

# Intercambio electrónico de información: el estándar XML



Jornades Catalanes d'Informació i Documentació



Col·legi Oficial  
de Bibliotecaris-  
Documentalistes  
de Catalunya



CometaTechnologies, S.L.  
Soluciones abiertas en tecnologías de la información

---

Alberto Otero García (alberto@cometatech.com)

- Introducción a XML
- Esquemas XML
- Transformación de XML
- APIs de trabajo con XML
- Servicios web
- Caso de estudio
- Conclusiones
- Referencias

# Introducción a XML

---

- El *eXtensible Markup Language*, **XML**, se ha convertido en la forma estándar de intercambio de datos más utilizada
- Este éxito viene dado no sólo por XML en sí, sino por las tecnologías que lo amplían, por ejemplo:
  - Esquemas: definición de la **estructura** de documentos XML
  - Transformación: **conversión** de documentos XML en otros XML
  - APIs: **acceso** estándar a documentos XML a través de software
  - Servicios web: descubrimiento e **invocación** de métodos remotos a través de XML

# Introducción a XML

---

- ¿Qué es XML?

XML es un lenguaje de marcas diseñado para describir datos, desarrollado por el W3C (*World Wide Web Consortium*)

- ¿Cuáles son las principales características de XML?

- Los documentos XML están en formato de texto
- Las marcas (*tags*) utilizadas en los documentos XML no están predefinidas
- Los documentos XML son diseñados para describir datos, y no para definir cómo estos se mostrarán

# Introducción a XML

---

- Documento XML

```
<?xml
  version="1.0"
  encoding="ISO-8859-15"?>

<noticia>
  <titulo>UMTS, probablemente la
    tecnología más esperada</titulo>
  <descripcion>UMTS es probablemente
    la tecnología más esperada en
    estos momentos, una de sus
    aplicaciones mas importantes es
    en los móviles de tercera
    generación.</descripcion>
  <pub hora="16:24">25/11/2004</pub>
  <autor>
    <nombre>Andrés</nombre>
    <apellido>Bou</apellido>
  </autor>
</noticia>
```

- Documento HTML

```
<html>
<body>
  <p>
    <b>Título:</b> UMTS, probablemente
    la tecnología más esperada <br>
    <b>Descripción:</b> UMTS es
    probablemente la tecnología más
    esperada en estos momentos, una
    de sus aplicaciones más
    importantes es en los móviles de
    tercera generación.
  </p>
  <p>
    Publicación: 16:24h., 25/11/2004
  <br>
    Autor: Andrés Bou
  </p>
</body>
</html>
```

# Introducción a XML

---

- ¿Qué ventajas conlleva el **almacenar la información** en XML, pero no cómo ésta será mostrada?
  - Permite **concentrarse** en una de las tareas, almacenamiento de datos o su presentación (HTML, PDF, SVG, etc.)
  - Facilita el intercambio de información entre diferentes sistemas, XML es un **estándar** (p. ej. a diferencia de los formatos utilizados por SGBDs)
  - Facilita el manejo de datos, formato texto **fácil de manipular** e independiente de plataforma
  - Facilita el intercambio de información relacionada con cualquier **área funcional** (finanzas, marketing, etc.)

# Introducción a XML

---

- ¿Qué características tiene un documento XML?
  - Primera línea de declaración (versión estándar XML y codificación de caracteres utilizada)
  - Marcas y marcas hijas
  - Las marcas pueden tener atributos
  - El contenido está dentro de marcas y atributos

- Ejemplo:

