

Banco de Dados na Web e XML

Banco de Dados: Teoria e Prática

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

Motivação

Estudo de Caso

- Extrair automaticamente de páginas *Web*, informações sobre lançamentos de livro, como:
 - Assunto
 - Autor
 - Título

Editora A



DIABO E A TERRA DE SANTA CRUZ, O

Feitiçaria e religiosidade popular no Brasil colonial

Laura de Mello e Souza

408 páginas

R\$ 41,00

Primeiro estudo realizado no Brasil sobre a feitiçaria nos tempos coloniais, baseado em crônicas da época, devassas eclesiásticas e processos da Inquisição.



ENIGMA E COMENTÁRIO

Ensaio sobre literatura e experiência

Davi Arriguuci Jr.

240 páginas

R\$ 32,50

Davi Arriguuci analisa o processo de transposição simbólica da experiência de vida para o plano literário em autores brasileiros e hispano-americanos - Jorge Luis Borges, Manuel Bandeira, Fernando Gabeira e Rubem Braga, entre outros.



ANO DA MORTE DE RICARDO REIS, O

José Saramago

416 páginas

R\$ 43,00

Depois de uma temporada de auto-exílio no Brasil, o heterônimo de Fernando Pessoa está de volta a Lisboa. O ano é 1936, e ele tem de pôr de lado sua índole contemplativa para poder se situar em meio aos acontecimentos políticos de uma Europa em ebulição.



BOCA DO INFERNO

Ana Miranda

336 páginas

R\$ 32,00

O jogo da ambição e do poder na Bahia colonial de Gregório de Matos, numa trama ágil em que homens e mulheres se dilaceram entre o prazer e o pecado, o céu e o inferno. Livro de estreia da atriz, pesquisadora, editora e poeta. Lançado em 1989, renovou o romance histórico brasileiro.



ORIENTALISMO

O Oriente como invenção do Ocidente
Edward Said

Professor de literatura na Universidade de Columbia, Said mostra como o conceito de "Oriente" decorre de uma visão essencialista e superficial, e como a representação dos países

Editora B

ARTE-EDUCAÇÃO

BIOLOGIA

CIÊNCIAS

FÍSICA

GEOGRAFIA

HISTÓRIA

INGLÊS

LÍNGUA
PORTUGUESA

LITERATURA

MATEMÁTICA

PEDAGOGIA E
EDUCAÇÃO

QUÍMICA

NOTÍCIAS

digite seu e-mail para
receber nossas notícias

ok

ÚLTIMOS LANÇAMENTOS



O diário da rua
Esmeralda Ortiz
Literatura
Salamandra



Português - Uma proposta para o letramento 2 - consumível
Magda Soares
Didáticos
Editora Moderna



Coleção base matemática - 2ª edição
Manoel Paiva
Didáticos
Editora Moderna



Coleção base física - 2ª edição
Vários Autores
Didáticos
Editora Moderna



Caixa Mágica - Linguagem 2
Edições Educativas da
Editora Moderna
Didáticos
Editora Moderna



Ciclos, seriação e avaliação - Confrontos de lógicas
Luiz Carlos de Freitas
Apoio didático
Editora Moderna

Editora C

História da Arte | Graça Proença |

Um panorama completo e didático de toda a arte ocidental.
Um instrumento de trabalho completo e sério para o Professor e o aluno de Arte e Educação Artística.

- Da Pré-História ao Pós-Moderno...

[...] Veja Mais

História do Brasil | Nelson Piletti |

O livro didático que inaugurou uma nova maneira de apresentar a História do Brasil. Conteúdo crítico, que levanta denúncias sobre a situação de miséria de milhões de brasileiros.

Contém nume...

[...] Veja Mais

História Moderna e Contemporânea - Volume Único | Alceu Luiz Pazzinato, Maria H. Valente Senise |

Em uma linguagem clara, o livro apresenta a História de uma maneira sintonizada com as mudanças do mundo atual. Seu texto é baseado em produção historiográfica recente e valoriza a contextualização, a...

[...] Veja Mais

Inglês - Edição Compacta - Série Novo Ensino Médio | Amadeu Marques | Novo Ensino Médio - Edição Compacta

Este livro compõe-se de exercícios sobre os principais pontos gramaticais da língua inglesa, complementados por textos. Esses pontos são trabalhados sem

Como um ser humano vê



DIABO E A TERRA DE SANTA CRUZ, O

Fetiçaria e religiosidade popular no Brasil colonial

Laura de Mello e Souza

408 páginas

R\$ 41,00

Primeiro estudo realizado no Brasil sobre a feitiçaria nos tempos coloniais, baseado em crônicas da época, devassas eclesíásticas e processos da Inquisição.

Adicionar à minha sacola de compras

Livro

Título

Autor



ENIGMA E COMENTÁRIO

Ensaio sobre literatura e experiência

Davi Arrigucci Jr.

240 páginas

R\$ 32,50

Davi Arrigucci analisa o processo de transposição simbólica da experiência de vida para o plano literário em autores brasileiros e hispano-americanos - Jorge Luis Borges, Manuel Bandeira, Fernando Gabeira e Rubem Braga, entre outros.

Adicionar à minha sacola de compras



ANO DA MORTE DE RICARDO REIS, O

José Saramago

416 páginas

R\$ 43,00

Depois de uma temporada de auto-exílio no Brasil, o heterônimo de Fernando Pessoa está de volta a Lisboa. O ano é 1936, e ele tem de pôr de lado sua índole contemplativa para poder se situar em meio aos acontecimentos políticos de uma Europa em ebulição.

Adicionar à minha sacola de compras

Como uma máquina vê

<a href>



DIABO E A TERRA DE SANTA CRUZ, O

Feitiçaria e religiosidade popular no Brasil colonial

Laura de Mello e Souza

408 páginas

R\$ 41,00

Primeiro estudo realizado no Brasil sobre a feitiçaria nos tempos coloniais, baseado em crônicas da época, devassas eclesiásticas e processos da Inquisição.

 Adicionar à minha sacola de compras



ENIGMA E COMENTÁRIO

Ensaio sobre literatura e experiência

Davi Arrigucci Jr.

240 páginas

R\$ 32,50

Davi Arrigucci analisa o processo de transposição simbólica da experiência de vida para o plano literário em autores brasileiros e hispano-americanos - Jorge Luis Borges, Manuel Bandeira, Fernando Gabeira e Rubem Braga, entre outros.

 Adicionar à minha sacola de compras



ANO DA MORTE DE RICARDO REIS, O

José Saramago

416 páginas

R\$ 43,00

Depois de uma temporada de auto-exílio no Brasil, o heterônimo de Fernando Pessoa está de volta a Lisboa. O ano é 1936, e ele tem de pôr de lado sua índole contemplativa para poder se situar em meio aos acontecimentos políticos de uma Europa em ebulição.

 Adicionar à minha sacola de compras

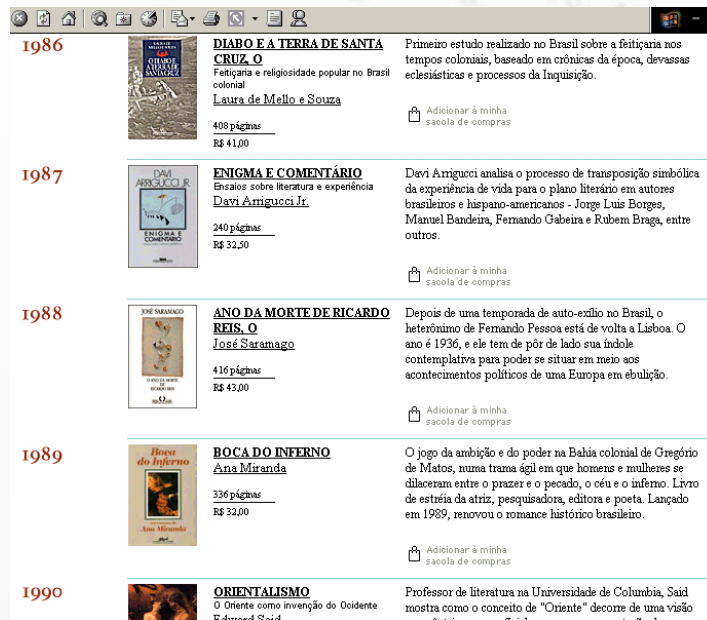
<tr>

<td>

<table>

Web - consumo humano

- Linguagem projetada principalmente para a apresentação de dados e definição de *links* entre documentos.
- Amplo uso de linguagem natural.



1986 **DIABO E A TERRA DE SANTA CRUZ, O**
Fetição e religiosidade popular no Brasil colonial
Laura de Mello e Souza
408 páginas
R\$ 41,00

1987 **ENIGMA E COMENTÁRIO**
Ensaio sobre literatura e experiência
Davi Arrigucci Jr.
240 páginas
R\$ 32,50

1988 **ANO DA MORTE DE RICARDO REIS, O**
José Saramago
416 páginas
R\$ 43,00

1989 **BOCA DO INFERNO**
Ana Miranda
336 páginas
R\$ 32,00

1990 **ORIENTALISMO**
O Oriente como invenção do Ocidente
Edward Said

Primeiro estudo realizado no Brasil sobre a feitiçaria nos tempos coloniais, baseado em crônicas da época, devassas eclesiásticas e processos da Inquisição.

Davi Arrigucci analisa o processo de transposição simbólica da experiência de vida para o plano literário em autores brasileiros e hispano-americanos - Jorge Luis Borges, Manuel Bandeira, Fernando Gabeira e Rubem Braga, entre outros.

Depois de uma temporada de auto-exílio no Brasil, o heterônimo de Fernando Pessoa está de volta a Lisboa. O ano é 1936, e ele tem de pôr de lado sua índole contemplativa para poder se situar em meio aos acontecimentos políticos de uma Europa em ebulição.

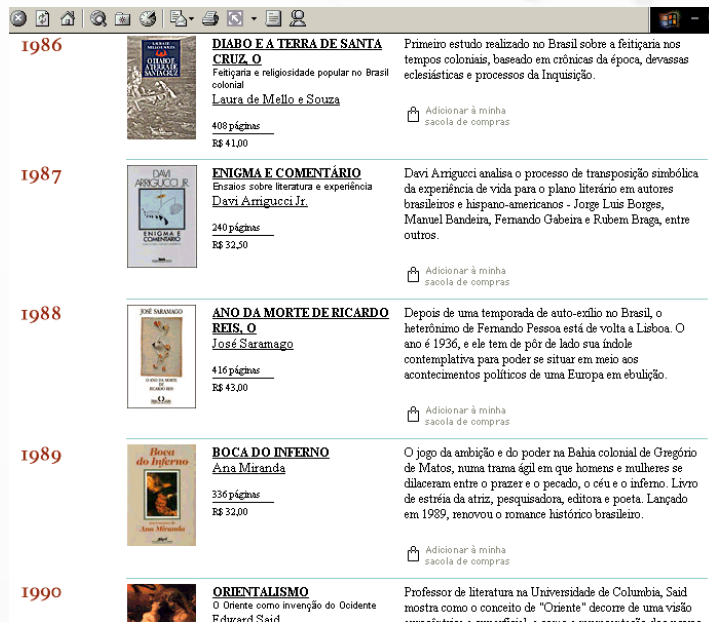
O jogo da ambição e do poder na Bahia colonial de Gregório de Matos, numa trama ágil em que homens e mulheres se dilaceram entre o prazer e o pecado, o céu e o inferno. Livro de estria da atriz, pesquisadora, editora e poeta. Lançado em 1989, renovou o romance histórico brasileiro.

