

XML

ADOLFO SANZ DE DIEGO

MÁSTER UAH

2 ACERCA DE

2.1 AUTOR

- Adolfo Sanz De Diego
 - Blog: asanzdiego.blogspot.com.es
 - Correo: asanzdiego@gmail.com
 - GitHub: github.com/asanzdiego
 - Twitter: twitter.com/asanzdiego
 - LinkedIn: in/asanzdiego
 - SlideShare: slideshare.net/asanzdiego

3 INTRODUCCIÓN A XML

3.1 ¿QUÉ ES?

- XML (Extensible Markup Language) es un metalenguaje que **permite definir lenguajes de marcado.**
- Los lenguajes de marcado permiten describir la estructura de los contenidos de un documento.

3.2 ETIQUETAS

- Un lenguaje de marcado está formado por un conjunto de etiquetas que se encierran entre **corchetes angulares**, `<>`, y se usan en pares.
- Cada par de etiquetas delimita el comienzo y el final de una porción de documento a la que se refiere la etiqueta. Por ejemplo:

```
<asignatura>Bases de datos</asignatura>
```

3.3 EJEMPLO XML

```
1 <banco>
2 <cuanta>
3   <numero_cuenta> C-101</numero_cuenta>
4   <nombre_sucursal> Centro</nombre_sucursal>
5   <saldo>500</saldo>
6 </cuanta>
7 <cuanta>
11 </cuanta>
12 <cuanta>
16 </cuanta>
17 <cliente>
18   <nombre_cliente> González</nombre_cliente>
19   <calle_cliente> Arenal</calle_cliente>
20   <ciudad_cliente> La Granja</ciudad_cliente>
21 </cliente>
22 <cliente>
26 </cliente>
27 <impositor>
28   <numero_cuenta> C-101</numero_cuenta>
29   <nombre_cliente>González</nombre_cliente>
30 </impositor>
31 <impositor>
34 </impositor>
35 <impositor>
38 </impositor>
39 </banco>
```

Ejemplo XML

3.4 VENTAJAS

- Permite que la información esté autodocumentada.
- Formato no rígido pues dispone de la capacidad de reconocer e ignorar nuevas etiquetas.
- Las etiquetas pueden aparecer varias veces facilitando la representación de atributos multivaluados.
- Permite el anidamiento de etiquetas.

3.5 EJEMPLO API

- <http://www.thomas-bayer.com/sqlrest/CUSTOMER/>
- <http://www.thomas-bayer.com/sqlrest/CUSTOMER/3/>

4 ESTRUCTURA BÁSICA

4.1 PROLOGO

- Consta de dos declaraciones:
 - La declaración XML que indica la versión de XML utilizada y el tipo de codificación de caracteres.
 - La declaración de tipo de documento que asocia el documento a una DTD o XSD respecto a la cual el documento es conforme.

4.2 ELEMENTOS

- Es un par de etiquetas de comienzo y final coincidentes que delimita una **porción de información**.

```
<titulo>introducción</titulo>
```

4.3 ELEMENTOS VACÍOS

- Existen elementos vacíos que no contienen contenido.

```
<Nombre etiqueta/>  
<Nombre etiqueta></Nombre etiqueta>
```

4.4 ELEMENTOS ANIDADOS

- Los elementos se pueden anidar:
 - Un texto aparece en el contexto de un elemento si aparece entre la etiqueta de inicio y final de dicho elemento.
 - Las etiquetas se anidan correctamente si toda etiqueta de inicio tiene un única etiqueta de finalización coincidente que está en el contexto del mismo elemento padre.
- Un elemento puede aparecer varias veces en un documento XML.

