

XML: Espacios de nombres

Con el XML se pueden crear las etiquetas con el nombre que se quiera en el momento en que sean necesarios. Por lo tanto, si tenemos un documento XML en el que se especifican “habitaciones en alquiler” podríamos tener un documento como este:

```
<alquiler>
  <direccion>
    <calle> Escalante, 23 </calle>
    <ciudad> Sevilla </ciudad>
    <codi_postal> 17600 </codi_postal>
  </direccion>
  <habitacionn>
    <rectanguloangulo>
      <ventanas> 2 </ventanas>
      <puertas> 1 </puertas>
    </rectanguloangulo>
  </habitacionn>
</alquiler>
```

O sea, en el último documento XML estamos especificando que hay una habitación de alquiler en la calle Escalante, 23, de Sevilla, que tiene forma rectánguloangular con dos ventanas y una puerta.

Aunque teóricamente el sistema parece perfecto, en la práctica tiene el problema, ya que el lenguaje humano es limitado y además muchas veces hay palabras con varios sentidos. Esto hace que mucha gente pueda elegir la misma etiqueta para hacer cosas que sean totalmente diferentes. El problema surge cuando hay que mezclar documentos que vienen de fuentes diferentes.

En XML hay un lenguaje estándar que sirve para representar gráficos 2D llamado SVG (Scalable **V**ector **G**raphic). **SVG** sirve para definir gráficos vectoriales, o sea, que en vez de guardar los puntos que forman las imágenes se guarda como se deben dibujar los gráficos utilizando figuras geométricas. Se pueden ver algunos de los elementos de SVG en la tabla.

Etiquetas básicas del formato SVG para representar gráficos vectoriales

etiqueta	uso
svg	La raíz de los documentos SVG.
line	Sirve para definir líneas de un punto a otro.
rect	Se utiliza para definir rectángulos a partir de cuatro puntos.
circle	Con esta etiqueta se definen círculos a partir del punto central y el radio.
ellipse	Nos permite definir elipses a partir del punto central y los dos radios.
polygon	Permite dibujar polígonos a partir de un grupo de puntos.

Por lo tanto, en el ejemplo de “habitaciones en alquiler”, alguien podría decidir que el documento mejoraría si se añade una representación gráfica de la planta de los pisos, y que se puede aprovechar el lenguaje SVG para hacerla.

Como el XML permite mezclar diferentes vocabularios, simplemente se añade el nuevo vocabulario.

Para hacerlo más estructurado, además, se ha definido la etiqueta <imagen>:

```
<alquiler>
  <direccion>
    <calle> Escalante, 23 </calle>
    <ciudad> Sevilla</ciudad>
    <codi_postal> 17600 </codi_postal>
  </direccion>
  <habitacion>
    <rectangulo>
      <ventanas> 2 </ventanas>
      <puertas> 1 </puertas>
    </rectangulo>
  </habitacion>
  <imagen>
    <svg>
      <rectangulo x = "0" y = "0" width = "30" height = "60" />
    </svg>
  </imagen>
</alquiler>
```

Para un lector humano al leerlo no habría ningún problema; rápidamente puede detectar que la imagen está dentro del elemento <imagen>. Pero para un programa, determinar si la etiqueta es del vocabulario original o bien de SVG, es prácticamente imposible. Esto significa que deberíamos tener un **sistema que permita definir a qué vocabulario pertenece cada etiqueta**. Esto es lo que hacen los **espacios de nombres**.

Los espacios de nombres **permiten mezclar lenguajes XML** en el mismo documento y además definir claramente a qué vocabulario pertenece cada etiqueta.

Lo que hacen los espacios de nombres es cambiar los nombres de las etiquetas para que sean únicos. En teoría se podría utilizar cualquier combinación de caracteres, pero como habría el mismo problema que con las etiquetas (depende de quien lo use), generalmente se hace por medio de una **URL (Uniform Resource Locator)**, que en principio es **única**.

