La Güeb de Joaquín
Mi sitio de Internet

Apuntes Tácticos

Sindicación RSS

Información del documento

	La Güeb de Joaquin (http://joaquin.medina.name)
Categoría	Programación
Tema.:	Sindicación RSS
Titulo	Sindicación RSS
Autor.:	Joaquin Medina Serrano (mailto:joaquin@medina.name)
Archivo	Informática/Documentación/Lógica/Normas Y estándares/sindicación/RSS
Sumario	La Sindicación de Contenidos por medio de canales RSS es un medio para distribuir contenidos en la Web que brinda al usuario la posibilidad de personalizar los contenidos que desea acceder. RSS es la abreviatura de RDF Site Summary, Rich Site Summary, Really Simple Sindication pero también se lo utiliza para nombrar el formato Atom. Este artículo aborda una serie de aspectos acerca de la Sindicación de Contenidos y los canales RSS. Primero se realiza una introducción como crear manualmente documentos RSS Bien formados, a continuación se describen una a una las etiquetas que se utilizan para marcar los contenidos en un documento RSS. También se da una pincelada de la estructura y el aspecto de un documento RSS en sus diferentes formatos hasta llegar a los formatos utilizados actualmente, luego se describe la estructura básica de las versiones RSS 1.0, 2.0 y Atom 1.0.
Numero de paginas	44
Fecha de creación.	08/01/2010 11:29:00
Fecha última modificación.	22/01/2010 13:06:00

Sobre el autor

- Joaquín Medina Serrano [Web] [Mail]
- Fotógrafo erótico conceptual
- Autor de libros sobre Visual Basic, Cobol y Office
- He desarrollado diversas aplicaciones de gestión, utilizando herramientas como Clipper, Visual Basic 6 y Visual Basic .NET.
- Autor de artículos sobre Programación Orientada a Objetos, y Visual Basic .NET
- El último curso de capacitación realizado ha sido el Máster en Servicios Web, Seguridad Informática y Aplicaciones de Comercio Electrónico de la [Universidad de Zaragoza]

Tabla de contenido

1 Una breve historia de RSS 5
2 Crear orígenes de datos RSS desde el principio 7
2.1 Una visión preliminar de XML 8
2.2 Empezar por la declaración XML 8
2.3 Crear el elemento de documento 9
2.4 Creación de elementos XML 10
2.5 Creación de atributos XML 10
2.6 Anidar elementos XML 10
2.7 Documentos XML bien formados y válidos 12
3 Escribir documentos RSS 0.91 14
3.1 Declaración XML y definición DTD 14
3.2 El elemento <rss> 15
3.3 El elemento <channel> 15
3.3.1 El elemento <copyright> 16
3.3.2 El elemento <pubDate> 16
3.3.3 El elemento <lastBuildDate> 17
3.3.4 El elemento <docs> 17
3.3.5 El elemento <description> 18
3.3.6 El elemento <link> 18
3.3.7 El elemento <title> 19
3.3.8 El elemento <language> 20
3.3.9 El elemento <Image> 21
3.3.10 El elemento <managingEditor> 23
3.3.11 El elemento <webMaster> 24
3.3.12 El elemento <rating> 25
3.3.13 El elemento <skipHours> 25
3.3.14 El elemento <skipDays> 26
3.3.15 El elemento <item> 27
3.3.16 El elemento <textinput> 29
4 Escribir documentos RSS 2.0 33
4.1 El elemento <channel> 34
4.1.1 El elemento <category> del elemento <channel> 35
4.1.2 El elemento <generator> del elemento <channel> 35
4.1.3 El elemento <cloud> del elemento <channel> 35

4.1.4 El elemento <ttl> del elemento <channel> 36
4.1.5 El elemento <item> 37
4.2 Ampliar RSS 2.0 40
5 Escribir documentos Atom 41
6 Enlazar un origen de noticias a nuestra página HTML 42
6.1 Utilizando un enlace 42
6.2 Utilizar la técnica del Autodescubrimiento 43

1 Una breve historia de RSS

¿De dónde viene RSS? ¿Cuánto tiempo lleva entre nosotros? RSS ha tenido bastantes historias, algunas de las cuales entran en la categoría de folclore, con lo cual no puedo garantizar que mi comprensión de la historia de RSS sea cien por cien perfecta. De hecho, incluso el significado de RSS ha cambiado a lo largo de los años. Hay tres significados de RSS según la versión en la que introducidos:

- Rich Site Summary (Compendio enriquecido de sitios) para la versión 0.91 de RSS.
- RDF Site Summary (Compendio RDF de sitios) para la versión 0.90 y la versión 1.0.
- Real/y Simple Syndication (Sindicación realmente sencilla) para la versión 2.0 de RSS.

Hoy en día, la nomenclatura más aceptada es la última.

Así pues, ¿de dónde viene la idea de la sindicación? (suscripción al contenido de un sitio). Ya antes de RSS, existían otros formatos (podrían llamarse lenguajes) que soportaban la sindicación. Puede que recuerde el formato CDF (Channel Definition Format, formato de definición de canales) de Microsoft, con el que funcionaba la característica de canales activos de Internet Explorer. Utilizando CDF podíamos suscribirnos a varios sitios con publicaciones online. Internet Explorer comprobaba estas publicaciones de forma regular y descargaba los nuevos datos a medida que estaban disponibles. CDF se introdujo en 1997 y, en el mismo año, un desarrollador de Userland.com llamado Dave Winer, introdujo su propio formato de sindicación basado en XML en su blog de noticias Scripting News. Userland.com, cuya primera aparición tuvo lugar en 1997, se especializó en software para publicaciones Web y ha dejado una profunda huella en la historia de RSS.

La primera versión verdadera de RSS fue creada por Dan Libby de Netscape. Creada en 1999, se convirtió en la versión 0.90 (diseñada para emplearse en el portal de My Netscape). Enjulio de 1999, la versión 0.90 se modificó dando lugar a la aparición de la primera versión popular, la 0.9 1. En ese momento, RSS se traducía conocía con el nombre de Rich Site Summary (Compendio enriquecido de sitios). La versión 0.91 adoptó parte del formato de Dave Winer y se convirtió en una de las principales versiones de RSS.

Sin embargo, Netscape abandonó RSS no mucho después de eso dado que a pesar de que tenía algún uso, nadie quería hacerse cargo de él. El resultado fue el caos, como cabía esperar. Y esto fue lo que sucedió: apareció una lista de correo llamada RSS-DEV que se hizo muy popular. Al mismo tiempo, Dave Winer creó una versión modificada de RSS 0.9 1, que ya se había utilizado en el software de Userland.

Con el tiempo, las aproximaciones de RSS del grupo RSS-DEV y de Dave Winer fueron divergiendo, hecho que se dio a conocer en el mundillo con el nombre de bifurcación RSS. El grupo RSS-DEV creó la versión 1.0 de RSS en diciembre de 2000. Esta versión se alejaba bastante de las versiones anteriores y estaba fuertemente ligada al lenguaje Resource Description Format (Formato de descripción de recursos) basado en XML o en RDF (http://www.w3.org/TR/rdf-primer/), especialmente en la extensión RDF de Dublin Core (http://dublincore.org/). Así pues, RSS tomó el nombre de RDF Site Summary (Compendio RDF de sitios).

Sólo tres semanas después del anuncio de RSS 1.0, Dave Winer lanzó la versión 0.92 de RSS. RSS 1.0 fue también una versión importante, pero su sintaxis fue tan diferente de las otras versiones de RSS que la gente la encontró demasiado difícil de utilizar.

Entonces, las cosas se pusieron bastante turbulentas. En abril de 2001, Dave RS Winer lanzó RSS 0.93, similar a la versión 0.92. En agosto de 2001, apareció ni un borrador de la versión

0.94, que eliminaba los cambios realizados en la versión 0.93 e introducía unos cuantos cambios menores.
Finalmente, en septiembre de 2002, Dave Winer lanzó un sucesor de la versión 0.92 conocida como versión 2.0, ya que el nombre de versión 1.0 estaba ya cogido. En ese momento, RSS comenzó a conocerse como Real y Simple Syndication (Sindicación realmente sencilla). De hecho, las cosas se complicaron bastante ya que aparecieron varias versiones de RSS 2.0, aunque ahora ya se han estabilizado. RSS 2.0 era una ampliación de RSS 0.92, e incluía formas de que los usuarios ampliaran RSS por ellos mismos. RSS 2.0 se ha convertido en la principal versión de RSS.
En julio de 2003, Dave Winer transmitió la propiedad de RSS 2.0 al Berkman Center for Internet & Society de la Harvard Law School (en la dirección http://cyber.law.harvard.edu/), donde se encontraba trabajando en ese momento. Esta acción calmó las aguas en gran medida, y RSS 2.0 se estabilizó desde entonces.
Pero siguieron sucediendo cosas con RSS. En enero de 2005, Sean B. Palmer y Christopher Schmidt lanzaron un borrador de RSS 1 .1, que simplificaba el lenguaje en cierta medida. Sin embargo, en el momento de escribir este libro, RSS 1.1 todavía no había despegado. En noviembre de 2005, Microsoft propuso algunas extensiones de RSS, llamadas de forma informal Real Simple Synchronization (Sincronización sencilla real).
Hace aproximadamente unos cinco años varios desarrolladores, entre los que se encontraba Sam Ruby, ofrecieron al mundo un nuevo formato que se llamó Atom (primero fue conocido como Echo, luego como Pie y, finalmente, como Atom) como sustituto del formato RSS. Atom está basado en XML, pero es bastante más difícil de aprender y es más complejo que RSS. Atom mejora RSS añadiendo capacidades de internacionalización, estandarizando la sintaxis y permitiendo que los usuarios añadan sus propias características. Sin embargo, Atom todavía tiene que realizar un largo camino, en comparación con RSS.
Así, como puede ver, el desarrollo de RSS sigue en marcha, pero la situación se ha estabilizado. Hoy en día, las versiones 0.91, 1.0 y 2.0, así como Atom se utilizan con gran profusión.

2 Crear orígenes de datos RSS desde el principio

Hay muchos editores RSS que nos permiten crear un origen de datos RSS y publicarlo en un sitio Web. Pero si pretende una mayor sofisticación a la hora de manejar RSS, habrá ocasiones en las que prefiera hacer el trabajo por sí mismo. Por ejemplo, puede que necesite hacer algunas cosas que su editor no le permita o simplemente configurar una plantilla para sus orígenes de datos RSS y editar periódicamente dichos componentes.

Trabajar directamente con RSS, pone a nuestra disposición toda la potencia que nos ofrecen todo este conjunto de lenguajes. Pero ahora la pregunta es, ¿qué versión de RSS queremos utilizar?