Professor de literatura na Universidade de Columbia, Said mostra como o conceito de "Oriente" decorre de uma visão essencialmente essencial, a ser a representação dos ocidentais.



Web - consumo por computadores

- Papel dos computadores limitado à leitura e apresentação dos documentos.
- Carência de estruturas de dados adequadas para outros propósitos.



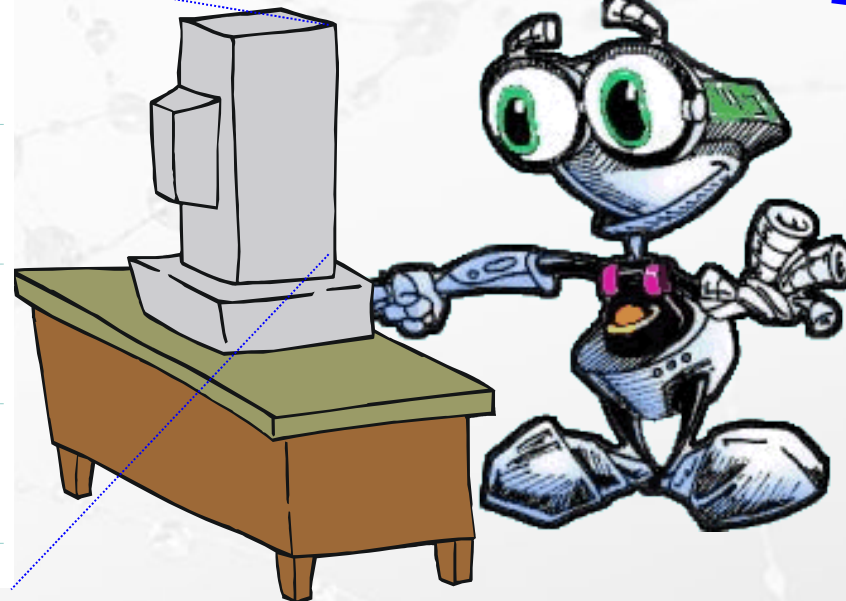
1986 **DIABO E A TERRA DE SANTA CRUZ, O** Primeiro estudo realizado no Brasil sobre a feitiçaria nos tempos coloniais, baseado em crônicas da época, devassas eclesásticas e processos da Inquisição.
Laura de Mello e Souza
408 páginas
R\$ 41,00

1987 **ENIGMA E COMENTÁRIO** Davi Arrigucci analisa o processo de transposição simbólica da experiência de vida para o plano literário em autores brasileiros e hispano-americanos - Jorge Luis Borges, Manuel Bandeira, Fernando Gabeira e Rubem Braga, entre outros.
Davi Arrigucci Jr.
240 páginas
R\$ 32,50

1988 **ANO DA MORTE DE RICARDO REIS, O** Depois de uma temporada de auto-exílio no Brasil, o heterônimo de Fernando Pessoa está de volta a Lisboa. O ano é 1936, e ele tem de pôr de lado sua índole contemplativa para poder se situar em meio aos acontecimentos políticos de uma Europa em ebulição.
José Saramago
416 páginas
R\$ 42,00

1989 **BOCA DO INFERNO** O jogo da ambição e do poder na Bahia colonial de Gregório de Matos, numa trama ágil em que homens e mulheres se dilaceram entre o prazer e o pecado, o céu e o inferno. Livro de estreia da atriz, pesquisadora, editora e poeta. Lançado em 1989, renovou o romance histórico brasileiro.
Ana Miranda
336 páginas
R\$ 32,00

1990 **ORIENTALISMO** Professor de literatura na Universidade de Columbia, Said mostra como o conceito de "Oriente" decorre de uma visão essencialista e essencialista a como representação dos países



Conteúdo

```
<header> Galeria </header>
<section>
  <aside>
    Página [01]
    ...
  </aside>
  <section class="ce">
    ...
    <figure><img src="" />
    ...
  </section>
</section>
<footer>
  Cabeças de Dinossauros
  ...
</footer>
```

Estilo

<header>

<section>

Apresentação

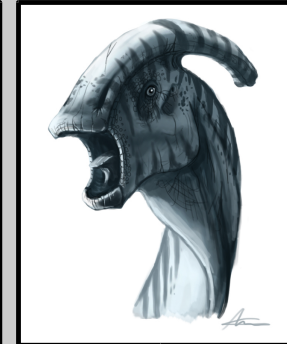
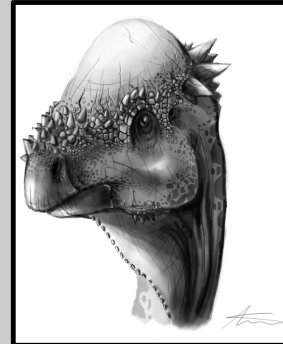
<aside>

Galeria

Página [01]

próxima >

< anterior



Cabeças de Dinossauros

Autor: [*highdarktemplar](#)

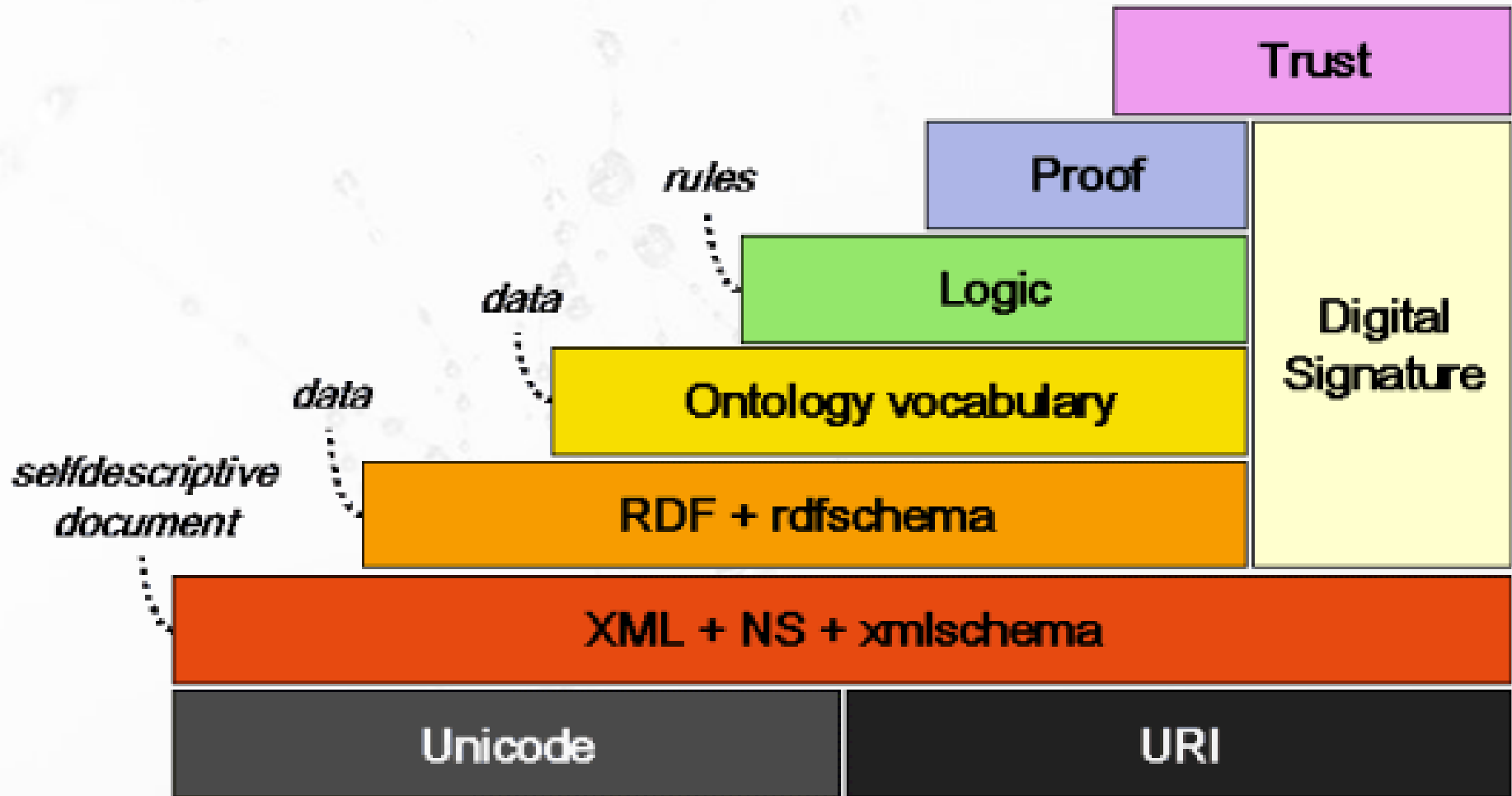
[<http://highdarktemplar.deviantart.com/>]

- **Cabeçalho**
 - Galeria
- **Secundário**
 - Número da página
 - Navegação: próximo/anterior
- **Central**
 - Sequência de imagens
- **Rodapé**
 - Dados do autor

