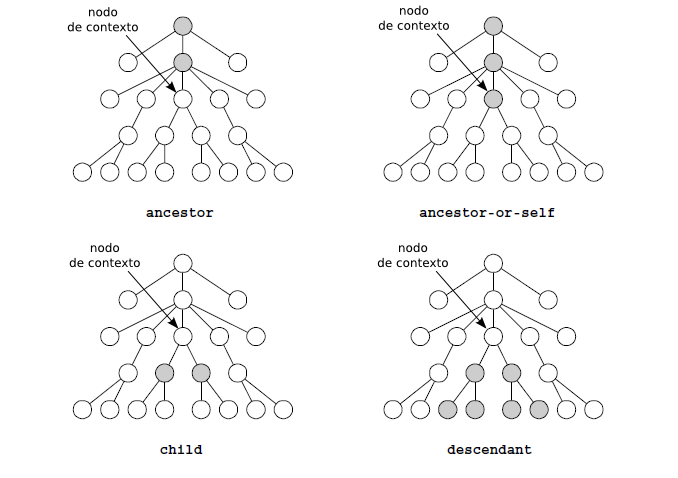
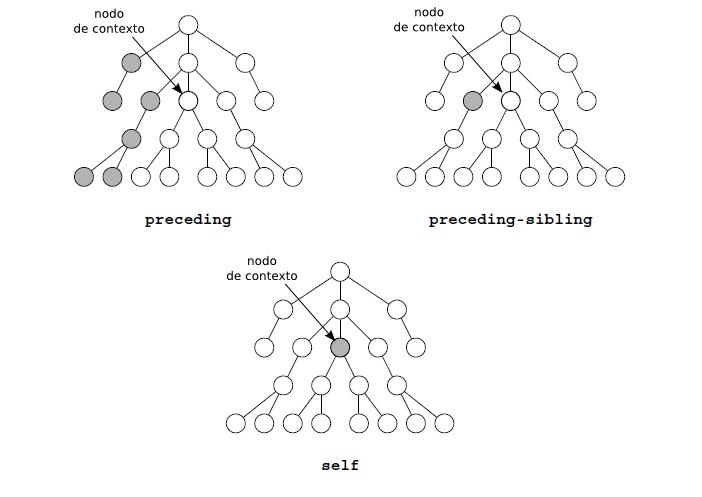
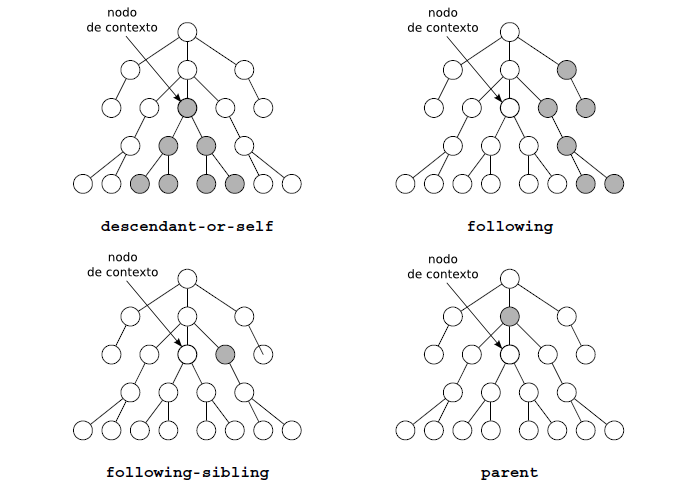
**LOS EJES DE XPATH**

XPath tiene un total de 13 ejes, que son usados para navegar a través del árbol de nodos del modelo de datos XPath. XSLT soporta todos estos ejes. Los ejes son "child","attribute", "ancestor", "ancestor-or-self", "descendant", "descendant-or-self", "following", "following-sibling", "namespace", "parent", "preceding", "preceding-sibling" y "self".

Los siguientes gráficos muestra los nodos pertenecientes a cada eje:

*NOTA: Los nombres de los ejes están en minúscula, y así se deben escribir siempre. XML es casesensitive.*

Vamos uno por uno.

**EJE CHILD**

El eje por defecto en XPath es "child". Selecciona los nodos que son hijos inmediatos del nodo contexto. Por ejemplo:

**<Factura>**

**<Fecha> 02-01-2011 </Fecha>**

**<Item cantidad="4">AS11</Item>**

**<Item cantidad="5">AS13</Item>**

**</Factura>**

Si el nodo de contexto es el nodo elemento "Factura", la ruta "child::Item" (o en su modo abreviado "Item") devolverá un "node-set" con los nodos "Item", hijos del elemento "Factura".

