



Tecnología XML

Unidad: 3
Laboratorio de Programación

Universidad Nacional de la Patagonia Austral
Unidad Académica Río Gallegos



Indice

- Introducción: HTML, SGML
- XML: concepto, características, comparación con HTML
- Herramientas: editor de texto, navegador y XML Copy Editor
- Tipos de documentos XML
 - Bien formados
 - Válidos
- Reglas sintácticas XML
- Técnicas de validación: DTD y XML Schemas
- DTD: concepto, elementos, atributos y declaraciones
- Ejemplos y ejercitación

XML Laboratorio de Programacion 1

Qué es XML?

Repasemos lo conocido...



HTML (Hypertext Markup Language)

- Lenguaje para transmitir por la red documentos sencillos y genéricos (**cabeceras, párrafos, listas, ilustraciones, etc., algo de hipertexto y multimedia**)
- Lenguaje de visualización (**apariciencia del documento**)
- Su sencillez le proporciona éxito inmediato (**revolución en Internet**)
- HTML evoluciona: **tablas, formularios..., mapas de imagen..., scripts y applets...**

XML Laboratorio de Programacion 3



HTML (Hypertext Markup Language)

Limitaciones:

- limitación para describir documentos complejos o datos
- elementos estructurales iguales para cualquier documento
- estructura demasiado flexible para algunos documentos (datos...)
- enlaces hipertextuales muy simples
- presentación pobre de los documentos a pesar de las ampliaciones
- ausencia de normalización

Necesidad de un nuevo lenguaje para la web

XML Laboratorio de Programacion 4



SGML (Standard Generalized Markup Language)

- Estándar internacional para definir la estructura y contenido de múltiples tipos de documentos electrónicos
- Metalenguaje: permite crear lenguajes de marcas propios que describan la información contenida en múltiples clases de documentos
- El Origen de todos los lenguajes de marcas existentes
- GML (1969) y SGML (1986)
- Norma de facto para el intercambio de documentos extensos, complejos y de tipos muy diferentes (documentación técnica de aviones, historiales médicos, notaciones musicales...)
- Lenguaje robusto y potente / complejo, exhaustivo, y caro de implantar y mantener

XML Laboratorio de Programacion 5

SGML (Standard Generalized Markup Language)

- Problemas de SGML:
 - Es complicado.
 - Es costoso.
 - El diseño de documentos es costoso.
 - La mano de obra es cara.
 - Aunque hay herramientas gratuitas, las que no lo son muy costosas.

XML

Laboratorio de Programacion

6

Se requiere algo nuevo...

- Barato, veloz y sencillo:
 - Para crear documentos.
 - Para procesar documentos.
 - Para presentar documentos.
- Extensible:
 - Un conjunto de reglas, no un conjunto de etiquetas.
- Compatible con el HTML:
 - Debe tener una manera sencilla de convertir HTML.
- Compatible con el SGML:
 - Debe de conservar su potencia sin contener complejidades no necesarias.

XML

Laboratorio de Programacion

7

XML

Qué es XML?

- El Lenguaje de Marcado Extensible (*Extensible Markup Language, XML*).
 - Un metalenguaje de marcado (lenguaje de marcas).
 - Marca: en XML es una etiqueta que ayuda a organizar la estructura de un documento y define semánticamente la información que identifica.
- Una recomendación técnica del W3C.
 - Es un estándar del W3C, no de alguna compañía.
 - W3C: es un consorcio de industrias internacionales, constituido con el objetivo de desarrollar protocolos comunes para Internet.
- Multiplataforma, simple, fácil de aprender.
 - Apto para cualquier plataforma, SO o entorno de trabajo (documentos en ASCII de 7 u 8 bits)
 - Optimizado para usarse en Internet.
- XML es una versión simplificada de SGML (Standard Generalized Markup Language)
- Libre (y gratuito)

XML

Laboratorio de Programacion

9

Qué es XML?...

