

Linguagem de Consulta XPath

Caminhos, predicados, ancestrais e funções

Prof. Walmes Zeviani

walmes@ufpr.br

Laboratório de Estatística e Geoinformação
Departamento de Estatística
Universidade Federal do Paraná

Introdução

Motivação

- ▶ **XPath** é uma linguagem de consulta para XML.
- ▶ Ela é baseada em notação de caminhos e é bastante flexível.
- ▶ Com XPath pode-se:
 - ▶ Utilizar regras lógicas em atributos e conteúdo.
 - ▶ Descrever caminhos baseados em relação vertical e horizontal.
 - ▶ Aplicar funções numéricas e de texto.
- ▶ Existem outras formas de apontar para extração mas geralmente são subótimas comparadas ao XPath.

Introdução

Objetivos

- ▶ Introduzir o básico da sintaxe XPath para consultas.
- ▶ Apresentar uso dos predicado e genealogia em consultas XPath.
- ▶ Ilustrar o uso de predicados, operadores e funções.

XPath

Características

- ▶ É uma **linguagem de consulta** para endereçamento e extração de informação em documentos XML/HTML.
- ▶ É um padrão do *World Wide Web Consortium* (W3C).
- ▶ Só funciona com a representação DOM do documento e descreve caminhos pelos nós e ramos da árvore.
- ▶ Uma instrução XPath retorna o conteúdo que bate com o caminho descrito.
- ▶ Em alguns aspectos, se assemelha a Expressões Regulares.

Documentação do XPath no W3C

Documentação

1. XPath está na versão 3.1.
2. Documentação para cada versão:
<https://www.w3.org/TR/xpath/all/>.
3. De acordo com o **código fonte**, a `libxml2` usa o XPath 1.0.

Tutoriais

3. https://www.w3schools.com/xml/xpath_intro.asp.
4. <https://docs.marklogic.com/guide/xquery/xpath;>
5. <https://www.tutorialspoint.com/xpath/>.
6. <http://www.utools.nl/downloads/XPathReference.pdf>.

Cartões de referência e testadores online

Cartões de referência

1. <http://ricostacruz.com/cheatsheets/xpath.html>.
2. <http://xpath.alephzarro.com/content/cheatsheet.html>.
3. http://www.mulberrytech.com/quickref/XSLT_1quickref-v2.pdf.

Testadores online de XPath

1. <http://xpather.com/>.
2. <http://codebeautify.org/Xpath-Tester>.
3. <http://www.webtoolkitonline.com/xml-xpath-tester.html>.
4. <http://www.freeformatter.com/xpath-tester.html>.

Descendant selectors

<code>h1</code>	<code>//h1</code>	?
<code>div p</code>	<code>//div//p</code>	?
<code>ul > li</code>	<code>//ul/li</code>	?
<code>ul > li > a</code>	<code>//ul/li/a</code>	
<code>div > *</code>	<code>//div/*</code>	
<code>:root</code>	<code>/</code>	?
<code>:root > body</code>	<code>/body</code>	

Figura 1. Seleção de descendentes. Fonte: <https://devhints.io/xpath>.

Prefixes

Prefix	Example	What
//	//hr[@class='edge']	Anywhere
./	./a	Relative
/	/html/body/div	Root

Begin your expression with any of these.

Figura 2. Significado dos prefixos. Fonte: <https://devhints.io/xpath>.

Steps and axes

```
//      ul      /      a[@id='link']
```

```
Axis      Step      Axis      Step
```

Axes

Axis	Example	What
/	//ul/li/a	Child
//	//[@id="list"]//a	Descendant

Separate your steps with /. Use two (//) if you don't want to select direct children.

Figura 3. Combinação de seleção e prefixos. Fonte: <https://devhints.io/xpath>.

Child axis

```
# both the same
//ul/li/a
//child::ul/child::li/child::a
```

`child::` is the default axis. This makes `//a/b/c` work.

```
# both the same
# this works because `child::li` is truthy, so the predicate succeeds
//ul[11]
//ul[child::11]
```

```
# both the same
//ul[count(li) > 2]
//ul[count(child::li) > 2]
```

Figura 4. Seleção de filhos. Fonte: <https://devhints.io/xpath>.