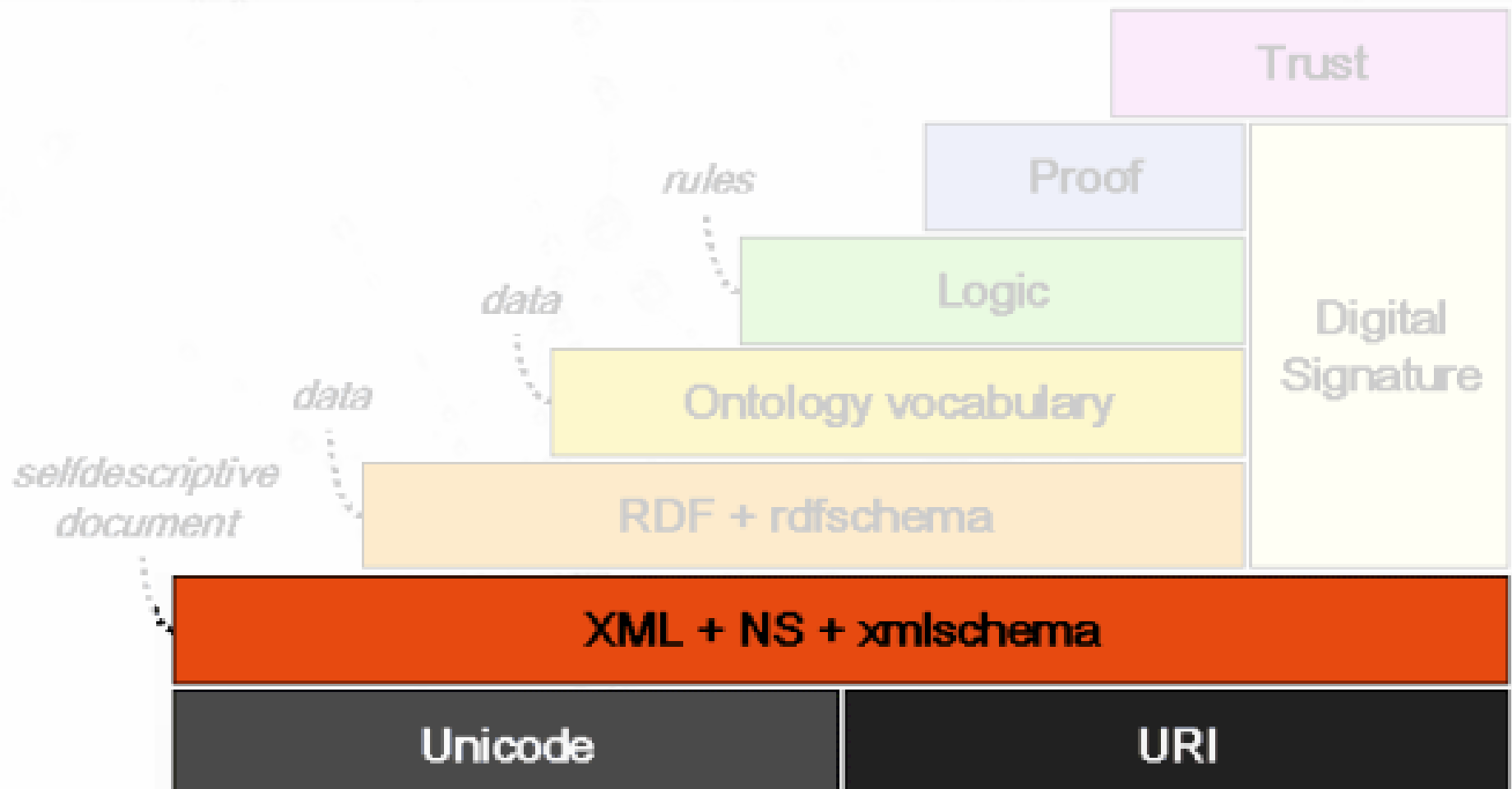
Plano

Web Semântica

Semantic Web



Foco desta Aula

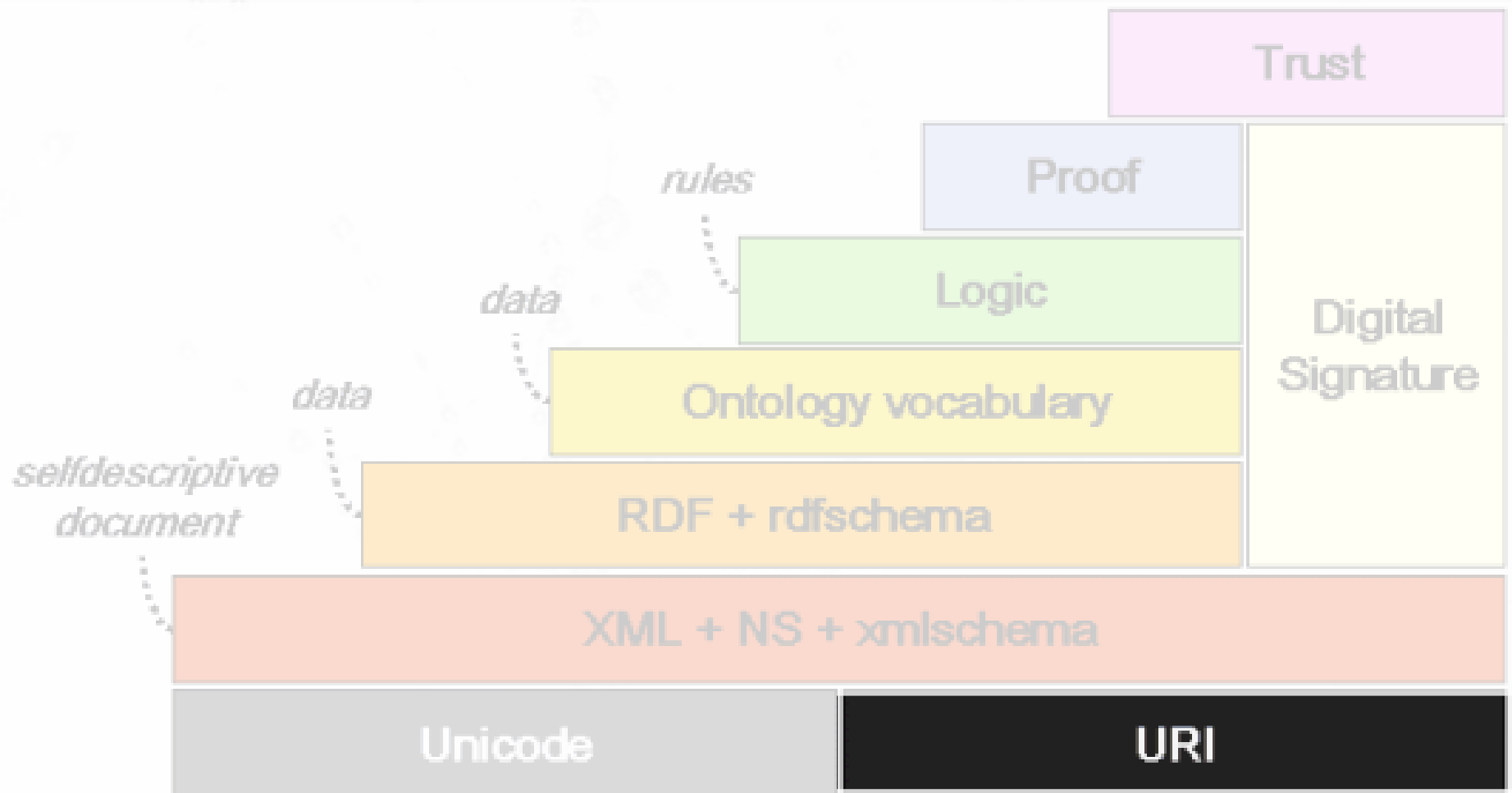


Web dos Dados

Web of Data

- A Web também pode ser vista como um grande **Repositório de Dados** que pode ser acessado pelas máquinas
- Por isso a Web Semântica também é chamada **Web dos Dados**

Componente 1: URI





URI

- A identificação de um recurso é feita através de um URI - Uniform Resource Identifier.
- URI = URL ou URN



URI

- URL (*Uniform Resource Locator*): identifica recursos por meio de sua localização física na Internet.

Ex.: <http://www.paleo.org>

<ftp://ftp.unicamp.br>

<mailto:horacio@paleo.org>

- URN (*Uniform Resource Names*): identificador é relacionado indiretamente com sua localização física na rede (exige um resolver).

Ex.: <urn:ogc:def:uom:celsius>

<urn:mpeg:mpeg21:dii:iswc:T-041.220.506-1>

Usando uma URN OGC

■ Como definir uma unidade Celsius?

`urn:ogc:def:uom:celsius`

1. É uma definição OGC
2. É uma unidade de medida (baseado na tabela)
3. Temperatura Celsius

crs:	coordinate reference systems
datum:	datums
meridian:	prime meridians
ellipsoid:	ellipsoids
cs:	coordinate systems
axis:	coordinate system axes
coordinateOperation:	coordinate operations
method:	operation methods
parameter:	operation parameters
group:	operation parameter groups
derivedCRSType:	derived CRS type codes
verticalDatumType:	vertical datum type codes
pixelInCell:	PixelInCell codes
rangeMeaning:	meaning codes
axisDirection:	axis direction codes
uom:	units of measure

Combinação de URL e URN

- URL Persistente:
 - Tal como URN: Identificador relacionado indiretamente ao endereço real
 - Tal como URL: Sob a forma de URL indica o resolver

Ex.: <http://purl.org/dc/elements/1.1/>
<http://doi.acm.org/10.1145/274440.274441>

Possibilidades da URI

- Fazer referência a um recurso.



<http://www.paleo.org/dinos.html>

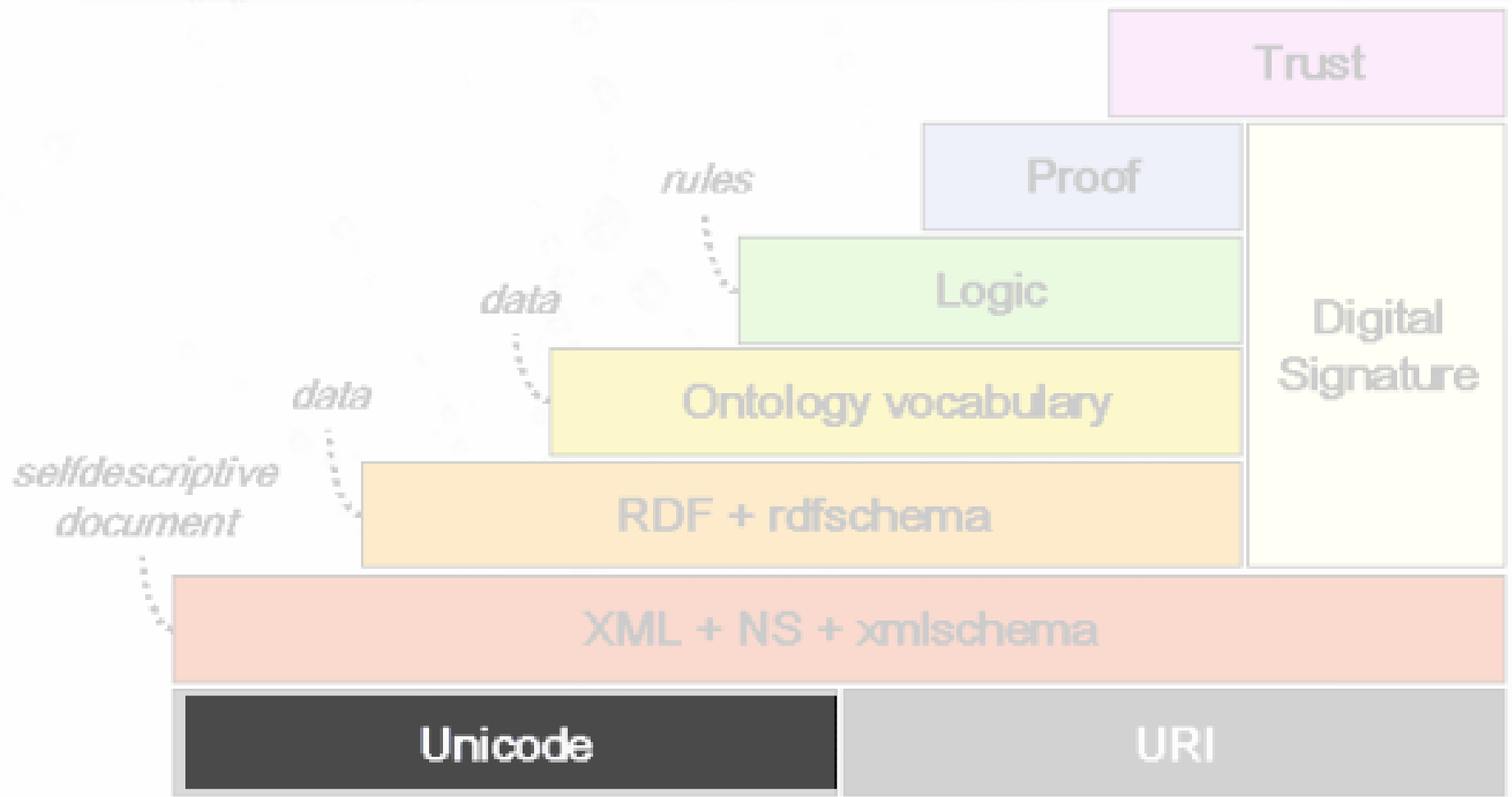
Possibilidades da URI

- Fazer referência a um elemento dentro de um documento.

```
<__>  
  <__>  
    <__>.....</__>  
    <__ id="elem">  
      ...  
    </__>  
  <__/>  
</__>  
<__> ..... </__>  
</__>
```

