Describir los sistemas de recomendación a partir de indicadores generados en reportes.
- Definir los requisitos funcionales y no funcionales de la solución.
- Definir el modelo del sistema.
- Diseñar el sistema mediante el uso de una herramienta de modelado
- Desarrollar el módulo de reporte y notificación
- Validar la solución mediante el método de prueba que más se adecúe a la solución.

Como resultado de todas las tareas y actividades desarrolladas se esperan los siguientes resultados:

- Artefactos resultantes de los flujos de trabajo que define la metodología utilizada.
- Documento de Trabajo de Diploma.

---

<sup>5</sup> Marco de trabajo

- Módulo de reportes y notificación.

En el transcurso de la investigación se han utilizado diferentes métodos científicos tales como:

- **Histórico - lógico:** Se basan en el estudio histórico del fenómeno, ponen de manifiesto la lógica interna de su desarrollo, de su teoría y hay un conocimiento más profundo de su esencia. Estos métodos expresan en forma teórica la esencia del objeto, explican la historia de su desarrollo, reproducen el objeto en su forma superior y permiten unir el estudio de la estructura del objeto de investigación con su concepción histórica (León, y otros, 2002). Se empleó para realizar un estudio analítico de la trayectoria histórica del objeto de estudio de la presente investigación, con el objetivo de conocer la evolución, el estado y las tendencias actuales de los sistemas generadores de reportes y notificación de eventos en diferentes entornos en el mundo, con los cuales se le podría dar solución al problema expuesto con anterioridad.
- **Analítico - Sintético:** Permite la división mental del fenómeno en sus relaciones y componentes para facilitar su estudio. Establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas (León, y otros, 2002). Este método se empleó para descomponer el problema de investigación en elementos por separado y de esta manera profundizar en el estudio de cada uno de ellos, para luego sintetizarlos para la confección de la solución propuesta. Es decir, se descompuso primeramente el problema en dos objetos a estudiar: la generación de reportes y la notificación de eventos, luego cada uno de estos por separado fue descompuesto en: primero investigar sobre los diferentes tipos de reportes que se pueden generar y posteriormente en las eventualidades que se necesiten notificar. Finalmente el resultado de todos estos elementos estudiados por separado se unen para dar una respuesta al problema en análisis.
- **Modelación:** Es el método mediante el cual se crean abstracciones con vistas a explicar la realidad. El modelo como sustituto del objeto de investigación. En el modelo se revela la unidad de lo objetivo y lo subjetivo. La modelación es el método que opera en forma práctica o teórica con un objeto, no en forma directa, sino utilizando cierto sistema intermedio, auxiliar, natural o artificial. La modelación que se utilizó fue la teórica que utiliza símbolos para designar las propiedades del sistema que se desea estudiar, representa las características y las relaciones fundamentales del objeto (Ramírez, 2003). Este método se emplea en el modelado de la solución que se hace mediante la herramienta Visual Paradigm utilizando UML como lenguaje de modelado.

El presente documento consta de tres capítulos, los cuales están estructurados de la siguiente manera:

**Capítulo 1:** Se expone una descripción detallada del objeto de estudio y se enuncian conceptos asociados que ayuden a un mejor entendimiento del mismo. Se caracterizan algunas soluciones existentes que ayuden a un mejor entendimiento de lo descrito en la situación problemática.

**Capítulo 2:** Se realiza un estudio de la metodología, el lenguaje y tecnologías que posteriormente se utilizan para el desarrollo de la aplicación. Se describe como se llevan a cabo los procesos de generación de reportes y notificaciones de eventos en la plataforma, PRIMICIA. Se definen los requerimientos funcionales y no funcionales que tendrá la solución propuesta, se representan los casos de usos y los diagramas de secuencias. También se muestran las clases del análisis y diseño, el modelado de datos y las clases persistentes en el sistema.

**Capítulo 3:** Se seleccionan los patrones a utilizar para el desarrollo del módulo tanto arquitectónico como de diseño. Se muestran los principales artefactos generados en la implementación. Abarca las validaciones aplicadas al sistema detallando las pruebas unitarias aplicadas mostrando los resultados obtenidos a partir de las mismas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SISTEMA DE REPORTES Y NOTIFICACIÓN DE EVENTOS.

## 1.1 Introducción

En el presente capítulo se explicarán un conjunto de conceptos fundamentales que se utilizarán a lo largo del documento para un mejor entendimiento de la investigación. Se abordarán las características del proyecto, PRIMICIA, para entender el ámbito en el cual será desarrollada la propuesta de solución. Se caracterizan algunas soluciones existentes para fundamentar lo descrito en la situación problemática y se expone una descripción detallada del objeto de estudio.

## 1.2 Conceptos Fundamentales. Sistema de reportes y notificación de eventos

### **Sistema:**

En informática, este término utilizado sin otra palabra que lo adjective designa un conjunto de hardware y software específico (Schmuller, 2000).

Conjunto de elementos interrelacionados y regidos por normas propias, de modo tal que pueden ser vistos y analizados como una totalidad. El sistema se organiza para producir determinados efectos, o para cumplir una o varias funciones (Schmuller, 2000).

Se llega a la conclusión de que un sistema es un conjunto de hardware, software y soporte humano interrelacionados que forman parte de una empresa u organización. Incluyen ordenadores, más los programas necesarios para procesar datos y las personas encargadas de su manejo.

### **Notificación:**

Comunicación oficial que hace una autoridad sobre una conclusión o determinación a la que ha llegado con relación a cierto tema (Dictionary, 2010).

Acción y efecto de notificar. Comunicación de un aviso o una noticia con propósito cierto. En el ámbito de la informática, las notificaciones aparecen relacionadas con alertas que emiten ciertos programas o servicios para advertir algo al usuario (Definicion, 2008).

Se concluye que una notificación es una alerta o aviso que hace el sistema sobre una acción que ocurrió en el mismo, la cual puede ser una advertencia sobre una anomalía que puede afectar su funcionamiento.

**Reporte:**

Un reporte es un informe o una noticia. Este tipo de documento (que puede ser impreso, digital, audiovisual, etc.) pretende transmitir una información, aunque puede tener diversos objetivos. Existen reportes divulgativos, persuasivos y de otros tipos (Definicion, 2008).

El reporte es aquel documento que se utilizará cuando se quiera informar o dar noticia acerca de una determinada cuestión (ABC, 2008).

Se concluye que los reportes son informes que organizan y exhiben la información y su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

**Evento:**

Conjunto de acciones que proceden de la ejecución o activación de otra acción. La combinación de estas acciones también puede dar como resultado un evento en particular o una serie de eventos (Schmuller, 2000).

Acontecimiento imprevisto o que puede acaecer aunque no existe seguridad sobre eso. Una eventualidad es algo que no estaba en los planes que sucediera (Definicion, 2008).

Se concluye además que un evento es una alerta acerca de la ocurrencia de una acción en el sistema que puede provocar un mal funcionamiento del mismo.

***Feedback*<sup>6</sup>:**

El término '*feedback*' proviene del inglés y podría ser traducido correctamente al castellano como 'retroalimentación', aunque de todos modos suele usarse en idioma inglés en la mayoría de los países de habla hispana (ABC, 2007).

El *feedback* o retroalimentación es el proceso mediante el cual se realiza un intercambio de datos, informaciones, hipótesis o teorías entre varias partes que intervienen en este proceso (ABC, 2007).

---

<sup>6</sup> Retroalimentación.



El *feedback* es además un proceso que puede darse de manera continua entre la(s) parte(s) que manda los datos y la(s) que los recibe, posiciones que pueden intercambiarse durante el proceso una y otra vez (ABC, 2007).

Se concluye que el *feedback* es el proceso que resulta de la conexión continua que se da entre las partes que intervienen, conexión que incluye y supone el traspaso de datos, información u otro tipo de elementos.

### **Módulo:**

Pieza o conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica (Dictionary, 2010).

Pieza que forma parte de un conjunto pero que también puede considerarse por separado (Dictionary, 2010).

Profundizando en el tema se concluye que un módulo es una parte de un sistema que suele estar conectado de alguna manera con el resto de los componentes del mismo y a través del cual se obtienen una serie de procedimientos o funciones capaces de asumir una gran diversidad de tareas para dar solución a un problema dado.

### **Sistema generador de reportes:**

Es un programa (rutina de computadora) que prepara un informe o exhibición de datos. Dicho programa selecciona en una base de datos la información necesaria para construir el informe y la formatea en renglones y columnas definidos por el usuario (Leon, 1993).

Los generadores de reportes son herramientas complementarias de los sistemas de información. Utilizan una especie de lenguaje transparente para el usuario por medio del cual éste realiza consultas a la base de datos y obtiene información de ella en forma de reporte (Leon, 1993).

Así mismo se llega a la conclusión de que un sistema generador de reportes es un conjunto organizado de información automatizada, en el que se puede encontrar los datos generales de un sistema computacional, se utiliza para agrupar la información que se desee de acuerdo a un interés en específico.

### **Sistema de notificación de eventos:**

Un Sistema de Notificación hace referencia a los procesos y la tecnología implicada en la estandarización, formato, comunicación, retroalimentación, análisis, aprendizaje, respuesta y difusión del aprendizaje

generado por el registro de eventos. Es la acción que se adopta tras el registro, la respuesta constructiva, y no el registro en sí lo que provoca los cambios y aporta valor (Seguridad, 2009).

Los sistemas de notificación de eventos son una parte de la “cultura de seguridad” donde se entienden los eventos adversos como una oportunidad para aprender y mejorar (Baneres, 2006).

Por lo que se llega a la conclusión de que un sistema de notificación de eventos es un conjunto organizado de información relacionada con las acciones que se realizan en una aplicación, es el encargado de mantener informados a los usuarios que interactúen con el sistema sobre las acciones que se efectúan.

### **1.3 Los procesos de monitoreo a través de la generación de reportes y sistemas de notificación de eventos**

#### **1.3.1 Descripción General del objeto de estudio**

Los sistemas informáticos son capaces de realizar diferentes procesos tales como generar reportes, notificar eventos además de brindar recomendaciones. Los sistemas de reportes permiten generar una serie de informes que pretenden transmitir una información solicitada, la cual organizan y exhiben, son datos relevantes guardados o generados por la misma aplicación de tal manera que se vuelvan útiles. El contenido de un reporte no se manipula o modifica directamente, sino que tiene que ser afectado en alguna otra parte del Sistema para que se reflejen los cambios una vez que sea generado nuevamente. Su función es aplicar un formato determinado a las informaciones que incluye para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (WebHelp, 2003).

Un sistema de notificación de eventos habilita las comunicaciones para responder ante eventos críticos como interrupciones en la producción, amenazas a la seguridad y desastres. Cuando ocurre un evento, se envían las alertas a grupos o individuos predeterminados, quienes pueden responder rápidamente con el estado de la situación. Los mensajes son soportados virtualmente en cualquier tipo de teléfono móvil, dispositivo para mensajes de texto y por correo electrónico (AVAYA, 2009). Para prepararse ante un evento o amenaza significativa, se emplean estos sistemas de notificaciones para definir procesos e identificar a las personas que se deben contactar durante la crisis, los cuales son notificados mediante el envío de un mensaje.

Uno de los principales retos que afrontan los sistemas de información es la gestión eficaz del gran volumen de documentos que almacenan, para poder facilitar a los consumidores de información el acceso a recursos que satisfagan sus necesidades de una manera sencilla y ágil. Esta necesidad se vuelve más acuciante en una sociedad como la actual, donde el nivel de exigencia de los usuarios es cada vez mayor. Por lo cual, surgen los sistemas de recomendaciones que no son más que un asistente virtual que es capaz de seleccionar aquellos productos o contenidos que se adaptan más a sus intereses (Martinez, 2009). Estas recomendaciones pueden ser mostradas por el sistema mediante formularios o a través de notificaciones, es aquí la relación existente entre los sistemas de recomendación y los sistemas de notificación.

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

En la actualidad es muy amplio el número de aplicaciones que incluyen un sistema de reportes, de notificación de eventos o ambas al mismo tiempo. Por lo que se realizó un estudio de diferentes aplicaciones existentes similares a la solución que se propone para tener una visión de la evolución que han tenido estos sistemas y apoyarnos en algunas características de estos que se adecuen a lo que se quiere lograr.

### **1.4.1 Ámbito internacional**

#### **Infogestión:**

Sistema de reportes estadísticos de la gestión del SIBUDEC (Sistema de Bibliotecas de la Universidad de Concepción), a partir de datos generados en el Sistema ALEPH.

Es una aplicación Web desarrollada en el año 2003 por la Dirección de Tecnologías de Información cuya principal característica es la generación dinámica y en línea de informes a partir de la definición de sentencias en Lenguaje de Consulta Estándar (SQL) sobre la base de datos corporativos de la Universidad de Concepción, también permite relacionar datos y armar reportes de acuerdo a la necesidad del operador. Durante el año 2004 la Dirección de Bibliotecas, plantea la necesidad de contar con un sistema de consulta y generación de reportes estadísticos de las actividades automatizadas desarrolladas en el software ALEPH, por lo que se toma entonces la decisión de emplear Infogestión para atender dicha necesidad. Este sistema se encuentra aún en desarrollo, pero en sus funciones básicas ha cumplido con las expectativas impuestas (Picon, y otros, 2007).