"Lenguaje de **marcado**" basado en marcas:

- **Marcas** [mark-up] construcciones con etiquetas. Comienzan con "<" y terminan con ">".
- **Datos** resto de contenido del documento que se encuentra entre marcas
- Ejemplo:
 - `<autor>Deepak Chopra</autor>`

XML

Laboratorio de Programacion

10

Qué es XML?...

"**Meta-lenguaje**":

- Se utiliza para crear lenguajes de anotación
- Permite al usuario definir sus propios lenguajes de anotación adaptados a sus necesidades. Por ejemplo:
 - Información (marca): Deepak Chopra
 - Meta-información (dato): `<autor>`

XML

Laboratorio de Programacion

11



Qué NO es XML?...

- NO es una versión mejorada de HTML
- NO es un lenguaje para hacer mejores páginas Web
- NO es un lenguaje sustituto de HTML
- NO es un lenguaje difícil

XML Laboratorio de Programacion 12



Características

- Extensible.
 - Se pueden definir nuevas etiquetas
- Estructurado.
 - Se puede modelar datos a cualquier nivel de complejidad, y su definición está en una DTD, Document Type Definition .
- Validable.
 - Cada documento se puede validar frente a una DTD, o en su defecto, se puede declarar bien formado
- Independiente del medio.
 - Para publicar contenidos en múltiples formatos
- Independiente de fabricante y plataforma.
 - Para poder utilizar cualquier herramienta estándar

XML Laboratorio de Programacion 13



XML: puntos importantes

1. XML sirve para estructurar datos
2. XML se parece a HTML
3. XML es texto, pero no está pensado para ser leído
4. XML es una familia de tecnologías
5. XML es nuevo, pero no tanto
6. XML lleva a HTML hacia XHTML
7. XML es modular
8. XML es la base de la Web Semántica
9. XML es gratuito e independiente de la plataforma y está bien soportado

XML Laboratorio de Programacion 14



Para qué sirve?

- Intercambio de datos entre aplicaciones heterogéneas.
- Configuración de aplicaciones
- Crear protocolos para el intercambio de datos entre miembros de una industria.
- Facilitar el procesamiento de datos usando software barato.
- Generación de vistas (HTML, WML, PDF, etc.) a partir de datos en documentos XML.
- Proporcionar metadatos que mejoran la calidad de la búsqueda de información.
- Etc.

XML Laboratorio de Programacion 15



HTML vs. XML

HTML	XML
<ul style="list-style-type: none"> ■ HTML es una aplicación de SGML. ■ HTML describe tanto estructura (ej. <body>, <h2>, <p>) como apariencia (ej.
, ,) ■ HTML usa un conjunto limitado de etiquetas ■ Escasa capacidad de procesamiento, el browser es un mero visor de página ■ El problema de la 'no compatibilidad' y las diferencias entre browsers ha alcanzado un punto en el que la solución es difícil. 	<ul style="list-style-type: none"> ■ XML es un subconjunto de SGML. ■ XML describe solo contenido o semántica ■ En XML, se puede crear sus propias etiquetas ■ Gran capacidad para procesar documentos, el browser es una plataforma para el desarrollo de aplicaciones ■ Fin de la guerra de los navegadores y etiquetas propietarias

XML Laboratorio de Programacion 16



Conceptos básicos XML

Separación lógica de **contenido**, descripción de la **estructura** y **presentación**. Esto permite máxima independencia y flexibilidad.

- *Contenido*: datos. Documento XML.
- *Estructura*: reglas de estructura de los datos.
 - DTD y XML Schema.
- *Presentación*: el formato para mostrar la información. Hojas de estilo

XML Laboratorio de Programacion 17

Conceptos básicos XML...

