

Bloque 5 - XML



Como recordatorio traemos a este tema las diapositivas del Bloque 1 en el que definimos XML, los principios de diseño, la estructura y la sintaxis básica de los documentos XML y definición de XML bien formados.

XML: Definición

- Al crecer el uso de la web, crecieron las presiones para ampliar HTML, pero W3C decidió que la solución no era ampliar HTML, sino crear unas **reglas para que cualquiera pudiera crear lenguajes de marcas adecuados a sus necesidades**, pero manteniendo unas estructuras y sintaxis comunes que permitieran **compatibilizarlos y tratarlos con las mismas herramientas**. Ese conjunto de reglas y tecnologías es **XML**.
- XML (Extensible Markup Language) deriva de SGML (ISO 8879) → **simple y flexible**. Diseñado para la **publicación a gran escala y el intercambio de datos en la Web**.
- [Guía: XML en 10 puntos](#)
- **XML** no es un lenguaje de marcado, **es un meta-lenguaje para “definir” lenguajes de marcado**. → XML proporciona una serie de reglas para que **cualquiera pueda definir su propio conjunto de etiquetas y atributos y pueda definir las relaciones que existen entre esas etiquetas**.
- La especificación ha sido desarrollada por [W3C](#) (comenzó en 1996) → [XML1.0](#) y [XML1.1](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<países>
  <pais>
    <nombre>España</nombre>
    <capital>Madrid</capital>
  </pais>
  <pais>
    <nombre>Francia</nombre>
    <capital>Paris</capital>
  </pais>
</países>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<employees>
  <employee>
    <id>1</id>
    <firstName>Amr</firstName>
    <lastName>Khan</lastName>
    <photo>http://imgd.s3.amazonaws.com/dhoom-3-fan-made-poster.jpg</photo>
  </employee>
  <employee>
    <id>2</id>
    <firstName>Anna</firstName>
    <lastName>Smith</lastName>
    <photo>http://imgd.s3.amazonaws.com/nH436FvL.jpg</photo>
  </employee>
  <employee>
    <id>3</id>
    <firstName>Peter</firstName>
    <lastName>Jones</lastName>
    <photo>http://imgd.s3.amazonaws.com/7VVKFPpQ.jpg</photo>
  </employee>
</employees>
```

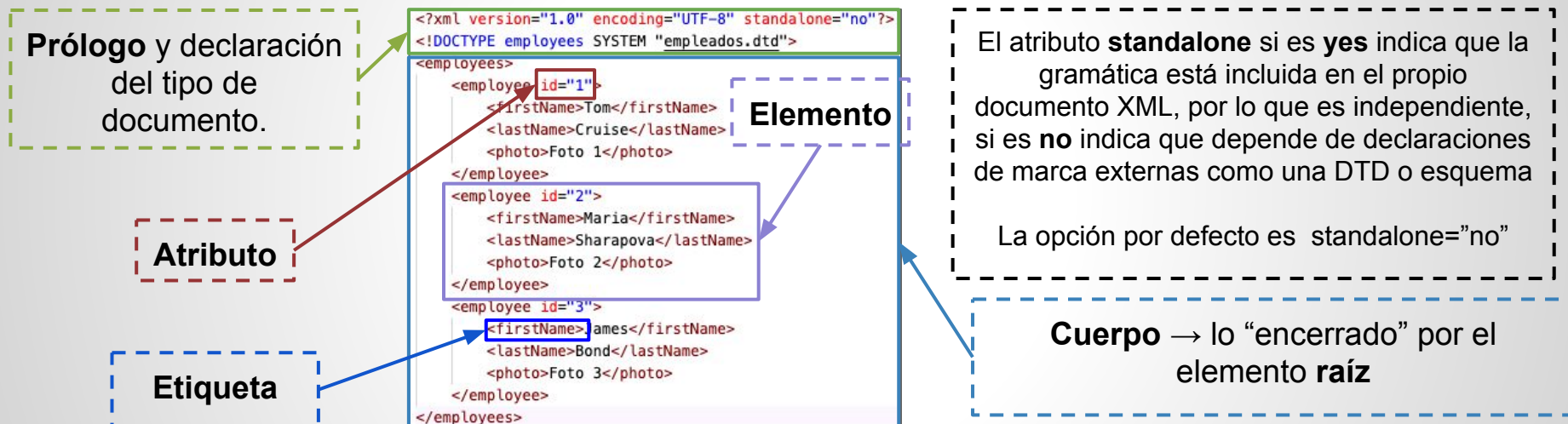
XML: principios de diseño



- Directamente utilizable en Internet (*XML shall be straightforwardly usable over the Internet*).
- Soportar una amplia variedad de aplicaciones (*XML shall support a wide variety of applications*).
- Compatible con SGML (*XML shall be compatible with SGML*).
- Fácil de utilizar por los programadores (*It shall be easy to write programs which process XML documents*).
- El número de características opcionales en XML debe mantenerse al mínimo absoluto , idealmente cero (*The number of optional features in XML is to be kept to the absolute minimum, ideally zero*).
- **Los documentos XML deben ser legible por las personas y razonablemente claro** (*XML documents should be human-legible and reasonably clear*).
- Diseño rápido (*The XML design should be prepared quickly*).
- **El diseño tiene que ser formal** y conciso (*The design of XML shall be formal and concise*).
- Fácil de crear documentos XML (*XML documents shall be easy to create*).

XML: Sintaxis básica

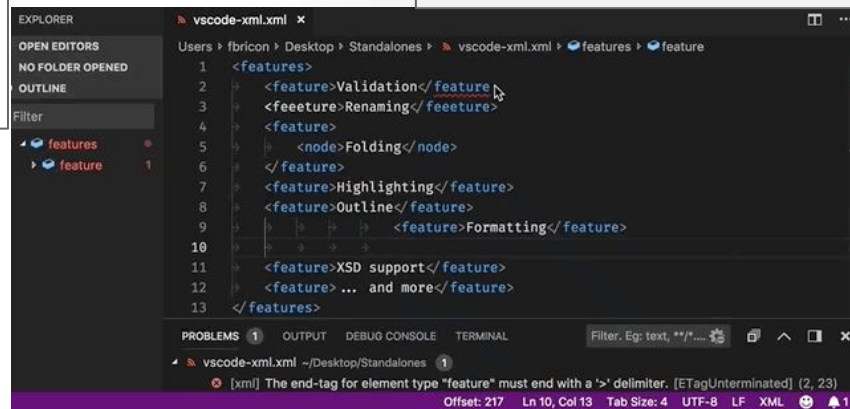
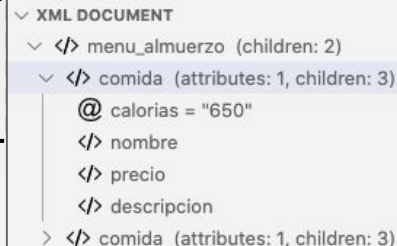
- **DTD** → Definición del tipo de documento: descripción en **lenguaje comprensible por las máquinas** de la estructura, elementos y atributos correctos del documento XML.



Aunque no es obligatorio, un documento XML debe comenzar con la declaración XML, que es una instrucción de procesamiento que declara al documento como un documento XML. Esta declaración indica la versión de XML, el juego de caracteres empleado y si tiene asociado un DTD o esquema externo

Si no se incluye se asume versión 1, utf-8 y standalone no.

- Los ficheros tienen que tener **extensión xml**
- Al pulsar **Ctrol + Space** podremos autocompletar el código



Extensión para facilitar el trabajo con XML

Tiene estar Java (JDK o JRE 8 o más reciente) instalado y configurado:

Archivo → Preferencias → buscar
java.home, abrir "Edit in settings.json"
y poner la ruta en la variable xml.java.home



