



# **JAVA WEB SERVICES**

## **eXtensible Markup Language**

**Eric Gustavo Coronel Castillo**  
**[www.desarrollasoftware.com](http://www.desarrollasoftware.com)**  
**[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)**



## Temas

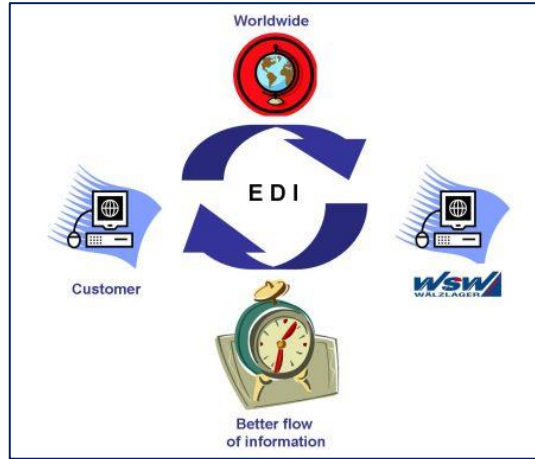
- INTRODUCCIÓN
- XML
- DTD – DOCUMENT TYPE DEFINITION
- XML NAMESPACES





# INTRODUCCIÓN

## Tecnologías de Intercambio de Datos

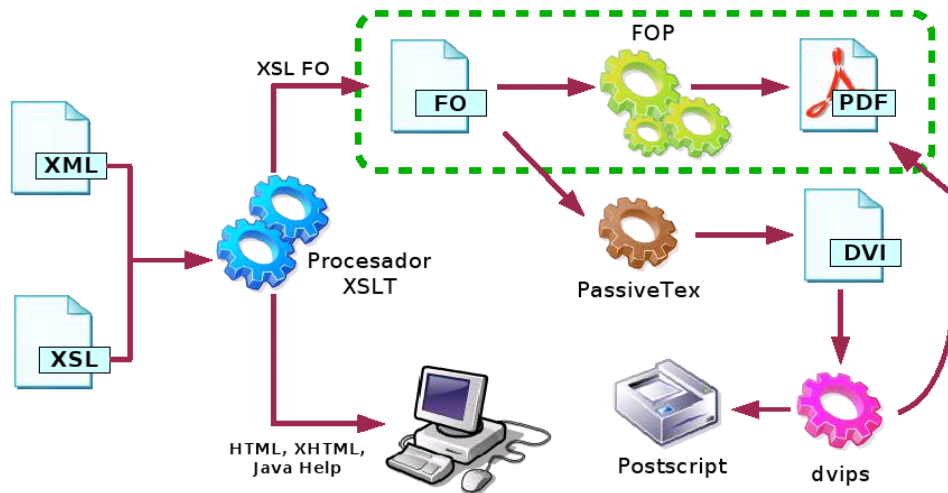


Es el intercambio electrónico de datos de computadora a computadora entre Socios Comerciales (B2B), Consumidores potenciales (B2C) todo esto para el soporte de ERP, CRM, BPM con la finalidad de ahorrar tiempo al eliminar los tradicionales métodos de preparación y envío de documentos a través de mensajería.



# INTRODUCCIÓN

## Tecnologías de Intercambio de Datos

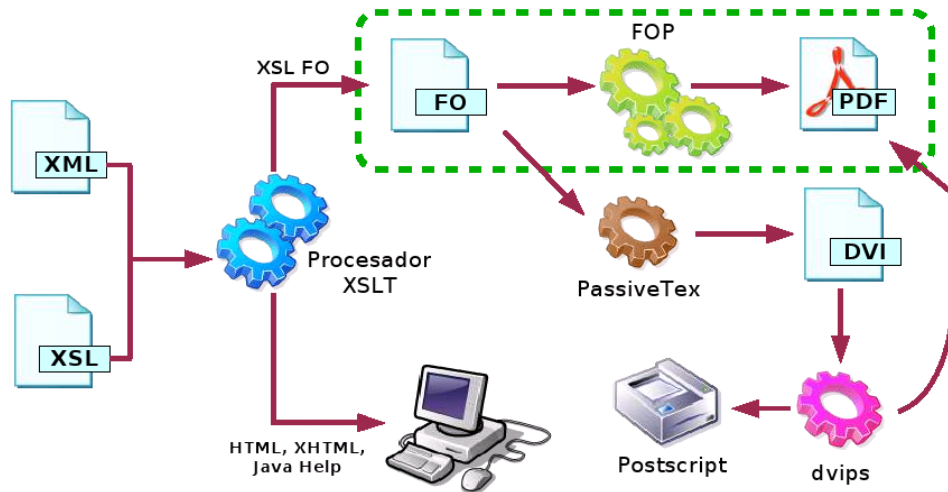


XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.