Entre sus objetivos principales se encuentran (Picon, y otros, 2007):

- Facilitar al personal del SIBUDEC la obtención de reportes estadísticos.
- Contar con un sistema de reportes actualizado diariamente.

En estos momentos Infogestión cuenta con 12 reportes entre los cuales se encuentran (Picon, y otros, 2007):

- Estadística general de préstamos del SIBUDEC.
- Estadística anual de préstamos por operador.
- Estadística anual de préstamos por tipo de material.
- Estadística anual de préstamos por estatus de ítem.
- Estadística anual de préstamos por sexo del usuario.
- Estadística anual de préstamos por clasificación Dewey.

No se puede usar para la solución del problema en análisis debido a que su implementación está dirigida directamente a la Universidad donde fue desarrollada y maneja la información solamente de su base de datos.

### **Generador de reportes GETrep:**

Es una herramienta que permite generar diferentes informes de tipo gerencial, bajo un diseño personalizado que el propio usuario puede definir y mantener de acuerdo a las necesidades particulares de cada empresa (SYNET, 2006).

Entre sus principales funciones se encuentran (SYNET, 2006):

- El usuario define el número de informes a generar así como sus formatos de salida y el nivel de detalle de los mismos.
- Otorga las herramientas necesarias para generar informes comparativos entre saldos de diferentes períodos (año/mes).
- Registro de ajustes por período, los cuales pueden reflejarse en los diferentes informes a definir.
- Almacena la información en MN (Moneda Nacional) y ME (Moneda Extranjera), permitiendo de esta manera informes en varias monedas.

- Genera informes financieros en los cuales se reflejan saldos consolidados de todas las compañías existentes en la base de datos.
- Gráfica de estadísticas definibles por el usuario.

Esta aplicación no es factible usarla debido a que maneja información que tiene que ver solamente con finanzas, saldos y costos en sistemas empresariales.

### **AdaptaPro:**

Es un generador de reportes que permite la creación de reportes escritos en código o léxico DpXbase de forma manual o auto-construido a través de un potente asistente visual que facilita la creación de reportes de manera instantánea con un mínimo de conocimientos y esfuerzo. A través de él se pueden crear desde simples listados generalizados, hasta complejos reportes especificando rangos y criterios de búsqueda que AdaptaPro realiza en pocos instantes (DataPro, 1987).

Es un potente generador de Reportes declarativo con un entorno de diseño basado en diccionario de datos que facilita la construcción de: sentencia SQL, enlaces entre tablas, rangos, criterios, grupos, totales y finalmente 100% del código fuente del reporte (DataPro, 1987).

Entre sus principales características se encuentran (DataPro, 1987):

- Permite generar en forma automáticas las sentencias SQL, necesarias para obtener las consultas de diversas tablas de la base de datos, requerido para lograr el modelo de consulta que será utilizado en la definición de cada columna que conforme el reporte.
- Luego de crearse el reporte se puede seleccionar la salida que se desea: vista previa (pantalla), impresión así como archivos de textos para el Sistema Operativo Windows.
- Permite crear un catálogo para la clasificación de reportes, facilitando la administración, ubicación y ejecución al estar en conjunto.

Es la que más se acerca al modelo deseado pero no es funcional en PRIMICIA porque es de licencia privativa y además podría haber incompatibilidad en las base de datos de ambas aplicaciones.

### **Redmine:**

Es una herramienta para la gestión de proyectos que incluye un sistema de seguimiento de incidentes con seguimiento de errores. Otras herramientas que incluye son calendario de actividades, diagramas de

Gantt para la representación visual de la línea del tiempo de los proyectos, foro, visor del repositorio de control de versiones, RSS, control de flujo de trabajo basado en roles e integración con correo electrónico. Además establece un sistema de notificaciones para los usuarios mediante correo electrónico ya sea porque se le ha asignado una tarea o porque una parte del proyecto ha cambiado o se ha actualizado. Está escrito usando el marco de trabajo *Ruby* con *Rails*. Es software libre y de código abierto, disponible bajo la Licencia Pública General de GNU v2. Este sistema cuenta con más de 100 tipos de reportes, los cuales una vez generados se pueden exportar a distintos tipos de formatos (*pdf*<sup>7</sup>, *csv*<sup>8</sup>, *excel*, *html*<sup>9</sup>). También estos muestran diferentes tipos de gráficos, ya sean de barras, pastel o de líneas (Cabeio, 2009). No es factible el uso directo de esta aplicación en la plataforma como sistema generador de reportes debido a que los reportes que genera están vinculados principalmente con la gestión del proyecto en sí de manera general y no con la gestión de los procesos y recursos que se administran en la misma.

#### 1.4.2 Ámbito nacional

##### **GDR (Gestor Dinámico de Reportes):**

El GDR ofrece una solución de reporte que le brinda a los negocios una herramienta reditual e inteligente, destinada a mejorar la rapidez y calidad de la adopción de decisiones en todos los niveles de una organización. Constituye, también, una herramienta que posibilita la inteligencia de negocios ya que genera vistas agregadas de datos para mantener a la gerencia informada sobre el estado de su negocio, además de que (Generador dinámico de reportes, 2012):

- Proporciona soporte a todo el ciclo de vida de los reportes.
- Provee accesibilidad a la información emitiendo reportes a través de un navegador Web u otros formatos estándares.
- Está orientada al usuario final, brindando herramientas como el Diseñador de reportes que permite la confección de informes de forma interactiva en la Web.

---

<sup>7</sup> formato de documento portátil.

<sup>8</sup> Valores Separados por Comas.

<sup>9</sup> Lenguaje de marcado de hipertexto.

### **Arquitectura y RIA (Aplicaciones Ricas de Internet):**

Su arquitectura flexible, modular y escalable permite a los programadores de software y a las empresas integrar el conjunto de herramientas con sus sistemas heredados, portales empresariales o aplicaciones personalizadas.

El sistema Generador Dinámico de Reportes constituye una aplicación RIA, puesto que combina las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales o de escritorio. Se evidencia en las características y potencialidades del Diseñador de reportes, que se extiende al máximo las posibilidades de diseño del usuario. Este diseñador permite personalizar al detalle la salida de los reportes y la forma en que la información será visualizada (Medina, y otros, 2012).

GDR en su versión 1.8, contempla funcionalidades como el Diseñador de modelos, reportes y consultas así como el Visor de reportes, que pueden ser insertadas al proceso de generación reportes de la Plataforma de Televisión Informativa, PRIMICIA sobre todo con aquellos reportes que se centran en la recuperación de información a través de un origen de datos. Sin embargo no todos los reportes que se conciben en la plataforma pueden ser generados a través de este sistema, debido a que el origen de datos de GDR siempre es un gestor de base de datos y en la plataforma no todas las fuentes de información provienen de una base de datos. Además cabe destacar que GDR no contempla un mecanismo de notificación a partir de los resultados que hace mostrar.

### **1.5 Descripción actual del dominio del problema**

En la UCI se encuentra el centro GEYSED con el objetivo de desarrollar productos, servicios y soluciones informáticas en el campo del procesamiento de Señales Digitales y en el mundo de la Geo-ciencia, permitiendo un posicionamiento en el mercado nacional y latinoamericano. Este centro cuenta con el Departamento de Señales Digitales el cual es un colectivo formado por profesores y estudiantes de la UCI, que se encargan del desarrollo de productos, sistemas, servicios y soluciones informáticas en el campo del procesamiento digital de imágenes y señales. Uno de sus objetivos lo constituye el desarrollo de componentes de alto valor agregado relacionados con el tema de procesamiento digital de imágenes y señales para ser comercializados con productos de otros centros. Además promueve la implantación de aplicaciones especializadas en diversos sectores bajo plataforma de software libre.

El departamento orienta las investigaciones en tres líneas fundamentales que tributan a la formación de pregrado y postgrado, además de guiar el desarrollo de los productos:

- Sistemas de Transmisión de Señales.
- Recuperación, Gestión y Procesamiento de Medias.
- Procesamiento digital de imágenes y señales.

Dentro de los proyectos que se desarrollan en este departamento se encuentra la Plataforma de Televisión Informativa, PRIMICIA. Dicha plataforma es abordada en la investigación en curso. Esta plataforma en su etapa de Gestión de Cambios y Configuración realiza la mejora y configuración del producto para lograr una mayor automatización.

## 1.6 Características de la Plataforma de Televisión Informativa, PRIMICIA

PRIMICIA es un producto informático que se ha venido desarrollando desde el año 2005 por la UCI, ha sido desarrollado y es soportado completamente en software libre. Se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios, estos son:

*Tabla 1: Subsistemas de la plataforma PRIMICIA*

<b>Subsistemas</b>	<b>Tipo de aplicación</b>
Transmisión	Escritorio
Administración	Web

Esta plataforma está desarrollada en dos entornos al mismo tiempo, web y escritorio. En un principio se desarrolló una versión para la AIN (Agencia de Información Nacional) que incluyó una pequeña funcionalidad para generar algunos reportes, pues existía la necesidad de mantener controlada un grupo de información como eran: las noticias (redactadas, corregidas y publicadas), los usuarios (administradores, redactores y correctores), las medias (imagen, audio y video) y las secciones temáticas. Esta funcionalidad fue suprimida de las versiones próximas debido al negocio que se enfrentaban donde no era considerada. Actualmente se retoma la idea, no como una funcionalidad sino como un módulo que se integrará a la plataforma no solo para generar reportes de los procesos y recursos que se administran en la misma sino también para notificar eventos y sugerir algunas recomendaciones ante determinadas situaciones que se presenten durante la administración de dicha plataforma.



## 1.7 Conclusiones parciales

Después de un estudio de los principales conceptos asociados a la investigación así como de algunas aplicaciones similares existentes para una mejor comprensión del problema en cuestión se concluye que:

- A través de una descripción de los procesos de generación de reportes y sistemas de notificación de eventos se logró tener una mejor visión de lo que se quiere alcanzar con el módulo de reportes y notificaciones de eventos.
- A través del estudio de otras soluciones existentes similares a la que se quiere, las cuales poseen características que se pueden tener en cuenta a la hora de implementar el módulo de reportes y notificaciones de eventos se logró una mejor comprensión del problema en cuestión.
- La incorporación a la plataforma de un módulo de reportes y notificaciones de eventos la dotará de una serie de funcionalidades que la harían capaz de permitir la recuperación y obtención de información deseada, así como la notificación de eventos que ocurren en el sistema.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### **2.1 Introducción**

En el presente capítulo se explicarán las herramientas, lenguaje de programación, metodología de desarrollo de software y tecnologías fundamentales que se utilizarán a lo largo de la implementación de la aplicación. Se describe como se llevan a cabo los procesos de generación de reportes y notificaciones de eventos en la plataforma, PRIMICIA. Se definen los requisitos funcionales y no funcionales que tendrá la solución propuesta, se representan los casos de usos así como se muestran las clases del análisis y diseño, el modelado de datos y las clases persistentes en el sistema.

### **2.2 Metodología, lenguajes de programación, herramientas y tecnologías**

#### **2.2.1 Metodología de Desarrollo de Software**

##### **RUP**

Proceso Unificado de Desarrollo es un proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de software. Su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales. Describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura (Jacobson, y otros, 2000).

Esta metodología se escoge para el desarrollo del Módulo de reportes y notificaciones de eventos porque permite ser utilizada en proyectos donde las personas no cuentan con experiencia en la producción de software, no depende de la presencia constante del cliente, permite el desarrollo de proyectos a largo plazo y cuenta con una documentación detallada para un mejor entendimiento y trabajo con la misma.

#### **2.2.2 Lenguaje Unificado de Modelado**

Se ha seleccionado UML como lenguaje para la modelación de la solución propuesta, debido a que sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML tiene una notación gráfica muy expresiva que permite

representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases y objetos, hasta la implementación y configuración con los diagramas de despliegue (Larman, 2004).

Se escoge UML para el modelado del Módulo de reportes y notificación de eventos porque brinda la posibilidad a los analistas de definir las características principales del sistema, describe las decisiones y todos los pasos necesarios representados gráficamente por modelos precisos y completos que proporcionan una mejor comprensión para los desarrolladores.

### **2.2.3 Lenguaje PHP**

Es un lenguaje de interpretado de alto nivel, habitualmente se usa para el desarrollo de aplicaciones web. El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO<sup>10</sup> para solucionar las carencias de las anteriores versiones (Angelismo, 2006).

Se escoge PHP5 como lenguaje de programación del lado del servidor porque es multiplataforma, de código libre por lo que es una alternativa de fácil acceso para todos y posee una amplia documentación, permite las técnicas de Programación Orientada a Objetos, consta de una biblioteca nativa de funciones y no requiere definición de tipos de variables. También este lenguaje proporciona una fácil integración del acceso a bases de datos en páginas HTML. Estas características lo hacen funcional para el desarrollo de la solución propuesta para resolver problema en análisis.