```
"xml.java.home": "C:\\Program Files\\Java\\jdk1.8.0_121"
```


XML bien formados (I): Definición

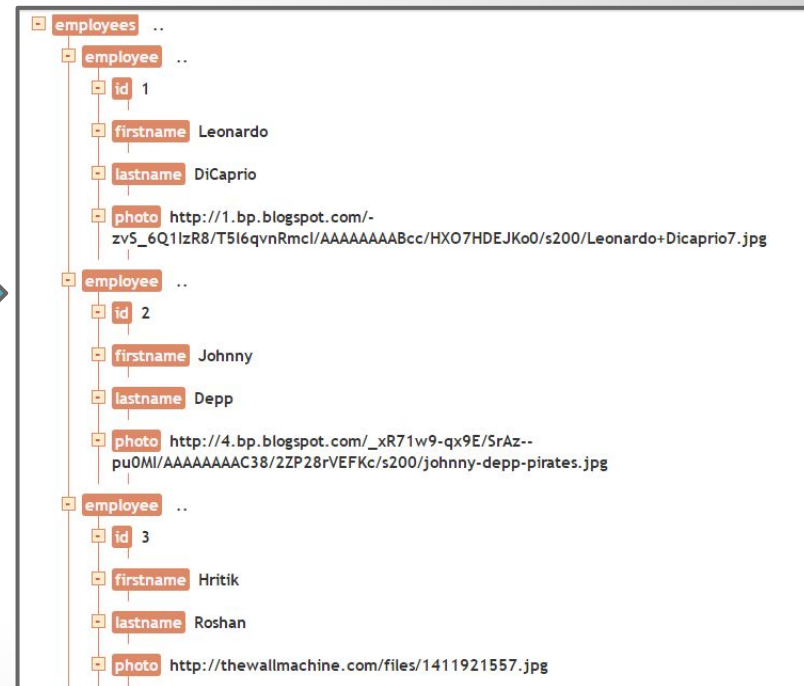
- Un documento se considera **bien formado (well formed)** si cumple la sintaxis especificada en la recomendación XML [XML1.0](#) o [XML1.1](#), en resumen:
 - Todos los elementos tienen que tener **etiqueta de apertura y cierre** → `<Modulo>LMSGI</Modulo>`
 - Los **elementos vacíos**, aquellos que no encierran texto, **también tienen que estar cerrados** → `
` o `
</br>`
 - Es **sensible a mayúsculas/minúsculas** (case sensitive) → `<Modulo>LMSGI</modulo>` → error de sintaxis
 - **Solo hay un elemento raíz**
 - Los elementos tienen que estar **correctamente anidados** → `<Modulo><Nombre>LMSGI<Centro></Nombre>IES Valle del Jerte</Centro></Modulo>` → error de sintaxis
 - Los **valores de los atributos** tienen que estar entre comillas simples o dobles → `<Empleado id="1"><Nombre>Manuel</Nombre></Empleado>`
 - **Pueden analizarse correctamente por cualquier parser (analizador sintáctico) que cumpla con la especificación XML.**
 - En caso de incluirse, el prólogo y la declaración del tipo de documento siguen la sintaxis correcta.

Si un documento XML no está bien formado,
no es un documento XML.

Un documento puede estar bien formado
pero no ser válido, y para ser válido tiene
que estar bien formado.

XML bien formados (II): Ejemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<employees>
  <employee>
    <id>1</id>
    <firstName>Leonardo</firstName>
    <lastName>DiCaprio</lastName>
    <photo>http://1.bp.blogspot.com/-zvS_6Q1IzR8/T5l6qvnRmcl/AAAAAAAABcc/HX07HDEJKo0/s200/Leonardo+Dicaprio7.jpg</photo>
  </employee>
  <employee>
    <id>2</id>
    <firstName>Johnny</firstName>
    <lastName>Depp</lastName>
    <photo>http://4.bp.blogspot.com/_xR7lw9-qx9E/SrAz--pu0MI/AAAAAAAAC38/2ZP28rVEFKc/s200/johnny-depp-pirates.jpg</photo>
  </employee>
  <employee>
    <id>3</id>
    <firstName>Hritik</firstName>
    <lastName>Roshan</lastName>
    <photo>http://thewallmachine.com/files/1411921557.jpg</photo>
  </employee>
</employees>
```