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<noticia>
  <titulo>UMTS, probablemente la tecnología más esperada</titulo>
  <pub hora="16:24">25/11/2004</pub>
  <autor>
    <nombre>Andrés</nombre>
    <apellido>Bou</apellido>
  </autor>
</noticia>
```

# Introducción a XML

---

- ¿Qué **reglas sintácticas** deberá cumplir un documento XML? (**bien formado**)
  - Primera línea de declaración y elemento raíz
  - Distinción entre mayúsculas y minúsculas
  - Marcas cerradas y anidadas correctamente
  - Marcas con nombres con letras, números y otros caracteres. No pueden empezar por un número, un signo de puntuación, etc.
  - Contenido de los atributos entrecomillado y sin " o ` que interfieran (&quot; y &apos;)
  - No utilizar dentro del contenido los caracteres < y & (&lt; y &amp;)
  - Comentarios acotados con <!-- y -->



# Introducción a XML

---

- ¿Cómo ayuda XML en el proceso de **estandarización** real o de facto?

Dada la potencia y **versatilidad** del estándar XML, en la actualidad cada vez más **vocabularios** diseñados con el fin de describir datos (para intercambiarlos, almacenarlos, transformarlos, etc.) se basan en la utilización de XML para definir su **sintaxis**

- Ejemplos:
  - eXtensible HyperText Markup Language (**XHTML**): el nuevo HTML
  - **DocBook**: descripción de documentos, en especial libros y artículos de carácter técnico

# Introducción a XML

---

- Ejemplos:
  - Scalable Vector Graphics (**SVG**): descripción de gráficos bidimensionales
  - RDF Site Summary (**RSS**): sindicación de contenido web (noticias, *blogs*, etc.), describiendo metadatos
  - Electronic Business eXtensible Markup Language (**ebXML**): intercambio entre empresas de ofertas, registros, etc. relacionados con sus procesos de negocio
  - Voice eXtensible Markup Language (**VoiceXML**): desarrollo de aplicaciones que utilizan tecnologías de reconocimiento y síntesis de voz
  - Lenguajes relacionados con las propias tecnologías XML: **XSL**, **XML Schema**

# Esquemas XML

---

- ¿Cómo se define **qué debe contener** un documento XML?
  - Para la mayoría de aplicaciones es necesario no sólo que los documentos XML estén bien formados, sino que cumplan ciertos **requisitos** definidos previamente (que se atengan a **vocabularios** XML)
  - Se considera que un documento XML es **válido** cuando los elementos que contiene son acordes a la definición que se ha hecho de ellos en lo que llamaremos el **esquema** XML (la definición de un vocabulario)
  - El esquema recoge las **marcas** y **atributos** incluidos en el vocabulario, la **estructura** que deben seguir éstos, y el formato que tendrá su **contenido**

# Esquemas XML

---

- ¿Qué **ventajas** aporta la utilización de esquemas XML?
  - Asegurar de forma sencilla que un documento XML cumple con unas **normas** establecidas previamente (usa un vocabulario y lo hace de forma correcta)
  - Definir y **compartir** descripciones de documentos XML. Los vocabularios públicos creados utilizando el estándar XML (p. ej., SVG, XSL, ebXML) disponen de sus respectivos esquemas
- ¿Cómo podemos **definir** un esquema?

Existen múltiples **lenguajes** de definición de esquemas, por ejemplo: DTD, XML Schema, Relax NG, Schematron

# Esquemas XML

---

- Ejemplo (XML Schema):

```
<?xml version="1.0" encoding="ISO-8859-15"?>

<xsd:schema
  xmlns="http://cometatech.com/noticia"
  targetNamespace="http://cometatech.com/noticia"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="autor" type="autorTipo" />

  <xsd:complexType name="autorTipo">
    <xsd:sequence>
      <xsd:element name="nombre" type="xsd:string" />
      <xsd:element name="apellido" type="xsd:string" maxOccurs="2" />
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

# Transformación de XML

---

- ¿Cómo podemos **aprovechar** las aplicaciones disponibles para los diferentes vocabularios XML?  
La **transformación** de un documento XML en otro implicará la traducción entre vocabularios, y por tanto poder sacar partido a las ventajas de éstos
- Ejemplo: transformación de un documento XML propio, "noticias", en
  - Un documento XML que sigue un vocabulario estándar como RSS para sindicación de contenido
  - Un documento XHTML que podemos mostrar en un navegador web
  - Un documento VoiceXML que nos permite oír el contenido mediante una aplicación telefónica

