



XML y Adjuntos

**Ministerio de Economía Fomento y
Reconstrucción**

Santiago, octubre 2009

ÍNDICE

1. Objetivo.....	2
2. Incorporación de adjuntos en documentos XML.....	2
2.1. Incluir el adjunto o apuntar a una ubicación del adjunto	2
2.1.1. Incluir el adjunto	2
2.1.2. Apuntar o referenciar a la dirección del adjunto	5
3. Visualización de un documento XML con adjunto	6
3.1. Data URI scheme	6
3.2. Desarrollo a medida.....	7
4. Elección de formatos de adjuntos.....	8
5. Envío de archivos en los servicios Web	8
5.1. Estándares.....	9
5.2. MTOM	9
5.2.1. Capas de un Servicio WEB utilizando MTOM.....	10
5.2.2. Flujo de la optimización MTOM.....	10
6. Consideraciones de archivos XML	11
7. Referencias.....	12

1. Objetivo

El presente documento tiene como objetivo explicar las alternativas técnicas para el envío de documentos adjuntos en una estructura XML y algunos métodos de visualización, como por ejemplo en el caso de la estructura de oficio, como incluir una imagen, un pdf, etc..

2. Incorporación de adjuntos en documentos XML

Para completitud de información en algunos documentos XML se requiere incorporar piezas adjuntas o documentos como parte de la estructura de datos a manejar, unos de los ejemplos más frecuente es adjuntar documentos escaneados en formato de imagen.

El primer análisis está enfocado en poder saber si se pueden modelar estos datos en formato XML y evaluar técnicas de captura de estos datos.

En muchos casos no se puede por la naturaleza misma de la fuente como por ejemplo, un documento XML de aceptación de obra de construcción donde se adjuntan los planos.

Para resolver este tema se manejan dos alternativas de solución:

- Incluir los documentos adjuntos en el XML.
- Apuntar desde el XML a un repositorio donde se encuentren almacenados los documentos.

2.1. Incluir el adjunto o apuntar a una ubicación del adjunto

Al momento de modelar el documento XML en la etapa de creación del esquema, se deben considerar las características del proceso de negocio que se quiere desarrollar y de las características de los documentos a incluir para tomar la decisión correcta.

2.1.1. Incluir el adjunto

Al incluir un documento dentro de una representación XML se ve como una cadena de caracteres como se muestra a continuación:

```
<documentoXML>  
...  
    <boleta>TU0AKgAAQAgAA....</boleta>  
    <carta>TU0AKgAAQAgAA....</carta>  
</documentoXML>
```

Ventajas:

- El documento XML contiene todos los datos modelados. (Ej.: datos modelados+boleta+carta).
- Cuando se intercambia (envía/recibe) el documento XML, se transfiere un solo documento.
- Si el documento se debe firmar con Firma Electrónica Avanzada, la aplicación de firma tiene todos los elementos para el procesamiento.
- Agiliza la arquitectura de repositorio y el flujo de trabajo basados en el documento XML.

Desventajas:

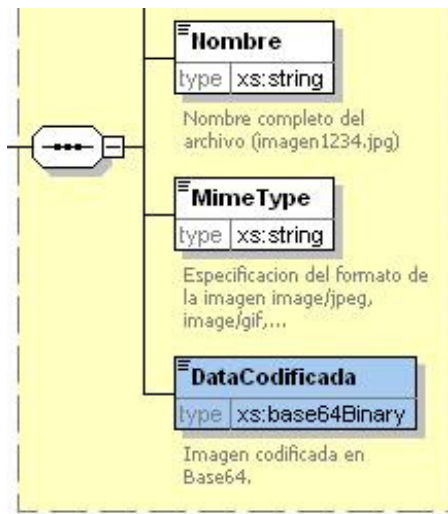
- La incorporación de archivos binarios puede aumentar el tamaño del archivo XML hasta un punto que dificulta el procesamiento o intercambio.

Modelo técnico

XML permite la inclusión de archivos binarios a través del tipo de datos: `xs:base64Binary` definido por la W3C (<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#base64Binary>), este tipo de dato representa al archivo binario codificado en base64.