XML bien formados (y III): VSCode

Conforme escribimos muestra los errores en el propio documento, en algunos casos muestra sugerencias para poder solucionarlo.

```
<?xml version="1.0" encoding="UTF-8"?>
<menu_almuerzo>
  <comida calorías="650">
    <nombre>Waffles</nombre>
    <precio>2.00</Precio>
  </comida>
  <comida calorías="1500">
    <nombre>Hamburguesa</nombre>
    <precio>5.00
    <descripcion>La hamburguesa mas común de McDonalds</descripcion>
  </comida>
</menu_almuerzo>
```

Replace with 'precio' closing tag

prueba.xml 4 de 5 problemas

Expected end tag name to match start tag's name ('precio').

13 </menu_almuerzo>

Ctrl+Shift+M

Show Problems panel

F8

Go to next error or warning

Shift+F8

Go to previous error or warning

PROBLEMAS 5 SALIDA TERMINAL

prueba.xml 5

Missing element's end tag ('</menu_almuerzo>'). [2, 2]

Expected end tag name to match start tag's name ('precio'). [5, 17]

The element type "precio" must be terminated by the matching end-tag "</precio>". xml(ETagRequired) [5, 17]

Expected end tag name to match start tag's name ('precio'). [12, 7]

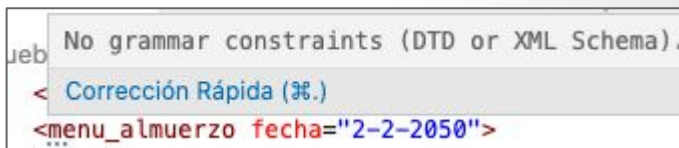
Expected end tag name to match start tag's name ('comida'). [13, 3]

Desde el menú “Ver” →
“Problemas”(Ctrl
podemos consultar todos
los errores

XML Válidos

- Un documento **es válido** si:
 - Está **bien formado**
 - Tiene **asociado** una declaración de tipo de documento con la **gramática**
 - Todos los elementos y atributos **cumplen la gramática** asociada
 - No hay elementos ni atributos que no estén en la gramática

En el elemento raíz muestra un error cuando no tiene asociado una gramática



Si tiene una gramática asociada VSCode muestra los errores en el código y también en "PROBLEMAS"

PROBLEMAS 1 SALIDA TERMINAL CONSOLA DE DEPURACIÓN

prueba.xml 1

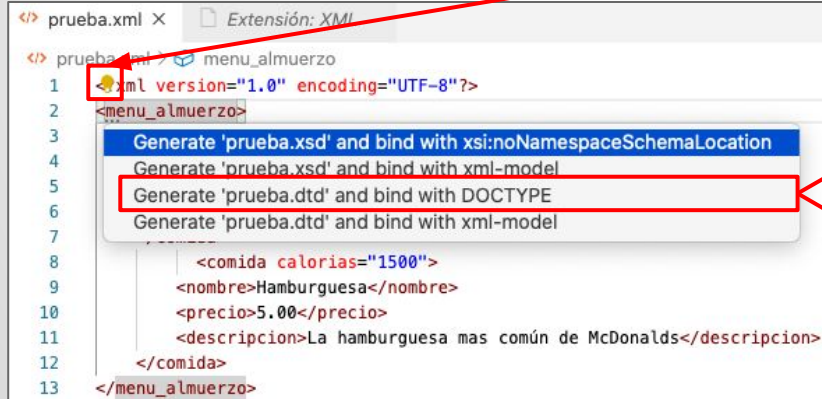
Attribute "fecha" must be declared for element type "menu_almuerzo". xml(MSG_ATTRIBUTE_NOT_DECLARED) [3, 16]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE menu_almuerzo SYSTEM "prueba1.dtd">
<menu_almuerzo fecha="2-2-2050">
  <comida calorias="650">
    <nombre>Waffles</nombre>
    <precio>2.00</precio>
    <descripcion>Waffles baratos de McDonalds</descripcion>
  </comida>
  <comida calorias="1500">
    <nombre>Hamburguesa</nombre>
    <precio>5.00</precio>
    <descripcion>La hamburguesa mas común de McDonalds</descripcion>
  </comida>
</menu_almuerzo>
```

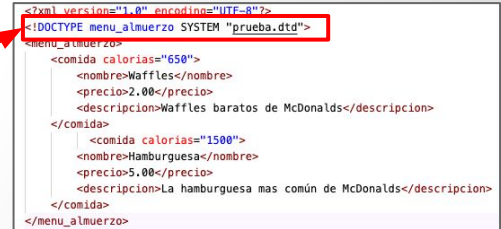
Generar DTD de XML en VSCode

- Desde VSCode podemos **generar automáticamente un DTD** partiendo de un documento **XML bien formado**
- Para **asociar el DTD** al documento podemos consultar el DTD generado, incluye un comentario (líneas con <!-- comentario -->) con la declaración del tipo de documento **<!DOCTYPE**

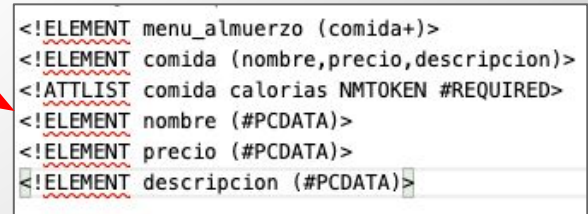
Al poner el ratón sobre el elemento raíz y pulsar en la “bombilla” nos dará las opciones para crear el DTD o XSD y asociarlo al documento XML