# Transformación de XML

---

- ¿Cómo podemos **transformar** documentos XML?  
A través del lenguaje eXtensible Stylesheet Language (**XSL**) podremos transformar marcas, atributos y contenido de los documentos XML
- ¿En qué consiste la **especificación** XSL?  
XSL define tres lenguajes:
  - **XPath**: permite hacer referencia a partes de documentos XML
  - XSL Transformation (**XSLT**): permite transformar un documento XML en otro documento XML
  - XSL Formatting Objects (**XSL-FO**): permite describir cómo se mostrará la información de un documento XML en un medio con paginación

# Transformación de XML

---

- ¿Para qué sirve **XSLT**?

El lenguaje XSLT permite **expresar** cómo un documento XML se debe transformar en otro documento XML

- ¿Qué **permite** hacer XSLT?
  - Transformar unos elementos (marcas, atributos, contenido) en otros
  - Añadir nuevos elementos al documento XML resultado de la transformación
  - Eliminar elementos del documento XML original, que no aparecerán en el resultado de la transformación
  - Reordenar elementos del documento XML original



# Transformación de XML

---

- ¿Cómo se **usa** XSLT?

La transformación de un documento XML con XSLT se hace estableciendo **correspondencias** entre sus elementos y el resultado (los elementos producto de la transformación)

- **Ejemplo:**

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="noticia">
    <titular>
      <xsl:value-of select="titulo"/>, por
      <xsl:value-of select="autor/nombre" />
      <xsl:value-of select="autor/apellido"/>
    </titular>
  </xsl:template>
</xsl:stylesheet>
```

# Transformación de XML

---

- **Ejemplo:**

