

# 1. XPath

---

## 1.1 Introducción

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas:

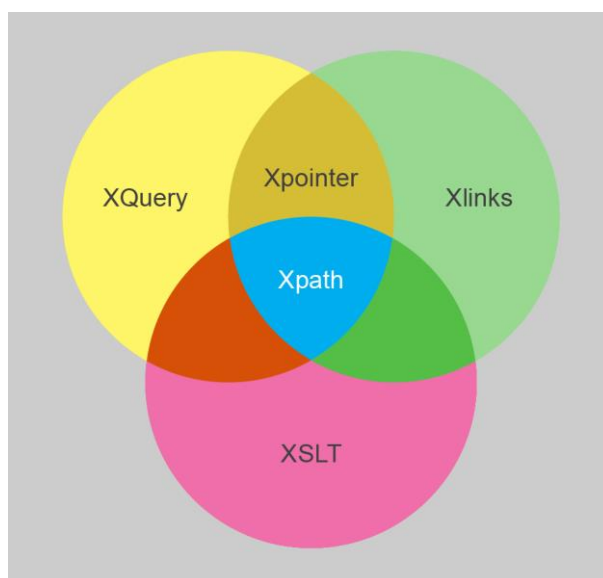
- Linguaxes para o procesamento de documentos XML.
- Estrutura en árbore dun documento XML. O modelo DOM XML.
- Sintaxe das expresións XPath. Rutas e pasos de localización.
- Operadores e funcións da linguaxe XPath.
- Creación de expresións con XPath.

## 1.2 Actividade

### Linguaxes para o procesamento de documentos XML

Nas actividades anteriores traballamos con documentos XML, e vimos a súa sintaxe para poder crear documentos ben formados. Tamén estudamos dúas linguaxes (*DTDs* e *XML Schemas*) que nos permiten definir gramáticas específicas que podemos empregar para validar o contido dos documentos XML.

Ao redor da linguaxe XML definíronse un conxunto de tecnoloxías que se empregan para procesar e obter información dos documentos XML. Algunhas destas tecnoloxías son:

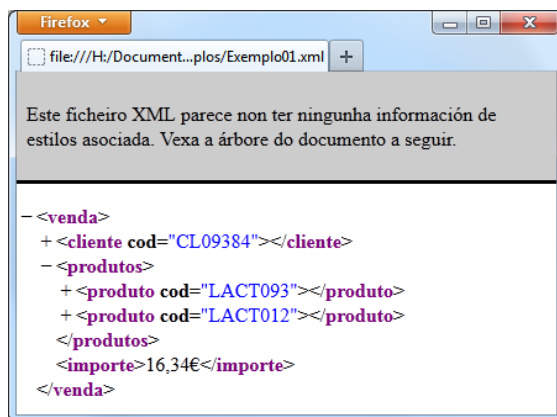


- **XPath**. É unha linguaxe que permite extraer información dun documento XML.
- **XLink**. É unha linguaxe para a creación de hipervínculos nun documento XML.
- **XPointer**. É unha linguaxe que permite que os hipervínculos enlacen con partes específicas dun documento XML.

- **XQuery**. É unha linguaxe que permite realizar consultas a documentos XML, do mesmo xeito que SQL o fai coas bases de datos relacionais.
- **XSLT**. É unha linguaxe que permite transformar documentos XML, obtendo novos documentos XML con diferente estrutura ou documentos noutros formatos.

### Procesamento dun documento XML: **estrutura en árbore**

Moitas aplicacións integran un *parser* XML. Un *parser* é un programa capaz de procesar un documento XML, e xeralmente o resultado obtido é a representación deste documento en forma de árbore. Por exemplo, os navegadores web integran un *parser* XML, e amosan a árbore obtida como resultado do procesamento:



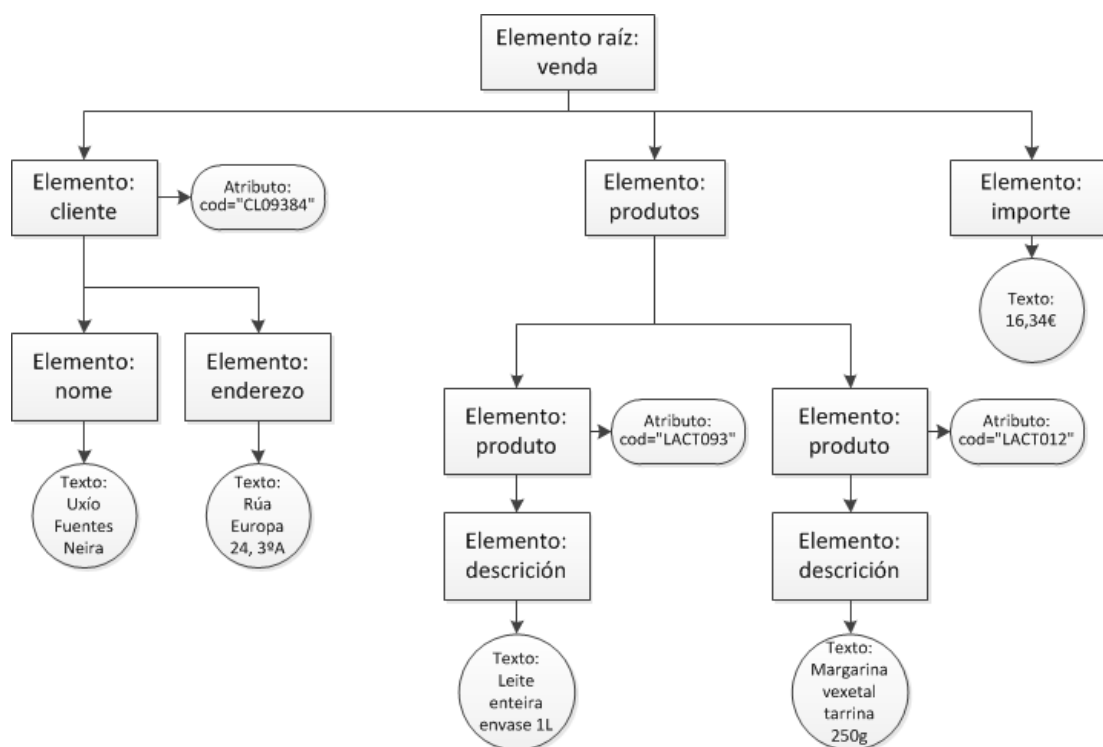
O **modelo máis empregado para almacenar e procesar as árbores XML é o DOM** (*Document Object Model*, Modelo de Obxectos do Documento). O DOM é un estándar do W3C (*World Wide Web Consortium*) que tamén se emprega no procesamento dos documentos HTML que se empregan nas páxinas web. O estándar do DOM divídese en tres partes:

- DOM base. É un modelo estándar para calquera documento estruturado.
- DOM HTML. É o modelo específico para documentos HTML.
- **DOM XML**. É o modelo específico para documentos XML.

