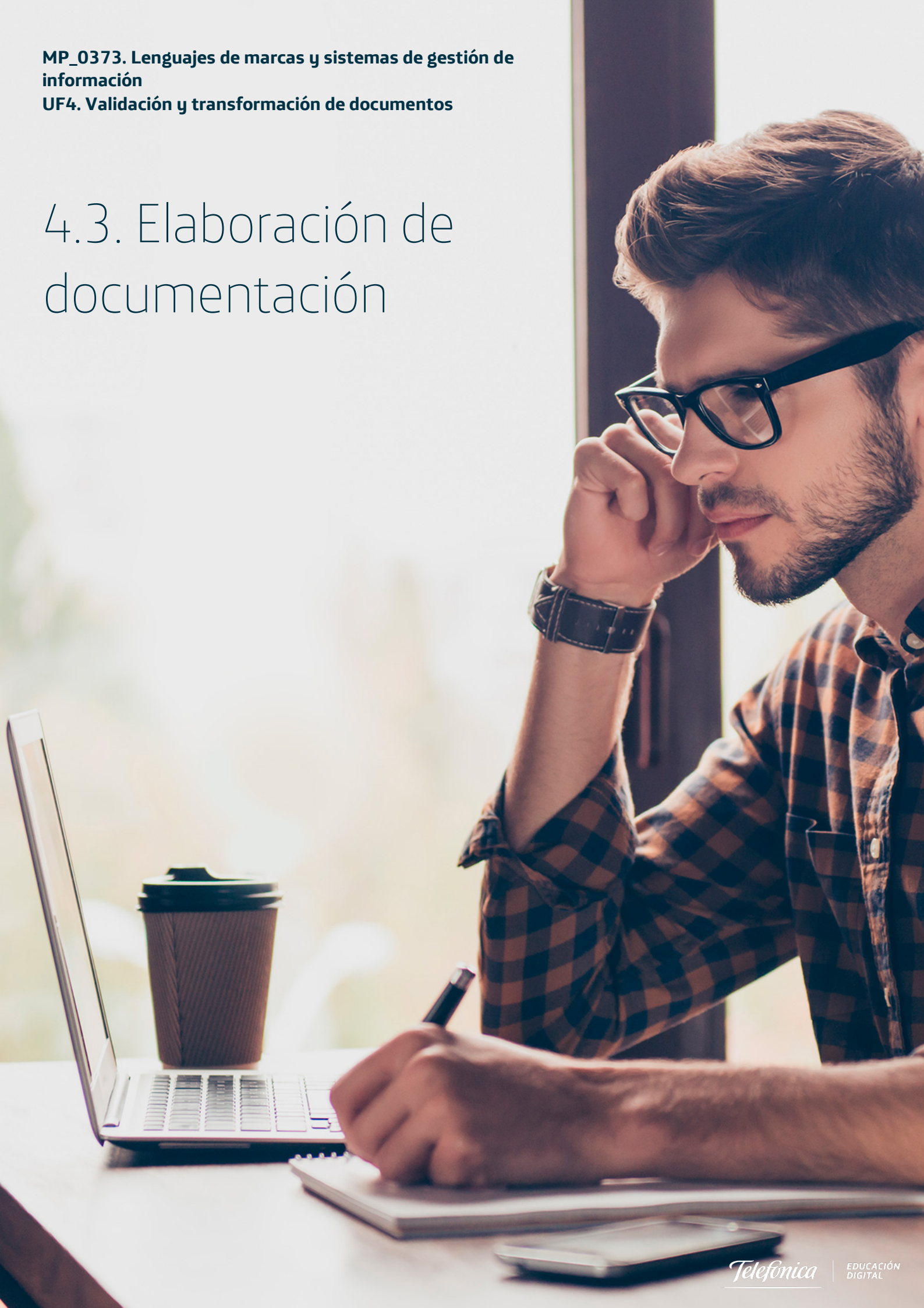


## 4.3. Elaboración de documentación



# Índice

---

Objetivos .....	3
Introducción .....	4
XML para elaborar documentación .....	4
Documentación en la realización de software.....	4
Base de datos y plantillas XSL .....	4
Transformación a documentos .....	5
Generación de noticias .....	5
Formatting objects XSL-FO .....	7
XSL-FO.....	7
Estructura.....	7
Ejemplo de página inicial XSL-FO.....	8
Jerarquía de elementos XSL-FO.....	9
Páginas.....	10
Secuencias de páginas.....	11
Tipos de contenido .....	11
Cajas.....	12
Contenedor.....	12
Bloque.....	12
Línea.....	13
Inclusiones.....	13
Ejemplo.....	14
Imágenes .....	15
Listas .....	15
Enlaces .....	16
Despedida .....	17
Documentación y ejemplos.....	17
Resumen.....	17

# Objetivos

En esta lección perseguimos los siguientes objetivos:

- Conocer los usos para la generación de documentación en base a XML.
- Conocer el lenguaje Formatting Objects XSL-FO.
- Manejar la sintaxis de XSL-FO.