```
<?xml version="1.0" encoding="UTF-8"?>
<menu_almuerzo>
  <comida calorías="1500">
    <nombre>Hamburguesa</nombre>
    <precio>5.00</precio>
    <descripcion>La hamburguesa mas común de McDonalds</descripcion>
  </comida>
</menu_almuerzo>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE menu_almuerzo SYSTEM "prueba.dtd">
<menu_almuerzo>
  <comida calorías="650">
    <nombre>Waffles</nombre>
    <precio>2.00</precio>
    <descripcion>Waffles baratos de McDonalds</descripcion>
  </comida>
  <comida calorías="1500">
    <nombre>Hamburguesa</nombre>
    <precio>5.00</precio>
    <descripcion>La hamburguesa mas común de McDonalds</descripcion>
  </comida>
</menu_almuerzo>
```



```
<!ELEMENT menu_almuerzo (comida+)>
<!ELEMENT comida (nombre,precio,descripcion)>
<!ATTLIST comida calorías NMTOKEN #REQUIRED>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT descripcion (#PCDATA)>
```

XML: espacios de nombre (namespaces) (I)

- Un **espacio de nombres XML** es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. Un archivo XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual, la homonimia. Los nombres de elementos dentro de cada espacio de nombres deben ser únicos (fuente [wiki](#)) → no conviene inventar términos “raros” para evitarlos, de forma que se pueda hacer referencia **sin conflicto**
- Un **espacio de nombres XML (namespace)** es un conjunto de nombres, identificados por una referencia **URI** (identificador de recursos uniforme, cadena de caracteres que identifica los recursos de una red de forma unívoca [\[RFC2396\]](#)), que se utilizan en documentos XML.
- Especificación** → [namespace para XML 1.0](#) y [namespace para XML 1.1](#)
- Más información http://www.w3schools.com/xml/xml_namespaces.asp y

<https://diego.com.es/xml-principios-basicos>

Se llaman table pero se refieren a objetos distintos →
Al unirlos generarán conflicto

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```



```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

Utilizando namespace (xmlns) se
evita el conflicto

```
<root
  xmlns:h="http://www.w3.org/TR/html4/"
  xmlns:f="http://www.w3schools.com/furniture">

  <h:table>
    <h:tr>
      <h:td>Apples</h:td>
      <h:td>Bananas</h:td>
    </h:tr>
  </h:table>

  <f:table>
    <f:name>African Coffee Table</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>

</root>
```

Prefijo

Namespace por defecto
 ("http://www.w3.org/2005/Atom"), utilizado
 etiquetas sin prefijo

```
<?xml version="1.0" encoding="UTF-8"?><feed
xmlns="http://www.w3.org/2005/Atom"
xmlns:thr="http://purl.org/syndication/thread/1.0"
xml:lang="es-ES"
xml:base="http://informatica.iesvalledeljerteplasencia.es/wordpress/wp-atom.php"
>
<title type="text">Ciclos Formativos IES Valle del Jerte &#8211; Plasencia</title>
<subtitle type="text">Contenidos elaborados por alumnos</subtitle>

<updated>2017-11-08T16:05:14Z</updated>

<link rel="alternate" type="text/html" href="http://informatica.iesvalledeljerteplasencia.es/wordpress/" />
<id>http://informatica.iesvalledeljerteplasencia.es/wordpress/feed/atom/</id>
<link rel="self" type="application/atom+xml" href="http://informatica.iesvalledeljerteplasencia.es/wordpress/feed/atom/" />

<generator uri="https://wordpress.org/" version="4.8.2">WordPress</generator>
<icon>http://informatica.iesvalledeljerteplasencia.es/wordpress/wp-content/uploads/2016/10/cropped-ic_sin fondo-32x32.png</icon>
<entry>
<author>
<name>Alegre</name>
</author>
<title type="html"><![CDATA[Tuquito]]></title>
<link rel="alternate" type="text/html" href="http://informatica.iesvalledeljerteplasencia.es/wordpress/tuquito/" />
<id>http://informatica.iesvalledeljerteplasencia.es/wordpress/?p=15545</id>
<updated>2017-11-07T15:18:04Z</updated>
<published>2017-11-07T14:51:33Z</published>
<category scheme="http://informatica.iesvalledeljerteplasencia.es/wordpress" term="Sin categoria" />
<summary type="html"><![CDATA[Tuquito.odp Pasos para
instalar tuquito: https://es.scribd.com/document/123432004/Tutorial-Tuquito]]></summary>
<content type="html" xml:base="http://informatica.iesvalledeljerteplasencia.es/wordpress/tuquito/"><![CDATA[<p><a href="http://informatica
.iesvalledeljerteplasencia.es/wordpress/wp-content/uploads/2017/11/Tuquito.odp.pptx">Tuquito.odp</a></p>
<p><a href="https://es.scribd.com/document/123432004/Tutorial-Tuquito">Pasos para instalar tuq</a><a href="https://es.scribd.com/document/123432004/Tutorial-Tuquito"
>uito:</a></p>
<p><a href="https://
<p><a class="a2a_dd
";title=Tuquito"
/buttons/favi.com
```