Por exemplo, o seguinte **documento XML**:

```
<?xml version="1.0" encoding="utf-8"?>
<venda>
  <cliente cod="CL09384">
    <nome>Uxío Fuentes Neira</nome>
    <endereço>Rúa Europa 24, 3ºA</endereço>
  </cliente>
  <produtos>
    <produto cod="LACT093">
      <descrición>Leite enteira envase 1L</descrición>
    </produto>
    <produto cod="LACT012">
      <descrición>Margarina vexetal tarrina 250g</descrición>
    </produto>
  </produtos>
  <importe>16,34€</importe>
</venda>
```

Preséntase do seguinte xeito empregando o **modelo DOM XML**:



Nunha **árbore DOM XML**, o **documento XML** represéntase empregando **nodos**. Tódolos compoñentes dun documento XML son nodos. Para crear a árbore **debe terse en conta a orde na que aparecen os elementos dentro do documento XML**; por exemplo, o produto de código "LACT093" é anterior ao de código "LACT012". Non é relevante a orde dos atributos dentro dun elemento.

O estándar DOM XML define tamén unha interface de programación orientada a obxectos, cos obxectos correspondentes aos distintos nodos dun documento XML, e as propiedades e os métodos para acceder a eles (obter o seu contido, modificalos, eliminalos ou engadir novos nodos).

### **Tipos de nodos** no modelo DOM XML

No modelo DOM XML, unha árbore correspondente a un documento XML está composta por nodos. Estes nodos poden ser de distintos tipos. Os principais son:

- Nodo *Document* (**Documento**). Representa ao documento XML enteiro. Ten un único fillo de tipo elemento (o nodo raíz).
- Nodo *Element* (**Elemento**). Representa un elemento dun documento XML. Poden ter identificadores únicos (pódese especificar no documento de validación) para acceder a eles de xeito directo.
- Nodo *Attr* (**Atributo**). Representa un atributo dun elemento. Aínda que se lles denomina nodos, na estrutura de árbore considérase aos atributos como unha información engadida aos nodos "Elemento" e non como fillos deles.
- Nodo *Text* (**Texto**). Representa ao texto dun elemento. Contén todos os caracteres que non están dentro dalgunha etiqueta.
- Nodo *Comment* (**Comentario**).
- Nodo *namespace* (**espazo de nomes**).

- Nodo *ProcessingInstruction* (Instrución de Procesamento).

As relacións entre os nodos dunha árbore DOM XML son as seguintes:

- Soamente existe un *nodo raíz*, de tipo "Elemento".
- Un nodo "Elemento" pode ter ou non *nodos fillos*. Un nodo "Elemento" sen fillos é un *nodo folia*. Os nodos dos outros tipos nunca teñen fillos.
- Tódolos nodos a excepción do raíz teñen un e soamente un *nodo pai*. Tampouco teñen pai os nodos de tipo "Atributo", que como xa dixemos cólganse na árbore do nodo "Elemento" que os contén, pero non se consideran fillos deste.
- Chámanse *nodos irmáns* a aqueles nodos "Elemento" que teñen o mesmo pai.

### A interface de programación DOM XML

Como xa dixemos, DOM XML define tamén unha interface de programación para manexar os nodos da árbore correspondente a un documento XML. Esta interface de programación inclúe propiedades como:

- *nodeName*, para obter o nome dun nodo.
- *parentNode*, para obter o pai dun nodo.
- *attributes*, para obter os atributos dun nodo.

E tamén métodos como:

- *getElementsByTagName(nome)*, obtén os elementos cunha etiqueta determinada.
- *hasAttributes()*, indica se un elemento ten ou non atributos.
- *removeChild(nodo)*, elimina a un fillo dado dun nodo.

Así por **exemplo**, se quixéramos eliminar do documento anterior o nodo correspondente ao enderezo do cliente poderíamos facer:

```
// supoñemos o documento XML cargado en docXML
endereço=docXML.getElementsByTagName("endereço")[0];
cliente=endereço.parentNode;
cliente.removeChild(endereço);
```

As expresións que empregan DOM para percorrer e localizar nodos nunha árbore XML son en moitos casos complexas e moi largas. Como veremos a continuación, **a linguaxe XPath ofrece unha forma moito máis sinxela e efectiva de facer esta tarefa.**

### Expresións XPath

XPath (*XML Path*) é unha **linguaxe para acceder ás distintas partes dun documento XML**. Empregando XPath podemos seleccionar e facer referencia a texto, elementos, atributos e calquera outra información contida dentro dun documento XML. Non é unha linguaxe XML; **ten a súa propia sintaxe**.

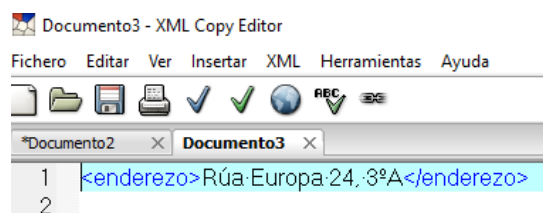
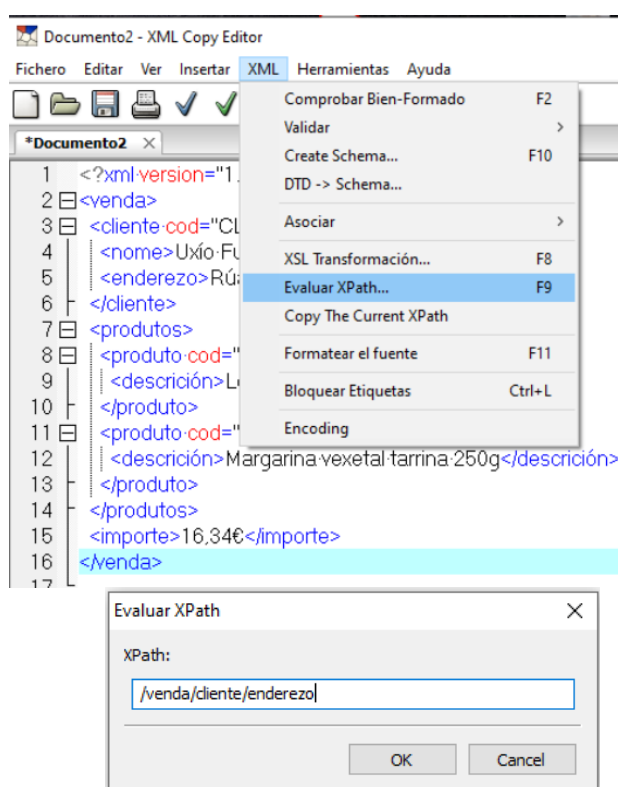
Por **exemplo**, no documento XML anterior poderíamos empregar a seguinte expresión para obter o enderezo do cliente:

/venda/cliente/enderezo

A versión actual Xpath 3.1 está especificada na [recomendación de W3C do 21 de marzo de 2017](#).

Na presente actividade, para avaliar as expresións XPath podemos utilizar *XML Copy Editor*, *XPathBuilder* ou probadores online como [Probador/evaluador XPath1](#) e [Probador/evaluador XPath2](#).

