

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 9



Desarrollo del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA

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

Autores:

Daniel Barrios Cardoso

Marisleidy Torres Camejo

Tutor:

Ing. Rafael Lorente Miranda

Ciudad de La Habana, julio de 2010

“Año 52 de la Revolución”

Declaración de Autoría

Declaramos ser autores del trabajo de diploma titulado: Desarrollo del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA. Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Marisleidy Torres Camejo

Daniel Barrios Cardoso

Firma del Autor

Firma del Autor

Ing. Rafael Lorente Miranda

Firma del Tutor

Resumen

En los últimos años el uso de la tecnología digital ha transformado el proceso de captación, edición y emisión de noticias audiovisuales. Estos procesos han evolucionado, permitiéndole al redactor hacer su trabajo con mayor eficacia a través de las nuevas posibilidades que ofrece la “redacción digital integrada”. Cuba, no exento de estas tecnologías y con el objetivo de perfeccionar sus medios de comunicación, desarrolla en el marco físico de la Universidad de las Ciencias Informáticas (UCI), el sistema PRIMICIA. Este sistema ha venido desarrollando su propia estrategia orientada a la gestión de un canal televisivo, con las mismas características relacionadas con el proceso de “redacción digital integrada”, y sobre el cual se focaliza la investigación. De tal manera, en el presente trabajo se persigue la realización de una API (interfaz programable de aplicaciones) o motor XML que medie en el proceso de generación y carga de los ficheros de noticias. Dicho motor se integra a la herramienta de maquetación o edición de noticias, y constituye la base del proceso de integración entre los subsistemas que componen la plataforma PRIMICIA. Para cumplir con dicho objetivo, se presentan los estudios que dan soporte al desarrollo de la investigación, así como también se abordan los aspectos técnicos tomados en cuenta en los procesos de selección de herramientas y métodos informáticos que dan vida a la solución en cuestión. Específicamente se realizó un análisis de las características asociadas al desarrollo de la API, se estudió la factibilidad que aporta el uso del lenguaje XML para la concepción de un estándar de noticias, y finalmente se materializaron las tareas de diseño e implementación del prototipo funcional concerniente a la investigación.

Palabras Claves

PRIMICIA, XML esquema, XML, API, audiovisuales, DTD.

Índice de Tablas

Tabla 1: Descripción de la jerarquía de elementos que componen el formato XML de la plataforma PRIMICIA.....	31
Tabla 2: Definición de los actores del sistema	38
Tabla 3: Especificación del caso de uso Gestionar Escena	39
Tabla 4: Especificación del caso de uso Gestionar Acción.....	39
Tabla 5: Especificación del caso de uso Gestionar Recurso	39
Tabla 6: Especificación del caso de uso Gestionar Efecto	40
Tabla 7: Especificación del caso de uso Gestionar Estilo.....	40
Tabla 8: Especificación del caso de uso Validar Datos	40
Tabla 9: Especificación del caso de uso Búsqueda de Elementos	41
Tabla 10: Especificación del caso de uso Generar XML	41
Tabla 11: Especificación del caso de uso Cargar XML.....	41
Tabla 17: Modelo extendido del caso de uso Validar Datos	46
Tabla 18: Modelo extendido del caso de uso Buscar Elemento	47
Tabla 19: Modelo extendido del caso de uso Generar XML	48
Tabla 20: Modelo extendido del caso de uso Cargar XML.	50

Índice de Figuras

Figura 1: XML versión abreviada de SGML optimizada, HTML nace a partir de SGML.....	7
Figura 2: Ciclo de vida de RUP	24
Figura 3: Jerarquía de elementos que conforman en formato XML de la plataforma PRIMICIA	30
Figura 4: Modelo de dominio del sistema	33
Figura 5: Diagrama del caso de uso Gestionar Escena.....	42
Figura 6: Diagrama del caso de uso Gestionar Acción.....	42
Figura 7: Diagrama del caso de uso Gestionar Recurso	43
Figura 8: Diagrama del caso de uso Gestionar Efecto	43
Figura 9: Diagrama del caso de uso Gestionar Estilo.....	44
Figura 10: Diagrama del caso de uso Generar XML	44
Figura 11: Diagrama del caso de uso Cargar XML.....	44
Figura 12: Diagrama de paquetes del diseño	55
Figura 13: Distribución de clases del paquete Property.....	56
Figura 14: Distribución de clases del paquete Properties.....	57
Figura 15: Distribución de clases del paquete Interfaces.	57
Figura 16: Distribución de clases del paquete Attributes	58
Figura 17: Distribución de clases del paquete Resources.....	58
Figura 18: Distribución de clases del paquete Elements.....	59
Figura 19: Distribución de clases del paquete Effect.....	59
Figura 20: Diagrama de clases del diseño (Distribución global).	60
Figura 21: Distribución física global de la API.	61
Figura 22: Distribución interna del nodo Resorces.....	62
Figura 23: Distribución del nodo físico Elements.....	62

Índice

Introducción	1
Capítulo 1. Fundamentación teórica.....	5
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Redacción Digital Integrada	5
1.2.2 Librerías de software.....	6
1.2.3 Lenguaje estándar Extensible Markup Language (XML)	6
1.2.4 Definición de tipo de documento: DTD	8
1.2.5 Esquemas XML.....	9
1.2.6 Estándares del XML.....	10
1.3 Objeto de Estudio	11
1.3.1 Descripción General.....	11
1.3.2 Descripción actual del dominio del problema	13
1.3.3 Situación Problemática	14
1.4 Análisis de otras soluciones existentes.....	15
1.4.1 Ámbito Internacional	15
1.5 Síntesis Temática.....	18
1.6 Conclusiones.....	19
Capítulo 2. Tendencias y tecnologías actuales.	20
2.1 Introducción.....	20
2.2 Lenguajes de programación propuestos.....	20
2.2.1 Lenguaje PHP (Hypertext Pre-processor).....	20

2.2.3 Lenguaje de modelado UML	21
2.3 Metodologías de desarrollo de software.	22
2.3.1 Fundamentación de la selección	23
2.3.2 Proceso Unificado de Rational (RUP; Rational Unified Process).....	23
2.4 Herramientas de Modelado de Software.....	25
2.4.1 Fundamentación de la selección	25
2.4.2 Visual Paradigm.....	26
2.5 Entorno de Desarrollo Integrado.....	27
2.5.1 Fundamentación de la selección	27
2.5.2 Eclipse PDT	27
2.6 Conclusiones.....	28
Capítulo 3. Presentación de la solución propuesta.....	29
3.1 Introducción.....	29
3.2 Descripción de la solución propuesta.	29
3.2.1 XML como formato para el almacenamiento de datos.....	29
3.2.2 Fichero XML de configuración de noticia. Estructura Básica.	30
3.2.3 Aspectos Generales.....	31
3.2.4 Modelo de Objetos del Documento (DOM).....	32
3.3 Diseño del Sistema.....	32
3.3.1 Modelo del Dominio	32
3.2.2 Captura de los requerimientos	34
3.3.3 Modelo de Casos de Uso	38
3.3.4 Diagramas de Casos de Uso del Sistema	42
3.3.5 Descripción de casos de uso en formato extendido	45

3.4 Análisis.....	50
3.5 Conclusiones.....	50
Capítulo 4: Construcción de la solución propuesta.....	52
4.1 Introducción.....	52
4.2 Diseño	52
4.2.1 Modelo de Diseño	52
4.2.2 Patrones de Diseño.....	52
4.2.3 Diagramas de clases del diseño.....	55
4.3 Implementación	60
4.3.1 Diagrama de despliegue	60
4.3.2 Modelo de implementación	61
4.4 Conclusiones.....	63
Conclusiones	64
Recomendaciones	65
Trabajos citados.....	66
Glosario de términos.....	70

Introducción

Los medios de comunicación han ido evolucionando a pasos agigantados en las últimas décadas, debido a los avances tecnológicos obtenidos en la informática y las comunicaciones (TIC), de esta forma se ha logrado captar una mayor cantidad de seguidores, gracias a las ventajas que ofrecen estos adelantos.

La televisión es un medio de comunicación audiovisual muy difundido en todo el mundo, debido a los múltiples servicios que brinda. Entre los servicios que soporta el medio se encuentran: programas educativos, de negocios, informaciones culturales, deportivas, el acontecer nacional e internacional, e infinidad de beneficios que permiten un acercamiento a cualquier parte del mundo para mantenerse al tanto de todo lo que resulte interesante. Por dichas razones el entorno audiovisual se encuentra en la élite de las mejoras tecnológicas existentes.

En la actualidad los sistemas de canales televisivos enfrentan un nuevo paradigma, cimentado en la introducción de la tecnología digital y sus ramas asociadas. El establecimiento y consolidación de dicha tecnología ha revolucionado el medio en nuevas formas orientadas a la interactividad, velocidad de actualización de las noticias, el continuo flujo de la información y el abaratamiento de los costos. Dichos cambios han propiciado un vertiginoso avance en materia de edición de audiovisuales y noticias multimedia que a su vez son insertados en un proceso denominado “redacción digital integrada”, término utilizado para designar un sistema basado en el almacenamiento de la información en servidores centrales, que se maneja en el proceso de producción de programas informativos, ya sean textos, gráficos, imágenes o sonidos. Estos sistemas dotan a los editores o periodistas de herramientas con un alto nivel de configuración y facilidades de edición de noticias, así como también capacidades para el trabajo simultáneo desde diferentes terminales. (1)

La extensibilidad que propone esta tecnología ha llevado a grandes agencias de noticias y diversas empresas a invertir en el desarrollo de sistemas asociados al uso de las tecnologías digitales. La principal razón se basa en la diferencia entre la tecnología analógica y la digital, la cual radica fundamentalmente en la capacidad con que cuenta esta última de recibir imágenes, sonidos y datos, sin que estos se encuentren necesariamente asociados a la programación audiovisual. En dicha capacidad se apoya lo que

hoy se conoce como servicios interactivos, y gracias a los cuales se debe el avance acelerado de esta tecnología.

Dada la importancia de dicha tecnología, se hace necesario ejemplificar el uso de la misma mediante la presentación de algunos de los productos más exitosos o con mayor soporte a nivel internacional. Este es el caso de sistemas como: Sonaps -producto líder de Sony- que cuenta con un completo entorno en red que integra todos los aspectos del proceso de producción de informativos, desde la planificación, captación y edición, hasta la emisión, distribución y archivo; Multicon -una solución completa para la emisión automática de TV multicanal- que cuenta además con una *solución de digitalización global*, integrada con sistemas de producción, redacción y archivo como Avid®, Digition®, Estructure®, EVS®, Profile® y Tarsys®, productos también líderes en materia de edición.

En tanto, Cuba demostrando su capacidad de desarrollo de tecnologías informáticas orientadas al perfeccionamiento de sus propios medios de comunicación, y a la exportación de software calificados en el área de la televisión digital, produce, en el marco físico de la Universidad de las Ciencias Informáticas (UCI), el sistema PRIMICIA. Este sistema consta de características similares a los llamados sistemas de producción de noticias, pudiendo identificarse dentro del mismo un proceso parcial de redacción digital integrada.

Actualmente el producto PRIMICIA se encuentra en vías de perfeccionamiento. Si bien la última versión del mismo ofrece una interfaz sencilla para la edición y transmisión de las noticias, también es cierto que carece de elementos configurables que permitan la diversidad de visualización de los contenidos multimedia y las plantillas que dan soporte a los mismos. Por tal motivo se concibieron un conjunto de mejoras en la plataforma que solucionen las deficiencias visuales actuales para dar lugar a un proceso total de redacción digital integrada. Características arquitectónicas de la plataforma determinaron la aplicación de mejoras en los dos subsistemas que la estructuran. De ahí que partiendo de dicha situación se defina como **problema** de investigación:

¿Cómo lograr la integración entre el proceso de *redacción de noticias* y el *subsistema de transmisión*, usando como medio el formato XML de la Plataforma de Televisión Informativa PRIMICIA?

Tomando como punto de partida el problema planteado con anterioridad, se define entonces como **objeto de estudio**, el uso de ficheros XML como eje fundamental del proceso de "redacción digital integrada".

Para obtener un mayor ángulo de visualización y capacidad para integrar un prototipo bien diseñado, se hace necesario definir y profundizar en las áreas de aplicación sobre las que incide el prototipo en cuestión. De tal manera queda definido el **campo de acción**, en el uso de ficheros XML como medio de integración entre el proceso de redacción y el subsistema de transmisión de la Plataforma de Televisión Informativa PRIMICIA.

Se propone como **objetivo general** el desarrollo de un prototipo funcional que permita la integración entre el proceso de *redacción de noticias* y el *subsistema de transmisión* de la Plataforma de Televisión Informativa PRIMICIA.

Para dar cumplimiento al objetivo general del trabajo de diploma en cuestión se identificaron los siguientes **objetivos específicos**:

- ✓ Validar el formato XML propuesto para la Plataforma de Televisión Informativa PRIMICIA.
- ✓ Caracterizar los conceptos asociados a la solución planteada.
- ✓ Realizar el análisis y diseño del prototipo funcional encargado de gestionar los ficheros de configuración de noticias.
- ✓ Implementar el prototipo funcional.

Teniendo en cuenta los distintos enfoques investigativos planteados, se define como **idea a defender** que la implementación del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA, posibilitará la integración entre el proceso de *redacción de noticias* y el subsistema de transmisión de dicha plataforma.

Para dar cumplimiento a los objetivos propuestos se identificaron las siguientes **tareas de investigación**:

- ✓ Caracterizar el formato XML de la Plataforma de Televisión Informativa PRIMICIA.

- ✓ Describir el estado del arte asociado a la solución planteada.
- ✓ Definir una metodología adecuada para el desarrollo del prototipo funcional.
- ✓ Modelar el prototipo funcional encargado de la integración entre el proceso de redacción y el subsistema de transmisión de la Plataforma de Televisión Informativa PRIMICIA.
- ✓ Obtención del prototipo funcional.

Los **Métodos Teóricos** utilizados para dar cumplimiento a las tareas previstas son:

- ✓ **Analítico-Sintético:** Para identificar los conceptos y las definiciones más importantes relacionadas con la descripción de documentos que permitan luego generar una propuesta adecuada a la situación planteada, la tecnología estudiada y las exigencias del objeto de estudio.
- ✓ **Histórico lógico:** El método lógico para investigar el comportamiento y desarrollo general de los fenómenos en estudio.
- ✓ **Modelación:** Para representar por medio de diagramas, todos los procesos de gestión, obteniendo como resultado un mejor entendimiento de la posible solución a implementar. Se utilizará la modelación teórica, la cual tiene la capacidad de representar las características y relaciones fundamentales del fenómeno, proporcionar explicaciones y servir como guía para generar hipótesis teóricas.

Los **Métodos Empíricos** permiten extraer de los fenómenos analizados las informaciones que se necesitan a través de observaciones. Por consiguiente se aplicaron en el seguimiento de la presente investigación los siguientes:

- ✓ **Revisión de Documentos:** Para lograr un mayor conocimiento del proceso de descripción, lograr una mejor comprensión del desempeño y funcionamiento de distintas aplicaciones que existen actualmente y la manera en que operan, así como para determinar la forma de trabajo.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En el presente capítulo se realiza el estudio de los contenidos teóricos que sustentan la investigación en cuestión. De forma general se tratan conceptos ligados al ámbito de la investigación, se realiza el estudio de la situación problemática y se exploran soluciones previas.

1.2 Conceptos asociados al dominio del problema

1.2.1 Redacción Digital Integrada

Con el desarrollo de la tecnología de ordenadores, ha devenido una serie de grandes cambios en la historia del desarrollo científico humano. Gracias a los ordenadores se crearon nuevas vías para el desarrollo de la ciencia junto con todas sus vertientes, de forma tal que es posible mencionar que constituye la base del desarrollo científico-tecnológico actual.

La introducción de los ordenadores en la sala de *redacción*, y gracias a los avances de la tecnología digital, cambia el paradigma asociado a las formas estandarizadas de hacer periodismo. Con ello surge un nuevo concepto “redacción digital integrada” (1).

Las televisiones digitales de todo el mundo han adoptado sistemas digitales integrados, desterrando los ya obsoletos sistemas analógicos que generaban cuellos de botella en la cabina de edición y ralentizan el proceso de elaboración de las noticias. Las nuevas tecnologías digitales permiten el manejo de grandes bases de datos y redes, así como también agilizan el proceso de redacción e integración de la noticia con el sistema (2).

Desde hace ya tiempo las redacciones para televisión utilizan componentes digitales tales como los servicios de agencia, generación de grafismo por ordenador, los equipos DNG (Digital News Gathering) y los sistemas de edición no lineal. De tal forma la “redacción digital integrada” permite reunir todos los recursos asociados al proceso de redacción de la noticia, dígame texto, gráficos, fotografía, clips de video o sonido, y almacenarlos en servidores comunes, a los que tienen acceso todos los periodistas que

interactúan con el sistema.(1) Este proceso de interacción entre los redactores y las noticias se puede realizar desde cualquier terminal u ordenador, a través de los cuales se pueden editar los distintos recursos de forma no lineal, dígase edición de imágenes y sonido, redactar un texto ajustado a un tiempo, acceder a fuentes o archivos multimedia, etc. Para terminar el proceso se configura el listado de las noticias, organizado de acuerdo a la escaleta¹ del informativo, prevista para la emisión.