Namespace por defecto
 (“http://www.w3.org/2005/Atom”), **utilizado por las**
etiquetas sin prefijo

Ejemplo de uso de namespace en un documento Atom

XML: espacios de nombre (namespaces) (y III)

```
<rdf:RDF
  xmlns:dct="http://purl.org/dc/terms/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://www.w3.org/1999/xhtml/vocab#"
  xmlns:dcap="http://www.w3.org/ns/dcap#"
  xmlns:j.1="http://a9.com/~spec/opensource/1.1/"
  xmlns:api="http://purl.org/linked-data/api/vocab#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  <api:Page rdf:about="http://datos.gob.es/apidata/catalog/dataset/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior.rdf">
    <dct:hasPart rdf:resource="http://datos.gob.es/apidata/catalog/dataset/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior.rdf"/>
    <api:definition rdf:resource="http://datos.gob.es/apidata/catalog/meta/dataset/_id.rdf"/>
    <api:items rdf:parseType="Collection">
      <dcap:Dataset rdf:about="http://datos.gob.es/catalogo/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior">
        <dct:title xml:lang="es">Estadísticas de la Consejería de Turismo, Regeneración, Justicia y Administración Local</dct:title>
        <dcap:theme rdf:resource="http://datos.gob.es/kos/sector-publico/sector/legislacion-justicia"/>
        <dcap:theme rdf:resource="http://datos.gob.es/kos/sector-publico/sector/seguridad"/>
        <dcap:keyword>Estadísticas</dcap:keyword>
        <dct:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2019-07-25T06:42:37+02:00</dct:modified>
        <dcap:distribution>
          <dcap:Distribution rdf:about="http://datos.gob.es/catalogo/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior/resource/5efdf2d8-3957-4809-a37f-12a547d6e969">
            <dct:title xml:lang="es">CSV - Estadísticas de la Consejería de Turismo, Regeneración, Justicia y Administración Local</dct:title>
            <dct:format>
              <dct:IMT rdf:about="http://datos.gob.es/catalogo/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior/resource/5efdf2d8-3957-4809-a37f-12a547d6e969/format">
                <rdf:value>text/csv</rdf:value>
              </dct:IMT>
            </dct:format>
            <dcap:accessURL rdf:resource="http://juntadeandalucia.es/export/icms/jda_estadisticas_cji.csv"/>
          </dcap:Distribution>
        </dcap:distribution>
        <dct:publisher rdf:resource="http://datos.gob.es/recurso/sector-publico/org/Organismo/A01002820"/>
        <dcap:distribution>
          <dcap:Distribution rdf:about="http://datos.gob.es/catalogo/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior/resource/e8b02d4a-7c7a-44b8-88e5-9af104120fab">
            <dct:title xml:lang="es">JSON - Estadísticas de la Consejería de Turismo, Regeneración, Justicia y Administración Local</dct:title>
            <dct:format>
              <dct:IMT rdf:about="http://datos.gob.es/catalogo/a01002820-estadisticas-de-la-consejeria-de-justicia-e-interior/resource/e8b02d4a-7c7a-44b8-88e5-9af104120fab/format">
                <rdf:value>application/json</rdf:value>
              </dct:IMT>
            </dct:format>
            <dcap:accessURL rdf:resource="http://www.juntadeandalucia.es/icms/rest/datasets/jda_estadisticas.json?query=icms\field_taxonomias_buscador_value:organismos/justiciae">
            </dcap:Distribution>
          </dcap:distribution>
        </dct:publisher>
      </api:items>
    </api:Page>
  </rdf:RDF>
```


<![CDATA[.....]]>

- En un documento XML, una sección CDATA evita que el texto que contiene sea interpretado por el analizador sintáctico. No hay diferencia semántica entre una cadena de caracteres dentro de una sección CDATA y la sintaxis usual en la que "<" y "&" estarían representados por "& lt;" y "& amp;", respectivamente. **Fuente:** <https://es.wikipedia.org/wiki/CDATA>

```
<?xml version="2.0">
<channel>
  <title>RSS de noticias de portada</title>
  <link href="https://www.hoy.es/rss/2.0/portada" rel="alternate" type="application/rss+xml" title="RSS de noticias de portada"/>
  <description> RSS de noticias de portada </description>
  <language>es-ES</language>
  <ttl>60</ttl>
  <item>
    <title>
      <![CDATA[
        La tensión vuelve a Cataluña tras la histórica sentencia a los líderes
      ]]>
    </title>
    <guid>
      https://www.hoy.es/nacional/tension-vuelve-cataluna-historica-sentencia
    </guid>
    <pubDate>2019-10-15T01:42:11+02:00 </pubDate>
    <description>
      <![CDATA[
        El tribunal sostiene que su estrategia ha sido un engaño culminado co
      ]]>
    </description>
  </item>
  <item>
    <title>
      <![CDATA[
        Normalidad en El Prat tras el bloqueo y las cancelaciones
      ]]>
    </title>
    <guid>
      https://www.hoy.es/nacional/normalidad-prat-protestas-proces-incidentes-20191015094328-ntnc.html
    </guid>
    <pubDate>2019-10-15T09:45:08+02:00 </pubDate>
    <description>
      <![CDATA[
        El servicio del AVE en Girona sigue interrumpido por daños en la infraestructura
      ]]>
    </description>
```