### **2.2.4 Lenguaje JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa *Sun Microsystems* (Valdelli, 2006).

Se escoge JavaScript como lenguaje de programación del lado del cliente porque es dinámico, responde a eventos en tiempo real, se consigue la validación de la información como son los datos y permite crear menús interactivos, además de ser un lenguaje sencillo de aprender. Además vale destacar que se ha

---

<sup>10</sup> Programación Orientada a Objetos.

empleado en las versiones anteriores del producto, una razón más para fundamentar su uso en la nueva versión.

### **2.2.5 Visual Paradigm como herramienta CASE**

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios (Carrera, 2007).

Es una herramienta que emplea las últimas notaciones del Lenguaje Unificado de Modelado, ingeniería inversa, generación del código, importación de *Rational Rose*, exportación/importación XML e integración con el Visio. Además soporta aplicaciones web, genera código para el lenguaje Java y exporta en formato HTML, es una tecnología libre y está disponible en varios idiomas, en conjunto con esto, es fácil de instalar y fácil de actualizar. Por último, admite compatibilidad con las demás versiones (FILEHEAVEN 2008).

Se escoge Visual Paradigm pues además de emplear UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre, permite crear diferentes tipos de diagramas es muy sencillo de usar, fácil de instalar y actualizar. Es muy poderosa pues permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requerimientos y de base de datos. Se adecua al entorno de desarrollo permitiendo una mayor organización y claridad en el trabajo.

### **2.2.6 Symfony 2.0 como marco de trabajo de desarrollo**

Symfony es un marco de trabajo PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto. Symfony2 es la versión más reciente de Symfony. Se anunció por primera vez a principios de 2009 y supone un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Se selecciona como marco de trabajo para el desarrollo del módulo ya que ha sido ideado para aprovechar al límite todas las nuevas características de PHP 5.3 y por eso es uno de los marcos de trabajo PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en un proyecto (Hoplite, 2005).

Se escoge Symfony2 como marco de trabajo debido a que es independiente del sistema gestor de bases de datos, emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación Web, además que cuenta con una amplia documentación traducida al español. Por las razones expuestas anteriormente se considera que es la mejor opción para el desarrollo del módulo ya que la calidad que brinda al código fuente y la cantidad de documentación disponible es favorable para desarrollar el mismo.

### **2.2.7 JQuery como biblioteca Javascript**

Se escoge JQuery como biblioteca JavaScript para crear una presentación estéticamente aceptable y amigable para los usuarios porque permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Lycanthrop, 2007).

### **2.2.8 Netbeans como IDE<sup>11</sup>**

NetBeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Elimina el uso de los comandos de consola que se necesitaban para instalar Symfony, esto se debe a que presenta una interfaz visual para desarrollar dicha instalación. Posee un fuerte editor de código fuente, sobre todo el de PHP, ya que provee todas las ventajas que presenta PHP 5.3, además de ser robusto y ágil. NetBeans es gratuito y de código abierto (NetBeans, 2009).

Se escoge como Entorno Integrado de Desarrollo para el desarrollo del módulo porque tiene soporte para crear interfaces gráficas de forma visual y desarrollar aplicaciones web, además estas funcionalidades pueden ser ampliables mediante instalación de paquetes adicionales. Presenta una fácil integración con Symfony que agiliza la elaboración de proyectos.

---

<sup>11</sup> Entorno Integrado de desarrollo.

### **2.2.9 PostgreSQL<sup>12</sup>**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto potente y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas la hacen una base de datos robusta y potente. Este gestor funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (Martinez, 2009).

La utilización de este SGBD en la Plataforma de Televisión Informativa, PRIMICIA obedece al propósito de la elaboración de un sistema robusto y alto nivel de escalabilidad. Trabajar con este gestor es factible debido a que es muy rápido, tiene buenas utilidades de administración, es confiable y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. La conectividad, velocidad y seguridad, lo hacen altamente conveniente para acceder a bases de datos en internet porque tiene mejor control de acceso a usuarios. Además los lenguajes que se proponen para el desarrollo del sistema logran una rica interacción con este gestor lo que resulta compatible y beneficioso a la hora de diseñar de forma rápida y eficaz aplicaciones dirigidas a bases de datos.

## **2.3 Proceso de generación de reportes y notificación de eventos en Primicia-v2.0**

Las noticias redactadas, la relación de carga de noticias por secciones temáticas y la relación de noticias redactadas por un redactor específico, así como una especificación de las noticias que estén usando un audio determinado, son factores que influyen en la Plataforma de Televisión Informativa, PRIMICIA a la hora de tener un control de los recursos que se administran. En la visualización de las noticias se emplea un audio, cuya música tiene un autor, al cual se le debe pagar el derecho de autor por parte de la empresa o entidad que está haciendo uso del producto, por lo que se tiene una cuota de cantidad de veces a utilizar un audio por noticias y así saber el costo del mismo.

---

<sup>12</sup> Sistema gestor de base de datos.

Cuando las prestaciones del servidor estén al límite se les pueden enviar, mediante notificaciones, algunas recomendaciones a los encargados de la administración de cómo pueden dar solución a esta situación, así como sugerir otro audio cuando el que se está empleando llegue al límite de uso. Otra forma de controlar los recursos y proceso en la plataforma es teniendo en cuenta algunos elementos de seguridad como son los usuarios, grupos de usuarios y sus permisos; a partir de la selección de la opción usuarios o grupos de usuarios se muestran los permisos concedidos a estos, al igual que si se selecciona un permiso se mostraran los usuarios y grupos que poseen el mismo. A partir del archivo trazas del sistema se muestran las acciones que han realizado los usuarios en la aplicación, a través del cual se puede tener un control de la seguridad de la misma.

A partir de la generación de reportes sobre los datos de los recursos y procesos que se producen en la plataforma se procede a la notificación del estado de los mismos, incorporando alguna recomendación sobre lo que se debe realizar. Esto se logra enviando un mensaje a los responsables de la administración sobre lo que está sucediendo y estos serán los encargados de responder ante esta situación.

## **2.4 Modelo del dominio**

**PRIMICIA** es una Plataforma de Televisión Informativa compuesta por dos sub-sistemas, el de **Administración** y el de **Transmisión**. En la **Administración** son gestionados todos los recursos con que cuenta la plataforma para la transmisión de noticias, para lo cual se realizan peticiones a la **Base de Datos** donde se encuentran almacenados dichos recursos. El sub-sistema de **Transmisión** es el encargado de transmitir el bloque de noticias programado en la administración y registra en la base de datos los errores ocurridos durante la transmisión del canal.

### 2.4.1 Diagrama de Clases del Modelo del Dominio

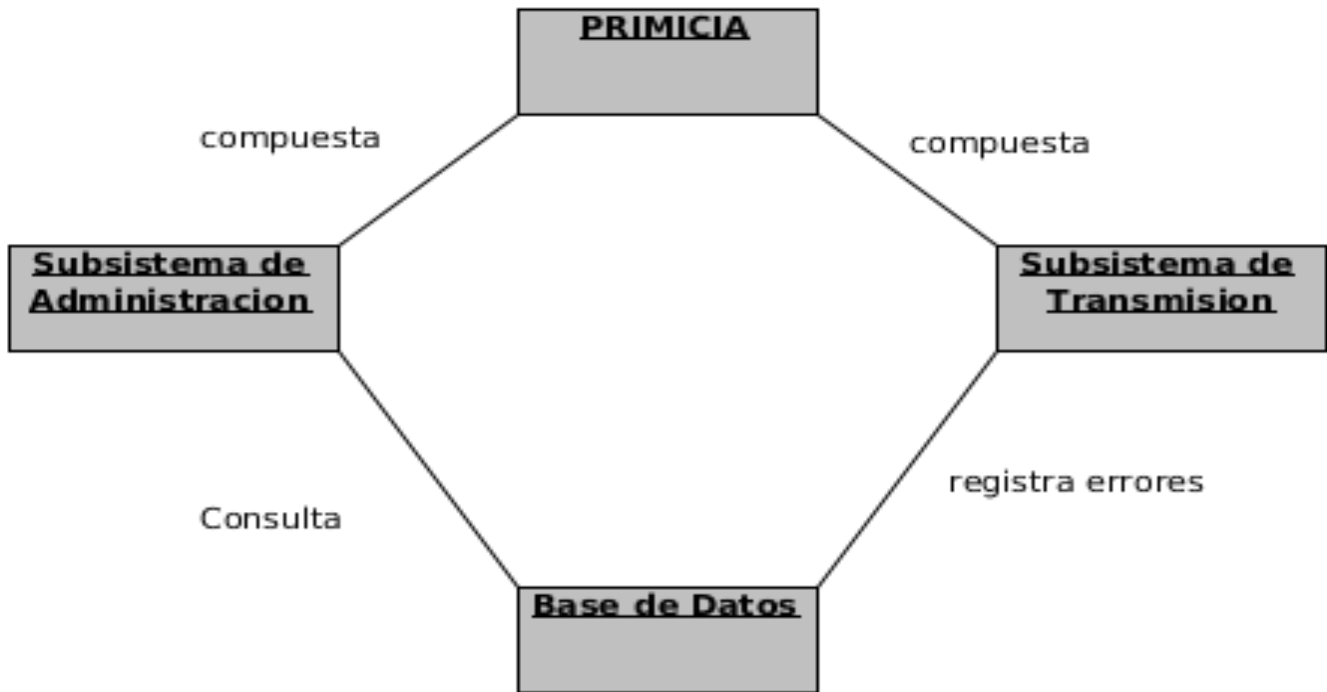


Figura 1. Diagrama de Clases del Modelo del Dominio.

### 2.4.2 Descripción de las clases del dominio

**Subsistema de Administración:** Es el encargado de la administración del canal y de gestionar todas las noticias, infocintas, secciones y señales, así como los reportes.

**Subsistema de Transmisión:** Realiza peticiones a la base de datos para transmitir las señales y es el encargado de la visualización de las informaciones asociadas a las noticias.

**Base de Datos:** En ella van a estar almacenadas la información asociada a las señales, noticias, infocintas y secciones posibles a transmitir, así como los errores de transmisión ocurridos durante la misma.

## 2.5 Requisitos del sistema

Los requerimientos del sistema son los que permiten que la aplicación cumpla con las exigencias del cliente, estos se dividen en funcionales y no funcionales (Mirabal, 2012). A continuación se enuncian los requisitos funcionales que debe cumplir el módulo de reportes y notificaciones de eventos:



### **2.5.1 Requisitos funcionales**

#### **RF1. Generar reporte de infocintas según un rango de fechas preseleccionado:**

Aquí se mostrará un listado de las infocintas que fueron transmitidas en un intervalo de tiempo según una fecha inicio y una fecha fin determinada.

#### **RF2. Generar reporte según la sección temática seleccionada a partir de un rango de fechas preseleccionado:**

Brindará la opción de seleccionar la temática de la que se quiere generar el reporte, una vez seleccionada mostrará un listado con las noticias de dicha temática publicadas en el intervalo de tiempo según una fecha inicio y una fecha fin determinada.

#### **RF3. Generar reporte de medias según un rango de fechas preseleccionado:**

Brindará la opción de seleccionar la media de la que se quiere el reporte (audio, video, imagen), una vez seleccionada se mostrará un listado con las noticias publicadas en un intervalo de tiempo según una fecha inicio y una fecha fin determinada que utilizan dicha media.

#### **RF4. Generar reporte de noticias a partir de un rango de fechas preseleccionado:**

Brindará la opción de seleccionar el estado de la noticia (redactada, corregida, publicada) del que se quiere el reporte, una vez seleccionado este, se mostrará un listado con las noticias que se encuentren en dicho estado preseleccionado a partir de un intervalo de tiempo según una fecha inicio y una fecha fin determinada.

#### **RF5. Generar reporte en forma de gráfico:**

Brindará la opción de mostrar algunos reportes en formas de gráficos los cuales pueden ser de barra.

#### **RF6. Generar reporte de notificaciones a partir de un rango de fechas preseleccionado:**

Se mostrará un listado con todas las notificaciones que se realizaron en un intervalo de tiempo según una fecha inicio y una fecha fin determinada.

**RF7. Generar reporte de señales a partir de un rango de fechas preseleccionado:**

Brindará la opción de seleccionar la señal de la cual se quiere el reporte, una vez seleccionada, se mostrará un listado con la programación transmitida en dicha señal en un intervalo de tiempo según una fecha inicio y una fecha fin determinada.

**RF8. Generar reporte a partir del análisis de las trazas:**

Muestra las acciones que se realizan en la aplicación en el entorno de producción.

**RF9. Exportar el reporte a formato pdf:**

Permitirá exportar el reporte a un documento con formato pdf.

**RF10. Notificar la existencia de medias obsoletas:**

Se enviará un mensaje a los administradores del sistema informándoles la existencia de medias que se encuentren obsoletas, la cuales pueden ser eliminadas.

**RF11. Notificar la existencia de noticias obsoletas:**

Se enviará un mensaje a los administradores del sistema informándoles la existencia de noticias que se encuentren obsoletas, la cuales pueden ser eliminadas.

**2.5.2 Requisitos no funcionales:**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable. Tienen que ver con las características que de una u otra forma puedan limitar el sistema. Describen una restricción sobre el sistema que limita nuestra elección en la construcción de una solución (Mirabal, 2012).