1.2.2 Librerías de software

Las bibliotecas o librerías de componentes de software son repositorios² donde se guardan o conservan documentos, ideas, fragmentos de código y todos aquellos componentes que participan en el desarrollo de un software, que puedan ser reutilizados, transformados o consultados. Se puede decir que tienen como objetivo asegurar la disponibilidad de activos, para apoyar el desarrollo de un producto de software (3). También permiten retener información relevante de cada componente tales como especificación técnica, clasificación, documentación, etc. Constituyen además una forma de acelerar los tiempos de programación, implementación, bajar costos y agregar funcionalidades utilizando soluciones ya probadas.

En todos los lenguajes de programación existen librerías de funciones que sirven para realizar diversas tareas a la hora de programar. Las librerías de los lenguajes de programación ahorran la tarea de escribir las funciones comunes que por lo general pueden necesitar los programadores.

1.2.3 Lenguaje de marcas extensible (XML)

El *Extensible Markup Language* (XML) o "Lenguaje de Marcas Extensible" es un metalenguaje -lenguaje que describe los datos y cómo estos se estructuran- mediante el cual, los desarrolladores pueden crear sus propios elementos para alcanzar sus propias necesidades de información (4). XML se usa fundamentalmente para crear meta vocabularios (conjuntos de etiquetas usados para representar elementos dentro de un documento XML) adaptados a las necesidades de cada empresa, entidad o industria.

¹ Se llama **escaleta** o **guión técnico** a un boceto, lista o minuta donde los productores escriben en orden, los pasos, segmentos o secciones que componen una noticia.

² Un **repositorio** es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

Un documento XML contiene solamente datos y etiquetas. Su objetivo es separar el contenido de la presentación del mismo, característica que le proporciona gran flexibilidad en el procesamiento y la visualización. La estructuración de datos usando XML permite acceder a datos a través de múltiples plataformas y aplicaciones, realizar búsquedas eficientes y utilizar diferentes "hojas de estilo" (XLS) para la visualización de datos mediante diversas interfaces (4).

El surgimiento del XML viene dado por el establecimiento del *SGML* (*Standard Generalized Markup Language*). Este último es un metalenguaje que permite definir lenguajes para determinar la estructura y el contenido de los documentos. La definición de la estructura y el contenido de un tipo de documento se realiza en las DTD³ ó en los XML esquemas⁴. En estas se definen los elementos que conformarán ese tipo de documentos y cómo tienen que estar organizados para que sea correcto.

XML no es ningún tipo de documento SGML, sino que es una versión abreviada de SGML optimizada para su utilización en Internet. Esto significa que con él se logran definir tipos de documentos personalizados, y por tanto se elimina la dependencia a un único documento HTML. XML ofrece el 80% de las ventajas del SGML con un 20% de su complejidad (5).

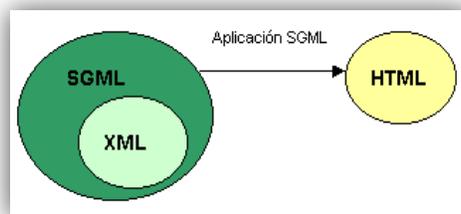


Figura 1: XML versión abreviada de SGML optimizada, HTML nace a partir de SGML

A continuación se enuncian las características principales del lenguaje:

³ **Document Type Definition (DTD)** se refiere a un documento concebido para la validación de estructura relacionada a un documento XML.

⁴ Un **esquema XML** es un lenguaje basado en el propio XML para la definición de estándares de documento.

- ✓ El XML no es un lenguaje de presentación, sino para describir datos, lo que facilita una búsqueda más precisa, permite a las empresas compartir datos de forma eficiente y hace la navegación más fácil.
- ✓ XML permite definir nuevas etiquetas y atributos en un documento donde las estructuras de contenido pueden anidarse a cualquier nivel de complejidad, lográndose un control absoluto sobre dichas estructuras y añadiendo atributos para incrementar el control de la información.
- ✓ La capacidad de separar la información (XML) de la (HTML) o alguna otra interfaz es atractivo para desarrolladores debido a que la información se puede mantener fresca en su propia base de datos y no en una página HTML estática. (4)
- ✓ Se pueden transportar enormes cantidades de datos bien organizados junto con descriptores que indican los elementos y la estructura de datos que contiene el documento.
- ✓ XML permitirá que los fabricantes eviten tener que desarrollar una GUI⁵ independiente para cada plataforma de administración de Red.
- ✓ Un documento XML puede contener una descripción opcional de su gramática para uso de las aplicaciones que necesiten realizar una validación de la estructura del mismo (6).

1.2.4 Definición de tipo de documento: DTD

Para mantener compatibilidad con SGML, el estándar XML mantiene el metalenguaje DTD de definición de lenguajes particulares de marcado. Las siglas DTD significan *Document Type Definition*, y se refieren a la definición de un tipo o esquema de documento.

La definición del tipo de documento puede hacerse:

⁵ La **interfaz gráfica de usuario**, conocida también como **GUI** (del inglés *graphical user interface*) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

- ✓ En un fichero separado, y poner la referencia en el DOCTYPE⁶.
- ✓ En el propio documento, dentro del DOCTYPE.
- ✓ Con una combinación de ambos métodos (6).

Mediante la DTD se define la estructura sintáctica que debe cumplir el documento XML, estructura que puede ser definida por un analista o programador. La definición del tipo de documento es como crear un lenguaje propio de marcado para una aplicación específica (5). El más claro ejemplo en el caso de la presente investigación, viene aparejado a la capacidad que brinda la DTD de definir la estructura sintáctica de las noticias que serán creadas por la Plataforma de Televisión Informativa PRIMICIA, pudiendo así crear un estándar de noticias aplicable a todos productos análogos que se desarrollan en el departamento de señales digitales. A partir de esta DTD se tendría una serie de elementos XML que permitirían definir noticias. La DTD define los tipos de elementos, atributos y entidades permitidas, pudiendo también expresar algunas limitaciones para combinarlos (4).

1.2.5 Esquemas XML

Un *esquema XML* es un lenguaje basado en el propio XML para la definición de estándares de documento. El mismo permite establecer reglas que brindan la posibilidad de restringir la estructura de los documentos XML que estén asignados a tal esquema. En este lenguaje los tipos de datos son clasificados en función de los elementos y atributos que contienen, elementos de tipo simple y complejo. Mediante estos grupos de especificación se permite determinar los tipos de datos válidos para cada elemento y atributo.

Los Esquemas XML superan muchas de las limitaciones y debilidades de los DTDs, tal es el caso del uso de “espacios de nombres”, lo cual permite definir elementos con igual nombre dentro del mismo contexto, siempre y cuando se anteponga un prefijo al nombre del elemento. El simple hecho de que un esquema de XML sea un documento XML soluciona otra de las restricciones que poseían los DTDs (7). Un

⁶ Una declaración de tipo de documento o **DOCTYPE** asocia un documento SGML o XML particular con una definición de tipo de documento.

esquema por si solo puede describir una clase de documentos, del cual puede haber varias instancias. Esta analogía es bien parecida a la existente entre clases y objetos en sistemas con tecnología orientada a objetos.

Una de las ventajas del uso del esquema XML es que al restringir el contenido de los ficheros XML posibilita intercambiar información entre aplicaciones con gran seguridad, disminuyendo así el trabajo de comprobar la estructura de los ficheros y el tipo de los datos. Además las definiciones de esquema pueden ser procesadas mediante lenguajes de parseado⁷ tales como DOM, SAX, XPath y XSLT, debido a esto han aparecido gran variedad de herramientas para el manejo de esquemas.

1.2.6 Estándares del XML

Para que los ordenadores puedan trabajar conjuntamente necesitan hablar el mismo lenguaje de manera tal que se logre una integración impecable entre las capas de aplicación. Muchas empresas han invertido en el desarrollo de estándares XML que resuelvan problemas cotidianos de comunicación entre aplicaciones de distintos cortes y funcionalidades.

El uso de estándares en el proceso de construcción de soluciones informáticas es muy importante. Su uso garantiza el correcto funcionamiento e integración de infraestructuras productivas implementadas en diferentes lenguajes y plataformas. XML por su parte, está provisto de los medios necesarios para la creación de estándares para la integración de sistemas con amplias prestaciones. Así pues, el mismo ha sido objeto para la creación de varios lenguajes estandarizados y globalmente utilizados. Algunos de los estándares más comunes son:

- ✓ RDF (*Resource Definition Format*) describe el contenido de sitios WEB mejorando su calificación (4).
- ✓ VML (*Vector Markup Language*), PGML (*Precision Graphics Markup Language*) ambos describen gráficos vectoriales.

⁷ El **parseado** no es más que el análisis de una oración o expresión en un lenguaje, en términos de sus constituyentes gramaticales, identificando las partes y sus relaciones sintácticas.

- ✓ ICE (*Information and Contents Exchange*) es un protocolo que servirá para syndicar datos entre varios sitios WEB.
- ✓ SMIL (*Synchronized Multimedia Integration Language*) proporciona un medio para enlazar textos, audio, gráficos y animaciones (1).

1.3 Objeto de Estudio

1.3.1 Descripción General

A lo largo de los últimos años la mayoría de las más importantes agencias de noticias han encontrado en el XML una vía para el desarrollo de la industria periodística debido a las capacidades que el mismo ofrece para la integración de aplicaciones y diversas tareas de configuración. La creciente necesidad de integrar aplicaciones tanto dentro de la empresa como con el resto de las organizaciones análogas ha promovido el lenguaje estándar de XML (*Extensible Markup Language*), considerado ya como la panacea del periodismo digital (1).

Al abordarse una investigación sobre la utilización de los estándares de XML en el desarrollo de plataformas de televisión, la documentación referente a las soluciones propuestas son escasas, llegando a los límites de la desinformación. Por tal motivo se hace necesario establecer nuevas líneas de investigación en las cuales se profundice en el uso de las tecnologías que soportan los medios de comunicación.

A nivel mundial muchas de las agencias de noticias basan la configuración de las noticias editables en metalenguajes basados en XML. Como tal en el periodismo tanto general como digital, tienen como unidad informativa fundamental la noticia con sus características asociadas. En sentido general el formato XML es ideal para describir la estructura de las noticias, proveyendo soporte a las mismas mediante las DTD o XML esquema, garantizando su integración con otras aplicaciones que usen el mismo estándar. La ventaja fundamental para la integración usando dicho lenguaje reside en que las distintas empresas o productos utilicen un mismo estándar de noticias. Por tal motivo muchos grupos han comenzado a trabajar en la creación de estándares generalizados de noticias que garanticen toda integración posible.

El proceso de redacción en sistemas informativos automatizados comienza desde la obtención de los recursos multimedia y la concepción de la historia, hasta la edición de la misma. Con la obtención de noticias y las funciones de búsqueda, tareas de asignación, investigación de historias y edición de los materiales audiovisuales, los profesionales de las noticias pueden controlar el sistema completo desde su ordenador. Las características más generales del proceso de redacción de estos sistemas son:

- ✓ *Precisión y rapidez en la obtención de la noticia y sus recursos asociados*, lo que permite a los redactores de noticias recopilar información de diferentes fuentes, como señales estándar, telnet, correo electrónico, espacios de red compartidos con grandes bases de datos y servicios que contienen todo lo necesario para conformar la noticia (2).
- ✓ *Capacidad de colaboración y espacios de trabajo compartido*, logrando que varios profesionales utilicen concurrentemente los mismos recursos multimedia en bruto para la elaboración de la noticia y en muchos casos también el trabajo concurrente en una misma noticia o escaleta de ellas.
- ✓ *Edición de todos los elementos de la noticia*, lo que posibilita la edición de textos e imágenes, hacer cortes en videos, sincronización de los recursos utilizando *una línea de tiempo (timeline)*⁸ y utilización de efectos de videos (1).
- ✓ *Digitalización de la noticia en diferentes formatos*, permitiendo una vez creada y editada la noticia, obtenerla en un formato digital (mpg2, avi, flv ó formato propio de la plataforma) (8).

Si lo antes expuesto es llevado a la práctica, es posible definir la noticia como un contenedor de información multimedia integrado por elementos como: **Texto** (que puede estar descrito por parte de los redactores en diferentes formatos, como HTML); **Imágenes** (que pueden tener formato GIF, JPEG, TIFF); **Sonido** (en formato AVI, AU, WAV) o **Vídeo** (AVI, MPEG), etc. La integración de todos estos elementos pertenecientes a una misma noticia da lugar a una misma **unidad informativa (fichero de noticia)** (1), en el cual se almacena información clave de configuración que permita ser gestionada mediante un gestor de contenidos que referencie los recursos en base de datos.

⁸ La *línea de tiempo* ó *timeline* organiza y controla el contenido de una película a través del tiempo.

Una vez seleccionado el estándar de noticias en XML, se realiza el proceso de presentación de la noticia. Existen numerosas vías para la presentación de las noticias, basadas en lenguajes de visualización en el cliente. Lenguajes como JavaScript, PHP/XHTML, Java, .Net y XML/XSL, permiten el parseado⁹ de los documentos en XML para obtener una información presentable.

1.3.2 Descripción actual del dominio del problema

A nivel mundial existen numerosos *sistemas de producción de noticias* que incluyen un proceso de “redacción digital integrada” total o parcialmente configurable. Todo esto originado por la creciente competencia entre productos de alta tecnología dedicados a la divulgación de noticias. Dichos procesos son estructurados en diversos sistemas con amplio soporte, basado en la mayoría de los casos en tecnología propietaria.

El dilema de monopolización de estos sistemas de alta tecnología supone un problema que enfrentan actualmente los medios televisivos de Cuba. Por tal razón, el país ha dirigido esfuerzos en el desarrollo de sus propias herramientas o sistemas para la gestión de sus propios medios de comunicación. De esa forma en el marco de la Universidad de las Ciencias Informáticas (UCI) se inician una serie de proyectos cuyo objetivo no es más que proveer las herramientas para dar soporte a la tecnología digital en nuestro país.

Entre los principales sistemas desarrollados en la Universidad de las Ciencias Informáticas (UCI), se encuentran *Señal 3* y *Señal ACN* -administrado por la Agencia Cubana de Noticias (ACN)-. Estos sistemas o productos han solucionado los principales problemas identificados en su área de aplicación, logrando la transmisión de noticias televisivas de una forma organizada y automatizada (8). De ahí que a partir de las experiencias acumuladas y la necesidad de un sistema integral de televisión informativa, se desarrolle el producto informático PRIMICIA.

Hoy en día PRIMICIA es un producto completamente desarrollado, y cuenta con numerosas funcionalidades genéricas basadas en la experiencia alcanzada por el grupo de desarrolladores durante

⁹ El **parseado** no es más que el análisis de una oración o expresión en un lenguaje, en términos de sus constituyentes gramaticales, identificando las partes y sus relaciones sintácticas.

varios años. Sin embargo, el desarrollo del producto estuvo sujeto a una escasa variedad práctica lo cual devino en poca generalidad de las funcionalidades integradas al mismo.

1.3.3 Situación Problemática

El panorama de desarrollo de los medios de comunicación y la globalización asociada a estos, impone un nuevo reto a las agencias periodísticas. Desde hace algunos años, la televisión, en la nueva era digital, se ha convertido en uno de los medios de difusión masiva de información con mayor alcance a nivel global. Es entonces que muchas empresas adoptan como una variante de distribución de las noticias al medio audiovisual digitalizado. De ahí el nacimiento de los *sistemas de producción de noticias*, cuyo objetivo no es más que proveer soluciones informáticas que haciendo uso por lo general de redes de televisión interna o envió de la señal vía satélite, administran la transmisión de un canal de televisión (9).

La creciente competencia entre los *sistemas de producción de noticias*, ha devenido en el desarrollo de las funcionalidades que los componen. Estas funcionalidades se centran en la rapidez, sencillez y alcance de los procesos de edición y transmisión de las noticias.

En Cuba se cuenta con una serie de sistemas que tributan a una solución integral dedicada a proveer un canal de televisión informativa completamente automatizado, cuyo nombre es PRIMICIA.

PRIMICIA es capaz de proveer un canal de televisión informativa que integra informaciones en formato texto, imagen, audio y video, posibilitando la transmisión automática de las informaciones de acuerdo a las publicaciones realizadas a través de la administración del sistema. El sistema fue desarrollado sobre plataformas libres y está compuesto por dos subsistemas: *administración* y *transmisión*. Mediante la administración del sistema es posible gestionar los usuarios de la plataforma, la transmisión del canal, los recursos multimedia para la confección de las noticias y permite llevar a cabo los procesos de redacción y publicación de la misma. Por su parte el subsistema de transmisión es el encargado de visualizar las noticias publicadas durante el proceso editorial (8).

Como objetivo actual de la Plataforma de Televisión Informativa PRIMICIA se perfila su distribución en diversas entidades, que a su vez cuenten con características diferentes. Este es el caso de: empresas, centros de convenciones, hospitales, hoteles, terminales, universidades, ministerios, y agencias de

prensa. El funcionamiento interno de cada entidad exige características visuales específicas para la distribución de las noticias. Esto trae como consecuencia que el proceso de redacción y la forma en que se presenta la noticia en el momento de la transmisión también varíe de un medio a otro.

Sin embargo este proceso se caracteriza por ser poco flexible y esquemático ya que la plataforma no le permite al redactor interactuar con los elementos visuales de la noticia. Por ejemplo, mediante el proceso de redacción no se le permite al redactor definir nuevos elementos para la presentación de la noticia, o bien no puede cambiar ni eliminar los que vienen definidos por defecto. Esto es debido a que las noticias están compuestas por pantallas y estas pantallas son de solo cuatro tipo: texto, texto-imagen, imagen y video, por lo que cualquier noticia que combine otros dos recursos actualmente no se puede elaborar. Por último otra deficiencia es la imposibilidad por parte del redactor de configurar el tiempo que se visualizará la pantalla, este valor es estático y se define internamente durante el desarrollo de la aplicación (8).

