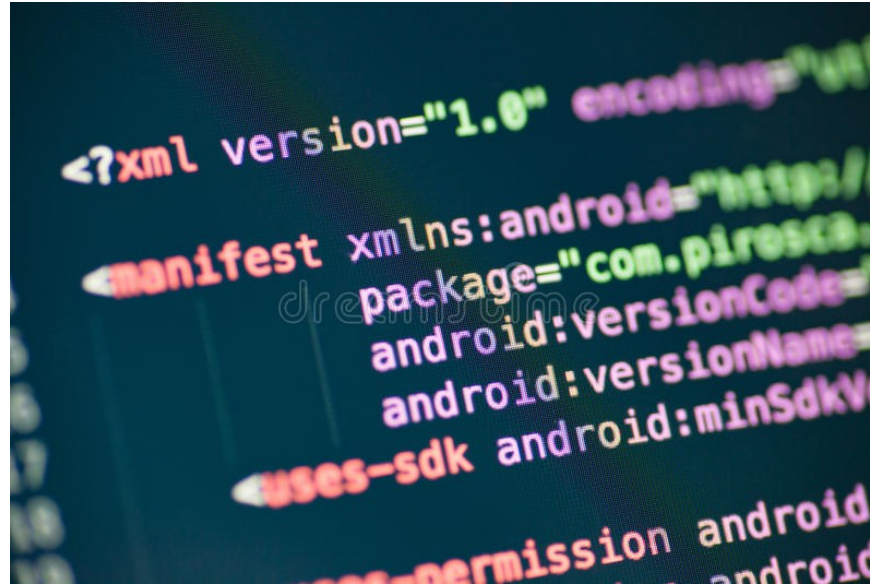


# Lenguajes de marcas y sistemas de gestión de la información



## UT06 – Conversión de documentos

### 2 – XSLT – Plantillas

# Múltiples plantillas en la misma XSLT

En una hoja XSLT podemos tener más de una plantilla, lo que permite dividir / organizar la XSLT de forma que sea más fácil de leer y mantener.

Similar a la forma en la que se crean tipos personalizados en XSD.

Habitualmente, se parte de una plantilla que trata el nodo raíz y se añaden plantillas para hacer más modular la XSLT.

Se usa `<xsl:apply-templates/>` para aplicar plantillas al elemento actual o a sus hijos.

Apply-templates puede usar el atributo “select” para filtrar los elementos a los que se aplican plantillas.

# Múltiples plantillas - Ejemplo A

En el ejemplo biblioteca-a.xslt, tenemos:

- Una plantilla “general” que se aplica a “/”
- Una plantilla para los elementos “libro”
- Una plantilla para los elementos “anio”

Se usa `<xsl:apply-templates />` para aplicar las plantillas en los puntos en que lo necesitamos.

Es necesario llamar a `<xsl:apply-templates />` dentro de cada plantilla para llamar a las “subplantillas”. Si no, no se aplican las plantillas en los elementos hijos.

Los ejemplos están disponibles en el repositorio de github:

<https://github.com/jllopezalvarez-profesor/DA1D1E-23-24-LLMM>

# Múltiples plantillas - Ejemplo B

En el ejemplo biblioteca-b.xslt, tenemos lo mismo que en el ejemplo A, pero utilizamos el atributo “select” en `<xsl:apply-templates />` para aplicar las plantillas sólo a ciertos elementos:

```
<xsl:apply-templates  
  select="biblioteca/libro[anio = 1986]" />
```

Esto hace que la plantilla se aplique sólo a los elementos seleccionados (libros del año 1986). El resto de los elementos no se procesan.

# Múltiples plantillas - Ejemplo C

En el ejemplo biblioteca-c.xslt, usamos el atributo “mode” para aplicar selectivamente dos plantillas diferentes a los libros:

- Una plantilla para generar un índice
- Otra para el detalle del libro.

Definición de plantillas

```
<xsl:template match="libro">  
<xsl:template match="libro" mode="indice">
```

Invocación de plantillas:

```
<xsl:apply-templates />  
<xsl:apply-templates mode="indice" />
```

# Múltiples plantillas - Orden

Cuando se usa “apply-templates” con “select” se pueden ordenar los elementos antes de aplicar las plantillas.

```
<xsl:apply-templates select="a/b/c">  
  <xsl:sort select="d" />  
</xsl:apply-templates>
```

En xsl:sort se puede usar el atributo “order” para especificar ascendente o descendente:

```
<xsl:apply-templates select="a/b/c">  
  <xsl:sort select="d" order="x" />  
</xsl:apply-templates>
```

# Plantillas con nombre

Una plantilla puede tener un nombre, lo que permite invocarla directamente en cualquier punto usando `<xsl:call-template>`, sin necesidad de que haya ni siquiera un elemento relacionado con la plantilla.

Definición de la plantilla (no tiene match):

```
<xsl:template name="pie">  
    .....  
</xsl:template>
```

Para llamar a la plantilla:

```
<xsl:call-template name="pie"/>
```

# Parámetros en plantillas con nombre

Los parámetros se definen en la plantilla con "xsl:param".

```
<xsl:template name="fCadenas">  
    <xsl:param name="cadena" />  
</xsl:template>
```

Cuando se invoca la plantilla con xsl:call-template, se usa xsl:with-param para pasar el parámetro a la plantilla.

```
<xsl:call-template name="fCadenas">  
    <xsl:with-param name="cadena" select="---"/>  
</xsl:call-template>
```

Podemos usar el parámetro anteponiendo "\$" a su nombre

```
<xsl:value-of select="$cadena"/>
```



# Parámetros en plantillas con nombre

Cuando se llama a una plantilla y se pasa un parámetro, se puede:

- Usar una expresión XPath en el atributo "select" para indicar de donde se extrae el valor del parámetro:

```
<xsl:call-template name="fCadenas">  
  <xsl:with-param name="cadena"  
select="./hijo/@atributo" />  
</xsl:call-template>
```

- Usar un valor fijo

```
<xsl:call-template name="fCadenas">  
  <xsl:with-param name="cadena">  
    Cadena parámetro  
  </xsl:with-param>  
</xsl:call-template>
```

# Múltiples plantillas - Ejemplo D

El ejemplo D tiene una plantilla para el pie de página, a la que se pasa un parámetro. Se llama con "xsl:call-template", y el parámetro se especifica con "xsl:with-param". Usamos \$ para mostrar el valor del parámetro.

Definición de la plantilla y el parámetro:

```
<xsl:template name="pie">
  <xsl:param name="nombreIES"></xsl:param>
  <footer><p><xsl:value-of select="$nombreIES"
/></p></footer>
```

Invocación de la plantilla:

```
<xsl:call-template name="pie">
  <xsl:with-param name="nombreIES"
select="biblioteca/@nombreIES"></xsl:with-param>
</xsl:call-template>
```