

Banco de Dados na Web e XML (resolução dos exercícios)

Banco de Dados: Teoria e Prática

André Santanchè e Patrícia Cavoto
Institute of Computing - UNICAMP
Outubro 2016

Exercício 1

Representar:

Sr. Horácio é autor da página <http://www.paleo.org/dino.html>
em XML

Exercício 1 (resolução)

Representar:

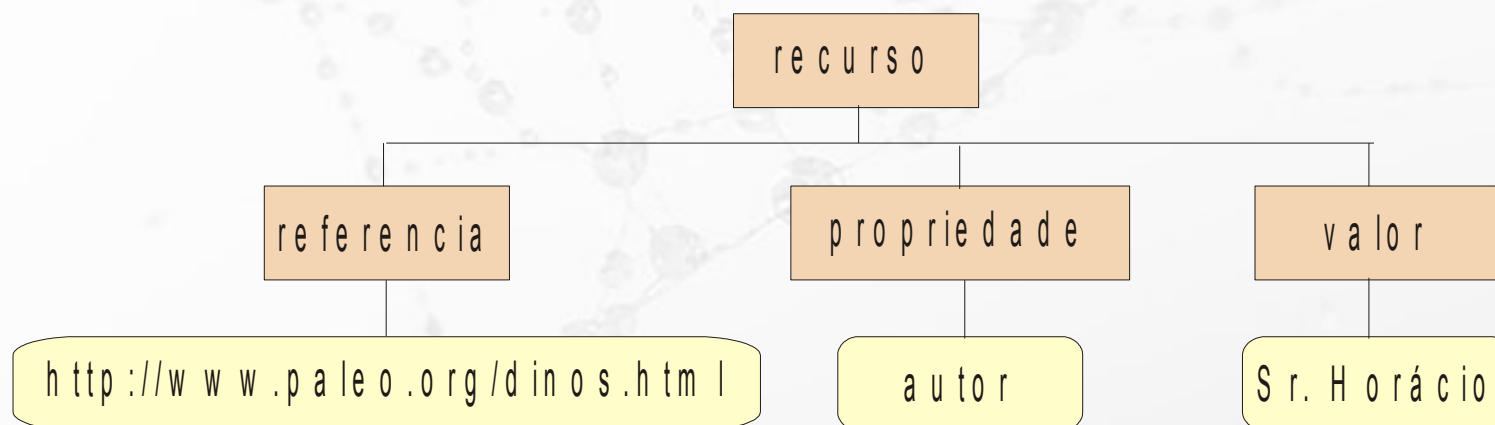
Sr. Horácio é autor da página <http://www.paleo.org/dino.html>
em XML

```
<escritor nome="Sr. Horácio">  
  <autorPagina href="http://www.paleo.org/dino.html"/>  
</escritor>
```

Exercício 1

Outra Possível Representação 1

```
<recurso>  
  <referencia>  
    http://www.paleo.org/dinos.html  
  </referencia>  
  <propriedade>autor</propriedade>  
  <valor>Sr. Horácio</valor>  
</recurso>
```



Exercício 1

Outra Possível Representação 2

```
<recurso referencia="http://www.paleo.org/dinos.html">  
  <propriedade rotulo="autor">Sr. Horácio</propriedade>  
</recurso>
```

referencia = http://www.paleo.org/dinos.html

rotulo = autor

recurso

propriedade

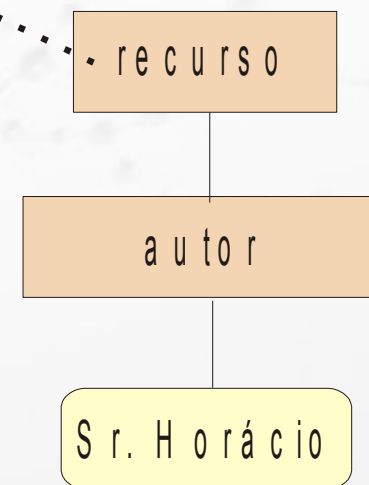
Sr. Horácio

Exercício 1

Outra Possível Representação 3

```
<recurso referencia="http://www.paleo.org/dinos.html">  
  <autor>Sr. Horácio</autor>  
</recurso>
```

referencia = http://www.paleo.org/dinos.html



Exercício 1

Outra Possível Representação 4

```
<recurso referencia="http://www.paleo.org/dinos.html" autor="Sr. Horacio"/>
```

referencia = http://www.paleo.org/dinos.html

rotulo = autor

recurso

Exercício 2

Representar:

Sr. Horácio é autor da página <http://www.paleo.org/dino.html>
em XML usando relacionamento entre
elementos

Exercício 2 (resolução) Representar:

Sr. Horácio é autor da página <http://www.paleo.org/dino.html>
em XML usando relacionamento entre
elementos

```
<registro>  
  <escritor nome="Sr. Horácio" ref="pg-dinos"/>  
  <pagina id="pg-dinos" href="http://www.paleo.org/dino.html"/>  
</fichario>
```

Exercício 3

- Escreva um modelo relacional compatível com este exemplo:

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Exercício 3

(resolução)

- Escreva um modelo relacional compatível com este exemplo:

Individuo(Nome, Idade, Genero)

```
CREATE TABLE Individuo (  
  Nome VARCHAR(80) NOT NULL,  
  Idade INTEGER,  
  Genero VARCHAR(9),  
  PRIMARY KEY(Nome)  
);
```

Exercício 4

- Construa um comando SELECT que retorne dados equivalentes a este XPath

```
//individuo[idade>20]/@nome
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
nome="Quincas Borba"
-----
nome="Doriana Margarina"
```

Exercício 4

(resolução)

- Construa uma comando **SELECT** que retorne dados equivalentes a este XPath

```
//individuo[idade>20]/@nome
```

```
SELECT nome  
  FROM Individuo  
  WHERE idade > 20;
```

Exercício 5

- Qual a outra maneira de escrever esta query sem o where?
- Qual a diferença?

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return {data($i/@nome)}
```

Exercício 5 (resolução)

- Qual a outra maneira de escrever esta query sem o where?
- Qual a diferença?

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return {data($i/@nome)}
```

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo[idade>17])  
return {data($i/@nome)}
```

Exercício 6

- Escreva uma consulta SQL equivalente ao XQuery:

```
let $fichariodoc := doc('icunicamp:fichario.xml')
for $i in ($fichariodoc//individuo)
where $i[idade>17]
return {data($i/@nome)}
```


Exercício 6 (resolução)

- Escreva uma consulta SQL equivalente ao XQuery:

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return {data($i/@nome)}
```