En pos al enriquecimiento y desarrollo del producto PRIMICIA se identifican entonces una serie de funcionalidades generales que serán integradas a la plataforma. Estas son:

- ✓ Edición de los elementos y recursos multimedia de la noticia.
- ✓ Incorporación de efectos de animación a las noticias.
- ✓ Pre visualización en todo momento de la edición en curso.

Según lo antes expuesto queda declarado como objetivo de esta investigación, la solución de una parte de la problemática asociada a la integración entre el proceso de redacción y el subsistema de transmisión de la Plataforma de Televisión Informativa PRIMICIA. Dentro del mencionado proceso de integración se planea profundizar en los métodos de interoperabilidad existentes que proporcionan una solución coherente y aplicable al marco de trabajo de la plataforma en cuestión.

1.4 Análisis de otras soluciones existentes

En los últimos años el metalenguaje de marcado XML ha alcanzado su mayor grado respecto al desarrollo de la industria periodística debido a las capacidades que el mismo ofrece para la integración de aplicaciones y diversas tareas de configuración. Este lenguaje ha sido objeto de las mejores soluciones en cuanto a *sistemas de producción de noticias* se trata. Así pues, en el presente punto se analizan algunas

de las soluciones existentes en los distintos ámbitos, teniendo en cuenta que las mismas sean lo más análoga posible a la referenciada en la presente investigación.

1.4.1 Ámbito Internacional

A nivel internacional existen numerosas soluciones basadas en el desarrollo de *sistemas de producción de noticias*. Muchas de estas se encuentran bajo la tutela de tecnología propietaria y por lo tanto no ha sido liberada la documentación correspondiente a las mismas. Sin embargo existen varias soluciones bien documentadas que sentarán las bases del presente estudio.

Este es el caso de Atos Origin, un sistema especialmente diseñado para entornos editoriales, una innovadora solución a las necesidades que presenta la gestión editorial en la nueva era. **D&M Press** (10) soporta el flujo de trabajo completo desde la maquetación hasta la gestión e impresión inteligente de ediciones, incluyendo la integración con sistemas crossmedia¹⁰ y con sistemas de gestión de contenidos.

D&M Press es una solución global dividida en un grupo de aplicaciones que cubren los diferentes ámbitos de las publicaciones. En particular interesa la herramienta **ProToWeb** –Herramienta de publicación web-, la cual constituye la versión del sistema que se encarga de la configuración y visualización de los contenidos mediante la web (10). Esta herramienta cuenta con una serie de características interesantes que deben ser analizadas en el desarrollo de la presente investigación.

Características fundamentales de la herramienta **ProToWeb**:

- ✓ Exportación XML de todo el bloque de información correspondiente a cada noticia a publicar en la web, o de toda la página.
- ✓ Almacenamiento y gestión básica de los contenidos a publicar y publicados.
- ✓ Definición de la plantilla a utilizar, de la estructura de la edición digital y de la correspondencia de secciones entre los dos soportes.

¹⁰ **Crossmedia** es la integración entre las tres plataformas de distribución existentes "Gráfica-Web-Video" logrando así un flujo de trabajo de mayor eficiencia.

- ✓ Publicación de las noticias de forma individual, por sección o para todo el periódico.
- ✓ Visualización y edición de las noticias incluidas en la edición digital.
- ✓ Posibilidad de incorporar noticias exclusivas de la edición digital.

De forma general, estos sistemas en función de superar las limitaciones que ofrecen las soluciones tradicionales, se encargan del desarrollo de servicios integrales que comparten un núcleo común especializado a cada uno de los diferentes tipos de terminal (WEB, WAP, i-Mode, Web-TV, Voice-XML). Sobre este núcleo se desarrollan servicios que presentan un punto común: permitir el acceso a los recursos, de forma tal que puedan ser integrados mediante archivos de configuración XML con un sistema que presente la estructura de las noticias.

Como punto común, muchos *sistemas de producción de noticias* basan sus soluciones en un motor XML como centro de edición, elaboración y visualización. Sus amplias ventajas ya mencionadas en apartados anteriores de este capítulo han devenido en diversas soluciones para diferentes escenarios. Sintetizando, las capacidades fundamentales del motor XML se centran en las siguientes funcionalidades:

1. Transformación de contenidos XML aplicando una cierta hoja de estilo XSLT. El objeto de esta transformación es doble:

- ✓ Adaptación a dispositivo de salida. El Motor XML cuenta con una serie de plantillas XSL que facilitan la generación de salidas en diferentes formatos en función del dispositivo. El motor tiene un módulo de reconocimiento del dispositivo, en función de la cual genera una salida adaptada a dicho dispositivo (10).

2. Administración del sistema. El Motor XML cuenta con un BackOffice de administración que permite las siguientes funciones:

- ✓ Gestión de plantillas. El Motor XML provee una interfaz para cargar y gestionar plantillas XSL con el fin de generar salidas adaptadas a diferentes terminales. En la implementación actual se cuenta con plantillas para PC, móvil (WAP e I-MODE), PDA, televisión y voz.

- ✓ Gestión de idiomas. A través de la herramienta de administración se puede definir qué idiomas soporta un servicio.
- ✓ Gestión de dispositivos. Permite administrar los dispositivos soportados y registrados en la plataforma, y especificar cuáles de ellos tiene implementado cada uno de los servicios.
- ✓ Gestión de contenidos estáticos. Permite el upload¹¹ y almacenamiento de contenidos estáticos de los servicios, tales como imágenes y elementos multimedia (10).

Esta arquitectura basada en XML aporta gran flexibilidad y un alto nivel de personalización tanto de las interfaces como de la estructura visual de la noticia mediante el uso de la combinación XML/XSL que se encarga de gestionar la personalización con gran rapidez y efectividad.

1.5 Síntesis Temática

En el epígrafe anterior se abordaron soluciones presentes a nivel internacional y los usos fundamentales prestados al metalenguaje XML dentro de las propuestas. Sin embargo en el marco nacional no se cuenta con experiencias de desarrollo en dicho campo que posibiliten marcar un camino a seguir en el perfeccionamiento de la plataforma. Por ende en el presente punto se explican las principales diferencias entre el modelo de diseño a seguir por el producto PRIMICIA y los modelos internacionales de sistemas digitales informativos.

Si bien los modelos internacionales plantean una solución eficiente, extensible a distintos dispositivos y medios de comunicación, la Plataforma de Televisión Informativa PRIMICIA aboga por un diseño mucho más dinámico y orientado a la televisión, donde el usuario es exigente en cuanto a la apariencia de la noticia. Una de las principales diferencias entre el modelo internacional y el modelo de diseño a seguir por PRIMICIA es que este basa su proceso de transmisión en el modelo broadcast¹², por lo cual no se hace necesaria una personalización a nivel de usuario sino a nivel de noticia. De esa manera es posible

¹¹ **Upload** es transmitir datos de un ordenador a un foro, red

¹² **Broadcast** (informática), transmisión de un paquete que será recibido por todos los dispositivos en una red.

descartar el modelo basado en XSL o XSLT (lenguaje de transformación adherido a la estructura XSL) y abogar por modelos de interpretación basados en lenguajes de parseado.

Así pues, de los proyectos relacionados que sí han optado desde un principio por la utilización de estándares XML en sus desarrollos, la gran mayoría ha escogido usar tecnologías como SAX, DOM o similares como XSL/XSLT. Cada tecnología presenta diferencias importantes en cuanto a potencia, eficiencia y nivel de abstracción. En el caso de SAX, se plantea una tecnología con un bajo nivel de abstracción pero eficiente para tareas sencillas sobre datos voluminosos. También está el caso de XSLT, un lenguaje de especificación de transformaciones, que pese a su aceptación resulta sofisticado y tedioso cuando las reglas de transformación adquieren complejidad (bucles, variables tipadas¹³, condicionales y expresiones). Por último DOM resulta ser la técnica más potente, pero también requiere un mayor conocimiento y experiencia.

Según lo antes expuesto, tomando como punto de partida la existencia de frameworks¹⁴ que potencian el trabajo y la sencillez de la tecnología DOM, es posible plantear entonces la utilización de la misma. Actualmente muchos de los lenguajes tradicionales de implementación de tecnologías web poseen soporte para DOM. La tarea es definir entonces el lenguaje que provee las mejores prácticas de integración en el desarrollo del prototipo en cuestión

1.6 Conclusiones

En el presente capítulo se abordaron los principales métodos estructurales de desarrollo para sistemas de producción de noticias, así como el papel que desempeñan en la actualidad las tecnologías basadas en XML y sus estándares asociados. También se abordaron los conceptos fundamentales de cada una de las tecnologías que soportan la investigación, así como la caracterización de algunas de ellas.

¹³ El término **variables tipadas** se refiere a la capacidad que poseen los lenguajes de programación de declarar el patrón de valores que recibe una variable, usualmente asociado a una definición interna del lenguaje o personalizada por el programador.

¹⁴ **Framework** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Capítulo 2. Tendencias y tecnologías actuales.

2.1 Introducción

En este capítulo se realiza un análisis de las tendencias y tecnologías de desarrollo actuales entorno a la solución. Se abordan conceptos y presentan los estudios que soportan cada propuesta tecnológica y herramienta seleccionada.

2.2 Lenguajes de programación propuestos.

Un lenguaje de programación no es más que un conjunto de símbolos, reglas sintácticas y semánticas, mediante las cuales el programador puede interactuar con el hardware para obtener un resultado observable. Existen un gran número de lenguajes, pero ninguno se puede considerar el más capacitado o el más adecuado para creación de aplicaciones. En el presente epígrafe se abordan los lenguajes propuestos para dar solución a los objetivos planteados y los indicadores que posibilitaron su selección.

2.2.1 Lenguaje PHP (Hypertext Pre-processor).

PP acrónimo de “PHP Hypertext Pre-processor”, es un lenguaje multiplataforma por lo que puede ser ejecutado en la mayoría de los sistemas operativos, como es el caso de Windows, Mac OS, Linux, Unix, y otros. Este lenguaje se utiliza en la generación de páginas Web dinámicas muy robustas, brindando la posibilidad de conectarse a diferentes servidores de bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.

Otras ventajas que caracterizan a este lenguaje son:

- ✓ Independencia del navegador, pues como se ejecuta en el servidor no necesita que éste lo soporte.
- ✓ Logra la integración con varias bibliotecas externas, lo que permite desde generar documentos en PDF (Acrobat Reader) hasta generar y analizar código XML.
- ✓ Rápida ejecución utilizando poca memoria, debido a que está completamente escrito en C.

- ✓ Portabilidad: PHP está disponible para una gran cantidad de sistemas operativos diferentes. Se puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix, como Linux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIX o en las diferentes versiones de Microsoft Windows (11).

2.2.2 Lenguaje de modelado UML

El Lenguaje Unificado de Modelado (UML), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas” (12).

UML se ha convertido en el verdadero estándar global para representar y modelar la información con la que se trabaja en las fases de análisis y de diseño. 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.

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura de hardware donde se ejecuten. Otro objetivo de este modelado visual es lograr la independencia del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

Los principales objetivos de UML son:

- ✓ **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.

- ✓ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados (13).
- ✓ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

2.3 Metodologías de desarrollo de software.

Durante la construcción de todo sistema o proyecto de software se corren riesgos y se presentan situaciones que en muchos casos se hacen difíciles de controlar. Como respuesta a semejante problema, se hicieron estudios que devinieron en la creación de las *metodologías de desarrollo de software* y en diversos estándares. Dichas metodologías y estándares proporcionan las guías para un desarrollo controlado y basado en soluciones exitosas, de forma tal, que gane en agilidad el proceso y en calidad el producto (14). Formalmente se pueden definir estas metodologías como un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software (15).

No obstante, cada software cuenta con necesidades y entornos diferentes, que deben tenerse en consideración. Por otra parte cada metodología tiene características propias del ambiente de desarrollo, constituyendo estos los principales aspectos que rigen el proceso de selección de la metodología adecuada. Una metodología tiene que ser usada de forma simple y efectiva, sin hacer las tareas de diseño excesivamente complejas (15).

La comparación y clasificación de metodologías no es una tarea fácil. La diversidad de propuestas, diferencias, información y alcance de cada una, complica el proceso de selección. A grandes rasgos se definen dos clasificaciones muy generales en las que se encapsulan las metodologías existentes: metodologías estructuradas y metodologías orientadas a objetos. Por otra parte, considerando la filosofía de desarrollo se definen dos nuevas clasificaciones comúnmente conocidas como: las metodologías ágiles y las pesadas.

Las metodologías ágiles nacen para solventar la necesidad de un desarrollo de software rápido y con la calidad requerida. Estas son una agrupación de las prácticas tradicionales de producción de código pero llevadas al extremo, mediante las cuales se buscan entregas rápidas y oportunas hasta la entrega final del

producto. Su mayor fortaleza se evidencia en la resistencia a los cambios sin un incremento radical del costo inicial de producción (16).

Por otro lado las metodologías tradicionales ó pesadas cuentan con un proceso mucho más controlado. Están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo y se centran en el alcanzar una arquitectura estable. No obstante se ven altamente afectadas por los cambios, tienen altos costos de producción y mucho más trabajo de documentación (15).

2.3.1 Fundamentación de la selección

En la actualidad no es recomendable realizar un desarrollo software de una manera lenta. Las exigencias de los clientes actuales conllevan a la necesidad de implementar soluciones rápidas y que cumplan con los requerimientos planteados, por lo que el Desarrollo Rápido de Aplicaciones (RAD) es una de las características que más impacto tiene en la actualidad. Para solventar esto se deben utilizar herramientas basadas en este nuevo enfoque.

En el caso particular de RUP, por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto mediante su configuración previa a aplicarse, realizando una configuración adecuada, podría considerarse ágil. Por lo tanto utilizar una metodología RUP, en la que se definan claramente los objetivos de cada fase, los entregables y sus actividades permitiría tener un entorno predictivo, en el que se agilicen las prácticas y técnicas necesarias para llevar a cabo la integración continua manteniendo diseños simples.

Según lo antes expuesto y teniendo en cuenta la experiencia en el uso de las tecnologías antes descritas, se acordó el uso de la metodología RUP orientándose a un desarrollo configurado que agilice el proceso de desarrollo.

2.3.2 Proceso Unificado de Rational (RUP; Rational Unified Process)

Las siglas RUP en inglés significan *Rational Unified Process* (Proceso Unificado de Rational). Este es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción

del software de alta calidad que resuelva las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.

La metodología RUP proporciona disciplinas en las cuales se encuentran artefactos con lo cual se podrá contar con guías para poder documentar e implementar de una manera fácil y eficiente, todas las guías para un buen desarrollo, todo esto dentro de las respectivas fases con las cuales cuenta (15).

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software (14).

Cada fase en RUP puede descomponerse en iteraciones. Una *iteración* es un ciclo de desarrollo completo dando como resultado una entrega de producto ejecutable (interna o externa).

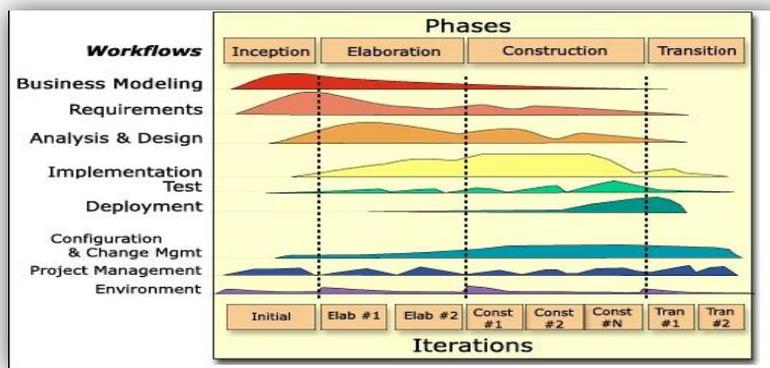


Figura 2: Ciclo de vida de RUP

Características de RUP, metodología centrada en:

- ✓ Realizar un levantamiento exhaustivo de requerimientos.
- ✓ Buscar detectar defectos en las fases iniciales.

- ✓ Reducir el número de cambios tanto como sea posible.
- ✓ Realizar el análisis y diseño, tan completo como sea posible.
- ✓ Diseño genérico, intenta anticiparse a futuras necesidades.
- ✓ El cliente no pertenece al grupo de desarrollo.

2.4 Herramientas de Modelado de Software

Las herramientas CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Ordenador), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

De acuerdo con Kendall y Kendall¹⁵ la ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas (19).

2.4.1 Fundamentación de la selección

Actualmente **Visual Paradigm** constituye una de las herramientas de modelado más utilizadas en el mundo. La misma es una herramienta completamente integrada al entorno de desarrollo Eclipse SDK. Soporta los últimos estándares de Java y UML. Provee mecanismos de re-ingeniería o ingeniería inversa y generación de código. Por tales razones se seleccionó dicha herramienta.

¹⁵ *Kendall y kendall es un libro que presenta de manera precisa los métodos, herramientas y técnicas de desarrollo de sistemas con un toque humorístico y fácil de entender*

2.4.2 Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (20).

La herramienta también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Está diseñada para dar soporte a arquitectos de sistemas, diseñadores, desarrolladores, analistas de procesos de negocio y modeladores de datos en los procesos de desarrollo de software.