<http://www.paleo.org/doc.xml#elem>

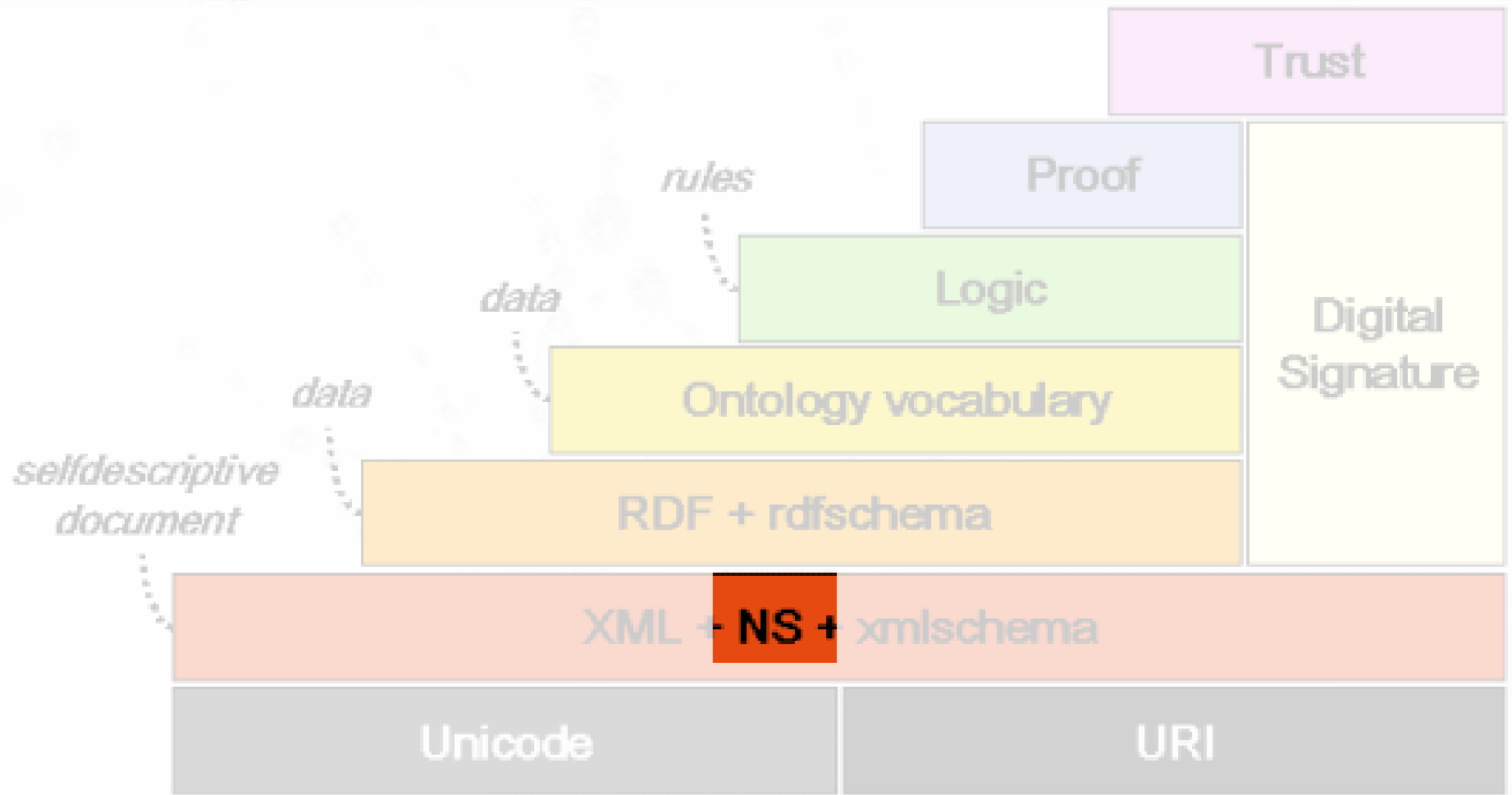
Componente 2: Unicode



Unicode

- Padrão para codificação de caracteres em texto
- Suporte à internacionalização e localização
- Mais de 128.000 caracteres
- 135 scripts (conjunto de caracteres para suporte a línguas) modernos e históricos, bem como símbolos

Componente 3: Namespaces



Namespaces

- URIs Dublin Core

<http://purl.org/dc/elements/1.1/creator>

<http://purl.org/dc/elements/1.1/title>

<http://purl.org/dc/elements/1.1/publisher>

- URIs vCard

<http://nwalsh.com/rdf/vCard#Name>

<http://nwalsh.com/rdf/vCard#Address>

<http://nwalsh.com/rdf/vCard#Organization>

Namespaces

- Demarca vocabulários
- Aumenta a legibilidade

Namespaces

- URIs Dublin Core

dc: ⇒ <http://purl.org/dc/elements/1.1/>

<http://purl.org/dc/elements/1.1/creator>

<http://purl.org/dc/elements/1.1/title>

<http://purl.org/dc/elements/1.1/publisher>

dc:creator

dc:title

dc:publisher

Namespaces

- URIs vCard

vcard: ⇒ <http://nwalsh.com/rdf/vCard#>

<http://nwalsh.com/rdf/vCard#Name>

<http://nwalsh.com/rdf/vCard#Address>

<http://nwalsh.com/rdf/vCard#Organization>

vcard:Name

vcard:Address

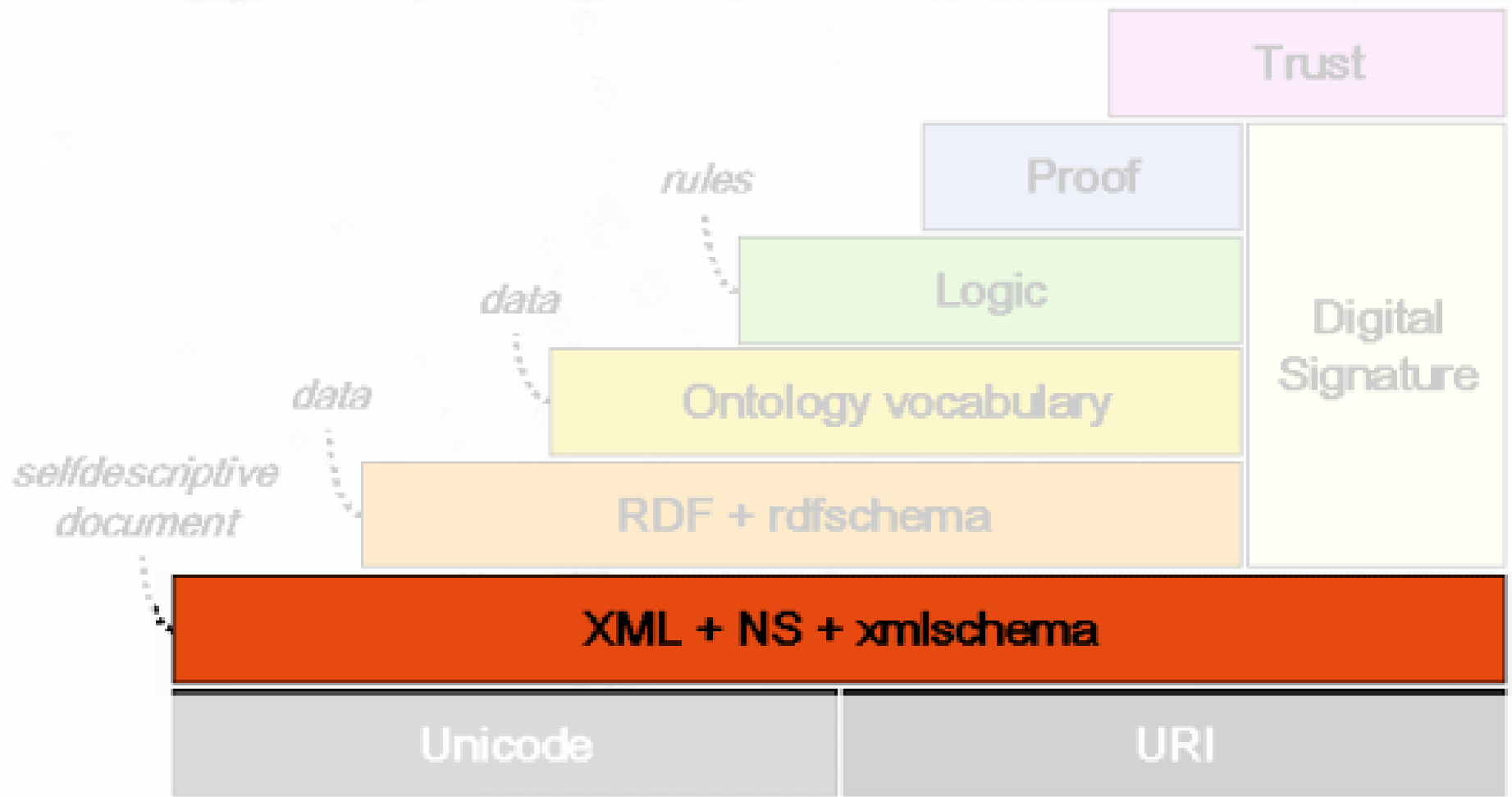
vcard:Organization

Namespaces

- Qualifica nomes de elementos e/ou atributos, conforme o vocabulário a que pertencem.

```
<ger:sentença xmlns:publ='http://www.publicar.org/esquema/'  
              xmlns:ger='http://www.gerais.org/vocab#'>  
  <publ:autor cpf="487.526.548-74"> Horácio </publ:autor>  
  <ger:ação> escreveu o  
    <publ:publicação>  
      <tipo> livro </tipo>  
      <título> Vida dos Dinossauros </título>  
    </publ:publicação>  
  </ger:ação>  
</ger:sentença>
```

Componente 4: XML



XML - eXtensible Markup Language

XML

- Lançada em 1996 como uma versão simplificada da SGML (*Standard Generalized Markup Language*), para ser utilizada na *Web*.

XML

Dimensões

- Modelo Lógico Hierárquico, baseado em Documentos
- Metalinguagem e Linguagem para a Web

XML

**Modelo Lógico Hierárquico, baseado
em Documentos**

O Caso da Nota Fiscal



Green Leaf Design
111 Main Street
Pleasanton, CA 99999
555.555.4444
info@greenleafdesign.com

Bill To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)
555-555-2225 (Fax)

Ship To:
Aqua Hotel and Resort
1040 Boulevard
Anytown, CA 94558
555-555-2222 (Phone)

Shipment Details:
Carrier: USPS
Method: Overnight Priority
Tracking#: w126a5s4321sas
Ship Date: 03/02/2009

INVOICE

Invoice : 00016
Date: 03/11/2009
Due Date: 04/10/2009
Balance Due \$: 1,937.01
Customer PO#:

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System Includes logo, layout for letterhead, 2nd sheet, A10 envelope, and business card	3,500.00	project	1	3,500.00	8.25%
Design Services - Additional Concepts Includes thumbnail sketches for one additional concept	125.00	each	1	125.00	8.25%

Pre-tax Total: 3,625.00

Tax: 299.06

Shipping: 12.95

Total: 3,937.01

Payments: -2,000.00

Balance (\$): 1,937.01

Notes

Thanks for the work! Art hard copies and CD mailed to printer per instructions.

