

XPath

XPath

▶ `/html/body/div[2]/div/section[2]/main/div/p[7]`

XPath

- ▶ Especificação: <http://www.w3.org/TR/xpath>
- ▶ Uma expressão XPath seleciona um conjunto de nodos
- ▶ Operadores principais:
 - ▶ / para dar um “passo” na árvore XML (percorrer uma relação pai-filho)
 - ▶ // para dar vários “passos” de uma vez (percorrer uma relação ascendente-descendente)

Exemplo

► /empregados/empregado

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Contexto

- ▶ Cada / muda o contexto atual da consulta:
/empregados

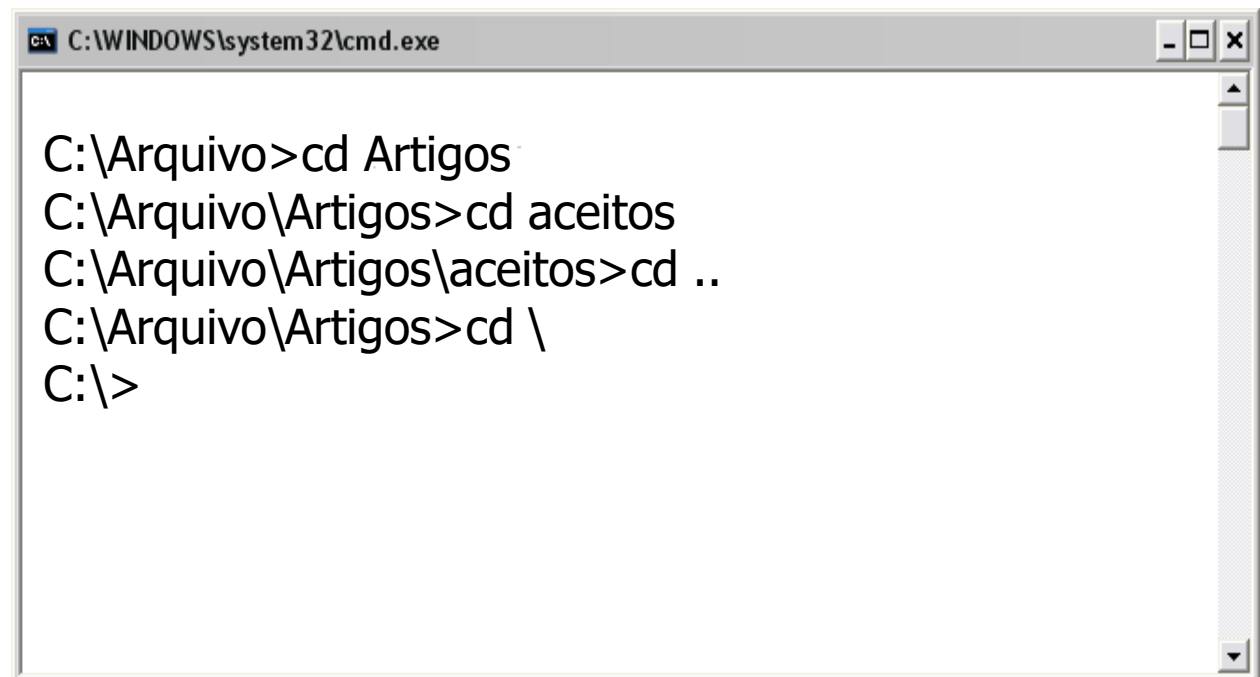
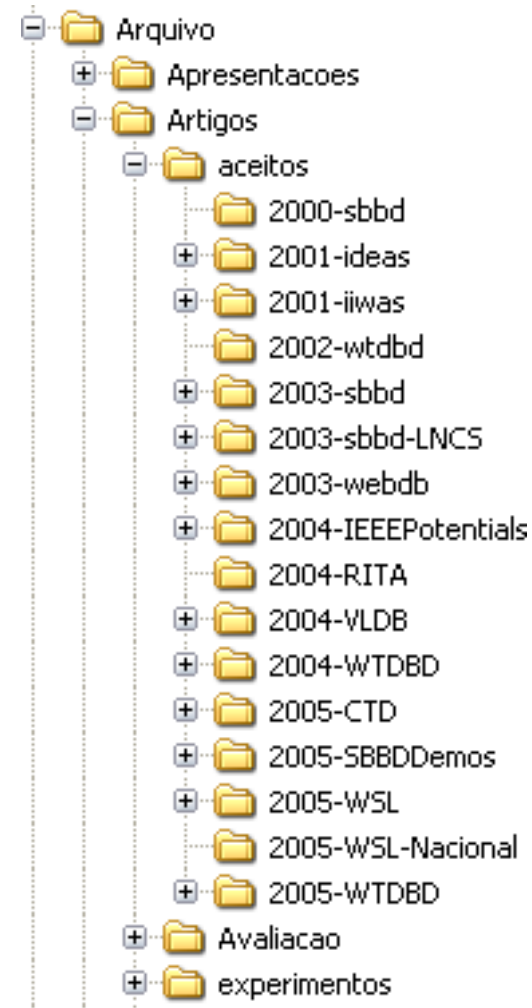
```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Contexto

- ▶ Cada / muda o contexto atual da consulta:
/empregados/empregado

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

É como uma árvore de diretórios...



Retorno da Expressão

- Conjunto de nodos retornados é sempre o especificado pelo último passo do caminho

/empregados/empregado/nome

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Operador //

- ▶ Retorna os descendentes a partir do contexto atual
/empregados//nome //sobrenome

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Atributos

- Usa-se “@” na frente do nome do atributo
/empregados/empregado/@cod

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Operador ..

- ▶ Seleciona o pai do contexto atual
/empregados/empregado/..

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Operador *

- Substitui um passo do caminho

//empregado/*

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

XPath na prática...



Prática 1

- ▶ Usando o documento XML fornecido, crie expressões XPath para as seguintes consultas:
 - a) Selecionar as instituições dos autores do artigo
 - b) Selecionar todos os parágrafos das seções do artigo
 - c) Selecionar nomes dos autores do artigo propriamente dito e das referencias bibliográficas
 - d) Selecionar pai do elemento endereço
 - e) Selecionar avô do elemento paragrafo
 - f) Selecionar todas as ocorrências de endereço



Filtros

- ▶ Restringem o conjunto de nodos selecionados
- ▶ Podem ser colocados em qualquer passo do caminho

Filtros

- Sintaxe: Expressão booleana entre colchetes

//empregado[@cod="E01"]

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Contexto do Filtro

- ▶ Sempre o último passo percorrido antes do filtro
- ▶ Retorno nunca é o que está no filtro, mas o último passo do caminho da expressão

//empregado[nome="Ana"]

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Filtro de Posição

//empregado[1]

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Filtro de Posição

//nome[1]

ATENÇÃO: o filtro de posição leva em conta o pai do nó como contexto, por isso neste exemplo ele retorna os dois elementos nome, e não apenas o primeiro que aparece no documento

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

AND, OR e NOT são aceitos

//empregado[inicial-meio="S." and nome="João"]/sobrenome

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Testes de elementos

- ▶ O nome de um elemento pode aparecer representando um elemento que deve estar presente como um filho
- ▶ Selecionar um elemento empregado se ele contém um sub-elemento inicial-meio: `//empregado[inicial-meio]`

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Prática 2

- a) Selecionar o autor cujo nome é Maria Ana
- b) Selecionar a obra da bibliografia cujo ano é 1999 e o local é University of Pennsylvania
- c) Selecionar a seção cujo número é s2 e que contém um parágrafo cujo conteúdo é ...
- d) Selecionar o atributo título das seções



Funções

- ▶ XPath possui muitas funções
- ▶ A maioria delas é utilizada dentro dos filtros
- ▶ Uma lista completa está disponível aqui:
<http://www.w3.org/TR/xpath#corelib>

Testes de posição

- ▶ Função `position()` retorna a localização sequencial do elemento testado
 - ▶ Selecionar somente o primeiro parágrafo dentre os já selecionados pelo padrão