Características principales:

- ✓ Modelar procesos de negocio.
- ✓ Administrar requerimientos.
- ✓ Importar archivos desarrollados con Rational Rose.
- ✓ Importar y exportar archivos XML.
- ✓ Generar código e ingeniería inversa.
- ✓ Generar una capa Objeto- Relacional fiable, escalable, y de alto rendimiento. Modelar visualmente el diseño lógico y físico de datos.
- ✓ Automatizar el mapeo entre el modelo de objetos y el modelo de datos.
- ✓ Soporta una amplia gama de bases de datos donde se incluyen: Oracle, Microsoft SQL Server, PostgreSQL, MySQL y otros. Se puede integrar a los principales IDEs: Eclipse, Borland JBuilder, NetBeans, Sun ONE, IntelliJ IDEA, Oracle JDeveloper, BEA WebLogic Workshop. Es multiplataforma, disponible para los Sistemas Operativos Linux, Windows, y Mac OS.

2.5 Entorno de Desarrollo Integrado

Integrated Development Environment (entorno de desarrollo integrado), no es más que un editor de código que además puede servir para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación (21). Una de las tareas más notables de cualquier desarrollador, lo constituyen la selección del lenguaje y entorno de desarrollo. En muchos casos esta selección define el propio alcance y calidad del proyecto final. En el presente epígrafe se presentan los fundamentos tomados en cuenta para la selección del entorno de desarrollo según las capacidades de integración con los lenguajes de implementación propuestos. Dígase soporte para el lenguaje PHP y capacidades para el trabajo con XML.

2.5.1 Fundamentación de la selección

Eclipse constituye el entorno de desarrollo mayormente integrado a las características del prototipo en cuestión. Eclipse es un IDE (Entorno de Desarrollo Integrado) tan potente como popular, que incorpora diversas herramientas para el diseño integradas al espacio de trabajo del mismo.

Entre las características que sentaron las bases de la selección figuran:

- ✓ Herramienta Open Source y libre de licencias privativas.
- ✓ Entorno completamente integrado a la herramienta de modelado Visual Paradigm, plugin que brinda muy nutridas herramientas de desarrollo.
- ✓ Extensos frameworks y APIs que permiten a los desarrolladores fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP.
- ✓ Integración con el entorno de desarrollo Aptana Studio. Herramienta con amplio soporte para Javascript.

2.5.2 Eclipse PDT

Eclipse es una poderosa herramienta que permite integrar diferentes aplicaciones para construir un entorno integrado de desarrollo (IDE). Es un proyecto de desarrollo de software de código abierto,

construida en base a plugins¹⁶. Este mecanismo permite desarrollar, integrar y correr nuevas funcionalidades garantizando extensibilidad e independencia del tipo de contenido o lenguaje (21).

Entre los añadidos para proveer nuevas utilidades al programa enfocadas a diversos usos se encuentra un módulo para programación en PHP. Dicho módulo es conocido como PDT (PHP Development Tools), en el cual trabaja para proveer un IDE completamente funcional para PHP.

PDT (PHP Development Tools), es un proyecto con gran repercusión entre los desarrolladores de PHP. Entre sus principales características se encuentran:

- ✓ Editor sensible al contexto, el cual provee de exaltamiento de código, asistente de código y autocompletado de código.
- ✓ Integración con el modelo del proyecto Eclipse, que permite para inspeccionar el uso de las vistas del contorno del fichero y del proyecto, así como la nueva vista PHP Explorer.
- ✓ Soporte para el debug¹⁷ incremental del código de PHP
- ✓ Extensos frameworks y APIs que permiten a los desarrolladores fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP.
- ✓ Soporte para el desarrollo de soluciones basadas en XML.

2.6 Conclusiones

En este capítulo se realizó el análisis de las tecnologías y herramientas más actuales en el desarrollo de sistemas basados en estándares web y soporte XML. En el mismo se presentaron los estudios que sirvieron de base para su selección, así como también se profundizó en algunas de las áreas de interés para la investigación.

¹⁶ Un **plug-in** es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

¹⁷ **Debug** no es más que la capacidad de controlar la ejecución de un programa o un código en cierto lenguaje de programación.

Capítulo 3. Presentación de la solución propuesta

3.1 Introducción

En el presente capítulo se muestran las distintas investigaciones que soportan el desarrollo de la solución propuesta. En el mismo se presenta la caracterización actualizada de los elementos configurables que conforman el formato XML de la Plataforma de Televisión Informativa PRIMICIA. Se comienza el proceso de negocio y exploración de las características o viabilidades del sistema, y se culmina con la descripción de todos los artefactos generados mediante los procesos antes mencionados.

3.2 Descripción de la solución propuesta.

En el presente epígrafe se abordan conceptos asociados al dominio de la solución en cuestión. Como propuesta de soluciones técnicas de este trabajo se propone entonces, el desarrollo de una Interfaz de Programación de Aplicaciones (API) que permita generar noticias configuradas mediante el formato XML de la plataforma PRIMICIA. Dicha API se encarga de proveer un medio sencillo de interacción entre el espacio visual de la noticia sostenido por el sistema de redacción y el fichero de configuración en formato XML de la propia noticia. Es decir, se brinda un medio altamente escalable y flexible que facilite las tareas de configuración de los elementos visuales de la noticia, para posteriormente generar el fichero de configuración en formato XML.

3.2.1 XML como formato para el almacenamiento de datos

El almacenamiento de la información que genera la aplicación se estructuró un fichero global que contiene toda la configuración de la noticia. Como formato para el almacenamiento de dicha información se escogió el estándar XML. Dicho formato permite la integración entre aplicaciones a distintos niveles, además de ser un formato estandarizado para archivar información de cualquier índole. Como lenguaje provee extensibilidad, y una extraordinaria capacidad de validación mediante las DTD y XML esquemas. En el presente caso se hizo uso de la extensibilidad y flexibilidad del mismo para elaborar un estándar de noticias que permita la fácil integración y configuración de noticias entre los módulos fundamentales de la plataforma.

3.2.2 Fichero XML de configuración de noticia. Estructura Básica.

En el presente sub-epígrafe se aborda el conjunto de elementos que componen la jerarquía de la noticia. Se presenta información gráfica y textual con el objetivo de lograr un mayor entendimiento de la estructura básica de la noticia, y se culmina con la definición del papel que desempeña cada uno dentro de dicha estructura.

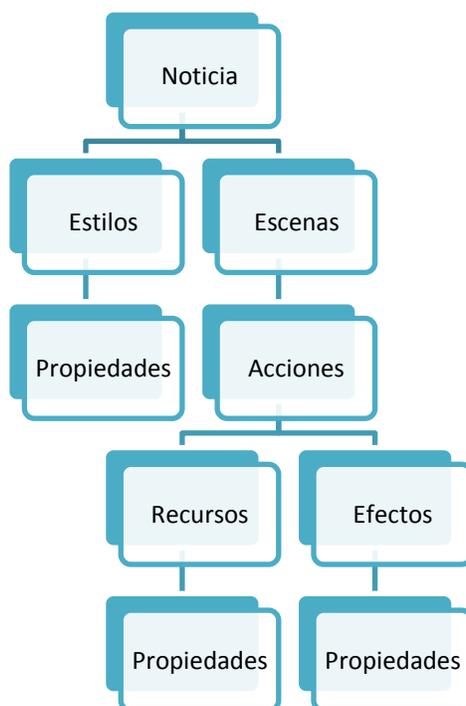


Figura 3: Jerarquía de elementos que conforman en formato XML de la plataforma PRIMICIA

Nodo del XML	Definición
Noticia	Nodo padre en la Jerarquía del XML, sus nodos hijos son las Escenas y Estilos, constituye el elemento que engloba toda la configuración.
Escenas	Constituye un elemento contenedor de un grupo de Acciones. Su papel principal es englobar las Acciones presentes en un

	instante de tiempo. Potencia la reutilización de componentes.
Estilos	Constituyen un elemento global que se utilizará para almacenar los posibles Estilos aplicables a los componentes texto presentes en la noticia. La declaración global de este elemento se realizó por motivos de reutilización.
Acciones	Constituye un elemento de documento XML y presenta como su principal atributo el tiempo en que se inicia la Acción. Tiene como elementos hijos un Recurso (de carácter obligatorio) y puede tener o no, un Efecto asociado.
Efectos	Efecto de animación de las Acciones. Este elemento engloba también un conjunto de atributos que dependen del efecto visual en cuestión.
Recursos	Este elemento representa todos los elementos configurables de la noticia.
Propiedades	Representa todas las propiedades de los elementos configurables, como su nombre lo indica no representa un elemento sino los atributos de estos.

Tabla 1: Descripción de la jerarquía de elementos que componen el formato XML de la plataforma PRIMICIA

3.2.3 Aspectos Generales

Para exportar el formato XML de noticias de la Plataforma de Televisión Informativa PRIMICIA se hará uso de las facilidades que brinda el modelo de objetos de documento (DOM), mediante el lenguaje PHP. El mismo cuenta con un conjunto de clases y métodos para el manejo del formato XML.

3.2.4 Modelo de Objetos del Documento (DOM)

DOM define un conjunto estándar de comandos que los parsers devuelven para facilitar el acceso al contenido de los documentos XML desde sus programas. Un analizador de XML compatible con DOM toma los datos de un documento XML y los expone mediante un conjunto de objetos que se pueden programar.

El Modelo de Objetos del Documento (DOM) es una 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. En la especificación del DOM, el término "documento" se utiliza en un sentido amplio. XML se utiliza cada vez más como un medio para representar muchas clases diferentes de información que pueden ser almacenadas en sistemas diversos, sin embargo XML presenta estos datos como documentos, pudiendo así utilizar DOM para manipularlos.

Con el Modelo de Objetos del Documento los programadores pueden construir documentos, navegar por su estructura, y añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier cosa que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando el Modelo de Objetos del Documento, salvo algunas excepciones. En particular, aún no se han especificado las interfaces DOM para los subconjuntos internos y externos de XML.

DOM para XML es un modelo de objetos estándar (propuesto por el W3C) que muestra el contenido de un documento XML. La Especificación del Modelo de Objeto de documento (DOM) del W3C define actualmente lo que debería mostrar un DOM como propiedades, métodos y eventos.

3.3 Diseño del Sistema

3.3.1 Modelo del Dominio

El modelo de dominio es proporcionado por el *proceso unificado* para capturar y expresar el entendimiento ganado del contexto del sistema como paso previo al diseño. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado como una vía para comprender el medio al cual está dirigido.

El modelo de dominio se describe mediante diagramas de UML, en los cuales se muestran las clases del dominio y cómo se relacionan unas con otras mediante asociaciones.

Dentro de los objetivos principales del modelo de dominio se encuentra proporcionar una mejor comprensión de los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

¿Por qué hacer modelo del dominio?

Debido a que no se tienen bien definidos los procesos del negocio se realizará una modelación del dominio y se procederá a explicar cada uno de los conceptos que forman parte del mismo. El modelo del dominio se considera en RUP un subconjunto del llamado *modelo de objetos del negocio*.

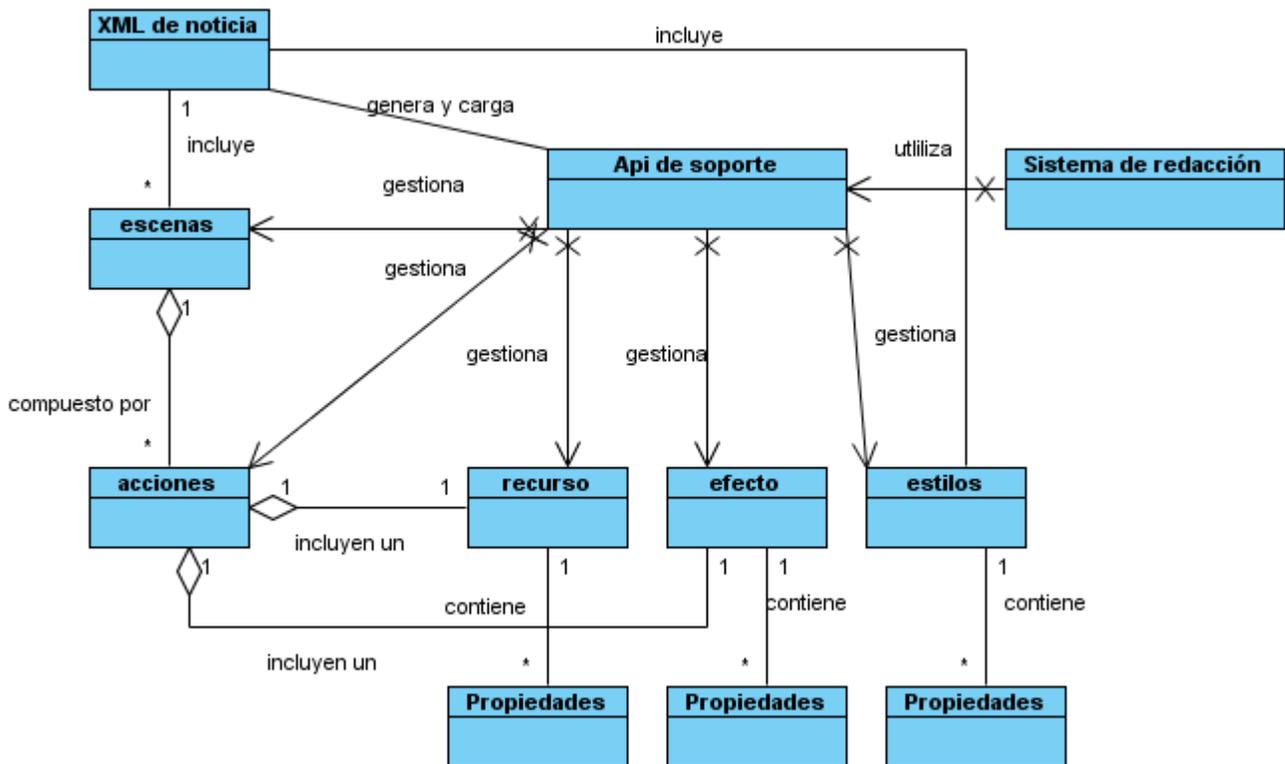


Figura 4: Modelo de dominio del sistema

3.3.2 Captura de los requerimientos

Los requerimientos son de gran importancia para el desarrollo del software ya que determinan lo se quiere construir, representan las necesidades de los usuarios y los objetivos del sistema. Pueden dividirse en requerimientos funcionales y requerimientos no funcionales. Deben ser necesarios, concisos, completos, consistentes, verificables y no pueden ser ambiguos.

Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir (2). A continuación se relacionan los requerimientos funcionales que debe cumplir el sistema:

Requisito funcional 1

Adicionar Escenas: La API debe permitir agregar escenas. El mismo es un elemento encapsulador de acciones, definido por el formato XML de la plataforma PRIMICIA.

Requisito funcional 2

Modificar Escenas: La API debe permitir modificar escenas previamente agregadas a la noticia.

Requisito funcional 3

Eliminar Escenas: La API debe permitir eliminar escenas previamente agregadas a la noticia. Este proceso de eliminación incluye todos los elementos que encapsula dicho elemento.

Requisito funcional 4

Adicionar Acciones: La API debe permitir crear acciones. Dicho elemento es un encapsulador de recursos y efectos, cuya funcionalidad se expresa en la especificación del formato XML de la plataforma PRIMICIA.

Requisito funcional 5

Modificar Acciones: La API debe ser capaz de modificar las propiedades de las acciones.

Requisito funcional 6

Eliminar Acciones: La API debe permitir eliminar acciones tomando en cuenta los procesos realizados a mayor nivel. Este proceso está vinculado al proceso de eliminación del recurso encapsulado por dicha acción.

Requisito funcional 7

Adicionar Recurso: La API debe permitir agregar recursos. Dicho elemento permite anexar información de los distintos tipos de recursos, ya sean multimedia (video, imagen, sonido, texto) ó complejos (infocinta, clima, hora), a la estructura definida por el formato XML de la plataforma PRIMICIA.

Requisito funcional 8

Modificar Recurso: La API debe permitir modificar las propiedades de un recurso ya creado.

Requisito funcional 9

Eliminar Recurso: La API debe ser capaz de eliminar un recurso de ser necesario. Anexo a este proceso se ejecuta la eliminación de la acción que encapsula al mismo.

Requisito funcional 10

Adicionar Efecto: La API debe permitir establecer o agregar un efecto visual a un recurso. Se debe anexar el efecto a la acción que encapsula el recurso al que se aplicará el efecto, tal y como establece la especificación del formato XML de la plataforma PRIMICIA.

Requisito funcional 11

Modificar Efecto: La API debe permitir modificar o agregar nuevas propiedades a un efecto ya creado.

Requisito funcional 12

Eliminar Efecto: La API debe permitir eliminar por completo un efecto si así se precisa.

Requisito funcional 13

Adicionar Estilo La API debe permitir establecer o agregar estilos para la visualización de los textos. El estilo es un elemento declarado en un ámbito global, por motivos de reutilización.

Requisito funcional 14

Modificar Estilo: La API debe permitir modificar o agregar nuevas propiedades a un estilo ya creado.

Requisito funcional 15

Eliminar Estilo: La API debe permitir eliminar por completo un estilo si así se precisa.

Requisito funcional 16

Validar: La API debe validar los datos de todos los elementos que componen el formato XML de la plataforma PRIMICIA. Esta tarea se llevará a cabo aún cuando deben existir otros filtros a mayor nivel que garanticen la corrección de los datos. Dicho requisito llevará a cabo dos funcionalidades:

- ✓ Validación de estructura.
- ✓ Validación de datos.
- ✓ Validación de existencia.