# INTRODUCCIÓN

## Tecnologías de Intercambio de Datos

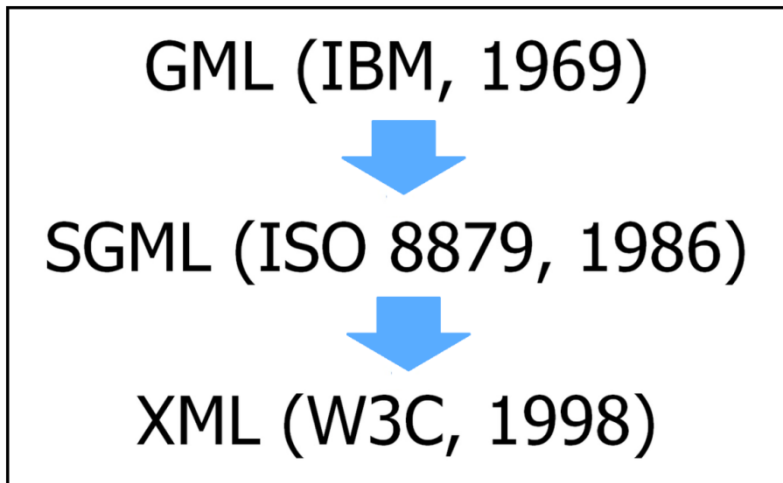


En la actualidad todo intercambio de información implementa XML, entonces podemos afirmar que los sistemas B2B, B2C, ERP, CRM, BPM y muchos sistemas de información de cualquier plataforma y de cualquier lenguaje usan XML.



# INTRODUCCIÓN

## Introducción a XML



### GML - Generalized Markup Language

- Inventado por IBM.

### SQXML - Standard Generalized Markup Language

- Normalizado por ISO en 1996.

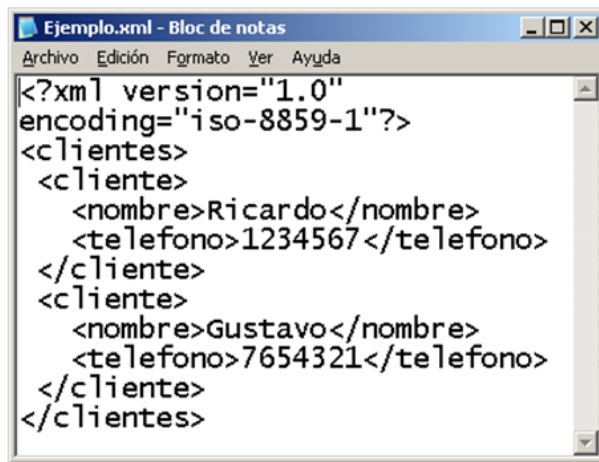
### XML - eXtensible Markup Language

- Creado por la W3C a partir de SGML.
- Aprobado en febrero de 1998.
- Es un método para introducir datos estructurados en un fichero de texto.
- Se parece al HTML pero no es HTML.
- XML es gratis, independiente de la plataforma y ampliamente distribuida.

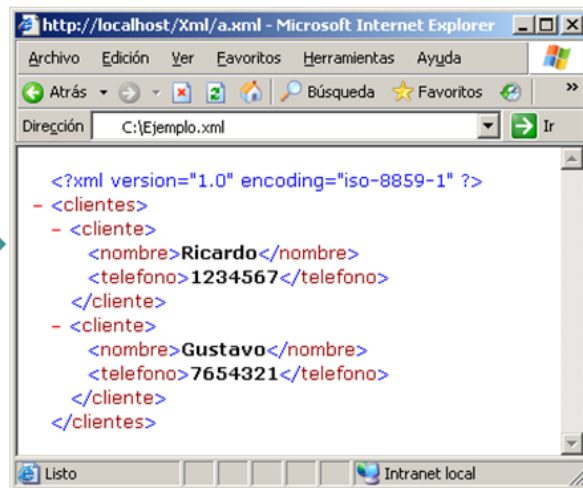


# INTRODUCCIÓN

## Introducción a XML



```
<?xml version="1.0"
encoding="iso-8859-1"?>
<clientes>
  <cliente>
    <nombre>Ricardo</nombre>
    <telefono>1234567</telefono>
  </cliente>
  <cliente>
    <nombre>Gustavo</nombre>
    <telefono>7654321</telefono>
  </cliente>
</clientes>
```



```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <clientes>
  - <cliente>
    <nombre>Ricardo</nombre>
    <telefono>1234567</telefono>
  </cliente>
  - <cliente>
    <nombre>Gustavo</nombre>
    <telefono>7654321</telefono>
  </cliente>
</clientes>
```

