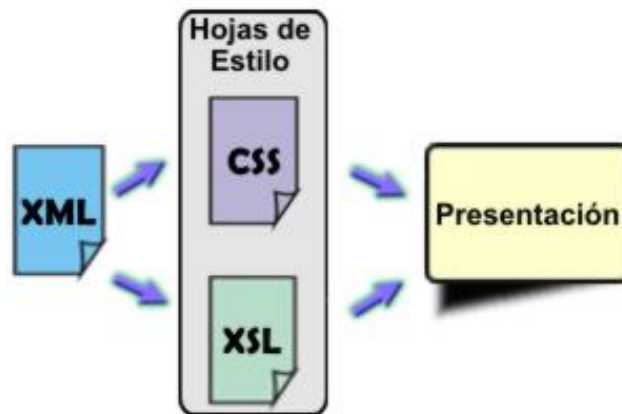


# Capítulo IV

## Lenguajes de estilo



**Lenguajes de Estilo**

**Hojas de estilos**

**Mecanismos de Hojas de estilos previos a XSL**

**Lenguaje de estilo XSL**

**Comparación entre CSS y XSL**

**Transformación XML/XSL en aplicativos web**

XML será el gran protagonista en una próxima generación de la Web, permitiendo a cada comunidad de usuarios crear una estructura de datos propia e integrarla a una infraestructura basada en XML. Sin embargo XML y todas las aplicaciones centradas en él tratan sobre el modo de **organizar y facilitar el intercambio de información**, marcando el contenido de un documento, pero no se ocupan de la apariencia que tomará esa información a la hora de ser mostrada al usuario, sea en un navegador web u otra aplicación.

Esto no es una desventaja de XML, más bien es parte de su poder, ya que un mismo documento XML puede ser utilizado para agregar datos en un sistema de bases de datos, o puede ser manipulado para presentarlo visualmente en un navegador Web, o enviado a un sistema que transforme el texto de XML en audio, o tal vez enviarlo a una impresora.

Para ocuparse de la presentación de XML se utilizan los lenguajes de estilo, destinados a manipular la presentación de documentos basados en etiquetas y cuya aplicación no afecta el sistema de etiquetas de la página, de esta forma la estructura del documento queda separada de su forma de presentación.

Si el uso de XML estuviera enfocado únicamente para ser un medio de intercambio de información estructurada entre empresas u organismos, entonces XML por sí solo sería suficiente, pero por lo general no es así, los usuarios del Web esperan que XML también proporcione un método por medio del cual se pueda presentar esa información de manera atractiva a los lectores, así que para eso existe una tecnología que nace a la par de XML llamada eXtensible Stylesheet Language (Lenguaje de Estilo Extensible) o simplemente XSL. Los lenguajes de estilo permiten definir **Hojas de Estilo**, que son el mecanismo para aplicar estilos de formato a documentos XML.

XSL permite especificar hojas de estilo XSL, que físicamente son archivos que contienen las instrucciones necesarias para que el software que manipula los documentos XML, pueda transformar los datos a un medio no específico, esto es visual o no visual. Las Hojas de Estilos XSL, proveen el mecanismo por medio del cual los documentos XML pueden presentarse con un formato “agradable” en diferentes medios de presentación.

XSL no es el único mecanismo de hojas de estilo existente, existen otros como son las hojas de estilo en cascada (CSS - *Cascading StyleSheet*) y también existe una especificación para aplicar estilos de formato a documentos SGML en la cual se basa XSL, esta es DSSSL (*Lenguaje de semántica y especificación de estilo de documento - Document Style and Semantics Specification Language*).

En este capítulo se abordarán XSL, CSS y DSSSL, enfocándonos en XSL, que es una especificación del W3C y es la que se aplica a documentos XML.

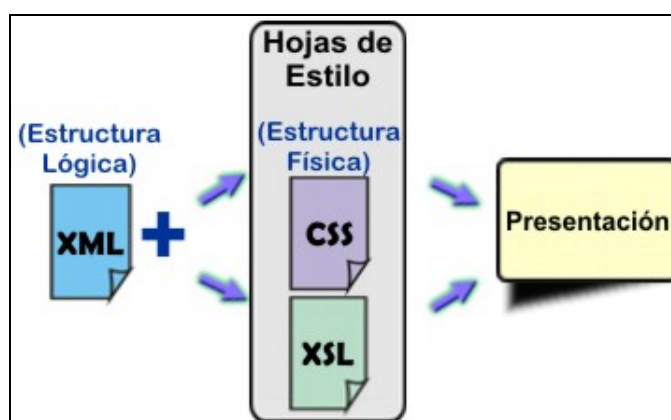
#### 4.1. Hojas de estilo

El lenguaje HTML está limitado a la hora de aplicarle formato a un documento, debido a que fue concebido para otros usos (científicos sobretodo), distintos a los actuales que son mucho más amplios. Para solucionar estos problemas los diseñadores han utilizado ciertas técnicas tales como la utilización de tablas, imágenes, utilización de etiquetas que no son estándares de HTML y otras. Estas "trampas" han causado a menudo problemas en las páginas a la hora de su visualización en distintas plataformas.

Debido a estas causas y con la intención de volver a la idea original de separar el contenido de la presentación y de ofrecer mayores y más fáciles posibilidades de presentación, el W3C empezó a discutir la utilización de Hojas de Estilo en la Web.