Para estar seguro de que conoce cómo trabajan las distintas versiones de RSS, tendrá que leer las especificaciones correspondientes. Estas especificaciones le dirán cómo escribir el código XML real para dicha versión de RSS. Puede encontrar las especificaciones de las distintas versiones de RSS y Atom en los siguientes sitios:

- RSS 0.91. http://backend.userland.com/rss091
- RSS 0.92. http://backend.userland.com/rss092
- RSS 0.93. http://backend.userland.com/rss093
- RSS 1.0. http://web.resource.org/rss/1.0/spec
- RSS 1.1. http://inamidst.com/rss1.1/
- RSS 2.0. http://cyber.law.harvard.edu/rss/rss.html
- Atom. http://xml.coverpages.org/draft-ietf-atompub-format-11.txt

No podemos incluir todas las especificaciones en este capítulo, pero cubriremos la versión con la que empezó la explosión de RSS, la versión 0.91, que sigue utilizándose ampliamente. También incluiremos la versión 2.0 de RSS, la más popular con diferencia entre las que se utilizan hoy en día.

¿Qué pasa con la versión 1.0? Aunque también es de uso común, RSS 1.0 es un lenguaje diferente y no forma parte del árbol original de RSS. Basado principalmente en RDF (Resource Description Framework), RSS 1.0 se aleja del resto de las versiones de RSS. (RSS 2.0 es más un descendiente de RSS 0.91 de lo que lo es RSS 1.0.) Le proporcionaremos una visión preliminar de RSS 1.0. Si desea más detalles, eche un vistazo a la especificación de RSS 1.0 de la lista anterior.

También proporcionaremos ligeras nociones sobre Atom. Atom difiere de las especificaciones de RSS principalmente porque fue desarrollado por un equipo de estándares verdadero, el IETF (Internet Engineering Task Force). Sin embargo, Atom es un lenguaje completo y todavía no se ha adoptado con demasiada profusión. Las encuestas de la Web muestran que Atom todavía sigue teniendo un uso relativamente pequeño en comparación con RSS.

Mi opinión particular es que mientras que siempre habrá un lugar para Atom, es demasiado complejo para la mayoría de los desarrolladores de orígenes de datos RSS. Pocos tienen la paciencia o el tiempo necesario para trabajar con una especificación formal de más de 50 páginas antes de empezar a escribir. El encanto de RSS ha sido siempre su facilidad de uso y las complejas especificaciones de Atom se traducen en un tiempo excesivo para dominar el lenguaje. Los usuarios tuvieron los mismos problemas cuando el W3C (World Wide Web Consortium) introdujo XHTML (Extensible Hypertext Markup Language) como sucesor de

HTML. XHTML es tan complejo que nunca ha llegado a despegar. Por supuesto, Atom si podría llegar a despegar y si utilizamos editores RSS capaces de escribir código Atom, el proceso completo se hace muy sencillo. Atom tiene todavía un largo camino por recorrer, pero es una especificación importante, por lo que la trataremos en este capítulo.
RSS y Atom son lenguajes basados en XML y los archivos RSS y Atom son, en realidad, archivos XML. Así pues, para escribir RSS y Atom desde el principio, tendremos que conocer algunas de las reglas de XML. Y por ahí será por donde empecemos. Por una visión preliminar de XML.

2.1 Una visión preliminar de XML

Las siglas XML significan Extensible Markup Language (Lenguaje de etiquetas ampliable) y la palabra clave aquí es "ampliable". La idea es que podemos escribir y ampliar nuestro propio lenguaje desde el principio, siempre que sigamos las reglas de XML. XML es como HTML en el sentido en que está basado en texto encerrado dentro de distintos elementos. Pero en XML, somos nosotros mismos los que creamos estos elementos (a diferencia de HTML, donde los elementos ya están predefinidos.

> **Truco:** XML fue creado y mantenido por el W3C. Puede encontrar la especificación formal de la versión más reciente, XML 1.1, en la dirección http://www.w3.org/TR/xml11/ y la especificación de XML 1.0 (la versión utilizada para crear documentos RSS y Atom) en la dirección Web http://www.w3.org/TR/REC-xml/

2.2 Empezar por la declaración XML

La creación de documentos XML se comprende mejor con un ejemplo. Digamos que tenemos un grupo de empleados y que queremos llevar un seguimiento de los proyectos en los que están trabajando. Podríamos hacerlo con un documento XML y, como en cualquier otro documento XML (incluyendo los documentos RSS y Atom), tendremos que empezar con una declaración XML.

```
<?xml version = "1.0" encoding = "utf-8" ?>
```

Esta es la forma en la que deben empezar los documentos RSS y Atom, con la declaración que nos dice que se trata de un documento XML y que proporciona el número de versión con el que deseamos trabajar. Esta declaración tiene una forma especial, que empieza con <?xml y termina con ?>. Como los elementos HTML, esta declaración puede contener atributos, es decir, elementos tales como el número de versión o el tipo de codificación, como vemos en el ejemplo anterior.

> Nota: En la declaración XML, es obligatorio el número de versión, mientras que el tipo de codificación no lo es.

El atributo de versión define la versión de XML que deseamos utilizar para crear el documento y siempre debe ser 1.0 para RSS y Atom. El atributo de codificación define la codificación de caracteres, es decir, el conjunto de caracteres que se utiliza en el documento. UTF-8 (formato de transformación Unicode de 8 bits) es una buena elección para este atributo y, de hecho, es el valor por defecto para los documentos XML. Subconjunto de Unicode, UTF-8 coincide con el conjunto de caracteres ASCII que utiliza el programa WordPad de Microsoft Windows y la mayoría de los editores de texto de los países

de habla inglesa. Si queremos escribir nuestro origen de datos RSS en lenguajes con otros conjuntos de caracteres, tales como el español, tendremos que utilizar una codificación diferente.
Observe que cada atributo tiene asignado una cadena de texto entre comillas. En nuestro ejemplo, version=”1.0”. En XML y, por lo tanto, en RSS y Atom, también asignaremos cadenas de texto entrecomilladas a nuestros atributos. En HTML no necesitamos estas comillas y algunos atributos ni siquiera necesitan tener valores asignados. En XML, si utilizamos un atributo debemos asignarle un valor.

2.3 Crear el elemento de documento

Ahora que ya hemos empezado con la declaración XML, podemos añadir los elementos que conforman el cuerpo de un documento XML. Como en HTML, almacenaremos los datos de texto en elementos XML. A diferencia de HTML, construiremos nosotros los elementos que queramos utilizar en XML. Por ejemplo, si queremos crear un elemento XML para almacenar una dirección, podríamos hacerlo creando un nuevo elemento XML: <address>. Igual que en HTML, los elementos XML tienen una etiqueta de apertura y otra de cierre. Así pues, almacenaríamos la dirección de una persona de la siguiente forma:

```
<address>
    14 Picklewood Avenue
</address>
```

Hay algunas reglas para los nombres de elementos en XML: no pueden contener espacios, no pueden empezar por un número y no pueden empezar con un símbolo de puntuación. Estos son algunos nombres de elementos XML no válidos:

```
<phone number="">8905555</phone number>
<5feettall> John</5feettall>
```

El texto dentro del elemento se conoce como contenido del mismo y, si un elemento tiene algún contenido, necesita una etiqueta de apertura y una de cierre en XML. A diferencia de HTML, donde algunos elementos no necesitan una etiqueta de cierre (tales como los elementos <img> o <input>), XML requiere una etiqueta de cierre para los elementos que tienen algún contenido.
Los elementos que no tienen contenido se conocen como elementos vacíos en XML. Un elemento vacío no tiene ningún contenido de texto entre las etiquetas de apertura y de cierre, pero puede tener atributos. Los elementos vacíos de XML no necesitan una etiqueta de cierre, pero podernos utilizar el símbolo /> para finalizar un elemento vacío si queremos.

```
<data language = "es-ES" />
```

Nota: Los datos de los elementos vacíos se almacenan como a tributos, no como contenido entre las etiquetas de apertura y de cierre.

El primer elemento en todo documento XML es el elemento documento. Este elemento contiene a los restantes elementos del documento (como en HTML, un elemento que contiene a otros). Por ejemplo:

```
        <identidad>
            <apellidos>Connery</apellidos>
```

```
            <nombre>Sean</nombre>
        </identidad>
```

En nuestro ejemplo, llamaremos al elemento documento <documento>, para dejar claro que se trata de dicho elemento, pero podemos emplear cualquier nombre de elemento válido en XML (en un documento RSS, por ejemplo, utilizamos <rss> como elemento documento):

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<documento>
        . . .
        . . .
</documento>
```

Ahora, podemos empezar a añadir a nuestro documento los elementos que contendrán los datos que deseamos almacenar.

2.4 Creación de elementos XML

Digamos que queremos almacenar algunos datos de nuestros dos empleados. Podríamos crear dos elementos <empleado> y añadirlos a nuestro nuevo documento:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<documento>
    <empleado>
        . . .
        . . .
    </empleado>
    <empleado>
        . . .
        . . .
    </empleado>
</documento>
```

Ahora, disponemos de dos nuevos elementos XML en nuestro documento. Hasta ahora, todo va bien.

2.5 Creación de atributos XML

Como sucede con HTML, en XML podemos añadir atributos a un elemento, empezando por la etiqueta de apertura. Por ejemplo, podríamos querer añadir un atributo de estado para el elemento <empleados> que indique el estado del empleado (en activo o jubilado):

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<documento>
    <empleado status="retirado">
        . . .
        . . .
    </empleado>
    <empleado status="activo">
        . . .
        . . .
    </empleado>
</documento>
```

2.6 Anidar elementos XML

¡Fantástico! Ahora estamos en el camino. En XML, los elementos pueden contener bien datos de texto u otros elementos, de forma que podríamos añadir los datos que necesitamos sobre nuestros empleados utilizando un elemento <identidad> (para el nombre

completo) que contenga a su vez dos elementos anidados: <apellidos> (para los apellidos) y <nombre> (para el nombre de pila):

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<documento>
    <empleado status="retirado">
        <identidad>
            <apellidos>Connery</apellidos>
            <nombre>Sean</nombre>
        </identidad>
        . . .
        . . .

    </empleado>
    <empleado status="activo">
        <identidad>
            <apellidos >Hepburn</apellidos>
            <nombre>Audrey</nombre>
        </identidad>
        . . .
        . . .

    </empleado>
</documento>
```