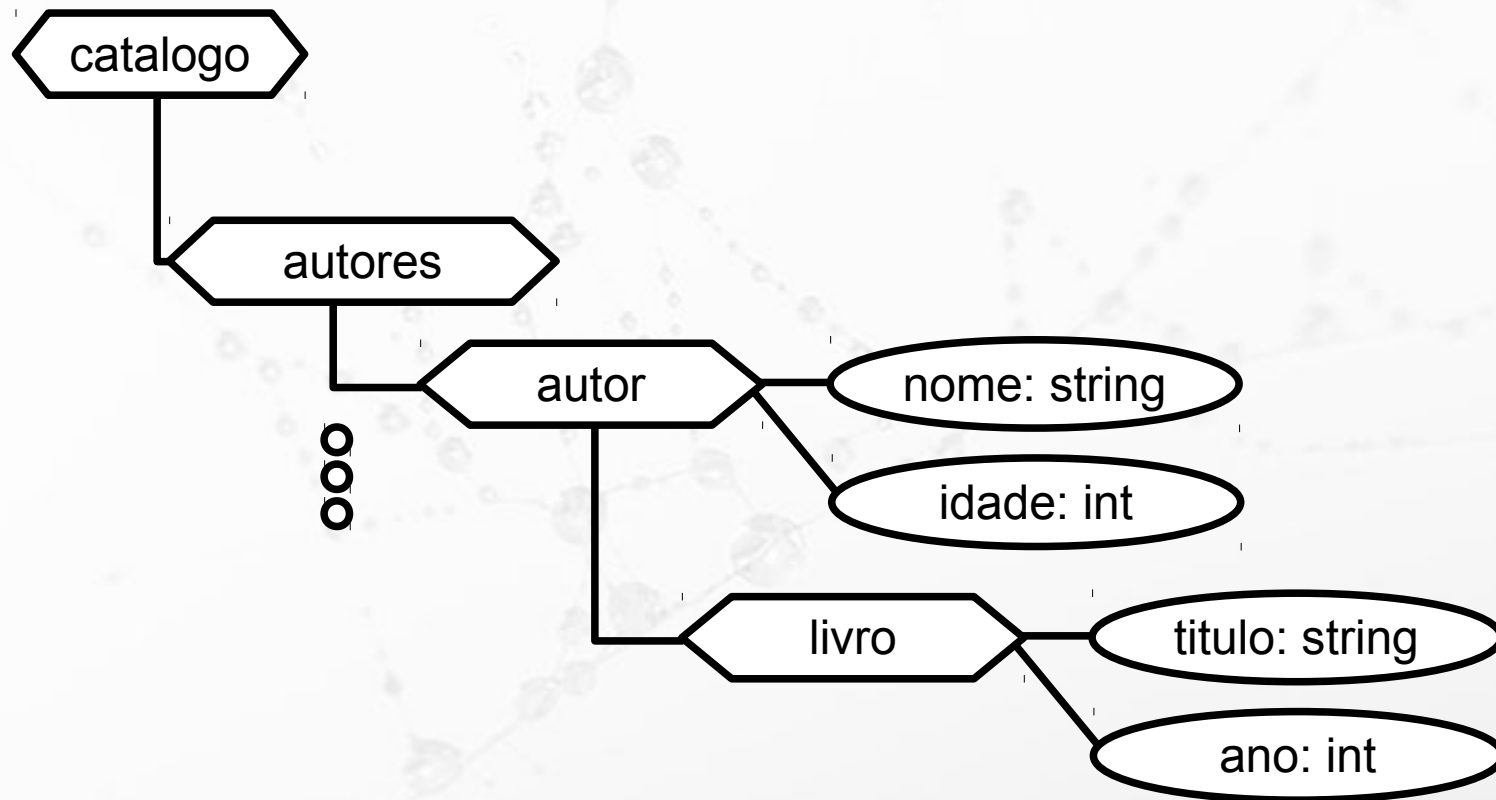
```
SELECT nome  
FROM Individuo  
WHERE idade > 17;
```

Exercício para Casa 1

- Escreva um XQuery que transforme o arquivo XML em uma sequência de INSERTS.