- Separación de procesamiento, presentación, estructura y contenido:

The diagram illustrates the flow of XML processing. It is divided into three main categories: **CONTENIDO** (Content), **ESTRUCTURA** (Structure), and **PRESENTACION** (Presentation). Under **CONTENIDO**, there is an 'XML' box. Under **ESTRUCTURA**, there are 'DTD' and 'Schema' boxes. Under **PRESENTACION**, there are 'XSL' and 'CSS' boxes. Arrows from all these boxes point to a central 'PROCESAMIENTO' (Processing) box. From 'PROCESAMIENTO', an arrow points to an 'HTML' box, which is labeled 'SALIDA WEB' (Web Output).

XML Laboratorio de Programacion 18

Conceptos básicos XML...

- Muchas formas de presentación a partir de un documento XML:

The diagram shows an 'XML' box on the left. An arrow points from it to a vertical bar labeled 'PROCESAMIENTO'. From the top of this bar, an arrow points to a stack of output format boxes: 'HTML', 'VML', 'RTF', 'PDF', and 'TXT'.

XML Laboratorio de Programacion 19

Ejemplos (1)

- Un Ejemplo de HTML:


```
<B>Deepak Chopra</b>
<br><B><i>El sendero del Mago</i></B></i>
<BR>precio: <b>$ 80</b>
```
- El mismo Ejemplo en XML:


```
<libro>
<autor>Deepak Chopra</autor>
<titulo>El sendero del Mago</titulo>
<precio moneda="pesos">80</precio>
</libro>
```

XML Laboratorio de Programacion 20

XML

Trabajar con XML

Qué es un archivo XML?

- Un archivo de texto ASCII
- Un archivo con extensión ".xml"
- Un archivo cuyo contenido sigue las reglas sintácticas de XML

The diagram shows a box labeled 'XML' with an arrow pointing to a box labeled 'libro.xml'.

XML Laboratorio de Programacion 22

Herramientas necesarias

- Un Editor de Texto
- Un Explorador o Navegador de Internet
 - Internet Explorer
 - Mozilla
 - Opera
 - Netscape, etc
- Un herramienta de procesamiento y transformación para XML o parser
 - XML CopyEditor**
 - Xerces
 - MSXML (Microsoft), etc

XML Laboratorio de Programacion 23

Herramientas necesarias

Parser:
Analizador sintáctico

- Procesa el contenido de un archivo XML para:
 - Validar
 - Transformar
- Pueden incluir validación o no
- Pueden realizar transformaciones o no
- Pueden exponer la información de diferentes formas (DOM, SAX)
- Existen para la mayoría de lenguajes y plataformas de desarrollo (VB,Php,Perl,Java, etc.)

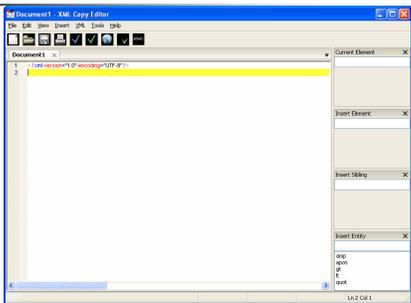
XML Laboratorio de Programacion 24

Herramientas necesarias

- Usaremos el parser de XML Copy Editor
 - Realiza validaciones contra DTD
 - Realiza transformaciones con soporte XSL
 - Gratuito y redistribuible libremente

XML Laboratorio de Programacion 25

XML Copy Editor



XML Laboratorio de Programacion 26

Ejemplo

```
<?xml version="1.0">
<libro>
  <autor>Deepak Chopra</autor>
  <titulo>El sendero del Mago</titulo>
  <precio moneda="pesos">30.000</precio>
</libro>
```

XML Laboratorio de Programacion 27

Tipos de documento XML

- Hay dos tipos de documentos XML:
 - Bien formados (*well formed*)
 - Aquellos que cumplen con las especificaciones del lenguaje respecto a las **reglas sintácticas**
 - Una condición básica para trabajar con un documento XML es que esté bien formado
 - Válidos
 - Aquellos que cumplan con la estructura y una semántica determinada por alguna regla de validación, como:
 - DTD
 - XML Schema