**RNF1. Seguridad:**

El componente no presentará un sistema de autenticación debido a que el mismo se integrará a la plataforma PRIMICIAv2.0 la cual ya incluye uno. Al componente tendrán acceso sólo las personas autorizadas para hacer uso del mismo.

**RNF2. Soporte:**

Se utilizará como sistema gestor de Base de Datos PostgreSQL y como servidor web Apache para el trabajo con PHP y algunas otras herramientas necesarias.

**RNF3. Usabilidad:**

El módulo podrá ser usado por cualquier persona que posea un conocimiento básico de computación.

**RNF4. Software:**

Tanto las estaciones de trabajo como el servidor deben tener instalados sistemas operativos Linux que reciban soporte por la universidad.

**RNF5. Requisitos de restricciones en el diseño y la implementación:**

- Lenguajes de programación JavaScript para el lado del cliente y PHP5 para el lado del servidor.
- Como Entorno Integrado de Desarrollo (IDE) NetBeans.
- Como herramienta CASE Visual Paradigm.

**RNF6. Requisitos de apariencia e interfaz externa:**

El sistema deberá contar de manera general, con una interfaz sencilla, que sea intuitiva y de fácil comprensión garantizando la correcta interacción con el sistema.

## **2.6 Descripción de la solución propuesta**

Después de haber realizado el levantamiento de requisitos del sistema y tener en cuenta varias restricciones, se propone el desarrollo de un módulo que genere reportes de los recursos y procesos que se administran en la plataforma PRIMICIA, además de notificar el estado de estos. Dicho módulo dará solución al problema de monitoreo de los recursos y procesos que se administran en la plataforma, manteniendo informados por vía notificación a los administradores de los mismos.

### **2.6.1 Generación de reportes en la Plataforma de Televisión Informativa, PRIMICIA**

Los reportes en la Plataforma de Televisión Informativa, PRIMICIA se realizarán a partir de consultas realizadas a la base de datos que posee dicha plataforma, en la cual se encuentra toda la información de los recursos que se administran en la misma. Los recursos de los que se obtendrán los reportes son usuarios, medias, noticias, secciones temáticas, notificaciones, infocintas, señales y trazas. En el caso del reporte de usuarios, se mostrarán todas las noticias en las cuales ha trabajado un usuario específico en un determinado intervalo de tiempo. En el reporte de medias se mostrarán todas las noticias que estén utilizando la media determinada. Para el reporte de noticias se determinará el intervalo de tiempo d el que

se quiere el reporte y el estado de la noticia (redactada, corregida, publicada o archivada), mostrándose un listado con todas las noticias que se encuentren en el estado seleccionado. Otro de los reportes es el de las secciones temáticas, el cual mostrará un listado de las noticias que pertenecen a dicha temática. También está el reporte de notificaciones donde se mostrarán todas las notificaciones realizadas en un intervalo de tiempo predeterminado. Otro de los reportes que se generará es el de infocintas, donde según un intervalo de fechas se mostrará un listado con las infocintas que se transmitieron en ese periodo. En el reporte de señales, se mostrará la programación asociada a dicha señal en el intervalo en el intervalo de tiempo que se indique, y por último el reporte a partir del análisis de las trazas aquí se mostrará las acciones realizadas en la aplicación solo en el entorno de producción. Una vez generado uno de estos reportes se pueden exportar a un documento con formato pdf o seleccionar la opción de imprimir, algunos de estos reportes tendrán incluidos la opción de ser mostrados en forma de gráfico de barras como es el caso del reporte de trazas.

### **2.6.2 Notificaciones en la Plataforma de Televisión Informativa, PRIMICIA**

En la Plataforma de Televisión Informativa, PRIMICIA una de las vías para mantener a los responsables de su administración informados es mediante la notificación. Se realizarán notificaciones cuando se genera un reporte, cuando un audio está al límite de uso o cuando se encuentran noticias o medias obsoletas. Dentro de las notificaciones se enviarán algunas recomendaciones, para lo que se definieron algunas reglas de recomendación. Una de las reglas de recomendación es cuando un audio se pasa del límite de uso, este deja de ser usado y se le muestra al usuario, vía notificación, un listado de los audios disponibles facilitando la selección de uno de ellos. Otra de las reglas es cuando se encuentran noticias o medias obsoletas, se envía una notificación recomendando la eliminación de estas y así aprovechar el espacio libre para otros recursos.

## 2.7 Casos de uso del sistema

### 2.7.1 Diagrama de Casos de Uso del Sistema

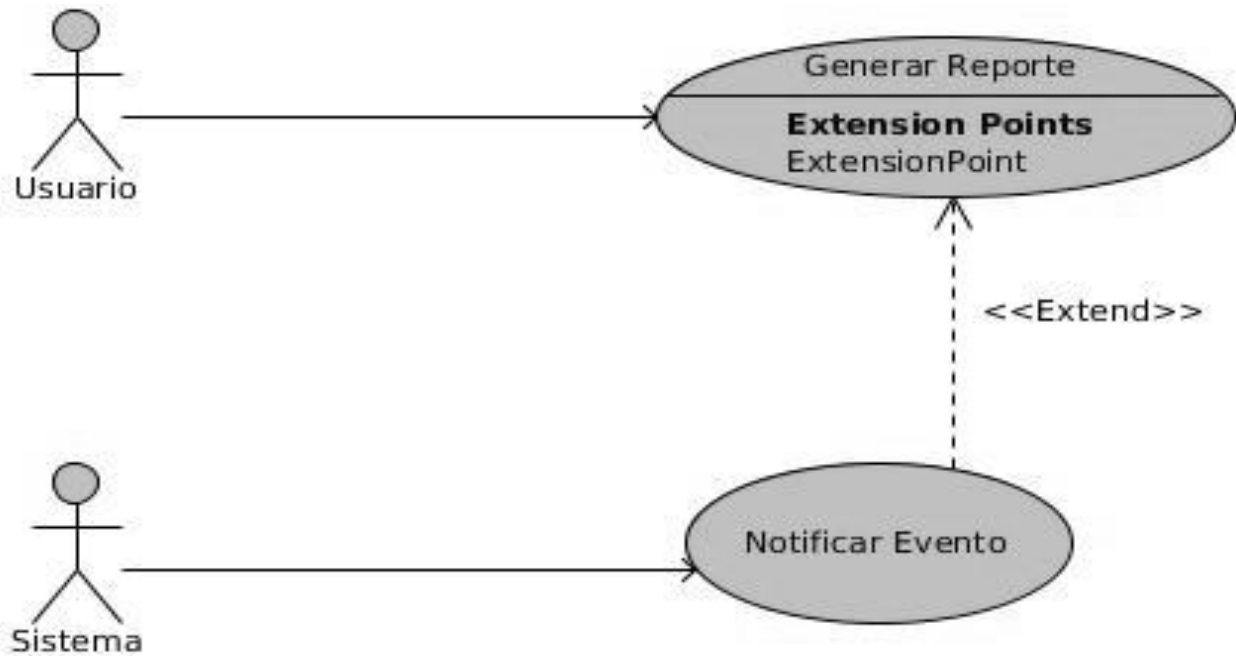


Figura 2. Diagrama de Casos de Uso del Sistema.

### 2.7.2 Caso de uso Generar Reporte

Tabla 2. Descripción del caso de uso Generar reportes.

<b>Objetivo</b>	Generar el reporte deseado por el usuario que se encuentre interactuando con el sistema.
<b>Actores</b>	Usuario: (Inicia) Seleccionando el reporte que desea que genere el sistema.
<b>Resumen</b>	El caso de uso inicia cuando el usuario selecciona en el menú principal la opción del reporte que desee obtener.

<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Usuario autenticado. Sistema PRIMICIA disponible.	
<b>Pos condiciones</b>	Reporte generado según la operación solicitada.	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Generar reporte.		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona en el menú principal la opción Reporte.	
2.		Muestra las opciones de los reportes.
3.	Selecciona el reporte que desea que genere el sistema.  Infocintas. Ver Sección 1.  Temático. Ver Sección 2.  Medias. Ver Sección 3.  Noticias. Ver Sección 4.  Notificaciones. Ver Sección 5.  Señales. Ver Sección 6.  Trazas. Ver Sección 7.	
4.		El sistema muestra las acciones que

		<p>el usuario puede realizar una vez generado el reporte.</p> <p>Exportar el reporte a pdf. Ver sección 8.</p> <p>Mostrar reporte en forma de gráfico. Ver sección 9.</p>
5.	El usuario selecciona la acción que desea realizar.	
6.		El sistema realiza la acción seleccionada por el usuario. Finaliza el caso de uso.

### Flujos alternos

	<b>Actor</b>	<b>Sistema</b>

### Sección 1: “Reporte de Infocintas”

#### Flujo básico: Reporte de Infocintas

	<b>Actor</b>	<b>Sistema</b>
3.		Muestra la página del reporte de infocintas.
4.	Introduce el intervalo de fechas de uso de las infocintas.	

5.		Muestra las infocintas usadas en ese intervalo de tiempo. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		
4a. Las fechas introducidas no son correctas.		
	<b>Actor</b>	<b>Sistema</b>
4a.	Introduce fechas incorrectas.	Muestra un mensaje de que las fechas introducidas están incorrectas. Vuelve al paso 3 del flujo básico: Reporte de Infocintas.
<b>Sección 2: “Reporte Temático”</b>		
<b>Flujo básico: Reporte Temático</b>		
	<b>Actor</b>	<b>Sistema</b>
3.		Muestra la página del reporte temático.
4.	Selecciona la temática de las noticias e introduce el intervalo de fechas que desee para el reporte.	
5.		Muestra un listado con las noticias, pertenecientes a la temática seleccionada, publicadas en el intervalo de tiempo preseleccionado. Regresa al paso 4 del flujo básico:



		Generar reporte.
<b>Flujos alternos</b>		
4a. Datos introducidos incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
4a.	Introduce fechas incorrectas.	Muestra un mensaje de que las fechas introducidas están incorrectas. Regresa al paso 3 del flujo básico: Reporte Temático.
<b>Sección 3: "Reporte de medias"</b>		
<b>Flujo básico: Reporte de medias</b>		
	<b>Actor</b>	<b>Sistema</b>
3.		Muestra la página del reporte de medias.
4.	Selecciona la media de la cual desea se muestre el reporte e introduce el intervalo de fechas en que fue empleada la misma.	
5.		Muestra un listado con las noticias que utilizan la media seleccionada en el intervalo de tiempo preseleccionado. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		

<b>4a. Datos introducidos incorrectos.</b>		
	<b>Actor</b>	<b>Sistema</b>
4a.	Introduce fechas incorrectas.	Muestra un mensaje de que las fechas introducidas están incorrectas. Regresa al paso 3 del flujo básico: Reporte de medias.
<b>Sección 4: "Reporte de noticias"</b>		
<b>Flujo básico: Reporte de noticias</b>		
	<b>Actor</b>	<b>Sistema</b>
3.		Muestra la página del reporte de noticias.
4.	Selecciona el tipo de noticia que dese se muestren e introduce el intervalo de fechas de las mismas.	
5.		Muestra un listado con los datos de las noticias según el tipo seleccionado en ese intervalo de tiempo. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		
<b>4a. Datos introducidos incorrectos.</b>		
	<b>Actor</b>	<b>Sistema</b>

4a.	Fechas introducidas incorrectas.	Muestra un mensaje de que las fechas introducidas están incorrectas. Regresa al paso 3 del flujo básico: Reporte de noticias.
<b>Sección 5: “Reporte de notificaciones”</b>		
<b>Flujo básico: Reporte de notificaciones</b>		
	<b>Actor</b>	<b>Sistema</b>
3.		Muestra la página del reporte de notificaciones.
4.	Introduce el intervalo de fechas de las notificaciones.	
5.		Muestra un listado con los datos de las notificaciones que se realizaron en ese intervalo de tiempo preseleccionado. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		
<b>4a. Datos introducidos incorrectos.</b>		
	<b>Actor</b>	<b>Sistema</b>
4a.	Fechas introducidas incorrectas.	Muestra un mensaje de que las fechas introducidas están incorrectas. Regresa al paso 3 del flujo básico: Reporte de notificaciones.

<b>Sección 6: “Reporte de señales”</b>		
<b>Flujo básico:</b> Reporte de señales		
3.		Muestra la página del reporte de señales.
4.	Selecciona la señal que desee se muestre su programación.	
5.		Muestra la programación de la señal seleccionada. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		
<b>4a.</b> Datos introducidos incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
	Introduce una señal incorrecta.	Muestra un mensaje de que la señal introducida es incorrecta. Regresa al paso 3 del flujo básico: Reporte de señales.
<b>Sección 7: “Reporte de trazas”</b>		
<b>Flujo básico:</b> Reporte de trazas		
3.		Muestra la página del reporte de trazas.
4.	Introduce el intervalo de tiempo.	