Podemos añadir más información sobre cada empleado, tal como su fecha de contratación y los proyectos en los que está trabajando:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<documento>
    <empleado status="retirado">
        <identidad>
            <apellidos>Connery</apellidos>
            <nombre>Sean</nombre>
        </identidad>
        <fechaAlta>October 15, 2006</fechaAlta>
        <proyectos>
            <proyecto>
                <producto>Printer</producto>
                <id>111</id>
                <precio>111.00€</precio>
            </proyecto>
            <proyecto>
                <producto>Laptop</producto>
                <id>222</id>
                <precio>989.00€</precio>
            </proyecto>
        </proyectos>
    </empleado>
    <empleado status="activo">
        <identidad>
            <apellidos >Hepburn</apellidos>
            <nombre>Audrey</nombre>
        </identidad>
        <fechaAlta>Octolber 20, 2006</fechaAlta>
        <proyectos>
            <proyecto>
                <producto>Desktop</producto>
                <id>333</id>
                <precio>2995.00€</precio>
            </proyecto>
            <proyecto>
                <producto>Scanner</producto>
                <id>444</id>
                <precio>200.00€</precio>
            </proyecto>
        </proyectos>
    </empleado>
</documento>
```

Este es un documento XML completo. Para crear nuestro archivo, podemos utilizar cualquier editor de texto normal del que dispongamos, como WordPad en Windows. Asegúrese de guardar su documento como un archivo de texto sencillo. Cuando guarde su documento en WordPad, elija la opción Documento de texto, no el formato "Formato de texto enriquecido (RTF)", que incluye todo tipo de códigos de formato que convertirían el documento en un formato XML no válido.

2.7 Documentos XML bien formados y válidos

Existen dos criterios más: los documentos XML deben estar bien formados y ser válidos. Hay varias reglas que necesitamos seguir para hacer que nuestros documentos XML estén bien formados y que podemos encontrar en las especificaciones de XML. La regla más importante dice que cada documento XML debe tener solamente un elemento documento y que dicho elemento debe contener los restantes elementos del documento. También debemos evitar cualquier error de anidación. Eche un vistazo al siguiente código XML, donde todo está anidado correctamente.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
        <proyectos>
            <proyecto>
                <producto>Printer</producto>
                <id>111</id>
                <precio>111.00€</precio>
            </proyecto>
            <proyecto>
                <producto>Laptop</producto>
                <id>222</id>
                <precio>989.00€</precio>
            </proyecto>
        </proyectos>
```

El siguiente, no es un documento XML bien formado, ya que el primer elemento <proyecto> no termina antes de que empiece el siguiente, creando así un error de anidación. En otras palabras. Los dos elementos <proyecto> se mezclan:

```
<proyectos>
    <proyecto>
        <producto>Desktop</producto>
        <id>333</id>
        <precio>2995.00€</precio>
    <proyecto>
    </proyecto>
            <producto>Scanner</producto>
            <id>444</id>
            <precio>200.00€</precio>
        </proyecto>
</proyectos>
```

¿Qué hace que un documento XML sea válido? Cuando creamos un documento XML, podemos especificar su gramática o sintaxis. Por ejemplo, ¿qué atributos puede tener el elemento <empleado>? ¿Qué elementos debe contener el elemento <proyecto>? Y así, sucesivamente. Hoy en día hay dos formas de especificar la gramática de un documento XML: podemos utilizar una definición DTD (Document Type Definition, definición de tipo de documento) o emplear un esquema XML. Ambos métodos nos permitirán comprobar si un documento RSS o Atom se ajusta a la sintaxis correcta. Todo lo que tendremos que hacer es recurrir a un validador XML (como http://www.stg.brown.edu/service/xmlvalid/) y comprobar nuestro documento con los estándares de la definición DTD de RSS o Atom o con

el esquema XML. Dado que podemos encontrar validadores RSS online (como http://www.feedvalidator.org/), no es necesario que entremos aquí en detalle sobre las definiciones DTD o los esquemas XML, pero es altamente recomendable que valide siempre sus orígenes de datos RSS o Atom.

En esta introducción de XML, le hemos proporcionado los fundamentos básicos que necesitará para escribir documentos RSS y Atom. Por supuesto, no hemos cubierto todas las reglas de XML. Hay libros enteros dedicados a este tema si desea más información, O bien, eche un vistazo online a las especificaciones de XML, si tiene estómago para hacerlo.

XML en sí no es realmente un lenguaje, a pesar de su nombre. Realmente, es un conjunto de reglas que podemos utilizar para escribir nuestro propio lenguaje. Así es como RSS y Atom vieron la luz: sus autores crearon un conjunto de elementos, tales como el elemento <rss>, que se ajustan a las reglas de XML. En otras palabras, utilizaron las reglas de XML para crear nuevos lenguajes (RSS y Atom) con sus propios elementos prefabricados.

Ahora, pasemos a echar un vistazo a RSS, empezando por su versión 0.91

3 Escribir documentos RSS 0.91

Aquí tiene un documento RSS 0.91 más complejo, al que nos referiremos a lo largo de esta sección:

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>Copyright 2005. </copyright>
        <pubDate>Wed, 14 Dec 2005 07:00:00 GMT</pubDate>
        <lastBuildDate>Mon, 12 Dec 2005 07:00:00 GMT</lastBuildDate>
        <docs>http://www.rssmanjac.com/steve/jntohtml</docs>
        <description>This feed contains news from Steve!</description>
        <link>
            http: //www.rssmaniac.com/steve
        </link>
        <title>Steve's News!</title>
        <language>en-us</language>
        <image>
            <title>Steve's News</title>
            <uri>http://www.rssmariiac.com/steve/Image.jpg</uri>
            <link>http://www.rssmaniac.com/steve</link>
            <description>Steve s News</description>
            <width>144</width>
            <height>36</height>
        </image>
        <managingEditor>steve@rssmaniac .com (Steve) </managingEditor>
        <webMaster>steve@rssmaniac.com (Steve) </webMaster>
        <skipHours>
            <hour>8</hour>
            <hour>9</hour>
            <hour>1O</hour>
        </skipHours>
        <skipDays>
            <day>Sunday</day>
        </skipDays>
        <item>
            <title>Steve shovels the snow</title>
            <description>It snowed once again.</description>
            <link>http://www.rssmaniac.com/ateve</link>
        </item>
        <textinput>
            <title>Search for other items</title>
            <description>What do you want to find?</description>
            <name>search </name>
            <link>http//www.rssmaniac.com/tind.php</link>
        </textinput>
    </channel>
</rss>
```

Echemos un vistazo a este documento pieza a pieza.

3.1 Declaración XML y definición DTD

Como deben hacerlo todos los documentos RSS, este empieza con una declaración XML. Esta declaración incluye solamente el atributo de versión obligatorio, al que debe asignarse el valor "1.0" en RSS 0.91. Observe también el elemento <DOCTYPE>.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
```

El elemento <!DOCTYPE> especifica la definición DTD de RSS 0.91 (que puede encontrarse en http: //my.netscape.com/publish/formats/rss-091.dtd), de forma que nuestro lector RSS pueda comprobar la sintaxis correcta de nuestro documento si quiere hacerlo. Aunque el elemento <!DOCTYPE> que hemos escrito aquí es oficialmente obligatorio en RSS 0.9 1, mucha gente lo omite y muchos validadores RSS no insisten demasiado en ello.
Aquí se plantean algunos aspectos técnicos. La declaración XML dispone de un atributo opcional, la codificación, que nos permite especificar el conjunto de caracteres que queremos emplear en nuestro origen de datos. Por ejemplo, si fuéramos japoneses especificaríamos una codificación que soportase caracteres japoneses. El valor por defecto es UTF-8, que es una versión condensada de Unicode que incluye todos los caracteres ASCII estándar que es emplean la mayoría de los editores de texto. Los nombres válidos son cualquier nombre estándar IANA o nombre MIME (por ejemplo ISO-8859-1, Windows-1251, UTF-8)

3.2 El elemento <rss>

El elemento <rss> es el elemento documento y da comienzo a la parte de almacenamiento de datos de un documento RSS 0.91.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
        . . .
</rss>
```

Recuerde que, dado que este es el documento elemento, contiene todos los restantes elementos del documento (la declaración XML y el elemento <!DOCTYPE> no son parte del elemento documento). El atributo de versión es obligatorio y debe tener el valor 0. 91 para un documento RSS 0.91.
Hay un elemento hijo del elemento <rss>, el elemento <channel>, que es obligatorio.

3.3 El elemento <channel>

El elemento <channel> contiene toda la información necesaria para configurar un canal en particular. Cada elemento <rss> debe contener sólo un elemento <channel>. El elemento <channel> no tiene atributos.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        . . .
    </channel>
</rss>
```

El elemento <channel> tiene varios elementos hijo, unos obligatorios y otros opcionales. Los elementos obligatorios son los siguientes:

- <description>.
- <language>.
- <link>.
- <title>.

Y estos son los elementos hijos opcionales que se incluyen dentro de un elemento <channel>:

- <copyright>.
- <docs>.
- <image>.
- <item> (se permite un máximo de 15 elementos <item>).
- <lastBuildDate>.
- <managingEditor>.
- <pubDate>.
- <rating>.
- <skipDays>.
- <skipHours>.
- <textinput>.
- <webMaster>.

3.3.1 El elemento <copyright>

El elemento <copyright> es uno de los elementos opcionales que hay dentro del elemento <channel>. Contiene la información de copyright del origen de datos.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        . . .
    </channel>
</rss>
```

Este elemento no tiene atributos y permite a su vez elementos hijos.

3.3.2 El elemento <pubDate>

El elemento <pubDate>, otro de los elementos hijos opcionales del elemento <channel>, contiene la fecha en la que se ha publicado el canal:

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        . . .
    </channel>
</rss>
```

No hay ningún formato especial para la fecha y la hora. Queda en manos del usuario. El elemento <pubDate> no tiene elementos hijos ni atributos.

Más información sobre Formato de Fechas
El consorcio W3c ha publicado un estándar para la publicación de fechas en internet cuya dirección es: http://www.w3.org/TR/NOTE-datetime y también el documento "Date and Time Formats" ("Formatos para Fechas y Horas"), Nota del W3C, M. Wolf y C. Wicksteed, 15 de septiembre de 1997. Revisada el 27 de agosto de 1998. Este documento se encuentra en http://www.w3.org/TR/1998/NOTE-datetime-19980827 que refleja el estándar ISO 8601 - ISO 8601:2004, publicada el 3 de diciembre de 2004. http://es.wikipedia.org/wiki/ISO_8601 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874

3.3.3 El elemento <lastBuildDate>

Como su nombre indica, el elemento <lastBuildDate> contiene la fecha de la última publicación del documento.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        . . .
    </channel>
</rss>
```

El elemento <lastBuildDate> es un elemento hijo opcional del elemento <channel>. No tiene elementos hijos ni atributos, así que es muy sencillo. Y no tiene ningún formato especial para registrar la fecha. (ver comentario sobre fechas en la etiqueta <PubDate>)
La última fecha de publicación puede ser muy útil. Entre otras cosas, nos dice la frecuencia con la que se actualiza el origen de datos (así que, si no actualiza sus orígenes de datos con mucha frecuencia, puede que desee omitir este elemento).

3.3.4 El elemento <docs>

El elemento <docs> contiene la dirección URL en la que se puede obtener más información y una descripción sobre el canal.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
```

```
        <docs>
http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        . . .
    </channel>
</rss>
```

Este elemento, otro de los elementos hijos opcionales del elemento <channel>, no tiene elementos hijos ni atributos. Simplemente contiene una dirección URL con documentación sobre el canal de datos (de ahí su nombre). Aquí es donde podemos ofrecer a los lectores más información sobre el canal, como archivos de ayuda en un sitio Web para que puedan obtener el máximo aprovechamiento.
Observe que las direcciones deben empezar por http: // o por ftp: //, no por www.