XML Laboratorio de Programacion 28

Reglas sintácticas XML

- Elemento** y Contenido

Contenido del Elemento

```
<autor>Deepak Chopra</autor>
```

Nombre del Elemento Etiqueta fin del Elemento

 - Siempre comienzan con una <etiqueta>
 - Siempre termina con una </etiqueta> del mismo nombre
 - Elemento vacío <etiqueta/>
 - Es "case-sensitive", es decir <autor> no es igual a <Autor>
 - No se permiten espacios en blanco ni saltos de línea en el nombre

XML Laboratorio de Programacion 29

Reglas sintácticas XML

Cómo escribir el **Prólogo**?

- El prólogo es opcional
- La primera línea debe comenzar con `<?xml` y terminar con `?>`.
- La primera línea especifica la **versión** y la **codificación** de caracteres del documento:
 - UTF-8 (por defecto) de 7 bits
 - ISO-8859-1 de 8 bits (caracteres no estándar ASCII)
- La primer línea también indica si es del tipo standalone
 - yes => El documento no necesita ningún otro (p.e. DTD)
 - no (por defecto) => El documento requiere algún otro
- La segunda línea especifica documentos asociados (DTD, Schema, XSL)

XML Laboratorio de Programacion 36

Ejemplos de declaraciones XML

- `<?xml version="1.0" ?>`
- `<?xml version="1.0" encoding="ISO-8859-7"?>`
- `<?xml version="1.0" encoding="UTF-16" standalone="yes" ?>`
- `<?xml version="1.0" encoding="UTF-16" standalone="no" ?>`
`<!DOCTYPE clima SYSTEM "clima.dtd">`
- `<?xml version="1.0">`
`<?xml:stylesheet type="text/xsl" href="prueba.xls"?>`

XML Laboratorio de Programacion 37

Reglas sintácticas XML

- Estructura

XML Laboratorio de Programacion 38

Reglas sintácticas XML

El cuerpo de un archivo XML tiene una estructura **jerárquica**:

- Estructura de árbol correctamente anidados
- No se pueden superponer elementos
- Debe haber un elemento raíz por documento XML
- Todas las etiquetas deben cerrarse

XML Laboratorio de Programacion 39

Reglas sintácticas XML

Ejemplos:

- Incorrectamente anidado


```
<libro>
<autor>Deepak Chopra
<titulo>El sendero del Mago</autor></libro></titulo>
```
- Correctamente anidado


```
<libro>
<autor>Deepak Chopra</autor>
<titulo>El sendero del Mago</titulo>
</libro>
```

XML Laboratorio de Programacion 40

Reglas sintácticas XML

- Estructura Jerárquica de Árbol

```
<libro>
<autor>Deepak Chopra</autor>
<titulo>El sendero del Mago</titulo>
<precio moneda="pesos">80</precio>
</libro>
```

XML Laboratorio de Programacion 41

Comprobación: XML bien formado

- Uso un analizador sintáctico (parser)
 - Por ejemplo:
 - Internet Explorer
 - Aplicación para editar XML (por ejemplo XML Spy)
 - Analizadores en línea (w3c)
 - XML Copy Editor

XML Laboratorio de Programacion 42

Otro ejemplo

```

<pizzas>
<pizza nombre="Napolitana" precio="25">
<ingrediente nombre="Tomate" />
<ingrediente nombre="Queso" />
</pizza>
</pizzas>

```

Bien formado

```

<pizzas>
<pizza nombre="Margarita" precio="6">
<ingrediente nombre="Tomate" >
</pizzas>

```

Mal formado

XML Laboratorio de Programacion 43

Documentos XML válidos

- Técnicas de validación
 - Qué se valida?
 - La estructura: nombres y valores de los elementos, atributos y entidades
 - Para qué validar?
 - Consistencia de datos, compartir datos, uso de estándares.