<sender>Teresa Melo</sender>

El texto contenido dentro de las etiquetas **sender** es interpretado como elemento XML. Sin embargo si se escribe de esta manera:

<![CDATA[<sender>Agapito Morones</sender>]]>

Es interpretado como si se hubiera escrito así:

<sender>Agapito Morones</sender>

[https://www.w3.org/TR/REC-xml/
#sec-cdata-sect](https://www.w3.org/TR/REC-xml/#sec-cdata-sect)

Estructura y sintaxis de documentos XML: ampliación



- Diferencia mayúsculas de minúsculas
- Los **nombres de los elementos**:
 - **Primer carácter**: una letra del alfabeto inglés (A-Z, a-z) o el guión bajo (_)
 - **Resto**: una letra del alfabeto inglés (A-Z, a-z), el guión (-), el guión bajo (_), el punto (.) o números (0-9)
 - Se puede utilizar cualquier nombre a excepción de xml
 - No pueden contener espacios
- [Best Naming Practices](#)
- **Atributos reservados**:
 - El prefijo **xml:** se reserva para las especificaciones de XML, por lo que un atributo que definamos no puede iniciar con los caracteres xml
 - **xml:lang** → idioma del elemento
 - **xml:space: default o preserve** → procesamiento normal o conservar los espacios en blanco
 - **xml:id** → identificador único del elemento en todo el documento
- Comentarios → <!-- Comentario -->
- Recuerda, la gramática es el conjunto de reglas de un lenguaje.
- La versión que utilizaremos es XML 1.0, la recomendada por W3C

Al igual que HTML en XML podemos utilizar entidades de carácter que nos permiten utilizar caracteres reservados.

Entidad	Caracter	Descripción
<	<	Menor que
>	>	Mayor que
&	&	Ampersand
"	"	Comilla doble
'	'	Apóstrofe (o comilla sencilla)

IC Informática y Comunicaciones
IES Valle del Jerte
Placencia
informatica.iesvalledeljerte@participacion.es

- ```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

 <xs:simpleType name="stringtype">
 <xs:restriction base="xs:string"/>
 </xs:simpleType>

 <xs:simpleType name="inttype">
 <xs:restriction base="xs:positiveInteger"/>
 </xs:simpleType>

 <xs:simpleType name="decetype">
 <xs:restriction base="xs:decimal"/>
 </xs:simpleType>

 <xs:simpleType name="orderidtype">
 <xs:restriction base="xs:string">
 <xs:pattern value="..." />
 </xs:restriction>
 </xs:simpleType>

 <xs:complexType name="...">
 <xs:sequence>
 <xs:element name="..." />
 <xs:element name="..." />
 <xs:element name="..." />
 <xs:element name="..." />
 </xs:sequence>
 </xs:complexType>

 <xs:complexType name="...">
 <xs:sequence>
 <xs:element name="..." />
 <xs:element name="..." />
 <xs:element name="..." />
 </xs:sequence>
 </xs:complexType>

 <xs:complexType name="...">
 <xs:sequence>
 <xs:element name="..." />
 <xs:element name="..." />
 <xs:element name="..." />
 </xs:sequence>
 </xs:complexType>

 <xs:complexType name="...">
 <xs:sequence>
 <xs:element name="..." />
 <xs:element name="..." />
 <xs:element name="..." />
 </xs:sequence>
 </xs:complexType>

 <xs:attribute name="..." />
</xs:schema>

```

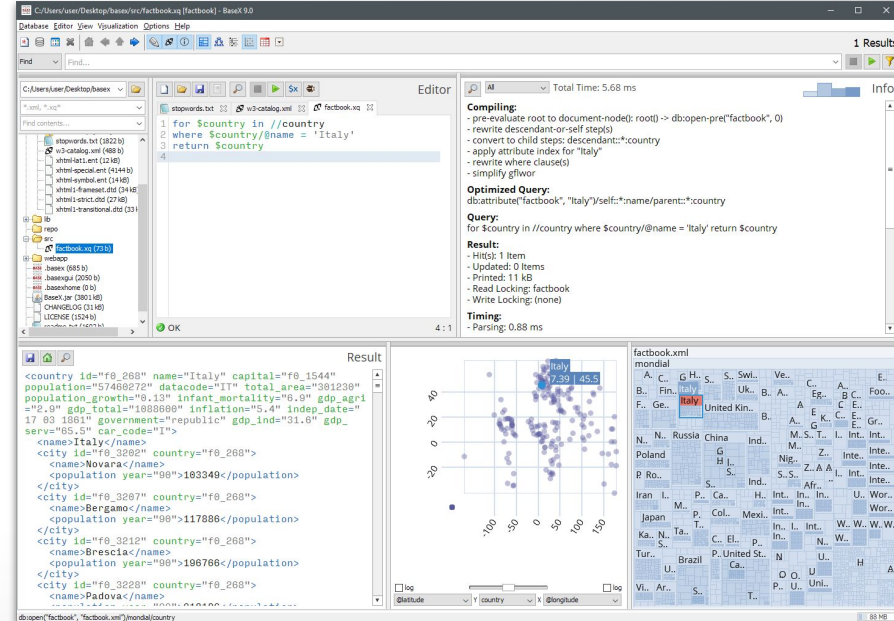
# Almacenamiento de información

Bases de datos nativas XML (NXD - Native XML Database) → almacena documentos XML y facilita funcionalidad sobre los mismos similar a la que podemos encontrar en un sistema gestor de base de datos relacional.