# Introducción

## XML para elaborar documentación

El uso de XML para elaborar documentos es una forma de creación de documentación bastante estandarizada en muchos contextos y entidades.

Los escenarios más comunes en este tipo de configuraciones son:

- Generación de documentación en la **realización de software**.
- Generación de documentos con origen en **base de datos** y **plantillas XSL**.
- Transformación a **documentos electrónicos** en base a plantillas.
- Generación de noticias en base a **XMLNews**.

## Documentación en la realización de software

En muchos sistemas de generación de código fuente se permite trasvasar la información del proyecto y de dicho código a formato XML estandarizado, de forma que la documentación de dicho código siga unas estructuras prefijadas.

Las aplicaciones **actualmente necesitan de mucha documentación**, explicando los procesos que realizan o sus métodos. Para ello los entornos de desarrollo disponen de herramientas que facilitan la creación de estos textos de forma ordenada. Normalmente usan los comentarios escritos por los desarrolladores, que con anotaciones especiales generan esta documentación.

Un ejemplo es la creación de documentación de aplicaciones como Javadoc.

### Uso de JavaDoc.

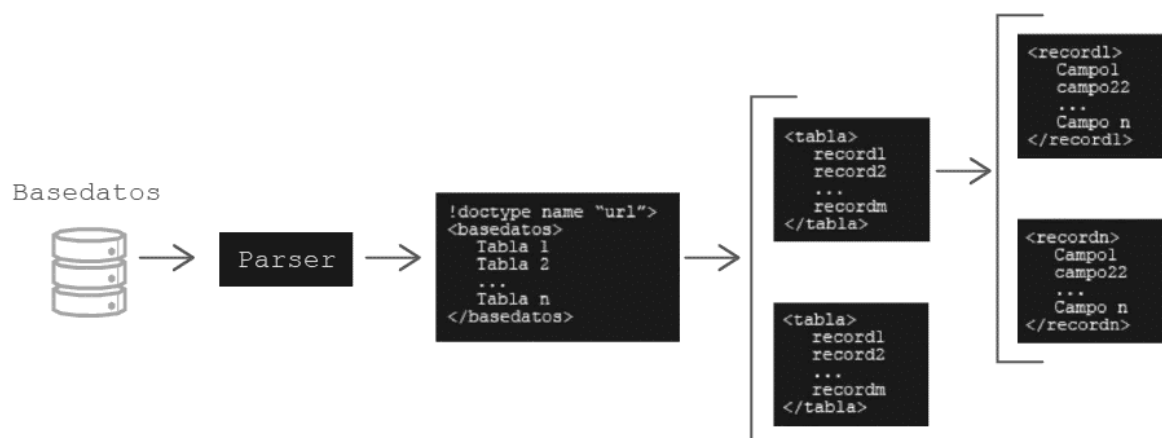
Generador de documentación con Java.

## Base de datos y plantillas XSL

La técnica para el **traspaso desde un modelo relacional** se basará en el desarrollo de un *parser*, que transformará cada registro de la base de datos en un nodo de un elemento XML.

Una **base de datos consiste en un conjunto de tablas**, que a su vez constan de registros, que a su vez se componen de campos.