Es un lenguaje para definir estructuras de los datos y permite el intercambio de datos estandarizado, no es un lenguaje que remplace a HTML, porque en XML el usuario define sus propias etiquetas.



# INTRODUCCIÓN

## Rol del W3C



W3C: Es una asociación internacional formada por organizaciones miembro del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web, cuyo propósito es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web.

El futuro expuesto por la W3C es: Web para todos, en cualquier cosa, en cualquier lugar, navegar a través de la vista, el oído, la voz y el tacto.

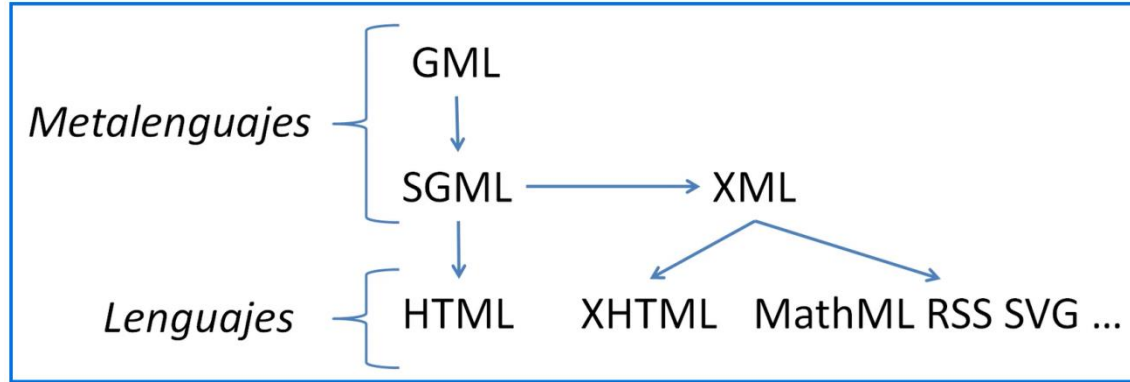
Tim Berners-Lee, quien inventó la World Wide Web en 1989 fundó junto con otros el W3C en 1994.





# XML

## ¿Qué es XML?



- XML es un meta-lenguaje de marcas, es decir, permite que el usuario diseñe sus propias marcas (tags) y les dé un significado, con tal de que siga un modelo coherente.
- XML define un conjunto de reglas para que cualquiera pueda definir su propio conjunto de etiquetas y atributos, y las relaciones que existen entre esas etiquetas.
- El conjunto de reglas del lenguaje es la gramática del lenguaje.



# XML

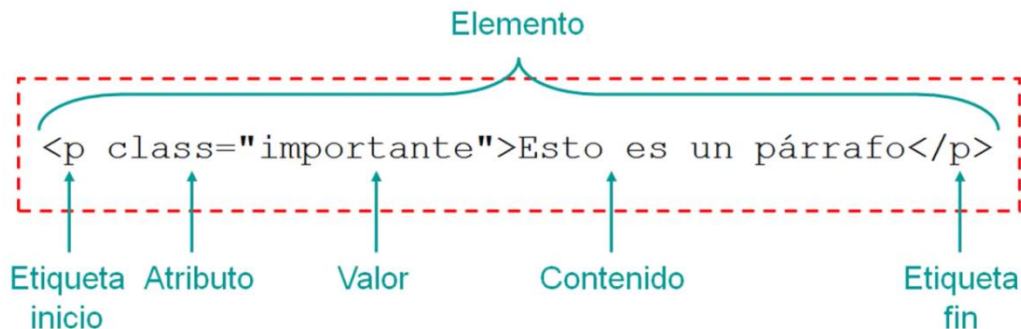
## ¿Qué es XML?

```
<!-- ejm01.xml -->
<Catalogo>
  <Articulo>
    <Nombre>Monitor</Nombre>
    <Precio>120.00</Precio>
  </Articulo>
  <Articulo>
    <Nombre>Teclado</Nombre>
    <Precio>24.00</Precio>
  </Articulo>
  <Articulo>
    <Nombre>Mouse</Nombre>
    <Precio>18.00</Precio>
  </Articulo>
</Catalogo>
```