Las hojas de estilo vienen a separar en un documento el estilo lógico del estilo físico. El estilo lógico se refiere a la lógica del documento, no se preocupa de la apariencia final, sino de la estructura del documento. Por el contrario, el estilo físico no se preocupa de la estructura del documento, sino por la apariencia final.

Las hojas de estilo tienen dos características fundamentales. Por un lado, permiten un mayor control en el diseño de páginas web y, por otro lado, separan el contenido de la página del código necesario para su presentación. Esto último es muy importante, pues de este modo se puede cambiar la presentación de una página, o grupo de páginas, sin cambiar una sola línea del código XML.



**Figura 4.1.** Aplicación de Hojas de estilos (CSS ó XSL) a documentos XML.

Una hoja de estilo define reglas de diseño que le indican a un navegador (u otra aplicación) como mostrar el contenido de un documento XML. Las reglas de las hojas de estilo se utilizan para traducir la estructura lógica de un documento a una forma que se adapte a la presentación. XML es en esencia, un metalenguaje basado en contenido, por lo que depende totalmente de las hojas de estilo para su visualización.

## **4.2. Mecanismos de hojas de estilo previos a XSL**

### **4.2.1. Lenguaje de semántica y especificación de estilo de documento – DSSSL (Document Style and Semantics Specification Language)**

DSSSL es el lenguaje oficial para dar formato o presentación a los documentos SGML y como tal es extremadamente poderoso, además es un estándar internacional ISO/IEC10179.

DSSSL es mucho más complejo que SGML. Es realmente un lenguaje de programación completo, pero al mismo tiempo es extremadamente complejo. Uno de los problemas de tener un lenguaje complicado es que muy pocos quieren aprenderlo, y por lo tanto, las compañías que se dedican al desarrollo de software saben de antemano que las ventas de ese producto van a estar bajas, ya que la complejidad del mismo lenguaje hace que el software sea difícil de codificar, provocando un software muy caro y baja adopción del lenguaje.

DSSSL incluye la mayoría de las características que son necesitadas por empresas editoriales profesionales. De cualquier forma, DSSSL no ha sido diseñado pensando en los requerimientos del Web, razón por la que XSL adquiere muchas de sus características más importantes, modificando lo necesario para adaptarlo de la mejor manera posible al Web.

Es importante aclarar, que no es parte de esta tesis profundizar en el DSSSL. Lo único relevante es que es un estándar internacional, que por medio de éste, los documentos de SGML tienen formato, y que XSL se basa en éste estándar para su desarrollo.

### **4.2.2. Hojas de estilo en Cascada – CSS (Cascade StyleSheet)**

Aunque XSL (*Lenguaje de Estilo Extensible*) es el estándar actual para la aplicación de formatos a los documentos XML, también se pueden utilizar hojas de estilo CSS.

CSS viene de las siglas "**Cascade StyleSheet**", hojas de estilo en cascada. Las hojas de estilo en cascada son una especificación del W3C que está ya en su segunda versión.

Tiene como objetivo la especificación de un modelo de formato que permite especificar cómo debe visualizarse un documento con etiquetas como contenido.

Formalmente, el lenguaje de Hojas de Estilo en Cascada ha sido definido mediante dos recomendaciones del W3C: la primera versión (CSS1), publicada en octubre de 1996, y CSS2, publicada en mayo de 1998, en tanto se encuentran en desarrollo los trabajos para el lanzamiento de una tercera versión más avanzada y completa del lenguaje.

Las Hojas de Estilo en Cascada o CSS constituyen un mecanismo para asociar estilos de composición a documentos estructurados como HTML o XML. Inicialmente fue desarrollado como el estándar para añadir información de estilo a los documentos HTML. Pero este uso estaba restringido sólo a las etiquetas propias de HTML.

Con XML las CSS pueden aplicar las reglas de estilo a todos los elementos XML. Un documento XML puede estar formateado de muchas diferentes formas y propósitos por lo que sería muy rentable utilizar un sólo documento XML y diferentes posibles formatos dependiendo del uso que se le vaya a dar al documento.

Un rasgo distintivo de las CSS es la posibilidad de ser combinadas entre sí, esta característica es la que da origen a su denominación en cascada, de modo que la composición de la página puede verse afectada por hojas de estilo de distinta procedencia. Como en esta interacción inevitablemente aparecerán conflictos entre las fuentes, existe un orden específico de jerarquías y preferencias.

#### **4.2.2.1. Ventajas**

- Simplifican la edición y mantenimiento de los documentos, con una Hoja de Estilo CSS se puede alterar la presentación de cada elemento sin tocar el código HTML o XML, ahorrando esfuerzo y tiempo de edición. De este modo no sólo simplificamos el mantenimiento del sitio sino que además se reducen las posibilidades de cometer errores.
- La simplicidad, el lenguaje de Hojas de Estilo en cascada es relativamente sencillo y fácil de aprender.
- El rendimiento, los documentos que usan CSS generalmente resultan más compactos. Moviendo los atributos de formato a un archivo externo obtendremos páginas más livianas. Consume poca memoria y tiempo de proceso, pues no construye una representación en árbol del documento.