Cargando o arquivo anterior no *XML Copy Editor*, preme "XML" para abrir os menús, seleccione Evaluar XPath... e poderá avaliar a expresión anterior obtendo co seguinte resultado:



## XPathBuilder:

XPathBuilder

/def:centro/def:ciclo/al:alumno[al:nome='Xoan']/al:módulo/@id

Evaluate ☒ Evaluate when typing ☐ Evaluate on button click

MP0370  
MP0371

Tarefa4\_3\_a.xml

```
<?xml version="1.0" encoding="utf-8"?>
<centro xmlns="http://www.centro.es/ciclos" xmlns:al="http://www.alumnos.es/alumnos">
  <ciclo id="ASIR" grao="CS">
    <módulo id="MP0370">
      <nome>Planificación e administración de redes</nome>
      <horas>213</horas>
    </módulo>
    <módulo id="MP0371">
      <nome>Fundamentos hardware</nome>
      <horas>107</horas>
    </módulo>
    <al:alumno>
      <al:nome>Xoan</al:nome>
      <al:apelidos>Cancedo da Foz</al:apelidos>
      <al:datanacemento>06/11/1991</al:datanacemento>
      <al:nota>Xa fixo outro ciclo</al:nota>
      <al:módulo id="MP0370" nota="6" />
      <al:módulo id="MP0371" nota="8" />
    </al:alumno>
    <al:alumno>
      <al:nome>Carla</al:nome>
      <al:apelidos>Castro Castro</al:apelidos>
      <al:datanacemento>06/10/1993</al:datanacemento>
      <al:nota>Ten un grado</al:nota>
      <al:módulo id="MP0370" nota="9" />
      <al:módulo id="MP0371" nota="7" />
    </al:alumno>
  </ciclo>
  <ciclo id="DAW" grao="CS">
    ...
  </ciclo>
  ...
</centro>
```

E:\LM\UD6 XML\Actividades 1 2\Tarefa4 3 a.xml

## Probador / Evaluador XPath

Le permite probar sus expresiones / consultas XPath en un archivo XML. Esta herramienta funciona mejor que otras herramientas en línea de XPath existentes, ya que admite la mayoría de las funciones de XPath (cadena (), número (), nombre (), longitud de cadena () etc.) y no lo limita a trabajar contra nodos. Es totalmente compatible con la especificación XPath 2.0 / 3.0. Vea la sección de [Ejemplos XPath](#) para más detalles.

El probador XPath es totalmente compatible con los espacios de nombres XML. Vea la sección de Ejemplos XPath para más detalles. El prefijo de espacio de nombres "fn" y "matemáticas" están reservados para las funciones XPath.

\* El límite de tamaño máximo para cargar archivos es de 2 megabytes. Los resultados superiores a 500k se escribirán en una nueva ventana por razones de rendimiento y para evitar que su navegador no responda.

### Entrada XML

Opción 1: copie y pegue su documento XML aquí

```
<?xml version="1.0" encoding="utf-8"?>
<venda>
  <cliente cod="CL09384">
    <nome>Uxío Fuentes Neira</nome>
    <endereço>Rúa Europa 24, 3ºA</endereço>
  </cliente>
```

Opción 2: O suba su documento XML

Seleccionar archivo

Ningún archivo seleccionado

UTF-8

### Expresión XPath

/venda/cliente/endereço

Incluya el tipo de elemento XML en la salida:

No

PRUEBA XPATH

PRUEBA XPATH EN NUEVA VENTANA

### Resultado XPath:

```
<endereço> Rúa Europa 24, 3ºA </endereço>
```

# Template / XPath 3.0 / XQuery 3.0 / CSS 3 Selector / JSONiq Online Tester

(Puede encontrar la documentación a continuación)

Archivo de entrada HTML / XML: xml

```
1 <? xml version = "1.0" encoding = "utf-8"?>
2 <venda>
3   <cliente cod = "CL09384" >
4     <nome> Uxío Fuentes Neira </nome>
5     <endereço> Rúa Europa 24, 3ªA </endereço>
6   </cliente>
7   <produtos>
8     <produto cod = "LACT093" >
9       <descricao> Leite inteira envase 1L </ descricao>
10    </produto>
11    <produto cod = "LACT012" >
12      <descricao> Margarina vexetal tarrina 250g </ descricao>
13    </produto>
14  </produtos>
15  <importe> 16,34 € </importe>
16</venda>
17
```

☒ Modelo ☒ XPath 3.0 ☒ 2.0 ☒ XQuery 3.0 ☒ 1 ☒ Selectores CSS 3.0 ☒ Detección automática

1 / venda / cliente / endereço  
2  
3

☐ desactivar la actualización automática ☐ desactivar el resaltado de sintaxis

Opciones de salida : formato de nodo :: texto  Formato de salida:: ad hoc  ☐ Mostrar tipos ☐ Ocultar nombres de variables

Compatibilidad : Habilitar todas las extensiones

Trabajo en progreso: ☒ XPath 3.1 ☐ XQuery 3.1

Activar Windows  
Ve a Configuración para activar Windows.

Resultado de la expresión anterior aplicada al archivo HTML anterior:

```
1 Rúa Europa 24, 3ªA
2
```

En XPath, as **expresións** poden empregar e devolver os seguintes tipos de datos:

- **Números en punto flotante**.
- Valores **booleanos** (verdadeiro ou falso).
- **Cadeas de caracteres** codificadas en Unicode.
- **Conxuntos de nodos**, que poden conter cero, un ou máis nodos.

## Rutas de localización

O tipo máis común de expresión en XPath é a ruta de localización. Unha ruta de localización **permite a selección dun conxunto de nodos, partindo dun nodo contexto no que se avaliará a expresión**. O resultado de avaliar unha ruta de localización é sempre un conxunto de nodos (de distintos tipos, non soamente nodos "Elemento").

As rutas de localización **analízanse de esquerda a dereita**, e **poden ser absolutas ou relativas**:

- As **rutas de localización relativas** poden comezar en calquera outro nodo da estrutura da árbore, onde o punto de partida recibe o nome de nodo de contexto. Por exemplo, poderíamos avaliar unha expresión tendo como contexto:



```
<produto cod="LACT012">
  <descricao>Margarina vexetal tarrina 250g</descricao>
</produto>
```

De xeito que a expresión "produto/descrición" obteña o nodo "descricao" do produto.

- Unha **ruta absoluta** comeza no nodo de documento e as expresións comezarán cunha barra diagonal "/" (ruta absoluta), o que indica que o contexto da expresión é o nodo "Documento" (o documento XML enteiro).

As expresións XPath tamén **empregan a barra "/" para separar as diferentes partes das que se compoñen**. Estas partes chámanse **pasos de localización**. Cada un dos pasos de localización vai refinando a procura de datos a través dos nodos da árbore.

Por exemplo, a ruta de localización "/venda/cliente/enderezo" componse de tres pasos de localización. O primeiro fai referencia ao nodo raíz "venda", que colga do nodo documento "/"; o seguinte fai referencia ao nodo "cliente" que é fillo de "venda"; e o mesmo para o nodo "enderezo".



**Tarefa 1. Rutas de localización sinxelas.** Nesta tarefa, sobre un documento XML dado, empregaremos rutas de localización sinxelas para obter conxuntos de nodos.