Requisito funcional 17

Búsqueda de Elementos: La API debe ser capaz de localizar cualquier tipo de elemento contenido en la estructura de objetos de la misma. Dicho requisito es fundamental para la ejecución de otros requerimientos explicados en el documento en cuestión.

Requisito funcional 18

Generar XML: El objetivo fundamental de la API es brindar las herramientas para registrar los eventos producidos en el sistema de redacción, que finalizará con la completa generación de la configuración de la noticia mediante el formato XML de la plataforma PRIMICIA. Por tanto es necesario proveer los mecanismos para la generación de toda la configuración.

Requisito funcional 19

Cargar XML: Con el objetivo de proporcionar flexibilidad al manejar las diferentes noticias generadas, la API es debe ser capaz de cargar en su propia estructura de objetos interna un fichero de noticia ya

generado. Dicha funcionalidad elevará la potencia de la misma para brindar soporte a los sistemas que la utilicen.

Requerimientos no funcionales

Los requerimientos no funcionales hacen relación a las características del sistema que se aplican de manera general como un todo, más que a rasgos particulares del mismo. Estos requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema, y corresponden a aspectos tales como la disponibilidad, capacidad de manutención, flexibilidad, seguridad, facilidad de uso, entre otros. De forma generalizada dichos requerimientos representan las características que hacen al producto atractivo, usable, rápido o confiable.

En muchos casos los requerimientos no funcionales son fundamentales para alcanzar un diseño altamente funcional y extensible, todo es cuestión de identificarlos puntualmente desde las primeras fases correspondientes al diseño de un sistema o software. Debido a las características del prototipo en cuestión se definen requerimientos no funcionales orientados mayormente a la escalabilidad y facilidad de uso, desechando aspectos de soporte, seguridad, y requisitos de hardware. A continuación se relacionan los mismos.

Escalabilidad

El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible, para ello deben incorporarse aspectos de reutilización de componentes.

El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

Usabilidad

El prototipo en cuestión deberá simplificar el número de interacciones necesarias para llevar a cabo la configuración de la noticia mediante el formato XML de la plataforma PRIMICIA. El objetivo es lograr altas prestaciones y un incremento de la facilidad de uso en la presente solución.

Portabilidad

La herramienta debe ser capaz de correr en cualquier sistema operativo. Por consiguiente es necesario para la implementación del mismo el uso de lenguaje multiplataforma.

Restricciones en el Diseño y la Implementación

- ✓ Se usará como lenguaje de programación PHP 5 por razones de integración con el sistema de redacción.
- ✓ Como herramienta CASE Visual Paradigm para el modelado de los artefactos que se generen.
- ✓ Se utiliza UML como lenguaje de modelado.

3.3.3 Modelo de Casos de Uso

El modelo de casos de uso permite visualizar de una forma sencilla las funcionalidades que deberá tener el sistema propuesto. En el mismo se definen los actores y casos de uso correspondientes.

Definición de los actores del Sistema

Actores	Justificación
Sistema de redacción	Es el que procede a iniciar todos los procesos de gestión presentes en el modelo de clases de la API.

Tabla 2: Definición de los actores del sistema

Casos de Uso del sistema

CU-1	Gestionar Escena
Actor	Sistema de redacción
Descripción	Se encarga de gestionar la creación, modificación y eliminación de las Escenas dentro del dominio de una noticia.

Referencia	RF-1, RF-2, RF-3
-------------------	------------------

Tabla 3: Especificación del caso de uso Gestionar Escena

CU-2	Gestionar Acción
Actor	Sistema de redacción
Descripción	Se encarga de gestionar la creación, modificación y eliminación de las Acciones encapsuladas en una Escena dentro del dominio de una noticia.
Referencia	RF-4, RF-5, RF-6

Tabla 4: Especificación del caso de uso Gestionar Acción

CU-3	Gestionar Recurso
Actor	Sistema de redacción
Descripción	Se encarga de gestionar la creación, modificación y eliminación de un Recurso encapsulado en una Acción.
Referencia	RF-7, RF-8, RF-9

Tabla 5: Especificación del caso de uso Gestionar Recurso

CU-4	Gestionar Efecto
Actor	Sistema de redacción
Descripción	Se encarga de gestionar la agregación, modificación y eliminación de un Efecto asignado a un Recurso. El Efecto es gestionado dentro del dominio

	que encapsula el Recurso (<i>Acción</i>).
Referencia	RF-10, RF-11, RF-12

Tabla 6: Especificación del caso de uso Gestionar Efecto

CU-5	Gestionar Estilos
Actor	Sistema de redacción
Descripción	Se encarga de gestionar las agregación, modificación y eliminación de Estilos aplicables a elementos de tipo texto. El objetivo de dichos Estilos es maximizar la reutilización y crear estándares tipográficos aplicables a varias noticias.
Referencia	RF-13, RF-14, RF-15

Tabla 7: Especificación del caso de uso Gestionar Estilo

CU-6	Validar Datos
Actor	Sistema de redacción
Descripción	Garantiza que la construcción de la configuración de la noticia se realice de manera explícita según el formato XML de PRIMICIA.
Referencia	RF-16

Tabla 8: Especificación del caso de uso Validar Datos

CU-7	Búsqueda de Elementos
-------------	------------------------------

Actor	Sistema de redacción
Descripción	Se encarga de la búsqueda de los distintos elementos que componen el formato XML de PRIMICIA. El objetivo de esta funcionalidad es dar soporte a las operaciones de gestión y validación.
Referencia	RF-17

Tabla 9: Especificación del caso de uso Búsqueda de Elementos

CU-8	Generar XML
Actor	Sistema de redacción
Descripción	Crea un fichero XML con la configuración de una noticia, según el formato XML de la plataforma PRIMICIA.
Referencia	RF-18

Tabla 10: Especificación del caso de uso Generar XML

CU-9	Cargar XML
Actor	Sistema de redacción
Descripción	Carga un fichero XML con la configuración de una noticia al formato interno de objetos de la API, según el formato XML de la plataforma PRIMICIA.
Referencia	RF-19

Tabla 11: Especificación del caso de uso Cargar XML

3.3.4 Diagramas de Casos de Uso del Sistema

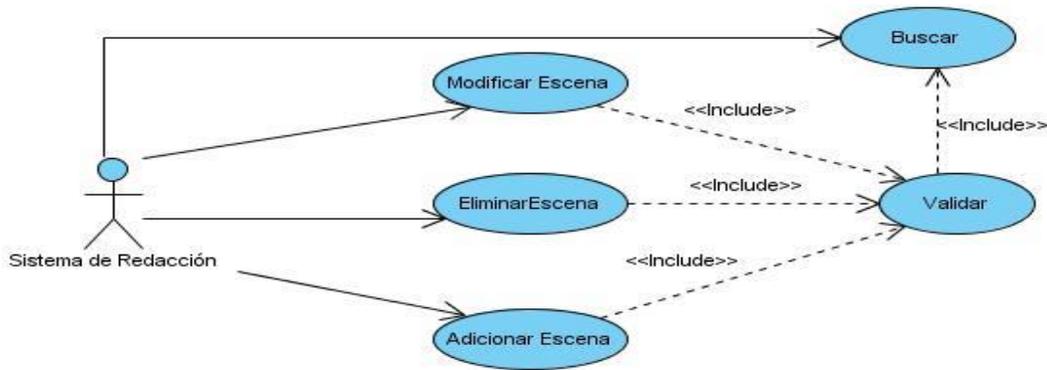


Figura 5: Diagrama del caso de uso Gestionar Escena

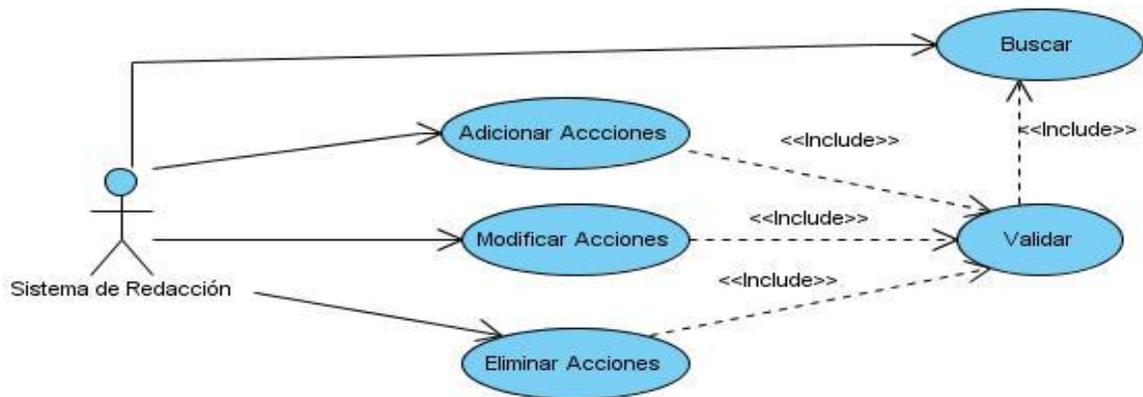


Figura 6: Diagrama del caso de uso Gestionar Acción

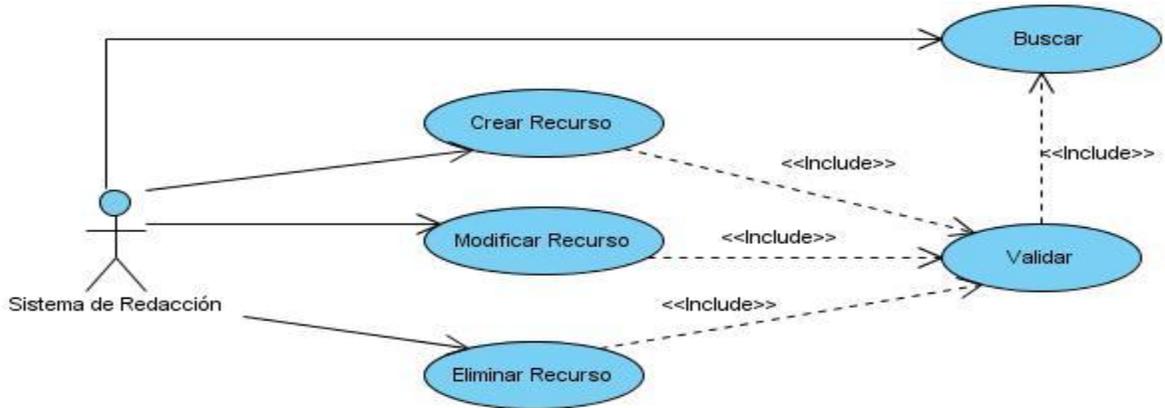


Figura 7: Diagrama del caso de uso Gestionar Recurso

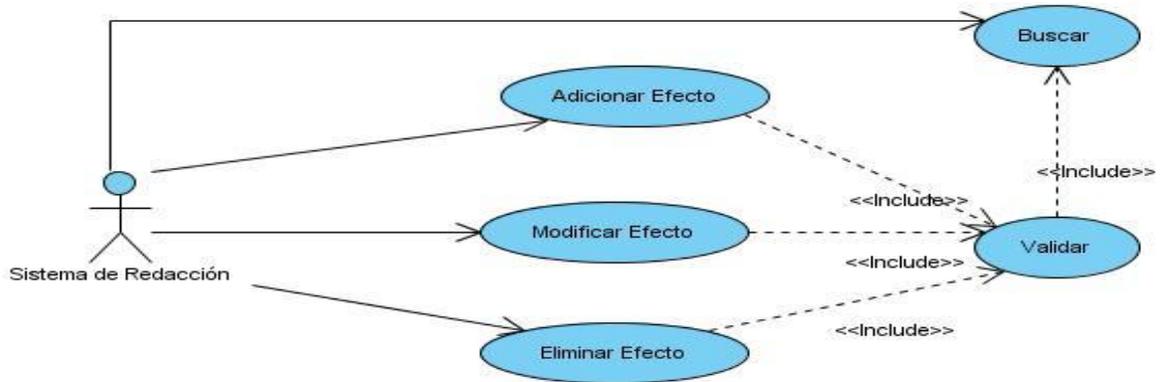


Figura 8: Diagrama del caso de uso Gestionar Efecto

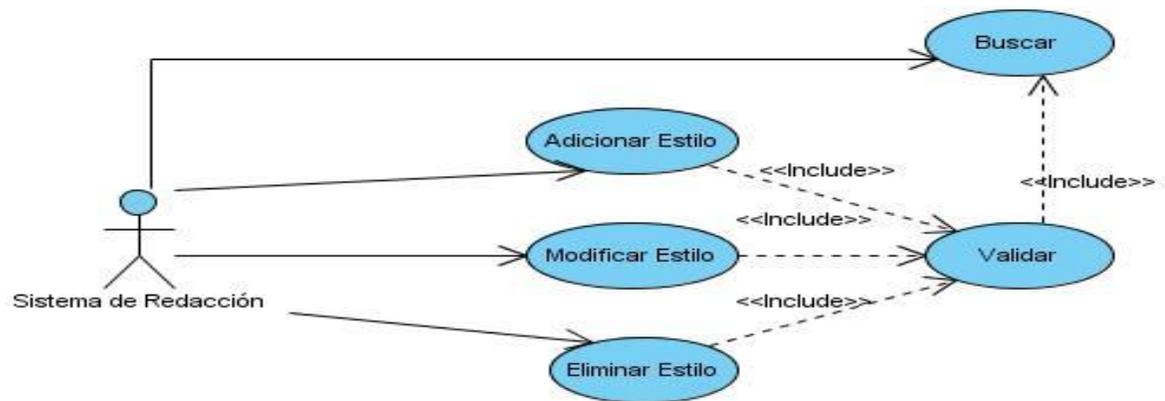


Figura 9: Diagrama del caso de uso Gestionar Estilo



Figura 10: Diagrama del caso de uso Generar XML

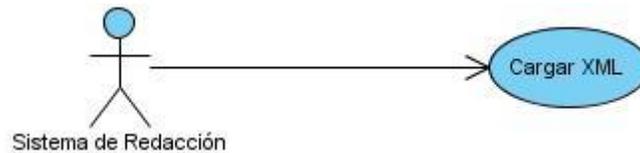


Figura 11: Diagrama del caso de uso Cargar XML

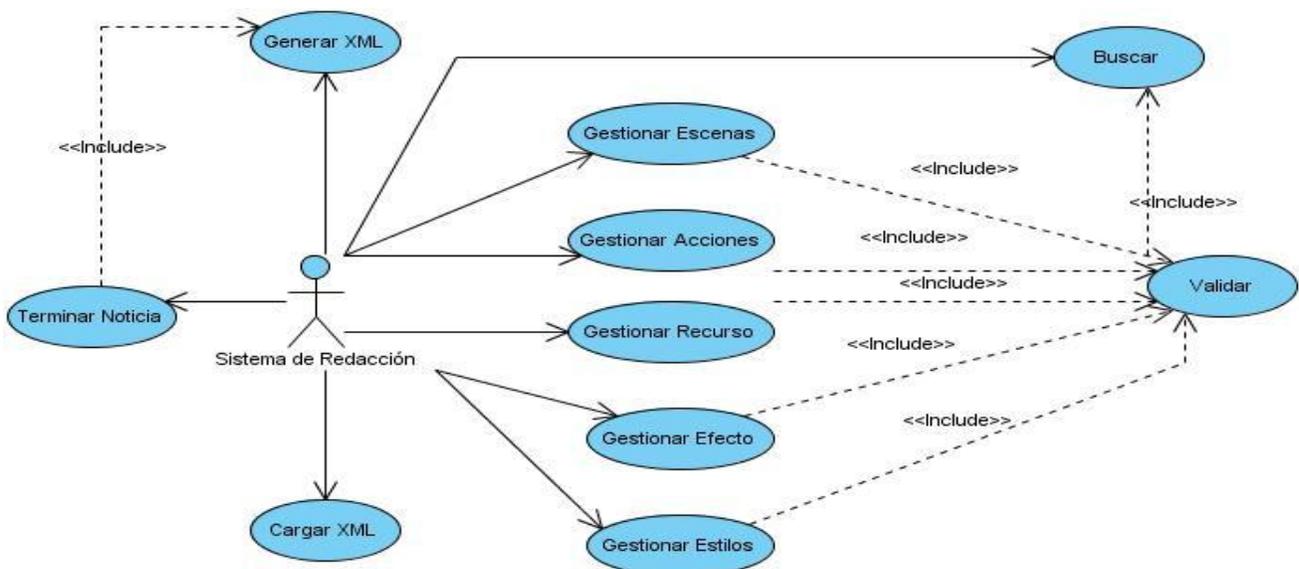


Figura 13: Diagrama general de casos de uso del sistema

3.3.5 Descripción de casos de uso en formato extendido

Para cumplimentar el desarrollo de la presente tarea de manera eficiente se muestran a continuación solo la descripción de las funcionalidades principales de la API.

Caso de Uso:	1- Validar Datos	
Actores:	Sistema de redacción	
Resumen:	El caso de uso se inicia cuando el actor accede a las funcionalidades de gestión de elementos. Dichos elementos pueden ser de varios tipos: <i>Escena, Acción, Recurso, Efecto y Estilo</i> ; cada uno de los mismos posee una especificación que será tomada como base para la validación. A su vez la validación se divide en escenarios: <i>Validación de Estructura, Validación de Propiedades, Validación de Existencia</i> .	
Precondiciones:	Se ha creado el manejador de la noticia.	
Referencias:	RF-16	
Prioridad:	Alta	
CU asociados:	CU-1, CU-2, CU-3, CU-4, CU-5, CU-6,	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. Identifica los escenarios de validación: <ul style="list-style-type: none"> • Validación de Estructura. • Validación de datos. 	