```
//empregado[position()=1]
```

ou

```
//empregado[1]
```



Testes de posição

- ▶ Função `last()` localiza o último elemento (retorna o número da posição do último elemento)
 - ▶ Selecionar o último empregado
`//empregado[last()]`
- ▶ Função `count()` retorna o número de ocorrências de um elemento
 - ▶ Selecionar seções que contenham apenas dois parágrafos
`//secao[count(paragrafo)=2]`



Função NOT

- ▶ Função `not()` para reverter o resultado do teste
 - ▶ Selecionar todas as notas, exceto a terceira
`//nota[not(position())=3]`
 - ▶ Selecionar uma nota que não contém um elemento título
`//nota[not(titulo)]`
 - ▶ Selecionar todos os capítulos, exceto aquele que tenha o atributo número com valor 10
`//capitulo[not(@numero='10')]`



Comparações

- ▶ Selecionar todos paragrafos, mas não o último

`//paragrafo[position() != last()]`

- ▶ Outras comparações:

`//paragrafo[position() > 2]`

`//paragrafo[position() >= 3]`

`//paragrafo[position() > 2 and position() < last()]`

`//paragrafo[position() = 2 or position() = 4]`



Tratamento de Strings

- ▶ Função `contains(par1, par2)` retorna true se `par1` contém o texto em `par2`

`par1` pode ser `text()` ou `.`



Função contains()

- ▶ Usando "text()", testa somente o conteúdo textual do elemento
- ▶ Selecionar título que contenha a palavra "relacional"

```
//titulo[contains(text(), "relacional")]
```

```
<titulo>Modelo relacional</titulo>
```



Tratamento de Strings

- ▶ Usando ".", testa o elemento secao e seus subelementos
- ▶ Selecionar secao que contenha a palavra "relacional" em seu texto ou no texto de algum de seus descendentes