#### 4.2.2.2. Desventajas

- En cuanto a las desventajas en el uso de las Hojas de Estilo CSS, la de mayor importancia es el soporte irregular que tienen las CSS por parte de los navegadores. Ciertas propiedades que funcionan en un navegador no funcionan en otros, o existen diferencias en un mismo navegador según sea el sistema operativo.
- Sólo permite acceder al contenido de los elementos. No permite manipular un documento como añadir o borrar elementos, acceder a atributos, instrucciones de proceso, realizar ordenaciones, etc.

### 4.3. Lenguaje de Estilo Extensible XSL - (eXtensible Stylesheet Language)

Es lógico pensar que para aprovechar las características de XML falta tener un estándar paralelo y similar asociado a él. De este modo el W3C creó la recomendación XSL.

Según el W3C, XSL es "un lenguaje para transformar documentos XML", así como un vocabulario XML para especificar semántica de formateo de documentos [WWW013].

Las XSL definen o implementan el lenguaje de estilo de los documentos XML, este estándar está basado en DSSSL y, por otro lado, se consideran más potentes que las hojas de estilos en cascada (CSS).

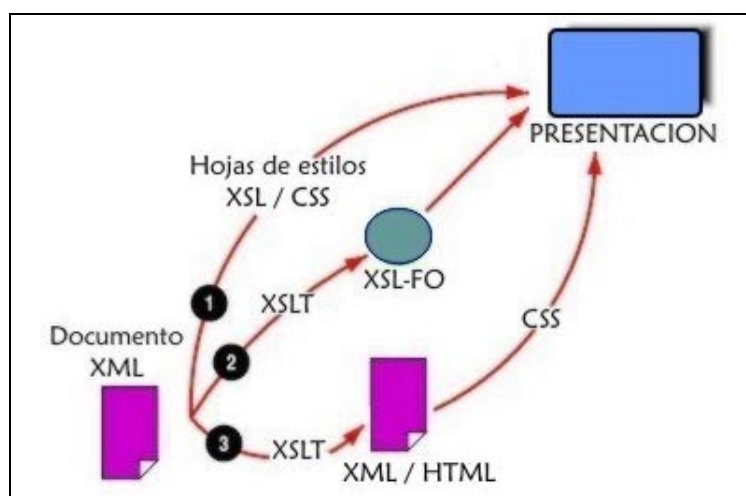
Se espera que las hojas CSS sean usadas para visualizar simples estructuras de documentos XML, ofreciendo una forma rápida de visualización y, por otra parte, las hojas de estilo XSL puedan ser utilizadas donde se requiera más potencia de diseño como documentos XML que encierran datos estructurados (tablas, organigramas, etc.).

El lenguaje de Estilo Extensible (XSL) incluye un lenguaje de transformación (*XSLT- XSL Transformation*) y un lenguaje de formateo (*XSL-FO - XSL Formatting Objects*), en esencia las XSL están conformadas de dos lenguajes, cada uno de éstos, naturalmente, es una aplicación de XML.

El lenguaje de transformación proporciona elementos que definen las reglas de cómo un documento XML es transformado en otro documento XML (ó HTML). El documento XML transformado puede usar las etiquetas y el respectivo DTD del documento original o puede usar un juego completamente diferente de etiquetas. En particular, también puede usar las etiquetas definidas por la segunda parte de XSL, los objetos de formateo (*XSL-FO*).

La transformación y formateo, pueden funcionar independientemente entre sí. El lenguaje de transformación puede transformar un documento XML en un archivo HTML bien formado, e ignorar completamente los objetos de formateo XSL.

La habilidad del lenguaje de transformación de mover los datos desde una representación XML a otra, lo constituye como un componente importante del comercio electrónico basado en XML, permitiendo el intercambio electrónico de datos, intercambio de metadata, y cualquier aplicación que necesita convertir los mismos datos entre las diferentes representaciones de XML.

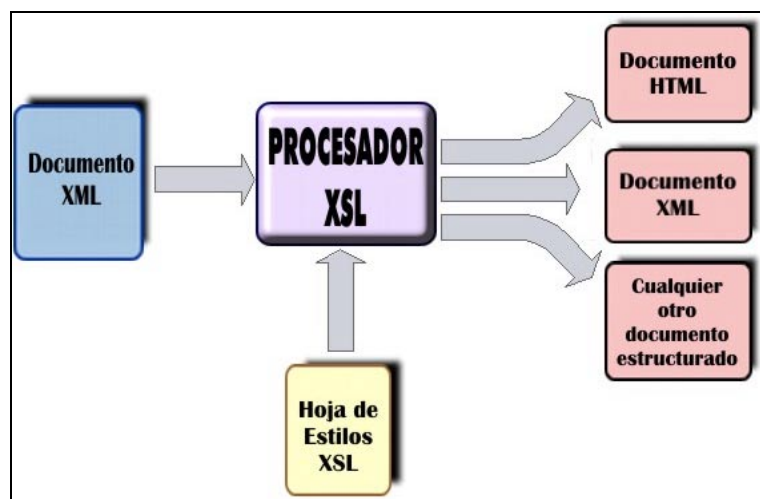


**Figura 4.2** Aplicación de hojas de estilos a documentos XML.

Por consiguiente, muchas aplicaciones tempranas de XSL dan enfoque exclusivamente a la parte de la transformación e ignoran los objetos de formateo. Éstas son aplicaciones incompletas, pero no obstante útiles.