Se codifica en base64 para tener una cadena de caracteres donde ningún carácter sea prohibido por el lenguaje XML como son: `<` &

Por ejemplo de estructura esquema para una imagen corresponde a:



En una instancia XML:

```
<Adjuntos>
  <imagen>
    <Nro>1</Nro>
    <Nombre>Resolucion-02478-01.jpg</Nombre>
    <MimeType>image/jpeg</MimeType>

    <DataCodificada>/9j/4AAQSkZJRgABAAEAYADIAAD//gAfTEVBRCBUZWNobm9sb2dpZX
    MgSW5jLiBWMS4wMQD/2wCEAAgFBgcGBQgHBgcJCAgJDBQNDA sLDBgREg4UHRkeHhwZHB
    sgJC4nIClrIhscKDYoKy8xMzQzHyY4PDgyPC4yMzEBCAkJDAoMFw0NFzEhHCExMTExMT
    ExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMf/EAaIAAAEFAQE
    BAQEBAAAAAAAAAAABA gMEBQYHCAkKcwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcl
    CQoLEAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVl
    R8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSINUVVZlZXWFlaY2RlZmdoaWpzdH
    V2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX
    2Nna4eLj5OXm5+jp6vHy8/T19vf4+foRAAIBAgQEAwQHBQQEAAECdwABA gMRBAUhmQYSQVE
    HYXETljkBCBRckaGxwQ .....
    .....</DataCodificada>
  </imagen>
</Adjuntos>
```

2.1.2. Apuntar o referenciar a la dirección del adjunto

Apuntar o referenciar un documento adjunto, significa indicar la URL en la cual se encuentra dicho documento. La forma correcta de referenciar los documentos en un XML, puede ser de las siguientes formas:

- **href**

```
<documentoXML>  
    <boleta href="http://aem.gov.cl/repo/doc123.jpg"/>  
    <carta href="http://aem.gov.cl/repo/doc987.doc"/>  
</documentoXML>
```

- **XLink**

```
<documentoXML>  
    <boleta xmlns:xlink="http://www.w3.org/XML/XLink/0.9" xlink:type="simple"  
    xlink:href="http://aem.gov.cl/repo/doc123.jpg"/>  
    <carta xmlns:xlink="http://www.w3.org/XML/XLink/0.9" xlink:type="simple"  
    xlink:href="http://aem.gov.cl/repo/doc987.doc"/>  
</documentoXML>
```

Ventajas:

- Permite no cargar la instancia XML en término de tamaño.
- Permite tener extensibilidad en caso de lista (el documento XML puede apuntar tanto a 10 documentos como a 10.000)

Desventajas:

- El documento XML no contiene todos los datos modelados.
- Cuando se intercambia (envía/recibe) el documento XML, se transfiere un documento. Luego, el receptor debe pedir los documentos relacionados según necesidades.
- Si el documento se firma con Firma Electrónica Avanzada, la aplicación de firma debe pedir todos los documentos antes de generar una representación del objeto a firmar. Esto en el caso de que los adjuntos sean parte de los datos que se requiere firmar.
- Requiere de un repositorio permanentemente disponible y de simple acceso, especialmente cuando se intercambian estos documentos XML.
- Requiere una gestión de cambios de los adjuntos y permanencia de todas las versiones en repositorio.

- Si un documento en formato XML firmado con Firma Electrónica Avanzada apunta a adjuntos en un repositorio, al momento de verificar la firma electrónica 4 años después de su creación, los adjuntos a los cuales apunta el XML deben ser los originales no modificados.

3. Visualización de un documento XML con adjunto

El método básico de visualización de instancias XML es XHTML mediante una transformación XSL (T). Este método supone la visualización en un navegador de Internet como son Microsoft Internet Explorer, Mozilla Firefox, Safari, Google Chrome, etc.

Los navegadores no tienen métodos de representación para todos los tipos de archivos binarios, pero en XHTML se puede incluir imágenes dentro del flujo de las páginas o hacer referencia a un archivo binario.

Algunos programas instalan plugins en los navegadores para obtener una representación del tipo de archivo que representan (plugin flash, plugin pdf, plugin MS word, etc.) y así lograr una “integración” mas cercana con el navegador y entregar una mejor experiencia al usuario.

Estas limitaciones son válidas también para las representaciones destinadas a la captura de los datos en formato distinto a XHTML.