5		Muestra un listado con las acciones que se han realizado en la aplicación en el entorno de producción. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Flujos alternos</b>		
4a. Datos introducidos incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
	Introduce fechas incorrectas.	Muestra un mensaje de fechas incorrectas. Regresa al paso 3 del flujo básico: Reporte de trazas.
<b>Sección 8: “Exportar el reporte a pdf.”</b>		
<b>Flujo básico:</b> Exportar el reporte a pdf.		
3.	Selecciona exportar el reporte a pdf.	
4.		Exporta el reporte a pdf. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Sección 9: “Mostrar el reporte en forma de gráfico”</b>		
<b>Flujo básico:</b> Mostrar el reporte en forma de gráfico		
3.	Selecciona la opción mostrar el reporte en forma de gráfico.	
4.		Muestra los datos del reporte en

		forma de gráfico. Regresa al paso 4 del flujo básico: Generar reporte.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
<b>Relaciones</b>	<b>CU Extendidos</b>	Notificar evento. Ver CU "Notificar evento".
<b>Requisitos funcionales</b>	<b>no</b>	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6.
<b>Asuntos pendientes</b>		No procede.

### 2.7.2.1 Diagrama de clases del análisis para el caso uso Generar Reporte

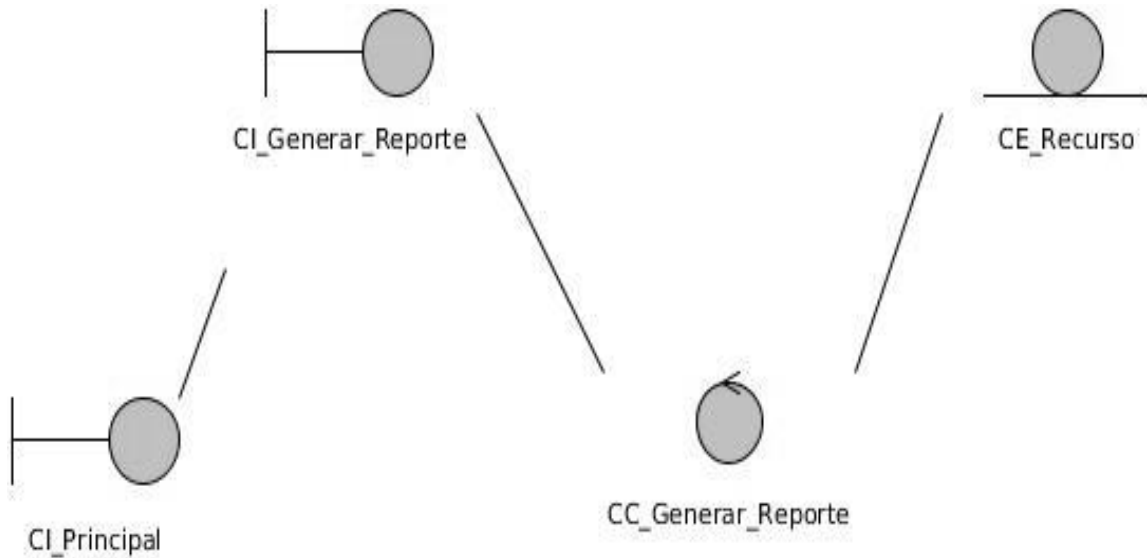


Figura 3. Diagrama de Clases del Análisis Caso de Uso Generar Reporte.

### 2.7.3 Caso de Uso Notificar Eventos

*Tabla 3. Descripción del caso de uso Notificar eventos.*

<b>Objetivo</b>	Este caso de uso tiene como objetivo enviar mensajes a los encargados de la administración sobre irregularidades encontradas en los reportes realizados y en los procesos ocurridos en el sistema.	
<b>Actores</b>	Sistema PRIMICIA: (Inicia) Notificación de eventualidades de la plataforma.	
<b>Resumen</b>	El caso de uso inicia cuando el sistema encuentra alguna irregularidad en los reportes generados o en los procesos que ocurren en la plataforma.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Ocurrencia de un error. Sistema PRIMICIA disponible.	
<b>Pos condiciones</b>	Notificación de errores o eventos ocurridos en el sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Notificar eventos.		
	<b>Actor</b>	<b>Sistema</b>
1.		Envía un mensaje, por correo electrónico, a los responsables de la administración cada vez que sea generado un reporte.

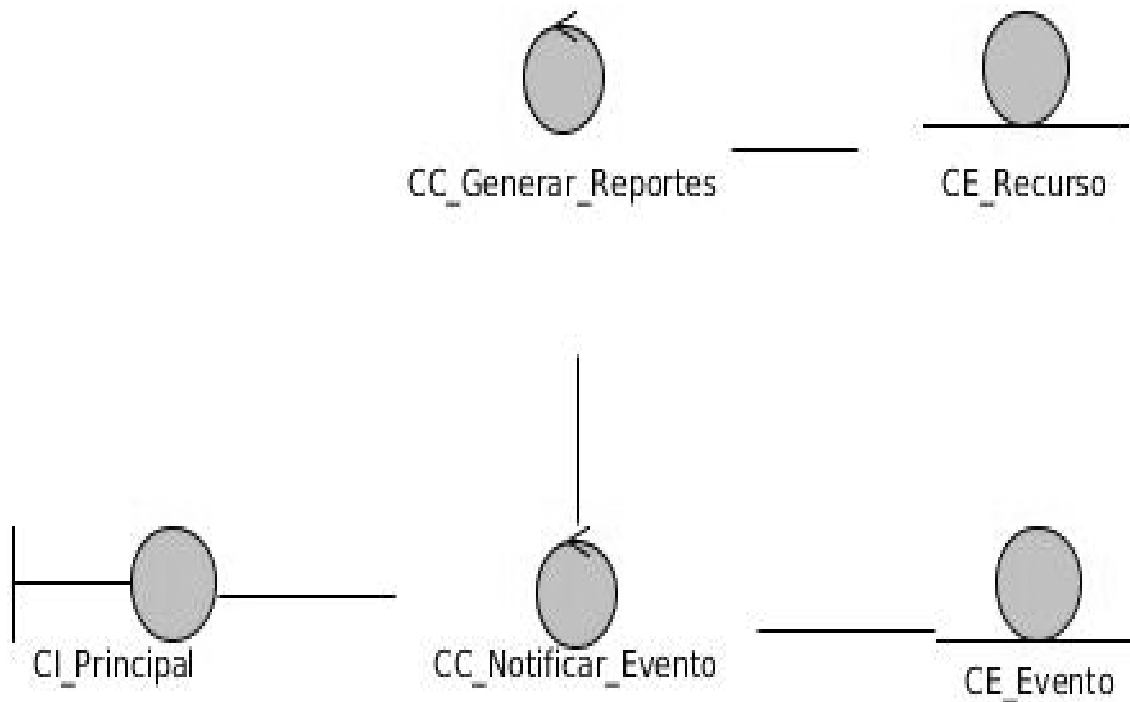
2.		<p>Verifica los datos de los recursos que puedan dar lugar a la notificación de eventos (medias o noticias obsoletas).</p> <p><b>Sección 1:</b> Notificar la existencia de medias obsoletas.</p> <p><b>Sección 2:</b> Notificar existencia de noticias obsoletas.</p>
3.	Resuelve el problema notificado.	
4.		Finaliza el caso de uso.
<b>Flujos alternos</b>		
1a. El sistema no encuentra ninguna irregularidad.		
	<b>Actor</b>	<b>Sistema</b>
1a.		Regresa al paso 4 del flujo básico de eventos: Notificar eventos.
<b>Sección 1:</b> “Notificar la existencia de medias obsoletas.”		
<b>Flujo básico:</b> Notificar la existencia de medias obsoletas.		
3.		Envía un mensaje notificando la existencia de medias obsoletas, y recomendando su eliminación.
4.	Elimina las medias obsoletas.	
5.		Regresa el paso 4 del flujo básico de



		eventos: Notificar eventos.
<b>Flujos alternos</b>		
1a.El sistema no encuentra medias obsoletas.		
	<b>Actor</b>	<b>Sistema</b>
		Regresa al paso 4 del flujo básico de eventos: Notificar eventos.
<b>Sección 2: “Notificar la existencia de noticias obsoletas”</b>		
<b>Flujo básico:</b> Notificar la existencia de noticias obsoletas		
3.		Envía un mensaje notificando la existencia de noticias obsoletas, y recomendando su eliminación.
4.	Elimina las noticias obsoletas.	
5.		Regresa al paso 4 del flujo básico de eventos: Notificar eventos.
<b>Flujos alternos</b>		
1a.El sistema no encuentra ninguna noticia obsoleta.		
	<b>Actor</b>	<b>Sistema</b>
		Regresa al paso 4 del flujo básico de eventos: Notificar eventos.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.

<b>Relaciones</b>	<b>CU Extendidos</b>	No procede.
<b>Requisitos funcionales</b>	no	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6.
<b>Asuntos pendientes</b>		No procede.

**2.7.3.1 Diagrama de clases del análisis para el caso de uso Notificar Eventos**



*Figura 4. Diagrama de Clases del Análisis Caso de Uso Notificar Eventos.*

## 2.8 Conclusiones parciales

Una vez terminado el capítulo se sentaron las bases para el desarrollo de la solución propuesta, además se logró:

- Se logró mostrar en términos de componentes, el análisis y el diseño, se especificaron los requisitos de software, se definieron los casos de uso del sistema, diagrama de caso de uso del sistema y la descripción de los mismos y se logró llegar a una comprensión total del sistema propuesto.
- Siguiendo las tendencias actuales y las políticas de migración hacia el software libre del país, se seleccionaron las herramientas para el desarrollo del módulo de reportes y notificaciones de eventos. Además una vez realizado el capítulo ya están creadas las bases para comenzar la implementación de dicho módulo.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE REPORTES Y NOTIFICACIONES.

### 3.1 Introducción

En este capítulo se especifica el patrón arquitectónico a utilizar así como los patrones de diseño utilizados para el desarrollo del Módulo de reportes y notificaciones de eventos para PRIMICIAv2.0. Se muestran en términos de componentes el modelo de implementación así como las pruebas aplicadas y los resultados obtenidos.

### 3.2 Descripción de la Arquitectura de software

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones (Reynoso, 2004).

#### 3.2.1 Patrón arquitectónico

Los patrones arquitectónicos expresan un esquema organizativo estructural fundamental para sistemas software y definen las reglas generales de organización, las restricciones en la forma y la estructura de un grupo numeroso de sistemas de software. La selección de un patrón arquitectónico es una decisión fundamental de diseño en el desarrollo de un sistema de software (Reynoso, 2004).

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC<sup>13</sup>, que está formado por tres niveles:

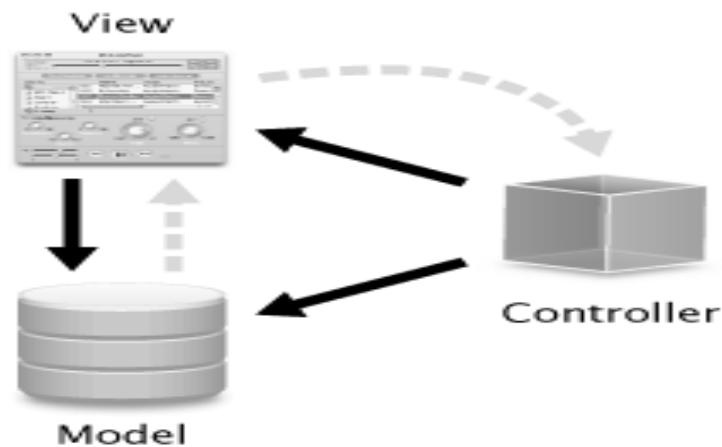
- El Modelo representa la información con la que trabaja la aplicación, es decir, el acceso a los datos.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.

---

<sup>13</sup> Modelo Vista Controlador.

- 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 y la presentación 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 o correo). 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 (LIBROSWEB, 2006).



*Figura 5.Arquitectura Modelo-Vista-Controlador*

### **3.2.2 Patrones de diseño**

Un patrón de diseño constituye una solución estándar para un problema común de programación en el desarrollo del software. Además es una técnica muy eficaz para flexibilizar el código haciéndolo satisfacer ciertos criterios, así como permite una manera más práctica de describir ciertos aspectos de la organización de un programa (Pressman, 2006).

### 3.2.2.1 Patrones GRASP<sup>14</sup>

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades).

- **Experto:** El patrón experto en información soluciona el problema ¿De qué forma podemos saber que responsabilidad delegar a cada objeto?, es el principio básico de asignación de responsabilidades. Este patrón se encarga de asignar una responsabilidad al experto en información, es decir, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad (Larman, 2004). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón se encuentra presente en la clase **TrackController.php** encargada de todo lo referente con los reportes de trazas.
- **Creador:** Soluciona el problema de ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?, ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase (Larman, 2004). Este patrón se evidencia en todas las clases.
- **Bajo Acoplamiento:** Soluciona el problema de ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?, asigna una responsabilidad para mantener bajo acoplamiento, es una medida de fuerza con que una clase esta conectada a otras clases, con que las conoce y con que recurre a ellas (Larman, 2004). Se utilizó para establecer un uso mínimo de clases entrelazadas de modo que si se hacen modificaciones solamente afecten a la menor cantidad de clases. Esto se evidencia en la estructura de componentes que se ha empleado, como el **jquery-charts.js** y el **sfPdf.php**, los cuales pueden ser reutilizados de manera independiente sin que sus funcionalidades se vean afectadas por estar separados del resto de la estructura del módulo.
- **Alta Cohesión:** Soluciona el problema de ¿Cómo mantener la complejidad dentro de límites manejables?, asigna una responsabilidad de modo que la cohesión siga siendo alta, es una

---

<sup>14</sup> Patrones generales de software para asignar responsabilidades (General Responsibility Assignment Software Patterns).

medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. La información que almacena una clase debe ser coherente y estar en la mayor medida relacionada con la clase (Larman, 2004).