3.3.5 El elemento <description>

Como un campo HTML en el que introducimos texto, el elemento <description> almacena una descripción de texto para un canal, un componente o un control de entrada de texto. Es un elemento hijo obligatorio del elemento <channel>. Cada elemento <channel> debe tener un elemento <description> hijo. El elemento <description> también es un elemento hijo obligatorio de los elementos <item> y <textinput>. En nuestro ejemplo, este elemento describe nuestro nuevo canal.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        . . .
    </channel>
</rss>
```

El elemento <description> no tiene ningún elemento hijo ni ningún atributo. Es un elemento importante, de forma que deberemos utilizarlo para añadir el texto apropiado para nuestro canal. Este es el texto que se mostrará en el lector RSS cuando el usuario consulte las propiedades de nuestro origen de datos.

3.3.6 El elemento <link>

El elemento <link> es un elemento hijo obligatorio de los elementos <channel>, <image>, <item> y <textinput> y representa una dirección URL sobre la que el usuario puede hacer clic. Cuando se utiliza como elemento hijo del elemento <channel>, normalmente almacena la dirección de la página de inicio del creador del canal o de su empresa. Cuando se utiliza como elemento hijo de un elemento <item>, es un vínculo a la información completa en un

sitio Web. En nuestro documento RSS 0.91, lo hemos utilizado para definir un vínculo para el nuevo canal:

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
            http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>

http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010_Links_RSS20.xml
        </link>
        . . .
    </channel>
</rss>
```

Observe que la dirección URL debe empezar por http: / / o ftp: / /, no por www.

3.3.7 El elemento <title>

El elemento <title> es un elemento obligatorio de los elementos <channel>, <image>, <item> y <textinput> y, como ya sabe, almacena el titular del canal, componente o control de entrada de texto. El elemento <title> se utiliza para dar un título al nuevo canal.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
           http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>

http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010_Links_RSS20.xml
        </link>
        <title>La Gueb de Joaquin, Noticias</title>
        . . .
    </channel>
</rss>
```

El título del canal aparece en la ventana de orígenes del lector RSS. El titular de un elemento aparece en la ventana de titulares. El título de un control de entrada de texto, se utiliza para la etiqueta de dicho control.

3.3.8 El elemento <language>

El elemento <language>, un elemento hijo obligatorio del elemento <channel>, especifica el idioma del canal. En nuestro ejemplo, hemos especificado un idioma español de España.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
            http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>

http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010_Links_RSS20.xml
        </link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        . . .
    </channel>
</rss>
```

El elemento <language> no tiene atributos ni elementos hijos. El contenido de este elemento (por ejemplo es-ES), se conoce como código de idioma. Hay numerosos códigos de idioma disponibles para RSS 0.91

Nota: Los lectores RSS utilizan los códigos de idioma para determinar el idioma que utiliza un origen de datos, pero para seleccionar un conjunto de caracteres para dicho idioma, no olvide configurar el atributo de codificación de la declaración XML.

Más información sobre Lenguajes

[ISO639]

Códigos para la representación de nombres de idiomas, ISO 639:1988. Para más información consulte http://www.iso.ch/cate/d4766.html. Puede consultar también http://www.oasis-open.org/cover/iso639a.html.

[ISO3166]

Códigos para la representación de nombres de países, ISO 3166:1993.

[W3c - Etiquetas de idioma en HTML y XML]

http://www.w3.org/International/articles/language-tags/Overview.es.php

La sintaxis de las etiquetas de idioma se encuentra definida en la BCP 47 de la IETF. BCP significa 'Best Current Practise' (mejor práctica actual) y es un nombre duradero para una serie de RFC cuyos números cambian conforme se actualizan. El último RFC donde se

describe la sintaxis de las etiquetas de idiomas es el RFC 4646, Tags for the Identification of Languages, que reemplaza a los RFC anteriores 3066 y 1766. Antes, para hallar las etiquetas había que consultar las listas de códigos presentes en diversos estándares de la ISO, pero ahora se pueden hallar todas las etiquetas en el IANA Registro de subetiquetas de idioma.

3.3.9 El elemento <Image>

Como sabe, el elemento <image> se utiliza para conectar una imagen al canal que se muestra en el lector RSS.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
            http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>

http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010_Links_RSS20.xml
        </link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
       <image>
            <title>La Güeb de Joaquín</title>
            <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
            <url>http://jms32.eresmas.net/web2008/graficos/red/logo_gj01.gif</url>
            <description>Este es el Logotipo de mi canal</description>
            <width>186</width>
            <height>33</height>
        </image>
        . . .
    </channel>
</rss>
```

El elemento <image> no tiene atributos, pero puede tener varios elementos hijos. Los elementos hijos obligatorios son los siguientes:

- <title>.
- <url>.
- <link>.

Y estos son los elementos hijos opcionales:

- <description>.
- <width>.
- <height>.

A continuación, echaremos un vistazo a algunos de estos elementos. Algunos, tales como <width> y <height> tienen restricciones particulares que debería conocer.

3.3.9.1 El elemento <title> del elemento <image>

El elemento <title> proporciona un título para la imagen.

```
<image>
    <title>La Güeb de Joaquín</title>
</image>
```

Los lectores RSS a menudo muestran este título si, por alguna razón, no se puede cargar la imagen. El elemento <title>, que es obligatorio en el elemento <image>, no tiene ningún elemento hijo ni ningún atributo.

3.3.9.2 El elemento <url> del elemento <image>

La función del elemento <url> no nos depara ninguna sorpresa. Almacena la dirección URL de la imagen y es un elemento hijo obligatorio del elemento

```
<image>
    <title>La Güeb de Joaquín</title>
    <url>http://jms32.eresmas.net/web2008/graficos/red/logo_gj01.gif</url>
</image>
```

Este elemento no tiene ningún elemento hijo ni ningún atributo. Observe que la dirección URL debe empezar por http: // o por ftp: //, no por el prefijo www.

3.3.9.3 El elemento <link> del elemento <image>

El elemento<link> es un elemento hijo obligatorio del elemento <image> y contiene la dirección URL a la que nos llevará el lector RSS si el usuario hace clic sobre el vínculo.

```
<image>
    <title>La Güeb de Joaquín</title>
    <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
    <url>http://jms32.eresmas.net/web2008/graficos/red/logo gj01.gif</url>
</image>
```

La dirección URL del elemento <link> debe empezar por http: // o por ftp: //. Normalmente, esta dirección URL apunta a la página principal del sitio al que corresponde el origen de datos o a una página con una explicación más detallada del contenido. h

3.3.9.4 El elemento <description> del elemento <image>

El elemento <description> es un elemento hijo opcional del elemento <image> y, como habrá adivinado, está diseñado para incluir una descripción del origen de datos que normalmente colocamos en el elemento <title> (obligatorio) de la imagen.

```
<image>
    <title>La Güeb de Joaquín</title>
    <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
    <url>http://jms32.eresmas.net/web2008/graficos/red/logo gj01.gif</url>
    <description>Este es el Logotipo de mi canal</description>
</image>
```

3.3.9.5 El elemento <width> del elemento <image>

Podemos informar al lector RSS del ancho de una imagen pero, igual que sucede con los navegadores Web, no es necesario que lo hagamos, ya que el lector RSS puede calcularlo una vez que la imagen se ha cargado por completo.

```
        <image>
             <title>La Güeb de Joaquín</title>
             <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
             <url>http://jms32.eresmas.net/web2008/graficos/red/logo_gj01.gif</url>
             <description>Este es el Logotipo de mi canal</description>
             <width>186</width>
         </image>
```

El texto contenido en este elemento es un valor entero positivo que define la anchura de la imagen en píxeles y su valor debe encontrarse entre 1 y 144, ambos incluidos (si, 144 es el valor máximo). Así pues, no podemos utilizar una imagen de 800 píxeles de ancho en nuestro origen de datos. Algunos lectores utilizan un valor por defecto si no especificamos la anchura de nuestra imagen. Por alguna razón, este valor es 88 píxeles. Este elemento no tiene elementos hijos ni atributos.

3.3.9.6 El elemento <height> del elemento <image>

El elemento <height> del elemento <image> es opcional y especifica la altura de la imagen en píxeles.

```
        <image>
             <title>La Güeb de Joaquín</title>
             <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
             <url>http://jms32.eresmas.net/web2008/graficos/red/logo_gj01.gif</url>
             <description>Este es el Logotipo de mi canal</description>
             <width>186</width>
             <height>33</height>
         </image>
```

El texto contenido en este elemento debe ser un entero positivo entre 1 y 400, ambos inclusive. Esto significa que las imágenes pueden ser bastante más altas que anchas. El valor por defecto para documentos RSS 0.91, por alguna razón, es de 31 píxeles.
Este elemento no tiene elementos hijos ni atributos.

3.3.10 El elemento <managingEditor>

El elemento <managingEditor> sirve para ofrecer a los lectores de nuestros orígenes de datos información de contacto. Es un elemento hijo opcional del elemento <channel>.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
            http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
```

```
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>
            http://jms32.eresmas.net/indicesAnuales/2010 Links RSS20.xml
        </link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        <image>
            . . .
        </image>
        <managingEditor>joaquin@medina.name</managingEditor>

        . . .
    </channel>
</rss>
```

Formalmente hablando, este elemento debería contener la dirección de correo electrónico del editor del origen de datos, no sólo un nombre. Si alguien lo desea, debería poder utilizar esta dirección de correo electrónico. Este elemento no tiene elementos hijos ni atributos.

3.3.11 El elemento <webMaster>

El elemento <webMaster>, otro elemento hijo del elemento <channel>, almacena la dirección de correo electrónico de la persona responsable de gestionar cualquier problema técnico con nuestro origen de datos. Es distinto del elemento <managingEditor>, que almacena la dirección de correo electrónico de la persona responsable del contenido del sitio.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>
            http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010/info.html
        </docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>

http://jms32.eresmas.net/web2008/indicesAnuales/2010/2010_Links_RSS20.xml
        </link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        <image>
            . . .
        </image>
        <managingEditor>joaquin@medina.name</managingEditor>
         <webMaster>departamento@tecnico.com</webMaster>

        . . .
    </channel>
</rss>
```

Este elemento no tiene elementos hijos ni atributos.
La dirección no lleva el protocolo [mailto:]

3.3.12 El elemento <rating>

En la práctica, el elemento <rating> se utiliza con muy poca frecuencia, motivo por el cual no lo hemos incluido en nuestro ejemplo de documento RSS 0.91. El elemento <rating>, un elemento hijo opcional del elemento <channel>, establece una clasificación PICS de terceros (plataforma para la selección de contenidos de Internet) de nuestro origen de datos RSS. Entre otras cosas, este sistema de clasificación se ha diseñado para evitar el acceso de los menores a contenido adulto. (Puede encontrar una lista de organizaciones de clasificación PICS en http://www.w3.org/PICS/raters.htm , pero la mayoría de los vínculos están difuntos.)

Si quiere utilizar este elemento, el texto que debe colocar normalmente empieza por PICS-1.1 “, seguido de la dirección URL de la agencia de clasificación y su clasificación, dando como resultado algo como esto:

```
        <rating>(PICS-1.1 "http://www.picswatcher.org/" (A+))</rating>
```

3.3.13 El elemento <skipHours>

El elemento <skipHours> nos permite configurar el intervalo de horas en el que no se actualizará nuestro origen de datos, si optamos por especificar este parámetro. Este elemento no está en sus mejores días pero, cuando se desarrolló RSS, los usuarios daban por sentado que los orígenes de datos debían actualizarse cada hora. Por esta razón, este elemento fue diseñado para ofrecer una forma de especificar a qué horas no se actualizaba un origen de datos.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>http://jms32.eresmas.net/2010/info.html</docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>http://jms32.eresmas.net/2010_Links_RSS20.xml</link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        . . .
        . . .
        <skipHours>
            . . .
        </skipHours>
        . . .
        . . .
    </channel>
</rss>
```

El elemento <skipHours> no tiene atributos, pero debe incluir al menos un elemento hijo <hour>.