Así se podría definir una URL única en nuestro espacio de nombres (<http://www.sevilla/habitacones>) y la URL de SVG (<https://www.w3.org/TR/SVG/>) permitiendo que los elementos y el programa no tengan problemas para determinar qué vocabulario pertenece cada etiqueta:

XML: Espacios de nombres

```
<http://www.sevilla/alquiler:alquiler>
  <http://www.sevilla/alquiler:direccion>
    <http://www.sevilla/alquiler:calle> Escalante, 23 </http://www.sevilla/alquiler:calle>
    <http://www.sevilla/alquiler:ciudad> Sevilla</http://www.sevilla/alquiler:ciudad>
    <http://www.sevilla/alquiler:codi_postal>17600 </http://www.sevilla/alquiler:codi_postal>
  </http://www.sevilla/alquiler:direccion>
  <http://www.sevilla/alquiler:habitacion>
    <http://www.sevilla/alquiler:rectangulo >
      <http://www.sevilla/alquiler:ventanas> 2 </http://www.sevilla/alquiler:ventanas>
      <http://www.sevilla/alquiler:puertas> 1 </http://www.sevilla/alquiler:puertas>
    </http://www.sevilla/alquiler:rectangulo>
  </http://www.sevilla/alquiler:habitacion>
  <http://www.sevilla/alquiler:imagen>
    <http://www.w3.org/2000/svg:svg>
      <http://www.w3.org/2000/svg:rectangulo x = "0" y = "0" width = "30" height = "60"/>
    </http://www.w3.org/2000/svg:svg>
  </http://www.sevilla/alquiler:imagen>
</http://www.sevilla/alquiler:alquiler>
```

Como podemos ver, escribir todas las direcciones hace que se pierda la legibilidad del documento; XML permite definir **alias** para cada una de las URL.

Los alias se definen en los elementos, usando el atributo **xmlns**; un alias se hereda a todo el contenido del elemento:

```
<elemento xmlns: alias = "http:// direccion" />
```

Si se define el atributo en la raíz del documento, se heredarán los alias a todos los elementos del documento:

```
<alquiler xmlns: alquiler = "http://www.sevilla/alquiler"
    xmlns: svg = "http://www.w3.org/2000/svg">
  <alquiler: direccion>
    <alquiler: calle> Escalante, 23 </alquiler: calle>
    <alquiler: ciudad> Sevilla</alquiler: ciudad>
    <alquiler: codi_postal> 17600 </alquiler: codi_postal>
  </alquiler: direccion>
  <alquiler: habitacionn>
    <alquiler: rectangulo>
      <alquiler: ventanas> 2 </alquiler: ventanas>
      <alquiler: puertas> 1 </alquiler: puertas>
    </alquiler: rectangulo>
  </alquiler: habitacionn>
  <alquiler: imagen>
    <svg: svg>
      <svg: rectangulo x = "0" y = "0" width = "30" height = "60" />
    </svg>
  </alquiler: imagen>
</alquiler: alquiler>
```

El atributo ***xmlns*** también permite un espacio de nombres por defecto, es decir, se podrán utilizar sus etiquetas sin especificar un alias. Si algún atributo *xmlns* no define alias se convierte en **el espacio de nombres predeterminado**.

```
<alquiler xmlns = "http://www.sevilla/alquiler"

  xmlns: svg = "http://www.w3.org/2000/svg">

  <direccion>

    <calle> Escalante, 23 </calle>

    <ciudad> Sevilla</ciudad>

    <codi_postal> 17600 </codi_postal>

  </direccion>

  <habitacionn>

    <rectangulo>

      <ventanas> 2 </ventanas>

      <puertas> 1 </puertas>

    </rectangulo>

  </habitacionn>

  <imagen>

    <svg: svg>

      <svg: rectangulo x = "0" y = "0" width = "30" height = "60" />

    </svg: svg>

  </imagen>

</alquiler>
```

XML: Espacios de nombres

Como los espacios de nombres se pueden definir en cualquier elemento, una posibilidad alternativa sería definir los espacios de nombres en las etiquetas adecuadas:

```
<alquiler xmlns = "http://www.sevilla/alquiler" >

  <direccion>

    <calle> Escalante, 23 </calle>

    <ciudad> Sevilla</ciudad>

    <codi_postal> 17600 </codi_postal>

  </direccion>

  <habitacionn>

    <rectangulo>

      <ventanas> 2 </ventanas>

      <puertas> 1 </puertas>

    </rectangulo>

  </habitacionn>

  <imagen>

    <svg xmlns = "http://www.w3c.org/2000/svg">

      <rectangulo x = "0" y = "0" width = "30" height = "60" />

    </svg>

  </imagen>

</alquiler>
```

De esta manera todos los descendientes del elemento <alquiler> siguen su espacio de nombres excepto cuando se llega al elemento <svg>, que cambia el espacio de nombres predeterminado para él y para sus descendientes.