#### 4.3.1. Procesador XSL

Para poder utilizar XSL se requiere de un software denominado XSL Engine (Motor XSL) o procesador XSL. Un procesador XSL es el software que convierte un archivo XML en resultado estructurado.

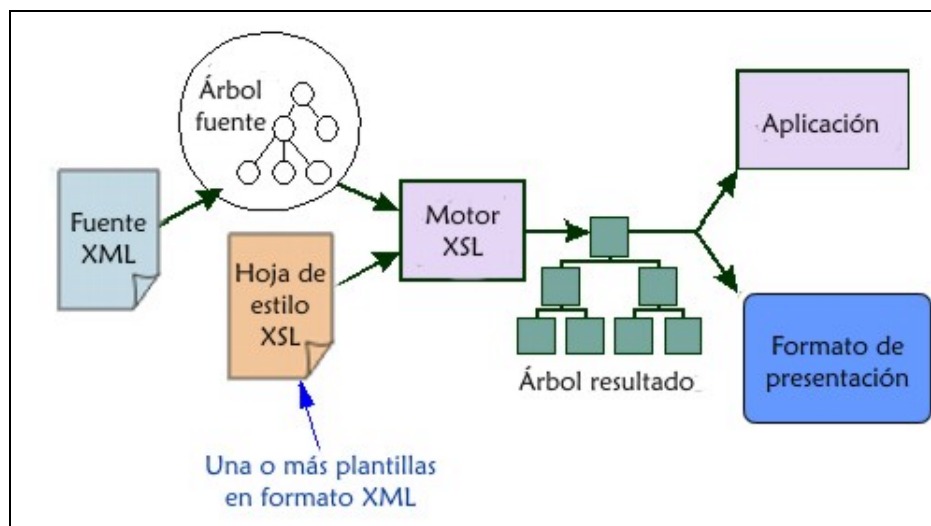


**Figura 4.3.** Procesador XSL

La forma de trabajo de un procesador XSL se resume así:

- Al iniciar cualquier transformación, el documento XML debe ser transformado mediante un "Parser XML" a un árbol. Dos cosas son necesarias antes de que un procesador XSL pueda procesar un documento: la representación del documento en árbol XML y la hoja de estilo XSL. La representación de un documento en árbol XML se obtiene analizando sintácticamente el documento, lo que significa que el procesador XSL debe emparejarse con un parser XML para que funcione.
- Una vez transformado el documento se aplica una hoja de estilo XSL al árbol. Ya que ha sido aplicada la hoja de estilo se genera otro árbol. El procesador comienza con el nodo raíz del árbol, utilizándolo para llevar a cabo el cotejo de los patrones en la hoja de estilo. El procesador analiza las plantillas de la hoja de estilo y los patrones asociados para procesar las distintas partes del árbol. El procesador XSL sigue las reglas de la plantilla para generar un árbol de resultados. Este punto es importante, ya que toma un árbol como entrada y genera otro árbol como salida.
- Finalmente la estructura resultante es transformada o enviada a una aplicación de servidor, bases de datos u otra según sea el caso.





**Figura 4.4.** Trabajo de un procesador XSL

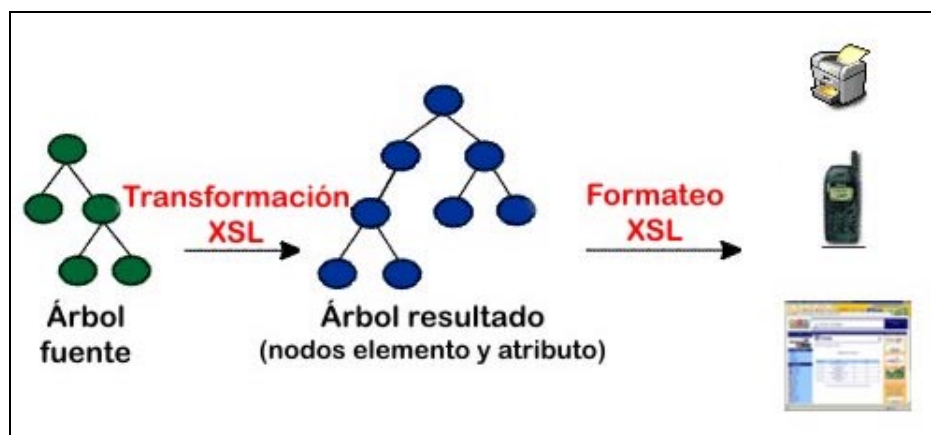
#### 4.3.1.1. Procesando una Hoja de Estilo XSL [WWW013]

Un procesador de hojas de estilo XSL acepta un documento XML y una hoja de estilo XSL y produce la presentación del documento fuente XML. Hay dos aspectos de este proceso de presentación: primero, la construcción de un árbol resultado del árbol fuente XML y segundo, la interpretación del árbol resultado para producir los resultados estructurados convenientes para la presentación en pantalla, en papel, o hacia otros medios. El primer aspecto se llama **transformación del árbol** y el segundo se llama **formateo**. El proceso de formateo es realizado por el **formateador**. Este formateador simplemente puede ser, un procesador(maquina-motor) dentro de un navegador web.

La transformación del árbol deja la estructura del árbol resultado significativamente diferente de la estructura del árbol fuente. Al construir el árbol resultado, el proceso de transformación del árbol agrega también la información necesaria para estructurar ese árbol resultado.