- **Controlador:** El patrón controlador resuelve el problema de ¿Quién debería encargarse de atender un evento del sistema?, es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control (Larman, 2004). Este patrón es utilizado en todas las clases controladoras, como por ejemplo **Defaultcontroller.php**.

### 3.2.2.2 Patrones GOF<sup>15</sup>

- **Singleton (Instancia única):** En el diseño de clases es necesario aplicar la solución del patrón Singleton que no es más que garantizar la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. De esta forma se controla el acceso a las clases. El mismo es usado en la clase **PdfController.php**.
- **Observador:** Define una dependencia de uno-a-muchos entre objetos, de tal forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (Larman, 2004). Se evidencia en todo lo que se refiere a la notificación de eventos.
- **Fachada:** Este patrón brinda la posibilidad de utilizar una interfaz unificada de alto nivel. Permite ocultar toda la complejidad del sistema. Muestra solamente al usuario puntos de entrada y el acceso a introducir valores, estando siempre ajeno al funcionamiento interno del sistema. Este es usado cuando se pretende proporcionar una interfaz simple para un subsistema complejo (Larman, 2004). Este patrón se evidencia en todas las plantillas.

---

<sup>15</sup> Pandilla de los cuatro (Gang-of-Four)

### 3.3 Diagrama de Clases del Diseño

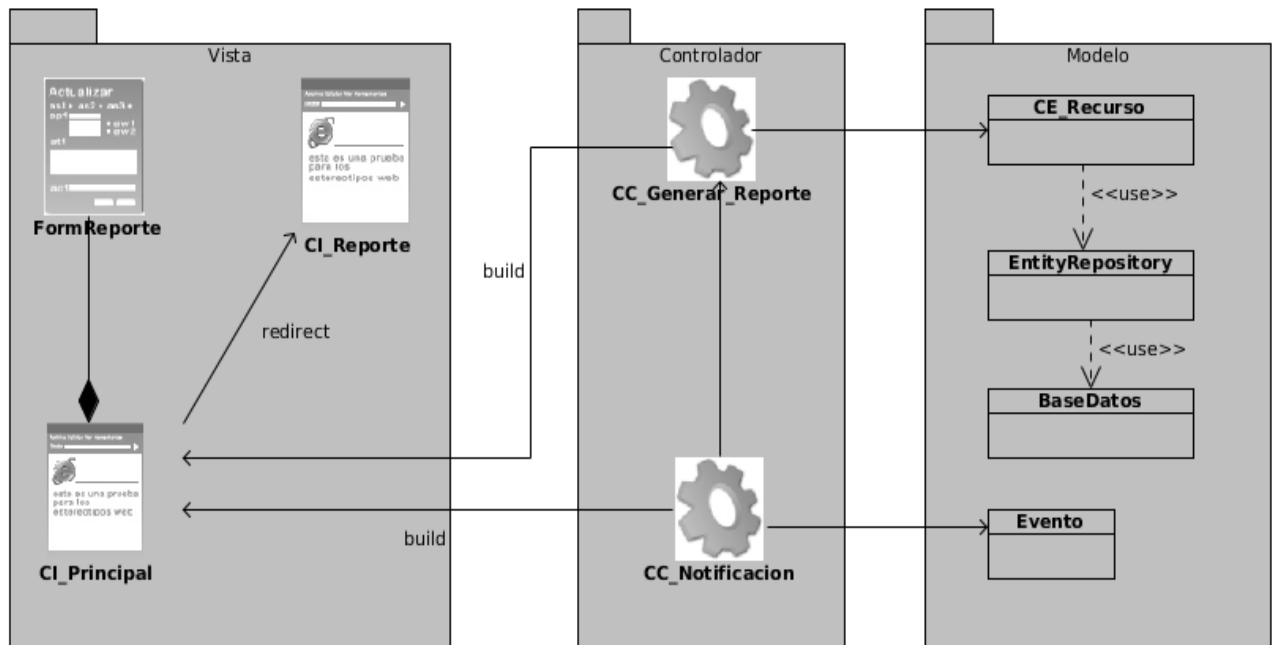


Figura 6. Diagrama de clases del diseño.

### 3.4 Modelo de Implementación

El Modelo de Implementación describe cómo los elementos del Modelo de Diseño, como las clases, se implementan en términos de componentes y subsistemas de implementación. Describe, además, la forma en que se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados. El Modelo de Implementación es la entrada principal de las etapas de prueba que siguen a la implementación (Jacobson, y otros, 2000).

#### 3.4.1 Modelo de datos

El modelo de datos es una representación abstracta de los datos de una organización y las relaciones entre ellos. El propósito de un modelo de datos es, por una parte, representar los datos y, por otra, ser comprensible. Elaborar un correcto modelo de datos es imprescindible para manejar y obtener correctamente los datos (Rosa-Rosario., 2011). A continuación se muestra el modelo de datos que



representa físicamente la base de datos del sistema. Esta fue diseñada para almacenar todos los datos referentes al proceso de generación de reportes y notificación de eventos. [Ver anexo.](#)

### 3.4.1 Diagrama de componentes

Por su parte, el diagrama de componentes es otro de los artefactos importantes que incluye el Modelo de Implementación. El mismo muestra elementos del modelo, tales como, los componentes y sus relaciones. Se utiliza para modelar la vista estática del sistema y muestra la organización y las dependencias lógicas entre los componentes de software, sean estos de código fuente, binarios o ejecutables (Jacobson, y otros, 2000).

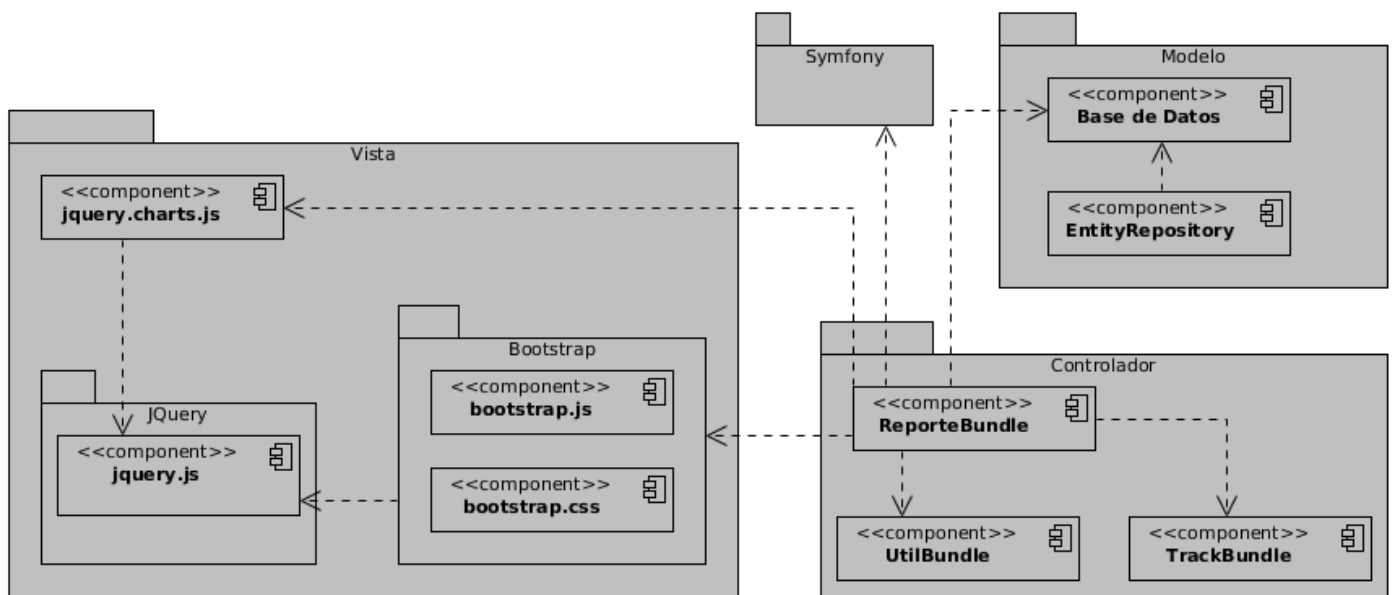


Figura 7. Diagrama de componentes.

### 3.4.2 Diagrama de despliegue

El diagrama de despliegue muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución en tiempo real en las instancias de los nodos de proceso (Larman, 2004).

El diagrama de despliegue que aparece a continuación muestra la disposición física de los distintos nodos que componen el sistema en tiempo de ejecución, los links de comunicación y las instancias de los componentes y objetos que residen en los mismos. Las conexiones establecidas en este diagrama son

asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Para el despliegue de la aplicación solo es necesario una PC cliente donde estará la plataforma y un servidor web donde estará también la Base de Datos.

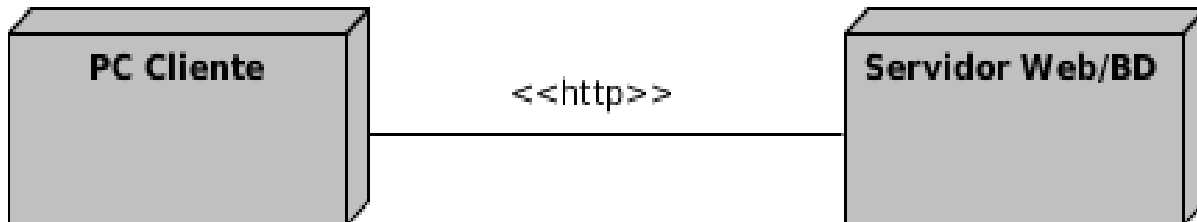


Figura 8. Diagrama de despliegue.

### 3.5 Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación. El objetivo fundamental de las pruebas es descubrir diferentes clases de errores con la menor cantidad de tiempo y de esfuerzo. Aunque las pruebas no pueden asegurar la ausencia de defectos; sí pueden demostrar que existen defectos en el software (Pressman, 2006).

Los objetivos principales de las pruebas de software son:

- Detectar un error.
- Tener un buen caso de prueba.
- Descubrir un error no descubierto antes.

La actividad de probar el sistema y verificar que se encuentre libre de defectos tiene muchos beneficios. La calidad es una variable muy importante para todo producto y uno de los caminos para garantizarla es siguiendo esta doctrina, además proporciona una medida del progreso del trabajo que se despliega (Pressman, 2006). Existen varios niveles de pruebas, entre ellos se encuentran: las unitarias, de sistemas, de integración, funcionales, no funcionales, aceptación así como de regresión. Las pruebas que serán aplicadas al módulo de reportes y notificaciones de eventos son caja negra y aceptación.

### 3.5.1 Método de prueba de caja negra

Estas pruebas pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Se centran en los requisitos funcionales del software y permiten encontrar errores de tipo de funciones incorrectas o ausentes, de interfaz, en la estructura de datos, de rendimiento y de inicialización y terminación (Pressman, 2006).

Al módulo de reportes y notificación de eventos para la Plataforma de Televisión Informativa PRIMICIA se le realizan pruebas de caja negra pues estas se centran en lo que se espera del mismo, es decir, intentan encontrar casos en que este no atiende a su especificación. Por ello se denominan pruebas funcionales y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo por dentro.

Con este tipo de pruebas se intenta encontrar (Pressman, 2006):

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para la realización de este método se utilizó la técnica de partición equivalente, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (Pressman, 2006).

Según RUP estas pruebas verifican el comportamiento de la unidad observable externamente. Cuando el número de entradas y salidas es grande, se dividen estas en clases de equivalencias. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman, 2006).

### 3.5.1.2 Diseños de casos de prueba

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de uso del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema (Pressman, 2006).

#### Secciones a probar en el caso de uso Generar Reportes

Tabla 4: Pruebas para el caso de uso Generar Reportes.

