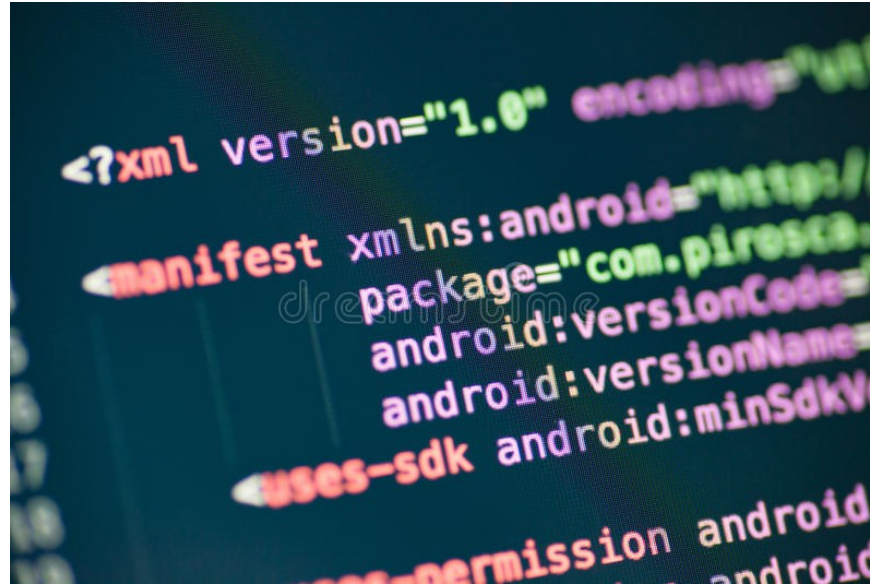


Lenguajes de marcas y sistemas de gestión de la información



UT04 – XML 2 – Namespaces

XML define vocabularios o dialectos

Como hemos visto XML es un metalenguaje, sirve para definir otros dialectos / lenguajes / estructuras / formatos de marcado.

Cada vez que creamos un formato XML para un uso específico estamos creando un dialecto o vocabulario, un conjunto de elementos / atributos que son válidos en cierto contexto, para crear ciertos documentos XML con un objetivo específico.

Cada formato / dialecto XML específico tiene su vocabulario.

Al ser completamente libre la elección de nombres de elementos y atributos, podemos encontrarnos con que dos vocabularios diferentes usan el mismo nombre para elementos o atributos que se refieren a cosas muy distintas.

Ejemplo de nombres coincidentes

Estos XML utilizan un mismo nombre de elemento para cosas distintas:

```
<países>
  <país>
    <nombre>España</nombre>
    <capital>Madrid</capital>
  </país>
</países>
<continente>EUR</continente>
```

```
<inversiones>
  <inversion>
    <capital moneda="euro">
      190000
    </capital>
  </inversion>
</inversiones>
```

Tenemos dos elementos "<capital>" con significados diferentes: uno es el nombre de la capital de un país, y el otro es un importe económico.

Esto no es un problema, hasta que se mezclan dos dialectos en un mismo documento XML

Ejemplo de nombres coincidentes

Otro ejemplo podría ser medios impresos: revistas y libros

```
<libro>
```

```
  <titulo>El código Da  
Vinci</titulo>
```

```
  <autor>Dan Brown</autor>
```

```
  <editorial>Doubleday</editorial>
```

```
  <publicacion a  
</libro>
```

```
<revista>
```

```
  <titulo>National Geographic</titulo>
```

```
  <editorial>National Geographic  
Society</editorial>
```

```
  <edicion numero="230" mes="marzo" año="2024" />  
</revista>
```

Potenciales problemas

Si mezclamos en un documento XML ambos vocabularios, revistas y libros, tendríamos elementos con el mismo nombre.

Al procesar un XML con ambos vocabularios pueden producirse confusiones de nombres.

Usar nombres únicos en los elementos y atributos es imposible, porque:

- ¿Quién gestiona si puedo o no usar un nombre porque ya está usado?
- La cantidad de nombres diferentes sería limitada.
- Si limitamos los nombres diferentes al mismo XML seguimos sin poder combinarlos.

La solución es "desambiguar" los nombres de elementos o atributos que puedan pertenecer a varios vocabularios, usando namespaces.