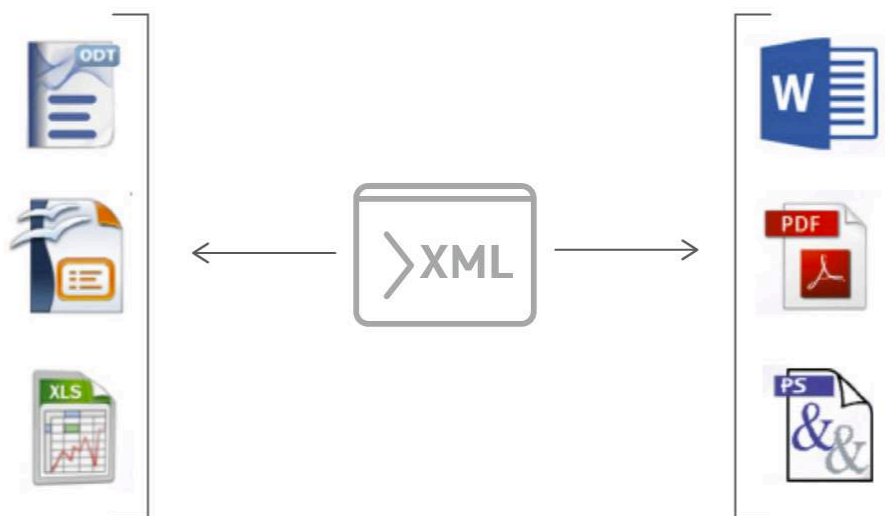
Podemos modelar la base de datos con un nodo de documento y sus nodos de elementos, que serán el reflejo de cada registro y un conjunto de campos para cada registro.



## Transformación a documentos

El uso de plantillas en las transformaciones desde documentos XML hacia otros formatos de documentos permite traspasar la información en formatos de lectura o impresión.

El uso de **XSL-FO** permite obtener documentos imprimibles y con un formato legible amigable.



## Generación de noticias

XMLNews es una especificación para el intercambio de noticias y otra información.

El uso de este estándar facilita, tanto a los productores de noticias como a los consumidores de noticias, producir, recibir y archivar cualquier tipo de información de noticias a través de diferente hardware, software y lenguajes de programación.

```
<?xml version="1.0" encoding="UTF-8"?>
<nitf>
  <head>
    <title>Colombia Earthquake</title>
  </head>
  <body>
    <headline>
      <h1>143 Dead in Colombia Earthquake</h1>
    </headline>
    <byline>
      <bytag>By Jared Kotler, Associated Press Writer</bytag>
    </byline>
    <dateline>
      <location>Bogota, Colombia</location>
      <date>Monday January 25 1999 7:28 ET</date>
    </dateline>
  </body>
</nitf>
```

# Formatting objects XSL-FO

## XSL-FO

**Formatting Objects (FO)** funciona de manera que, aplicado **a un fichero XML**, nos proporciona **otro fichero con la información** de este renderizada a algún formato de **visualización impresa**.

Sus principales características son:

- Es un **lenguaje de marcado de documentos XML** que es usado para generar archivos PDF u otros formatos de impresión y visualización.
- Es **parte de un conjunto de tecnologías W3C** diseñadas para la transformación de datos XML hacia otros formatos.
- La representación de datos a través de ficheros XML permite que puedan ser trasladados a otros formatos como el HTML, WML, etc.
- La principal diferencia entre un FO y otros XSLT es que **FO está pensado para generar directamente una renderización**, aunque ambos transformen documentos XML en otros documentos.
- Los formatos destino más conocidos pueden ser **PDF, PostScript, texto, word, etc.**
- XSL-FO, formalmente llamado XSL, se convirtió en una recomendación del W3C el 15 de octubre de 2001. Las otras partes de XSL son XSLT y XPath. La versión 1.1 de XSL-FO fue publicada en 2006.

## Estructura

**Los ficheros FO comienzan como un documento XML, con su declaración clásica.**

```
<?xml version="1.0" encoding="UTF-8"?>
```

El espacio de nombres se define de esta forma:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:fo="http://www.w3.org/XSL/Format/1.0"
  result-ns="fo">
```

Las propiedades especifican cómo será la renderización.

```
<fo:block font-family="Times, Times New Roman">...</fo:block>
```

Esta es la forma de indicar los tipos de letra que deseamos que tenga el bloque (se comienza por la primera y se continúa hasta encontrar una disponible), siendo *font-family* una propiedad bien conocida por los usuarios de CSS.

### Resumen de objetos de formato XSL-FO

Resumen del contenido y las propiedades de XSL Formatting Objects.



## Ejemplo de página inicial XSL-FO

En este ejemplo podemos ver los elementos principales de un documento XSL-FO, vamos a comentar algunas de sus declaraciones.

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="inicio">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="inicio">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hola Mundo!!</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

El elemento **<fo: root>** es el elemento raíz; declara el espacio de nombres para XSL-FO.

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!-- El contenido del documento va en este nivel-->
</fo:root>
```