Tomando como base o seguinte documento XML:

```
<?xml version="1.0" encoding="utf-8"?>
<equipos>
  <máquina nome="PC017">
    <hardware>
      <tipo>PC Sobremesa</tipo>
      <fabricante>Dell</fabricante>
      <procesador marca="Intel" num_nucleos="4" velocidde="3,1">i7</procesador>
      <memoria tecnoloxía="DDR3">8</memoria>
      <disco tecnoloxía="SATA" capacidade="2000"/>
      <gravadora tipo="DVD"/>
    </hardware>
    <config>
      <OS>Windows 7</OS>
      <IP>192.168.20.105</IP>
      <gateway>192.168.20.1</gateway>
    </config>
  </máquina>
  <máquina nome="GALILEO">
    <hardware>
      <tipo>Torre</tipo>
      <fabricante>Fujitsu-Siemens</fabricante>
      <procesador marca="Intel" num_nucleos="4" velocidde="3">Xeon</procesador>
      <memoria tecnoloxía="DDR2">2</memoria>
      <disco tecnoloxía="SCSI" capacidade="200"/>
      <disco tecnoloxía="SCSI" capacidade="200"/>
      <disco tecnoloxía="SCSI" capacidade="200"/>
      <lectora tipo="DVD"/>
    </hardware>
```

```

<config>
  <role>Servidor de dominio</role>
  <OS>Windows 2008 Server R2</OS>
  <IP>192.168.20.10</IP>
  <gateway>192.168.20.1</gateway>
</config>
</máquina>
</equipos>

```

Obtén as rutas de localización para obter os seguintes conxuntos de nodos:

- Esto no es el número que máquinas, sino el nombre... /equipos/máquina/@nome
- As máquinas que contén o documento.
  - A información correspondente ao hardware das máquinas. /equipos/máquina/hardware o // hardware
  - A información de configuración das máquinas. /equipos/máquina/config
  - A lista de fabricantes dos equipos. /equipos/máquina/hardware/fabricante
  - A lista de discos que conteñen. /equipos/máquina/hardware/disco
  - As direccións IP. /equipos/máquina/config/IP

### Pasos de localización

Os pasos de localización (*location steps*) son cada un dos elementos dunha ruta de localización. Co resultado obtido de avaliar un paso de localización, obtense o contexto do seguinte paso de localización. Constan dun **eixo** (*axis*), un **test de nodo** (*node test*), que é un filtro có que se delimita a un conxunto de nodos entre todos os que se sitúan no eixo, e opcionalmente dun **predicado**. A sintaxe é a seguinte:

```
eixo::test-nodo[predicado1] [predicado2]...
```

### Eixos

Os eixos indican, con respecto ao nodo contexto, o *conxunto de nodos sobre os cales se avaliará o test de nodo e o predicado se existe*. Basicamente trátase de realizar un primeiro filtro de nodos da árbore para obter o resultado cos datos buscados.

Os eixos que podemos empregar son:

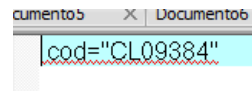
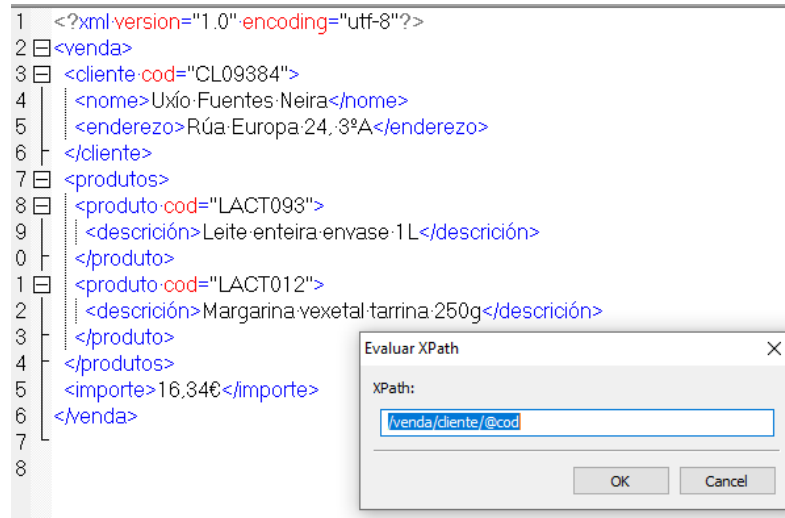
- **self**. O propio nodo contexto. Se corresponde co punto “.”.
- **child**. Os fillos do nodo contexto. Súa forma habitual é a barra, “/”, aínda que tamén pode poñerse: /child::.
- **parent**. O pai do nodo contexto. Para referirnos a el usamos os dous puntos, “..”.
- **ancestor**. Os antepasados do nodo contexto.
- **ancestor-or-self**. O nodo contexto e os seus antepasados.
- **descendant**. Os descendentes do nodo contexto.
- **descendant-or-self**. O nodo contexto e os seus descendentes.
- **following**. Os nodos seguintes ao nodo contexto, sen descendentes.

- **following-sibling.** Os nodos do mesmo nivel que seguen ao nodo contexto.
- **preceding.** Os nodos anteriores ao nodo contexto, sen antepasados.
- **preceding-sibling.** Os nodos do mesmo nivel que preceden ao nodo contexto.
- **attribute.** Os nodos atributo do nodo contexto.
- **namespace.** Os nodos de espazo de nomes do nodo contexto.

➤ Se non se pon ningún, o eixo por defecto é "child".

➤ Tamén se pode empregar o símbolo "@" no lugar do eixo "attribute". Por **exemplo**, as seguintes **expresións son equivalentes**:

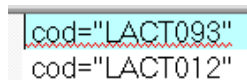
`/venda/cliente/@cod`      ← *Mais fácil de ver*  
`/child::venda/child::cliente/attribute::cod`



Veamos outros **exemplos**.

- Obter os códigos dos produtos que teñan unha descrición.

`/venda/produtos/produto[descricao]/@cod`



- Obter os códigos dos produtos que teñan unha descrición que empece por "Leite".

`/venda/produtos/produto[starts-with(descricao/text(),"Leite")]/@cod`





**Tarefa 2. Rutas de localización con diversos eixos.** Nesta tarefa, sobre o mesmo documento XML da tarefa anterior, crearemos expresións XPath.

- Os nomes das máquinas.
- Os nomes das máquinas que teñan gravadora (fíxate que tes que ir ata a gravadora e logo volver para obter o nome da máquina).
- O sistema operativo das máquinas nas que figure o "role".

Existen varias solucións posibles para cada unha das expresións que se solicitan.

### Tests de nodo

O test de nodo serve para, unha vez identificado un conxunto de nodos co eixo adecuado, especificar exactamente que nodos dese conxunto son os que queremos. Os test de nodo poden ser:

- `nome_dun_nodo`. Selecciona todos os nodos co nome indicado.
- `*`. Selecciona todos os elementos e atributos.
- `node()`. Selecciona todos os nodos (de calquera tipo).
- `text()`. Selecciona os nodos de texto.
- `comment()`. Selecciona os nodos de comentario.
- `processing-instructions()`. Selecciona os nodos de procesamento de instrucións.