```
//secao[contains(., "relacional")]
```

```
<secao>Esta secao apresenta...
```

```
  <paragrafo>O modelo relacional ...</paragrafo>
```

```
  <paragrafo>Como já mencionado, ...</paragrafo>
```

```
</secao>
```



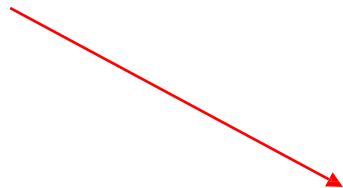
Tratamento de Strings

- ▶ Função `starts-with()` testa o texto no começo da string. Não pode haver espaço em branco.
 - ▶ Selecionar título que inicie com a palavra "Introdução"

```
//título[starts-with(., "Introdução")]
```

```
<título>Introdução a JSP</título>
```

```
<título>_Introdução a JSP</título>
```



Não funciona!



Tratamento de Strings

- ▶ Função `string()` converte o valor do argumento para string
 - ▶ Exemplo:

`string(//capitulo[1]/@numero)`, retorna o valor do atributo `numero` do primeiro capítulo, em formato string

- ▶ Função `normalize-space()`
 - ▶ No meio da string, reduz vários espaços em branco para um único caractere espaço
 - ▶ Remove completamente os espaços do início e fim da string

`//titulo[contains(normalize-space(.), "Introdução a JSP")]`

`<titulo> Introdução a JSP </titulo>`



Tratamento de Strings

- ▶ Função `translate()` converte caracteres de acordo com um esquema de mapeamento.
 - ▶ Uso: comparações case-insensitive
 - ▶ Parâmetros: string para converter, caracteres para modificar no texto fonte, e valores a serem colocados

```
//paragrafo[contains(translate(normalize-space(.),  
    "abcdefghijklmnopqrstuvwxyz",  
    "ABCDEFGHIJKLMNOPQRSTUVWXYZ"), "ELEMENTO")]
```

```
<paragrafo>EM XML, UM ELEMENTO É ...</paragrafo>  
<paragrafo>Um documento XML deve possuir um elemento raiz  
...</paragrafo>
```

- ▶ os dois elementos `paragrafo` são recuperados



Tratamento de Números

- ▶ Operadores + e - podem ser usados:
 - ▶ `//nota[3]` é equivalente a `//nota[1 + 2]`
- ▶ Função `mod()`
 - ▶ Fornece o resto de uma divisão truncada
 - ▶ Selecionar parágrafos pares:
`//paragrafo[position() mod 2 = 0]`



Filtros Múltiplos

- ▶ Usados para combinar um teste de posição e um outro tipo de teste
 - ▶ Selecionar nomes de companhias, e então extrair o terceiro nome da lista
`//nome[companhia][3]`

...

<nomes>

<nome>< Pessoa>...</ Pessoa></nome>

<nome>< companhia>...</ companhia></nome>

<nome>< companhia>...</ companhia></nome>

<nome>< companhia>...</ companhia></nome>

<nome>< companhia>...</ companhia></nome>

<nome>< Pessoa>...</ Pessoa></nome>

</nomes>

<nomes>

<nome>< Pessoa>...</ Pessoa></nome>

<nome>< companhia>...</ companhia></nome>

<nome>< Pessoa>... </ Pessoa></nome>

</nomes>

...



Filtros Múltiplos

- ▶ Usados para combinar um teste de posição e um outro tipo de teste
 - ▶ Selecionar o terceiro nome, fornecendo o nome da companhia (só seleciona se for uma companhia!)
`//nome[3][companhia]`

...

<nomes>

<nome><essoa>...</essoa></nome>

<nome><companhia>...</companhia></nome>

<nome><companhia>...</companhia></nome>

<nome><companhia>...</companhia></nome>

<nome><companhia>...</companhia></nome>

<nome><essoa>...</essoa></nome>

</nomes>

<nomes>

<nome><essoa>...</essoa></nome>

<nome><companhia>...</companhia></nome>

<nome><companhia>...</companhia></nome>

<nome><essoa>... </essoa></nome>

</nomes>

...



Prática 3

- ▶ Usando o mesmo documento XML, crie expressões Xpath
 - a) Retorne todas as seções do artigo que possuem pelo menos um subelemento figura
 - b) Selecione as seções ímpares
 - c) Selecione as seções ímpares que possuem pelo menos um subelemento figura
 - d) Encontre autores que possuam "Ana" no nome. Não devem estar nas referencias bibliográficas.
 - f) A versão do artigo
 - g) Selecione o parágrafo que tenha ambos os atributos "numero" e "tipo"



Prática 3

- j) Um parágrafo cujo tamanho da string que ele contém é 27
- k) Selecione o terceiro parágrafo de uma seção
- l) Selecione um parágrafo que contenha o nome do segundo autor do artigo e sua instituição. Entre estes dois dados existe a string "from"
- m) Selecione todo e qualquer parágrafo que tenha a string "two years". Trate o uso de maiúsculas e minúsculas!!
- n) Selecionar seções que contenham somente dois parágrafos
- o) Selecione os elementos "ano" descendentes de bibliografia
- p) Encontre a instrução de processamento de algum elemento paragrafo.