El formateo se habilita incluyendo la semántica de formateo en el árbol de resultado. La semántica de formateo se expresa en términos de un catálogo de clases de objetos de formateo. Los nodos del árbol resultado son objetos de formateo. Las clases de objetos de formateo denotan características como la página, párrafo, tabla, y así sucesivamente. El control de la presentación de estas abstracciones se proporciona por un juego de propiedades de formateo, como el control de sangrías, espaciado de palabras, párrafos y el control de la incorporación de guiones.

En XSL, las clases de objetos de formateo y las propiedades de formateo mantienen el vocabulario expresando el intento de la presentación.



**Figura 4.5.** Los dos procesos XSL: Transformación y Formateo.

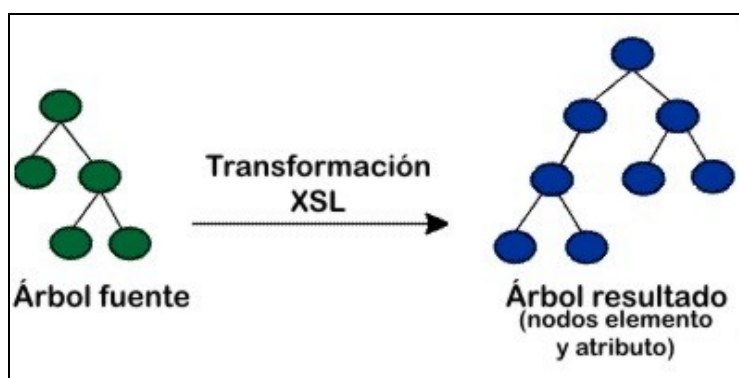
#### 4.3.2. Lenguaje de Transformación XSLT (XSL Transformation) [WWW013]

El **lenguaje de transformación** permite, partir de un documento XML y conseguir otro documento XML perteneciente a otro vocabulario distinto. El nuevo documento puede contener toda la información del primero o parte de ella. Este tipo de transformaciones garantiza la compatibilidad entre sistemas que utilicen distintos vocabularios, simplemente se ha de añadir la hoja de estilo XSL adecuada. Esto suele suceder en aplicaciones de comercio electrónico, intercambio electrónico de datos, metadatos, etc.

El lenguaje de transformación también podrá garantizar la visualización de documentos XML sobre navegadores HTML realizando un preproceso de los mismos, que puede ser transparente al usuario.

En una transformación de XSL, un procesador XSL lee un documento XML y una hoja de estilo XSL asociada a este documento, basándose en las instrucciones que el procesador encuentre en la hoja de estilo XSL, él procesador produce un nuevo documento XML.

Más precisamente, una transformación XSL acepta como entrada un árbol representado como un documento XML y produce como salida un nuevo árbol, también representado como un documento XML. En XSL, este árbol se llama **árbol de atributos y elementos**. En este árbol, un objeto de formateo se representa como un elemento XML, con las propiedades representadas por un juego de pares atributo-valor XML. El contenido del objeto de formateo es el contenido del elemento XML.



**Figura 4.6.** Proceso de transformación XSLT, de XML a XML.

Por consiguiente, la parte de la transformación de XSL también se llama la parte de construcción de un árbol. La entrada y la salida deben ser documentos XML. No se puede usar XSL para transformar a o desde formatos no XML. Se puede usar XSL para transformar a o desde formatos HTML y SGML que cuenten con reglas XML bien formadas. Es importante recordar que el lenguaje de transformación trabaja nada más para conversiones XML-a-XML.

La hoja de estilo XSL se usa en la transformación del árbol. Una hoja de estilo XSL contiene un conjunto de reglas de construcción del árbol, estas reglas tienen dos partes: un modelo o patrón (pattern), que se hace coincidir contra los elementos en la fuente, que especifica los árboles a los que se aplica la regla y una plantilla (template) que construye una porción del árbol resultado, que es el resultado cuando el patrón coincide. Esto permite a una hoja de estilo ser aplicable a una amplia clase de documentos que tienen estructuras de árbol fuente similares.

Cuando un procesador XSL estructura un documento XML usando una hoja de estilo XSL, examina el árbol del documento XML a través de cada sub-árbol. Como cada árbol en el documento XML es leído, el procesador lo compara con el modelo o patrón de cada plantilla de la regla en la hoja de estilo. Cuando el procesador encuentra un árbol que coincide con el patrón de una regla, se aplica la plantilla. Esta plantilla generalmente incluye algunas etiquetas, varios nuevos datos, y algunos datos copiados fuera del árbol del documento XML original.

XSL usa XML para describir estas reglas, plantillas, y patrones más adecuadamente, todos los elementos que son las instrucciones de XSL son parte del espacio de nombres de XSL. Los nombres de todos los elementos de XSL empiezan con (xsl:).

### **4.3.3. Objetos de Formateo XSL-FO (XSL Formatting Objects) [WWW013]**

Los Objetos de formateo XSL(XSL-FO) es la segunda mitad del Lenguaje de Estilo Extensible(XSL). Los Objetos de Formateo XSL constituyen el componente de formateo de la tecnología XSL. El lenguaje de formateo, presenta una serie de objetos de presentación, que asociados a los elementos del documento, les asignan un aspecto determinado. XSL-FO es una aplicación de XML que describe cómo las páginas aparecerán cuando sean presentadas a un lector.