3.3.13.1 El elemento <hour> del elemento <skipHours>

El elemento <hour>, un elemento hijo del elemento <skipHours>, contiene una hora del día, medida en el horario GMT del meridiano de Greenwich, en el que no se actualiza nuestro origen de datos.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>http://jms32.eresmas.net/2010/info.html</docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>http://jms32.eresmas.net/2010_Links_RSS20.xml</link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        . . .
        . . .
        <skipHours>
            <hour>8</hour>
            <hour>9</hour>
            <hour>10</hour>
        </skipHours>
        . . .
        . . .
    </channel>
</rss>
```

3.3.14 El elemento <skipDays>

Como el elemento <skipHours>, el elemento <skiplDays> nos permite indicar los días de la semana en que no se actualiza nuestro origen de datos.

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>http://jms32.eresmas.net/2010/info.html</docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>http://jms32.eresmas.net/2010_Links_RSS20.xml</link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        . . .
        . . .
        <skipDays> </skipDays>
            . . .
        . . .
```

```
    </channel>
</rss>
```

El elemento <skipDays> no tiene atributos pero, si lo utiliza, debe incluir al menos un elemento hijo <day>.

3.3.14.1 El elemento <day> del elemento <skipDays>

Mediante este elemento, especificamos un día de la semana en inglés (la especificación dice que debemos utilizar el inglés para los nombres de días). Por ejemplo, si nuestro origen de datos no se actualiza los domingos, podemos incluir un elemento <skipDays> con un elemento hijo <day> como este:

```
<?xml version="1.0"?>
<!DOCTYPE rss SYSTEM
"http=://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>http://jms32.eresmas.net/2010/info.html</docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>http://jms32.eresmas.net/2010_Links_RSS20.xml</link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
        . . .
        . . .
        <skipHours>
            <hour>10</hour>
        </skipHours>
        <skipDays>
            <day>Sunday</day>
        </skipDays>
        . . .
        . . .
    </channel>
</rss>
```

Podemos incluir hasta siete elementos <day> dentro de un elemento <skipDays>, cada uno de ellos correspondiente a un día diferente de la semana. El elemento <day> no tiene elementos hijos ni atributos.

3.3.15 El elemento <item>

Este es el más importante de todos, el elemento <item>, que representa una página Web o una sección de una página Web. Como sabe, este elemento se utiliza para crear los componentes de un origen de datos. En RSS 0.91, hay un límite de 15 elementos hijos <item> dentro del elemento <channel>.

```
        <item>
            <title>Implementación de la técnica Arrastrar y soltar</title>
            <description>
              . . .
            </description>
            <pubDate>2009-12-24T13:32:23</pubDate>
```

```
            <language>es-ES</language>
            <author>
                name=Joaquin Medina Serrano;
                uri=http://joaquin.medina.name/;
                email= joaquin@medina.name;
                companyia=jms32
            </author>
            <link>
              http://jms32.eresmas.net/documentos/2009_12_24_DragAndDrop.html
            </link>
            <guid />
            <category domain="">informática</category>
        </item>
```

El elemento <item> tiene dos elementos hijos obligatorios:

- <title>.
- <link>.

Y un elemento hijo opcional:

- <description>.

3.3.15.1 El elemento <title> del elemento <item>

Como ya vimos en el capítulo 3, un elemento <title> de un componente contiene el titular de dicho componente y es el texto que aparecerá en la ventana de titulares en un lector RSS.

```
        <item>
            <title>Implementación de la técnica Arrastrar y soltar</title>
        </item>
```

3.3.15.2 El elemento <description> del elemento <item>

El elemento <description> contiene una descripción del componente y, generalmente, incluye parte del texto completo del componente. Aunque es un elemento hijo opcional del elemento <item>, verá que está presente en prácticamente todos los componentes RSS de su lector RSS.

Este elemento no tiene atributos ni elementos hijos. Observe que es un elemento obligatorio dentro de un elemento <item>.

```
<item>
  <title>Implementación de la técnica Arrastrar y soltar</title>
    <description>
      Visual Basic .NET no permite el empleo automático de arrastrar y soltar.
      Hay que escribir código que inicie una operación de arrastrar y soltar
      en el control origen y  escribir código apara los  eventos  contenidos
      en el control de destino. De hecho, la única propiedad que se encuentra
      relacionada con arrastrar y soltar AllowDrop se debe definir como
      True para que un control desencadene  eventos  cuando trabaje como
      destino de una operación de arrastrar y soltar.
   </description>
</item>
```

3.3.15.3 El elemento <link> del elemento <item>

El elemento <link> es también obligatorio dentro de un elemento <item> y, como ya sabe, contiene la dirección URL con el contenido completo del componente.

```
        <item>
```

```
            <title>Implementación de la técnica Arrastrar y soltar</title>
            <description>
              . . .
            </description>
          <link>
http://jms32.eresmas.net/web2008/documentos/informatica/lenguajes/puntoNET/visualBasic/2009_12_24_DragAndDrop.html
          </link>
       </item>
```

El vínculo del componente aparece bajo su descripción en el lector RSS y el usuario puede hacer clic sobre él para acceder a la información adicional. Como sucede con otras direcciones URL, deben empezar por el prefijo http: / / o bien ftp://.

3.3.16 El elemento <textinput>

RSS 0.91 proporciona un elemento <textinput> opcional que los lectores pueden utilizar para realizar sus consultas, realizar búsquedas en nuestro sitio, proporcionar información, etc. Este elemento <textinput> aparece en forma de campo de texto. Los lectores pueden introducir texto y enviarlo con un clic en una dirección URL que debemos proporcionar. Este elemento es un elemento hijo opcional del elemento <channel> y su uso no está muy generalizado hoy en día. No encontrará muchos lectores RSS que lo soporten, ya que se ha convertido en un estándar la gestión de este tipo de interacción con los lectores dentro del propio sitio Web.

```
        <textinput>
            <title>Busqueda de Items</title>
            <description>¿Que es lo que quieres buscar?</description>
            <name>search</name>
            <link>http://jms32.eresmas.net/2010/find.php</link>
        </textinput>
```

El elemento <textinput> no tiene ningún atributo, pero incluye cuatro elementos hijos obligatorios:

- <title>.
- <description>.
- <name>.
- <link>.

3.3.16.1 El elemento <title> del elemento <textinput>

El elemento <title> nos permite mostrar un título para el control <textinput>. El título del control <textinput> de nuestro ejemplo es "`Busqueda de Items`".

```
        <textinput>
            <title>Busqueda de Items</title>
        </textinput>
```

Este elemento no tiene ningún elemento hijo ni ningún atributo.

3.3.16.2 El elemento <description> del elemento <textinput>

El elemento <description> del control <textinput> nos permite mostrar una descripción, tal como una etiqueta, para el control <textinput>.

```
<textinput>
    <title>Busqueda de Items</title>
    <description>¿Que es lo que quieres buscar?</description>
    <name>search</name>
    <link>http://jms32.eresmas.net/2010/find.cgi</link>
</textinput>
```

Este elemento no tiene elementos hijos ni atributos.
El elemento <name> del elemento <textinput>
Cuando enviamos datos a un programa online, la información de un control HTML se asocia con el nombre del control. Los script online y los programas pueden, de esta forma, recuperar los datos utilizando dicho nombre. En nuestro ejemplo, el nombre del control de entrada de texto es “search” (búsqueda).

```
<textinput>
    <title>Busqueda de Items</title>
    <description>¿Que es lo que quieres buscar?</description>
    <name>search</name>
</textinput>
```

Así pues, por ejemplo, si estamos enviando nuestros datos a un script PHP en el servidor, podríamos recuperar los datos que ha introducido el usuario en el control de entrada de texto con la expresión PHP $_REQUEST [“search”]. Este elemento tampoco tiene elementos hijos ni atributos.

3.3.16.3 El elemento <link> del elemento <textinput>

El elemento <link> del control <textinput> almacena una dirección URL. Esta es la dirección donde queremos enviar el texto que el usuario ha introducido en el control. Es la dirección URL del script o programa que se encargará de manejar el texto, tal como el programa que ejecuta la búsqueda del usuario. En nuestro ejemplo de documento RSS 0.91, los datos de texto que introduce el usuario se envían a un script PHP llamado find.php:

```
<textinput>
    <title>Busqueda de Items</title>
    <description>¿Que es lo que quieres buscar?</description>
    <name>search</name>
    <link>http://jms32.eresmas.net/2010/find.php</link>
</textinput>
```

Como es habitual, la dirección URL debe empezar por http: // o ftp: // (y, dado que estamos pasando los datos a un script, lo más probable es que queramos utilizar http: //). Este elemento no tiene elementos hijos ni atributos.
Con esto completamos la sintaxis de RSS 0.91. No está nada mal. Ya está en camino de convertirse en un profesional de RSS.