# XML

## Sintaxis XML



- XML usa etiquetas para definir el contexto de los datos
- La unidad básica de información es el elemento
- Nombres de los elementos
  - ✓ El primer carácter debe ser: [A-Z] [a-z] " "
  - ✓ El resto de caracteres: [A-Z] [a-z] [0-9] "-\_".
- XML es sensible a las mayúsculas
- Elementos vacíos
  - `<Libro></Libro>`
  - `<Libro/>`



# XML

## Documento XML

Instrucción de Procesamiento

```
<?xml version="1.0"?>
```

Comentario

```
<!-- Esto es un comentario -->
```

Elemento raíz

```
<empleado>
```

Elemento hijo

```
    <nombre>Juan Perez</nombre>
```

Elemento vacío

```
    <cargo />
```

Elemento con atributo

```
    <sueldo moneda="Soles">  
        3500
```

Fin de elemento raíz

```
    </sueldo>
```

```
</empleado>
```



# XML

## Prologo

- Los documentos XML deben empezar con unas líneas que describen la versión de XML, el tipo de documento, y otras cosas.
- Ejemplo 1

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mensaje SYSTEM "mensaje.dtd">
```
- Ejemplo 2

```
<?xml version="1.0" encoding="UTF-7"?>
<!DOCTYPE LABEL SYSTEM "http://www.empresa.com/dtds/label.dtd">
```



# XML

## Documento XML bien Formados

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ejm06.xml -->
```

Prólogo

```
<direccion>
  <nombre>
    <titulo>Mrs.</titulo>
    <nombres>Mary</nombres>
    <apellidos>McGoon</apellidos>
  </nombre>
  <calle>1401 Main Street</calle>
  <ciudad>Anytown</ciudad>
  <estado>NC</estado>
  <codigo-postal>34829</codigo-postal>
</direccion>
```

Cuerpo del

Documento

- Un documento XML esta bien formado si cumple con las reglas de sintaxis.
- Por ejemplo: solo una etiqueta raíz, todas las etiquetas debes cerrarse, etc.



# XML

## Documento XML Validos

```
<!-- ejm05.dtd -->
<!ELEMENT direccion (nombre, calle, ciudad,
    estado, codigo-postal)>
<!ELEMENT nombre (titulo, nombres, apellidos)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT nombres (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT codigo-postal (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ejm06.xml -->
<!DOCTYPE direccion SYSTEM "ejm05.dtd">
```

Prólogo

```
<direccion>
  <nombre>
    <titulo>Mrs.</titulo>
    <nombres>Mary</nombres>
    <apellidos>McGoon</apellidos>
  </nombre>
  <calle>1401 Main Street</calle>
  <ciudad>Anytown</ciudad>
  <estado>NC</estado>
  <codigo-postal>34829</codigo-postal>
</direccion>
```

Cuerpo del  
Documento

- Un documento XML es válido cuando, además de no tener errores de sintaxis, no incumple ninguna de las normas establecidas en su estructura.
- La estructura se puede definir utilizando distintos métodos, tales como: DTD, XML Schema, RelaxNG



# DTD - DOCUMENT TYPE DEFINITION

## Definición

```
<!-- ejm05.dtd -->
<!ELEMENT direccion (nombre, calle, ciudad,
    estado, codigo-postal)>
<!ELEMENT nombre (titulo, nombres, apellidos)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT nombres (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT codigo-postal (#PCDATA)>
```

- Describe la estructura y la sintaxis de un documento XML, es decir, de su gramática.
- Un problema de los DTD es que no es XML, este problema se resuelve con XML Schema.