Dos tipos:

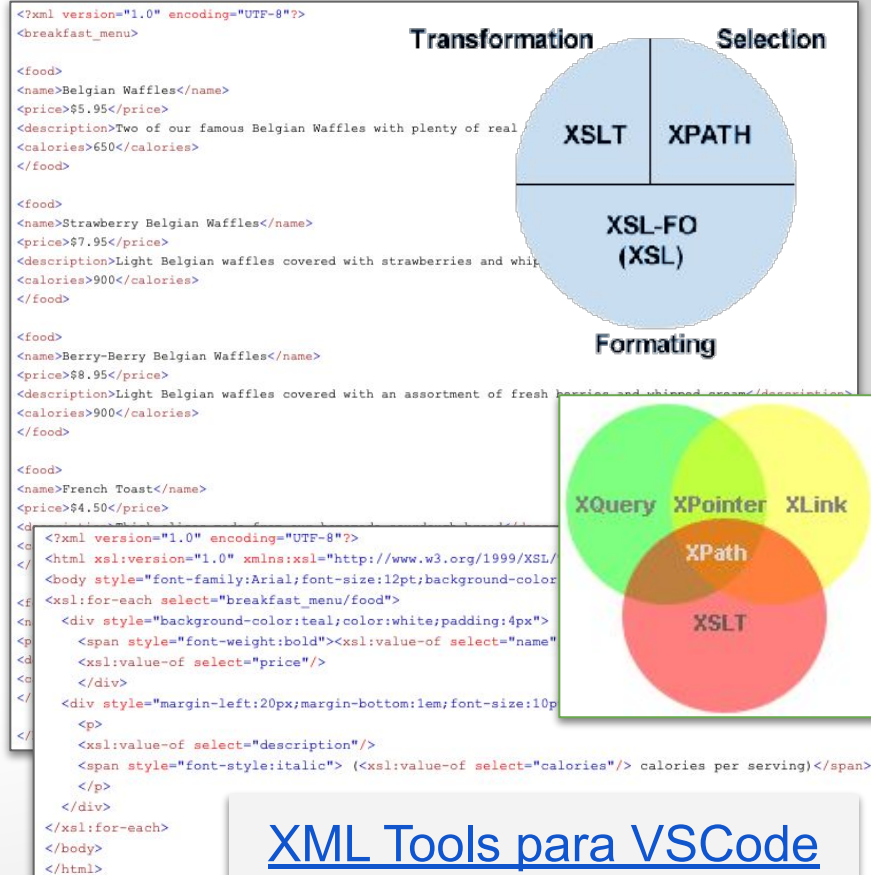
- **XML-enabled BD** → entrada y salida en formato XML, pero almacena los datos internamente como los sistemas tradicionales.
- **NXD - Native XML Database** → entrada y salida en XML, y además la unidad de almacenamiento son documentos XML.

<https://basex.org/>



# Conversión y adaptación de documentos XML (I)

- **XSL** - Extensible Stylesheet Language, "lenguaje extensible de hoja de estilo", cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil. Dividido en tres partes:
  - **XSLT** (siglas de Extensible Stylesheet Language Transformations, lenguaje de hojas extensibles de transformación), que permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).
  - **XSL-FO** (lenguaje de hojas extensibles de formateo de objetos), que permite **especificar el formato visual** con el cual se quiere presentar un documento XML, es **usado principalmente para generar documentos PDF**.
  - **XPath**, **XML Path Language** ([W3School](http://W3School)), es una sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML.
- **XQuery (XML Query)**: lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web. Es semánticamente similar a SQL. Utiliza XPath para acceder a las partes del documento XML. También se apoya en XML Schema.
- Otras especificaciones relacionadas: [XInclude](#), [XLink](#) y [XPointer](#)
- [También podemos utilizar CSS con algunas adaptaciones](#)