Terms and Conditions

Balance Due Net 30 days.

- Considere o modelo de nota fiscal ao lado. Proponha um formato de armazenamento para representá-la.

Fonte: WorkPoint

<http://www.workingpoint.com/features/invoicing>

Modelo Lógico

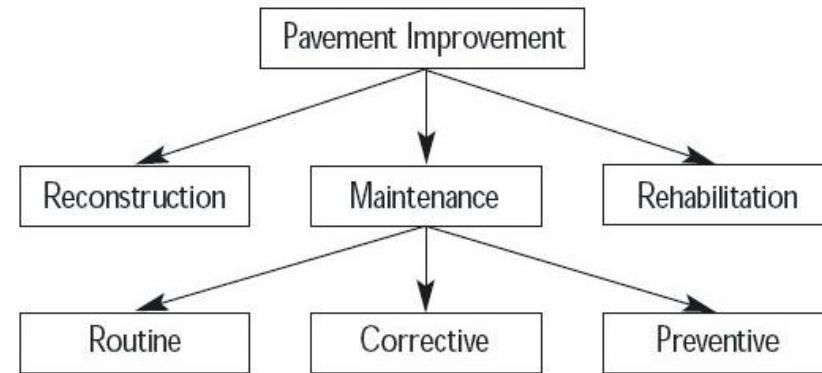
Flat File Model

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	
Record 3	SR-301	33	

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Hierarchical Model



Object-Oriented Model

Object 1: Maintenance Report Object 1 Instance

Date	01-12-01
Activity Code	24
Route No.	I-95
Daily Production	2.5
Equipment Hours	6.0
Labor Hours	6.0

Object 2: Maintenance Activity

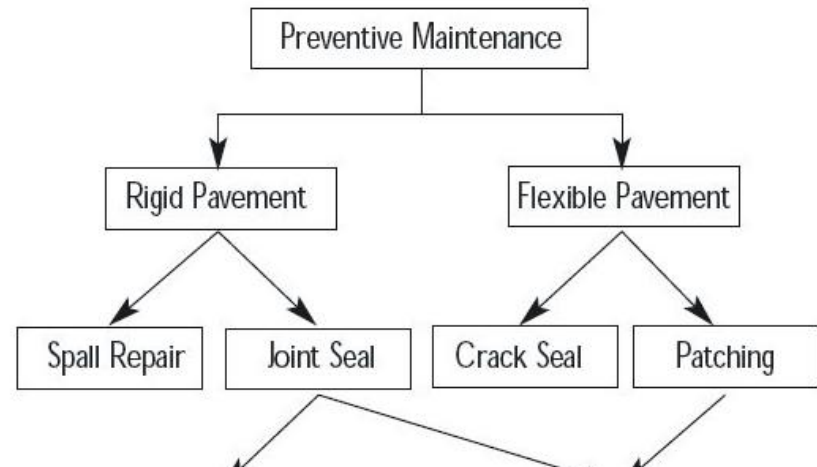
Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

By Marcel Douwe Dekker
https://en.wikipedia.org/wiki/File:Database_models.jpg

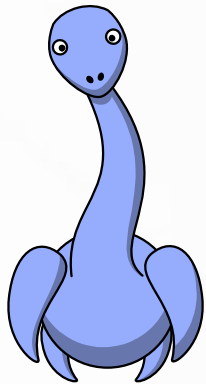
Network Model



Visão de Banco de Dados Modelo

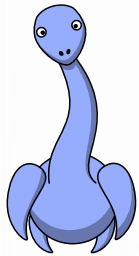
- Hierárquico
- Baseado em documentos
- Semi-estruturado

Describing Prehistoric Animals

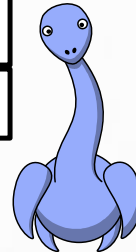


MNHN A. C. 8592		
Is a	Plesiosaurus dolichodeirus	
Origin	Lyme Regis	England
Recognized	1824	
Size	5	

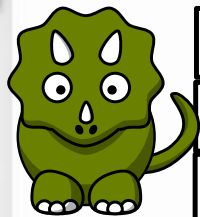
Describing Prehistoric Animals



SIPB R 90		
Is a	Plesiosaurus dolichodeirus	
Origin	Lyme Regis	England
Recognized	1830	
Size	5	



STC223		
Is a	Plesiosaurus gurgitis	
Origin	St. Croix	Switzerland
Recognized	1964	
Size	3.5	



MNHN 1912.20		
Is a	Triceratops horridus	
Origin	Lance Creek	EUA
Recognized	1889	
Size	9	



Sue

FMNH PR2081		
Is a	Tyrannosaurus rex	
Origin	Hell Creek	EUA
Recognized	1990	
Size	12.3	

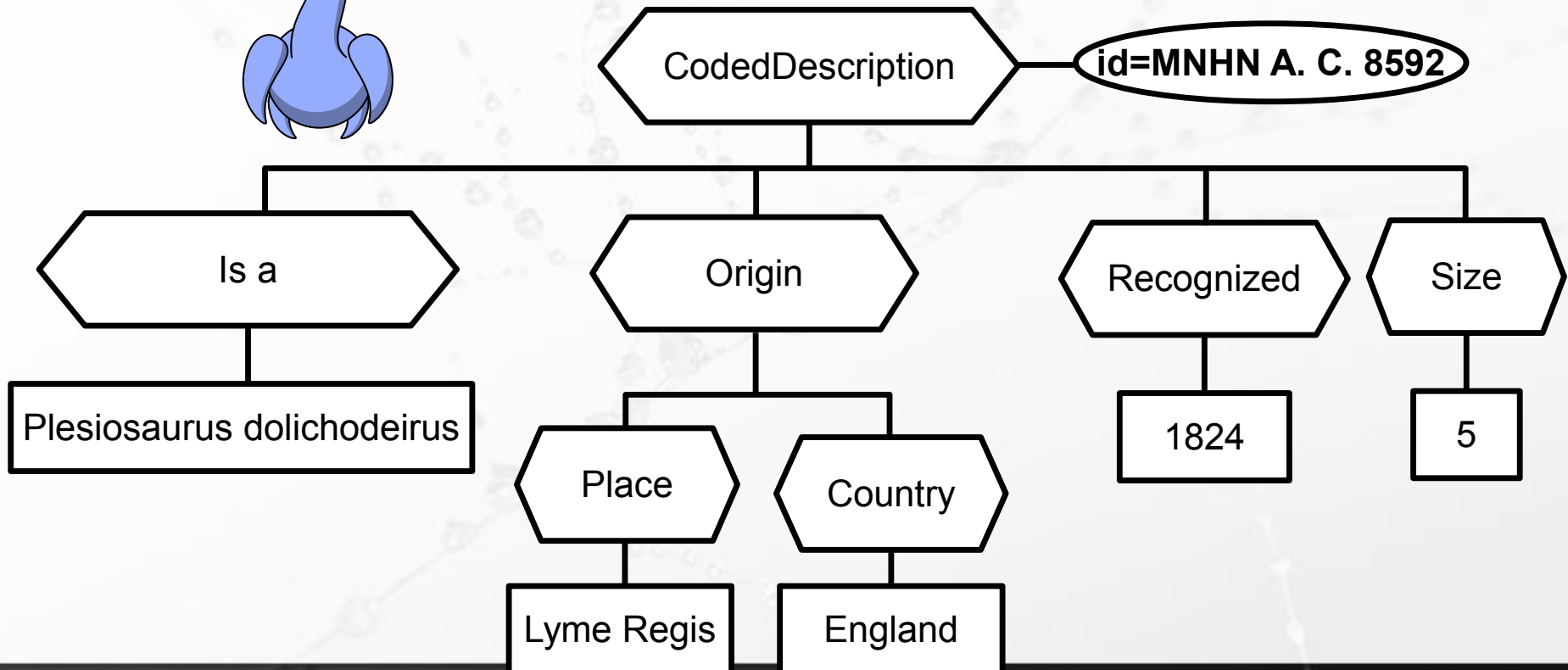
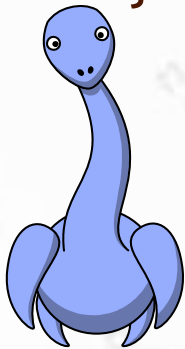
Table

Id	Is a	Origin Place	Origin Country	Recognized	Size
MNHN A. C. 8592	Plesiosaurus dolichodeirus	Lyme Regis	England	1824	5
SIPB R 90	Plesiosaurus dolichodeirus	Lyme Regis	England	1830	5
STC223	Plesiosaurus gurgitis	St. Croix	Switzerland	1964	3.5
MNHN 1912.20	Triceratops horridus	Lance Creek	EUA	1889	9
FMNH PR2081	Tyrannosaurus rex	Hell Creek	EUA	1990	12.3