XML Laboratorio de Programacion 44

Técnicas de validación

- Existen 2 formas de definir los elementos que contiene un documento XML a través de reglas gramaticales de los elementos, atributos y entidades:
 - **DTD (Document Type Definition)**
 - Archivos con extensión .dtd
 - **XML Schema**
 - Archivos con extensión .xsd

XML Laboratorio de Programacion 45

Técnicas de validación

- Ambas formas nos permiten crear nuestro propio lenguaje de marcado.
- Ambas formas pueden residir en un archivo externo y ser compartidos por varios documentos XML.
- Un XML que se ajusta a cualquier técnica es un "XML válido".
- Ambas técnicas son opcionales

XML Laboratorio de Programacion 46

DTD

Document Type Definition

DTD

- Permite especificar la estructura del documento XML
- Pueden expresar algunas limitaciones para combinarlos.
- Los documentos XML que se ajustan a su DTD son VALIDOS
- Una DTD describe
 - Elementos:** indican que etiquetas son permitidas
 - Estructura:** indica el orden en que van las etiquetas
 - Anidamiento:** indica que etiquetas van dentro de otra

XML Laboratorio de Programacion 48

Ejemplo DTD

- Un ejemplo de DTD: libro.dtd

```
<!ELEMENT libro (autor, titulo, isbn, editorial, sumario, precio, otro?)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT sumario (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ATTLIST precio moneda CDATA #REQUIRED>
<!ELEMENT otro (#PCDATA)>
```

XML Laboratorio de Programacion 49

DTD

- Uso Externo


```
<?xml version="1.0"?>
<!DOCTYPE libro SYSTEM "libro.dtd">
```
- Uso Interno


```
<?xml version="1.0"?>
<!ELEMENT libro (autor, titulo, isbn, editorial, sumario, precio, otro?)>
<!ELEMENT autor (#PCDATA)>
...
<!ELEMENT otro (#PCDATA)>
</libro>
...
```

XML Laboratorio de Programacion 50

Declaración de elemento ELEMENT

<!ELEMENT	Apertura y palabra clave.
Nombre_elemento	Nombre del elemento.
(. . .)	Modelo de contenido o contenido declarado.
PALABRACLAVE	Cierre.
>	

Palabras clave para el contenido declarado:

EMPTY	Sin elemento o contenido.
ANY	Cualquier combinación de elementos descendientes y datos caracter. (valor por defecto)

XML Laboratorio de Programacion 51

Contenido de elementos

(?)	= 0, 1 elemento
(*)	= 0 ó más elementos
(+)	= 1 ó más elementos
()	= alternativa
(.)	= secuencia
EMPTY	= vacío
ANY	= cualquier estructura de subelementos
#PCDATA	= cadena de caracteres analizados

```
<!ELEMENT pizza (ingrediente*, inventor?)>
<!ELEMENT servicio (domicilio | restaurante) >
<!ELEMENT ingrediente EMPTY>
<!ELEMENT inventor (#PCDATA)>
<!ELEMENT sección (titulo, (contenido | sección+))>
```

Recurividad

XML Laboratorio de Programacion 52

Contenido mixto

- Contenido que puede mezclar texto con elementos
- Ejemplo:

```
<!ELEMENT comentario (#PCDATA|enlace|persona) >
<!ELEMENT enlace (#PCDATA)>
<!ELEMENT persona (#PCDATA)>
<!ATTLIST enlace url CDATA #REQUIRED>
```

```
<comentario>
Este texto está relacionado con el <enlace
url="http://www.quijote.com">Quijote</enlace> y ha sido realizado por
<persona>Leopoldo Alas</persona>
</comentario>
```