Descendant-or-self axis

```
# both the same  
//div//h4  
//div/descendant-or-self::h4
```

// is short for the descendant-or-self:: axis.

```
# both the same  
//ul//[last()]  
//ul/descendant-or-self::[last()]
```

Figura 5. Seleção de descendentes. Fonte: <https://devhints.io/xpath>.

Using axes

```
//ul/li           # ul > li
//ul/child::li   # ul > li (same)
//ul/following-sibling::li # ul ~ li
//ul/descendant-or-self::li # ul li
//ul/ancestor-or-self::li # $('ul').closest('li')
```

Steps of an expression are separated by /, usually used to pick child nodes. That's not always true: you can specify a different "axis" with ::.

//	ul	/child::	li
Axis	Step	Axis	Step

Figura 6. Diferentes formas de caminho vertical e horizontal. Fonte: <https://devhints.io/xpath>.

Siblings

<code>h1 ~ ul</code>	<code>//h1/following-sibling::ul</code>	?
<code>h1 + ul</code>	<code>//h1/following-sibling::ul[1]</code>	
<code>h1 ~ #id</code>	<code>//h1/following-sibling::*[@id="id"]</code>	

Figura 7. Seleção de irmãos ou caminhos horizontais. Fonte: <https://devhints.io/xpath>.

Axis	Abbrev	Notes
ancestor		
ancestor-or-self		
attribute	@	@href is short for attribute::href
child		div is short for child::div
descendant		
descendant-or-self	//	// is short for /descendant-or-self::node()/
namespace		
self	.	. is short for self::node()
parent is short for parent::node()
following		
following-sibling		
preceding		
preceding-sibling		
There are other axes you can use.		

Figura 8. Outras especificações de eixos verticais e horizontais. Fonte: <https://devhints.io/xpath>.

Attribute selectors

<code>#id</code>	<code>//*[@id="id"]</code>	?
<code>.class</code>	<code>//*[@class="class"] ..kinda</code>	
<code>input[type="submit"]</code>	<code>//input[@type="submit"]</code>	
<code>a#abc[for="xyz"]</code>	<code>//a[@id="abc"][@for="xyz"]</code>	?
<code>a[rel]</code>	<code>//a[@rel]</code>	
<code>a[href^=' /']</code>	<code>//a[starts-with(@href, ' /']</code>	?
<code>a[href\$='.pdf']</code>	<code>//a[ends-with(@href, '.pdf']</code>	
<code>a[href*=': //']</code>	<code>//a[contains(@href, ': //']</code>	
<code>a[rel~='help']</code>	<code>//a[contains(@rel, 'help')] ..kinda</code>	

Figura 9. Seleção baseada nos atributos. Fonte: <https://devhints.io/xpath>.

Indexing

```
//a[1]           # first <a>
//a[last()]    # last <a>
//ol/li[2]     # second <li>
//ol/li[position()=2] # same as above
//ol/li[position()>1] # :not(:first-child)
```

Use [] with a number, or last() or position().

Figura 10. Seleção pela posição. Fonte: <https://devhints.io/xpath>.

Order selectors

<code>ul > li:first-child</code>	<code>//ul/li[1]</code>	?
<code>ul > li:nth-child(2)</code>	<code>//ul/li[2]</code>	
<code>ul > li:last-child</code>	<code>//ul/li[last()]</code>	
<code>li#id:first-child</code>	<code>//li[@id="id"][1]</code>	
<code>a:first-child</code>	<code>//a[1]</code>	
<code>a:last-child</code>	<code>//a[last()]</code>	

Figura 11. Mais seleção por posição. Fonte: <https://devhints.io/xpath>.

Predicates

```
//div[true()]  
//div[@class="head"]  
//div[@class="head"][@id="top"]
```

Restricts a nodeset only if some condition is true. They can be chained.

Figura 12. Uso de predicados nos nós. Fonte: <https://devhints.io/xpath>.

Operators

```
# Comparison
//a[@id = "xyz"]
//a[@id != "xyz"]
//a[@price > 25]

# Logic (and/or)
//div[@id="head" and position()=2]
//div[(x and y) or not(z)]
```

Use comparison and logic operators to make conditionals.

Boolean functions

```
not(expr)                # button[not(starts-with(text(),"Submit"))]
```

Figura 13. Operadores de comparação e lógicos. Fonte: <https://devhints.io/xpath>.

Unions

```
//a | //span
```

Use | to join two expressions.