El formateo es el proceso de convertir el resultado de una transformación de XSL en una forma tangible para el lector u oyente. Este proceso comprende varios pasos algunos de los cuales dependen de otros de una manera no secuencial.

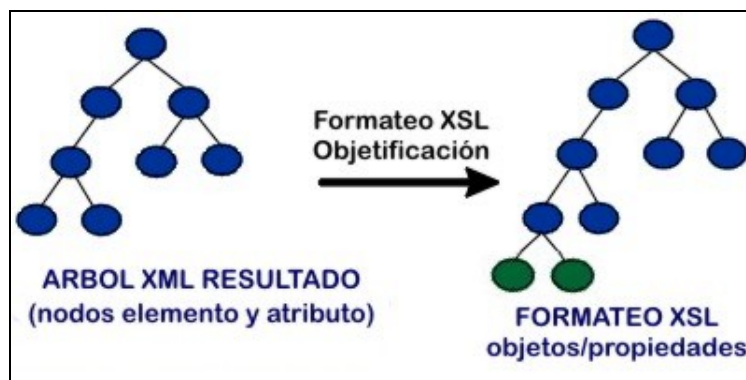
El formateador interpreta el árbol resultado en su forma de árbol de objetos de formateo para producir la presentación.

Cada objeto de formateo representa una especificación para una parte de la paginación, el diseño, e información de estilo que se aplicará al contenido de ese objeto de formateo como resultado de estructurar el árbol de resultado completo. Cada clase de objetos de formateo representa un tipo particular de conducta de formateo.

El formateo consiste en la generación de un árbol de áreas geométricas, llamado árbol de áreas. Las áreas geométricas se posicionan en una sucesión de una o más páginas (un navegador usa generalmente una sola página). Cada área geométrica tiene una posición en la página, una especificación de qué desplegar en esa área y puede tener un fondo, relleno, y bordes. Estas áreas pueden anidarse.

El formateador toma el árbol de áreas, el modelo abstracto de la presentación (en términos de las páginas y sus colecciones de áreas), y causa una presentación para ser desplegada en un medio pertinente, como una ventana del navegador, en una pantalla de computadora u hojas de papel.

El primer paso en el formateo consiste básicamente en convertir los elementos en nodos de objetos de formateo y los atributos en las especificaciones de propiedades. El resultado de este paso es el árbol de objetos de formateo.

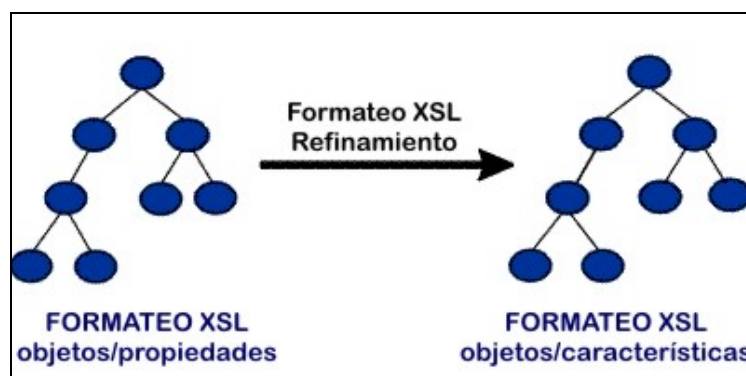


**Figura 4.7.** Construcción del árbol de objetos de formateo.

La segunda fase en el formateo es refinar el árbol de objetos de formateo para producir el árbol de objetos de formateo refinado. Éste es un proceso computacional que finaliza la especificación de propiedades basado en los valores del atributo en el árbol resultado XML.

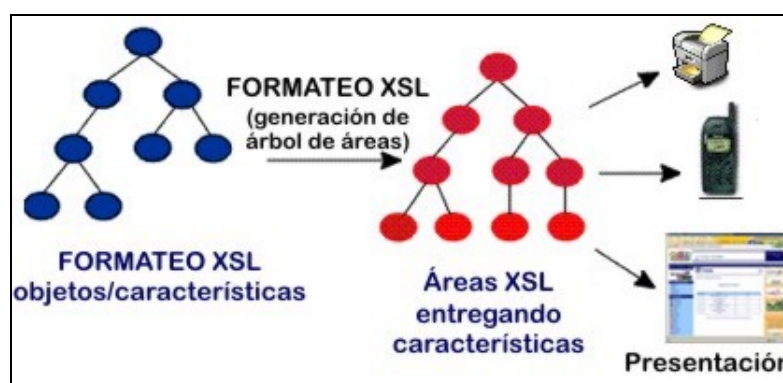
Aunque el árbol resultado XML y el árbol de objetos de formateo tienen estructura muy similar, es útil pensar en ellos como entidades conceptuales separadas. El refinamiento involucra:

- Generación de varios valores heredados de propiedades (implícitamente y aquellos con un valor de atributo de "herede").
- Evaluación de expresiones en las especificaciones de valor de propiedad en valores reales, que se usan para determinar el valor de las propiedades.
- Convertir numéricos relativos a numéricos absolutos.
- Construir algunas propiedades compuestas de más de un atributo.