Nombre del caso de uso	Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Resultados Esperados	Resultados Obtenidos
Generar Reportes.	SC 1: Reporte de infocintas.	EC 1.1: Se obtiene el reporte de infocintas correctamente.	Se mostrará un listado de las infocintas que fueron transmitidas en un intervalo de tiempo según una fecha inicio y una fecha fin determinada.	El sistema muestra un listado de las infocintas que fueron transmitidas en un intervalo de tiempo determinado por el usuario.	<b>Satisfactorio</b>
		EC 1.2: No se obtiene el reporte de infocintas.	No se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que las fechas están incorrectas o que ha dejado campos en blanco que deben ser llenados.	<b>Satisfactorio</b>

	SC 2: Reporte temático.	EC 2.1: Se obtiene el reporte temático correctamente.	Brindará la opción de seleccionar la temática de la que se quiere generar el reporte, una vez seleccionada mostrará un listado con las noticias de dicha temática publicadas en el intervalo de tiempo según una fecha inicio y una fecha fin determinada.	El sistema muestra un listado con las noticias de dicha temática, publicadas en el intervalo de tiempo según una fecha inicio y una fecha fin determinada por el usuario.	<b>Satisfactorio</b>
		EC 2.2: No se obtiene el reporte temático.	No se selecciona la temática de la que se quiere el reporte o no se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que debe seleccionar una temática o que ha dejado campos en blanco que deben ser llenados.	<b>Satisfactorio</b>
	SC 3: Reporte de medias.	EC 3.1: Se obtiene el reporte de medias correctamente.	Brindará la opción de seleccionar la media de la que se quiere el reporte, una vez seleccionada se	El sistema muestra un listado con las noticias publicadas en el intervalo de tiempo seleccionado por el	<b>Satisfactorio</b>

			mostrará un listado con las noticias publicadas en un intervalo de tiempo según una fecha inicio y una fecha fin determinada que utilizan dicha media.	usuario.	
		EC 3.2: No se obtiene el reporte de medias.	No se selecciona la media de la que se quiere el reporte o no se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que debe seleccionar la media de la que quiere el reporte o que ha dejado campos en blanco que deben ser llenados.	<b>Satisfactorio</b>
SC 4: Reporte de noticias.	EC 4.1: Se obtiene el reporte de noticias correctamente.	Brindará la opción de seleccionar el estado de la noticia (redactada, corregida, publicada) del que se quiere el reporte, una vez seleccionado este, se mostrará un	El sistema muestra un listado con las noticias que se encuentren en el estado preseleccionado partir de un intervalo de tiempo seleccionado por el usuario.		<b>Satisfactorio</b>

			listado con las noticias que se encuentren en dicho estado preseleccionado a partir de un intervalo de tiempo según una fecha inicio y una fecha fin determinada.		
		EC 4.2: No se obtiene el reporte de noticias.	No se selecciona el estado de la noticia o no se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que debe seleccionar el estado de la noticia de la que quiere el reporte o que ha dejado campos en blanco que deben ser llenados.	<b>Satisfactorio</b>
	SC 5: Reporte en forma de gráfico.	EC 5.1: Se obtiene el reporte en forma de gráfica correctamente.	Brindará la opción de mostrar algunos reportes en formas de gráficas de barra.	El sistema muestra en forma de gráfica de barras el reporte.	<b>Satisfactorio</b>
		EC 5.2: No se obtiene el reporte en	No se selecciona el reporte en forma de	El sistema no muestra nada al	<b>Satisfactorio</b>

		forma de gráfica.	gráfica.	usuario.	
SC 6: Reporte de notificación.	EC 6.1: Se obtiene el reporte de notificación correctamente.	Se mostrará un listado con todas las notificaciones que se realizaron en un intervalo de tiempo según una fecha inicio y una fecha fin determinada.	El sistema muestra un listado con todas las notificaciones que se realizaron en el intervalo de tiempo preseleccionado por el usuario.	<b>Satisfactorio</b>	
	EC 6.2: No se obtiene el reporte de notificación.	No se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que debe llenar los campos en blanco.	<b>Satisfactorio</b>	
SC 7: Reporte de señales.	EC 7.1: Se obtiene el reporte de señales correctamente.	Brindará la opción de seleccionar la señal de la cual se quiere el reporte, una vez seleccionada, se mostrará un listado con la programación transmitida en dicha señal en un intervalo de tiempo según una fecha inicio y una fecha fin	El sistema muestra un listado con la programación transmitida en dicha señal, en el intervalo de tiempo preseleccionado por el usuario.	<b>Satisfactorio</b>	



			determinada.		
		EC 7.2: No se obtiene el reporte de señales.	No selecciona la señal de la que quiere el reporte o no se selecciona el intervalo de fechas.	El sistema muestra un mensaje al usuario indicándole que debe seleccionar la señal de la que quiere el reporte o que debe llenar los campos en blanco.	<b>Satisfactorio</b>
SC 8: Reporte de trazas.	EC 8.1: Se obtiene el reporte de trazas correctamente.	Muestra las acciones que se realizan en la aplicación en el entorno de producción.	El sistema muestra un reporte con las acciones realizadas en la aplicación, solamente en el entorno de producción.	<b>Satisfactorio</b>	
	EC 8.2: No se obtiene el reporte de trazas.	No se selecciona el reporte de trazas.	El sistema no muestra nada al usuario.	<b>Satisfactorio</b>	
SC 9: Exportar a pdf.	EC 9.1: Se exporta el reporte a pdf correctamente.	Permitirá exportar el reporte a un documento con formato pdf.	El sistema exporta el reporte a pdf.	<b>Satisfactorio</b>	
	EC 9.2: No se	No selecciona la	El sistema no	<b>Satisfactorio</b>	

		exporta el reporte a pdf.	opción "Exportar reporte a pdf".	exporta el reporte a pdf.	
--	--	---------------------------	----------------------------------	---------------------------	--

### Secciones a probar en el caso de uso Notificar Evento

Tabla 5: Pruebas para el caso de uso Notificar Evento.

Nombre del caso de uso	Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Resultados Esperados	Resultados Obtenidos
Notificar Evento.	SC 1: Notificar medias obsoletas.	EC 1.1: Se notifica la existencia de medias obsoletas correctamente.	Se enviará un mensaje a los administradores del sistema informándoles la existencia de medias que se encuentren obsoletas, la cuales pueden ser eliminadas.	El sistema debe enviar una notificación por correo electrónico al usuario indicándole que hay medias obsoletas y recomendándole su eliminación.	<b>Satisfactorio.</b>
		EC 1.2: No se notifica existencia de medias obsoletas.	No se encuentran medias obsoletas.	El sistema no debe enviar notificación.	<b>Satisfactorio.</b>
	SC 2: Notificar noticias	EC 2.1: Se notifica la existencia de noticias	Se enviará un mensaje a los administradores del sistema	El sistema debe enviar una notificación por correo	<b>Satisfactorio.</b>

	obsoletas.	obsoletas correctamente.	informándoles la existencia de noticias que se encuentren obsoletas, la cuales pueden ser eliminadas.	electrónico al usuario indicándole que hay noticias obsoletas y recomendándole su eliminación.	
		EC 2.2: No se notifica existencia de noticias obsoletas.	No se encuentran noticias obsoletas.	El sistema no debe enviar notificación.	<b>Satisfactorio.</b>

### Resultados de la prueba de Caja negra

En la fase de pruebas se realizaron dos iteraciones al Módulo de reportes y notificaciones de eventos, en la primera se detectaron cuatro no conformidades. Los requisitos funcionales que no tenían un correcto funcionamiento eran: **RF2**. Generar reporte según la sección temática seleccionada a partir de un rango de fechas preseleccionado, **RF5**. Generar reporte de noticias a partir un rango de fechas preseleccionado, **RF7**. Generar reporte de notificaciones a partir un rango de fechas preseleccionado, **RF8**. Generar reporte de señales a partir un rango de fechas preseleccionado. Se realizó posteriormente una segunda iteración de prueba para verificar que se habían eliminado los problemas encontrados en estos requisitos o si se encontraban nuevas no conformidades en el funcionamiento del módulo. Esta segunda fase concluyó satisfactoriamente ya que no se encontró ningún otro problema en el Módulo de reporte y notificaciones de eventos.

### 3.5.2 Pruebas de aceptación

Las pruebas de aceptación de usuario validan el grado de satisfacción del cliente final. Este tipo de pruebas son comúnmente realizadas por el usuario final quien debe informar de todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente. Como mínimo se deben realizar dos tipos de pruebas, alfa y beta (Testhouse, 2011). Para el Módulo de reporte y notificación de

eventos se aplicaron las pruebas alfa que consisten en hacer pruebas funcionales como clientes en un entorno de desarrollo. Se trabaja en un entorno controlado y siempre se tiene un experto a mano para ayudarle a usar el sistema y para analizar los resultados (Testhouse, 2011).

## Resultados

Esta prueba fue realizada junto con el equipo de desarrollo de la Plataforma de Televisión Informativa, PRIMICIA puesto que ellos constituyen el cliente final y de esta manera se logró la aprobación del módulo por parte del proyecto. Durante la prueba intervinieron los miembros del equipo de desarrollo de la plataforma. Ver tabla 6.

*Tabla 6: Equipo de desarrollo de la Plataforma de Televisión Informativa, PRIMICIA.*

Equipo de desarrollo de la Plataforma de Televisión Informativa, PRIMICIA	
Rol que desempeña	Nombre y Apellidos
Líder del proyecto.	Félix Iván Romero Rodríguez.
Arquitecto de Software.	Yanary Hernández Sosa.
Programador.	Dorgis Montes de Oca Gómez.
Implantador de soluciones.	Rafael Lorente Miranda.

### 3.6 Conclusiones parciales

En este capítulo se seleccionó la arquitectura que tendrá el módulo de repostes y notificaciones de eventos en su desarrollo, los patrones de diseño y las pruebas aplicadas a dicho módulo por lo que se concluye que:

- La utilización del patrón arquitectónico MVC para el desarrollo del módulo, posibilitó separar la lógica de negocio y la presentación, lográndose un mantenimiento más sencillo del módulo.
- La selección de los patrones de diseño tanto GRASP como GOF permitió sentar la base para dar solución al problema en cuestión obteniendo un mejor funcionamiento del módulo de reportes y notificaciones de eventos.
- La realización de las pruebas de caja negra posibilitó validar el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo de la aplicación.
- Con la elaboración de la prueba de Aceptación se logró la aprobación del Moduló de reportes y notificación de eventos por parte del proyecto, PRIMICIA.

## CONCLUSIONES

Durante el desarrollo del presente trabajo de diploma se llevaron a cabo una serie de pasos que sirvieron de guía para dar cumplimiento al objetivo planteado, arribando a las siguientes conclusiones:

- Se logró conocer el estado del objeto de estudio de la investigación, permitiendo la comprensión de forma satisfactoria de toda la situación problemática de la misma, a partir de la utilización de los métodos de investigación científica tanto teóricos como empíricos.
- El estudio de otras aplicaciones generadoras de reporte y notificaciones permitió identificar conceptos fundamentales, herramientas y tecnologías relacionadas con la misma, así como llegar a la conclusión de por qué no se utilizan en la plataforma, PRIMICIA.
- La realización de pruebas de caja negra al módulo permitió comprobar el correcto funcionamiento del mismo.
- Se logró desarrollar el Módulo de reportes y notificación de eventos para la Plataforma de Televisión Informativa, PRIMICIAv2.0, dándole solución al problema a resolver que dio fundamento a la investigación realizada.

## RECOMENDACIONES

Luego de concluir con la implementación del Módulo de reportes y notificación de eventos para la plataforma PRIMICIAv2.0 se recomienda:

- Investigar e incluir otras vías de notificación como pueden ser sms<sup>16</sup> y viper<sup>17</sup>.
- Realizar otros tipos de prueba, como la de integración, para evaluar completamente el funcionamiento del Módulo de reportes y notificación de eventos.

---

<sup>16</sup> Servicio de Mensaje Corto.

<sup>17</sup> Dispositivo para mensajes de textos.

## BIBLIOGRAFÍA

ABC, D. (2007). *Definicion ABC*. Recuperado el 20 de noviembre de 2012, de Definicion ABC:  
<http://www.definicionabc.com/general/feedback.php>

ABC, D. (2008). *Definicion ABC*. Recuperado el 20 de noviembre de 2012, de Definicion ABC:  
<http://www.definicionabc.com/comunicacion/reporte.php>

Acuña, C. J. (2012). *Ingeniería de Software I*. Mexico.

Angelismo. (5 de Julio de 2006). *Skindario*. Recuperado el 12 de Febrero de 2013, de skindario.com:  
<http://skindario.com/768851/5423668-usos-y-ventajas-del-php/>

Baneres, J. (2006). *Los Sistema de Notificacion de Incidentes y Eventos Adversos*. Madrid: Fundacion Avedis Donabedian.

Burgos, U. d. (1999). *Introduccion a la Programacion Orientada a Objetos*. Burgos.

Cabeio, C. R. (2009). *Tecnología Pyme*. Recuperado el 20 de Noviembre de 2012, de [www.tecnologiapyme.com](http://www.tecnologiapyme.com):  
<http://www.tecnologiapyme.com/software/redmine-gestor-de-proyectos-de-codigo-libre-para-nuestras-empresas>

Carzaniga, A. (1997). *A characterization of the software deployment process and a survey of related technologies*.

Chaves, M. A. (2006). *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*. Costa Rica.