3.3.16.4 Escribir documentos RSS 1.0

La versión 1.0 de RSS es bastante diferente de la versión 0.9 1. Este es el aspecto que tendría nuestro documento RSS 0.91 anterior en formato RSS 1.0.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
    xmlns:rdf="http://wwww.3.org/1999/02/22-rdf-syfltax-ns#"
    xmlns="http://purl.org/rss/1.0/">
    <channel rdf:about="http://www.rssmaniac.com/steve">
        <title>Steve's News</title>
        <link>http://www.rssrnaniac.com/steve</link>
        <description>
            This feed contains news from Steve!
        </description>
        <image rdf:resource="http://www.rssmaniac.com/steve/Image.jpg" />
        <items>
            <rdf:Seq>
                <rdf:li resource= "http://xm1.com/pub/2000/08/09/xs1t/xs1t.html"
/>
            </rdf:Seq>
        </items>
    </channel>

    <image rdf:about="http://www.rssmaniac.com/steve/Image.ipg">
        <title>Steves News</title>
        <link>http://www.rssmaniac.com/steve</link>
        <url>http: //www. rssmaniac com/steve/Image.jpg</url>
    </image>
    <item rdf:about="http://www.rssmaniac.com/steve/about.html">
        <title>Steve shovels the snow</title>
        <link> http: //www. rssmaniac com/steve</link>
        <description>
            It snowed once again. Time to shovel!
        </description>
    </item>
</rdf:RDF>
```

Como puede ver, un documento RSS 1.0 varía mucho de un documento RSS 0.91. Por una parte, la organización de los datos es diferente. Por ejemplo, el elemento <channel> se cierra entes de empezar la lista de elementos <item>. También podemos crear un resumen de los elementos <item> que contiene el documento dentro de elementos <rdf : li>. En este sentido, el elemento documento en si es diferente. Empieza con <rdf:RDF> y no con <rss>. El redundante nombre del elemento <rdf:RDF> enfatiza la conexión RSS 1.0 con RDF, el formato de descripción de recursos.

Pero, ¿por qué algunos nombres de elementos empiezan por rdf: "? Este prefijo representa un espacio de nombres en XML. Dado que estamos construyendo nuestros propios nombres de elementos en XML, hay muchas posibilidades de que uno de nuestros nombres entre en conflicto por el nombre definido por alguna otra persona, de forma que es necesario introducir espacios de nombres. Los espacios de nombres corresponden a prefijos en documentos XML.

Así, aunque definamos un elemento llamado <address> y alguien más decida utilizar el mismo nombre de elemento, podemos utilizar prefijos de espacios de nombres tales como <mine:address> y<theirs:address> para que ambos elementos no entren en conflicto.

Podemos configurar un espacio de nombres en el elemento documento asociándolo a una cadena de texto única (dado que esta cadena es única, los elementos de nuestro espacio de nombres serán diferentes y no entrarán en conflicto con otros elementos). Normalmente, se utiliza una dirección URL como cadena de texto única, ya que los recursos Web deben tener una dirección URL única. Por ejemplo, esta URL podría ser nuestra página de inicio.

En el elemento documento<rdf:RDF>, un documento RSS 1.0 conecta el prefijo rdf con el conjunto estándar de elementos RDF y configura un espacio de nombres por defecto para el resto del documento. (Si omitimos este prefijo en un elemento, los lectores RSS asumirán que se encuentra en el espacio de nombres por defecto.)

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
    xmlns:rdf="http://wwww.3.org/1999/02/22-rdf-syfltax-ns#"
    xmlns="http://purl.org/rss/1.0/">
```

Ahora que hemos utilizado el elemento <rdf:RDF>, si omitimos cualquier prefijo, se asumirá que nuestros elementos pertenecen al espacio de nombres por defecto, que se corresponde con el espacio de nombres de RSS 1.0 (que en nuestro ejemplo se ha definido para corresponder con la dirección URL http://purl.org/rss/1.0/). Cuando utilicemos espacios de nombres, nuestros elementos XML no entrarán en conflicto con los de nadie y este espacio de nombres tiene una ventaja distinta: podemos crear y añadir nuestros propios elementos XML a un documento RSS 1.0. Todo lo que tendremos que hacer es definir un nuevo prefijo de espacio de nombres. A continuación, podremos construir y añadir nuestros propios elementos al documento siempre que los precedamos del nuevo prefijo. Por ejemplo, podríamos querer añadir una fecha a algunos elementos para mostrar en qué momento fueron creados:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
    xmlns:rdf="http://wwww.3.org/1999/02/22-rdf-syfltax-ns#"
    xmlns:joaquin="http://joaquin.medina.name/about.html"
    xmlns="http://purl.org/rss/1.0/">
    <channel rdf:about="http://www.rssmaniac.com/steve">
        <joaquin:date>2010/01/25</joaquin:date>
        <title>Steve's News</title>
        <link>http://www.rssrnaniac.com/steve</link>
        <description>
            This feed contains news from Steve!
        </description>
        <image rdf:resource="http://www.rssmaniac.com/steve/Image.jpg" />
        <items>
            <rdf:Seq>
                <rdf:li resource= "http://xm1.com/pub/2OOO/O8/O9/xs1t/xs1t.htm1"
/>
            </rdf:Seq>
        </items>
    </channel>

    <image rdf:about="http://www.rssmaniac.com/steve/Image.ipg">
        <title>Steves News</title>
        <link>http://www.rssmaniac.com/steve</link>
        <url>http: //www. rssmaniac com/steve/Image.jpg</url>
    </image>
    <item rdf:about="http://www.rssmaniac.com/steve/about.html">
        <joaquin:date>2010/01/25</joaquin:date>
        <title>Steve shovels the snow</title>
        <link> http: //www. rssmaniac com/steve</link>
        <description>
            It snowed once again. Time to shovel!
        </description>
    </item>
</rdf:RDF>
```

De hecho, existe una extensión de RDF llamada Dublin Core que ya dispone de numerosos elementos incorporados que podemos añadir a RDF, incluyendo un elemento <date> (fecha).Así, cuando utilizamos elementos de Dublin Core, normalmente los precedemos con el prefijo dc y conectamos dicho prefijo con el espacio de nombres definido para Dublin Core

(en la dirección http://purl.org/dc/elements/1.1/) De esta forma, podremos añadir fechas a cualquier elemento en RSS 1.0.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
    xmlns:rdf="http://wwww.3.org/1999/02/22-rdf-syfltax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns="http://purl.org/rss/1.0/">
    <channel rdf:about="http://www.rssmaniac.com/steve">
        <dc:date>2010/01/25</dc:date>
        <title>Steve's News</title>
        <link>http://www.rssrnaniac.com/steve</link>
        <description>
            This feed contains news from Steve!
        </description>
        <image rdf:resource="http://www.rssmaniac.com/steve/Image.jpg" />
        <items>
            <rdf:Seq>
               <rdf:li resource= "http://xml.com/pub/2000/08/09/xslt/xslt.html" />
            </rdf:Seq>
        </items>
    </channel>

    <image rdf:about="http://www.rssmaniac.com/steve/Image.ipg">
        <title>Steves News</title>
        <link>http://www.rssmaniac.com/steve</link>
        <url>http: //www. rssmaniac com/steve/Image.jpg</url>
    </image>
    <item rdf:about="http://www.rssmaniac.com/steve/about.html">
        <dc:date>2010/01/25</dc:date>
        <title>Steve shovels the snow</title>
        <link> http: //www. rssmaniac com/steve</link>
        <description>
            It snowed once again. Time to shovel!
        </description>
    </item>
</rdf:RDF>
```

Como puede ver, RSS 1.0 se basa en gran medida en RDF (no de la forma en la que se ha escrito la versión de RSS más popular, la 2.0). Muchos autores RSS quieren reclamar RSS como su propio lenguaje, no como un sublenguaje de RDF.

4 Escribir documentos RSS 2.0

Ahora, aquí tiene nuestro documento RSS en formato 2.0, tal como lo creamos con NewzAlert Composer en el capítulo 3. Observe que NewzAlert Composer rodea los datos de texto con las etiquetas XML <!CDATA y]] > para ocultarlo de los programas sensibles al análisis XML. No necesitamos utilizar estas etiquetas en nuestros documentos RSS 2.0. Basta con colocar el texto que deseamos almacenar directamente en nuestro documento.

```
<?xml version="1.0"?>
<rss version="2.0">
    <channel>
        <copyright>
            En la red desde 1997 -
            Joaquín Medina Serrano -
            Todos los Derechos reservados
        </copyright>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <docs>http://jms32.eresmas.net/2010/info.html</docs>
        <description>
            Este canal tiene los mejores artículos de Internet,
            es decir, todos los de mi Web
        </description>
        <link>http://jms32.eresmas.net/2010_Links_RSS20.xml</link>
        <title>La Gueb de Joaquin, Noticias</title>
        <language>es-ES</language>
```

```
        <generator uri="http://joaquin.medina.name/"
                   version="2008.10.04">
            DllSindicacion
        </generator>

        <item>
            <title>Implementación de la técnica Arrastrar y soltar</title>
            <description>
                 Visual Basic .NET no permite el empleo
                 automático de arrastrar y soltar.
            </description>
            <pubDate>2009-12-24T13:32:23</pubDate>
            <link>
http://jms32.eresmas.net/web2008/documentos/informatica/lenguajes/puntoNET/visualB
asic/2009_12_24_DragAndDrop.html
            </link>
        </item>
    </channel>
</rss>
```

RSS 2.0 es realmente el sucesor de RSS 0.91 y RSS 0.92, no de RSS 1.0.
Estas son las diferencias en RSS 2.0:

- Ya no hay una limitación de 15 elementos por canal: no hay ninguna limitación en cuanto al número de elementos en un canal.
- RSS adopta restricciones sobre el primer carácter que no es un espacio en los datos de los elementos <link> y <url>. Las direcciones URL de los elementos <link> y < url > deben empezar por http: //, https : //, news : //, mailto: o ftp: // (ya no estamos limitados solamente a http: // o ftp: //).
- El elemento <!DOCTYPE> ya no es obligatorio.
- Debemos utilizar <rss version= “2.0O”> en lugar de <rss version= “0.91”> como elemento documento.
- Hay nuevos elementos y atributos.

Siga leyendo para un examen detallado de los nuevos elementos y atributos de RSS 2.0.

4.1 El elemento <channel>

Aquí tiene una lista de los elementos obligatorios del elemento <channel> en RSS 2.0. Observe que el elemento <language>, que era obligatorio en RSS 0.91, ahora es un elemento hijo opcional del elemento <channel>.

- <title>.
- <link>.
- <description>.

Y estos son los elementos hijos opcionales del elemento <channel> en la versión 2.0 de RSS:

- <language>.
- <copyright>.
- <managingEditor>.
- <webMaster>.
- <pubDate>.
- <lastBuildDate>.
- <category>.
- <generator>.
- <docs>.
- <cloud>.
- <ttl>.
- <image>.

- <rating>.
- <textInput>.
- <skipHours>.
- <skipDays>.

De estos elementos, <category>, <generator>, <cloud> y <ttl> son nuevos y no aparecen en la versión 0.91.

4.1.1 El elemento <category> del elemento <channel>

A media que nos suscribimos a más y más orígenes de datos en un lector RSS, puede que queramos que nuestro lector nos permita organizarlos en varias carpetas o categorías, tal como lo hace RSSReader. El elemento <category> ofrece una sugerencia de categoría para el origen de datos y podemos incluir tantos elementos <category> como queramos. Este sería el aspecto de un elemento <category> en un documento RSS 2.0:

```
<?xml version="1.0"?>
<rss version="2.0">
    <channel>
        <category domain="">informática</category>
```

Este elemento funciona en cierto modo como las categorías que se establecen al final de algunos libros y que utilizan las bibliotecas para su ordenación (ficción/misterio y cosas similares).
Este elemento es opcional en el elemento <channel> y tiene un atributo también opcional, <domain>, que normalmente especifica una dirección URL donde podemos encontrar más información sobre la categoría.