```
<?xml version="1.0" encoding="ISO-8859-15"?>
```

```
<titular>
```

```
    UMTS, probablemente la tecnología más esperada, por Andrés Bou
```

```
</titular>
```

- ¿Cómo se hará **referencia** a los elementos de un documento XML a transformar mediante XSLT?

Las expresiones **XPath** permitirán identificar elementos concretos, conjuntos, valores, etc.

- ¿Cómo se podrá transformar un documento XML para **presentar** su contenido?

Mediante **XSL-FO** generaremos un documento XML en el que se exprese cómo éste se mostrará (márgenes, encabezados, tipos de letra, etc.)

# APIs de trabajo con XML

---

- ¿Cómo podremos **automatizar** el trabajo con documentos XML?

El trabajo con documentos XML se hará a través de **software** diseñado para esta labor

Existen multitud de librerías de programación que ofrecen **APIs** (Application Programming Interface) para realizar operaciones sobre documentos XML, como **crearlos**, **consultarlos**, **modificarlos**, testar si son **válidos** y están **bien formados**, **transformarlos**, etc.

En la actualidad todo **lenguaje de programación** que se precie dispone de un conjunto de librerías para trabajar con XML

# APIs de trabajo con XML

---

- Ejemplos:
  - LibXML: conjunto de utilidades y librerías para utilizar desde programas en lenguaje **C**, del proyecto Gnome (licencia MIT)
  - Xerces, Xalan: conjunto de utilidades y librerías para utilizar desde programas en lenguaje **C++** o **Java**, de la Apache Software Foundation (licencia Apache)
  - Expat: conjunto de librerías programadas en C para utilizar desde programas lenguaje **PHP**, por ejemplo (licencia MIT)
  - XML::Parser: extensión de la interfaz de Expat para utilizar desde programas en lenguaje **Perl** (licencia MIT)

# APIs de trabajo con XML

---

- ¿Qué tipo de **software** nos permitirá trabajar con documentos XML?

Uno de los principales elementos que todo software de trabajo con XML debe tener es un **parser**, es decir, el encargado de **leer** los documentos XML (teniendo en cuenta su **formato**, **características**, **elementos** que pueden aparecer, etc.) y permitir **acceder** a los datos que contienen

- ¿Cómo podremos **utilizar** un *parser* ya existente, con el fin de **ahorrar** recursos?

Mediante interfaces de programación estándares, como **DOM** (*Document Object Model*) y **SAX** (*Simple API for XML*)

# APIs de trabajo con XML

---

- ¿Qué es **DOM**?

Es una interfaz de programación que permite acceder y modificar los datos contenidos en documentos XML mediante un modelo de **objetos jerarquizado**

DOM proporciona un **árbol** de objetos correspondientes a los diferentes **elementos** de los documentos XML con que se trabaja

- ¿**Porqué** DOM?

El planteamiento concuerda con la **estructura** jerárquica en la que está organizada la información en XML (elemento raíz, hijos, etc.)

# APIs de trabajo con XML

---

- ¿Qué es **SAX**?

Es una interfaz de programación que permite acceder y modificar los datos contenidos en documentos XML mediante un modelo basado en **eventos**

El *parser* que lea el documento XML generará ciertos **eventos** conforme vaya encontrando los **elementos** que lo forman

- ¿Porqué SAX?

Cada uno de los **eventos** tendrá asociado un **método** que se ejecutará cuando el mencionado evento ocurra, no existirá una representación en **memoria** de la jerarquía del documento XML

# Servicios web

---

- ¿Cómo ayuda XML al **intercambio** de datos entre **aplicaciones**?

Los **servicios web** son una colección de protocolos que usan **XML** con el fin de posibilitar el **intercambio** de información entre aplicaciones

Una aplicación **ofrece** a otras un servicio web, una **función** o método que se puede **invocar** remotamente a través de redes de ordenadores

- ¿Qué estándares engloban los servicios web?
  - Simple Object Access Protocol (**SOAP**)
  - Web Services Description Language (**WSDL**)
  - Universal Description, Discovery and Integration (**UDDI**)



# Servicios web

---

- ¿Qué es **SOAP**?

Es un **vocabulario** estándar (W3C) que permite **intercambiar** datos entre aplicaciones

Posibilita la invocación de métodos remotos, pasando los **parámetros** que sean necesarios, y recogiendo los **resultados** que éstos devuelvan

- ¿Qué **ventajas** aporta SOAP?

Al basar la comunicación entre aplicaciones en XML permite que ésta sea **independiente** de la **plataforma** y **lenguaje de programación** utilizados (la mayoría disponen de librerías de abstracción)

Se apoya en protocolos de comunicación de **Internet** (p. ej. HTTP) para el envío de mensajes

# Servicios web

---

- **Ejemplo:**

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://cometatech.com/noticias">
    <m:GetAutor>
      <m:titulo>UMTS, probablemente la tecnología más esperada</m:titulo>
    </m:GetAutor>
  </soap:Body>
</soap:Envelope>
```

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://cometatech.com/noticias">
    <m:GetAutorResponse><m:autor>Andrés Bou</m:autor></m:GetAutorResponse>
  </soap:Body>
</soap:Envelope>
```

# Servicios web

---

- ¿Qué es **WSDL**?

Es un **vocabulario** estándar (W3C) que permite **describir servicios web**, expresando cuáles hay **disponibles**, en qué **servidor** residen, qué **parámetros** reciben y qué **devuelven**

- ¿Cómo se **relacionan** WSDL y SOAP?

Los documentos **WSDL** (que son en sí mismos documentos XML) permiten obtener las **descripciones** de los **servicios web** que posteriormente se **invocarán** a través de **SOAP**

A partir del documento WSDL que describa los servicios web se podrán realizar las llamadas SOAP, conociendo los datos a enviar y recibir

- **Ejemplo:**