El elemento **<fo: layout-master-set>** contiene una o más plantillas de página.

```
<fo:layout-master-set>
  <!-- Las plantillas que se usen irán a este nivel. -->
</fo:layout-master-set>
```

Cada elemento **<fo: simple-page-master>** contiene una plantilla de página única. Cada plantilla debe tener un nombre único (*master-name*).

```
<fo:simple-page-master master-name="plantilla4">
  <!-- Cada plantilla única va a este nivel -->
</fo:simple-page-master>
```

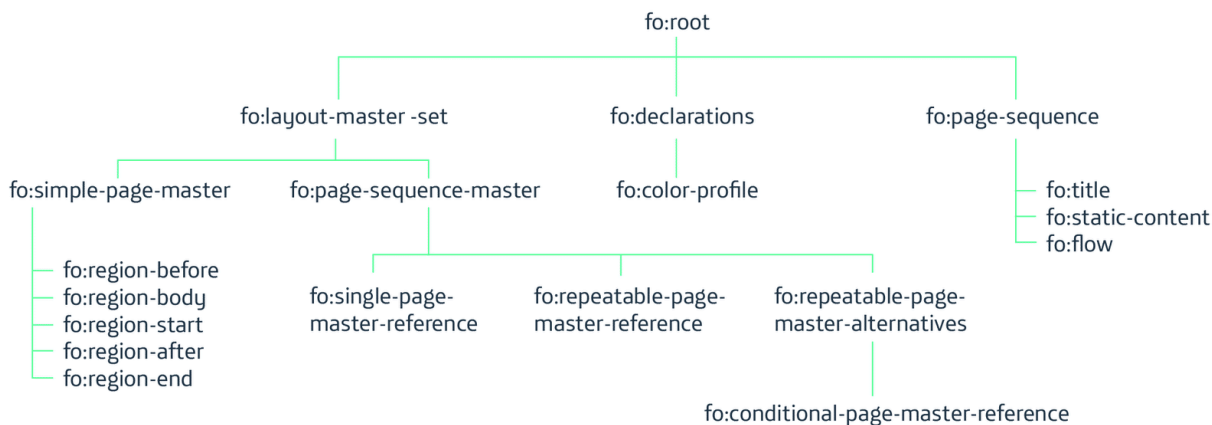
Uno o más elementos **<fo: page-sequence>** describen el contenido de la página. El atributo *master-reference* se refiere a la plantilla de página simple con el mismo nombre.

```
<fo:page-sequence master-reference=" plantilla4 ">
  <!-- La página de contenidos va a este nivel-->
</fo:page-sequence>
```



## Jerarquía de elementos XSL-FO

Veamos la jerarquía de los elementos en un documento XSL-FO.



### fo:root

El elemento `<fo:root>` es el elemento raíz y también declara el espacio de nombres para XSL-FO.

### <fo:layout-master-set>

Define la geometría y secuencia de las páginas. Sus descendientes son:

- **<fo:page-masters>**. Describen las subdivisiones de la página y su geometría.
- **<fo:page-sequence-masters>**. Describen la secuencia en la que un conjunto de páginas maestras serán usadas para generar las páginas tras aplicar el formato. La página maestra de la secuencia interna puede especificarse mediante una referencia a una *single-page-master* o como una repetición de una o más *page-masters*:
  - **<fo:single-page-master-reference>**. Secuencia que consiste en una *single-page-master*.
  - **<fo:repeatable-page-master-reference>**. Especifica la repetición de una *single-page-master*.
  - **<fo:repeatable-page-master-alternatives>**. Especifica la repetición de un conjunto de *page-masters*.

### <fo:page-sequences>

Para cada una solo es utilizada una secuencia de paginas, ya definidas en el `<fo:page-sequence-masters>` que satisfaga las condiciones dadas en el *page-masters*.

Sus descendientes son:

- **<fo:flows>** y **<fo:static-content>** que proporcionan el contenido que se pondrá en las páginas.

### fo:simple-page-master

Cada elemento `<fo:simple-page-master>` contiene una plantilla de página única. Cada plantilla debe tener un nombre único (*master-name*).

**<fo: page-sequence>**

Uno o más elementos *<fo: page-sequence>* describen el contenido de la página. El atributo *master-reference* se refiere a la plantilla de página simple con el mismo nombre.