Por exemplo, se quixéramos obter o conxunto de todos os atributos do documento, poderíamos facer:

```
//node()/@*      ← Mais fácil de ver  
//descendant-or-self::node()/@*
```

Isto é, seleccionamos todos os nodos do documento, e despois quedámonos cos seus atributos.

Se quixéramos soamente os atributo de nome "cod", poderíamos facer:

```
//node()/@cod      ← Mais fácil de ver  
//descendant-or-self::node()/@cod
```

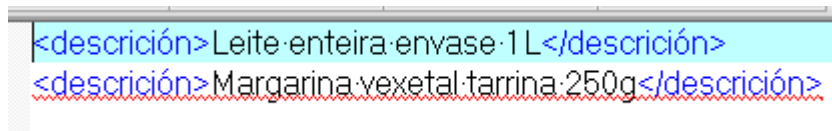
E calquera das seguintes formas é válida para obter os textos das descrições dos produtos:

```
/venda/produtos/produto/descripción/text()      ← Mais fácil de ver  
/descendant-or-self::node()/descripción/text()
```

Obsérvase que non é o mesmo obter como resultado un conxunto de nodos sen filtrar

```
//node()/descricaoión ← Mais fácil de ver
```

```
/descendant-or-self::node()/descricaoión:
```

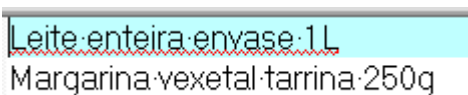


```
<descricaoión>Leite inteira envase 1 L</descricaoión>
<descricaoión>Margarina vexetal tarrina 250g</descricaoión>
```

que indicar que queremos obter soamente os nodos de tipo "Texto"

```
//node()/descricaoión/text() ← Mais fácil de ver
```

```
/descendant-or-self::node()/descricaoión/text()
```



```
Leite inteira envase 1 L
Margarina vexetal tarrina 250g
```

**Debido ao seu frecuente uso, existen abreviaturas para algunhas rutas de localización:**

- `"//"` equivale a `"descendant-or-self::node()"`
- `"."` equivale a `"self::node()"`
- `".."` equivale a `"parent::node()"`
- `"@"` equivale a `"attribute::@"`

Isto é, tamén poderíamos ter escrito a anterior expresión como:

```
//descricaoión/text()
```

### Predicados

Os predicados son condicións opcionais nun paso de localización, e introdúcense entre corchetes despois do test de nodo. Axúdannos a axustar a busca no conxunto de nodos que nos interesan.

Cada predicado pode conter unha ruta de localización relativa ao nodo actual. Unha forma habitual que se emprega nos predicados fai uso dunha característica especial de XPath: calquera conxunto de nodos non baleiro é tratado de forma booleana como "verdadeiro", mentres que a un conxunto de nodos baleiro asígnaselle o valor booleano "falso". Desta forma, na expresión:

```
//produto[child::descricaoión]
```

O predicado filtra os produtos obtendo soamente aqueles que teñen un nodo elemento fillo de nome "descricaoión". O mesmo poderíase facer da seguinte forma abreviada:

```
//produto[descricaoión] ← Mais fácil de ver
```

```

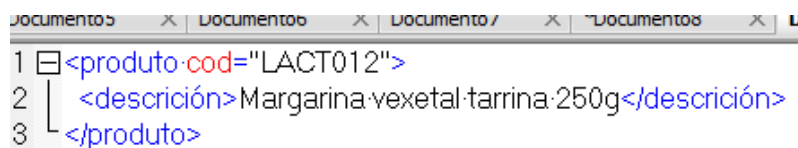
<produto·cod="LACT093">
  <descricao>Leite·enteira·envase·1 L</descricao>
</produto>
<produto·cod="LACT012">
  <descricao>Margarina·vexetal·tarrina·250g</descricao>
</produto>

```

Tamén podemos poñer condicións ao resultado obtido pola ruta de localización indicada no predicado. Por exemplo, a expresión:

```
//produto[@cod="LACT012"]
```

obtería os nodos "produto" cun atributo "cod" con valor "LACT012".



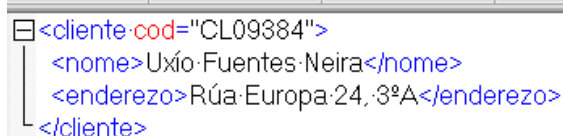
```

1 <produto·cod="LACT012">
2   <descricao>Margarina·vexetal·tarrina·250g</descricao>
3 </produto>

```

E para obter os clientes cun nome concreto, faríamos:

```
//cliente[nome/text()='Uxío Fuentes Neira']
```



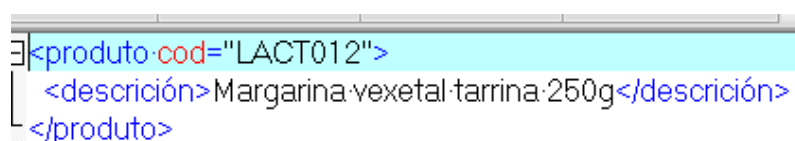
```

<cliente·cod="CL09384">
  <nome>Uxío·Fuentes·Neira</nome>
  <endereço>Rúa·Europa·24, 3ªA</endereço>
</cliente>

```

Un mesmo paso de localización pode conter cero, un ou varios predicados; neste último caso, poranse un a continuación do outro, cadanseu cos seus propios corchetes:

```
//produto[descricao][@cod="LACT012"]
```

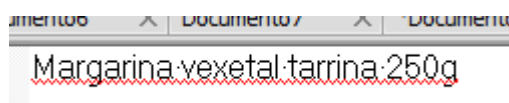


```

<produto·cod="LACT012">
  <descricao>Margarina·vexetal·tarrina·250g</descricao>
</produto>

```

```
//produto[descricao][@cod="LACT012"]/descricao/text()
```



```

Margarina·vexetal·tarrina·250g

```



**Tarefa 3. Rutas de localización con tests de nodo e predicados.** Nesta tarefa, sobre un documento XML dado, empregaremos rutas de localización que inclúan tests de nodo e predicados.