```
<?xml version="1.0" encoding="UTF-8"?>
[...
  <wsdl:message name="GetAutorRequest">
    <wsdl:part name="titulo" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="GetAutorResponse">
    <wsdl:part name="autor" type="xsd:string"/>
  </wsdl:message>
[...
  <wsdl:operation name="GetAutor">
    <wsdl:input name="GetAutorRequest">
      <wsdlsoap:body namespace="http://cometatech.com/noticias" />
    </wsdl:input>
    <wsdl:output name="GetAutorResponse">
      <wsdlsoap:body namespace="http://cometatech.com/noticias" />
    </wsdl:output>
  </wsdl:operation>
[...]
```

# Servicios web

---

- ¿Qué es **UDDI**?

Es un **servicio de directorio** (definido por un consorcio de empresas como IBM, HP o MS) mediante el cual las organizaciones pueden **registrar** y **buscar** servicios web

Mediante UDDI las organizaciones acceden a registros en los cuales pueden **hacer pública información** acerca de sí mismas y de los servicios web que ofrecen (para **qué** sirven, **cómo** invocarlos, etc.)

- ¿Qué **ventajas** aporta UDDI?

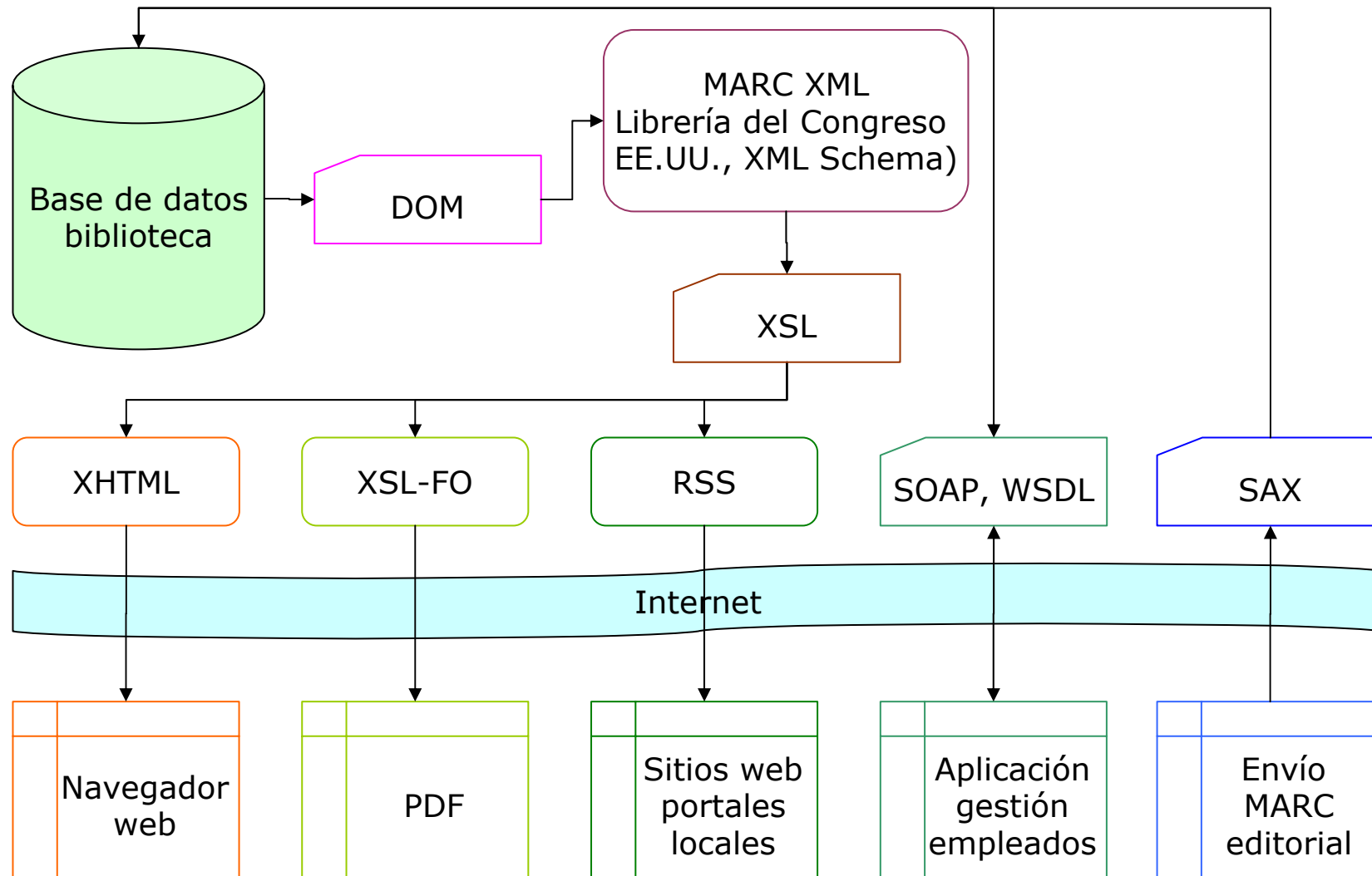
Hace posible la formación de un **mercado** de servicios web global

# Caso de estudio

---

- **Biblioteca** que tiene una base de datos en un formato **propietario**, quiere:
  - Que se pueda **consultar** su **catálogo** a través de internet, mediante un navegador **web**, y descargarlo en **PDFs**
  - Que los sitios web de los **portales** locales de su área geográfica dispongan de información sobre los **últimos títulos** incorporados al catálogo
  - Que sus **empleados** puedan trabajar con la **aplicación de gestión** del catálogo desde su casa
  - Que se puedan **incorporar** al catálogo los datos de los libros adquiridos de forma automática, a partir de **información** suministrada por los **proveedores**