4.1.2 El elemento <generator> del elemento <channel>

El elemento opcional <generator> almacena un texto que identifica el programa que se ha utilizado para crear el archivo. Por ejemplo, nuestro documento RSS 2.0 se creó con NewzAlert Composer.

```
        <title>La Gueb de Joaquin, Noticias</title>
        <category domain="">informática</category>
        <generator> DllSindicacion -  version: 2008.10.04> </generator>
```

Este elemento es un elemento hijo opcional del elemento <channel> y no tiene atributos.

4.1.3 El elemento <cloud> del elemento <channel>

El elemento <cloud> nos permite interactuar con una aplicación Web que soporta la interfaz rssCloud. Como mucho, se asume que los orígenes de datos RSS deben actualizarse cada hora pero, en algunas ocasiones, esto no es suficiente. Por dicha razón, podemos registrarnos en una interfaz rssCloud para obtener actualizaciones más rápidas. Los programas registrados con la interfaz reciben inmediatamente la notificación de las nuevas actualizaciones.

Referencia [http://www.elnuevoparquet.com/blog.1/tag/rsscloud/]
¿Alguien sabe cómo funciona la sindicación RSS? Casi nadie, en realidad, pero si hubiera que describirlo de manera bien sencilla diríamos que un cliente averigua cada cierto rato si se ha

publicado material nuevo en una 'fuente' de nuestro interés. Por otra parte, las noticias tardan un tiempo en publicarse. A veces es cuestión de minutos, a veces cuestión de horas. ¿Hay una manera de recibir esa noticia de forma inmediata? Pues sí, la tecnología RSSCloud, una especie de complemento elimina esta demora, permitiendo a las 'fuentes de noticias' eliminar la demora que existe entre la publicación, y la aparición en nuestro lector RSS

Para que RSSCloud funcione, es necesario que tanto el servidor de noticias (por ejemplo un blog), como el lector (por ejemplo Google Reader, Safari, Outlook, lo que sea), tengan soporte para esta tecnología para actualizar tus fuentes al instante en vez de chequear cada X minutos. Esta mañana (07/ene/2010) el único cliente que lo soportaba era River 2, [http://newsriver.org/river2.html] pero hace pocas horas LazyFeed se subió al carro [http://www.readwriteweb.com/archives/lazyfeed_1st_independent_rss_aggregator_declares_s.php] y declaró que también soportarán RSS Cloud.

Este elemento tiene cinco atributos obligatorios: domain, el dominio de la interfaz, port, el puerto del servidor por el que se realiza la comunicación, path, la carpeta de la aplicación de registro en el servidor, registerProcedure, el nombre de la aplicación de registro y protocol, el protocolo de comunicación online que, generalmente, es XML-RPC (Remote Procedure Cali, llamada de procedimiento remoto) o SOAP (Simple Object Access Protocol, protocolo de acceso a objetos sencillos).

```
<?xml version="1.0"?>
<rss version="2.0">
    <channel>
       <title>La Gueb de Joaquin, Noticias</title>
       <category domain="">informática</category>
       <generator uri="http://joaquin.medina.name/"
                  version="2008.10.04">DllSindicacion</generator>
        <cloud domain="rssmaniac.com"
               port="80"
               path="/RPC2"
               registerprocedure="register"
               protocol="soap" />
```

4.1.4 El elemento <ttl> del elemento <channel>

El último de los nuevos elementos opcionales el <ttl>. TTL significa Time To Live (tiempo de vida) y representa el número de minutos que puede permanecer almacenado un canal en la caché antes de intentar actualizarlo desde su origen. Siempre se ha asumido que los canales se actualizan cada hora pero, si escribimos nuestro propio origen de datos, probablemente esto sea demasiado asumir (a menos, claro está, que seamos unos completos fanáticos).
Este sería el código para actualizar un origen de datos cada dos horas (120 minutos) en un documento RSS 2.0:

```
<?xml version="1.0"?>
<rss version="2.0">
    <channel>
        <title>La Gueb de Joaquin, Noticias</title>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <ttl>120</ttl>
```

4.1.5 El elemento <item>

También hay cambios en el elemento <item> en RSS 2.0. Estos son los elementos hijos obligatorios del elemento <item>:

- <title>.
- <link>.

Los elementos hijos opcionales del elemento <item> son los siguientes:

- <description>.
- <author>.
- <category>.
- <comments>.
- <enclosure>.
- <guid>.
- <pubDate>.
- <source>.

¿Cuáles son los elementos nuevos? Los elementos opcionales <author>, <category>, <comments>, <enclosure>, <guid> y <source>.

4.1.5.1 El elemento <author> del elemento <item>

El elemento <author> contiene no el nombre del autor, sino su dirección de correo electrónico. La idea es que los lectores de nuestro origen de datos puedan estar en contacto con el autor de un componente determinado, especialmente si el autor no es el responsable de todo el canal.

```
<item>
      <title>Implementación de la técnica Arrastrar y soltar</title>
      <description>
          Visual Basic .NET no permite el empleo
          automático de arrastrar y soltar.
      </description>
      <pubDate>2009-12-24T13:32:23</pubDate>
      <author>joaquin@medina.name</author>
      . . .
      . . .
  </item>
```

Este es un elemento muy necesario si tiene varias personas escribiendo para el mismo origen de datos, como cada vez es más frecuente. Si el lector tiene información de contacto sólo par el canal, tendrá que contactar con el responsable del canal para llegar a un colaborador específico. Proporcionar la dirección de correo electrónico del autor de cada componente en concreto, es mejor idea.
Este elemento opcional no tiene elementos hijos ni atributos.

4.1.5.2 El elemento <category> del elemento <item>

El elemento hijo <category> del elemento <item> nos permite definir una categoría para un componente, de forma que se puedan organizar los distintos componentes de nuestro origen de datos. Este elemento opcional podría almacenar una categoría tal como <category>Música</category> que proporcione información sobre dónde encaja el componente. Podemos definir tantos elementos <category> como queramos.

```
<item>
      <title>Implementación de la técnica Arrastrar y soltar</title>
      <description>
          Visual Basic .NET no permite el empleo
          automático de arrastrar y soltar.
      </description>
      <pubDate>2009-12-24T13:32:23</pubDate>
      <author>joaquin@medina.name</author>
      <category domain="">informática</category>
      <category domain="">lenguajes</category>
      <category domain="">NET</category>
      <category domain="">System</category>
      <category domain="">Windows</category>
      <category domain="">Forms</category>
      . . .
      . . .
  </item>
```

El elemento <category> tiene un atributo opcional, domain. Este atributo normalmente es una dirección URL donde podemos obtener más información sobre la categoría.

4.1.5.3 El elemento <comments> del elemento <item>

Podríamos pensar que el elemento <comments> del elemento <item> contiene comentarios sobre un componente, pero no es esta la forma en que está diseñado. En lugar de ello, se supone que este elemento debe contener una dirección URL que apunte a una página con comentarios sobre el componente.

```
<item>
      <title>Implementación de la técnica Arrastrar y soltar</title>
      <description>
          Visual Basic .NET no permite el empleo
          automático de arrastrar y soltar.
      </description>
      <pubDate>2009-12-24T13:32:23</pubDate>
      <author>joaquin@medina.name</author>
      <category domain="">informática</category>
      <comments>
           http://jms32.eresmas.net/web2008//comxnents .html
      </comments>
      . . .
      . . .
  </item>
```

Este elemento opcional no tiene elementos hijos ni atributos.

4.1.5.4 El elemento <enclosure> del elemento <ítem>

El elemento hijo <enclosure> del elemento <item> es de vital importancia. Este es el elemento que hace posible las técnicas de podcast (añadir multimedia a nuestros orígenes de datos.”).

```
<item>
      <title>Implementación de la técnica Arrastrar y soltar</title>
      <description>
          Visual Basic .NET no permite el empleo
          automático de arrastrar y soltar.
      </description>
      <pubDate>2009-12-24T13:32:23</pubDate>
      <author>joaquin@medina.name</author>
      <category domain="">informática</category>
      <comments>
           http://jms32.eresmas.net/web2008//comxnents .html
      </comments>
      <enclosure url="http://jms32.eresmas.net/web2008/shoveljng.mp3"
```

```
                    length="4823902"
                    type="audio/mpeg" />
          . . .
          . . .
      </item>
```

Este elemento opcional nos permite incluir, por ejemplo, archivos MP3 en nuestro origen de datos. Hay tres atributos obligatorios en este elemento. El atributo un nos ofrece la localización online del adjunto, el atributo length contiene su longitud en bytes y el atributo type define el tipo MIME del adjunto, como “audio/mpeg” para MP3 (para una lista de los tipos MIME posibles, véase la dirección http://iana.org/assignments/media-types/).
En este elemento, la dirección URL debe empezar por http: //.

4.1.5.5 El elemento <guid> del elemento <ítem>

Es posible (especialmente si tiene varios autores escribiendo en su origen de datos) que pueda enviar dos componentes con el mismo título. Algunos lectores RSS simplemente echaran un vistazo al título, decidirán que ya se ha leído y seguirán adelante. ¿Cómo podemos evitar esto?
Podemos asignar a cada componente de nuestro origen de datos un identificador único global o guid (globaly unique identifler). Un elemento <guid> está compuesto por una cadena de texto que identifica de forma única a un componente. Los lectores RSS pueden comprobar los elementos <guid> para tener la certeza de que dos componentes son diferentes.
Un elemento <guid> puede contener cualquier cadena de texto, siempre que sea única. Por ejemplo, podría ser la dirección URL de nuestro sitio Web (y, dado que las direcciones URL son únicas, funcionaría correctamente, siempre que no volvamos a reutilizarla).

```
    <item>
          <title>Implementación de la técnica Arrastrar y soltar</title>
          <description>
              Visual Basic .NET no permite el
              empleo automático de arrastrar y soltar.
          </description>
          <pubDate>2009-12-24T13:32:23</pubDate>
          <author>joaquin@medina.name</author>
          <category domain="">informática</category>
          <comments>http://jms32.eresmas.net/web2008//comxnents.html </comments>
          <enclosure url="http://jms32.eresmas.net/web2008/shoveljng.mp3"
                    length="4823902"
                    type="audio/mpeg" />
          <guid>a0f50cbe-f38f-42c7-abe0-eeb15aa58d65</guid>
          . . .
          . . .
      </item>
```

Otra posibilidad es elegir simplemente un número aleatorio grande (así es como Microsoft crea los identificadores que se utilizan como etiquetas de volumen en los discos formateados) o una cadena de caracteres aleatorios grande. Si hay suficientes dígitos o letras aleatorias en un identificador, es muy probable que no coincida con ningún otro.
No hay reglas para crear nuestros propios identificadores en RSS 2.0. Solamente deben ser cadenas de texto únicas.
El elemento <guid> tiene un atributo llamado isPermaLink, que podemos configurar como true (verdadero) o false (falso). Si configuramos este atributo como true, el lector RSS sabrá que la cadena de identificación es una dirección URL, un vínculo permanente (una URL que no cambia con el tiempo, al menos no tan rápidamente como lo hacen otras direcciones).