XML Laboratorio de Programacion 53

Atributos

Consisten en un nombre del atributo y un valor asociado

Tipos de datos		Valor de los Atributos	
CDATA = Cadena de caracteres	#REQUIRED Obligatorio	#REQUIRED Obligatorio	
NMTOKEN = Palabra (sin espacios)	#IMPLIED Opcional	#IMPLIED Opcional	
NMTOKENS = Lista de palabras	#FIXED Constante	#FIXED Constante	
Enumeración separada por	Valor Valor por defecto	Valor Valor por defecto	

```

<!-- Atributos de ejemplo -->
<!-- pizza nombre CDATA #REQUIRED -->
<!-- ingrediente nombre CDATA #REQUIRED -->
<!-- precio moneda (euros) #REQUIRED -->
<!-- canOrigen (litro) #FIXED -->
<!-- impuesto tipo CDATA #FIXED "IVA" -->

<!-- Ejemplo de uso -->
<!-- pizza nombre="4 estaciones" -->
<!-- ingrediente nombre="Jamón" -->
<!-- precio moneda="euros" valor="7" -->
<!-- canOrigen="1" -->
<!-- impuesto="IVA" -->

```

XML Laboratorio de Programacion 54

Valores ID e IDREF

- ID = Nombre único (sin duplicados)
- IDREF = Su valor debe apuntar a un ID

```

<!-- Atributos de ejemplo -->
<!-- persona código ID #REQUIRED -->
<!-- dueño código IDREF #REQUIRED -->

<!-- Ejemplo de uso -->
<!-- persona código="23" nombre="Juan" -->
<!-- persona código="35" nombre="Pepe" -->
<!-- persona código="37" nombre="Luis" -->
<!-- dueño código="35" -->

```

XML Laboratorio de Programacion 55

Declaración de documento DOCTYPE en XML

```

<!DOCTYPE
Elemento_raiz
PALABRACLAVE
"dtd.dtd"
[ ... ]
>

```

Apertura y palabra clave. Nombre del elemento raíz. SYSTEM o PUBLIC y una DTD XML externa o declaraciones internas. Cierre.

XML Laboratorio de Programacion 56

Ejemplos de declaraciones de documento

- Declarado en el mismo documento XML:

```

<!DOCTYPE novela [
<!-- ELEMENT novela (titulo, parrafo+) -->
<!-- ELEMENT titulo (#PCDATA) -->
<!-- ELEMENT parrafo (#PCDATA) -->
]>

```
- Declarado fuera del documento XML:

```

<!DOCTYPE novela SYSTEM "novela.dtd" >

```

XML Laboratorio de Programacion 57

Ejemplo completo (XML+DTD)

DTD incluida con el XML.

```

<?xml version="1.0"?>
<!DOCTYPE clima [
<!-- Nombre Modelo contenido -->
<!-- ELEMENT clima (ciudad+) -->
<!-- ELEMENT ciudad (nombre, reporte) -->
<!-- ELEMENT nombre (#PCDATA) -->
<!-- ELEMENT reporte (alta, baja, precip?) -->
<!-- ELEMENT alta (#PCDATA) -->
<!-- ELEMENT baja (#PCDATA) -->
<!-- ELEMENT precip EMPTY -->
<!-- ATRIBUTO precip total_dia CDATA #REQUIRED -->
<!-- ATRIBUTO tipo (lluvia | nieve) #IMPLIED -->
]

```

Código XML.

```

<ciudad>
<nombre>Santa Cruz</nombre>
<reporte>
<alta>20</alta>
<baja>-5</baja>
<precip total_dia="0" tipo="lluvia" fuerza="ligera"/>
</reporte>
</ciudad>
<ciudad>
<nombre>Clubot</nombre>
<reporte>
<alta>25</alta>
<baja>5</baja>
<precip total_dia="0" tipo="lluvia" fuerza="fuerte"/>
</reporte>
</ciudad>
</clima>

```

2 elementos de segundo nivel de ejemplo.