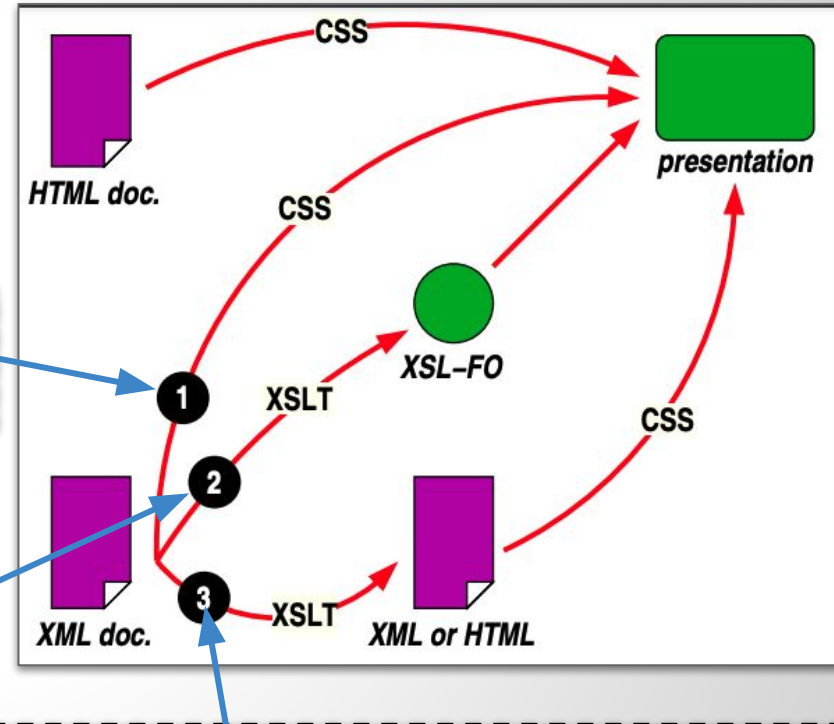


# Conversión y adaptación de documentos XML (y II)

<https://www.w3.org/Style/CSS-vs-XSL.en.html>

Si el documento no tiene que ser transformado,  
lo recomendable es utilizar CSS

Si el documento tiene que ser transformado,  
utiliza XSLT junto a XSL-FO (por ejemplo para  
presentarlo en formato PDF)



Si el documento tiene que ser transformado, utiliza XSLT para  
convertirlo en otro XML o HTML y después con CSS presentarlo



# The Extensible Stylesheet Language Family (XSL) - W3C



<https://www.w3.org/Style/XSL/>

## The Extensible Stylesheet Language Family (XSL)

**XSL** is a family of recommendations for defining XML document transformation and presentation. It consists of three parts:

[XSL Transformations](#) (XSLT)

a language for transforming XML;

[The XML Path Language](#) (XPath)

an expression language used by XSLT (and many other languages) to access or refer to parts of an XML document;

[XSL Formatting Objects](#) (XSL-FO)

an XML vocabulary for specifying formatting semantics.

An XSLT stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses a formatting vocabulary, such as (X)HTML or XSL-FO. For a more detailed explanation of how XSL works, see the [What Is XSL](#) page.

XSLT is developed by the W3C [XSLT Working Group \(members only\)](#) whose [charter](#) is to develop the next version of XSLT. XSLT is part of W3C's [XML Activity](#), whose work is described in the [XML Activity Statement](#).

XPath is developed jointly by the [XPath](#) and XSLT Working Groups.

The XSL-FO work at W3C was taken over by the [XML Print and Page Layout Working Group](#), which has now been closed.

### News

#### 2017-06-08: XSLT 3.0 is a W3C Recommendation

[XSLT 3.0 is a W3C Recommendation.](#)

#### 2017-03-21: XQuery 3.1, XPath 3.1, XDM 3.1 etc W3C Recommendations

[XPath 3.1, XQuery 3.1, XQueryX 3.1 and supporting documents now W3C Recommendations.](#)

#### 2014-04-12: XQuery 3.0, XPath 3.0, XDM 3.0 etc W3C Recommendations

[XPath 3, XQuery 3 etc now W3C Recommendations.](#)

#### 2011-05-04: Liquid XML Editor New 2011 Functionality

[Liquid XML Editor 2011](#) now features an advanced XSLT debugger and an XSLT-aware editor with validation, supporting XSLT 1 and 2.

### Specifications

[XSLT 3.0](#)  
[XSLT 2.0](#)  
[XSLT 1.0](#)  
[XPath 3.1](#)  
[XPath 3.0](#)  
[XPath 2.0](#)  
[XPath 1.0](#)  
[XQuery 3.1 and XPath 3.1 Data Model \(XDM\)](#)  
[XQuery 3.0 and XPath 3.0 Data Model \(XDM\)](#)  
[XQuery 1.0 and XPath 2.0 Data Model](#)

### Tutorials

[XSL-FO tutorial](#) by RenderX  
[How to Develop Stylesheets for XML to XSL-FO Transformation](#) by Antenna House  
[Introduction to XSL](#) by Miloslav Nic  
Tutorial from iX magazine ([German](#), [English](#))

### Reference

[XSL-FO 1.1 in PDF](#) produced by AltSoft.

# Propuestas de trabajos voluntarios

- Generación de documentación automatizada, por ejemplo: [xs3p](#), [XSDdoc](#) o [xsd doc](#) ([ejemplos](#)), ...
- Visualización gráfica de schema, por ejemplo: [xsdvi](#), [XSD Diagram](#), ...
- Software específico: [Liquid Studio](#), [oxygen](#), [XSD Diagram](#),...

## Overview Namespace Component Index Help

DETAILS: [DOCUMENTATION](#) | [ELEMENTS](#) | [ATTRIBUTES](#) | [SOURCE](#)

### http://www.w3.org/2001/XMLSchema element documentation

#### Documentation

#### Properties

This component is not nillable.

#### Model

```
<documentation
 lang = xml:language
 source = xs:anyURI
 (any attribute from any other namespace than current namespace)>
 (any element from any namespace) *
</documentation>
```

## Attribute Summary

|                                                                                |                        |
|--------------------------------------------------------------------------------|------------------------|
| <a href="#">xml:language</a>                                                   | <a href="#">lang</a>   |
| In due course, we should install the relevant ISO 2- and 3-letter codes as the |                        |
| <a href="#">xs:anyURI</a>                                                      | <a href="#">source</a> |

## Attribute Detail

#### lang

In due course, we should install the relevant ISO 2- and 3-letter codes as the enumerated possi

#### Type:

[xml:language](#)

#### Use:

optional

#### Form:

unqualified

