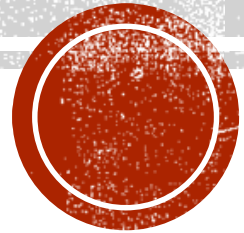
CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Técnicas de transformación de documentos XML.
- Descripción de la estructura y de la sintaxis.
- Utilización de plantillas.
- Utilización de herramientas de procesamiento.
- Elaboración de documentación.

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- En el ecosistema del metalenguaje de marcas XML existen tecnologías que permiten transformar los datos contenidos en los documentos XML
- Esta transformación tiene dos vías principales:
  - La transformación para el cambio de lenguaje de representación (por ejemplo, de XML a HTML) → genera un fichero en nuevo lenguaje
  - La adaptación para su presentación en diversos dispositivos o formatos físicos de salida → genera un nuevo documento visualizable
- **XSL** es una familia de lenguajes desarrollados por el W3C que permiten esta transformación e incluye las tecnologías **XSLT, XSL-FO y Xpath**

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Por tanto el **XSL** (**eXtensible Stylesheet Language**, «lenguajes de hojas de estilo extensible», en castellano) es una **familia de tecnologías desarrolladas por el W3C** que tienen como **objetivo** proporcionar herramientas para **transformar los documentos XML**

- Un documento XML se puede transformar en cualquier cosa que se pueda representar mediante cadenas de caracteres:
  - Lenguaje HTML
  - Formato CSV
  - Sentencias SQL
  - Programas informáticos
  - Imágenes vectoriales
  - Cualquier dialecto de XML
  - …

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

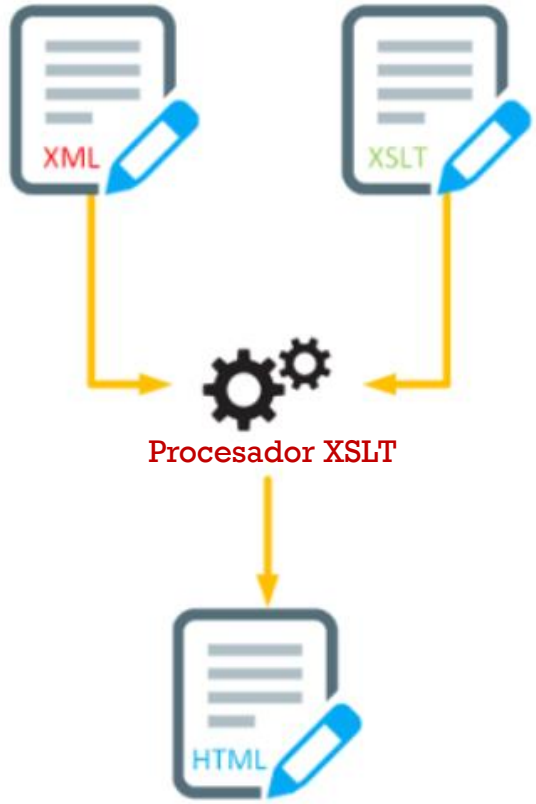
- La familia XSL está formada por tres elementos:
  - **XSLT** (XSL Transformation) **es el lenguaje de transformación propiamente dicho**. En muchas ocasiones XSL y XSLT hacen referencia al mismo concepto: la transformación de documentos XML
  - **XSL-FO** (XSL Formatting Objects) **es el lenguaje que permite indicar el formato que en que se presenta la información contenida en un documento XML** para mostrarse en diferentes dispositivos o medios, como pantallas o papel
  - **XPath** (XML Path Language) **es el lenguaje que permite obtener información de un documento XML utilizando expresiones**. Se utiliza en XSLT para seleccionar las diferentes partes de los documentos

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT
  - Un documento XML es un contenedor de datos estructurados, formado por elementos, atributos y dependencias jerárquicas
    - Estos datos están delimitados por unas etiquetas o marcas que no tienen más significado que el que les quiera asignar el creador del documento
  - En HTML las etiquetas están predefinidas en la especificación del lenguaje y tienen un significado asociado con un estilo de presentación (propio o asignado mediante CSS)
  - **Con XSLT se puede transformar un documento expresado en el formato XML en un documento equivalente expresado en otro formato, normalmente HTML**

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - El proceso de transformación consiste en seleccionar diferentes partes del documento XML e indicar en qué se quieren transformar
  - La selección se realiza mediante XPath y la transformación mediante XSLT. Las transformaciones XSLT se almacenan en ficheros XML con extensión .xsl
  - Los **componentes** que forman parte de una transformación son los siguientes:
    - Un **documento XML** de entrada
    - Un **fichero de transformación XSLT**
    - Un **procesador** con capacidad para aplicar las transformaciones (**un editor o un navegador web**) cuando se abre el documento xml con él



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - El documento XML debe incluir una referencia al documento XSL:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<?xml-stylesheet type="text/xsl" href="ejemplo1.xsl">
```

  - La estructura del documento XSL debe incluir la declaración del espacio de nombres de XSL y el elemento que indica que se debe aplicar la transformación a todo el documento XML o a partes del mismo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

...

</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - Una hoja de transformaciones debe ir compuesta por lo siguiente:
    - Declaración del documento XML.

```
<?xml versión=”1.0?”>
```

    - Elemento raíz.

