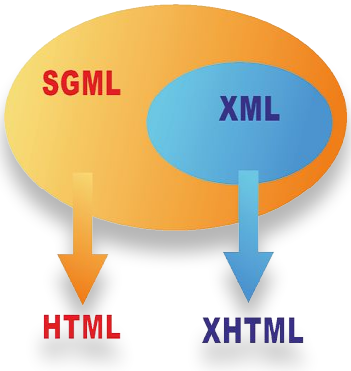
Lenguajes de marcas y Sistemas
de Gestión de la Información

¿Qué es XML?

- Wikipedia: "*Es un conjunto de reglas para codificar un documento en un formato legible por una máquina*"
- Deriva del lenguaje de marcas **SGML**
- Hay cientos de lenguajes de marcas que cumplen con las especificaciones de XML y se manejan de la misma forma
- Ejemplos: XHTML, RSS, Atom, Docbook, OpenDocument, OOXML, SVG, MathML, SOAP, …*(1)*
- Prácticamente todos los nuevos lenguajes de la web están basados en XML



(1) https://en.wikipedia.org/wiki/List_of_XML_markup_languages

Características de XML

- Sus siglas provienen de **eXtensible Markup Language**
- Puede **almacenar y organizar cualquier tipo de información**
- Es un **estándar internacional abierto (W3C)**
- Utiliza **Unicode** por defecto → Cualquier idioma
- Permite **revisar sintaxis** y **validar documentos**
- Es **fácil de leer** por personas y por aplicaciones o programas
- Es posible **exportarlo** a numerosos formatos

Usos de XML

- **Intercambio de información entre aplicaciones**
  - Almacena información en **texto plano => fácilmente legible** por cualquier software,
  - Utilizado por varios servicios en **Internet** para **ofrecer resultados de consultas**.
- **Documentación**
  - Muy utilizado para almacenar documentos, especialmente en el formato **ePUB** para eBooks, debido a su simplicidad y **gran capacidad semántica**.
- **Bases de datos**
  - Formato fundamental en las **BBDD empresariales**
  - Los lenguajes **XQuery** y **XPath** permiten **buscar y navegar datos** en los documentos XML.
  - Muchas BBDD tienen XML como un formato para almacenar atributos o columnas

Usos de XML

- **Formato de imagen vectorial**
  - **SVG** es un lenguaje XML utilizado para representar imágenes vectoriales que **no pierden calidad** al ampliar o reducirse, lo que lo convierte en el formato ideal para logotipos, líneas, formas e iconos de las páginas web.
- **Archivos de configuración**
  - Formato ideal para almacenar **información de configuración** y **archivos log** de diversos dispositivos hardware. Muchos switches, routers, impresoras y servidores utilizan XML para estructurar esta información de manera semántica y facilitar su modificación.

Tecnologías relacionadas

- DTD. Document Type Definition
- XML Schema
- Relax NG
- Namespacing
- XPath
- CSS
- XSLT
- XQuery
- DOM
- SAX
- XForms
- XLink
- XPointer

- **DTD**. *Document Type Definition*, **definición de tipo de documento**. Es un lenguaje que permite especificar reglas que han de cumplir los documentos XML a los que se asocien. Es decir, permite crear documentos de validación para archivos XML.
- **XML Schema**. La función que realiza esta tecnología es la misma que la anterior. La diferencia está en que los documentos XML Schema poseen una **sintaxis 100% XML** (DTD se basa en SGML), por lo que es un formato orientado a reemplazar al anterior.
- **Relax NG**. Otro formato de definición de validaciones para documentos XML. Es una alternativa a las dos anteriores. Tiene una **sintaxis más sencilla.**
- **Namespacing**, espacios de nombres. Es una norma que permite **conseguir nombres de elementos que carecen de ambigüedad**. Es decir nombres únicos para los distintos elementos que están dentro de los documentos XML.
- **XPath**. Lenguaje de consulta que permite **seleccionar o acceder a partes de un documento XML.**
- **CSS**. *Cascade StyleSheet*. Hojas de estilo en cascada. Permiten dar formato a los documentos XML o HTML.