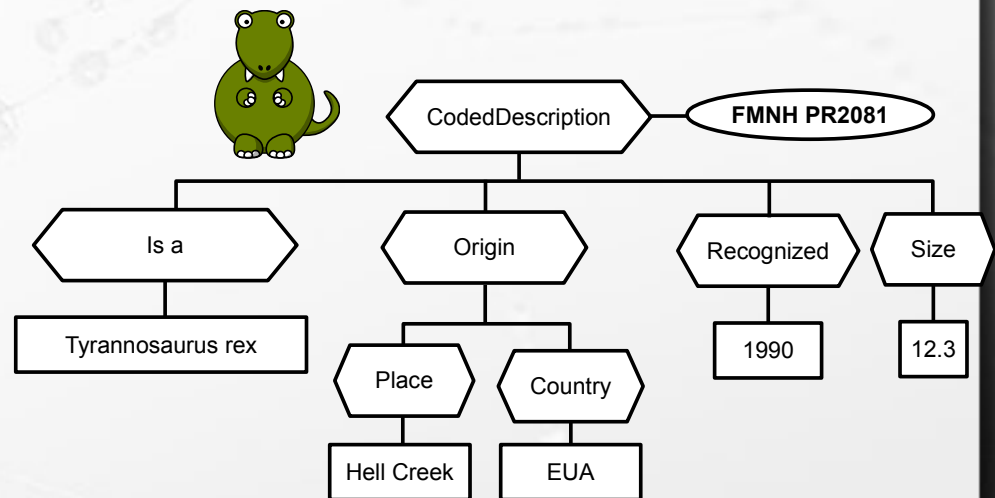
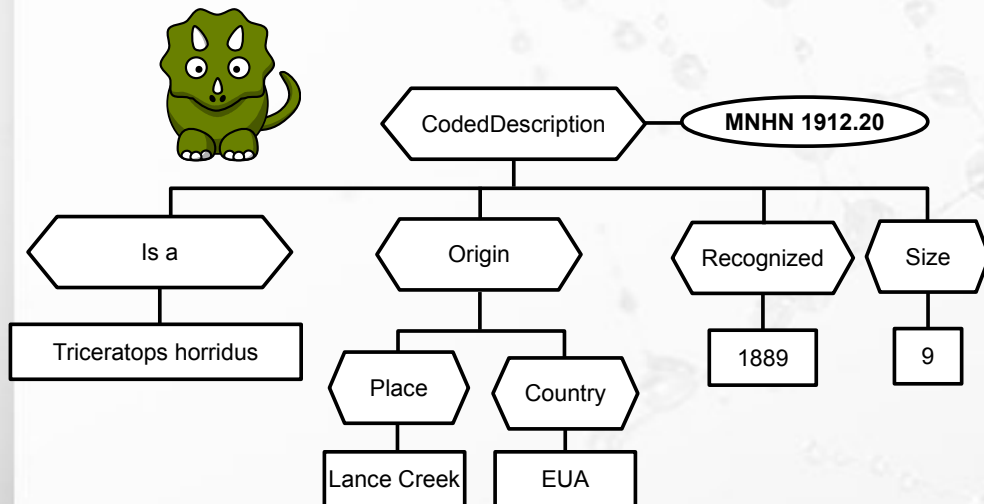
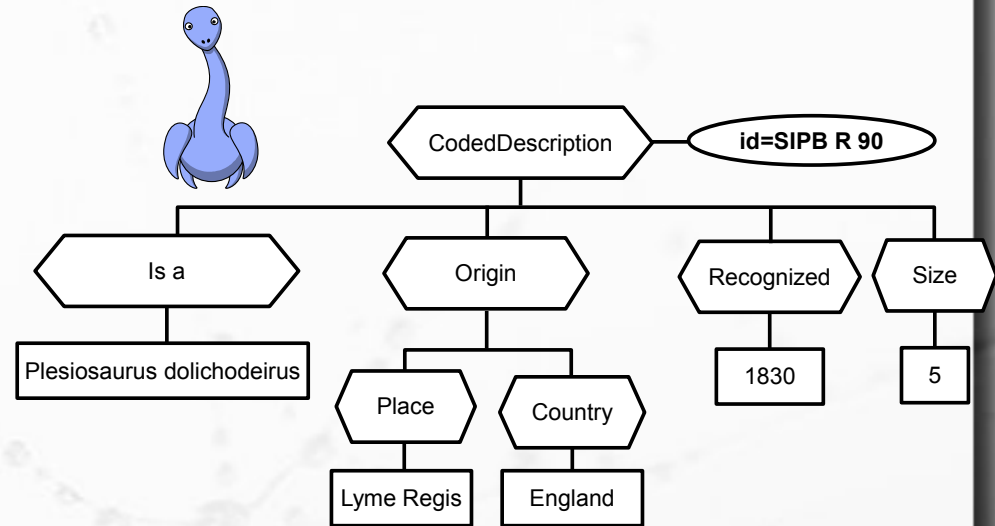
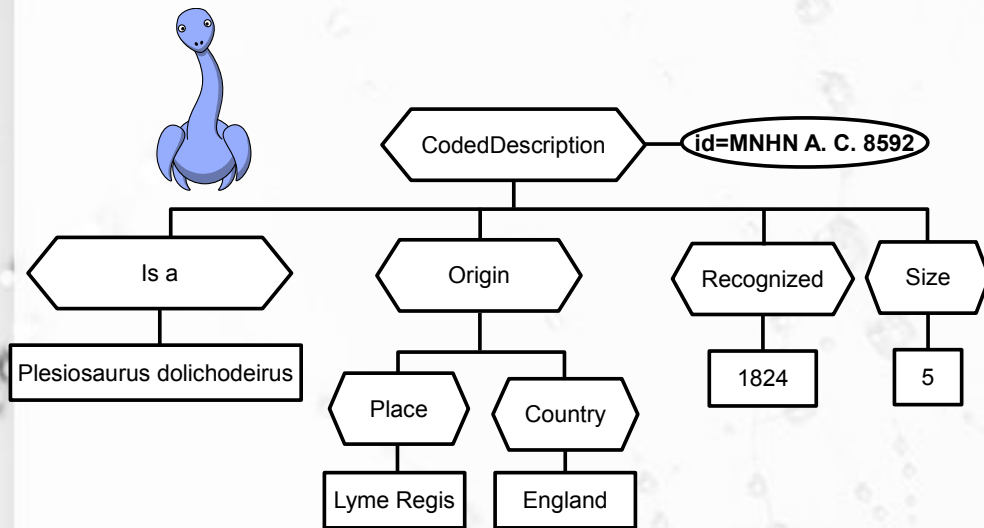


Modelo Hierárquico (Documentos/XML)

- Modelo autocontido adequado para a distribuição



Modelo Hierárquico (Documentos/XML)



XML

Modelo Lógico Semi-estruturado

Estruturado x Semi-estruturado

- Estruturado

- formato estrito
 - e.g., modelo relacional
- cada registro segue o mesmo formato

(Elmasri, 2010)

- Semi-estruturado

- itens de dados podem ter estruturas variadas
- grupos de itens compartilham estruturas

Processando Dados

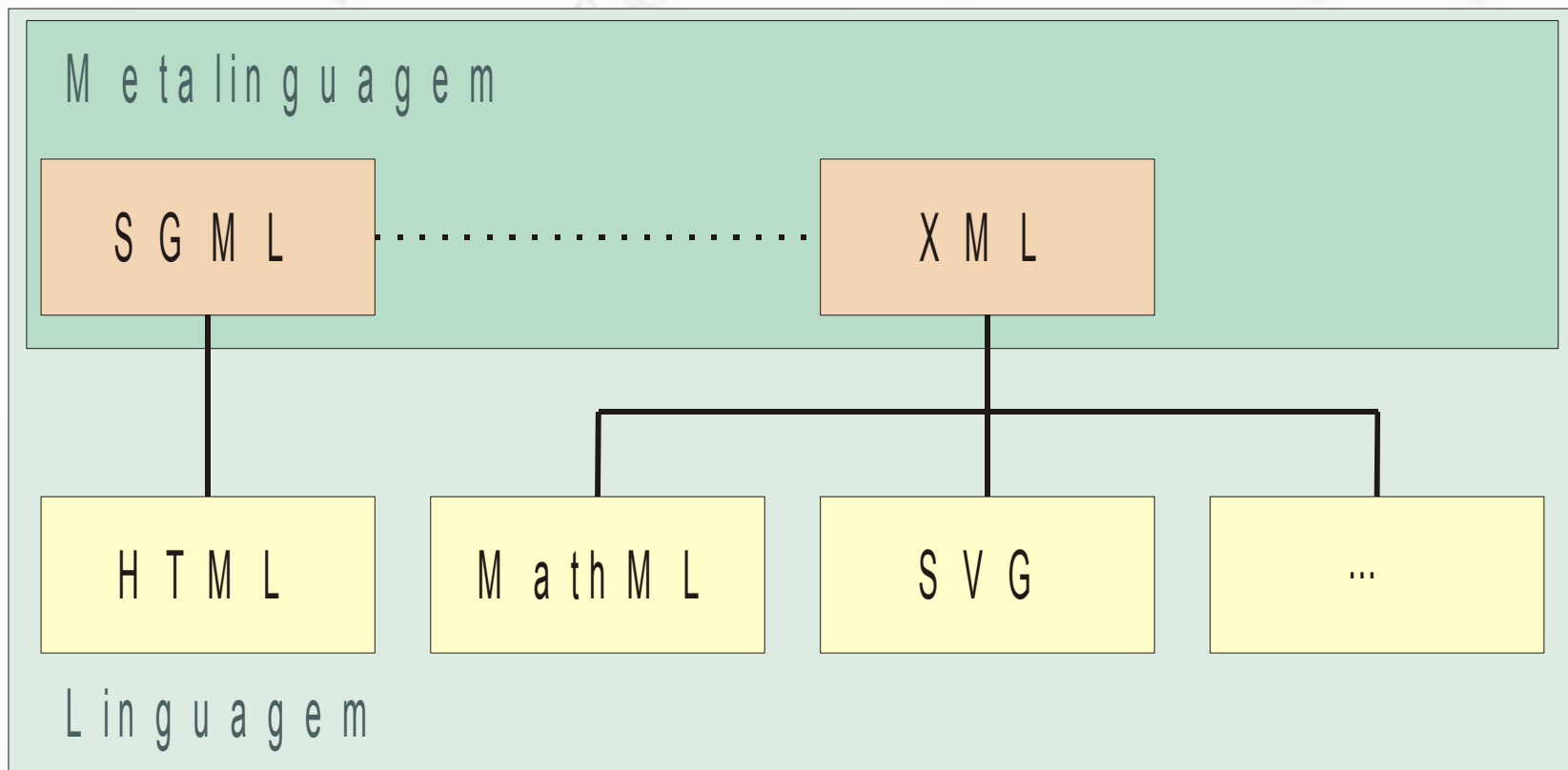
- Quanto mais “organizados” (estruturados) os dados, mas simples é o processamento
- Exemplo de dados **estruturados**: listas, tabelas, matrizes
- Exemplo de dados **não-estruturados**: texto, imagens, sons
- Exemplo de dados **semiestruturados**: árvores, grafos

XML

Metalinguagem e Linguagem para a Web

Metalinguagem

- Tal como SGML, XML é uma metalinguagem.
- HTML ao contrário, foi escrita em SGML.



SVG - Scalable Vector Graphics

- Formato XML para a representação de imagens vetoriais (Dahlström et al., 2011)
- Suporte nativo dos navegadores

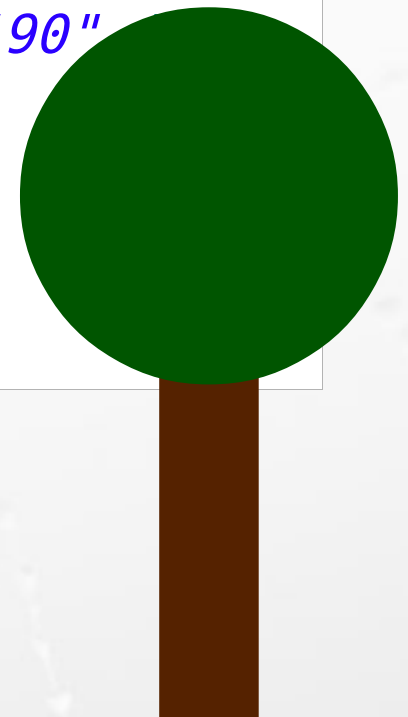
Árvore SVG

```
<svg id="desenho" version="1.1"
  xmlns="http://www.w3.org/2000/svg"
  width="205" height="370">

  <rect style="fill:#552200"
    x="77" y="179" width="50" height="190"

  <circle style="fill:#005500"
    cx="102" cy="106" r="95" />

</svg>
```



Linguagem de Marcação

- Utiliza marcadores para agregar informações adicionais a documentos.
- Tomemos como exemplo a seguinte frase:
Horácio escreveu o livro Vida dos Dinossauros.
- Desejamos agregar informações que identifiquem quem é o autor e qual a ação realizada.