Exercício 7

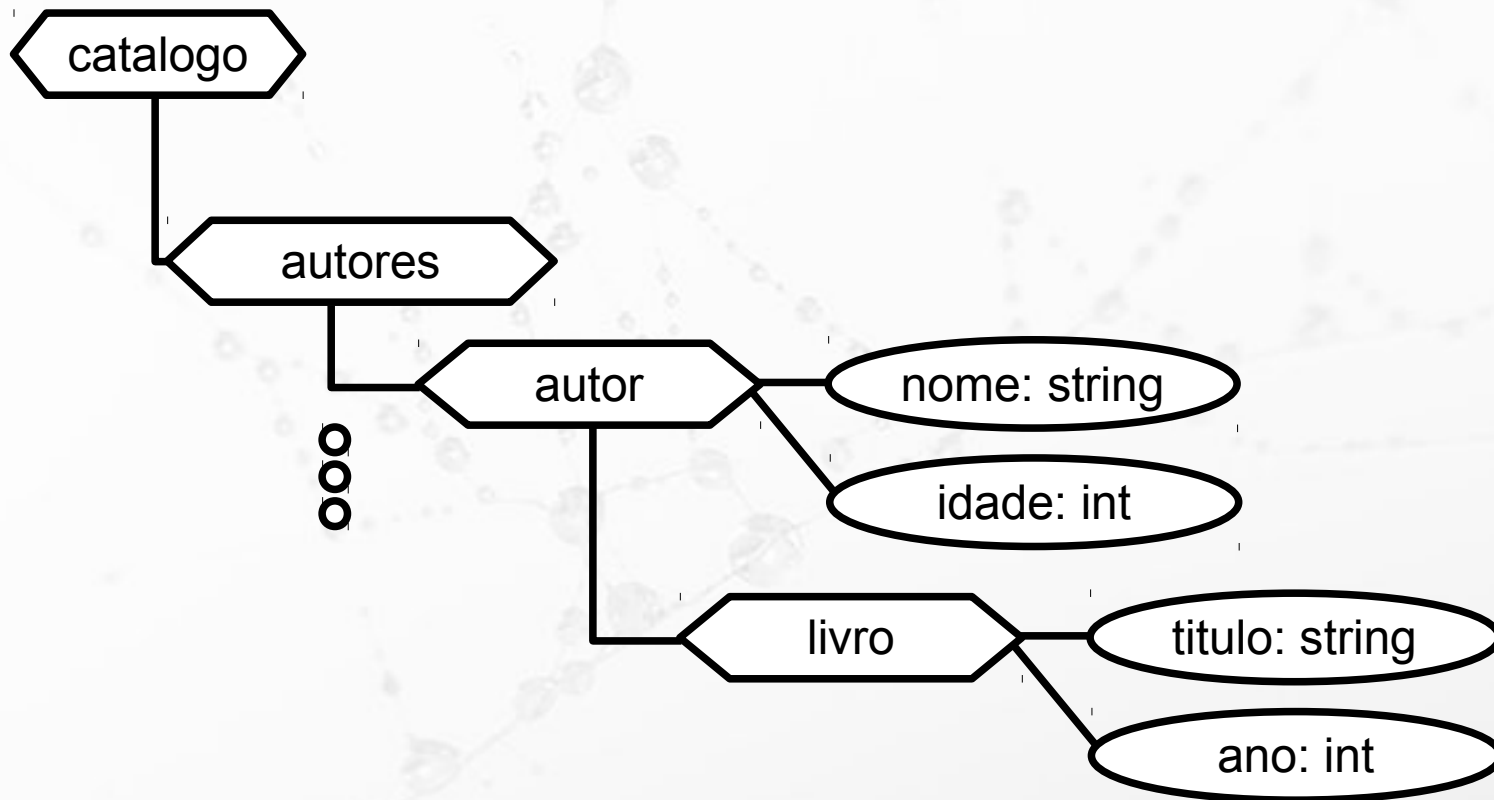
- Livros escritos após o ano 2000



Exercício 7

Autor/Livro Embedded

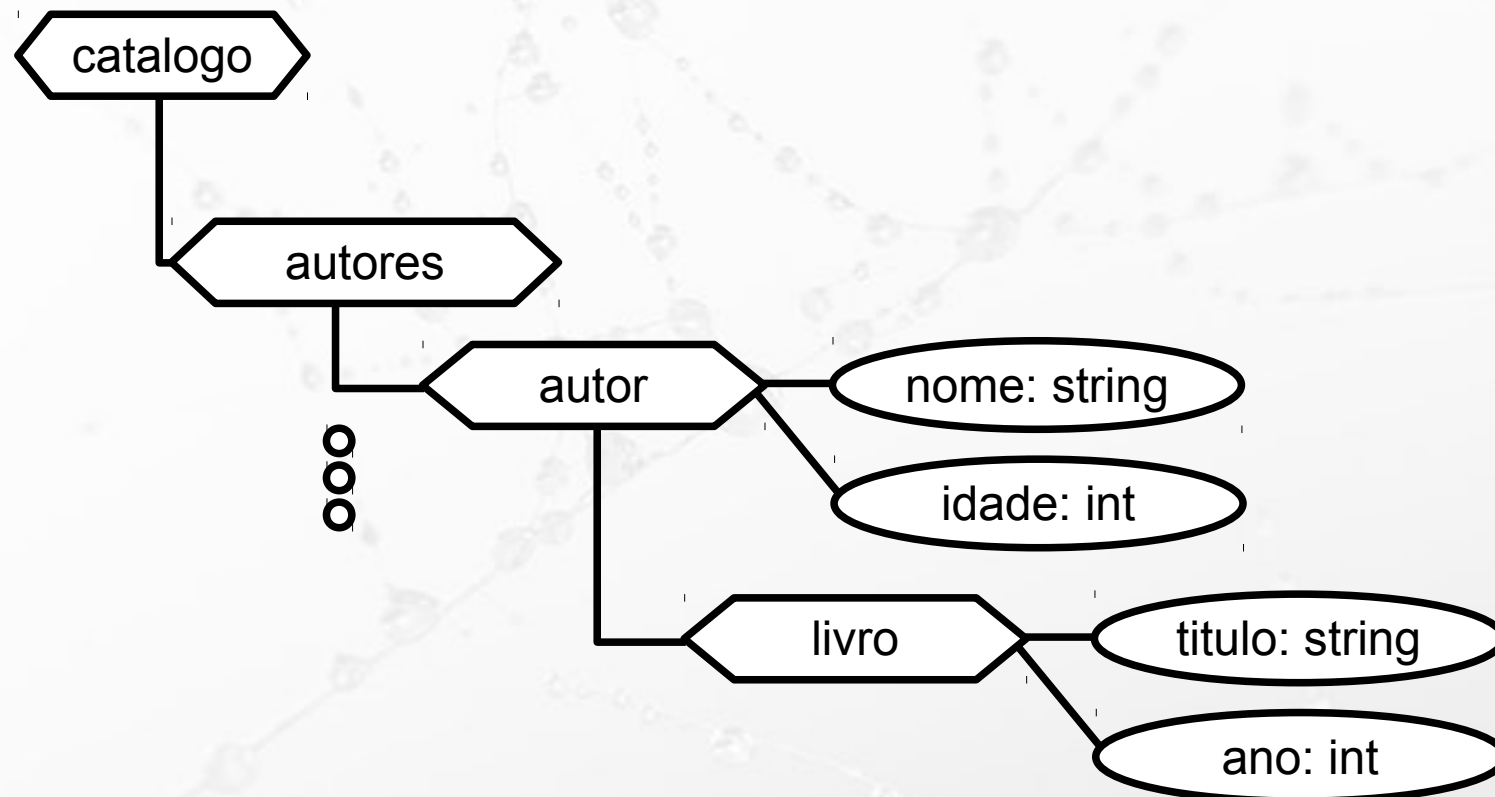
- Livros escritos após o ano 2000



Exercício 7

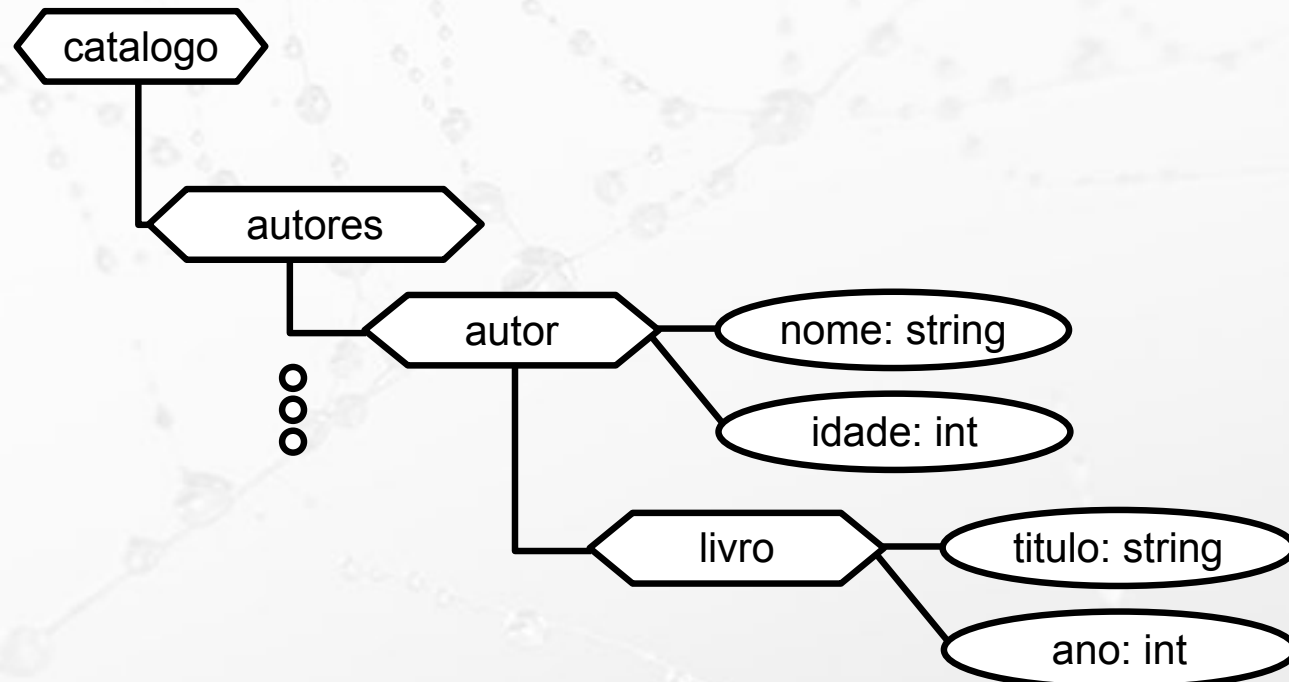
Autor/Livro Embedded

- Livros escritos após o ano 2000
- `return $publicadoc // livro[@ano>2000]`



Exercício 8

- Autores com mais de 50 anos e os livros que ele escreveu
 - use para o autor um tag especial <autorSenior>
 - associado ao tag coloque um atributo com seu nome



Exercício 8

(resolução)

- Autores com mais de 50 anos e os livros que ele escreveu
 - use para o autor um tag especial <autorSenior>
 - associado ao tag coloque um atributo com seu nome

```
let $fichariodoc := ...

for $a in ($fichariodoc//autor[@idade>50])
return <autorSenior nome="{data($a/@nome)}">
    {for $l in ($a/livro)
     Return <livro titulo="{data($l/@titulo)}"></livro>}
    </autorSenior>
```

Exercício 9