Tomando como base o seguinte documento XML:

```
<?xml version="1.0" encoding="utf-8"?>
<equipos>
  <máquina nome="PC017">
    <hardware>
      <tipo>PC Sobremesa</tipo>
      <fabricante>Dell</fabricante>
      <procesador marca="Intel" num_nucleos="4" velocidde="3.1">i7</procesador>
      <memoria tecnoloxía="DDR3">8</memoria>
      <disco tecnoloxía="SATA" capacidade="2000"/>
      <gravadora tipo="DVD"/>
    </hardware>
    <config>
      <OS>Windows 7</OS>
      <IP>192.168.20.105</IP>
      <gateway>192.168.20.1</gateway>
    </config>
  </máquina>
  <máquina nome="PC053">
    <hardware>
      <tipo>Semitorre</tipo>
      <memoria>0.5</memoria>
      <disco capacidade="40"/>
      <lectora tipo="CD"/>
    </hardware>
    <config>
      <OS>Windows XP</OS>
    </config>
  </máquina>
  <máquina nome="PC007">
    <hardware>
      <tipo>Semitorre</tipo>
      <memoria tecnoloxía="DDR">0.5</memoria>
      <disco capacidade="40"/>
      <lectora tipo="CD"/>
    </hardware>
    <config>
      <OS>Windows XP</OS>
    </config>
    <notas>Sin tarxeta de rede</notas>
  </máquina>
  <máquina nome="PR003">
    <hardware>
      <tipo>Impresora Inyección</tipo>
      <fabricante>Lexmark</fabricante>
    </hardware>
    <config/>
  </máquina>
  <máquina nome="PC011">
    <hardware>
      <tipo>Semitorre</tipo>
      <memoria>1</memoria>
      <disco capacidade="80"/>
    </hardware>
  </máquina>
</equipos>
```

```

    <lectora tipo="CD"/>
</hardware>
<config>
    <OS>Windows 2000 SP4</OS>
    <IP>192.168.10.221</IP>
</config>
</máquina>
<máquina nome="PC019">
    <hardware>
        <tipo>Semitorre</tipo>
        <procesador marca="AMD" velocidade="1.4">Athlon</procesador>
        <memoria>0.5</memoria>
        <disco capacidade="40"/>
        <gravadora tipo="CD"/>
    </hardware>
    <config>
        <OS>Mandriva 2007</OS>
        <IP>192.168.10.45</IP>
        <gateway>192.168.10.1</gateway>
    </config>
</máquina>
<máquina nome="PR007">
    <hardware>
        <tipo>Impresora Láser</tipo>
        <fabricante>OKI</fabricante>
    </hardware>
    <config/>
    <notas>Monocromo, dúplex, red</notas>
</máquina>
<máquina nome="COPERNICO">
    <hardware>
        <tipo>Torre</tipo>
        <fabricante>Fujitsu-Siemens</fabricante>
        <procesador marca="Intel" num_nucleos="4" velocidde="3">Xeon</procesador>
        <memoria tecnologia="DDR">2</memoria>
        <disco tecnologia="SCSI" capacidade="500"/>
        <disco tecnologia="SCSI" capacidade="500"/>
        <gravadora tipo="DVD"/>
    </hardware>
    <config>
        <role>Servidor de dominio</role>
        <OS>Windows 2003 Server R2</OS>
        <IP>192.168.20.11</IP>
        <gateway>192.168.20.1</gateway>
    </config>
</máquina>
<máquina nome="GALILEO">
    <hardware>
        <tipo>Torre</tipo>
        <fabricante>Fujitsu-Siemens</fabricante>
        <procesador marca="Intel" num_nucleos="4" velocidde="3">Xeon</procesador>
        <memoria tecnologia="DDR2">2</memoria>
        <disco tecnologia="SCSI" capacidade="200"/>
        <disco tecnologia="SCSI" capacidade="200"/>
        <disco tecnologia="SCSI" capacidade="200"/>
        <lectora tipo="DVD"/>
    </hardware>
    <config>
        <role>Servidor de dominio</role>
        <OS>Windows 2008 Server R2</OS>
        <IP>192.168.20.10</IP>

```



```

    <gateway>192.168.20.1</gateway>
  </config>
</máquina>
<máquina nome="KEPLER">
  <hardware>
    <tipo>Rack</tipo>
    <fabricante>HP</fabricante>
    <procesador marca="Intel" num_nucleos="2" velocidade="3">Core2
Duo</procesador>
    <memoria tecnoloxía="DDR2">4</memoria>
    <disco tecnoloxía="SATA" capacidade="500"/>
    <disco tecnoloxía="SATA" capacidade="500"/>
    <disco tecnoloxía="SATA" capacidade="500"/>
    <gravadora tipo="DVD"/>
  </hardware>
  <config>
    <role>Servidor de arquivos</role>
    <OS>Ubuntu 8.04 Server</OS>
    <IP>192.168.10.10</IP>
    <gateway>192.168.10.1</gateway>
  </config>
</máquina>
<máquina nome="NEWTON">
  <hardware>
    <tipo>Rack</tipo>
    <fabricante>HP</fabricante>
    <procesador marca="Intel" num_nucleos="2" velocidade="3">Core2
Duo</procesador>
    <memoria tecnoloxía="DDR2">4</memoria>
    <disco tecnoloxía="SATA" capacidade="500"/>
    <disco tecnoloxía="SATA" capacidade="500"/>
    <gravadora tipo="DVD"/>
  </hardware>
  <config>
    <role>Servidor web</role>
    <OS>Ubuntu 8.04 Server</OS>
    <IP>192.168.10.11</IP>
    <gateway>192.168.10.1</gateway>
  </config>
</máquina>
</equipos>

```

Algunhas **expresións XPath** para obter os seguintes conxuntos de nodos, serían.

- O nome do sistema operativo da máquina con IP "192.168.10.45".

```
//máquina/config[IP="192.168.10.45"]/OS
```

- Os textos (soamente) que figuran na configuración (elemento "config") do equipo de nome "COPERNICO".

```
//máquina[@nome="COPERNICO"]/config/*/text()
```

- Os fabricantes das máquinas que teñan 4GB de memoria "DDR2".

```
//máquina[hardware/memoria=4][hardware/memoria/@tecnoloxía="DDR2"]/hardware/fabricante
```

- O sistema operativo das máquinas nas que figure o número de núcleos do procesador.

```
//máquina[hardware/procesador/@num_nucleos]/config/OS
```

- Os nomes das máquinas que empreguen memoria con tecnoloxía "DDR2".

```
//máquina[hardware/memoria/@tecnoloxía="DDR2"]/@nome
```

- As máquinas con procesador da marca "Intel" e gravadora de DVD.

Dúas solucións:

```
//máquina[hardware/procesador/@marca="Intel"][hardware/gravadora/@tipo="DVD"]  
//procesador[@marca="Intel"]/../gravadora[@tipo="DVD"]/ancestor::máquina
```

**Escribe ti agora** as expresións XPath para obter os seguintes conxuntos de nodos.

- Os discos de tecnoloxía "SCSI".
- Os atributos que figuran nos procesadores da marca "AMD".
- As máquinas de tipo "Semitorre" con sistema operativo "Windows XP".
- A configuración daquelas máquinas nas que figura un *gateway*.



**Tarefa 4. Rutas de localización con tests de nodo e predicados (II).** Esta tarefa é similar á anterior, pero traballando sobre un documento XML diferente.