**Figura 4.8.** Refinamiento de árbol de objetos de formateo.

El tercer paso del formateo es la construcción del árbol de áreas. El árbol de áreas se genera como se describe en la semántica de cada objeto de formateo. Las características aplicables a cada clase de objetos de formateo controlan cómo se generan las áreas. Aunque cada propiedad de formato puede especificarse en cada objeto de formateo, para cada clase de objeto de formateo, sólo un subconjunto de las propiedades de formateo se usa para determinar los rasgos para los objetos de esa clase.



**Figura 4.9.** Construcción del árbol de áreas.

Al final del procedimiento, se han construido las áreas y sus características, y deben satisfacer las restricciones descritas en las definiciones de sus objetos de formateo asociados, y en la sección de modelo de área.

En particular, el tamaño y posición de las áreas estarán sujetos a la colocación y espaciado restringidos descritos en el modelo de área, a menos que la definición del objeto de formateo indique otra cosa.

#### 4.3.4. Ventajas de XSL.

- La salida no tiene por qué ser HTML para visualización en un navegador, sino que puede estar en muchos formatos y se los puede visualizar en otros dispositivos.
- Permite manipular de diversas maneras un documento XML: reordenar elementos, filtrar, añadir, borrar, etc.
- Permite acceder a todo el documento XML, no sólo al contenido de los elementos.
- XSL es un lenguaje XML, por lo que no hay que aprender nada especial acerca de su sintaxis.

#### 4.3.5. Desventajas de XSL.

- Su utilización es más compleja.
- Consume cierta memoria y capacidad de proceso, pues se construye un árbol con el contenido del documento.

#### 4.4. Comparación entre CSS y XSL

Una hoja de estilo XSL es considerablemente más potente y flexible que una hoja de estilo CSS. Mientras que CSS simplemente permite especificar el formato de cada elemento XML, una hoja de estilo XSL proporciona un control completo sobre la salida. Específicamente, XSL permite seleccionar de manera precisa los datos XML que se quieren visualizar, presentar dichos datos en cualquier orden o disposición y modificar o añadir información con total libertad. XSL proporciona acceso a todos los componentes XML (como elementos, atributos, comentarios e instrucciones de procesamiento), permite ordenar y filtrar fácilmente los datos XML, permite incluir scripts en la hoja de estilo y proporciona un conjunto de métodos de utilidad que pueden invocarse para trabajar con la información. XSL, sin embargo, es algo más complejo y difícil de aprender que CSS. También es una tecnología más reciente que CSS y, por tanto, dispone de menos soporte (y de un soporte menos coherente) entre los exploradores actuales. [LIB007]

También cabe señalar dos diferencias claves entre XSL y CSS:

- Las CSS se pueden usar para dar estilo a los documentos HTML, mientras que XSL no se usa con documentos HTML.
- XSL se puede usar para transformar documentos XML, mientras que las CSS no.

Al trabajar con XML la primera diferencia podría no importar mucho. Sin embargo, si consideramos que la mayoría de las aplicaciones XML implican en cierta medida a los documentos HTML, podría ser importante a la hora de elegir la tecnología de hojas de estilo más adecuada en un determinado proyecto.

La segunda diferencia es muy importante, ya que XSL proporcionan una forma directa de transformar documentos XML. Sin embargo, aunque no sea posible utilizar las CSS para transformar los documentos XML, si pueden utilizarse para darles estilo.

Las CSS fueron explícitamente diseñadas para ser utilizadas por los no programadores, lo que explica por qué son tan fáciles de aprender y usar. La desventaja obvia de la CSS es que no son más que un lenguaje de formato; adjunta propiedades de estilo a los

elementos de un documento XML/HTML. Como tales, las CSS tienen algunas limitaciones muy importantes:

- No pueden tomar una parte del documento y reutilizarla en otra parte.
- No conceptúan las relaciones hermanas entre los nodos.
- No soporta las estructuras de decisión (condicionales).
- No puede calcular cantidades o almacenar valores en variables.

XSL es una tecnología más potente que las CSS, pero la potencia añadida se incorpora también con una complejidad añadida. Si no importa utilizar un mayor esfuerzo en aprender XSL, las posibilidades de XSL de buscar y reorganizar el contenido del documento serán mayores.

#### 4.5. Transformación XML/XSL en aplicativos web [www014]

Hay tres maneras primarias en que los documentos XML se transforman en otros formatos, como HTML, con una hoja de estilo XSL:

1. El documento XML y la hoja de estilo asociada son servidos al cliente (navegador Web), que transforma el documento como está especificado por la hoja de estilo y es presentado al usuario.

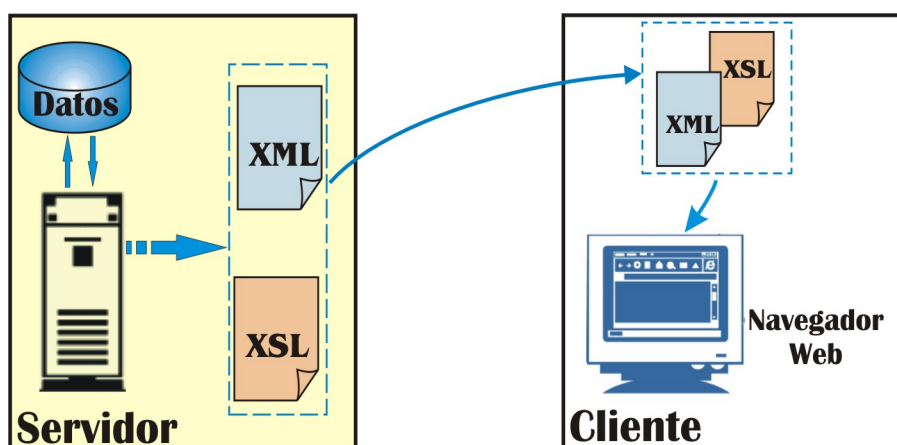


Figura 4.10. Transformación del lado del cliente.