4.1.5.6 El elemento <source> del elemento <ítem>

En RSS 2.0, los elementos <item> pueden contener un elemento <source>. Estos elementos contienen el título del canal RSS que origina el componente. Es un elemento útil cuando un origen de datos contiene componentes de otros orígenes de datos (por ejemplo, cuando creamos un nuevo origen de datos en base a un término de búsqueda). Siempre que nuestro origen de datos contenga componentes de otros orígenes de datos o fuentes, será una buena idea utilizar el elemento <source>.

```
<source url="http://jms32.eresmas.net/web2008/news.xml">
      Otras noticias interesantes
</source>
```

Este elemento tiene un atributo obligatorio, url, que contiene el archivo fuente XML del origen de datos (en otras palabras, la dirección URL que podemos utilizar para leer el origen de datos). Así pues, si utilizamos este elemento, tendremos que incluir no sólo el título del origen de datos del que procede el componente, sino también la dirección URL de dicho origen de datos (por supuesto, pidiendo permiso antes de incluir componentes de otras personas en nuestro origen de datos).

4.2 Ampliar RSS 2.0

Igual que sucede con RSS 1.0, podemos ampliar RSS 2.0. Si queremos crear y añadir nuestros propios elementos XML a un documento RSS 2.0, podemos hacerlo. Sólo tendremos que preocuparnos porque estos nuevos elementos tengan su propio espacio de nombres.
Por ejemplo, podríamos querer añadir un nuevo espacio de nombres utilizando el prefijo joaquin a un documento RSS. Así es como tendríamos que hacerlo (recuerde que debe asignar una cadena única como prefijo para su espacio de nombres).

```
<?xml version="1.0"?>
<rss version="2.0"
     xmlns:joaquin="http://jms32.eresmas.net/2010/about.html" >
    <channel>
        <title>La Gueb de Joaquin, Noticias</title>
        <pubDate>2010/01/25T12:54:32</pubDate>
        <lastBuildDate>2010/01/23T07:00:15</lastBuildDate>
        <joaquin:Afiliacion>Datos y Programas, SA.</joaquin:Afiliacion>
        . . .
        . . .
        <item>
            . . .
            . . .
        </item>
    </channel>
</rss>
```

Ahora, ya puede crear y añadir sus propios elementos, siempre que utilice como prefijo su nuevo espacio de nombres:
Observe que, incluso añadiendo nuestros nuevos elementos XML a RSS 2.0 de esta forma, no hay ninguna garantía de que los lectores RSS sepan cómo manejarlos. y si no pueden hacerlo, simplemente ignorarán nuestros orígenes de datos, a menos que utilicemos nuestro propio software especializado escrito para manejar estos nuevos elementos.

Nota: A diferencia de un documento RSS 1.0, un documento RSS 2.0 no define un espacio de nombres por defecto para todo el documento. La razón es la compatibilidad descendente de RSS 2.0 con las versiones 0.91 y 0.92, que no

contienen espacios de nombres por defecto. Así pues, un documento RSS 0.91 o RSS 0.92 también es un documento RSS 2.0 válido.

5 Escribir documentos Atom

Los documentos Atom requieren unos cuantos conocimientos más para crearlos partiendo desde cero, ya que la especificación de Atom es más complicada (tiene más de 50 páginas). Aquí tiene algunas diferencias entre RSS 2.0 yAtom:

- Los componentes de Atom se llaman entradas.
- No hay una etiqueta <channel> (de todas formas, no tendría ningún uso real).
- Se utilizan etiquetas <id> en lugar de etiquetas <guid>.
- Todos los elementos pertenecen al espacio de nombres por defecto de Atom.
- Los vínculos deben incluir los atributos rel, type y href.

Aquí tiene nuestro ejemplo RSS traducido a un formato de documento Atom 1.0 válido. Observe las diferencias entre los elementos comparados con un documento RSS 2.0.

```
<?xml version="1.0" encoding="utf-8" ?>
<feed xmlns="http://www.w3.org/2005/Atom/">
    <generator uri="http://joaquin.medina.name/"
               version="2008.10.04"
               xmlns="">DllSindicacion</generator>
    <id xmlns="" />
    <updated schema="ISO 8610:2004"
             xmlns="">2009-12-24T13:48:15</updated>
    <title type="text"
           xmlns="">La Gueb de Joaquin</title>
    <subtitle type="text"
              xmlns="">Paginas y documentos incluidos en mi web durante 2010 (La
web de Joaquin)</subtitle>
    <link rel="alternate"
          type="text/html"

href="http://jms32.eresmas.net/web2008/indicesAnuales/2008/2008_Links_ATOM10.xml"
          xmlns="" />
    <link rel="self"
          type="application/atom+xml"

href="http://jms32.eresmas.net/web2008/indicesAnuales/2008/2008_Links_ATOM10.xml"
          xmlns="" />
    <author xmlns="">
        <name>Joaquin Medina Serrano</name>
        <email>joaquin@medina.name</email>
        <uri>http://joaquin.medina.name/</uri>
        <companyia>jms32</companyia>
    </author>
    <contributor xmlns="" />
    <rights xmlns="">
        En la red desde 1997 - Joaquin Medina Serrano -
        Derechos reservados - Se puede copiar informacion
        de este sitio siempre que
        A) se haga mencion de la copia realizada y
        B) Se coloque un enlace al documento copiado
    </rights>
    <category xmlns="" />
    <logo xmlns="">
        <title>La Gueb de Joaquin</title>
        <link>http://jms32.eresmas.net/web2008/indicePagina.html</link>
        <url>http://jms32.eresmas.net/web2008/graficos/red/logo_gj01.gif</url>
        <height>33</height>
        <width>186</width>
    </logo>
    <entry xmlns="">
        <id />
        <title type="text">Implementación de la técnica Arrastrar y soltar</title>
        <!-- Fecha &#250;ltima modificaci&#243;n: jueves, 24 de diciembre de 2009
a las 13:32:23 horas-->
        <updated schema="ISO 8610:2004">2009-12-24T13:32:23</updated>
        <author>
```

```
            <name>Joaquin Medina Serrano</name>
            <email>joaquin@medina.name</email>
            <uri>http://joaquin.medina.name/</uri>
            <companyia>jms32</companyia>
        </author>
        <link
href="http://jms32.eresmas.net/web2008/documentos/informatica/lenguajes/puntoNET/v
isualBasic/2009_12_24_DragAndDrop.html"
              type="text/xhtml+xml"
              title="Implementación de la técnica Arrastrar y soltar"
              charset="ISO 8859-1"
              length="49417" />
        <summary type="text">
            Visual Basic .NET no permite el empleo automático de arrastrar y
soltar.
            Hay que escribir código que inicie una operación de arrastrar y soltar
            en el control origen y  escribir código apara los  eventos  contenidos
            en el control de destino. De hecho, la única propiedad que se
encuentra
            relacionada con arrastrar y soltar AllowDrop se debe definir como True
            para que un control desencadene  eventos  cuando trabaje como destino
            de una operación de arrastrar y soltar.
        </summary>
        <category term="informática" />
        <category term="lenguajes" />
        <category term="NET" />
        <category term="System" />
        <category term="Windows" />
        <category term="Forms" />
        <!-- Fecha publicacion: jueves, 24 de diciembre de 2009 a las 13:32:23
horas-->
        <published schema="ISO 8610:2004">2009-12-24T13:32:23</published>
        <rights type="text">
              © Copyright Joaquin 'jms32®' Medina Serrano 1.997-2010 -
              Reservados todos los derechos.
        </rights>
    </entry>
</feed>
```

Los orígenes de datos Atom son un poco más complejos y largos que los documentos RSS 2.0, pero si echa un vistazo a nuestro documento de ejemplo, debería resultarle comprensible o, al menos, familiar.

6 Enlazar un origen de noticias a nuestra página HTML

6.1 Utilizando un enlace

Para hacer accesible tu canal RSS debes poner un enlace a él desde tu sitio web. Bastará con que insertes una línea como la que se muestra a continuación en el código HTML de la página en la que quieras que aparezca el enlace:

```
<a type="application/rss+xml" href="nombre de tu canal RSS.xml">
  <img src="imagen que quieres que sirva de enlace.gif">
</a>
```

Mediante los atributos de la etiqueta <a> indicamos que la página a la que enlaza es un documento RSS y la dirección web en la que se haya dicho documento.

Con la etiqueta <img> señalamos la fuente de la imagen que nos va a servir de botón de enlace a nuestro feed.

6.2 Utilizar la técnica del Autodescubrimiento

Los navegadores de la actualidad pueden descubrir los orígenes de datos RSS en nuestras páginas. Normalmente, nuestro origen de datos muestra un icono para hacerle saber al visitante de nuestra página que puede suscribirse a él. Pero para hacer que un lector RSS o navegador desabran nuestro origen de datos automáticamente (una técnica que se conoce con el nombre de autodescubrimiento), tenemos que incluir un elemento <link> dentro de la sección <head> de nuestra página

Si nuestro origen de datos es un documento RSS tenemos que incluir el siguiente código HTML

```
<link rel="alternate"
      type="application/rss+xml"
      title="RSS"
      href="http://jms32.eresmas.net/web2008/indicesAnuales/2010_Links_RSS20.xml"
/>
```

Observe que el atributo ”href” de nuestro origen de datos debe contener una dirección URL , el atributo “rel”, debe contener el valor “altérnate” y el atributo “type” contener el valor “appliction/rss+xml” (lo que contenga el atributo “title” es cosa nuestra)

Si tenemos un origen de datos ATOM , deberemos utilizar el siguiente código HTML

```
<link rel="alternate"
      type="application/atom+xml"
      title="Atom"
      href="http://jms32.eresmas.net/web2008/indicesAnuales/2010_Links_ATOM10.xml"
/>
```

Los orígenes ATOM y RSS pueden convivir junto, y además no están limitados a uno solo, puede haber varias etiquetas <link>. Véase el siguiente código HTML

```
<!-- Descubrimiento RSS y ATOM -->

<link rel="alternate"
      type="application/rss+xml"
      title="RSS"
      href="http://jms32.eresmas.net/indicesAnuales/2010_Links_RSS20.xml" />

<link rel="alternate"
      type="application/rss+xml"
      title="RSS"
      href="http://jms32.eresmas.net/indicesAnuales/2010_LinksExternos_RSS20.xml"
/>

<link rel="alternate"
      type="application/atom+xml"
      title="Atom"
      href="http://jms32.eresmas.net/indicesAnuales/2010_Links_ATOM10.xml" />

<link rel="alternate"
      type="application/atom+xml"
      title="Atom"
      href="http://jms32.eresmas.net/indicesAnuales/2010_LinksExternos_ATOM10.xml"
/>
```

Referencia Bibliográfica
Los secretos de RSS
Steven Holzner
Anaya multimedia
ISBN 978-84-415-2130-8

Página en blanco