Tomando como base o seguinte documento XML:

```
<?xml version="1.0" encoding="utf-8"?>  
<videoteca data_creación="24/02/2009">  
  <película id="1">  
    <importe moneda="dólar">13.56</importe>  
    <título>El santo</título>  
    <títuloorixinal>The Saint</títuloorixinal>  
    <ano>1997</ano>  
    <director>Phillip Noyce</director>  
    <xénero>Acción</xénero>  
    <duración>111</duración>  
    <!-- Elisabeth Shue -->  
    <actúa id="51"/>  
    <!-- Val Kilmer -->  
    <actúa id="156"/>  
  </película>  
  <película id="3">  
    <importe moneda="euro">22.18</importe>  
    <título>Leaving Las Vegas</título>  
    <títuloorixinal>Leaving Las Vegas</títuloorixinal>  
    <ano>1995</ano>  
    <director>Mike Figgis</director>  
    <xénero>Drama</xénero>  
    <duración>107</duración>  
    <!-- Nicolas Cage -->  
    <actúa id="187"/>  
    <!-- Elisabeth Shue -->  
    <actúa id="51"/>  
  </película>  
  <película id="4">  
    <importe moneda="dólar">11.52</importe>
```

```

<título>¿A quién ama Gilbert Grape?</título>
<títuloorixinal>What's Eating Gilbert Grape?</títuloorixinal>
<ano>1993</ano>
<director>Lasse Hallström</director>
<xénero>Drama</xénero>
<duración>118</duración>
<!-- Johnny Depp -->
<actúa id="139"/>
</película>
<actor id="51">
  <nome>Elisabeth Shue</nome>
  <sexo>muller</sexo>
  <datanacemento>06/10/1963</datanacemento>
  <url>http://www.imdb.com/name/nm0000223/</url>
</actor>
<actor id="139">
  <nome>Johnny Depp</nome>
  <sexo>home</sexo>
  <datanacemento>09/06/1963</datanacemento>
  <url>http://www.imdb.com/name/nm0000136/</url>
</actor>
<actor id="156">
  <nome>Val Kilmer</nome>
  <sexo>home</sexo>
  <datanacemento>31/12/1959</datanacemento>
</actor>
<actor id="187">>
  <nome>Nicolas Cage</nome>
  <sexo>home</sexo>
  <datanacemento>07/01/1964</datanacemento>
</actor>
</videoteca>

```

Escrebe as expresións XPath para obter os seguintes conxuntos de nodos.

- Os comentarios.
- Os comentarios da película "Leaving Las Vegas".
- As películas de xénero "Drama".
- Os nomes das actrices.
- Os elementos non baleiros (conteñen algún texto).

## Operadores

Nos predicados, ademais do operador "=", podemos utilizar outros operadores e funcións XPath para filtrar o conxunto de nodos en función do valor dalgũa estrutura do documento. XPath define os seguintes operadores que se poden empregar nas expresións.

### Operadores aritméticos Función

+ suma

- resta

\* multiplicación

div resultado entero de la división

mod módulo

### Operadores de comparación Función

= igual

!= no igual

< menor que; en XSLT, se necesita el operador &lt;

> mayor que; en XSLT, se recomienda usar el operador &gt;

<= menor que o igual; en XSLT, se necesita el operador &lt;

>= Mayor que o igual; en XSLT, se recomienda usar el operador &gt;

### Operadores lógicos Función

and Operador Y lógico

or Operador O lógico

## Funcións

XPath 1.0 define as seguintes funcións que se poden empregar nas expresións.

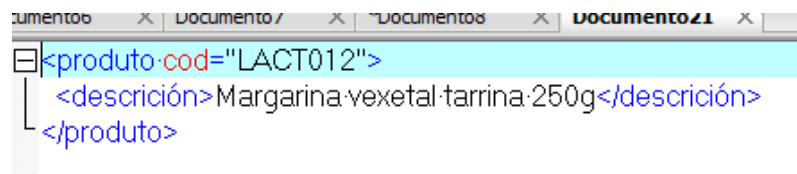
Función	Parámetros	Valor devolto	Descrición	Exemplo
boolean()	Conxunto de nodos, booleano, numérico ou cadea de caracteres	Booleano	Converte o parámetro a un valor booleano.	boolean(//produto)
false()		Booleano	Devolve falso.	false()
true()		Booleano	Devolve verdadeiro.	true()
not()	Conxunto de nodos, booleano, numérico ou cadea de caracteres	Booleano	Devolve verdadeiro se o valor do operando é falso, verdadeiro en caso contrario.	not(count(//produto) < 3)
lang()	Cadea de caracteres	Booleano	Devolve verdadeiro se a linguaxe definida con xml:lang coincide coa especificada no parámetro, falso en caso contrario.	lang("es")
ceiling()	Numérico	Numérico	Devolve o primeiro enteiro maior que o valor do parámetro.	ceiling(8 div 3)
floor()	Numérico	Numérico	Devolve o primeiro enteiro menor que o valor do parámetro.	floor(8 div 3)
round()	Numérico	Numérico	Devolve o enteiro máis próximo ao valor do parámetro.	round(8 div 3)
sum()	Conxunto de nodos	Numérico	Devolve a suma dos valores dos nodos que se pasan como parámetros.	sum(//produto/@importe)

concat()	Varias cadeas de caracteres	Cadea de caracteres	Concatena nunha cadea tódalas que se lle pasan como parámetros.	concat("Don ", //nome/text())
contains()	Dúas cadeas de caracteres	Booleano	Devolve verdadeiro se a primeira cadea contén á segunda, falso en caso contrario.	contains(//nome/text(), "Uxío")
normalize-space()	Cadea de caracteres	Cadea de caracteres	Devolve unha cadea como a que se lle pasa como parámetro, quitando os espazos ao comezo, ao final, e os duplicados.	normalize-space(//nome/text())
starts-with()	Dúas cadeas de caracteres	Booleano	Devolve verdadeiro se a primeira cadea comeza coa segunda, falso en caso contrario.	starts-with(//nome/text(), "Uxío")
string-length()	Cadea de caracteres	Númérico	Devolve o número de caracteres da cadea.	string-length(//nome/text())
substring()	1º: Cadea de caracteres 2º e 3º: Numérico	Cadea de caracteres	Da cadea que recibe como primeiro parámetro, devolve tantos caracteres como indique o terceiro parámetro, contando a partir da posición que indique o segundo parámetro.	substring(//nome/text(), 6, 7)
substring-after()	Dúas cadeas de caracteres	Cadea de caracteres	Devolve a cadea do primeiro parámetro a partir da primeira ocorrencia do segundo parámetro.	substring-after(//nome/text(), " ")
substring-before()	Dúas cadeas de caracteres	Cadea de caracteres	Devolve a cadea do primeiro parámetro anterior á primeira ocorrencia do segundo parámetro.	substring-before(//nome/text(), " ")
translate()	Tres cadeas de caracteres	Cadea de caracteres	Devolve a cadea do primeiro parámetro, substituindo tódalas ocorrencias dos caracteres do segundo parámetro polos caracteres do terceiro parámetro.	translate(//endereço/text(), " ", "-")
string()	Conxunto de nodos, booleano, numérico ou cadea de caracteres	Cadea de caracteres	Devolve o parámetro convertido a unha cadea de caracteres.	string(//nome)
count()	Conxunto de nodos	Númérico	Devolve o número de nodos do conxunto de nodos.	count(//produto)
id()	Cadea de caracteres	Conxunto de nodos	Devolve o nodo do elemento co ID especificado como parámetro.	id("G0097763")
last()		Conxunto de nodos	Devolve o número de nodos no contexto actual. Pódese empregar para acceder ao último nodo do contexto.	//produto[last()]
local-name()	Conxunto de nodos	Cadea de caracteres	Devolve o nome local (non o nome cualificado) do primeiro nodo no conxunto de nodos que se lle pasa como parámetro.	local-name(//nome)
name()	Conxunto de nodos	Cadea de caracteres	Devolve o nome cualificado do primeiro nodo no conxunto de nodos que se lle pasa como parámetro.	name(//nome)
namespace-uri()	Conxunto de nodos	Cadea de caracteres	Devolve o URI do espazo de nomes do primeiro nodo no conxunto de nodos que se lle pasa como parámetro.	namespace-uri(//produto)
position()		Númérico	Devolve a posición (comezando con 1) do nodo contexto no conxunto de nodos do contexto actual.	//produto[position()=2]

Podes encontrar mais información no seguinte [enlace](#).