	<ul style="list-style-type: none"> Validación de Existencia.
	2. Ejecuta las operaciones pertinentes y devuelve los resultados.
Pos condiciones	Se logra la validación de todos los elementos que componen el formato XML de la plataforma PRIMICIA.

Tabla 12: Modelo extendido del caso de uso Validar Datos

Caso de Uso:	2- Buscar Elemento	
Actores:	Sistema de redacción	
Resumen:	El caso de uso se inicia cuando el actor solicita las funcionalidades de búsqueda de elementos ya sea para la ejecución procesos de gestión u obtención de información de los mismos. Dentro del caso de uso se realiza la búsqueda de cualquier elemento de alto nivel (refiere elementos encapsuladores de funcionalidades internas, que sirven a su vez de fachada para el trabajo de los parámetros de elementos de nivel inferior) dentro de la noticia definido por el formato XML de la plataforma PRIMICIA.	
Precondiciones:	Se ha creado el manejador de la noticia.	
Referencias:	RF-17	
Prioridad:	Alta	
CU asociados:	CU-1, CU-2, CU-3, CU-4, CU-5, CU-6,	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

1. Solicita la búsqueda y provee los datos necesarios para su ejecución (identificador del elemento).	2. Se ejecuta la búsqueda del elemento según el identificador del mismo.
	3. Si existe el elemento devuelve los resultados de la búsqueda.
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
Acción 1. 1. Ejecuta la funcionalidad de búsqueda y provee los datos incorrectos o inexistentes.	2. Se ejecuta la búsqueda del elemento según el identificador del mismo.
	3. Devuelve mensaje de error.
Pos condiciones	Sirve de base para la validación y brinda soporte casos de uso dependientes.

Tabla 13: Modelo extendido del caso de uso Buscar Elemento

Caso de Uso:	3- Generar XML
Actores:	Sistema de redacción
Resumen:	El caso de uso se inicia cuando el actor acciona la funcionalidad del método generar XML de noticia. Seguidamente se ejecutan las operaciones necesarias para generar el fichero XML con la configuración de la noticia según el formato XML de la plataforma y termina el caso de uso.
Precondiciones:	Existe un flujo de noticia compuesto por Escenas, Acciones y Estilos.
Referencias:	RF-18

Prioridad:	Muy Alta
CU asociados:	-
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Acciona la funcionalidad encapsulada por él método generar XML.	2. Inicia el proceso de conversión de la noticia al formato XML de la plataforma PRIMICIA.
	3. Salva el fichero resultante en la localidad definida en la distribución de PRIMICIA.
Pos condiciones	Se obtiene una noticia configurada en el formato XML de PRIMICIA para su posterior utilización en el subsistema de transmisión.

Tabla 14: Modelo extendido del caso de uso Generar XML

Caso de Uso:	4- Cargar XML
Actores:	Sistema de redacción
Resumen:	El caso de uso se inicia cuando el actor acciona la funcionalidad Cargar XML de noticia. Seguidamente se ejecutan las operaciones necesarias para cargar la noticia en la estructura interna de objetos de la API; y validar que el XML en cuestión constituye un formato válido de noticia según el formato XML de la plataforma.
Precondiciones:	-
Referencias:	RF-19

Prioridad:	Muy Alta
CU asociados:	-
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Acciona la funcionalidad Cargar XML y proporciona la ubicación de una noticia según el formato XML de PRIMICIA.	2. Verifica si existe el fichero en la ubicación proporcionada.
	3. Inicia el proceso de comprobación de la noticia según el esquema del formato XML de la plataforma PRIMICIA.
	4. Si es un formato de noticia correcto, se ejecuta el proceso de carga de las configuraciones a los modelos de objetos definidos por la API.
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
Acción 1. 1. Ejecuta la funcionalidad Cargar XML provee los datos incorrectos o inexistentes.	2. Verifica si existe el fichero en la ubicación proporcionada.
	3. Si no se encuentra fichero, devuelve mensaje indicando error.
Acción 1. 1. Ejecuta la funcionalidad Cargar XML provee los datos incorrectos o inexistentes.	2. Verifica si existe el fichero en la ubicación proporcionada.

	3. Inicia el proceso de comprobación de la noticia según el esquema del formato XML de la plataforma PRIMICIA.
	4. Se detectó error en el formato XML proveído. Devuelve mensaje indicando error.
Pos condiciones	Se obtiene un flujo de noticia listo para su posterior modificación o extensión.

Tabla 15: Modelo extendido del caso de uso Cargar XML.

3.4 Análisis

El propósito del análisis es lograr un refinamiento y estructuración de los requisitos capturados en fases anteriores para lograr una mayor comprensión, reparación y modificación de los mismos. Por su parte el modelo de análisis especifica el comportamiento funcional del sistema, independientemente de los aspectos relativos al ambiente en el que va a ser finalmente implementado. Este modelo captura completamente y con exactitud los requerimientos. Por tanto es considerado una entrada primordial al modelo del diseño.

¿Por qué no realizar el modelado del análisis?

Aunque el análisis forma parte de uno de los flujos de trabajo más importantes de RUP, su empleo no es conveniente en todos los casos. Cuando los requisitos del software permanecen simples, no existen grandes riesgos y el proyecto no constituye una innovación, puede obviarse el modelado del análisis, para de tal forma, reducir el costo de tiempo. En el caso particular de la investigación en cuestión, se tomó la decisión de no realizar el análisis por las razones antes expuestas y debido a la selección de la metodología AUP. Dicha metodología constituye una versión ágil de RUP y como tal prioriza el ahorro de tiempo en actividades de diseño para incrementar el tiempo de implementación e integración.

3.5 Conclusiones

En el presente capítulo se profundizaron aspectos técnicos del diseño así como particularidades de la propuesta en cuestión. A partir de este punto se comenzaron a refinar e identificar funcionalidades

técnicas acerca de las herramientas que acompañan el proceso de implementación. Por tanto, a modo de conclusiones se puede decir que la solución propuesta se caracteriza por tener un diseño genérico donde los aspectos que comprenden la validación se apoyarán en herramientas externas como los XML esquemas.

Capítulo 4: Construcción de la solución propuesta

4.1 Introducción

En el presente capítulo se abordan los aspectos fundamentales asociados a la construcción de la solución en cuestión. Se abordan los modelos del análisis e implementación, las descripciones de los patrones de diseño identificados, así como una fundamentación teórica de cada modelo presentado.

4.2 Diseño

Durante la actividad de diseño se modela el sistema y define su forma, incluida su arquitectura para que soporte todos los requisitos. El diseño permite adquirir una comprensión profunda de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de interfaz de usuario, entre otras. Además crea una entrada apropiada para las actividades de implementación, y la descompone en partes más manejables separadas por interfaces definidas (paquetes, subsistemas de diseño) (22).

4.2.1 Modelo de Diseño

El Modelo de Diseño constituye la realización física de los casos de uso, teniendo en cuenta restricciones de implementación como el lenguaje de programación a utilizar así como componentes, sistemas operativos y tecnologías usadas en la realización del sistema, además de basarse en la comprensión de los aspectos relacionados con los requisitos no funcionales.

4.2.2 Patrones de Diseño

Los patrones de diseño, constituyen soluciones estándar que brindan respuesta a un problema común en materia de diseño de sistemas e implementación. Estos permiten establecer una gran comunicación entre los diseñadores para facilitar el aprendizaje del programador inexperto. Cada patrón sugiere además numerosas ventajas, dentro de ellas, la reutilización de código y la realización de un diseño extensible.

A cada patrón se le asigna un nombre que lo identifica, y a su vez son clasificados según su propósito como: creacionales, realizan la inicialización y configuración de objetos; estructurales, se encargan de las

relaciones entre clases, combinación y formación de estructuras mayores; y de comportamiento, este plantea la interrelación de comunicación entre los componentes (21).

A continuación se presentan los diferentes patrones identificados en el proceso de modelado de las clases del diseño. Así como también se precisan las filosofías de diseño de sistemas o patrones GRASP que se tuvieron en cuenta para aplicar cada patrón seleccionado.

Patrones Creacionales

Singleton: Plantea la existencia de una única instancia de una clase y un único punto de acceso global a esta. Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton, incluso en tiempo de ejecución (23). Las ventajas de este patrón se ven muy relacionadas al encapsulamiento de los objetos que este posee. El acceso controlado a la única instancia permite controlar de forma estricta cómo y cuándo se accede a ella.

Método de fábrica: Este patrón de diseño se encarga de definir una interfaz para crear un objeto de tipo genérico, pero a su vez permite que las subclases decidan qué clase instanciar. Es utilizado cuando una clase no puede anticipar qué objetos va a crear y además se usa cuando una clase quiere que sus subclases indiquen qué objetos crea. La utilización de este patrón garantiza el incremento de la flexibilidad del diseño.

Patrones Estructurales

Fachada: Este es un patrón clasificado como estructural, que implementa métodos convenientes para tareas comunes. Además provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces, permite ocultar toda la complejidad del sistema mostrando solamente al usuario puntos de entrada y el acceso a introducir valores estando siempre ajeno al funcionamiento interno del sistema.

Composite: Tiene como propósito estructurar subclases para que se puedan tratar de manera uniforme muchos objetos relacionados y que múltiples objetos puedan ser tratados como uno solo, es decir, no se centra en la decoración sino en la representación. No obstante su implementación puede devenir en un diseño demasiado genérico donde se dificulte la restricción de ciertos objetos. Por tal motivo en el caso presente se realizaron modificaciones en el diseño que suplieran estas deficiencias.

Patrones de Comportamiento

Cadena de Responsabilidad: La idea de este patrón es desacoplar a los emisores y a los receptores dándole a varios objetos la posibilidad de tratar una petición, que se pasa a través de una cadena de objetos hasta que es procesada por alguno de ellos. Para reenviar la petición a lo largo de la cadena, garantizando que los receptores permanezcan implícitos, cada objeto de la cadena comparte una interfaz común para procesar peticiones y para acceder a su sucesor en la cadena (13). Este patrón constituye la base de la generación y carga a XML con que cuenta la presente solución.

Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Estos patrones constituyen el fundamento de cómo se diseñará el sistema. (24)

Como tal todo sistema necesita de filosofías fundamentales que controlen ó condicionen la correcta ejecución del modelo de diseño de un sistema software. Estos patrones se centran en los principios de creación de objetos, el acoplamiento entre los mismos y la necesidad de clases que deleguen responsabilidades de tal forma que se logre un diseño limpio y flexible. A continuación explicarán dichos patrones:

Experto: Es un principio básico que suele utilizarse en el diseño orientado a objetos. Posee gran uso para asignar una responsabilidad al experto en información (la clase que cuenta con la información necesaria para cumplir la responsabilidad).

Alta Cohesión: Posee un alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. Este patrón soporta un aumento de la capacidad de reutilización. Además brinda como solución, el diseño de una clase con responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.

Bajo Acoplamiento: Este patrón estimula asignar una responsabilidad tratando de mantener el bajo acoplamiento. Ofrece como solución el diseño de clases más independientes, para lograr que no exista

una fuerte dependencia con el resto de las clases. Todo lo anterior posibilita reducir el impacto de los cambios y una mayor reusabilidad.

Creador: Ofrece como solución asignarle a una clase B la responsabilidad de crear una instancia de clase A. Guía la asignación de responsabilidades relacionadas a la creación de objetos y su intento básico es encontrar un creador que necesite estar conectado al objeto creado en un evento particular. Este patrón también beneficia el bajo acoplamiento, el mantenimiento y la reusabilidad del código.

Controlador: Es capaz de asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, entre otras). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

4.2.3 Diagramas de clases del diseño

Los diagramas de clases del diseño representan la estructura y organización de las clases que intervienen en el caso de uso desde un punto de vista orientado a la implementación y más específico donde están presentes los métodos a utilizar en cada clase y la forma de comunicación de las mismas. A continuación se presentará el diagrama de clases del diseño dividido por paquetes del diseño para facilitar el entendimiento del mismo.

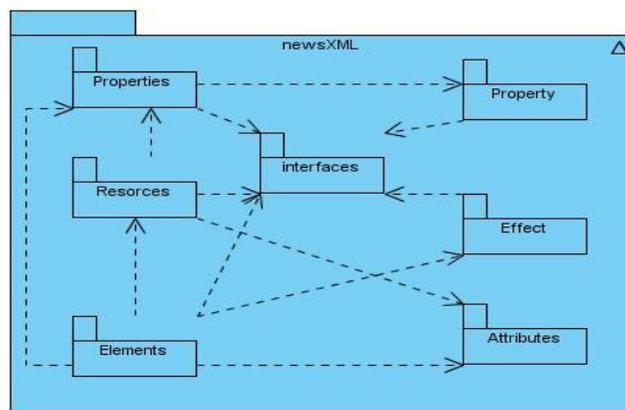


Figura 12: Diagrama de paquetes del diseño

Seguidamente se mostrará la distribución de clases contenida en los paquetes de diseño: Property, Properties, Interfaces, Attributes, Resources, Effect y Elements.

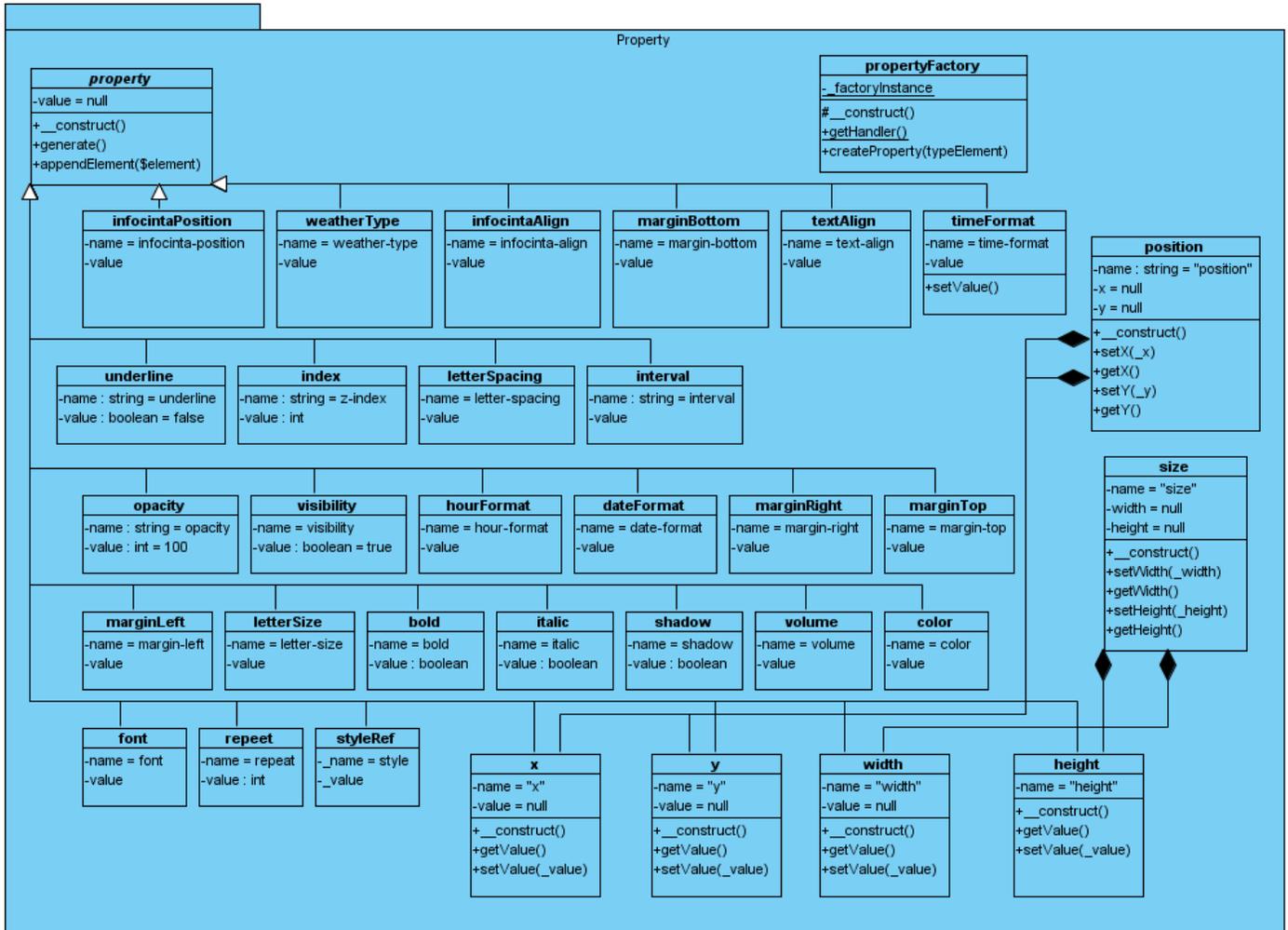


Figura 13: Distribución de clases del paquete Property.