Choque, G. (30 de Abril de 2012). *El Diario*. Recuperado el 20 de noviembre de 2012, de El Diario:  
[http://www.eldiario.net/noticias/2012/2012\\_04/nt120430/ciencia.php?n=6](http://www.eldiario.net/noticias/2012/2012_04/nt120430/ciencia.php?n=6)

Ciberaula. (2010). *Ciberaula Linux*. Recuperado el 12 de Febrero de 2013, de [ciberaula.com](http://linux.ciberaula.com):  
[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro)

DataPro. (1987). *DataPro*. Recuperado el 19 de febrero de 2013, de Datapro:  
[http://www.datapronet.com/index.php?id\\_seccion=110&tipo\\_seccion=section](http://www.datapronet.com/index.php?id_seccion=110&tipo_seccion=section)

Definicion. (2008). *Definicion.de*. Recuperado el 20 de noviembre de 2012, de Definicion.de:  
<http://definicion.de/notificacion/>

Dictionary, T. F. (2010). *The Free Dictionary*. Recuperado el 20 de noviembre de 2012, de The Free Dictionary:  
<http://es.thefreedictionary.com/modulo>

EffectCard. (23 de Agosto de 2005). *Skindario*. Recuperado el 12 de Febrero de 2013, de skindario.com:  
<http://skindario.com/768851/4098456-ventajas-del-php/>



Ferrer, J., & Garcia, V. (2010). *MMC*. Recuperado el 12 de Febrero de 2013, de [mmc.geofisica.unam.mx: http://mmc.geofisica.unam.mx/LuCAS/Manuales-LuCAS/doc-curso-html/doc-curso-html/introduccion-html.html](http://mmc.geofisica.unam.mx/LuCAS/Manuales-LuCAS/doc-curso-html/doc-curso-html/introduccion-html.html)

Gladys. (Mayo de 2011). *Buenas Tareas*. Recuperado el 19 de Febrero de 2013, de [www.buenastareas.com: http://www.buenastareas.com/ensayos/Caracteristicas-Principales-De-Uml/2234495.html](http://www.buenastareas.com/ensayos/Caracteristicas-Principales-De-Uml/2234495.html)

Hoplite. (2005). *Hoplite*. Recuperado el 5 de Febrero de 2013, de Hoplite: <http://www.hoplite.es/tecnologias/symfony>

IBM, C. (2006). *Ayuda de RUP*.

Ivar Jacobson, G. B. (2000). *El Lenguaje Unificado de Modelado. Manual de referencia*.

Jacobson. (1999). *El Proceso Unificado de Desarrollo de Software*.

Larman, C. (1999). *UML y Patrones*. Montevideo Uruguay.

Larman, C. (2004). *UML y patrones. Introduccion al analisis y diseno orientado a objetos. Tomo 1*. La Habana: Felix Varela.

Lee, H. W., & Bos, B. (5 de Marzo de 2001). *MBI*. Recuperado el 19 de Febrero de 2013, de [html.conclase.net: http://html.conclase.net/w3c/css1-es.html](http://html.conclase.net/w3c/css1-es.html)

Leon, G. B. (1993). *Generador de Reportes, espezifico para el centro de computacion del ministerio de defensa nacional*. Guatemala.

León, R. A., & González, S. C. (2002). *El paradirigma cuantitativo de la investigacion cientifica*. La Habana: EDUNIV.

LIBROSWEB. (2006). *LIBROSWEB*. Recuperado el 23 de Abril de 2013, de <http://librosweb.es>: [http://librosweb.es/symfony\\_1\\_2/capitulo\\_2/el\\_patron\\_mvc.html](http://librosweb.es/symfony_1_2/capitulo_2/el_patron_mvc.html)

Lycanthrop, L. (8 de Marzo de 2007). *CristaLab*. Recuperado el 12 de Febrero de 2013, de [cristalab.com: http://www.cristalab.com/tutoriales/tutorial-de-jquery-c214/](http://www.cristalab.com/tutoriales/tutorial-de-jquery-c214/)

Martinez, J. (3 de Diciembre de 2009). *Loogic*. Recuperado el 20 de Noviembre de 2012, de <http://loogic.com/sistemas-de-recomendacion/>

Martinez, R. (2009). *PostgreSQL.org*. Recuperado el 19 de Marzo de 2013, de [postgresql-es.org: http://www.postgresql-es.org/sobre\\_postgresql](http://www.postgresql-es.org/sobre_postgresql)

Masaryk. (2009). *Avaya, House 111 Piso 6 Col. Polanco 11570 Mexico D.F.* (Avaya Inc.) Recuperado el 22 de Enero de 2013, de <http://www.avaya.com>

- Mirabal, I. L. (2012). *Requerimientos funcionales y no funcionales*.
- NetBeans. (2009). *NetBeans*. Recuperado el 12 de Febrero de 2013, de netbeans.com:  
<https://netbeans.org/features/platform/index.html>
- Noguera, B. (5 de Julio de 2011). *Culturacion*. Recuperado el 12 de Febrero de 2013, de culturacion.com:  
<http://culturacion.com/2011/07/que-es-ajax/>
- Olave, R. P. (2003). *Los sistemas informaticos expertos de toma de decisiones y la voluntad de validez del negocio juridico*.
- Orallo, E. H. (2009). *El Lenguaje Unificado de Modelado*.
- PHP, P. d. (2008). *Patrone de diseno PHP*. Recuperado el 23 de Abril de 2013, de Patrone de diseno PHP:  
<http://patronesdiseniophp.blogspot.com/p/patron-observer.html>
- Picon, C. S., & Caro, L. F. (2007). *Infogestion*. Concepcion.
- Pinargote, E. G. (2012). *Tecnicas, Metodos empiricos y tipos de investigacion y de observacion*. Ecuador.
- Pressman, R. (2006). *Ingenieria de Software un enfoque practico*. s.l. : Dorset House.
- Pressman, R. (2005). *Ingenieria de Software. Un enfoque practico*. Espana: Mc Graw Hill.
- Ramirez, N. A. (2003). *Caracterizacion y diseno de los resultados cientificos como aporte de la investigacion educativa*. La Habana: Ciencias Sociales.
- Reynoso, C. B. (2004). *Introducción a la Arquitectura de Software*. Buenos Aires.
- Rodriguez, A. P. (2011). *Sistema de recomendacion colaborativo basado en el algoritmo de filtrado mejorado*.
- Rodriguez, L. C., & Alvarez, J. A. (2010). *Manual de GDR (Generador Dinamico de Reportes)*. UCI, La Habana.
- Rosa-Rosario., Dra. María G. 2011. Modelos de Bases de Datos. Base de Datos Relacionales. 2011.
- Schmuller, J. (2000). *Aprendiendo UML en 24 Horas*. Mexico: Pearson Educacion.
- Seguridad, C. d. (2009). *Sistema de Notificacion de Incidentes y eventos Adversos*. Madrid.
- Software, D. c. (2010). *Introduccion a RUP y UML*. La Habana.
- SYNET. (2006). *SYNET*. Recuperado el 19 de Febrero de 2013, de www.synet.com:  
<http://www.synet.com.pe/index.php/boletas/productos/contable/generador-de-reportes>

Testhouse. (2011). *Testhouse*. (Testhouse Consultores S.A) Recuperado el 7 de Mayo de 2013, de Testhouse:  
<http://www.es.testhouse.net/pruebas-de-aceptacion/>

TLDP. (2010). *Equipo de desarrollo del Manual de Usuario de PostgreSQL*. Recuperado el 25 de Marzo de 2013, de  
<http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>

Valdelli, I. (2006). *HTMLPOINT*. Recuperado el 12 de Febrero de 2013, de [htmlpoint.com](http://www.htmlpoint.com):  
[http://www.htmlpoint.com/javascript/corso/js\\_02.htm](http://www.htmlpoint.com/javascript/corso/js_02.htm)

Zayas, C. (1995). *Metodologia de la Investigacion Cientifica*. Santiago de Cuba.

# ANEXOS

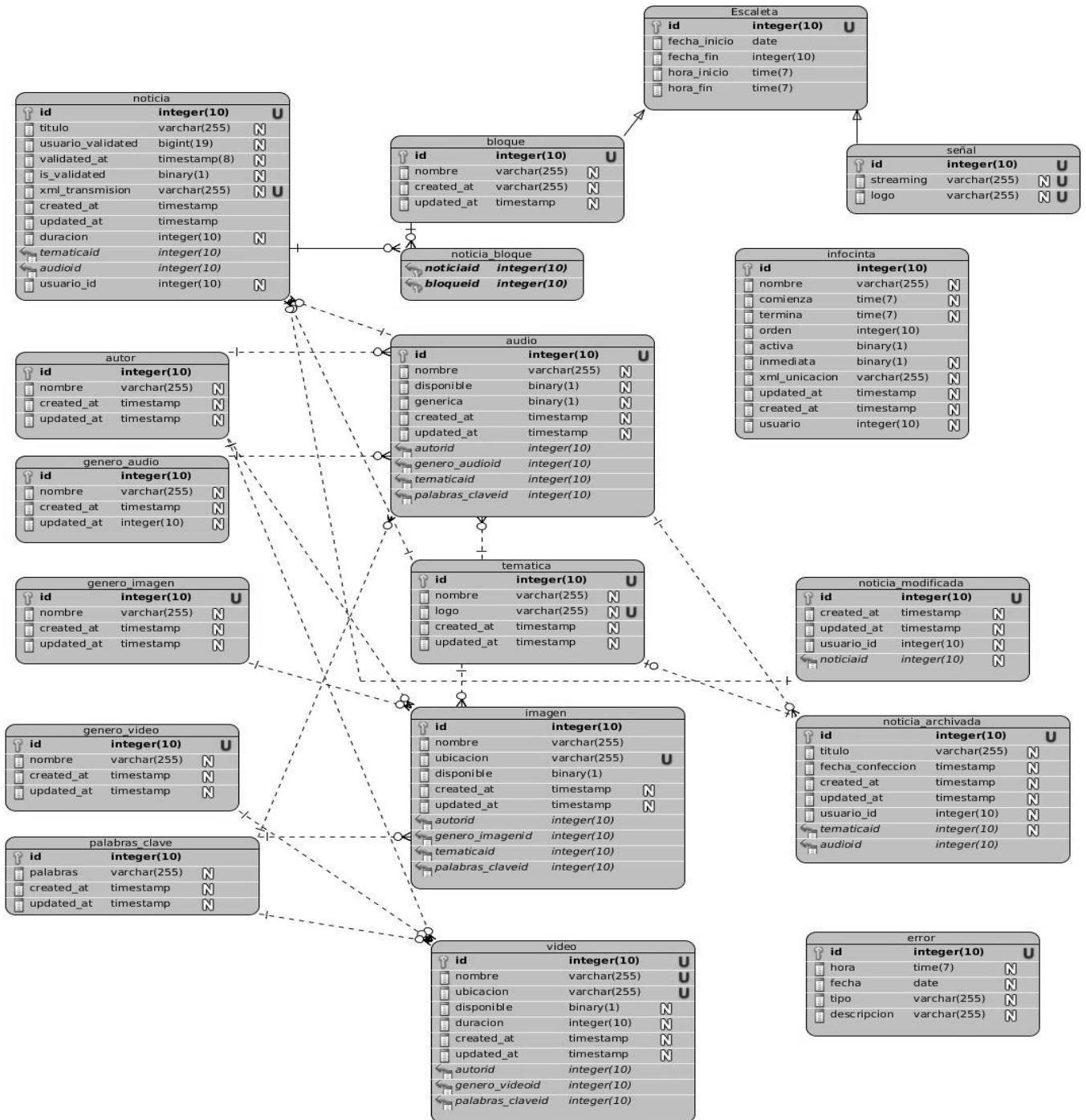


Figura 9. Modelo de datos.

## GLOSARIO

**Artefactos:** productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente.

**BSD:** Berkeley Software Distribution o Distribución de Software Berkeley, licencia de software que al contrario de la GPL permite el uso del código fuente en software no libre.

**CASE:** *Computer Aided Software Engineering* o Ingeniería de Software Asistida por Computadora, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

**CSS:** Cascade Style Sheet o Hoja de Estilo en Cascada.

**dBASE:** Fue el primer sistema de gestión de base de datos usado ampliamente para microcomputadoras.

**DOM:** Document Object Model (Modelo de Objetos del Documento), interfaz de programación de aplicaciones (API) para documentos HTML y XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula un documento.

**DpXbase:** Ambiente de desarrollo para léxico de tipo Xbase y diseñado para crear aplicaciones de tipo comercial de manera rápida y sencilla.

**HTML:** HyperText Markup Language o Lenguaje de Marcado de Hipertexto.

**IBM:** International Business Machines.

**PDO:** PHP Data Object o Objetos de Datos de PHP, define un interfaz ligera para poder acceder a bases de datos en PHP.

**PHP:** Page Hypertext Processor o Procesador de Páginas de Hipertextos.

**POO:** Object Oriented Programming o Programación Orientada a Objetos.

**Reverse Proxy:** Es un servidor proxy instalado en el domicilio de uno o más servidores web. Todo el tráfico entrante de Internet y con el destino de uno de esos servidores web pasa a través del servidor proxy.

**Roles:** papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso.

**RUP:** Rational Unified Process o Proceso Unificado de Desarrollo.

**SGML:** Standard Generalized Markup Language o Lenguaje de Marcado de Anotaciones Generales.

**UML:** Unified Modeling Language o Lenguaje Unificado de Modelado.

**WWW:** World Wide Web.

**xBase:** Es el término genérico para todos los lenguajes de programación que derivan del lenguaje de programación dBase.

**XHTML:** eXtensible HyperText Markup Language o Extensible Lenguaje de Marcado de Hipertexto.

**XML:** eXtensible Markup Language o Lenguaje de Marcas Extensible.

**XPath:** Lenguaje que permite seleccionar nodos de un documento XML y calcular valores a partir de su contenido.

**XSLT:** Extensible Stylesheet Language Transformation o lenguaje de transformación basado en hojas de estilo.