# Caso de estudio



# Conclusiones

---

- XML es el lenguaje a escoger para el **intercambio de información** en cualquier área funcional, ya que mediante éste, o las tecnologías asociadas a éste, podemos:
  - Definir y comunicar claramente el **formato** de la información con la que se trabaja, lo que da lugar a que se utilice en muchos estándares actuales
  - Realizar **transformaciones** entre diferentes formatos y aprovechar las ventajas de cada uno
  - Utilizar la mayoría de **lenguajes de programación** modernos, en los que existen multitud de librerías que suponen un ahorro de costes
  - Desarrollar **aplicaciones distribuidas** apoyadas en la utilización intensiva de redes de ordenadores



# Referencias

---

- [W3C](http://www.w3c.org/) (<http://www.w3c.org/>)
- [W3C Schools](http://www.w3schools.com/) (<http://www.w3schools.com/>)
- [ebXML](http://www.ebxml.org/) (<http://www.ebxml.org/>)
- [VoiceXML](http://www.voicexml.org/) (<http://www.voicexml.org/>)
- [Gnome XML](http://www.xmlsoft.org/) (<http://www.xmlsoft.org/>)
- [Expat](http://expat.sourceforge.net/) (<http://expat.sourceforge.net/>)
- [Apache XML](http://xml.apache.org) (<http://xml.apache.org>)
- [SAX](http://www.saxproject.org/) (<http://www.saxproject.org/>)
- [Apache web services](http://ws.apache.org/) (<http://ws.apache.org/>)
- [SOAP::Lite](http://www.soaplite.com/) (<http://www.soaplite.com/>)
- [UDDI](http://www.uddi.org/) (<http://www.uddi.org/>)

# Intercambio electrónico de información: el estándar XML



Jornades Catalanes d'Informació i Documentació



Col·legi Oficial  
de Bibliotecaris-  
Documentalistes  
de Catalunya



CometaTechnologies, S.L.  
Soluciones abiertas en tecnologías de la información

---

Alberto Otero García (alberto@cometatech.com)