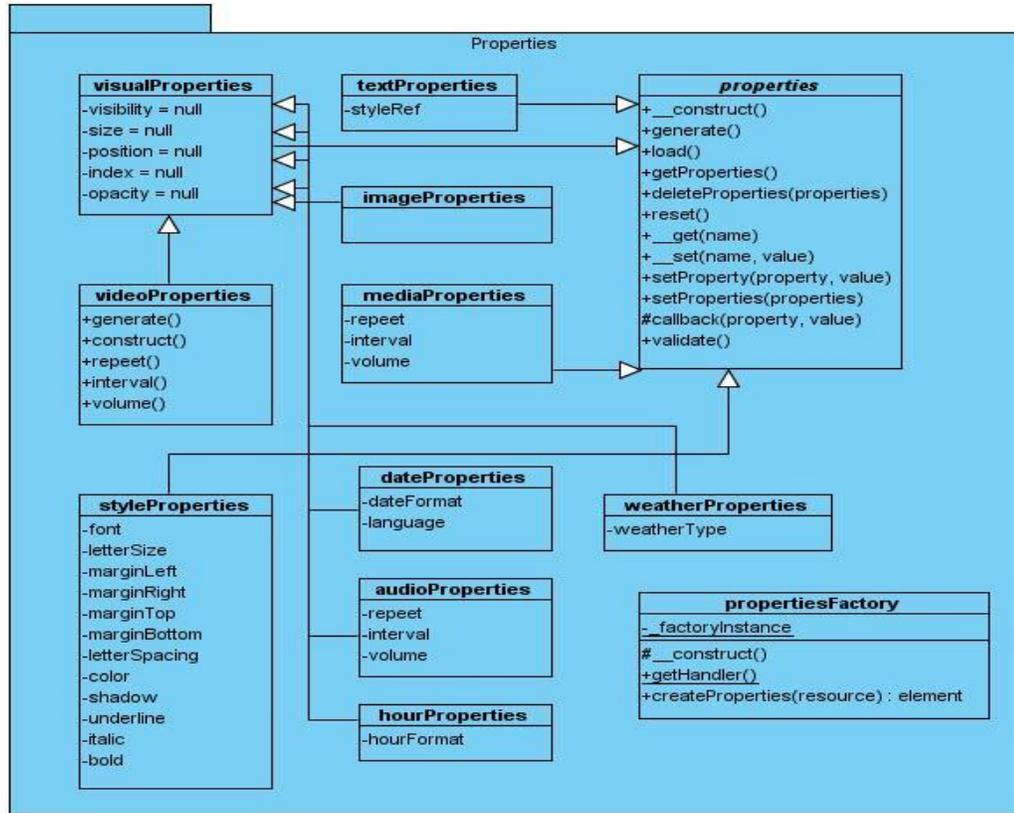


Figura 14: Distribución de clases del paquete Properties.

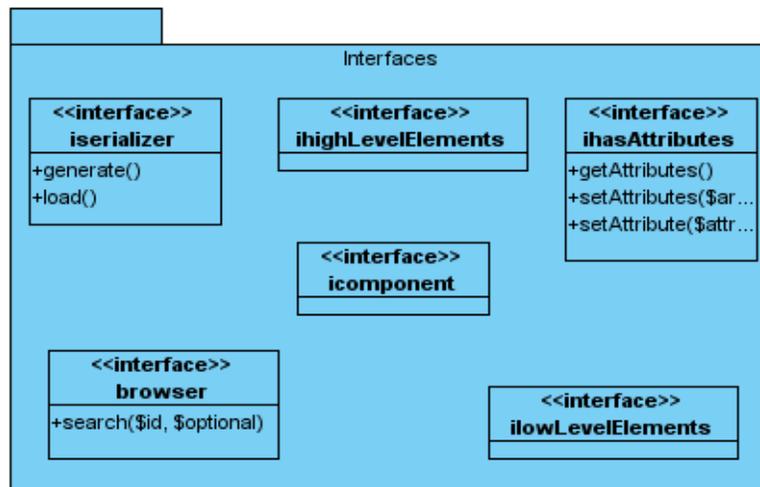


Figura 15: Distribución de clases del paquete Interfaces.

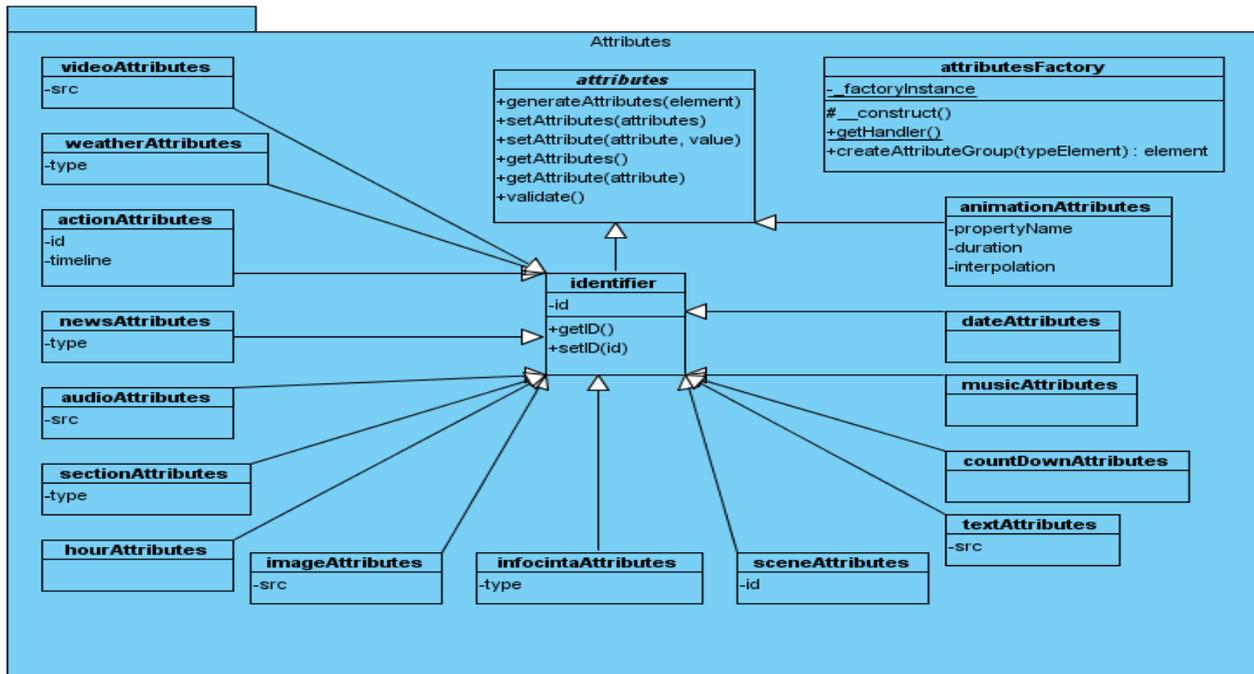


Figura 16: Distribución de clases del paquete Atributes

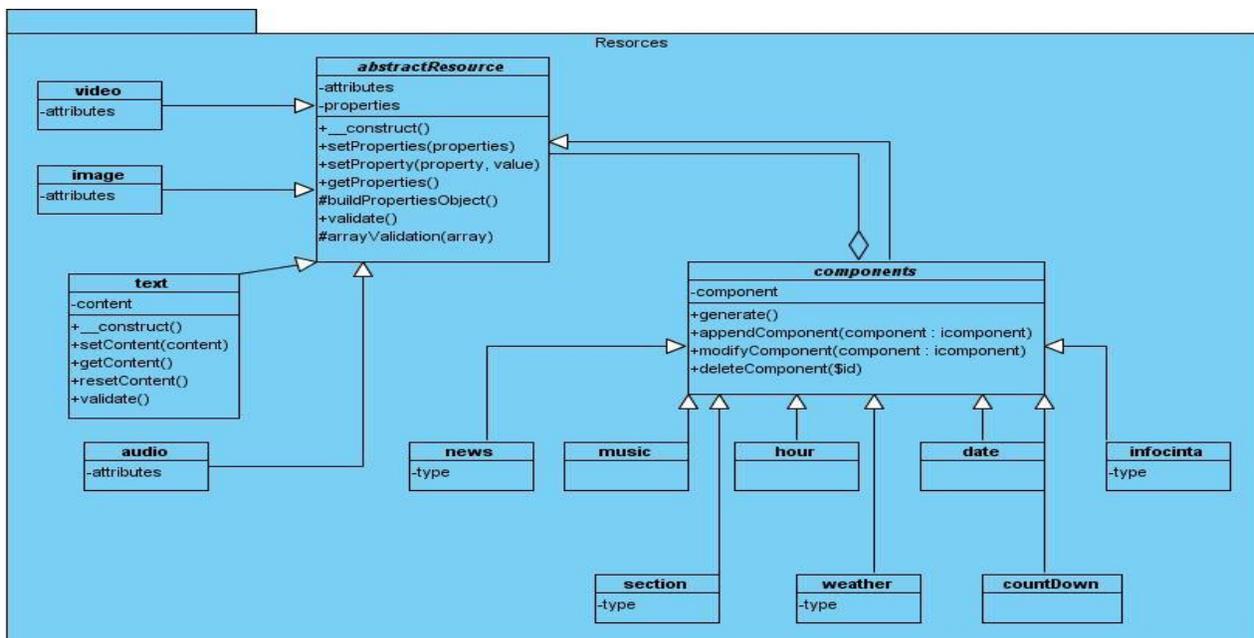


Figura 17: Distribución de clases del paquete Resources.

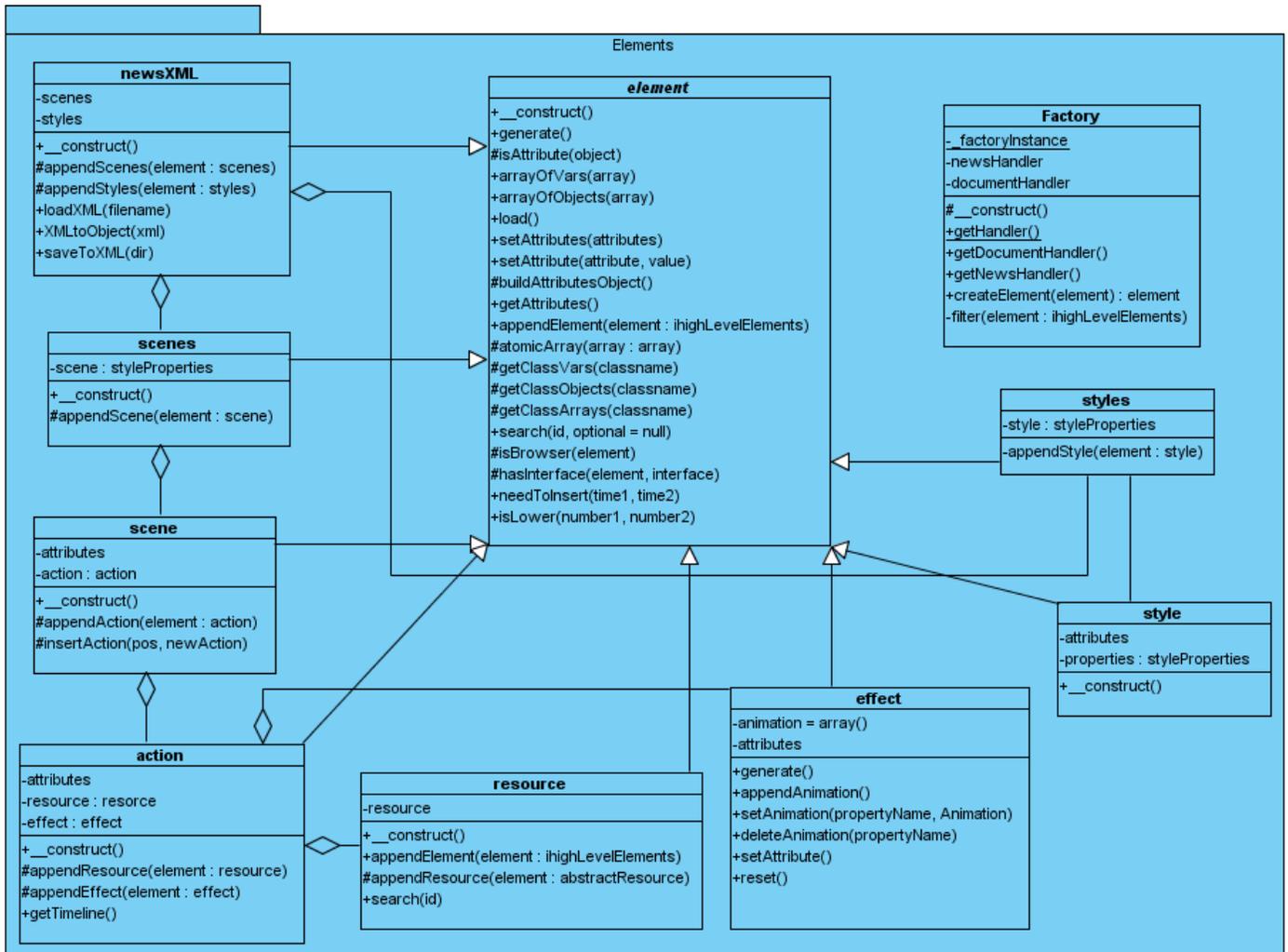


Figura 18: Distribución de clases del paquete Elements.

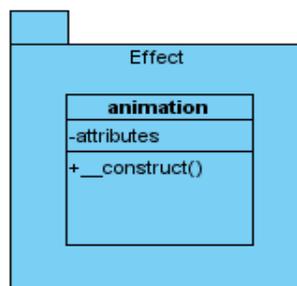


Figura 19: Distribución de clases del paquete Effect.

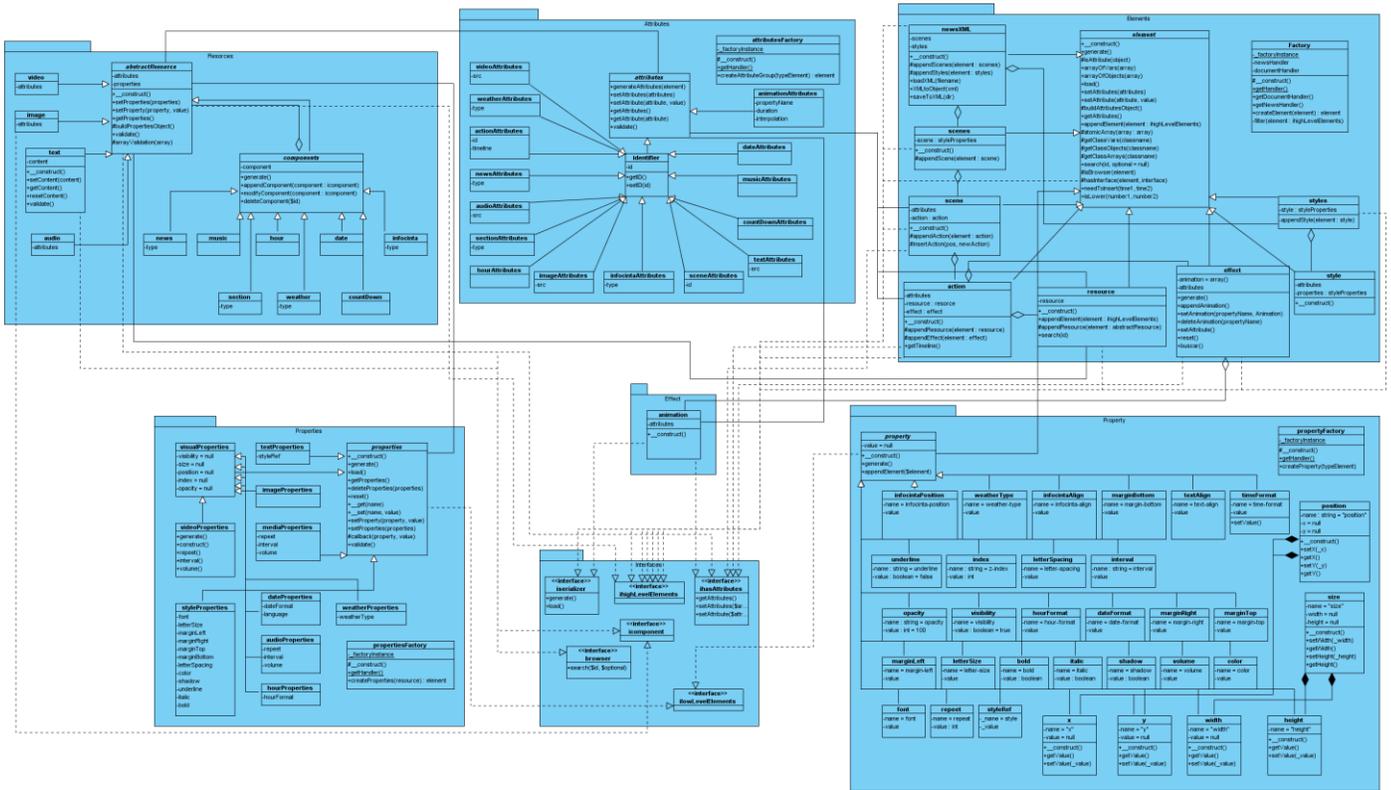


Figura 20: Diagrama de clases del diseño (Distribución global).

4.3 Implementación

La actividad de la implementación parte de los resultados del diseño para implementar el sistema en términos de componentes, entendiendo como tales a los componentes. La actividad fundamental durante la implementación es lograr la distribución del sistema entre los distintos nodos y la implementación de las clases del diseño.

4.3.1 Diagrama de despliegue

Dado que el sistema propuesto constituye un software de arquitectura monolítica se concluyó no realizar una representación del mismo pues la información que dicho diagrama provee es irrelevante para el diseño de la presente solución.

4.3.2 Modelo de implementación

El Modelo de implementación se constituye de una colección de componentes, y subsistemas de implementación que los contienen. Describe tanto los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Estos componentes incluyen: ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y el despliegue del sistema (25).