**<fo: region-body>**

Establece cómo será la apariencia de la región en la que se encuentra el contenido actual de la página.

## Páginas

Las páginas se representan con *<fo:simple-page-master>*, las cuáles son hijas de un *<fo:layout-master-set>*.

Para definir el aspecto de cada una de las páginas tenemos que indicar los tamaños de las regiones anterior, posterior, cuerpo, inicio y fin. Las etiquetas que se usan respectivamente son:

- *<fo:region-before>* - Región anterior.
- *<fo:region-after>* - Región posterior.
- *<fo:region-body>* - Cuerpo.
- *<fo:region-start>* - Inicio.
- *<fo:region-end>* - Fin.

```
<fo:simple-page-master master-name="HelloWorld">
  <fo:region-before extent="20mm"/>
  <fo:region-after extent="20mm"/>
  <fo:region-body margin-top="10mm" margin-bottom="10mm"/>
</fo:simple-page-master>
```

Ejemplo de regiones de impresión que definen áreas en la página.

Estas, a su vez, pueden ser rellenas de contenido usando *<fo:flow>* o *<fo:static-content>*.

La página es la unidad de impresión y tiene las siguientes áreas:

- Ancho de la página: *page-width*.
- Alto de la página: *page-height*.
- Márgenes: opcional, que se subdividen en cuatro regiones:
  - Arriba: *margin-top*.
  - Abajo: *margin-bottom*.
  - Izquierda: *margin-left*.
  - Derecha: *margin-right*.

```
<fo:simple-page-master master-name="HelloWorld"
  page-height="297mm"
  page-width="210mm"
  margin-left="12mm"
  margin-right="12mm"
  margin-top="20mm"
  margin-bottom="20mm">
```

Ejemplo.

## Secuencias de páginas

La secuencia de páginas es el segundo hijo de `<fo:layout-master-set>` y se denomina con la marca `<fo:page-sequence-master>`.

El elemento `<fo:page-sequence-master>` solo contiene un atributo, que determina el nombre de referencia (*master-name*). Las secuencias de página se definen por los elementos incluidos:

- `<fo:repeatable-page-master-reference>`.
- `<fo:repetible-page-master-alternatives>`.
- `<fo:page-master-reference>`.

```
<xsl:template match="/">
  <fo:root>
    <fo:layout-master-set>
      <fo:simple-page-master master-name="PageMaster.Content-right"
      ....
      </fo:simple-page-master>
      <fo:page-sequence-master master-name="Part-Pages">

      </fo:page-sequence-master>
      <fo:page-sequence-master master-name="Content-Pages">
        .....
      </fo:page-sequence-master>
    ...
  ...
</fo:root>
</xsl:template>
```

### Secuencia de páginas complejas

Ejemplo de aplicación de secuencia de páginas.

## Tipos de contenido

Existen dos tipos de contenido que podemos insertar; **dinámico y estático**.

### Contenido dinámico (*flows*)

`<fo:flow>` inserta contenido dinámico en el documento. Puede contener otros elementos como `<fo:block>`, `<fo:display-graphic>`, `<fo:display-link>`, `<fo:display-rule>` y otros elementos colocados al mismo nivel del elemento.

`<fo:flow>` contiene el atributo llamado *flow-name*, cuyos valores pueden ser: *xsl-body*, *xsl-after*, *xsl-before*, *xsl-start* y *xsl-end*, e indican en qué lugar de la página se coloca el contenido.

```
<fo:flow flow-name="region-after">
  <fo:block>Creado por Pepe Illo</fo:block>
</fo:flow>
```

Ejemplo `fo:flow`.

## Contenido estático (*static content*)

**<fo:static-content>** incluye contenido estático en las páginas e internamente tiene los mismos contenidos que **<fo:flow>**. Cuando se usan los contenidos estáticos, **deben aparecer antes de los elementos dinámicos** al definir la secuencia de elementos de la página.

```
<fo:static-content flow-name="region-after">
  <fo:block>(C) Telefónica Educación Digital 2018</fo:block>
</fo:static-content>
```

Ejemplo *fo:static-content*.

## Cajas

XSL-FO utiliza **cajas rectangulares (áreas)** para mostrar la salida.

Estos elementos están divididos formalmente en **cuatro tipos de áreas rectangulares**: contenedores, bloques, líneas e "inclusiones" (*inline-areas*).