```
<xsl:stylesheet versión=”1.0”
xmlns= “http://www.w3.org/1999/XSL/Transform”
…
/xsl:stylesheet>
```

    - Espacio de nombres.

`http://www.w3.org/1999/XSL/Transform` --> donde xsl corresponde al prefijo.

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - En el documento XSL se alternan texto libre (normalmente HTML) con elementos XSL
    - Los **textos libres** se volcarán en la salida sin sufrir ninguna modificación
    - Los **elementos XSL** son los que aportan las reglas de transformación que utilizan los datos contenidos en el documento XML
      - En el siguiente ejemplo, incrustado en el código HTML se encuentran un par de elementos XSL cuyo significado es:
        A. «considera el valor del elemento <nombre> que se encuentra dentro del elemento <tienda> del documento XML de entrada»
        B. «considera el valor del elemento <telefono> que se encuentra dentro del elemento <tienda> del documento XML de entrada»

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - Ejemplo
    - Documento XML:

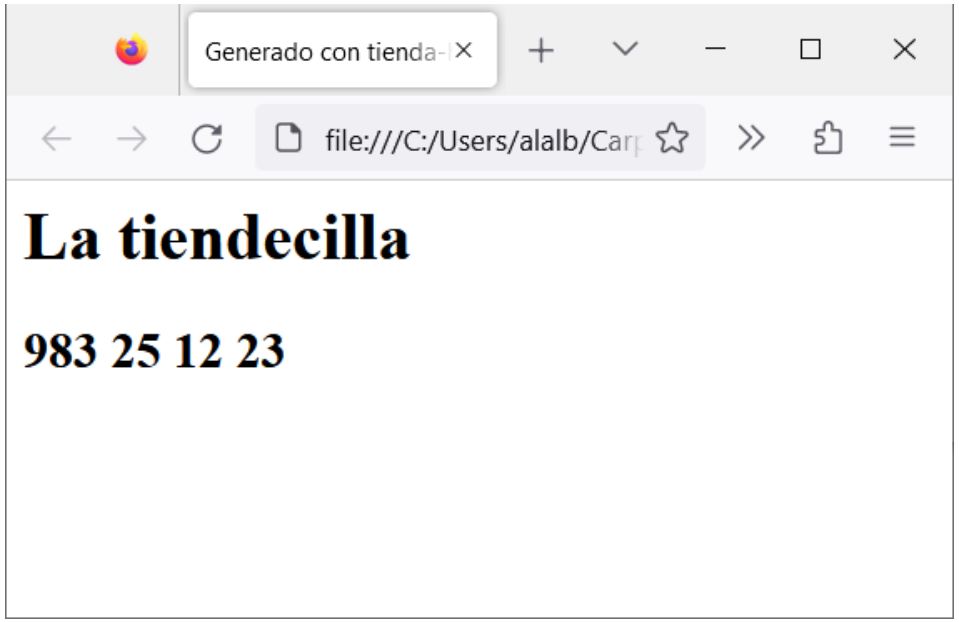
```
<?xml version="1.0" encoding='ISO-8859-1'?>
<?xml-stylesheet href="tienda-html.xsl" type="text/xsl"?>
<tienda>
   <nombre>La tiendecilla  </nombre>
   <telefono>983 25 12 23  </telefono>
</tienda>
```

    - Reglas de transformación XSLT (fichero tienda-html.xsl):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
  <html>
     <head><title>Generado con tienda-html.xsl</title></head>
     <body>
        <h1> <xsl:value-of select="tienda/nombre"/> </h1>
        <h2> <xsl:value-of select="tienda/telefono"/> </h2>
      </body>
   </html>
 </xsl:template>
 </xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - Como se puede observar, al abrir desde un navegador web (Edge por ejemplo) muestra una pantalla en blanco o no muestra el contenido como corresponde (en el caso de Firefox), esto es debido a la actualización de las políticas de seguridad de los navegadores web, los cuales ya no pueden ejecutar código XSL de archivos locales
  - Para que se muestre sin problemas se puede alojar el documento xml y el fichero .xls en un servidor web (local o remoto) y acceder al documento xml mediante el protocolo HTTP
  - Otra alternativa es generar el fichero HTML correspondiente con el procesador XSLT (por ejemplo XML Copy Editor)



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - En cualquiera de los casos que permiten la transformación del documento XML se realizan automáticamente las transformaciones declaradas en el fichero de transformación
  - Como se puede observar en el ejemplo el fichero .xsl contiene código HTML convencional junto con instrucciones XSLT, generando una vista equivalente a una página web convencional
  - En el ejemplo anterior si se genera el fichero html se obtiene:

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Generado con tienda-html.xsl</title>
  </head>
  <body>
    <h1>La tiendecilla  </h1>
    <h2>983 25 12 23  </h2>
  </body>
</html>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSLT. Estructura
  - Los elementos de XSL utilizados en el ejemplo anterior:
    - El elemento siguiente indica que nos encontramos ante una transformación y establece xsl como prefijo para las etiquetas del espacio de nombres de XSL:

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

    - El elemento <xsl:template> se utiliza para crear una plantilla relacionada con un atributo de coincidencia. En este caso, al utilizar **el valor "/" se está indicando que la plantilla afecta a todo el documento**:

```
<xsl:template match='/'>
```

    - Los elementos <xsl:value-of> indican que deben ser sustituidos por el valor del elemento obtenido como consecuencia de la consulta realizada con XPath y establecida en el atributo select

```
<h1> <xsl:value-of select="tienda/nombre"/> </h1>
<h2> <xsl:value-of select="tienda/telefono"/> </h2>
```

    - El resto es código HTML convencional en el cual se incrustan los valores proporcionados por los elementos <xsl:value-of>, lo que da lugar al documento HTML

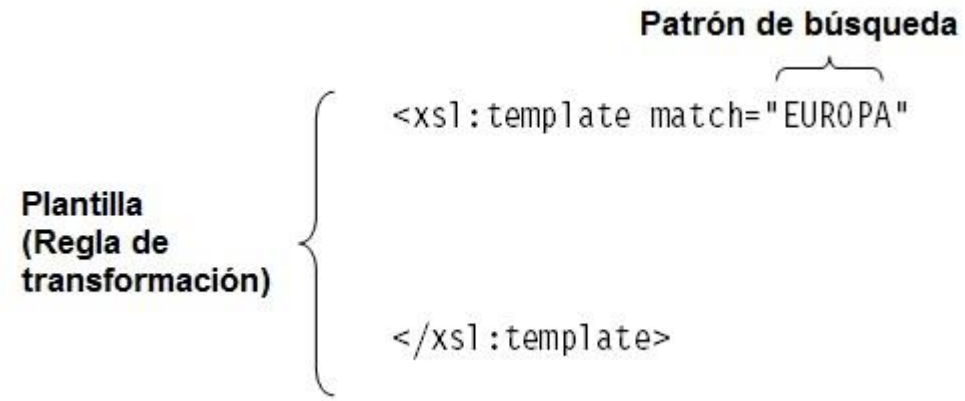
CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- La transformación no tiene por qué ser a **código HTML** aunque sea **la operación más habitual**
- Se puede realizar una transformación que genere un **documento JSON**
- También se puede generar **código java** con XSLT

- Se puede generar cualquier salida en formato texto a partir de un documento XML

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - **Las plantillas** (template) **son patrones** (patterns) **para la transformación del árbol fuente en el árbol resultado**
  - Se componen de dos partes, la primera se conoce como **patrón de búsqueda** y la segunda como **plantilla** propiamente dicha
  - Con el patrón se identifican los nodos del árbol fuente a los cuales se aplica la regla



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - Por tanto **una plantilla es un bloque que se define con el elemento**

    <xsl:template match="expresión XPath">
  - Que delimita una serie de contenidos y reglas XSL. Es un elemento destacado para definir cualquier transformación
  - Son muy importantes en XSLT ya que la transformación consiste en aplicar una colección de plantillas al documento de entrada para obtener el correspondiente documento de salida
  - El atributo match asociará la plantilla con un elemento XML, es una expresión Xpath

  - En el ejemplo indicado se ha usado la etiqueta <xsl:template match='/'>, que indica que se selecciona el documento completo, al utilizar el símbolo del elemento raíz

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - En el ejemplo:

```
<xsl:template match='/'>
      <html>
      <head>
                  <title>Título ejemplo</title></head>
      <body>
                  <h1> <xsl:apply-templates /> </h1>
      </body>
      </html>
</xsl:template>
```

  - La plantilla esta formada por contenidos que se desea incluir en el documento de salida:
    - Como las distintas etiquetas html o en su caso el valor correspondiente (“Título ejemplo” para para <title>)
    - Y las instrucciones propias de xslt, que son las que comenzarán con el prefijo declarado en el namespace y que en este caso es ‘xsl:'

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - Es decir, **todo lo que está dentro de la plantilla forma la salida o resultado**, trasladando todos los elementos normales o texto tal cual están, mientras que los que tienen el prefijo 'xsl:' indicarán al procesador que se debe hacer algún tratamiento
  - En nuestro ejemplo simple anterior hay una única plantilla, pero una hoja de estilo puede tener todas las plantillas que se considere necesario
  - **Su funcionamiento esta basado en el recorrido en forma de árbol de los nodos que coinciden con la expresión XPath** de las sucesivas plantillas, incorporándolos en el documento de destino

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - Ejemplo de aplicación de plantillas

Documento xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type="text/xsl" href=class.xsl"?>
<class>
   <student rollno = "393">
      <firstname>Dinkar</firstname>
      <lastname>Kad</lastname>
      <nickname>Dinkar</nickname>
      <marks>85</marks>
   </student>
   <student rollno = "493">
      <firstname>Vaneet</firstname>
      <lastname>Gupta</lastname>
      <nickname>Vinni</nickname>
      <marks>95</marks>
   </student>
   <student rollno = "593">
      <firstname>Jasvir</firstname>
      <lastname>Singh</lastname>
      <nickname>Jazz</nickname>
      <marks>90</marks>
   </student>
</class>
```