- **XSLT**. Sirve **para lo mismo que CSS**: dar formato a un documento XML. Tene muchas más posibilidades que CSS. Tiene la **capacidad de convertir** un documento **XML en un documento HTML** o incluso a tipos comerciales como **PDF**.
- **XQuery**. Permite **consultar datos** de los documentos XML, manejándolos **como si fueran parte de una base de datos**.
- **DOM**. *Document Object Model*, tecnología que permite acceder a la estructura jerárquica del documento. Normalmente para poder ser utilizada por un lenguaje de programación (como **JavaScript**) y poder dar dinamismo a su contenido.
- **SAX**. ***Simple API for XML***, permite el uso de herramientas para **acceder a la estructura jerárquica del documento XML a través de otro lenguaje**. Se usa mucho en la programación de aplicaciones en lenguaje **Java**.
- **XForms**. Formato de documentos pensados para ser usados **como formularios de introducción de datos**.
- **XLink**. Permite **crear hipervínculos en un documento XML**.
- **XPointer**. Semejante al anterior, especifica **enlaces a elementos externos al documento XML**.

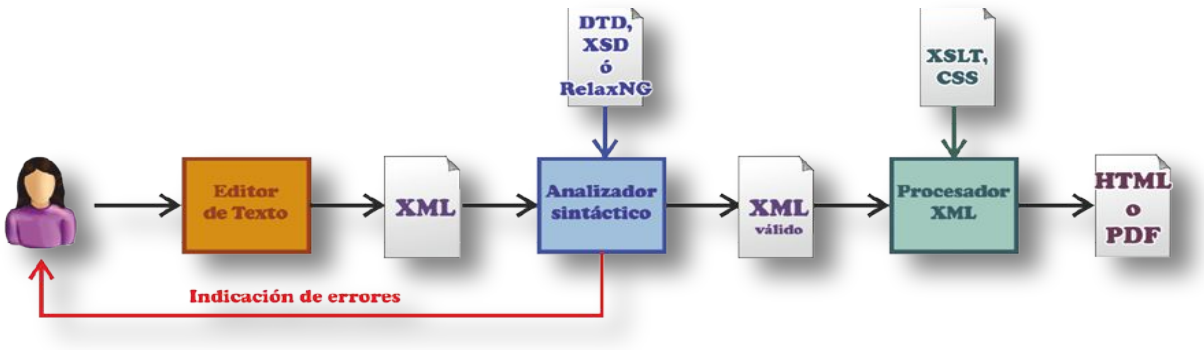
Software para producir XML

1. Un **editor de texto plano** para escribirlo
2. Un **Analizador sintáctico o parser** capaz de entender y validar XML
   - Apache Xerces quizá sea el validador más popular
3. Un **procesador XML** capaz de producir una presentación visual sobre el documento
   - Un navegador web puede hacerlo
   - Existe software que transforma XML en, por ejemplo, HTML (Apache Xalan o Saxon)

- **DTD**. *Document Type Definition*, **definición de tipo de documento**. Es un lenguaje que permite especificar reglas que han de cumplir los documentos XML a los que se asocien. Es decir, permite crear documentos de validación para archivos XML.
- **XML Schema**. La función que realiza esta tecnología es la misma que la anterior. La diferencia está en que los documentos XML Schema poseen una **sintaxis 100% XML** (DTD se basa en SGML), por lo que es un formato orientado a reemplazar al anterior.
- **Relax NG**. Otro formato de definición de validaciones para documentos XML. Es una alternativa a las dos anteriores. Tiene una **sintaxis más sencilla.**
- **Namespacing**, espacios de nombres. Es una norma que permite **conseguir nombres de elementos que carecen de ambigüedad**. Es decir nombres únicos para los distintos elementos que están dentro de los documentos XML.
- **XPath**. Lenguaje de consulta que permite **seleccionar o acceder a partes de un documento XML.**
- **CSS**. *Cascade StyleSheet*. Hojas de estilo en cascada. Permiten dar formato a los documentos XML o HTML.

- **XSLT**. Sirve **para lo mismo que CSS**: dar formato a un documento XML. Tene muchas más posibilidades que CSS. Tiene la **capacidad de convertir** un documento **XML en un documento HTML** o incluso a tipos comerciales como **PDF**.
- **XQuery**. Permite **consultar datos** de los documentos XML, manejándolos **como si fueran parte de una base de datos**.
- **DOM**. *Document Object Model*, tecnología que permite acceder a la estructura jerárquica del documento. Normalmente para poder ser utilizada por un lenguaje de programación (como **JavaScript**) y poder dar dinamismo a su contenido.
- **SAX**. ***Simple API for XML***, permite el uso de herramientas para **acceder a la estructura jerárquica del documento XML a través de otro lenguaje**. Se usa mucho en la programación de aplicaciones en lenguaje **Java**.
- **XForms**. Formato de documentos pensados para ser usados **como formularios de introducción de datos**.
- **XLink**. Permite **crear hipervínculos en un documento XML**.
- **XPointer**. Semejante al anterior, especifica **enlaces a elementos externos al documento XML**.