El modelo de procesamiento de XSLT consiste en la lectura de uno o múltiples nodos XML que se transforman según reglas generando un árbol XHTML de destino, por lo tanto al entregar a un navegador una instancia XML y un XSLT no es suficiente para procesar la extracción de archivos binarios del XML, el cual decidirá si es capaz de presentarlo o si debe llamar un plugin o si lo debe guardar en disco.

Existen dos alternativas para abordar este tipo de proceso:

- El uso de “Data URI scheme”
- Desarrollar un programa a medida.

3.1. Data URI scheme

Corresponde a un esquema URI que permite la inclusión de pequeños elementos de datos en línea, como si fueran referenciados hacia una fuente externa. Ver:

- <http://www.ietf.org/rfc/rfc2397.txt>
- http://en.wikipedia.org/wiki/Data_URI_scheme

Restricciones:

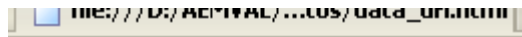
- Sirve sólo para imágenes en formato gif, jpg y png (restricción entregada por los navegadores)
- Todos los navegadores de Internet procesan Data URI schema, salvo Microsoft Internet Explorer.

Ejemplo:

```
<html><body><h2>ejemplo de uso data uri</h2>

<br/>ref: http://en.wikipedia.org/wiki/Data\_URI\_scheme#Examples</body></html>
```

Visualización en Firefox:



ejemplo de uso data uri

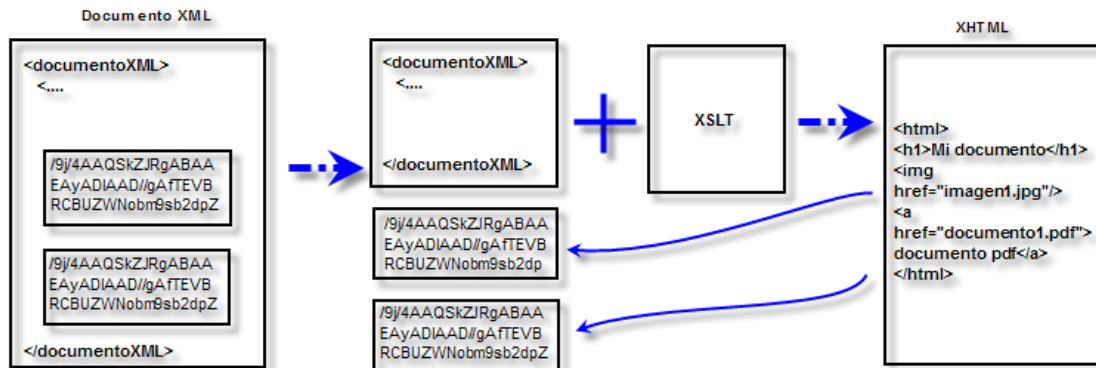


3.2. Desarrollo a medida

Debido a que los procesadores XSLT no están diseñados para crear visualización de elementos de tipo base64binary, es necesario agregar un motor adicional para estos efectos.

Por ejemplo, se puede ocupar un lenguaje de programación como java, php, .net, etc. para construir un programa o aplicación que separe los adjuntos de los documentos XML antes de la creación de la visualización.

Gráficamente se puede entender de la siguiente forma:



4. Elección de formatos de adjuntos

Según el uso del documento XML, se debe evaluar el formato de los adjuntos a incluir.

Si el documento XML está destinado a almacenar datos para su presentación, sin realizar cambios posteriores en los datos, como los son los actos administrativos, podría ser mejor incluir adjunto en formato imagen en vez de documentos Word, Excel, etc..

Para la evaluación, se recomienda considerar los métodos de presentación.

- Si los documentos van a ser publicados en XHTML, mediante una aplicación que ocupa XSLT, se puede lograr la visualización completa con imágenes.
- Si los archivos adjuntos tienen distintos formatos, la página XHTML tendrá vínculos a los adjuntos que se abrirán con sus programas respectivos, con el supuesto que el navegador tenga estos programas.

5. Envío de archivos en los servicios Web

En el proceso de intercambio de información entre instituciones, se debe resolver el problema de transferir un expediente de documentación asociado a un proceso de negocio. Este intercambio de información actualmente se realiza a través de Servicios Web con estructura SOAP.

La estructura SOAP es un documento XML, donde se pueden incluir otro documento XML como parte del SOAP o como archivo binario.