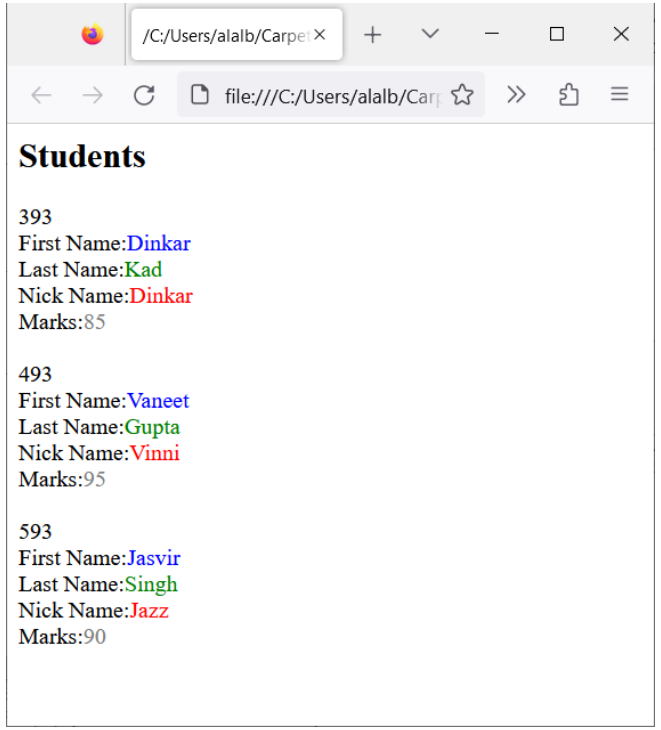
Documento xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
   xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
   <xsl:template match = "/">
      <html>
         <body>
            <h2>Students</h2>
            <xsl:apply-templates select = "class/student" />
         </body>
      </html>
   </xsl:template>

   <xsl:template match = "class/student">
      <xsl:apply-templates select = "@rollno" />
      <xsl:apply-templates select = "firstname" />
      <xsl:apply-templates select = "lastname" />
      <xsl:apply-templates select = "nickname" />
      <xsl:apply-templates select = "marks" />
      <br />
   </xsl:template>

   <xsl:template match = "@rollno">
      <span style = "font-size = 22px;">
         <xsl:value-of select = "." />
      </span>
      <br />
   </xsl:template>

   <xsl:template match = "firstname">
      First Name:<span style = "color:blue;">
         <xsl:value-of select = "." />
      </span>
      <br />
   </xsl:template>
   ...
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - Ejemplo de aplicación de plantillas – Resultado

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <body><h2>Students</h2><span style="font-size = 22px;">393</span><br/>
      First Name:<span style="color:blue;">Dinkar</span><br/>
      Last Name:<span style="color:green;">Kad</span><br/>
      Nick Name:<span style="color:red;">Dinkar</span><br/>
      Marks:<span style="color:gray;">85</span><br/><br/><span style="font-size = 22px;">493</span><br/>
      First Name:<span style="color:blue;">Vaneet</span><br/>
      Last Name:<span style="color:green;">Gupta</span><br/>
      Nick Name:<span style="color:red;">Vinni</span><br/>
      Marks:<span style="color:gray;">95</span><br/><br/><span style="font-size = 22px;">593</span><br/>
      First Name:<span style="color:blue;">Jasvir</span><br/>
      Last Name:<span style="color:green;">Singh</span><br/>
      Nick Name:<span style="color:red;">Jazz</span><br/>
      Marks:<span style="color:gray;">90</span><br/><br/></body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Plantillas
  - Si se incluye una **plantilla vacía** para un nodo de contexto el resultado es que no se muestra información para ese nodo de contexto:

```
<xsl:template select=”expresión XPath”/>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Xpath, patrones para la selección de información
  - **Los patrones pueden ser muy complejos** y como ya se ha indicado **se especifican en un lenguaje llamado XPath**. Es un lenguaje de consulta usado para identificar y seleccionar nodos (elementos) de un documento XML. Se caracteriza por:
    - Ser **declarativo** (vs. Procedimental o imperativo)
    - Ser **contextual**, los resultados dependen del nodo "actual"
  - Admite expresiones comunes: operadores de comparación, lógicos y matemáticos (=, <, and, or, *, +, etc.)

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Xpath, patrones para la selección de información
  - En la siguiente tabla se incluyen algunos de los operadores empleados más habitualmente para formar los patrones:

Operador	Descripción
/	Si está al principio de la expresión, indica el nodo raíz, si no, indica "hijo" (únicamente descendientes directos).
//	Selección de descendientes. Selecciona todos los descendientes.
.	Selección del elemento actual (el contexto).
*	Todos (en el sentido habitual de este operador)
@	Prefijo que se antepone al nombre de un atributo. Especificar /@ después de referencia de nodo si se utiliza con referencia de nodo
[ ]	Filtro sobre el conjunto de nodos seleccionado.

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Xpath, patrones para la selección de información
  - Algunos ejemplos:
    - ./AUTOR → Selecciona todos los elementos AUTOR dentro del contexto actual (todos los hijos del nodo actual que tengan como etiqueta AUTOR)
    - /LIBROS → Selecciona los elementos (posiblemente uno solo) con etiqueta LIBROS que cuelgan directamente del elemento raíz
    - //AUTOR → Selecciona todos los elementos AUTOR en cualquier parte del documento
    - /LIBROS[@TEMA="XML"] → Selecciona todos los elementos LIBROS que cuelgan directamente de la raíz de la raíz (los seleccionados antes) y de éstos selecciona únicamente aquellos que tengan el valor XML en el atributo TEMA

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Además de los elementos ya tratados en los puntos anteriores para explicar las declaraciones de documento y las plantillas, existen otros elementos pertenecientes a la tecnología XSLT de los que veremos una selección de los mismos correspondiente a los mas utilizados
  - Si consideramos otro ejemplo también simple que incluimos en el fichero ejemplo_contactos.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="ejemplo2.xsl" type="text/xsl"?>
<grupo>
    <nombre>Jose</nombre>
    <nombre>Ana</nombre>
    <nombre>Luis</nombre>
    <nombre>María</nombre>
</grupo>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:apply-templates>
    - **Se utiliza desde dentro de una plantilla para llamar a otras**, y si no hay otras se aplica la **plantilla por defecto** que lo que hace es incluir el contenido de las etiquetas en el documento de salida
    - Su sintaxis básica es:

      <xsl:apply-templates select=”expresión XPath”>

    - Si se especifica el atributo select, entonces se evalúa la expresión y el resultado se utiliza como nodo de contexto, y si no es especifica, se asume el nodo contexto vigente

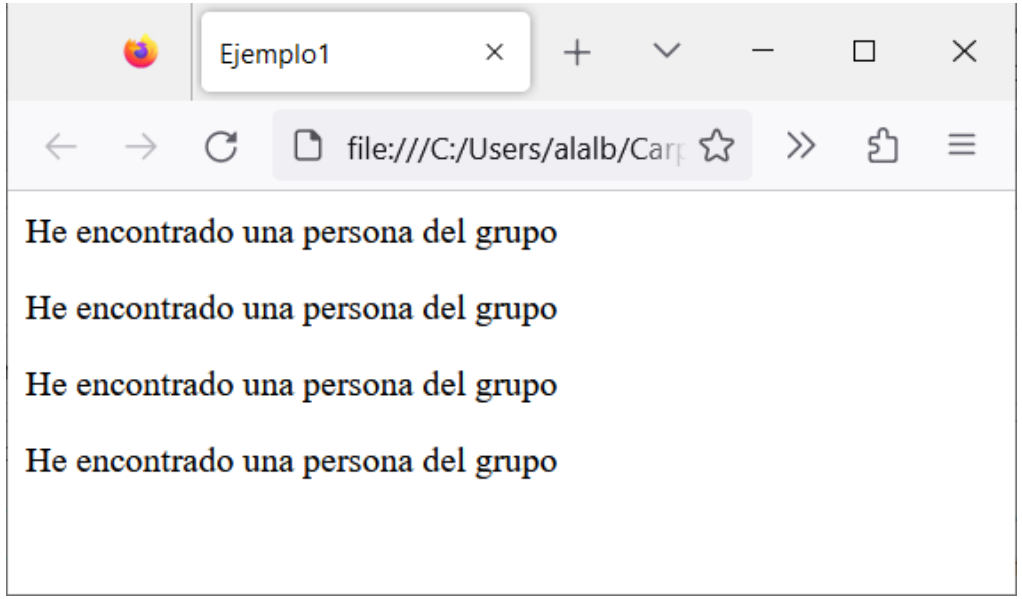
CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:apply-templates>
    - Si se crea el fichero xsl para el documento xml anterior, en el que se utiliza una plantilla para los elementos principales y otra plantilla para los elementos <nombre>

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
    <html>
            <head>
                    <title>Ejemplo1 </title></head>
            <body>
                    <xsl:apply-templates />
            </body>
    </html>
</xsl:template>
<xsl:template match="nombre">
<p> He encontrado una persona del grupo</p>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:apply-templates>
    - Si se incluye sin argumentos hace referencia a todos los hijos del nodo que se esté considerando
    - Para el fichero xsl de ejemplo considerado se obtien el resultado que se muestra en la imagen:
      - La primera plantilla genera la estructura de la página, con el título en <head> y el cuerpo principal de <body>
      - A continuación se aplica la siguiente plantilla, que crea un párrafo cada vez que se encuentra con el elemento <nombre> y escribe el texto previsto para este caso



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:value-of>
    - **Se utiliza para obtener el contenido de los nodos**, incluidos sus atributos
    - Su sintaxis básica es:

      <xsl:value-of select=”expresión XPath”>

    - El valor del atributo select es una expresión Xpath
      - La expresión Xpath utilizada puede servir para devolver tanto el valor del texto asociado al nodo como el de un atributo del nodo (para hacer referencia a un atributo deber ir precedido de @)

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

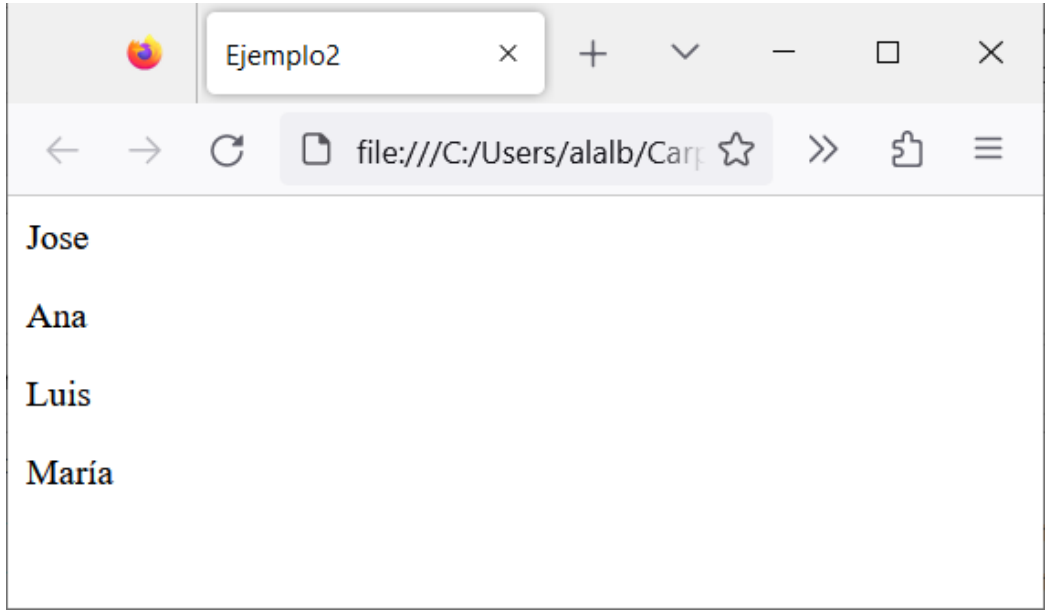
- Elementos XSLT
  - Elemento <xsl:value-of>
    - En el ejemplo anterior si se cambia