- Retorne quantas publicações são posteriores ao ano de 2011

Exercício 9 (resolução)

- Retorne quantas publicações são posteriores ao ano de 2011
- return
count(\$publicadoc // publication[year>2010])

Exercício 10

- Retorne a categoria cujo <label> em inglês seja 'e-Science Domain'

Exercício 10

(resolução)

- Retorne a categoria cujo <label> em inglês seja 'e-Science Domain'.
- for \$c in (\$publicadoc//categories/category)
where \$c/label[@lang='en-US'] = 'e-Science Domain'
return \$c

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.

Referências Bibliográficas

- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados**. Pearson, 6a edição em português.
- Lee, T.B. **Notation 3**. March 2006. on-line:
<http://www.w3.org/DesignIssues/Notation3.html>
- Lee, T.B.; Hendler, J. & Lassila, O. **The Semantic Web**. Scientific American, 2001, 284, 28-37
- Moats R. **URN Syntax**. Network Working Group, May 1997.
- Sollins, K. and Masinter, L. **Functional Requirements for Uniform Resource Names**. Network Working Group, December 1994.
- Wang, X.; Gorkitsky, R. & Almeida, J.S. **From XML to RDF: how semantic web technologies will change the design of 'omic' standards** Nat Biotech. 2005, 23, 1099-1103.
- Whiteside, Arliss. **URNs of definitions in ogc namespace**. version: 1.0.0, document: 05-010. January 2005.

Referências Bibliográficas

- Leise, F.; Fast, K.; Steckel, M. **What Is A Controlled Vocabulary?** Boxes and Arrows, Dezembro 2002, online: http://www.boxesandarrows.com/view/what_is_a_controlled_vocabulary_
- Amy J. Warner. **Taxonomy Primer**, online: <http://www.lexonomy.com/publications/aTaxonomyPrimer.html>, visitado em 20/08/2010.
- Wellisch, H. **Indexing from A to Z**. New York: H.W. Wilson, 1995. p. 214.
- Wilde, Erik. **XML Foundations** (slides). UC Berkeley iSchool, Aug 2006. <http://dret.net/lectures/xml-fall06/basics>



André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: <http://creativecommons.org/licenses/by-nc-sa/3.0/>