Proceso de funcionamiento productivo de un XML



10

- **DTD**. *Document Type Definition*, **definición de tipo de documento**. Es un lenguaje que permite especificar reglas que han de cumplir los documentos XML a los que se asocien. Es decir, permite crear documentos de validación para archivos XML.
- **XML Schema**. La función que realiza esta tecnología es la misma que la anterior. La diferencia está en que los documentos XML Schema poseen una **sintaxis 100% XML** (DTD se basa en SGML), por lo que es un formato orientado a reemplazar al anterior.
- **Relax NG**. Otro formato de definición de validaciones para documentos XML. Es una alternativa a las dos anteriores. Tiene una **sintaxis más sencilla.**
- **Namespacing**, espacios de nombres. Es una norma que permite **conseguir nombres de elementos que carecen de ambigüedad**. Es decir nombres únicos para los distintos elementos que están dentro de los documentos XML.
- **XPath**. Lenguaje de consulta que permite **seleccionar o acceder a partes de un documento XML.**
- **CSS**. *Cascade StyleSheet*. Hojas de estilo en cascada. Permiten dar formato a los documentos XML o HTML.

- **XSLT**. Sirve **para lo mismo que CSS**: dar formato a un documento XML. Tene muchas más posibilidades que CSS. Tiene la **capacidad de convertir** un documento **XML en un documento HTML** o incluso a tipos comerciales como **PDF**.
- **XQuery**. Permite **consultar datos** de los documentos XML, manejándolos **como si fueran parte de una base de datos**.
- **DOM**. *Document Object Model*, tecnología que permite acceder a la estructura jerárquica del documento. Normalmente para poder ser utilizada por un lenguaje de programación (como **JavaScript**) y poder dar dinamismo a su contenido.
- **SAX**. ***Simple API for XML***, permite el uso de herramientas para **acceder a la estructura jerárquica del documento XML a través de otro lenguaje**. Se usa mucho en la programación de aplicaciones en lenguaje **Java**.
- **XForms**. Formato de documentos pensados para ser usados **como formularios de introducción de datos**.
- **XLink**. Permite **crear hipervínculos en un documento XML**.
- **XPointer**. Semejante al anterior, especifica **enlaces a elementos externos al documento XML**.

Estructura (sintaxis) de un documento XML (I)

- En la **primera línea** se incluye la **declaración XML**:

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

- Normalmente incluye un **prólogo:**
  - Declaración del documento: indica el tipo de documento XML (versión y codificación)
  - Instrucciones para el procesado del documento
    `<?xml-stylesheet type="text/xsl" href="stylesheet.xsl"?>`
  - Comentarios `<!-- esto es un comentario -->`
  - Ruta hacia el documento DTD, XSD o Relax NG que para validar el documento XML actual
  - Indicación de otros documentos que afectan al actual. Como por ejemplo los documentos XSLT que sirven para dar formato al documento XML.

Estructura (sintaxis) de un documento XML (II)

- Incluye **elementos** que empiezan por una **etiqueta o tag de apertura**, a continuación se pone el contenido y termina con la **etiqueta o tag de cierre**:

```
<alumno>Pepa Ramírez Heredia</alumno>
```

- **Elemento raíz:** Todo el contenido debe incluirse dentro de este elemento obligatorio. Se abre tras el prólogo y se debe cerrar justo al final del documento.
- **Dentro de un elemento** (sea raíz o no), puede haber:
  - **Más elementos**
  - **Atributos**
  - **Texto normal**
  - **Entidades**
  - **Comentarios**

Estructura (sintaxis) de un documento XML (III)

- Los elementos pueden tener **atributos** (entrecomillados):

```
<alumno dni="45.123.123-J">Pepa Ramírez Heredia</alumno>
```

- Tiene **estructura jerárquica**:

```
<alumnos>
    <alumno dni="45.123.123-J">Pepa Ramírez Heredia</alumno>
    <alumno dni="41.321.321-H">José González Pons</alumno>
</alumnos>
```

Estructura (sintaxis) de un documento XML (IV)

- Los elementos vacíos tienen que **cerrarse siempre**:

  `<alumno dni="45.123.123-J">Pepa Ramírez Heredia</alumno>`

  `<becado></becado>` **ó** `<becado/>`
- Las etiquetas **distinguen mayúsculas de minúsculas**
- Los elementos deben estar **correctamente anidados**