```
<p> He encontrado una persona del grupo</p>
```

Por:

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

Se obtiene el resultado que se muestra en la figura

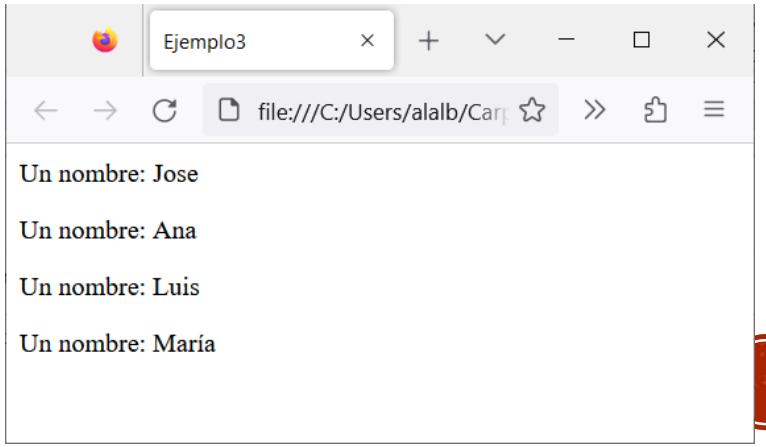


CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:text>
    - **Permite insertar texto**, aunque no es necesario ya que en las plantillas como se ha visto se puede escribir directamente el texto que se desea incluir en el árbol resultado
    - En el ejemplo anterior si se modifica la segunda plantilla de la siguiente forma:

```
</xsl:template>
<xsl:template match="nombre">
<p> <xsl:text > Un nombre: </xsl:text> <xsl:value-of select="." /> </p>
</xsl:template>
```

    - Se obtiene el resultado que muestra la figura



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:element>
    - Aunque se pueden incluir elementos en base a texto libre como se ha visto, el elemento <xsl:element> **permite insertar elementos de una forma más dinámica**. Su sintaxis básica es:

      <xsl:element name=”nombre_elemento”>

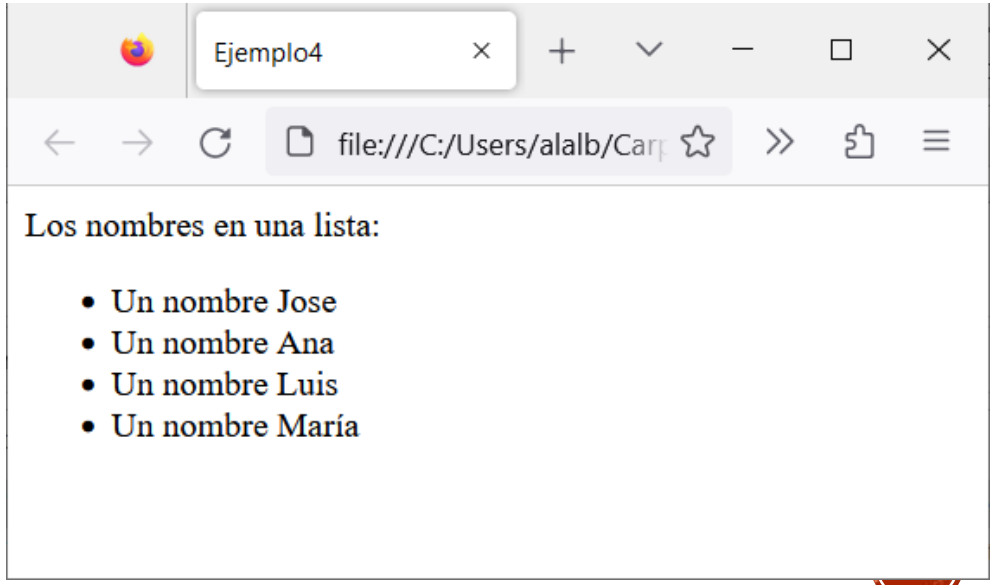
    - Si se realizan modificaciones en el ejemplo anterior:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
     <head><title>Ejemplo4 </title></head>
     <body>Los nombres en una lista:
          <xsl:element name="ul" > <xsl:apply-templates /> </xsl:element>
     </body>
</html>
</xsl:template>

<xsl:template match="nombre">
     <xsl:element name="li"> <xsl:text > Un nombre </xsl:text>
          <xsl:value-of select="." />
     </xsl:element>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:element>
    - Se obtiene un código html que se puede visualizar como se muestra en al figura y en el que se puede ver que se han incluido los elementos especificados

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo4 </title>
  </head>
  <body>Los nombres en una lista:
<ul>
<li> Un nombre Jose</li>
<li> Un nombre Ana</li>
<li> Un nombre Luis</li>
<li> Un nombre María</li>
</ul></body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:attribute>
    - Aunque también se pueden incluir atributos en base a texto libre como se ha visto, el elemento **permite insertar atributos de una forma más dinámica en el elemento en el que se incluye**. Su sintaxis básica es:

      <xsl:attribute name=”nombre_atributo”>

    - Si se realizan modificaciones en el ejemplo1 inicial:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
     <head>
               <title>Ejemplo 5 </title></head>
     <body>
     <xsl:apply-templates />
     </body>
</html>
</xsl:template>
<xsl:template match="nombre">
     <p><xsl:attribute name="class"> destacado </xsl:attribute>
     He encontrado una persona del grupo
     </p>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:attribute>
    - Se obtiene el siguiente código HTML, en el que tenemos elementos (en este caso párrafos HTML <p>) con el atributo "class" con valor "destacado", para que sea tenido en cuenta en una hoja de estilos CSS por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo 5 </title>
  </head>
  <body>
<p class=" destacado ">
He encontrado una persona del grupo
</p>
<p class=" destacado ">
He encontrado una persona del grupo
</p>
<p class=" destacado ">
He encontrado una persona del grupo
</p>
<p class=" destacado ">
He encontrado una persona del grupo
</p>
</body>
</html>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:copy-of>
    - Este elemento es **útil cuando se pretende transformar un XML en otro documento también XML**, si hay secciones largas en las que el contenido es exactamente igual, esta instrucción **permite tomar fragmentos del documento origen y traspasarlos íntegramente al destino, sin tener que crear los elementos**. Su sintaxis básica es:

      <xsl:copy-of select=”expresión XPath”>

    - Por ejemplo la siguiente transformación crea un documento XML idéntico al fichero XML origen:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match='/'>
            <xsl:copy-of select='.'/>
    </xsl:template>
</xsl:stylesheet>
```

    - Este elemento tiene más sentido cuando se utiliza dentro de elementos condicionales

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:copy>
    - Este elemento a diferencia del anterior **realiza una copia "hueca" que solo copia la etiqueta del elemento, mientras que los atributos y nodos hijos no se copian**
    - No tiene atributo select ya que siempre copia el elemento actual dentro de un "template" en un bucle o elemento de control. Además resulta útil cuando se precisa transformar un elemento a un XML similar pero eliminando los atributos. Su sintaxis básica es:

      <xsl:copy />

      O bien

      <xsl:copy> </xsl:copy>

    - Si se incluye como contenido del elemento <xsl:value-of select=".”> se obtiene el contenido del nodo de contexto más el de todos los nodos descendientes

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:copy>
    - Si en al ejemplo que se está considerando se le aplica un xsl de transformación:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:template match='/'>
                <xsl:apply-templates />
        </xsl:template>
<xsl:template match="grupo">
  <xsl:copy>
    <todos_mis_contactos>
      <xsl:value-of select="."/>
    </todos_mis_contactos>
  </xsl:copy>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos XSLT
  - Elemento <xsl:copy>
    - **Se obtiene otro fichero xml** de contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<grupo>
  <todos_mis_contactos>
Jose
Ana
Luis
María
</todos_mis_contactos>
</grupo>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - **En XSLT se pueden usar filtros y estructuras que facilitan las operaciones de control del contenido de los documentos**. Estas operaciones de control son habituales en los lenguajes de programación tradicionales, y se conocen como:
    - **Iteraciones** (o repeticiones o bucles)
    - **Selecciones** (o alternativas o condicionales)

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:for-each>, iteraciones
    - Este elemento **se usa para repetir la búsqueda de los nodos que coinciden con la expresión Xpath que se usa en el atributo select**, de forma que solo escribimos una vez el código que comprende la instrucción, pero se aplica a todos los casos en que se cumpla la expresión
    - Su sintaxis es:

```
<xsl:for-each select=”expresión XPath”>
…..
</xsl:for-each>
```

    - Permite en algunos casos simplificar el fichero de transformación al simplificar el flujo de llamada de plantillas y su definición

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

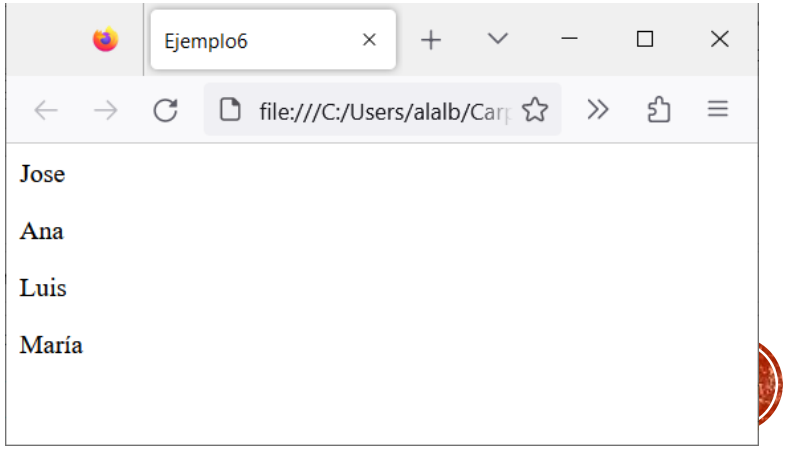
- Elementos de control XSLT
  - Elemento <xsl:for-each>, iteraciones
    - Si se considera el ejemplo1 inicial y se realizan modificaciones en el fichero de transformación xsl (ejemplo2):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
      <head>
                  <title>Ejemplo6 </title></head>
      <body>
                  <xsl:for-each select="grupo/nombre">
                              <p> <xsl:value-of select="." /> </p>
                  </xsl:for-each>
      </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:for-each>, iteraciones
    - En el que simplemente se ha sustituido la segunda plantilla por la estructura for-each completa que ya contiene el elemento seleccionado con value-of
    - Es decir, el contenido de <xsl:foreach…> hace la función de una plantilla anidada en otra anterior, con lo que se simplifica el documento
    - Se obtiene un resultado es idéntico al obtenido en el ejemplo 2 anterior:

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo6 </title>
  </head>
  <body>
    <p>Jose</p>
    <p>Ana</p>
    <p>Luis</p>
    <p>María</p>
  </body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:if test>, condicional
    - Es un elemento que permite decidir si una acción se realiza o no dependiendo de ciertos criterios, es decir, **permite generar contenido de forma condicional**
    - Su sintaxis básica es:

```
<xsl:if test="expression Xpath">
.....
</xsl:if>
```

    - El atributo test es obligatorio y contiene la condición que se tiene que cumplir utilizando sintaxis XPath
    - Este elemento **únicamente permite condiciones simples**, no alternativas como las existentes en los lenguajes de programación

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

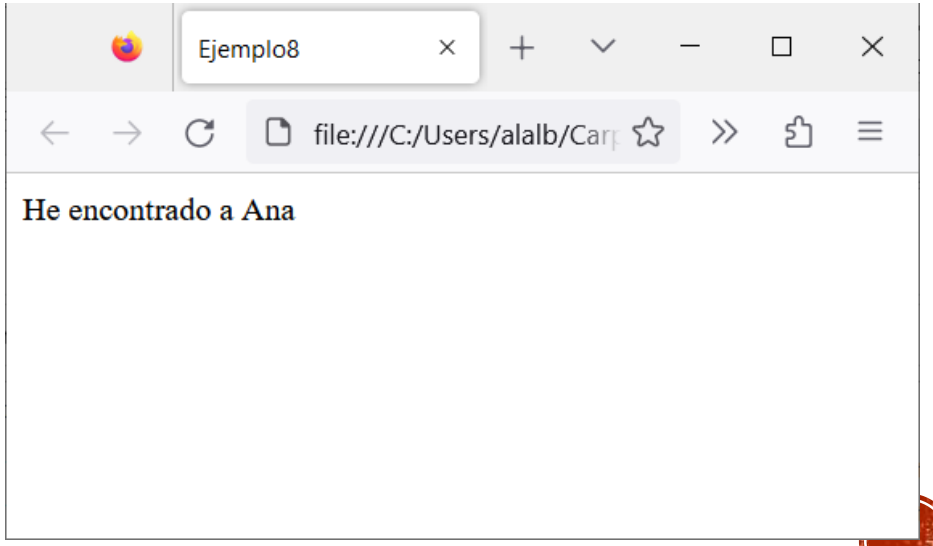
- Elementos de control XSLT
  - Elemento <xsl:if test>, condicional
    - Un ejemplo sencillo de su uso, considerando el fichero xml de los casos anteriores ejemplo_contactos.xml y aplicándole el siguiente fichero de transformación:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
     <head>
               <title>Ejemplo8</title></head>
     <body>
               <xsl:for-each select="grupo/nombre">
                         <xsl:if test=".='Ana'">
                                   <p>He encontrado a <xsl:value-of select="." /> </p>
                         </xsl:if>
               </xsl:for-each>
     </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:if test>, condicional
    - Notar que **se han tenido que alternar las comillas ' y “ para contemplar tanto las correspondientes a la expresión propia de la condición de test como a la del XPath** que contiene, ya que deben ser distintas al anidar unas comillas dentro de otras
    - Se obtiene la siguiente salida

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo8</title>
  </head>
  <body>
    <p>He encontrado a Ana</p>
  </body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:choose>, condicional
    - Este elemento con sus elementos secundarios <xsl:when> y <xsl:otherwise> **se corresponde con la estructura condicional switch/case** en los lenguajes de programación
    - Esta estructura **permite establecer varias condiciones**. También permite especificar una alternativa en caso de ninguna de éstas se cumpla
    - La estructura **debe comenzar con el elemento <xsl:choose>**
      - El elemento puede contener tantos elementos secundarios **<xsl:when>** como se desee y un único elemento <xsl:otherwise>
      - El elemento <xsl:when> dispone al igual que el elemento <xsl:if> de un atributo obligatorio test. Si se cumple la expresión contenida en el atributo test, se copiará el contenido en el árbol resultado
    - La diferencia más notable entre el elemento <xsl:if> y el <xsl:choose>/<xsl:when> es la posibilidad que ofrece esta última de introducir el elemento **<xsl:otherwise>**. Este elemento entra en acción cuando no se cumple ninguna de las condiciones establecidas por <xsl:when>

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:choose>, condicional
    - Su sintaxis básica es:

```
<xsl:choose>
        <xsl:when test="expresión booleana">
        …
        </xsl:when>
        <xsl:when test="expresión booleana">
        …
        </xsl:when>
        …
        …
        <xsl:otherwise>
        …
        </xsl: otherwise>
</xsl:xhoose>
```

    - El atributo test es obligatorio en los elementos <xsl:when> y contiene la condición que se tiene que cumplir utilizando sintaxis XPath
    - En esta estructura se pueden tener tantas condiciones como sean necesarias como en los lenguajes de programación

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

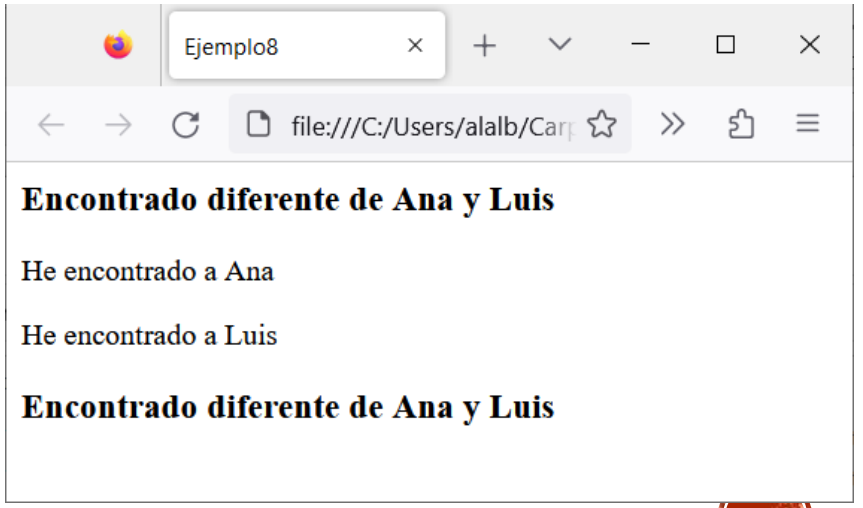
- Elementos de control XSLT
  - Elemento <xsl:choose>, condicional
    - Un ejemplo sencillo de su uso, considerando el fichero xml de los casos anteriores ejemplo_contactos.xml y aplicándole el siguiente fichero de transformación:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
      <head>
            <title>Ejemplo8 </title>
      </head>
      <body>
            <xsl:for-each select="grupo/nombre">
                  <xsl:choose>
                        <xsl:when test=".='Ana'">
                              <p>He encontrado a
                              <xsl:value-of select="." /> </p>
                        </xsl:when>
                        <xsl:when test=".='Luis'">
                              <p>He encontrado a
                              <xsl:value-of select="." /> </p>
                        </xsl:when>
                        <xsl:otherwise>
                              <h3>Encontrado diferente de Ana y Luis </h3>
                        </xsl:otherwise>
                  </xsl:choose>
            </xsl:for-each>
      </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elementos de control XSLT
  - Elemento <xsl:choose>, condicional
    - Se obtiene la siguiente salida