Nesting predicates

```
//section[//h1[@id='h1']]
```

This returns `<section>` if it has an `<h1>` descendant with `id='h1'`.

Figura 14. União de expressões e encadeamento. Fonte: <https://devhints.io/xpath>.

Node functions

```
name()           # //[starts-with(name(), 'h')]
text()           # //button[text()='Submit']
                 # //button/text()

lang(str)
namespace-uri()

count()          # //table[count(tr)=1]
position()       # //ol/li[position()=2]
```

Type conversion

```
string()
number()
boolean()
```

Figura 15. Funções aplicáveis aos nós. Fonte: <https://devhints.io/xpath>.

String functions

```
contains()           # font[contains(@class,"head")]
starts-with()       # font[starts-with(@class,"head")]
ends-with()         # font[ends-with(@class,"head")]

concat(x,y)
substring(str, start, len)
substring-before("01/02", "/")  #=> 01
substring-after("01/02", "/")   #=> 02
translate()
normalize-space()
string-length()
```

Figura 16. Funções de texto aplicáveis aos nós. Fonte: <https://devhints.io/xpath>.

Using nodes

```
# Use them inside functions
//ul[count(li) > 2]
//ul[count(li[@class='hide']) > 0]

# This returns `<ul>` that has a `<li>` child
//ul[li]
```

You can use nodes inside predicates.

Chaining order

```
a[1][@href='/']
a[@href='/'][1]
```

Order is significant, these two are different.

Figura 17. Combinando funções e precedência de seleção. Fonte: <https://devhints.io/xpath>.

Exemplos de uso do XPath

Usar esse conteúdo no <http://xpather.com/> para avaliar expressões XPath.

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>.
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//span
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//ul[@class = 'ingredientes']/li
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//ul[last()]/li[position() < last() - 2]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//ul/li[position() > 1 and position() < last()]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//ul[count(li) = 4]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//*[ @class and not(./li)]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//li[starts-with(text(), 'D')]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```



```
//p/parent::div
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//div[./p]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//p/ancestor::div
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

```
//span[contains(., 'reais') or contains(., 'pessoas')]
```

XML mode

Format Save

```
<body>
  <div type="orçamento">
    <p>
      Essa sobremesa custa <span class="valor">10 reais</span>
      e serve <span class="quantidade">6 pessoas</span>
    </p>
  </div>
  <div type="sobremesa">
    <ul class="ingredientes">
      <li>3 ovos.</li>
      <li>150 ml de leite.</li>
      <li>3 colheres de manteiga.</li>
      <li>Massa preparada de bolo.</li>
    </ul>
    <ul class="preparo">
      <li>Deixe o forno preaquecer por 20 min a 180 graus.</li>
      <li>Bata os ingredientes até uniformizar.</li>
      <li>Unte a forma.</li>
      <li>Despeje na forma.</li>
      <li>Leve ao fogo por aproximadamente 1 hora.</li>
    </ul>
  </div>
</body>
```

Resumo de recursos do XPath

- ▶ Prefixos e eixos:
 - ▶ `./, //, ///`.
 - ▶ Verticais: `child::`, `parent::`, `descendant::`, `ancestor::`, etc.
 - ▶ Horizontais: `following-sibling::`, `preceding-sibling::`.
- ▶ Metacacteres: `., . . ., @, *`.
- ▶ Predicados: `[...]`.
- ▶ Operadores de comparação: `=, !=, <, <=, >, >=`.
- ▶ Operadores lógicos: `and`, `or`, `not()` e `|`.
- ▶ Funções: `text()`, `name()`.
- ▶ Funções numéricas: `count()`, `position()`, `last()`, etc.
- ▶ Funções de texto: `contains()`, `starts-with()`, etc.

Resumo

- ▶ XPath é uma linguagem **flexível e poderosa** para consultar XML/HTML.
- ▶ HTML são profundos na hierarquia e requerem uso de predicados para **delimitar a consulta**.
- ▶ O domínio dos recursos do XPath são fundamentais para especificar buscas precisas, fáceis de criar e manter.
- ▶ O emprego de predicados, eixos e funções potencializa a aplicação do XPath.
- ▶ O mesmo resultado pode ser conseguido com diversas expressões.
- ▶ Os predicados podem ser combinados de várias formas.
- ▶ Como expressões regulares, e necessário equilibrar **especificidade** e **generalidade** para reduzir manutenção do código.