XML Laboratorio de Programacion 58

Ejemplo completo (XML+DTD)

```

...
<!DOCTYPE clima [
<!-- Nombre Modelo contenido -->
<!-- ELEMENT clima (ciudad+) -->
<!-- ELEMENT ciudad (nombre, reporte) -->
<!-- ELEMENT nombre (#PCDATA) -->
<!-- ELEMENT reporte (alta, baja, precip?) -->
<!-- ELEMENT alta (#PCDATA) -->
<!-- ELEMENT baja (#PCDATA) -->
<!-- ELEMENT precip EMPTY -->
<!-- ATRIBUTO precip total_dia CDATA #REQUIRED -->
<!-- ATRIBUTO tipo (lluvia | nieve) "lluvia" -->
<!-- ATRIBUTO fuerza (ligera | fuerte) #IMPLIED -->
]
...

```

XML Laboratorio de Programacion 59

Ejemplo completo (XML+DTD).

```

...
<clima>
  <ciudad>
    <nombre>Santa Cruz</nombre>
    <reporte>
      <alta>20</alta>
      <baja>5</baja>
      <precip total_dia="0" tipo="lluvia" fuerza="ligera"/>
    </reporte>
  </ciudad>
  <ciudad>
    <nombre>Chubut</nombre>
    <reporte>
      <alta>25</alta>
      <baja>5</baja>
      <precip total_dia="0" tipo="lluvia" fuerza="fuerte"/>
    </reporte>
  </ciudad>
</clima>

```

clima (ciudad+)
 ciudad (nombre, reporte)
 nombre (#PCDATA)
 reporte (alta, baja, precip?)
 alta (#PCDATA)
 baja (#PCDATA)
 precip EMPTY
 precip total_dia CDATA #REQUIRED
 tipo (lluvia | nieve) "lluvia"
 fuerza (ligera | fuerte) #IMPLIED

XML Laboratorio de Programacion 60

Otro ejemplo...

```

<?xml version="1.0"?>
<!DOCTYPE poema SYSTEM "poema.dtd">
<poema fecha="Abril de 1915">
  lugar="Granada">
    <titulo>Alba</titulo>
    <verso>Mi corazón oprimido</verso>
    <verso>siente junto a la alborada</verso>
    <verso>el dolor de sus amores</verso>
    <verso>y el sueño de las distancias.
  </verso>
</poema>

```

POEMA.DTD
 <!ELEMENT poema (titulo,verso)*>
 <!ELEMENT titulo (#PCDATA)>
 <!ELEMENT verso (#PCDATA)>
 <!ATTLIST poema fecha CDATA #REQUIRED
 lugar CDATA #IMPLIED>

Documento válido
 Está bien formado + Estructura encaja con DTD

XML Laboratorio de Programacion 61

Ejercicio: Construir documento XML y su DTD

- Analizar documento estructurado.
- Diseñar los elementos de información.
- Definir la estructura subyacente.
- Crear un documento XML correspondiente.
- Crear una representación de la estructura del documento en forma de un DTD.

XML Laboratorio de Programacion 62

Ejercicio...

```

graph TD
  Personas --> nombre
  Personas --> nacimiento
  Personas --> direccion
  Personas --> varon
  nombre --> JuanPerez["Juan Perez"]
  nacimiento --> dia["dia: 5"]
  nacimiento --> mes["mes: Mayo"]
  nacimiento --> anio["anio: 1980"]
  direccion --> calle
  direccion --> ciudad
  direccion --> provincia
  direccion --> cpostal
  calle --> AvRoca123["Av Roca 123"]
  ciudad --> RioGallegos
  provincia --> SantaCruz["Santa Cruz"]
  cpostal --> 9400
  
```

Nodos elementos
 Nodos hojas (contenido)
 Nodos de atributos

XML Laboratorio de Programacion 63

Consultas...

¿?

XML Laboratorio de Programacion 64

Próxima clase

- Repaso
- XML Schemas
- DOM
- Ejemplos y ejercitación

XML Laboratorio de Programacion 65