## Contenedor

Es la estructura de más alto nivel. Usando coordenadas se colocará en el área que lo contiene. Los contenedores pueden anidarse como secuencias de bloques y de espacios a visualizar.

Por ejemplo, una página de un texto o documento funcionará como un contenedor que incluye otros cinco contenedores: la cabecera, el cuerpo, el pie y los márgenes derecho e izquierdo.

Los FO que producen contenedores son: ***fo:region-body***, ***fo:region-before***, ***fo:region-after***, ***fo:region-start*** y ***fo:region-end***.

## Bloque

Sirve para crear una estructura bloque, que se podría asemejar a un párrafo o un apartado de una lista. Los bloques pueden ser incluidos de forma secuencial dentro del contenedor que los incluye. Los elementos que pueden incluirse dentro de un bloque son: líneas, espacios a visualizar, una imagen y otros bloques (incluir ruptura de línea).

Los FO que producen bloques son: ***fo:block***, ***fo:display-graphic***, ***fo:display-link***, ***fo:display-rule*** y ***fo:list-block***.

```
<fo:block>
  <fo:inline font-style="italic">
    Esto está en letra italica
  </fo:inline>
  Página: <fo:page-number/>
  <fo:inline>
    ¡Esto no lo está!
  </fo:inline>
</fo:block>
```

Los **<fo:block>** pueden estar contenidos en **<fo:flow>**, **<fo:static-content>** u otros **<fo:block>**. Ellos pueden contener a otros **<fo:block>**, así como **<fo:display-graphic>** y **<fo:display-rule>** (ambos a nivel de bloque), y también **<fo:inline>** y **<fo:page-number>** (a nivel de inclusiones). También pueden tener texto plano.

## Línea

Son las líneas de texto dentro de un bloque. Pueden contener inclusiones, así como inclusiones de espacios (*inline areas* e *inline spaces*, respectivamente). No existen FO que se correspondan con las líneas, sino que el motor de evaluación del formateo va generándolas conforme divide líneas debido, por ejemplo, al tipo de justificación elegida.

## Inclusiones

Son partes de una línea, como por ejemplo cada carácter, la referencia a una nota al pie o una expresión matemática. Cada inclusión puede contener otras inclusiones o inclusiones de espacio.

Es un área rectangular que puede contener texto u otras inclusiones. Se corresponde con inclusiones:

- *fo:bidirectional-override*.
- *fo:character*.
- *fo:first-line-marker*.
- *fo:inline-graphic*.
- *fo:inline-include-container*.
- *fo:inline-rule*.
- *fo:inline-sequence*.
- *fo:list-item-body*.
- *fo:list-item-label*.
- *fo:page-number*.
- *fo:page-number-citation*.

Algunas se corresponden con la versión de sus compañeras a nivel de bloque, tales como *fo:inline-rule* o *fo:inline-graphic*. Otras ya las hemos utilizado previamente, como *fo:list-item-body*, *fo:list-item-label* y *fo:page-number*.

## Ejemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- =====
      Hola.fo
      Joaquín Bravo Montero
      (c) Programación en Castellano
      http://www.programacion.net
      ===== -->

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

  <fo:layout-master-set>

    <fo:simple-page-master master-name="simple"
      page-height="29.7cm"
      page-width="21cm"
      margin-top="1cm"
      margin-bottom="2cm"
      margin-left="2.5cm"
      margin-right="2.5cm">
      <fo:region-body margin-top="3cm"/>
      <fo:region-before extent="3cm"/>
      <fo:region-after extent="1.5cm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-name="simple">

    <fo:flow flow-name="xsl-region-body">

      <fo:block font-size="18pt"
        font-family="sans-serif"
        line-height="24pt"
        space-after.optimum="15pt"
        text-align="center"
        padding-top="3pt">
        Mi primer XSL-FO
      </fo:block>

      <fo:block font-size="12pt"
        font-family="sans-serif"
        line-height="15pt"
        space-after.optimum="3pt"
        text-align="justify">
        Hola este es mi primer XSL-FO.
      </fo:block>

    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Si quieres ver más información de este ejemplo, puede ir a la fuente en: [Programación.net](http://www.programacion.net).

## Imágenes

El contenido que normalmente vamos a incluir es de tipo texto, aunque también se pueden incluir imágenes, casi de la misma forma que en HTML se hace con **<img>**.