A función `position()` emprégase habitualmente no predicado para seleccionar o nodo correspondente a unha posición determinada dentro do conxunto de nodos do contexto. Isto pódese facer tamén de forma abreviada indicando a posición directamente no predicado, de tal xeito que ámbalas dúas expresións seguintes son equivalentes:

```
//produto[position()=2]
//produto[2]
```

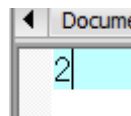


## Outras expresións

Aínda que a ruta de localización é o tipo máis común de expresión en XPath, podemos empregar os operadores e funcións anteriores para crear diversos tipos de expresións (moitas delas non devolven un conxunto de nodos, como por exemplo:

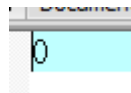
- Contar o número de produtos dunha venda (devolve un número):

```
count(//produto)
```



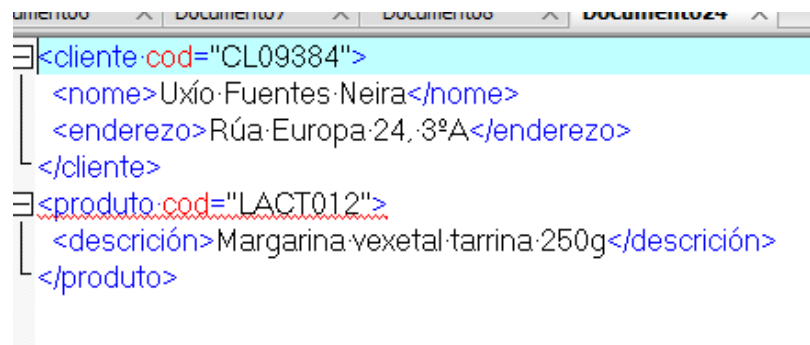
- Comprobar se o número de produtos dunha venda cumpre ou non certas condicións (devolve un valor booleano (ou 0, 1):

```
count(//produto) > 3 and count(//produto) < 7
```



- Obter os datos dun produto e os do cliente (devolve un conxunto de nodos):

```
//produto[2] | //cliente
```



- Comprobar o valor do código dun produto determinado (devolve un valor booleano ou 0,1):

```
//produto[2]/@cod = "LACT012"
```



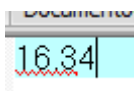
- Comprobar se existe algún produto cun código determinado (devolve un valor booleano ou 0,1):

```
//produto/@cod = "LACT012"
```



- Obter o importe da venda, cambiando a coma por un punto (devolve unha cadea de texto):

```
substring(translate(/venda/importe, ",", "."), 1, 5)
```



**Mais exemplos. Rutas de localización avanzadas.** Empregaremos expresións XPath avanzadas que inclúan algúns dos operadores e funcións propios de XPath

Tomando como base o mesmo **documento XML da tarefa 3**, escribiremos as expresións XPath para obter:

- As máquinas con nome comezando por "PC".

```
//máquina[starts-with(@nome, "PC")]
```

- O nome das máquinas con máis de un disco duro.

```
//máquina[count(hardware/disco)>1]/@nome
```

- A suma dos GB de memoria RAM de todas as máquinas.

```
sum(//memoria)
```

- A suma das capacidades dos discos duros de tipo "SCSI".

```
sum(//disco[@tecnoloxía="SCSI"]/@capacidade)
```

- Os nomes das máquinas das que se coñeza a súa cantidade de memoria, pero non a tecnoloxía desta.

```
//máquina[hardware/memoria][not(hardware/memoria/@tecnoloxía)]/@nome
```

- As máquinas con sistema operativo da familia Windows e gravadora de DVD.

```
//máquina[starts-with(config/OS, "Windows")][hardware/gravadora/@tipo="DVD"]
```

- As máquinas nas que non figure o sistema operativo.

```
//máquina[not(config/OS)]
```

- O nome da máquina e do SO daquelas máquinas con máis de un disco duro.

```
//máquina[count(hardware/disco)>1]/@nome |
//máquina[count(hardware/disco)>1]/config/OS
```

- A configuración daquelas máquinas nas que figura unha dirección IP pero non o *gateway*.

```
//máquina/config[IP and not(gateway)]
```

- As direccións IP dentro da rede 192.168.10.0/24.

Dúas solucións:

```
//IP[starts-with(., "192.168.10.")]
//IP[starts-with(text(), "192.168.10.")]
```

- Os nomes das máquinas con procesador multinúcleo e 2GB ou menos de memoria.

```
//máquina[hardware/procesador/@num_nucleos>1][hardware/memoria <= 2]/@nome
```

- As máquinas que teñan disco duro (un ou varios) con capacidade total maior de 1000GB.

```
//máquina[sum(hardware/disco/@capacidade)>1000]
```

- As máquinas que teñan disco duro (un ou varios) con capacidade total menor de 80GB.

```
//máquina[hardware/disco][sum(hardware/disco/@capacidade)<80]
```



### **Mais exemplos. Rutas de localización avanzadas (II).**

Tomando como base o mesmo **documento XML da tarefa 4**, escribiremos as expresións XPath para obter:

- As películas nas que o título sexa igual ao título orixinal.

```
//película[título=títuloorixinal]
```

- O atributo "*id*" correspondente á actriz "Elisabeth Shue".

```
//actor[nome="Elisabeth Shue"]/@id
```

- Os títulos das películas nas que participou o actor con atributo id=51.

```
//película[actúa/@id=51]/título
```

- Baseándote nos dous resultados anteriores, escribe a expresión correspondente aos títulos das películas nas que participou a actriz de nome "Elisabeth Shue".

```
//película[actúa/@id=//actor[nome="Elisabeth Shue"]/@id]/título
```



- O título da última película que aparece no documento.

Dúas solucións:

```
//película[count(//película)]/título  
//película[last()]/título
```

- A suma dos importes daquelas películas nas que o mesmo figura en dólares.

```
sum(//importe[@moneda="dólar"])
```

- A suma en euros dos importes das películas, considerando un cambio de  $1\text{€} = 1,4\text{\$}$ .

```
sum(//importe[@moneda="dólar"]) div 1.4 + sum(//importe[@moneda="euro"])
```