# DTD - DOCUMENT TYPE DEFINITION

## Función

```
<!-- ejm05.dtd -->
<!ELEMENT direccion (nombre, calle, ciudad,
    estado, codigo-postal)>
<!ELEMENT nombre (titulo, nombres, apellidos)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT nombres (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT codigo-postal (#PCDATA)>
```

- La función del DTD es describir la estructura de un documento XML, con el fin de mantener una estructura común y la consistencia entre todos los documentos que utilicen el misma DTD.
- El DTD permite de esta manera validar los documentos XML, puede comprobarse que son correctos, y pueden ser compartidos entre diferentes usuarios.
- Esta es la dirección para consultar los DTDs de XHTML:  
<https://www.w3.org/TR/xhtml1/dtds.html>



# DTD - DOCUMENT TYPE DEFINITION

## Uso de un DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- personal.dtd -->
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

- Un DTD define los elementos que son permitidos y el contenido de dichos elementos.
- También define los atributos que se pueden utilizar en cada elemento, así como también cuales son los valores validos a los atributos.
- Un DTD define la estructura valida de los elementos que pueden aparecer en un documento XML, el orden en el cual pueden aparecer, cómo pueden estar anidados y otros detalles de la estructura del documento XML.



# DTD - DOCUMENT TYPE DEFINITION

## Analizando un ejemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- personal.dtd -->
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

- **lista\_de\_personas** es un nombre de elemento válido. El "\*" indica que puede haber 0 o más elementos de persona.
- **persona** es un nombre de elemento válido. Éste contiene obligatoriamente el elemento nombre mientras que el resto son opcionales. Y lo son porque lo indica el símbolo "?".



# DTD - DOCUMENT TYPE DEFINITION

## DTD Interno

```
<!DOCTYPE Mensaje
[
    <!ELEMENT Mensaje (Frase,Arenga)>
    <!ELEMENT Frase (#PCDATA)>
    <!ELEMENT Arenga (#PCDATA)>
]
>
<!-- este es un comentario -->
<Mensaje>
    <Frase>Somos PERÚ</Frase>
    <Arenga>PERU es el campeón</Arenga>
</Mensaje>
```

- El DTD se encuentra en el propio documento XML.
- Para realizar la validación no hay que acudir a ningún documento externo.
- Se utiliza la declaración `<!DOCTYPE >`



# DTD - DOCUMENT TYPE DEFINITION

## DTD Externo

- En este caso, el DTD está en un archivo aparte, que es lo más lógico, porque es normal que se utilice el mismo DTD para crear varios documentos XML.
- A continuación se tiene la sintaxis para usar DTD externos:

- PUBLIC: global, un estándar

```
<!DOCTYPE elementoRaiz PUBLIC "nombreDTD">
```

```
<!DOCTYPE elementoRaiz PUBLIC "nombreDTD" "urlDTD">
```

- SYSTEM: local, definido por el usuario (privado)

```
<!DOCTYPE elementoRaiz SYSTEM "urlDTD">
```

- A continuación se tiene el ejemplo de uso de un DTD público que se utiliza en una página web:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- El siguiente ejemplo ilustra el uso de un DTD propio o local, donde el elemento raíz es NOTICIA en mayúsculas:

```
<!DOCTYPE NOTICIA SYSTEM "noticia.dtd">
```



# DTD - DOCUMENT TYPE DEFINITION

## DTD Externo

```
<!-- ejm06.dtd -->
<!ELEMENT direccion (nombre, calle, ciudad,
estado, codigo-postal)>
<!ELEMENT nombre (titulo, nombres, apellidos)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT nombres (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT calle (#PCDATA)>
<!ELEMENT ciudad (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT codigo-postal (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ejm06.xml -->
<!DOCTYPE direccion SYSTEM "ejm06.dtd">
<direccion>
  <nombre>
    <titulo>Mrs.</titulo>
    <nombres>Mary</nombres>
    <apellidos>McGoon</apellidos>
  </nombre>
  <calle>1401 Main Street</calle>
  <ciudad>Anytown</ciudad>
  <estado>NC</estado>
  <codigo-postal>34829</codigo-postal>
</direccion>
```

- Este es un ejemplo ilustrativo del uso de un DTO Externo.



# DTD - DOCUMENT TYPE DEFINITION

## DTD Externo

```
<!-- documento.dtd -->
<!ELEMENT documento (titulo, autor, seccion+)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT seccion (tema,parrafo+)>
<!ELEMENT tema (#PCDATA)>
<!ELEMENT parrafo (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- documento.xml -->
<!DOCTYPE documento SYSTEM "documento.dtd">
<documento>
  <titulo>Fundamentos de XML</titulo>
  <autor>Gustavo Coronel</autor>
  <seccion>
    <tema>Introducción</tema>
    <parrafo>Parrafo 1</parrafo>
    <parrafo>Parrafo 2</parrafo>
  </seccion>
  <seccion>
    <tema>XML Validos</tema>
    <parrafo>Parrafo 1</parrafo>
    <parrafo>Parrafo 2</parrafo>
  </seccion>
</documento>
```

- Este es otro ejemplo ilustrativo del uso de un DTO Externo.