Para seleccionar tanto elementos "Fecha" como "Item", usaremos la ruta "child::\*" (o en su modo abreviado "\*").

Para seleccionar todos los nodos hijo, incluyendo nodos comentario y nodos texto, podemos escribir la siguiente ruta: "child::node()" (o en su modo abreviado "node()").

Si lo que queremos es seleccionar específicamente los nodos texto hijos del nodo de contexto, podemos escribir "child::text()" (o en su modo abreviado "text()").

Como "child" es el eje por defecto, las siguientes rutas significan lo mismo:

**/child::Libros/child::Libro/child::Capítulos**

**/Libros/Libro/Capítulos**

*NOTA: Prueba a evaluar la siguiente ruta en el documento anterior sobre Libros: /Libros/Libro/Capítulos/Capítulo/text(). Con esta ruta estás recorriendo todos los nodos Capítulo del árbol y recuperando su nodo texto.*

**EJE ATRIBUTO**

Suponiendo que el nodo de contexto es un nodo elemento, la rutas siguientes significan lo mismo:

**attribute::\***

**@\***

Ambas rutas devolverán todos los nodos atributo asociados con el nodo elemento.

*NOTA: Si el nodo de contexto no es un nodo elemento, el eje atributo devolverá un node-set vacío.*

**PARA UN MOMENTO. ¡PONME UN EJEMPLO!**

En el siguiente ejemplo, vamos a ver un documento XML, y un documento XSLT que lo procesa. En el documento XSLT emplearemos tanto ejes "child" como "attribute".

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

**<DatosPersona>**

**<Persona FDN="25/11/1920">**

**<Nombre>Juan</Nombre>**

**<Apellido1>Pérez</Apellido1>**

**<Apellido2>García</Apellido2>**

**</Persona>**

**</DatosPersona>**

El documento XSLT, DatosPersona.xslt es el siguiente:

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

**<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">**

**<xsl:template match="/">**

**<html>**

**<head>**

**<title>Información sobre <xsl:value-of select="/DatosPersona/Persona/Nombre"/>**

**<xsl:text> </xsl:text>**

**<xsl:value-of select="/DatosPersona/Persona/Apellido1" />**

**</title>**

**</head>**

**<body>**

**<p><xsl:value-of select="/DatosPersona/Persona/Nombre" />**

**<xsl:text></xsl:text>**

**<xsl:value-of select="/DatosPersona/Persona/Apellido1" /> Nació en**

**<xsl:value-of select="/DatosPersona/Persona/@FDN" /></p>**

**</body>**

**</html>**

**</xsl:template>**

**</xsl:stylesheet>**

No entremos en detalles sobre XSLT. La idea es sencillamente que se entiendan las rutas:

<xsl:value-of select="/DatosPersona/Persona/Nombre"/> es una ruta absoluta, con sintaxis abreviada. Desde el nodo raíz "/", "DatosPersona" es buscado a través del eje "child" que es el eje por defecto. Es decir, se accede a través del eje "child" buscando un nodo llamado "DatosPersonaa". El proceso se repite para "Persona" y "Nombre".

<xsl:value-of select="/DatosPersona/Persona/@FDN"/> indica el atributo "FDN" del elemento Persona. De nuevo, la sintaxis abreviada indica que iremos navegando por los ejes "child" de cada nodo elemento. Desde el nodo raíz, entramos por Persona, luego por Nombre, y luego la @ indica que busquemos el nodo atributo con dicho nombre (FDN).

**Actividad 32. Podemos usar XML Copy Editor para evaluar rutas. Por ejemplo, para el documento XML sobre libros:**

* **Crea una ruta y evalúala para obtener el valor de los atributos año de los elementos Editorial.**
* **Crea una ruta y evalúala para obtener el valor de los atributos número de los elementos Capítulo.**

**EJE ANCESTOR**

El eje ancestor selecciona el nodo padre del nodo de contexto, el nodo "abuelo", el "bisabuelo" y así sucesivamente. Si el nodo de contexto es el nodo raíz, el eje ancestor devuelve un node-set vacío.

Por ejemplo:

**<Libro>**

**<Capítulo número="1">**

**<Sección>Primera sección. </Sección>**

**<Sección>Segunda sección. </Sección>**

**</Capítulo>**

**</Libro>**

Supongamos que el nodo de contexto es el segundo elemento "Sección" del capítulo 1. Entonces la ruta "ancestor::\*" devolvería el nodo elemento "Capítulo", que tiene un atributo "número" que vale 1. También devolverá el nodo elemento "Libro". Y por último devolverá el nodo raíz.