Namespace o "espacio de nombres"

Los namespaces permiten "calificar" los elementos y atributos, de forma que se concreta a que vocabulario pertenecen.

```
<libros xmlns:lib="http://ejemplo.com/libros"
        xmlns:rev="http://ejemplo.com/revistas">
  <lib:libro>
    <lib:titulo>El código Da Vinci</lib:titulo>
    <lib:autor>Dan Brown</lib:autor>
    <lib:editorial>Doubleday</lib:editorial>
    <lib:publicacion año="2003" />
  </lib:libro>
  <rev:revista>
    <rev:titulo>National Geographic</rev:titulo>
    <rev:editorial>National Geographic Society</rev:editorial>
    <rev:edicion numero="230" mes="marzo" año="2024" />
  </rev:revista>
</libros>
```

Declaración de namespaces

Una declaración de namespace tiene la forma:

```
xmlns:prefijo="URI"
```

El prefijo puede ser cualquier palabra siempre que:

- Sea un nombre válido para elementos, aplican las mismas reglas.
- No empiece con XML. Los nombres que empiezan con XML se reservan para prefijos estandarizados por el W3C.

La URI puede ser cualquier URI sintácticamente correcta. No tiene por qué existir realmente.

Se adoptó el uso de URI por la unicidad de los nombres de dominio, de modo que se puede conseguir fácilmente un valor único para cada espacio de nombres.

Uso de namespaces

Se antepone el prefijo que se ha declarado al nombre del elemento.

```
<prefijo:elemento>Contenido...</prefijo:elemento>
```

El tag de apertura y el de cierre deben contener el prefijo, no basta con ponerlo sólo en el de apertura.

Para atributos, se antepone el prefijo al nombre de un atributo:

```
<elemento prefijo:atributo="valor"></elemento>
```


Alcance y herencia de namespaces

El namespace se puede definir:

- En el elemento raíz del documento. En este caso:
 - El namespace será válido en todo el documento.
 - Se podrá utilizar en cualquier elemento o atributo, incluido el nodo raíz
- En otro elemento del documento, distinto del elemento raíz:
 - El namespace sólo aplicará y podrá utilizarse en los nodos y atributos que descenden de este elemento.

El namespace declarado en un elemento se hereda en todos los elementos descendientes, y puede utilizarse en cualquiera de ellos.

Sobrescritura de namespaces

Si un elemento declara un nuevo namespace con el mismo prefijo que uno definido anteriormente en un ancestro, el nuevo namespace sobrescribe al anterior solo para ese elemento y sus descendientes.

```
<raiz xmlns:info="http://info.com/info">
  <info:informaciones>
    <info:informacion>
      <detalles xmlns:info="http://info.com/details">
        <!-- El siguiente nodo, aunque se llama también
              información, no es el mismo que
                               el anterior información-->
        <info:informacion>Esto es otra cosa...</info:informacion>
      </detalles>
    </info:informacion>
  </info:informaciones>
</raiz>
```

Namespace por defecto

Se puede indicar cual es el namespace que debe aplicar a los elementos que no tienen un prefijo. Para esto se usa xmlns, sin ":prefijo".

```
<raiz xmlns="http://info.com/personas"
      xmlns:mascotas="http://info.com/mascotas">
  <persona edad="22">
    <nombre>Juan</nombre>
    <mascotas:mascota>
      <mascotas:nombre>Lassie</mascotas:nombre>
      <mascotas:raza>Rough Collie</mascotas:raza>
    </mascotas:mascota>
  </persona>
</raiz>
```

Sólo puede haber un namespace por defecto (sin ":prefijo"). Al fin y al cabo, xmlns es un atributo, y no puede repetirse

Aplicación de los namespaces

Se utilizan para poder utilizar en un XML un vocabulario de otro dialecto.

Por ejemplo, usar XSLT (dialecto) dentro de un XML (incompleta):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="libros">
    <html>
      <head><title>Lista de Libros</title></head>
      <body>
        <h1>Lista de Libros</h1>
        <ul><xsl:apply-templates select="libro" /></ul>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Aplicación de los namespaces

Otro ejemplo, definición de vistas en una app Android:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    ...
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        tools:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello,World" />
</LinearLayout>
```

Namespace de uso habitual

Algunos namespaces habituales

Prefijo	URI	Uso
xsl:	http://www.w3.org/1999/XSL/Transform	XSLT
fo:	http://www.w3.org/1999/XSL/Format	XSL formatting objects
xsd:	http://www.w3.org/2001/XMLSchema	Xml Schema Document
dt:	http://www.w3.org/2001/XMLSchema-datatypes	Xml Schema Document (tipos de datos)
svg:	http://www.w3.org/2000/svg	Scalable Vector Graphics
m: / mml:	http://www.w3.org/1998/Math/MathML	Mathematical Markup Language