2. El servidor aplica una hoja de estilo XSL a un documento XML transformándolo a algún otro formato (generalmente HTML) y envía el documento transformado al cliente (el navegador Web).

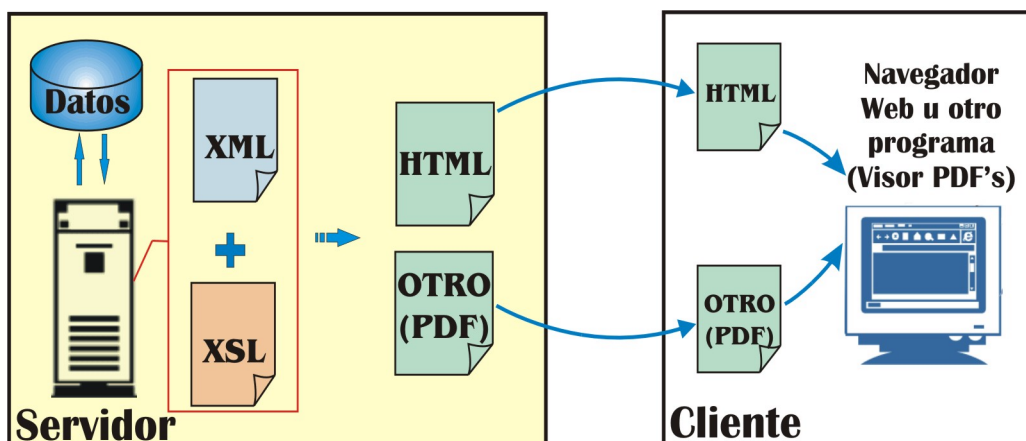


Figura 4.11. Transformación del lado del servidor.

3. Un tercer programa transforma los documentos XML originales en algún otro formato (a menudo HTML) antes de poner el documento en el servidor. El servidor y cliente sólo tratan con el documento post-transformado.

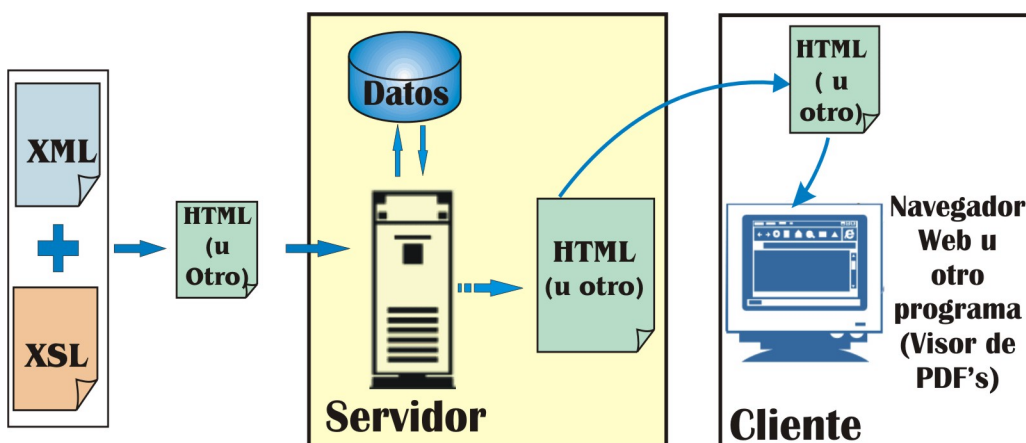


Figura 4.12. Transformación antes de colocar en el servidor.

Cada uno de estos tres acercamientos usan software diferente, aunque todos ellos usan el mismo documento XML y las hojas de estilos XSL. Un servidor Web ordinario que envía los documentos de XML a Internet Explorer 6.0 es un ejemplo del primer acercamiento. Un servidor Web compatible con servlets que usa Cocoon del proyecto Apache ejemplifica el segundo acercamiento. Un ejemplo del tercer acercamiento se da

al usar la línea de comandos del programa XT para transformar el documento XML a documento XHTML, y luego colocar los documentos XHTML en un servidor Web. Sin embargo, todos estos usan el mismo lenguaje XSL.

En esta tesis, se dará énfasis al segundo acercamiento, porque al momento se cuenta con la estructura de publicación Cocoon, software en el servidor web, que permite transformar el documento XML en HTML o PDF, aplicando hojas de estilos XSL. Además, esto proporciona mayor compatibilidad con navegadores Web y servidores considerando que el primer acercamiento requiere un navegador más reciente del que la mayoría de los usuarios usa y el tercer acercamiento requiere que se realice la transformación antes de ser colocados los documentos en el servidor web. En la práctica, sin embargo, requerir un software adicional en el servidor no es tan oneroso como requerir a un cliente particular.