En este ejemplo vemos cómo insertar un gráfico en un bloque. Las propiedades que incluye son: *href* (ubicación del fichero gráfico), *min-height*, *min-width*, *max-height*, *max-width*, *scale-to-fit*.

```
<fo:block>
  <fo:inline>
    <fo:external-graphic src="xml/Img/imagen1.bmp"/>
  </fo:inline>
  <fo:inline>
    <fo:external-graphic src="xml/Img/imagen1.bmp" content-width="26mm"/>
  </fo:inline>
  <fo:inline>
    <fo:external-graphic src="xml/Img/imagen1.bmp" content-width="25mm" content-
height="20mm"/>
  </fo:inline>
</fo:block>
```

## Listas

Un **<fo:list-block>** es el bloque que genera listas, pueden contener bien una serie de **<fo:list-item>**, o bien pares de **<fo:list-item-label>** y **<fo:list-item-body>**, pero no ambas cosas a la vez.

Tienen tres propiedades de especial relevancia:

- **provisional-label-separation:** se le dan tres valores de distancia separados por punto y coma, indicando valor máximo, mínimo y óptimo, respectivamente.
- **provisional-distance-between-starts:** distancia desde el comienzo del eje de la etiqueta y el comienzo del eje del cuerpo del elemento.
- **space-between-list-rows:** distancia entre los sucesivos elementos: máximo, mínimo y óptimo.

```
<fo:list-block>
  <fo:list-item>
    <fo:list-item-label>*</fo:list-item-label>
    <fo:list-item-body>Primer elemento de la lista</fo:list-item-label>
  </fo:list-item>
  <fo:list-item>
    <fo:list-item-label>*</fo:list-item-label>
    <fo:list-item-body>Segundo elemento de la lista</fo:list-item-label>
  </fo:list-item>
</fo:list-block>
```

Ejemplo de lista.



```
<fo:list-block>
  <fo:list-item-label>*</fo:list-item-label>
  <fo:list-item-body>Primer elemento de la lista</fo:list-item-label>
  <fo:list-item-label>*</fo:list-item-label>
  <fo:list-item-body>Segundo elemento de la lista</fo:list-item-label>
</fo:list-block>
```

Ejemplo de lista.

## Enlaces

Un enlace tiene sentido en la versión electrónica de dicho documento. Para introducir enlaces se usa la etiqueta `<fo:simple-link>`, que puede actuar como un elemento de bloque o de inclusión, según su contenido.

Para controlar su comportamiento se le han asignado seis atributos: ***external-destination***, ***internal-destination***, ***indicate-destination***, ***show-destination***, ***space-above-destination-block*** y ***space-above-destination-start***.

- Un enlace a un **documento remoto** se especifica asignándole su *URI* a *external-destination*.
- Un enlace a un **recurso del propio documento** se indica asignándole el valor del identificador de dicho recurso a la propiedad *internal-destination*.
- Un enlace **no puede tener estos dos atributos a la vez**.

***<fo:show-destination>*** acepta los valores *replace* (por defecto) o *new*, que permite cargar el enlace indicado en la misma ventana o en otra distinta.

Por último, ***<fo:space-above-destination-start>*** y ***<fo:space-above-destination-block>*** permiten indicar si cuando se cargue el documento destino se va a poner el punto exacto indicado por el enlace en la parte superior del navegador, o por el contrario se va a desplazar hacia abajo (no dejando espacio vacío, sino con el contenido que lo preceda).

# Despedida

## Documentación y ejemplos

La elaboración de documentos con XSL:FO, como has podido comprobar, tiene una gran cantidad de opciones y puede ser una herramienta muy poderosa para incorporar en nuestras aplicaciones.

A continuación encontrarás enlaces con ejemplos y documentación para que profundices en el tema.

### [Ejemplos XSL-FO](#)

Página con un listado de ejemplos, para ver el código y el resultado.

### [Más ejemplos de aplicación con XSL-FO](#)

Página de archivo con ejemplos de uso de XSL-FO.

### [Manual XSL-FO](#)

Manual en inglés de XSL-FO.

## Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

XML es un estándar muy abierto que permite almacenar información de cualquier tipo y con una estructura en árbol. En muchas tecnologías el uso de documentos XML, tanto para la salida como para la entrada de datos, es de uso común y está estandarizado.

Dentro de XSL se encuentra incluido el estándar para poder generar documentos electrónicos imprimibles o electrónicos con formato legible. Está aceptado por la W3C y es el XSL-FO. En este tema hemos visto su sintaxis y estructura.