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo8 </title>
  </head>
  <body>
    <h3>Encontrado diferente de Ana y Luis </h3>
    <p>He encontrado a
Ana</p>
    <p>He encontrado a
Luis</p>
    <h3>Encontrado diferente de Ana y Luis </h3>
  </body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elemento de ordenación XSLT
  - Elemento <xsl:sort>
    - Este elemento **se usa para alterar el orden de presentación de los elementos**, siguiendo un criterio de clasificación indicado en el atributo select
    - **Se añade anidado dentro de un elemento <xsl:foreach....> como primer hijo de ese elemento**
    - Su sintaxis básica es:

```
<xsl:sort select="expresión XPath" order="ascending|descending"/>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

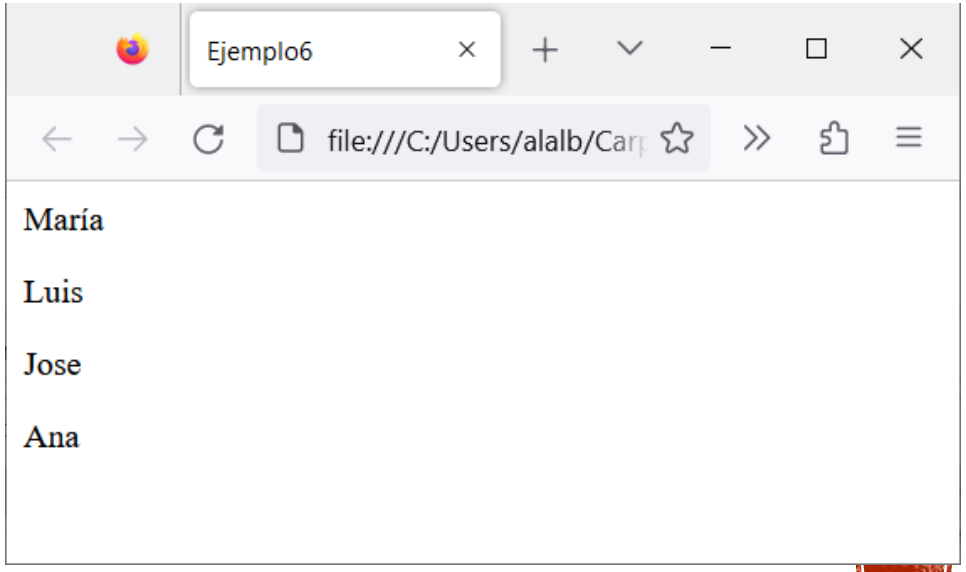
- Elemento de ordenación XSLT
  - Elemento <xsl:sort>
    - Un ejemplo sencillo de su uso, considerando el fichero xml de los casos anteriores, ejemplo_contactos.xml, y partiendo de la transformación del ejemplo6:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match='/'>
<html>
      <head>
                  <title>Ejemplo6 </title></head>
      <body>
                  <xsl:for-each select="grupo/nombre">
                              <xsl:sort select="." order="descending"/>
                              <p> <xsl:value-of select="." /> </p>
                  </xsl:for-each>
      </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

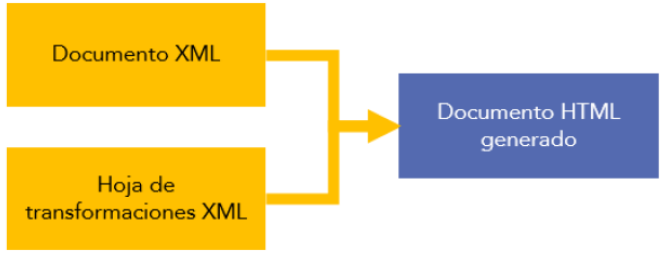
- Elemento de ordenación XSLT
  - Elemento <xsl:sort>
    - Se obtiene la siguiente salida

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <head>
    <title>Ejemplo6 </title>
  </head>
  <body>
    <p>María</p>
    <p>Luis</p>
    <p>Jose</p>
    <p>Ana</p>
  </body>
</html>
```



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elaboración de documentación
  - Como hemos visto hasta ahora mediante XSLT se puede generar código HTML



  - Mediante transformación XSLT se puede generar un nuevo documento XML a partir de un documento XML

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- Elaboración de documentación
  - Esta transformación se consigue mediante el uso de:
    - Elementos ya vistos que permiten la generación de elementos nuevos o elementos a partir de otros elementos: <xsl:element>, <xsl:attribute>, <xsl:copy-of>, <xsl:copy>
    - Y otros elementos como: <xsl:comment>, <xsl:processing-instruction>, …
  - Mediante el elemento <xsl:output> y su atributo method (que puede tomar los valores xml, html, text) se indica si la transformación se realiza a XML o a HTML
    - El valor por defecto es XML, pero si el primer elemento que se especifica que no sea un elemento xsl (es decir, que no lleva el prefijo xsl:) es <html> la transformación se realiza a html

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- **Visualización** en el navegador de documentos XML con **hojas de estilo CSS**
  - **Las hojas de estilos CSS** son archivos destinados a dar formato a las páginas web. Normalmente se enlazan a los archivos HTML, aunque **también pueden aplicarse a un documento XML** y visualizarse en el navegador
  - El W3C aprobó en junio de 1999 la recomendación que definen cómo vincular documentos XML con hojas de estilo CSS
    - De acuerdo con esta recomendación mediante la instrucción <?xml-stylesheet ?> de forma similar a como se hace en una página web XHTML con la etiqueta <link />
    - En ambos casos el atributo href incluye el camino absoluto o relativo a la hoja de estilo CSS