*NOTA: La sintaxis reducida para el eje "ancestor" es "..".*

Prueba a evaluar la ruta "/Libros/Libro/Capítulos/ancestor::\*" y la ruta "/Libros/Libro/Capítulos/ancestor::Libro" sobre el documento sobre libros.

**Actividad 33. Trata de explicar el resultado de evaluar la siguiente ruta XPath, y después evalúala para comprobar si has acertado:**

**/Libros/Libro/Nombre[text()="MySQL"]/../Capítulos/Capítulo[position()=2]**

**Reescribe la ruta anterior sin utilizar sintaxis reducida del nodo ancestor.**

**EJE DESCENDANT**

Dado el nodo de contexto, el eje descendant selecciona su nodo hijo, su nodo "nieto", su "bisnieto" y así sucesivamente.

En el ejemplo anterior, si el nodo de contexto es "Libro", la ruta "descendant::\*" seleccionaría tanto el nodo elemento "Capítulo", así como los dos nodos "Sección".

Si escribimos la ruta "/descendant::Sección", estaríamos describiendo todos los nodos elemento "Sección" que contiene el nodo de contexto.

*NOTA: Aquí no hay sintaxis abreviada.*

¿Qué nodos crees que devolvería la ruta "/Libros/Libro/descendant::Capítulos"? Prueba a evaluarla y comprueba si estabas en lo cierto.

**EJE DESCENTANT-OR-SELF**

Incluye todos los nodos en el eje "descendant" y también el nodo de contexto. La forma abreviada es "//".

Este eje nos permite encontrar nodos, independientemente de su posición.Es decir, si queremos todos los elementos "Capítulo" pero no estamos seguros de la jerarquía XML (o quizá el elemento "Capítulo" puede estar anidado), la ruta "//Capítulo" nos los devolverá todos. Esta ruta requiere un trabajo intensivo de búsqueda, lo que tiene un coste computacional.

¿Qué crees que devolvería la ruta "/Libros/Libro/Capítulos/descendant-or-self::\*? Prueba a evaluarla y comprueba si estabas en los cierto.

**Actividad 34. Definir la ruta necesaria para recuperar el nodo "Capitulo" así como todos sus descendientes.**

**EJES FOLLOWING**

El eje "following" contiene todos los nodos que vienen tras el nodo de contexto según el orden del documento, pero excluye todos los nodos descendientes así como cualquier nodo atributo y nodo namespace asociado con el nodo de contexto.

Un ejemplo:

**<Empleados>**

**<Persona>**

**<Nombre>Pedro</Nombre>**

**<Apellido1>Pérez</Apellido1>**

**<FechaDeNacimiento>12-12-1970</FechaDeNacimiento>**

**</Persona>**

**<Persona>**

**<Nombre>José</Nombre>**

**<Apellido1>García</Apellido1>**

**<FechaDeNacimiento>11-10-1984</FechaDeNacimiento>**

**</Persona>**

**<Persona>**

**<Nombre>Ángel</Nombre>**

**<Apellido1>Merino</Apellido1>**

**<FechaDeNacimiento>01-02-1980</FechaDeNacimiento>**

**</Persona>**

**</Empleados>**

La siguiente hoja de estilos, se llama "Empleados.xsl":

**<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">**

**<xsl:template match="/">**

**<html>**

**<head>**

**<title>Demostración del eje following.</title>**

**</head>**

**<body>**

**<h3>Demostración del eje following.</h3>**

**<xsl:apply-templates select="/Empleados/Persona[1]/Nombre" />**

**</body>**

**</html>**

**</xsl:template>**

**<xsl:template match="Nombre">**

**<xsl:for-each select="following::\*">**

**<p><xsl:value-of select="name(.)" /> -> Contiene el texto**

**"<xsl:value-of select="." />".</p>**

**</xsl:for-each>**

**</xsl:template>**

**</xsl:stylesheet>**

En este ejemplo, usamos el eje following en el elemento "xsl:for-each" hasta el final de la hoja de estilos XSLT.

En el documento resultante, se va avanzando por cada uno de los nodos elemento.

¿Qué crees que devolverá la ruta "/Libros/Libro/Capítulos/following::\*"? Prueba a evaluarla y comprobar si estabas en lo cierto.