En un mensaje SOAP, se ocupa la misma técnica de incorporación del archivo binario en formato Base64binary como se indicó anteriormente.

Este método es ocupado por los servidores de correo cuando transfieren los archivos adjuntos de un correo, dicha transformación se hace a nivel del servidor y es transparente al nivel del usuario.

Al nivel de la interfaz del servicio Web, se pueden especificar los métodos de envío que optimizan el mensaje cuando contiene archivos.

5.1. Estándares

Los estándares de optimización de transferencia de archivos binarios han ido evolucionando para incorporar compatibilidad con los estándares de Web Services WS-*

El estándar más reciente es MTOM: Message Transfer Optimization Mecanism (<http://www.w3.org/TR/soap12-mtom/>)

Cronológicamente, los estándares de transferencia de archivos en servicios Web SOAP son:

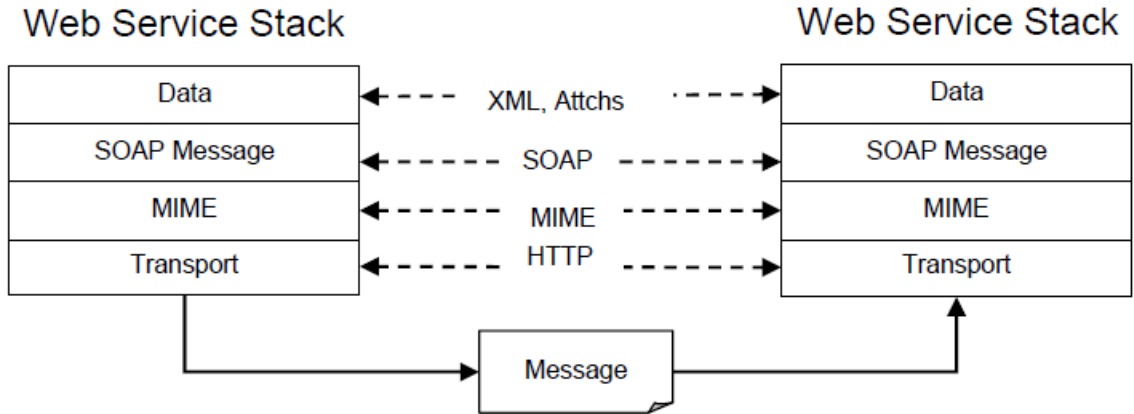
- SOAP Messages with Attachments (SwA)
- Direct Internet Message Encapsulation (DIME)
- WS-Attachments
- XML-binary Optimized Packaging (XOP)
- SOAP Message Transmission Optimization Mechanism (MTOM)

5.2. MTOM

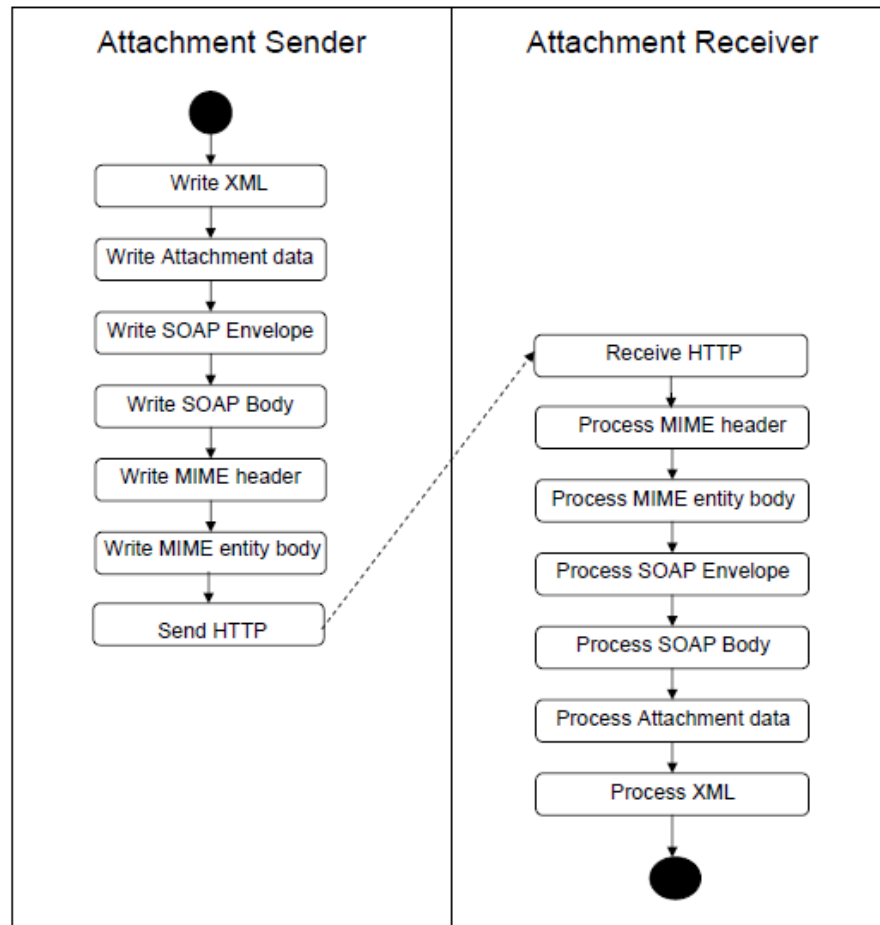
Se sugiere utilizar el estándar MTOM por que:

- MTOM optimiza la transferencia de binarios (de tipo base64Binary en el XML)
- MTOM es un mecanismo implementado al nivel del servicio Web (en la práctica, es el servidor que se encarga de implementar MTOM).
- MTOM opera al nivel de la transmisión del mensaje, eso es, entre el servicio web SOAP y el consumidor del servicio web.
- MTOM serializa el mensaje en formato “MIME Multipart/Related” utilizando XOP (<http://www.w3.org/TR/2005/REC-xop10-20050125/>)

5.2.1. Capas de un Servicio WEB utilizando MTOM



5.2.2. Flujo de la optimización MTOM



6. Consideraciones de archivos XML

La creación, almacenamiento, transferencia, presentación de documentos XML debe considerar factores tales como:

1. Los procesos de serialización de XML para distintos procesos como validación contra esquema, firma electrónica, indexación en repositorio, procesamiento usando XPath, XQuery, etc., son muy exigentes en términos de recursos (procesador, memoria) e influyen la planificación del tamaño de los archivos.
2. Se debe considerar los recursos del actor más “débil” en la cadena de uso del documento XML.
3. Una plataforma puede permitir crear, almacenar documentos XML de 100 MB, pero este XML difícilmente podrá ser entregado a un navegador junto con un XSLT para efectuar su transformación en formato XHTML.
4. Si se requiere el uso de Firma Electrónica, una aplicación de verificación de firma podría ver sus recursos sobrepasados por este tipo de tamaño.
5. En términos de transferencia de documentos vía Web Services SOAP, se debe considerar múltiples factores como:
 - Las plataformas de IT
 - El ancho de banda
 - Las capacidades operacionales de los receptores de los archivos (Ej.: Ministerio vs. Municipio, empresa multinacional vs. PYME, empresa vs. ciudadano)

Como base de comparación se puede tomar el intercambio de archivos a través de correo electrónico. Se deben responder las mismas preguntas:

- ¿Si creo un archivo de texto de 100 MB, podré enviarlo por correo?
- ¿Podrá la persona que lo recibe abrirlo en su computador?

Las respuestas dependen de la clarificación de los actores en la vida del documento. Es aceptable transferir planos de 1 GB de tamaño a un arquitecto, pero no necesariamente a la persona que compra la casa.

7. Referencias

- Uso de DIME : <http://msdn.microsoft.com/en-us/magazine/cc188797.aspx#S2>
- Migrar de DIME a MTOM: <http://msdn.microsoft.com/en-us/library/aa529283.aspx>
- Envío y recepción de archivos, grandes cantidad de datos en .NET:
<http://msdn.microsoft.com/en-us/library/aa528822.aspx>
- Web Services, Opaque Data, and the Attachments Problem:
<http://msdn.microsoft.com/en-us/library/ms996462.aspx>
- XML-binary Optimized Packaging : <http://www.w3.org/TR/2005/REC-xop10-20050125/>
- SOAP Message Transmission Optimization Mechanism:
<http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>
- Describing Media Content of Binary Data in XML: <http://www.w3.org/TR/xml-media-types/>
- XML Linking Language (XLink) Version 1.0: <http://www.w3.org/TR/xlink/>