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

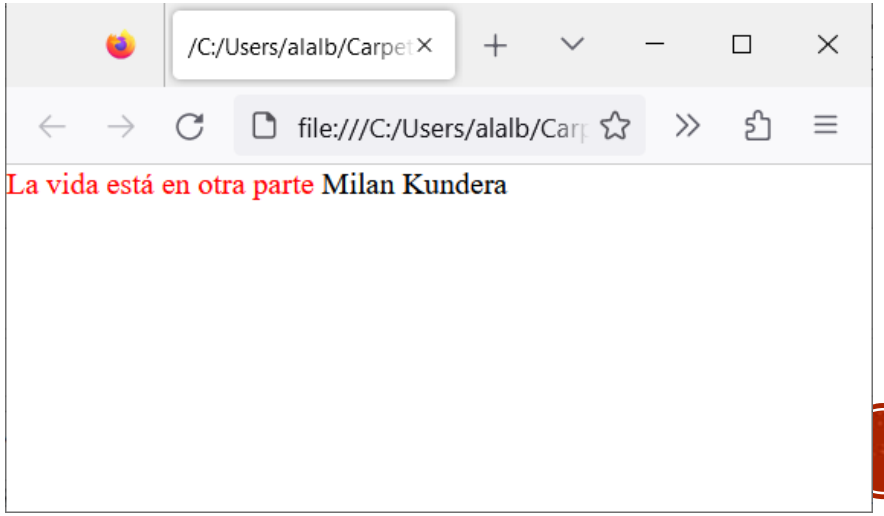
- Visualización en el navegador de documentos XML con hojas de estilo
  - Ejemplo:
    - Documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="ejemplo.css"?>
<libro>
  <titulo>La vida está en otra parte</titulo>
  <autor>Milan Kundera</autor>
  <fechaPublicacion año="1973"/>
</libro>
```

    - Hoja de estilo CSS:

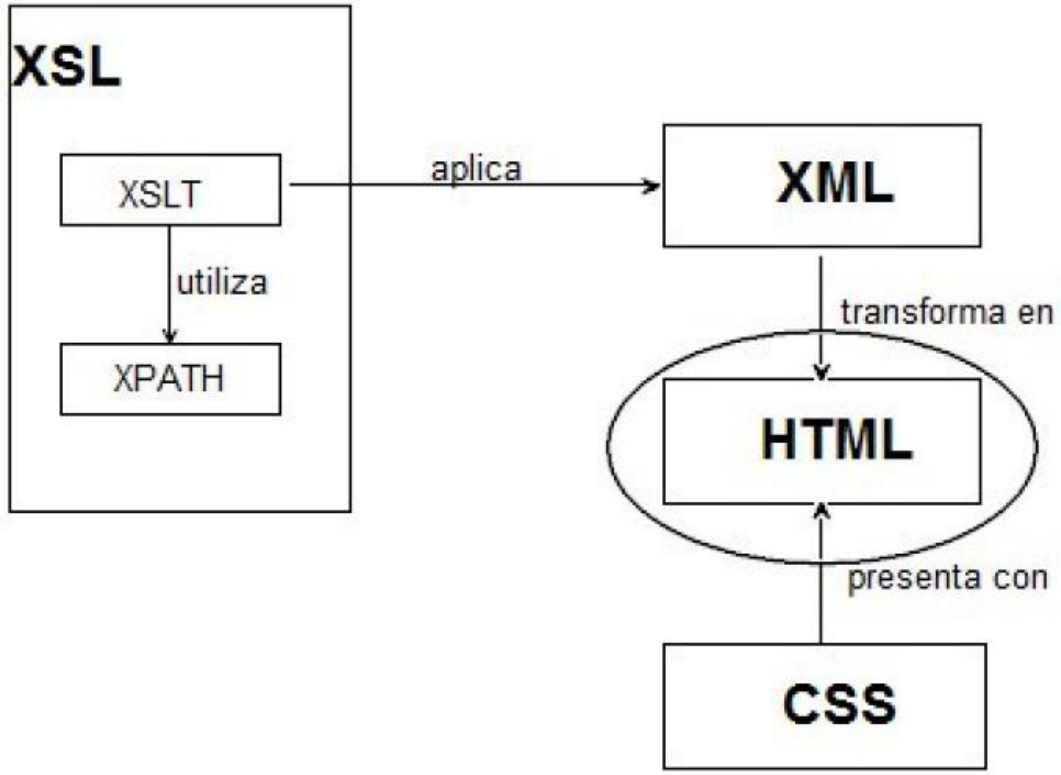
```
titulo {
  color: red;
}
```

    - El resultado se muestra en la figura



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- **XSL permite definir hojas de estilo más adecuadas para los documentos XML** que CSS (que utiliza un método de presentación adaptado al mundo HTML)
- Esto no quiere decir que XSL sustituya a las CSS
- **En la práctica XSL, HTML y CSS son complementarios**, de forma que el proceso habitual consiste en que **los datos de los documentos XML se utilizan en una página Web mediante su transformación a HTML con XSLT** y la ayuda de XPATH, **para finalmente presentar el documento HTML utilizando CSS**, como se muestra en la figura



CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSL-FO
  - El XSL-FO es **eXtensible Stylesheet Language Formatting Objects es un lenguaje de marcas basado en XML que forma parte de la familia de lenguajes XSL**
  - **Permite describir la forma de mostrar un conjunto de datos en un soporte determinado** (papel, pantalla de dispositivo electrónico concreto, …)
  - Con XSL-FO se definen aspectos como las **características de las pá**ginas (dimensiones, márgenes, secciones, …) **y los elementos que incluyen** (párrafos, tablas, …)
  - Para su uso es necesario disponer de un programa que permita procesar XSL-FO, la herramienta más utilizada para hacerlo es el **Apache FOP** (software libre)
  - Los ficheros XSL-FO tienen extensión .fo y .xml
  - Permite generar documentos en diferentes formatos como PDF, Word, PNG, PS (PostScript), PCL (Printer Command Language), etc.

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSL-FO
  - Existen dos formas de usarlo:
    - **Datos incluidos en el propio documento XSL-FO**
      - En este caso participa un único fichero en el proceso de transformación (el .fo o .xml que contiene las reglas de transformación y los datos)
    - **Datos en un documento XML del que se obtienen utilizando XSLT / Xpath** para conseguir la información del documento xml
      - En este caso participan dos ficheros:
        - Los datos a presentar se incluyen en un fichero xml de entrada
        - Las reglas de transformación se incluyen en un fichero XSLT (extensión .xsl)

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSL-FO
  - El proceso de transformación es muy sencillo
    - En el caso de que los datos estén incluidos en el documento XSL-FO la sintaxis con Apache FOP:

```
fop fichero_fo -tipo_formato_salida nom_fichero_salida
```

      - Ejemplo:

```
fop pedido20098.fo -pdf pedido20098.pdf
```

    - En el caso de que los datos estén en un documento XML distinto del que contiene las reglas XSL-FO (ficehro .xsl) la sintaxis con Apache FOP:

```
fop -xml fichero_xml -xsl fichero_xsl -tipo_formato_salida nom_fichero_salida
```

      - Ejemplo:

```
fop -xml pedido20098.xml -xsl formato_pedidos.xsl -pdf pedido20098.pdf
```

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

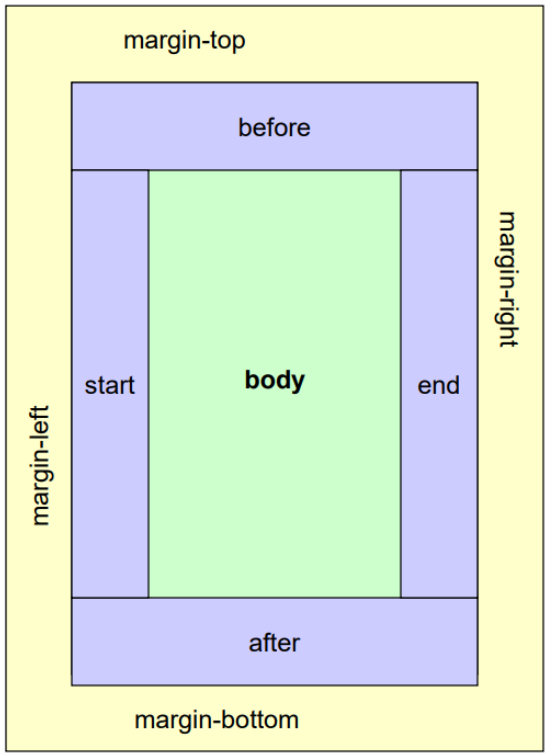
- XSL-FO
  - Los ficheros para la transformación XSL-FO deben incluir en su definición el name space XSL/Format (normalmente asociado al prefijo fo:):

    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

    - Que permite el uso de los **diferentes elementos del lenguaje** para implementar las reglas de transformación y que permiten definir:
      - La **configuración de páginas** (<simple-page-master>):
        - Se pueden definir varias **que pertenecen a un mismo diseño** (<layout-master-set>)
        - Se especifica un nombre, dimensiones (alto, ancho) y márgenes
        - **Se pueden especificar regiones** (cuerpo, cabecera, pie, lateral izquierdo y lateral derecho)
      - **Contenedores de objetos de página** (<page-sequence>), que pueden ocupar varias páginas del dispositivo si el tamaño del contenido lo requiere:
        - Incluyen elementos (<flow> y <static-content>) para indicar el formato de la información y especificar el contenido propiamente dicho

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

- XSL-FO
  - Esquema <simple-page-master>):



Atributos:

- master-name
- page-height
- page-width
- margin-left, margin-top, margin-right, margin-bottom

Ejemplo:

```
<fo:layout-master-set>
  <fo:simple-page-master
          master-name="pagina-normal"
          page-height="29.7cm"
          page-width="21cm"
          margin-left="3cm"
          margin-right="3cm"
          margin-top="2cm"
          margin-bottom="2cm">
  </fo:simple-page-master>
</fo:layout-master-set>
```