4.5 EJEMPLO ANIDADO

```
1 <pedido_compra>
2   <identificador> P-101</identificador>
3 <comprador>
4   <nombre> Coyote Loco</nombre>
5   <dirección> Mesa Flat, Route 66, Arizona 12345, EEUU</dirección>
6 </comprador>
7 <proveedor>
8   <nombre> Proveedores Acme SA</nombre>
9   <dirección> 1, Broadway, Nueva York, NY, EEUU</dirección>
10 </proveedor>
11 <lista_elementos>
12 <elemento>
13   <identificador>EAL</identificador>
14   <descripción> Trineo propulsado por cohetes atómicos</descripción>
15   <cantidad>2</cantidad>
16   <precio>199.5</precio>
17 </elemento>
18 <elemento>
19   <identificador>PF2</identificador>
20   <descripción> Pegamento fuerte</descripción>
21   <cantidad>1</cantidad>
22   <precio>29.95</precio>
23 </elemento>
24 </lista_elementos>
25 <coste_total> 429.85</coste_total>
26 <forma_de_pago> Reembolso</forma_de_pago>
27 <forma_de_envio>Avión</forma_de_envio>
28 </pedido_compra>
```

Elementos anidados

4.6 ATRIBUTOS

- Las etiquetas de los elementos pueden incluir 1 o más **atributos** que representan propiedades de los elementos de la forma Nombre atributo="Valor atributo"

```
<cuenta tipo_cuenta="corriente">
```

- Los atributos pueden aparecer solamente una vez en una etiqueta dada.

4.7 MEZCLA

- El texto en un documento XML puede estar mezclado con los subelementos de otro elemento.

```
1 <cuanta>  
2   Esta cuenta se usa muy rara vez, por no decir nunca  
3   <numero_cuenta> C-102</numero_cuenta>  
4   <nombre_sucursal> Navacerrada</nombre_sucursal>  
5   <saldo>400</saldo>  
6 </cuanta>
```

Mezcla texto con subelementos

4.8 RAÍZ

- Todo documento XML tiene **un único elemento raíz** que engloba al resto de elementos del documento.
- En el primer ejemplo el elemento era la raíz.

4.9 COMENTARIOS

- Es un texto que se escribe entre `<!-- y -->`
- La cadena "--" no puede aparecer dentro de un comentario.
- Los comentarios pueden aparecer en cualquier sitio salvo dentro de declaraciones, etiquetas y dentro de otros comentarios.

4.10 ESPACIO DE NOMBRES

- Es un mecanismo que permite especificar nombre únicos globalmente para que se usen como marcas de elementos en los documentos XML.
- Para ello se antepone a la etiqueta o atributo un identificador de recursos universal. En el ejemplo del banco podría ser `http:///www.BancoPrincipal.com`
- Para abreviarlo se declaran abreviaturas del espacio de nombres mediante el atributo `xmlns`

4.11 EJEMPLOS ESPACIO DE NOMBRES

```
1 <banco xmlns:BP="http://www.BancoPrincipal.com">
2   ...
3 <BP:sucursal>
4   <BP:nombre_sucursal> Centro </BP:nombre_sucursal>
5   <BP:ciudad_sucursal> Arganzuela</BP:ciudad_sucursal>
6 </BP:sucursal>
7   ...
8 </banco>
```

Ejemplo espacio de nombres

4.12 VARIOS ESPACIOS DE NOMBRES

- Un documento puede tener más de un espacio de nombres declarado como parte del elemento raíz, de manera que se puede asociar **elementos diferentes con espacios de nombres distintos**.

4.13 ESPACIO DE NOMBRE PREDETERMINADO

- Se puede definir un **espacio de nombres predeterminado** mediante el uso del atributo `xmlns` en el elemento raíz.
- Los elementos sin un prefijo de espacio de nombres explícito pertenecen entonces al espacio de nombres predeterminado.

4.14 CDATA

- A veces es necesario almacenar valores que contienen etiquetas sin que se interpreten como etiquetas XML, es decir como texto normal. Para ello se usa la construcción:

```
<![CDATA]<cuenta>...</cuenta>]]>
```