**EJE FOLLOWING-SIBLING**

El eje "following-sibling" incluye cualquier nodo en el eje "following" que comparte su nodo padre con el nodo de contexto. Vamos a ver un ejemplo aclaratorio. Partiendo del documento XML anterior, si en este caso elegimos el nodo "Nombre" como nodo de contexto, entonces el eje "following-sibling" incluirá a los nodos siguientes que comparten al mismo nodo elemento padre, es decir, "Apellido1" y "FechaDeNacimiento". Prueba a pasar esta hoja de estilos XLST al archivo XML anterior:

**<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >**

**<xsl:template match="/">**

**<html>**

**<head>**

**<title>Demostración del eje following-sibling.</title>**

**</head>**

**<body>**

**<h3>Demostración del eje following.</h3>**

**<xsl:apply-templates select="/Empleados/Persona[1]/Nombre" />**

**</body>**

**</html>**

**</xsl:template>**

**<xsl:template match="Nombre">**

**<xsl:for-each select="following-sibling::\*">**

**<p><xsl:value-of select="name(.)" /> -> Contiene el texto**

**"<xsl:value-of select="." />".</p>**

**</xsl:for-each>**

**</xsl:template>**

**</xsl:stylesheet>**

El documento resultante, solo contiene los elementos siguientes, que comparten al mismo padre.

¿Qué crees que devolverá la ruta "/Libros/Libro/Capítulos/Capítulo/following-sibling::\*? Prueba a evaluarla y comprobar si estabas en lo cierto.

**EJES PARENT**

Los ejes "parent" se usan para seleccionar los nodos padre del nodo contacto. Por ejemplo:

**<Partes>**

**<Parte número="ABC123"/>**

**<Parte número="DEF456"/>**

**</Partes>**

Si el nodo de contexto está en uno de los nodos elemento "Parte", entonces la ruta siguiente selecciona su nodo padre, es decir, el nodo elemento "Partes":

**parent::node()**

En cambio, si el nodo de contexto fuese el nodo elemento "Partes", la ruta anterior seleccionaría el nodo raíz.

*NOTA: Una forma de comprobar si estamos en el nodo raíz es comprobar si el nodo padre es "null". El nodo raíz es el único sin un padre.*

La sintaxis abreviada para el nodo padre es "..". Se parece a la forma de indicar el directorio padre en un sistema de archivos.

¿Qué crees que devolverá la ruta "/Libros/Libro/Capítulos/.."? Prueba a evaluarla para ver si estabas en lo cierto.

**EJE PRECEDING**

El eje "preceding" contiene todos los nodos anteriores al nodo de contexto, según el orden del documento, excluyendo los nodos del eje "ancestor" y los nodos atributo y namespace.

Si tomamos el ejemplo anterior de los empleados, y le pasamos la siguiente hoja de estilos:

**<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >**

**<xsl:template match="/">**

**<html>**

**<head>**

**<title>Demostración del eje preceding.</title>**

**</head>**

**<body>**

**<h3>Demostración del eje preceding.</h3>**

**<xsl:apply-templates select="/Empleados/Persona[3]/FechaDeNacimiento" />**

**</body>**

**</html>**

**</xsl:template>**

**<xsl:template match="FechaDeNacimiento">**

**<xsl:for-each select="preceding::\*">**

**<p><xsl:value-of select="name(.)" /> --< Contiene el texto**

**"<xsl:value-of select="." />".</p>**

**</xsl:for-each>**

**</xsl:template>**

**</xsl:stylesheet>**

Si probamos el ejemplo, veremos que se muestran todos los elementos anteriores al elemento "FechaDeNacimiento" del último elemento "Persona".

¿Qué crees que devolverá la ruta "/Libros/Libro/Nombre/preceding::\*"? Prueba a evaluarla para ver si estabas en lo cierto.

**EJE PRECEDING-SIBLING**

Este eje incluye todos los nodos que están dentro del eje "preceding" y que comparten padre con el nodo de contexto. O sea, que si en el ejemplo anterior, probamos a cambiar "prededing::\*" por "preceding-sibling::\*", obtendríamos solo los elementos "nombre" y "apellido1" del tercer elemento "persona".

¡Ala! A probarlo.

**EJE SELF**

El eje "self" selecciona el nodo de contexto. La forma no abreviada del eje self es "self::node()". Su forma abreviada es ".". Es decir, que las dos rutas siguientes serían equivalentes:

**<xsl:value-of select="." />**

**<xsl:value-of select="self::node()" />**