Estructura (sintaxis) de un documento XML (V)

- Las **etiquetas XML**:
  - **Pueden** contener **letras, números y caracteres especiales**
  - **No pueden empezar** por ningún carácter de **puntuación**
  - **No pueden empezar por xml o XML**
  - **No pueden** contener **espacios**

Estructura (sintaxis) de un documento XML (VI)

Ejemplo 1:

marcadores.xml

http://bit.ly/40fEmuJ

```
<?xml version="1.0" encoding="UTF-8"?>
<marcadores>
  <pagina fecha="16/2/2023">
    <nombre>GitHub</nombre>
    <descripcion>GitHub: Let's build from here</descripcion>
    <url>https://github.com/</url>
  </pagina>
  <pagina fecha="21/3/2023">
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina fecha="25/3/2023">
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<marcadores>
  <pagina>
    <nombre>Abrirllave</nombre>
    <descripcion>Tutoriales de informática.</descripcion>
    <url>http://www.abrirllave.com/</url>
  </pagina>
  <pagina>
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina>
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```

Atributos

- Los **atributos** son más naturales para expresar **metainformación**
- La utilización de atributos **simplifica la estructura del documento**
- Pero ...
    - Los atributos **no** pueden incluir **valores múltiples**
    - **No** pueden incluir **estructura compleja**
    - **No** son **fáciles de extender**

Atributos (II)

- Se definen **dentro de las etiquetas de apertura de los elementos**. Sirven para indicar propiedades de los elementos a los que se les asigna un determinado valor.
- Para ello se indica el **nombre del atributo seguido del signo = y del valor (entre comillas)** que se le da al atributo. Ejemplo:

```
<persona complejidad="alta">
    <nombre>Jorge</nombre>
    <apellido>Sánchez</apellido>
</persona>
```

- Un elemento **puede contener varios atributos**:

```
<persona privacidad="alta" tipo="autor">
    <nombre>Jorge</nombre>
    <apellido>Sánchez</apellido>
</persona>
```

Texto

- Está siempre entre una etiqueta de apertura y una de cierre. Eso significa que **todo texto es parte de un elemento XML**.
- Se puede escribir **cualquier carácter Unicode** en el texto, **pero no es válido utilizar caracteres** que podrían dar lugar a confusión como los signos separadores **<** y **>** por ejemplo

CDATA

- Disponemos de la posibilidad de **marcar texto para que sea procesado literalmente como texto y no como sintaxis de XML o código de otros lenguajes**. Para ello, el texto se coloca dentro de un elemento CDATA.

```
<! [CDATA [ texto no procesable… ]]>
```

```
<?xml version="1.0"?>
<documento>
    <título>Prueba</título>
    <ejemplo>
        <![CDATA[
            En HTML la negrita se escribe: <strong>
        ]]>
    </ejemplo>
</documento>
```

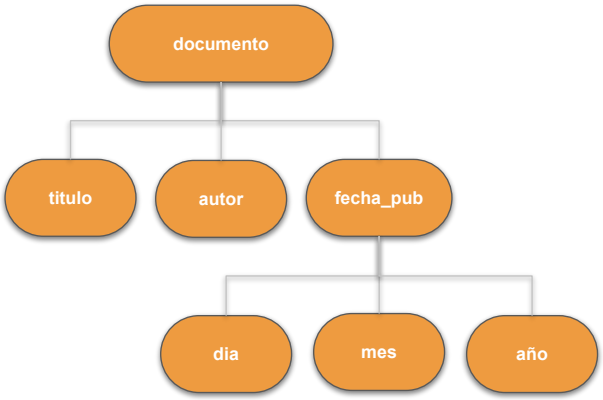
Entidades

- Representan caracteres individuales. Para representar caracteres especiales o bien caracteres inexistentes en el teclado habitual. Se trata de códigos que empiezan con el **signo &** al que sigue el nombre de la entidad o el número Unicode del carácter que deseamos representar.
- En XML hay definidas cinco entidades:
    - `>` Símbolo > (mayor que)
    - `<` Símbolo < (menor que)
    - `&` Símbolo &
    - `&quot;` Símbolo “ (comillas)
    - `&apos;` Símbolo ‘ (apóstrofe)

Jerarquía XML

- Los elementos de un documento XML establecen una jerarquía que estructura el contenido. Se puede representar en forma de árbol. Ejemplo:

```
<?xml version="1.0" ?>
<documento>
    <título>Apuntes de XML</título>
    <autor>Jorge Sánchez</autor>
    <fecha_pub>
        <dia>18</dia>
        <mes>Enero</mes>
        <año>2009</año>
    </fecha_pub>
</documento>
```



XML bien formado

- Representan caracteres individuales. Para representar caracteres especiales o bien caracteres inexistentes en el teclado habitual. Se trata de códigos que empiezan con el **signo &** al que sigue el nombre de la entidad o el número Unicode del carácter que deseamos representar.
- En XML hay definidas cinco entidades:
    - `>` Símbolo > (mayor que)
    - `<` Símbolo < (menor que)
    - `&` Símbolo &
    - `&quot;` Símbolo “ (comillas)
    - `&apos;` Símbolo ‘ (apóstrofe)

Ejercicio 01: Equipos de fútbol

Ejercicio 1:

Crea un documento XML llamado *equiposFutbol.xml*

que contenga, como mínimo, los siguientes elementos mínimos:

- 2 x Equipos
- 2 x Jugadores/Equipo
- Posibles posiciones: portero, defensa, lateral izquierdo, lateral derecho, centrocampista, delantero.

Jerarquía:

- Equipos de fútbol
  - Equipo
    - Nombre
    - Ciudad
    - Entrenador
    - Jugadores
      - Jugador (atributo: posición)
        - Nombre
        - Nacionalidad


```xml
<?xml version="1.0" encoding="UTF-8"?>
<marcadores>
  <pagina>
    <nombre>Abrirllave</nombre>
    <descripcion>Tutoriales de informática.</descripcion>
    <url>http://www.abrirllave.com/</url>
  </pagina>
  <pagina>
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina>
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```

Documentos bien formados y válidos

- Los documentos bien formados (*well formed*) son aquellos que **cumplen con las reglas de sintaxis** de XML.
- Lo **mínimo que se exige** a un documento XML es **que esté bien formado**.
- Si además cumple otra serie de relaciones y restricciones referidas a un dialecto concreto XML (XHTML, SVG, OpenDocument, Docbook, etc.), se dice que el documento es válido.
- Las **relaciones y restricciones** que especifican un determinado dialecto XML **se especifican en un documento externo**.

Lenguajes de Definición de Documentos o esquemas

- Se utilizan **para especificar** los nombres de los elementos, los atributos que pueden tener, la estructura del documento, tipos de datos que pueden contener, etc.)
- Existen **varios tipos**, los más usados son:
  - **DTD Document Type Definition (*.dtd*):** Heredado de SGML y con algunas limitaciones
  - **XML Schema (*.xsd*):** Evolución de DTD específico para XML y descrito por el W3C
  - **RELAX NG (*.rng / .rnc*):** Más intuitivo que XML Schema, desarrollado por OASIS y muy utilizado actualmente

Los archivos **DTD** (Document Type Definition) suelen tener la extensión de archivo **`.dtd`**.

Los archivos **XML Schema** suelen tener la extensión de archivo **`.xsd`**, que proviene de XML Schema Definition.

Los archivos Relax NG pueden tener dos formatos diferentes: el formato XML y el formato compacto.

Los archivos **Relax NG en formato XML** suelen tener la extensión de archivo **`.rng`**

Mientras que los archivos **Relax NG en formato compacto** suelen tener la extensión de archivo **`.rnc`**.

Espacios de nombres (XML namespaces)

- Al ser XML un lenguaje extensible en el que cada uno puede definir su dialecto y poner los nombres de elementos y etiquetas que desee, puede haber coincidencias de nombres con diferente significado.
- Un espacio de nombres es una **especificación de un dialecto** en el que los nombres de los elementos y atributos son únicos y **se especifica** en el documento XML **mediante una URI utilizando el atributo reservado *xmlns***, por ejemplo:

```
<html xmlns:xhtml="http://www.w3.org/1999/xhtml">
```

Ejemplos:

- **XML:**
  - marcadores.xml
- **DTD Document Type Definition:**
  - marcadores.dtd
- **XML Schema:**
  - marcadores.xsd
- **RELAX NG:**
  - marcadores.rng

Puedes descargar los ejemplos desde:

http://bit.ly/3zcqdSQ

Herramientas

- https://codebeautify.org/xmlviewer/
  - Vista en árbol
  - Embellecedor de código
  - Minimizador
  - Conversor a JSON
  - Exportación a CSV
- https://codebeautify.org/xmlvalidator
  - Validador de código XML