A continuación se presentará la distribución física global en términos de componentes, y los nodos que los contienen, y posteriormente la estructura interna de los nodos físicos más importantes.

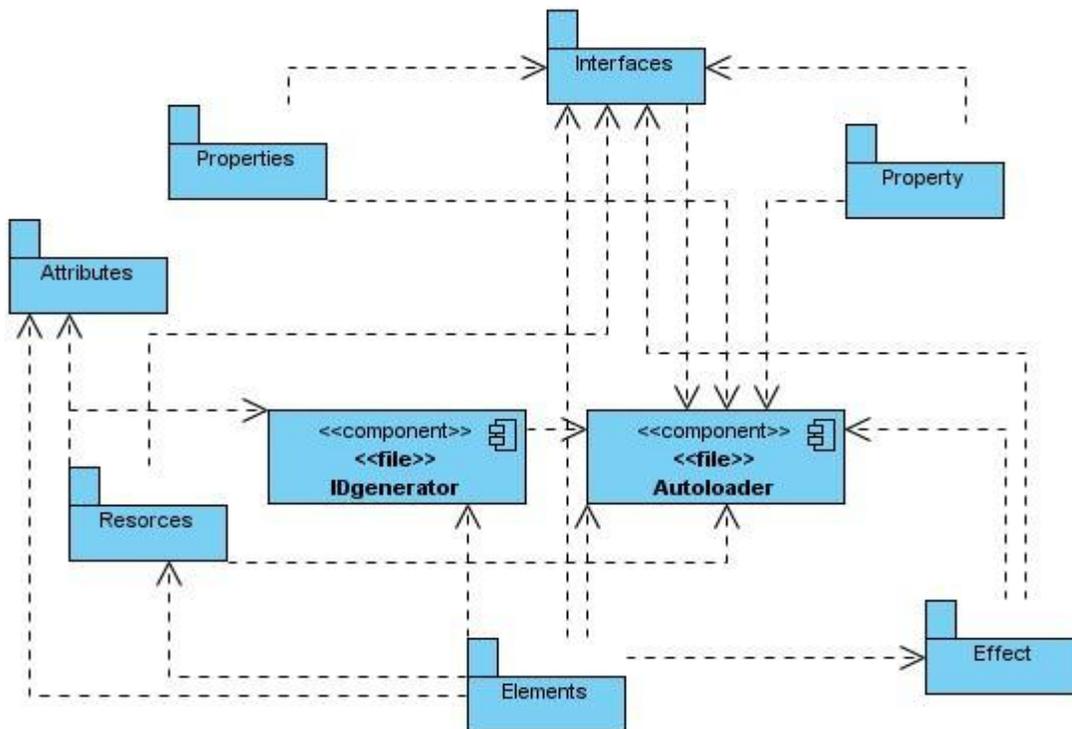


Figura 21: Distribución física global de la API.

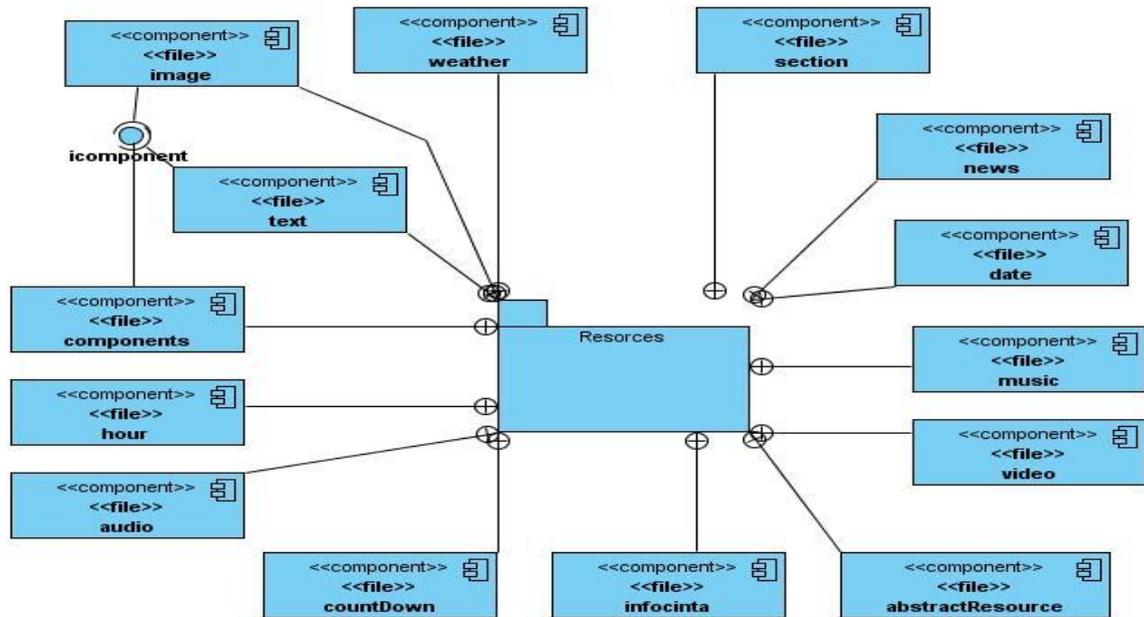


Figura 22: Distribución interna del nodo Resorces.

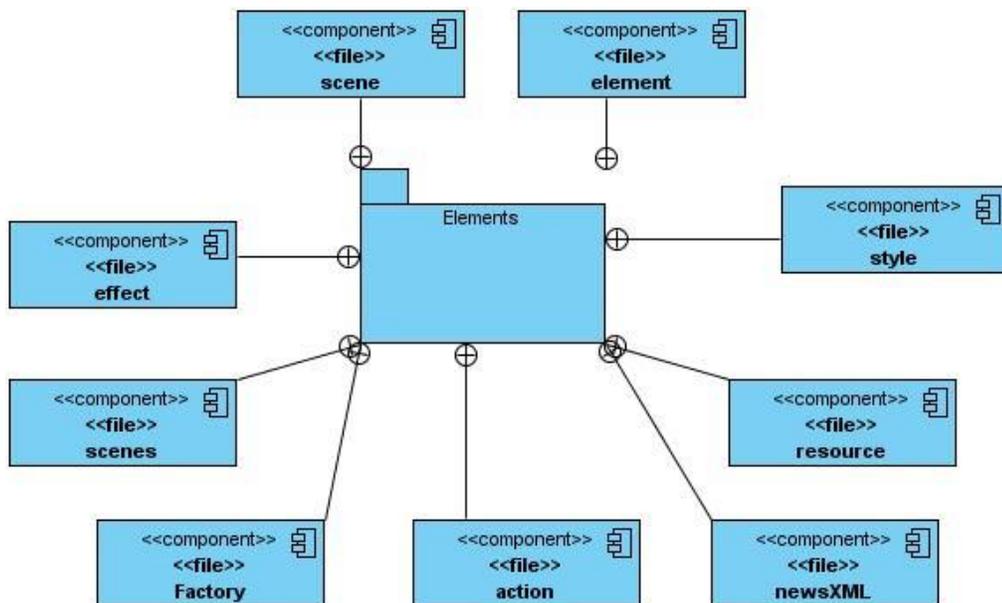


Figura 23: Distribución del nodo físico Elements

4.4 Conclusiones

Para cumplir los objetivos trazados para el presente capítulo, se realizó un estudio profundo de los aspectos fundamentales del diseño e implementación. Dicho estudio concluyó con la identificación de patrones de diseño aplicables al modelo propuesto, y la posible distribución física del sistema en cuestión. Para culminar se realizaron los modelos correspondientes a cada aspecto de diseño identificado y los modelos de distribución.

Conclusiones

Una vez cumplidos los objetivos trazados para la presente investigación, se obtuvo como resultado el prototipo funcional encargado de gestionar los ficheros de configuración de noticias para la plataforma PRIMICIA. Dicho prototipo constituye una API para la gestión, generación y carga de los elementos configurables que conforman el estándar de noticias de la plataforma.

En el desarrollo de la presente investigación se realizaron una serie de estudios enmarcados por el ámbito de la misma. A continuación se presentan algunos de los resultados y conclusiones parciales adquiridas en el proceso.

- ✓ El análisis de las herramientas y tecnologías asociadas permitió detectar las vías necesarias para dar vida a la solución. Se identificaron ventajas y desventajas, y se tuvo en cuenta el origen libre y multiplataforma de las mismas. De tal forma se hizo posible seguir la política del país de desarrollar sistemas informáticos con soberanía tecnológica.
- ✓ Durante el proceso de construcción de la solución fueron generados los artefactos relacionados al diseño. Para ello se tomó como base lo establecido por la metodología de desarrollo AUP (variante ágil de RUP), lo cual constituyó la base para la implementación del sistema.
- ✓ La herramienta se desarrolló de manera general utilizando el paradigma de la programación orientada a objetos. La elección estuvo sujeta a las características y filosofías de desarrollo de sistemas que soportan dicho paradigma, lo cual permitió la realización de un sistema genérico, extensible y flexible a cambios futuros en el estándar de noticias de la plataforma PRIMICIA.

Así pues, se concibe que tras la obtención de dicho prototipo sea posible la integración entre los módulos y los subsistemas que estructuran la plataforma.

Recomendaciones

Durante el desarrollo del presente trabajo se realizaron diversas investigaciones asociadas al cumplimiento de los objetivos planteados en la misma. Dichas investigaciones aportaron los elementos necesarios para la concepción de la presente y posteriores soluciones. Así pues, se recomienda:

- ✓ Dar continuidad al desarrollo y soporte del prototipo funcional resultante de la presente investigación. Se deben incorporar nuevas funcionalidades hasta obtener una solución estable integrada completamente a la plataforma PRIMICIA.
- ✓ Realizar un estudio profundo para identificar nuevos elementos configurables y la estructura serializable¹⁸ de los mismos.
- ✓ Por último, teniendo en cuenta los actuales usos de los estándares de noticias basados en XML, el avance incontenible de las redes y la tecnología digital, se propone el estudio y desarrollo de una nueva funcionalidad basada en un motor XML/XSL. Dicho motor permitiría a PRIMICIA extenderse al nuevo medio digital que constituye internet con una solución totalmente integral y extensible a diversos formatos de presentación.

¹⁸ **Serializable** es un término utilizado por algunos lenguajes de programación para expresar la capacidad que posee un objeto de ser archivado en formatos configurados por el programador. Usualmente se utilizan interfaces con nombres análogos que permitan identificar que el objeto posee dicha capacidad.

Trabajos citados

1. **Avilés, José Alberto García.** *Influencia de la Digitalización en los canales "Todo Noticias" de las televisiones públicas.* Universidad Miguel Hernández : s.n.
2. **BIGGS, WES.** "Web publishing with XML. Part I: Defining the definitions". "Online Journalism Review". . 2001.
3. **Lilian Teresa Castro Mecías, Rolando Bermúdez Peña.** *Desarrollo de una Librería de Componentes JavaScript basado en Dojo Toolki.* 2008.
4. **Dykes, Lucinda.** *XML for Dummies.*
5. **Ligth, Richard.** *Presenting XML.*
6. **Collado, Manuel.** *Introducción a las tecnologías y estándares XML.* 2006.
7. **Sturm, jack.** *Desarrollo de soluciones XML.*
8. **Lorente Miranda, Rafael.** *Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA.* Habana : s.n., 2008.
9. **García, Ruber Hernández.** *Sistema Automatizado de teletexto para la Plataforma de Televisión Satelital Cubana.* Habana : s.n.
10. **Atos Origen.** *Digital & Media Press. Descripción del producto.* 2009.
11. **Rodríguez, Maray Montero.** "Análisis y Diseño de un Sistema de Gestión y Control de Trámites para el Ministerio del Turismo (MINTUR)". Ciudad de la Habana : s.n., 2009.
12. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* s.l. : Prentice Hall.
13. **Larman, Craig.** *UML y Patrones.* 1999.

14. **Chacón, Julio César Rueda.** *Aplicación de la Metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE.* Guatemala : s.n., 2006.
15. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley.
16. **José H. Canós, Patricio Letelier, M Carmen Penadés.** *Metodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.
17. **Amaro Calderón, Sarah Dámaris, Valverde Rebaza. Jorge Carlos.** *Metodologías Ágiles.* Trujillo-Perú : s.n., 2007.
18. **Padrón, Jasmín Fuentes.** *Propuesta de marco de Fábricas de Software basado en las buenas prácticas de Métodos Ágiles.* 2008.
19. **Kendall, Kenneth E.** *Análisis Y Diseño De Sistemas.* 2010 : Prentice Hall.
20. Visual-paradigm.com. *Visual-paradigm.com.* [En línea] 2010. <http://www.visual-paradigm.com/product/vpuml/editions/standard.jsp>.
21. **Martínez, Jairo Chapela.** *Introducción al entorno de desarrollo Eclipse.* 2007.
22. **TORRES, M. L.** *Introducción al diseño con patrones.* 2009.
23. **Welicki, León.** *MSDN. El Patrón Singleton.* 2009.
24. **Gerónimo, Luis Fernando Alonso.** *Aplicación de patrones de diseño de software en el desarrollo de una aplicación de base de datos orientada a objetos.* guatemala : s.n., 2005.
25. **Lanvin, Daniel Fernández.** *Desarrollo de una metodología para un nuevo paradigma de desarrollo de software.* 2004.
26. **Gedda, R.** *Linux breaks desktop barrier in 2004.* 2004.

27. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* 2001.
28. Monografias.com. *Monografias.com.* [En línea] <http://www.monografias.com/trabajos14/internet-hist/internet-hist.shtml>.
29. **Aroche, Stephanie Falla.** *maestros del web. maestros del web.* [En línea] <http://www.maestrosdelweb.com/editorial/internetthis/>.
30. **Sterling, Bruce.** [En línea] 1992. http://biblioweb.sindominio.net/telematica/hist_internet.html.
31. **Sanchez, María A. Mendoza.** *Metodologías De Desarrollo De Software.* Perú : s.n., 2004.
32. **Pérez, Javier Eguíluz.** *Introducción a Ajax.*
33. *tejedoresdelweb.com.* [En línea] http://www.tejedoresdelweb.com/w/Parsing_XML.
34. **Fidalgo, Tomás Vilariño.** *Ajax.* 2007.
35. **Larman, Craig.** *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos.*

Bibliografía

1. **Gayo, Jorge Emilio Labra.** *Diseño de sitios Web mediante estándares.*
2. **W3C.** W3C. [En línea] <http://www.w3.org/TR/DOM-Level-2-Core/>.
3. **Rincón, Patricia Cruz.** *Animación para web con SMIL.*
4. **W3C.** W3C. [En línea] 2008. <http://www.w3.org/TR/2008/WD-timesheets-20080110/>.
5. —. W3C. [En línea] [Citado el: 20 de 11 de 2009.] <http://www.w3.org/XML>.
6. **P, Castell.** *La Web semántica.* Madrid : s.n., 2004.
7. **Kay, Michael.** *XSLT Programmer's Reference, 2nd edition.*
8. **E, Ander Egg.** *Técnicas de investigación social.* 1995.
9. **M, Bunge.** *La investigación científica.* 1969.
10. **M, González.** *Metodología de la investigación social.* 1997.
11. **Hernández, Fernandez y Batista.** *Metodología de la investigación.* 1991.
12. **Sierra, Alejandro Aguilar.** *Programación Extrema y Software Libre.* 2002.
13. **González, Victor.** *Introduction to the Microsoft Solutions Framework.*

Glosario de términos

UML: (Lenguaje Unificado de Modelado): Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

RUP: (Proceso de Desarrollo Unificado): Proceso de desarrollo de software, que constituye la metodología más usada para el análisis, implementación y documentación de sistemas con tecnología orientada a objetos.

XML: Por sus siglas en inglés de Extensible Markup Language («lenguaje de marcas extensible»). Propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, etc.

Base de datos: Conjunto de información relacionada que se encuentra agrupada ó estructurada.

CU *Caso de Uso*. Es una secuencia de pasos a seguir para la realización de un fin o propósito.

DHTML: HTML Dinámico que es una mejora de *Microsoft* de la versión 4.0 de HTML que le permite crear efectos especiales.

E-BUSINESS: Término utilizado para nombrar a los Negocios mediante la *Web*.

FRAMEWORKS: Plantillas predefinidas, que facilitan la programación.

RAD: *Desarrollo Rápido de Aplicaciones*. Metodología que permite a las organizaciones desarrollar sistemas estratégicamente importantes, de manera más rápida, reduciendo a la vez los costos de desarrollo y manteniendo la calidad. Existen varias tecnologías que utilizan esta metodología y que intentan reducir el tiempo de desarrollo.

STAKEHOLDER: Personas u organizaciones que están directamente envueltas en la elaboración o tomas de decisiones claves acerca de la funcionalidad y propiedades del Sistema.

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

PANTALLAS: Estructura que define un segmento de noticia, teniendo en cuenta el espacio disponible en la pantalla para mostrar los contenidos, dígame textos, imágenes o videos. Sección Temática: Idea general que es tratada indistintamente por las noticias, dándonos la posibilidad de agruparlas según su contenido.

CINTILLOS: Recurso visual utilizado para mostrar información acerca de la fotografía mostrada en las pantallas de tipo imagen.

API: Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

GUI: La interfaz gráfica de usuario, conocida también como GUI (del inglés graphical user interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

PARSEADO: Es el análisis de una oración o expresión en un lenguaje, en términos de sus constituyentes gramaticales, identificando las partes y sus relaciones sintácticas. En general el parseo significa realizar un examen detallado de un todo descomponiéndolo en sus componentes.