# DTD - DOCUMENT TYPE DEFINITION

## Ejercicio

```
<?xml version="1.0" encoding="UTF-8"?>
<curso>
  <titulo>Spring Framework</titulo>
  <unidad titulo="Spring Core">
    <lecciones>
      <leccion titulo="Introducción">En esta lección ...</leccion>
      <leccion titulo="Uso de XML">En esta lección ...</leccion>
      <leccion titulo="Uso de Anotaciones">En esta lección ...</leccion>
    </lecciones>
  </unidad>
  <unidad titulo="Spring MVC">
    <lecciones>
      <leccion titulo="Configuración">En esta lección ...</leccion>
      <leccion titulo="El Controlador">En esta lección ...</leccion>
      <leccion titulo="AJAX y JSON">En esta lección ...</leccion>
    </lecciones>
  </unidad>
</curso>
```

Definir un DTO para este documento XML.





# XML NAMESPACES

---

## Definición

- XML Namespaces, los espacios de nombre de XML, permite crear distintos espacios de nombres en un documento XML y que, por tanto, los elementos y atributos definidos en distintos módulos no entre en conflicto entre sí.
- En un documento XML donde se emplean múltiples vocabularios de etiquetado se pueden dar problemas de colisión de nombres y, por tanto, de interpretación: dos módulos distintos que emplean los mismos nombres para elementos o atributos.
- La solución a este problema sería que los vocabularios de etiquetado tuviesen nombres universales, cuyo ámbito se extendiese más allá del documento donde están definidos. Este mecanismo es el que proporciona XML Namespaces.
- Los XML NAMESPACES es un método para obtener nombres únicos en los vocabularios de etiquetado.



# XML NAMESPACES

## Caso 1

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:cartas xmlns:e1="http://www.egcc.com/carta1"
  xmlns:e2="http://www.egcc.com/carta2">

  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta>
    <e2:carnes>
      <e2:filete_de_tenera precio="50.00"/>
      <e2:solomillo_a_la_pimienta precio="65.00"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="80.00"/>
      <e2:merluza_en_salsa_verde precio="50.00"/>
    </e2:pescados>
  </e2:carta>

</e1:cartas>
```



# XML NAMESPACES

## Caso 2

```
<h:html xmlns:xdc="http://www.xml.com/books"
        xmlns:h="http://www.w3.org/HTML/1998/html4">
  <h:head><h:title>Revisiones de libros</h:title></h:head>
  <h:body>
    <xdc:bookreview>
      <xdc:title>XML en 10 minutos</xdc:title>
      <h:table>
        <h:tr align="center">
          <h:td>Autor</h:td><h:td>Precio</h:td>
          <h:td>Páginas</h:td><h:td>Fecha</h:td></h:tr>
        <h:tr align="left">
          <h:td><xdc:author>Sergio Luján Mora</xdc:author></h:td>
          <h:td><xdc:price>31.98</xdc:price></h:td>
          <h:td><xdc:pages>352</xdc:pages></h:td>
          <h:td><xdc:date>2003</xdc:date></h:td>
        </h:tr>
      </h:table>
    </xdc:bookreview>
  </h:body>
</h:html>
```



# XML NAMESPACES

## Caso 3

```
<html xmlns="http://www.w3.org/HTML/1998/html4"
      xmlns:xdc="http://www.xml.com/books">
<head><title>Revisiones de libros</title></head>
<body>
  <xdc:bookreview>
    <xdc:title>XML en 10 minutos</xdc:title>
    <table>
      <tr align="center">
        <td>Autor</td><td>Precio</td>
        <td>Páginas</td><td>Fecha</td></tr>
      <tr align="left">
        <td><xdc:author>Sergio Luján Mora</xdc:author></td>
        <td><xdc:price>31.98</xdc:price></td>
        <td><xdc:pages>352</xdc:pages></td>
        <td><xdc:date>2003</xdc:date></td>
      </tr>
    </table>
  </xdc:bookreview>
</body>
</html>
```



# XML NAMESPACES

## Caso 4

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h:form>
      . . .
      <h:commandButton ... />
    </h:form>
  </h:body>
</html>
```





# **JAVA WEB SERVICES**

**Gracias**

**Eric Gustavo Coronel Castillo**  
**[www.desarrollasoftware.com](http://www.desarrollasoftware.com)**  
**[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)**