Linguagem de Marcação

- Os marcadores se diferenciam do conteúdo pelos símbolos “<” e “>” (seguem o mesmo princípio de HTML):

```
<autor>Horácio</autor> <ação>escreveu o livro Vida dos Dinossauros</ação>
```

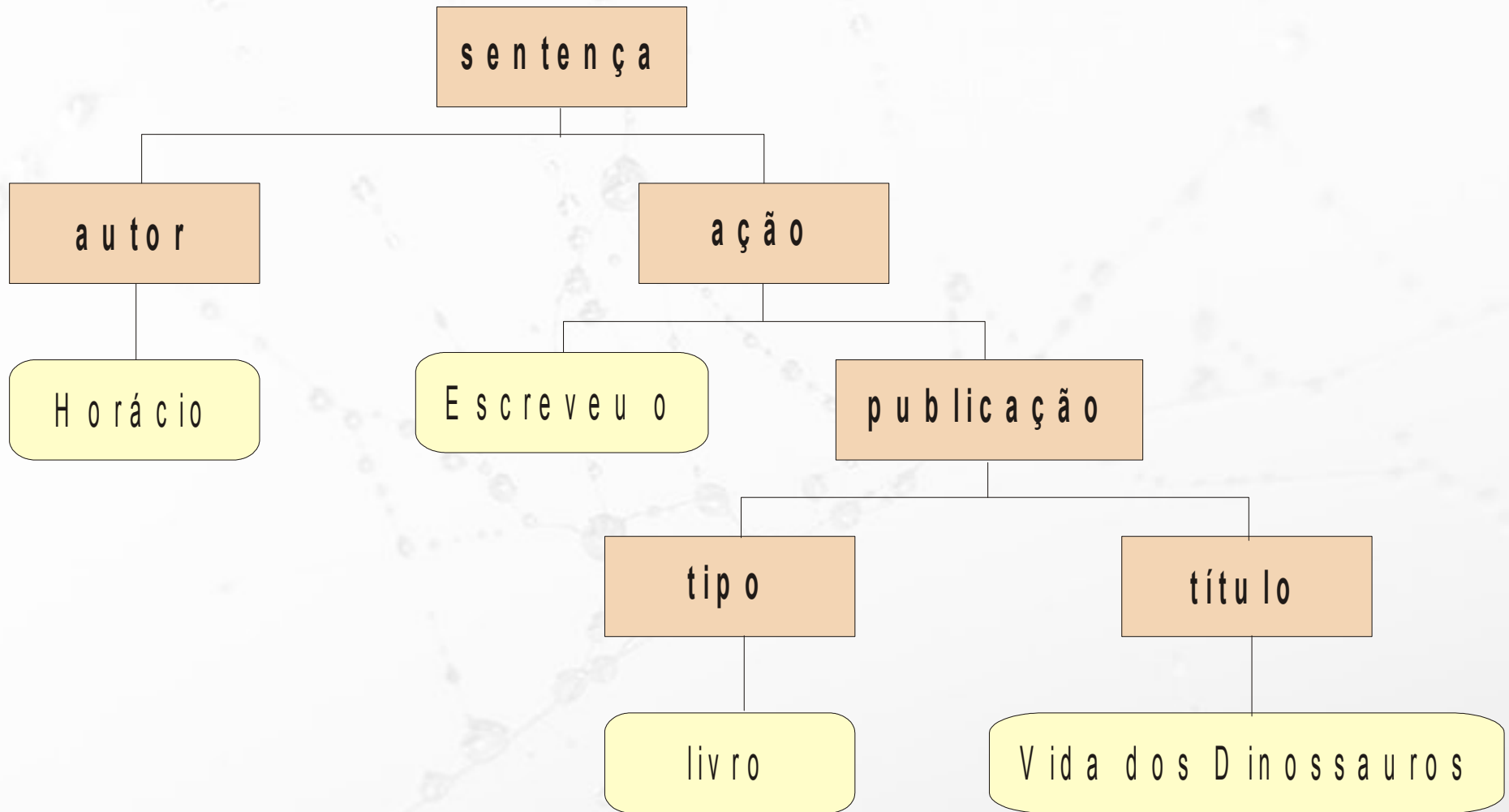
- Os marcadores delimitam unidades estruturais denominadas **elementos**.

Estrutura Hierárquica

- Marcações podem ser agrupadas hierarquicamente.
- Cada documento tem uma única raiz.
- A interpretação de cada marcador está subordinada a seu contexto.

```
<sentença>  
  <autor>Horácio</autor>  
  <ação>escreveu o  
    <publicação>  
      <tipo>livro</tipo>  
      <título>Vida dos Dinossauros</título>  
    </publicação>  
  </ação>  
</sentença>
```

Modelo de Dados XML



Elemento Vazio

- Não contém outros elementos ou texto aninhados

```
<esgotado/>
```

Atributos

- Elementos podem conter atributos
- Não há dois atributos de mesmo nome no mesmo elemento

```
<autor cpf="487.526.548-74" nascimento="12/5/1960"> Horácio </autor>
```

Links para Elementos

- # no final da URI mais id do link
- elemento no documento cujo do valor do id seja igual ao que sucede o #

<http://www.dominio.org/documento.xml#bibliografia>

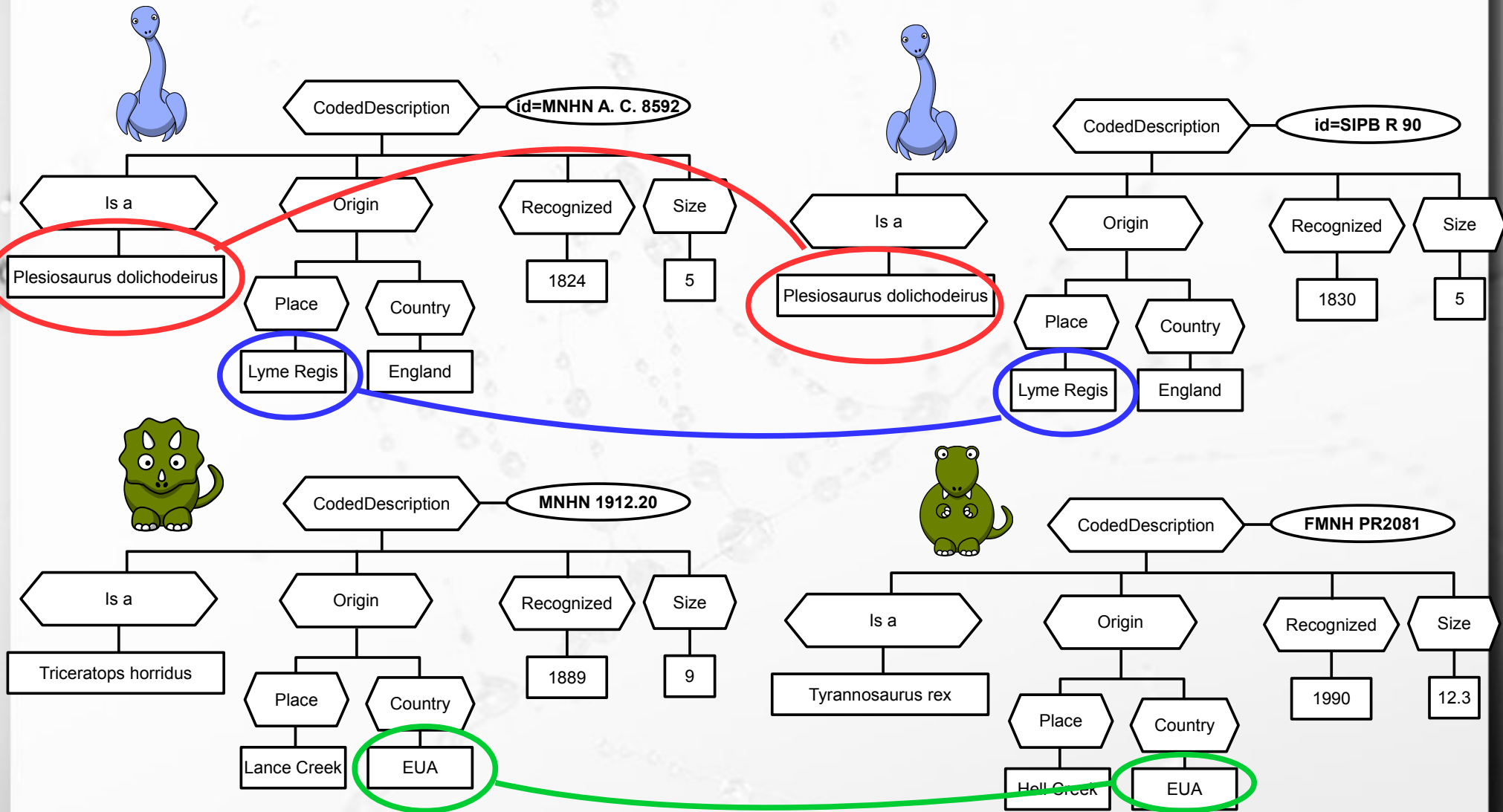
```
<__>  
  <__>  
    <__>.....</__>  
    <__ id="elem">  
      ...  
    </__>  
  <__/>  
</__>  
<__> ..... </__>  
</__>
```

<http://www.paleo.org/doc.xml#elem>

Possibilidades da URI

- Fazer referência a um elemento dentro de um documento.

Relações em XML



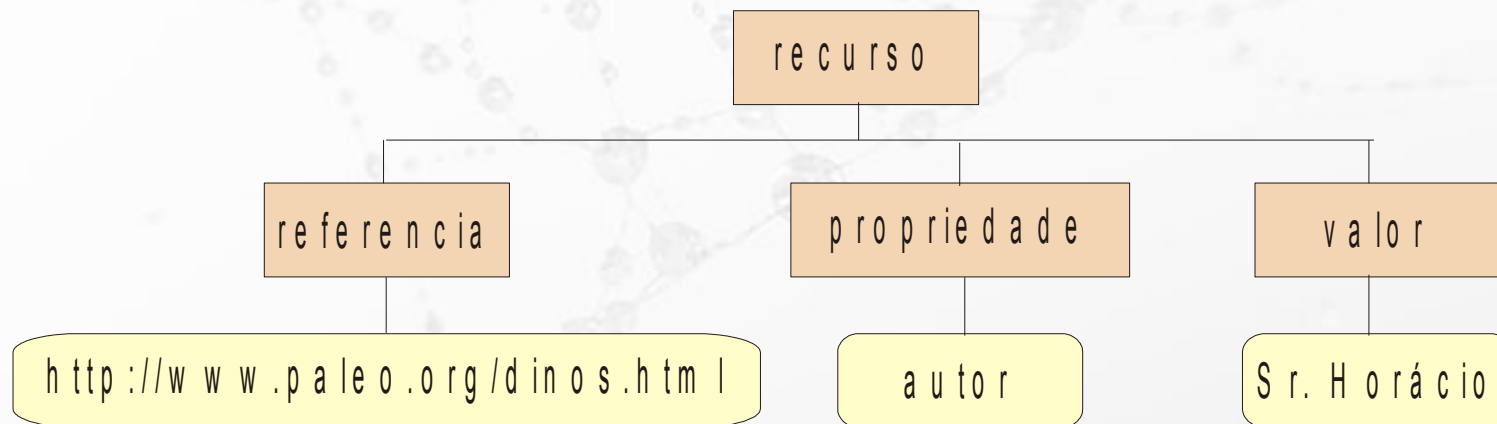
Exercício 1

Representar:

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

Possível Representação 1

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



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

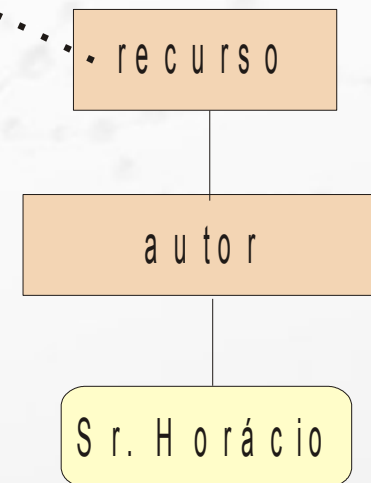
propriedade

Sr. Horácio

Possível Representação 3

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

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



Possível Representação 4

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

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

rotulo = autor

recurso

Questão 1

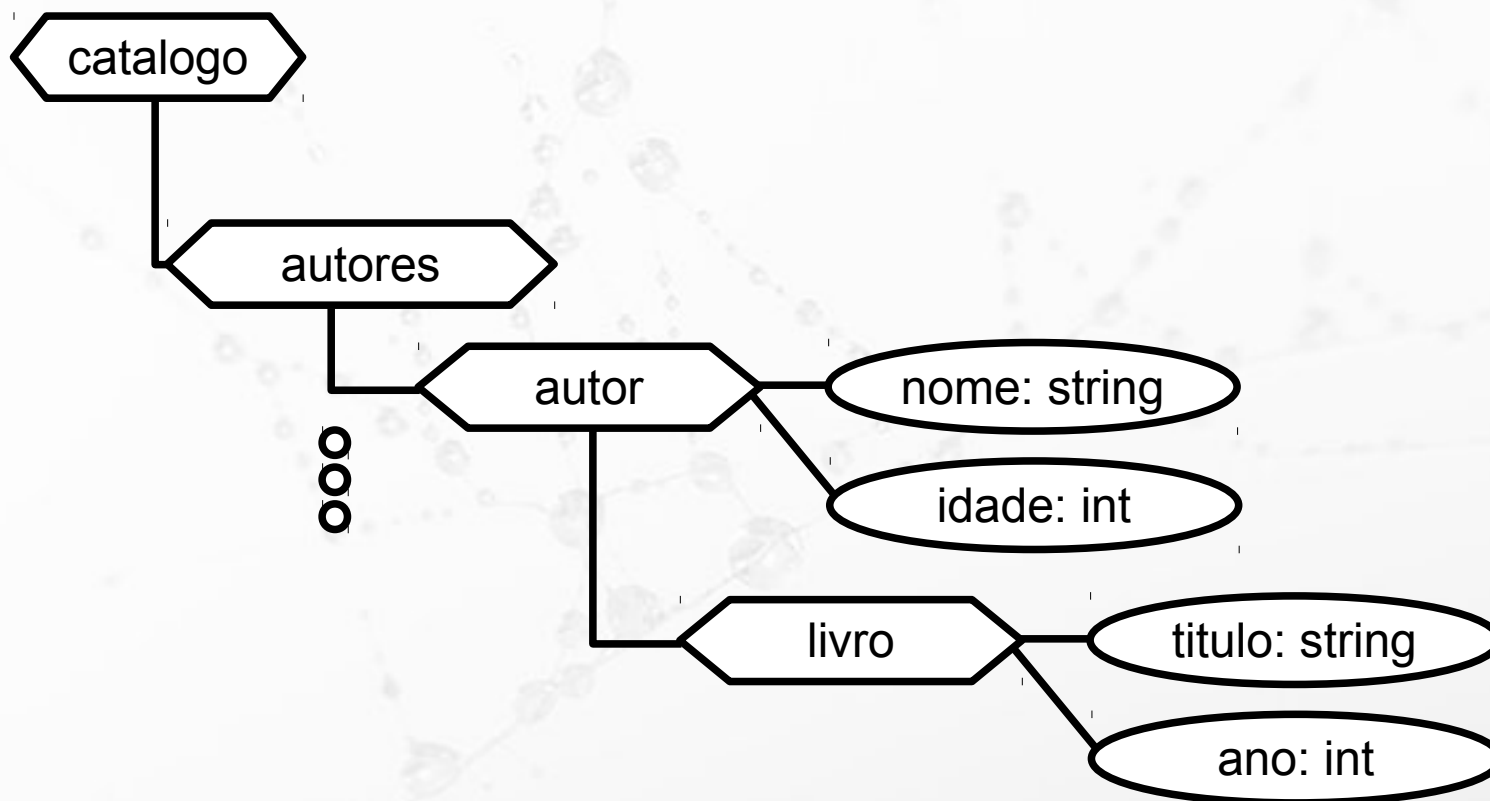
Qual a melhor forma de realizar a representação?

Modelos de Relações



Embedded

Autor/Livro Embedded



Livro Embedded em Autor

```
{  
  "nome": "Doriana",  
  "livro": {  
    "titulo": "Horacio o dino",  
    "ano": 2015  
  }  
}
```

563c871c45400d49a72753c4

nome: "Doriana"

livro

titulo: "Horacio o dino"
ano: 2015

Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Doriana">
```

```
    <livro titulo="Horacio o dino"  
ano="2015" />
```

```
  </autor>
```

```
  ...
```

```
</autores>
```

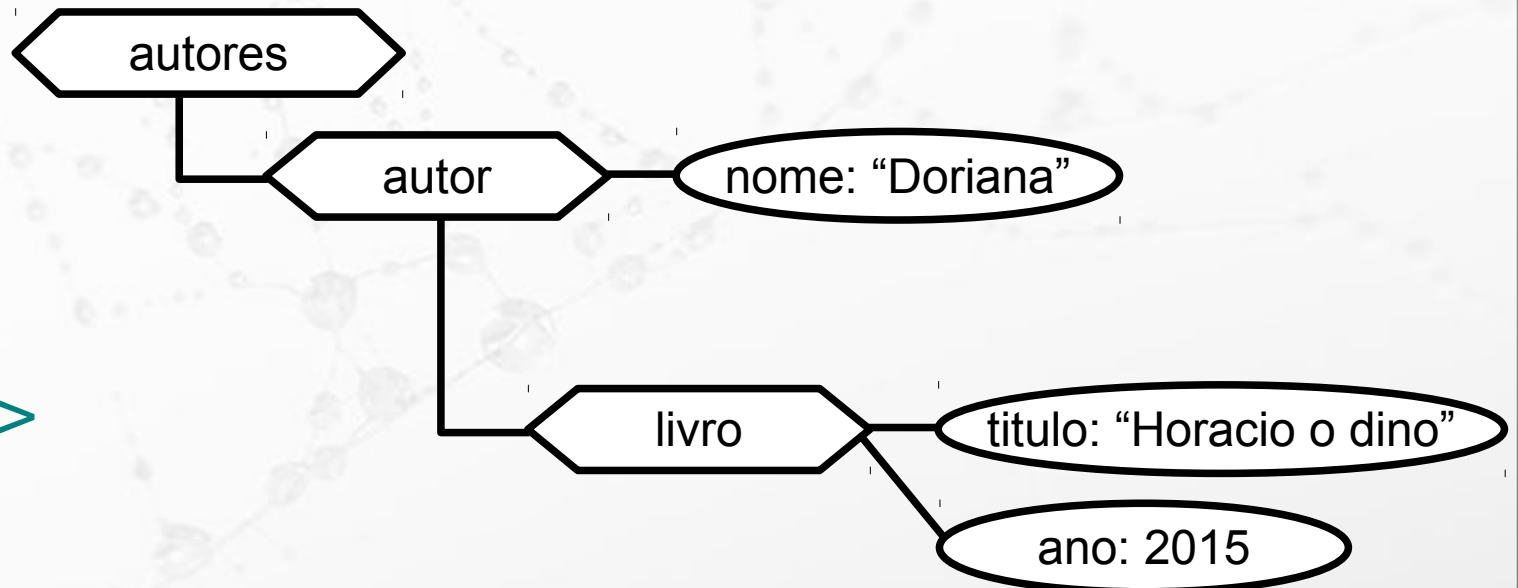
```
...
```

```
</catalogo>
```

563c871c45400d49a72753c4
nome: "Doriana"

↓ livro

titulo: "Horacio o dino" ano: 2015



Livro Embedded em Autor

```
{ "nome": "Asdrubal",  
  "idade": 25,  
  "livro": {  
    "titulo": "Vida sentimental dos  
dinossauros",  
    "ano": 2013  
  }  
}
```

563c873745400d49a72753c5

nome: "Asdrubal"
idade: 25

livro

titulo: "Vida sentimental dos dinossauros"
ano: 2013

Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Asdrubal" idade="25"
```

```
    <livro titulo="Vida sentimental dos dinossauros"  
ano="2013" />
```

```
  </autor>
```

```
...
```

```
</autores>
```

```
...
```

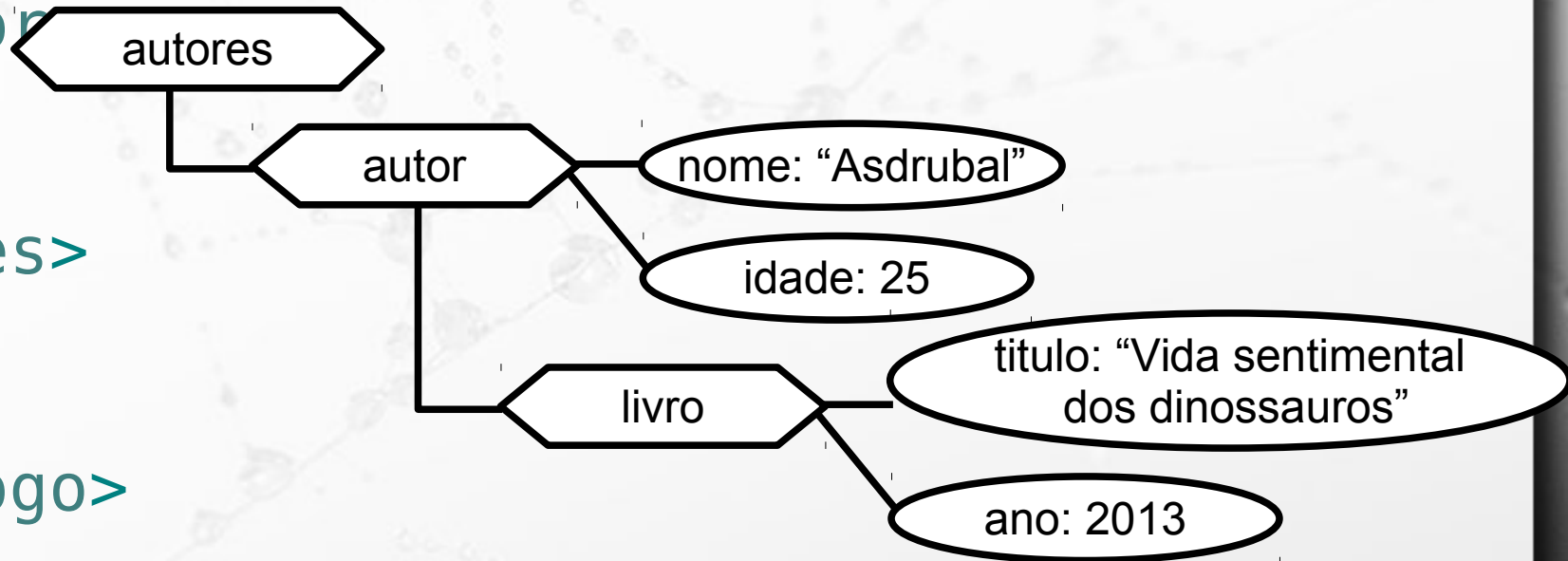
```
</catalogo>
```

563c873745400d49a72753c5

nome: "Asdrubal"
idade: 25

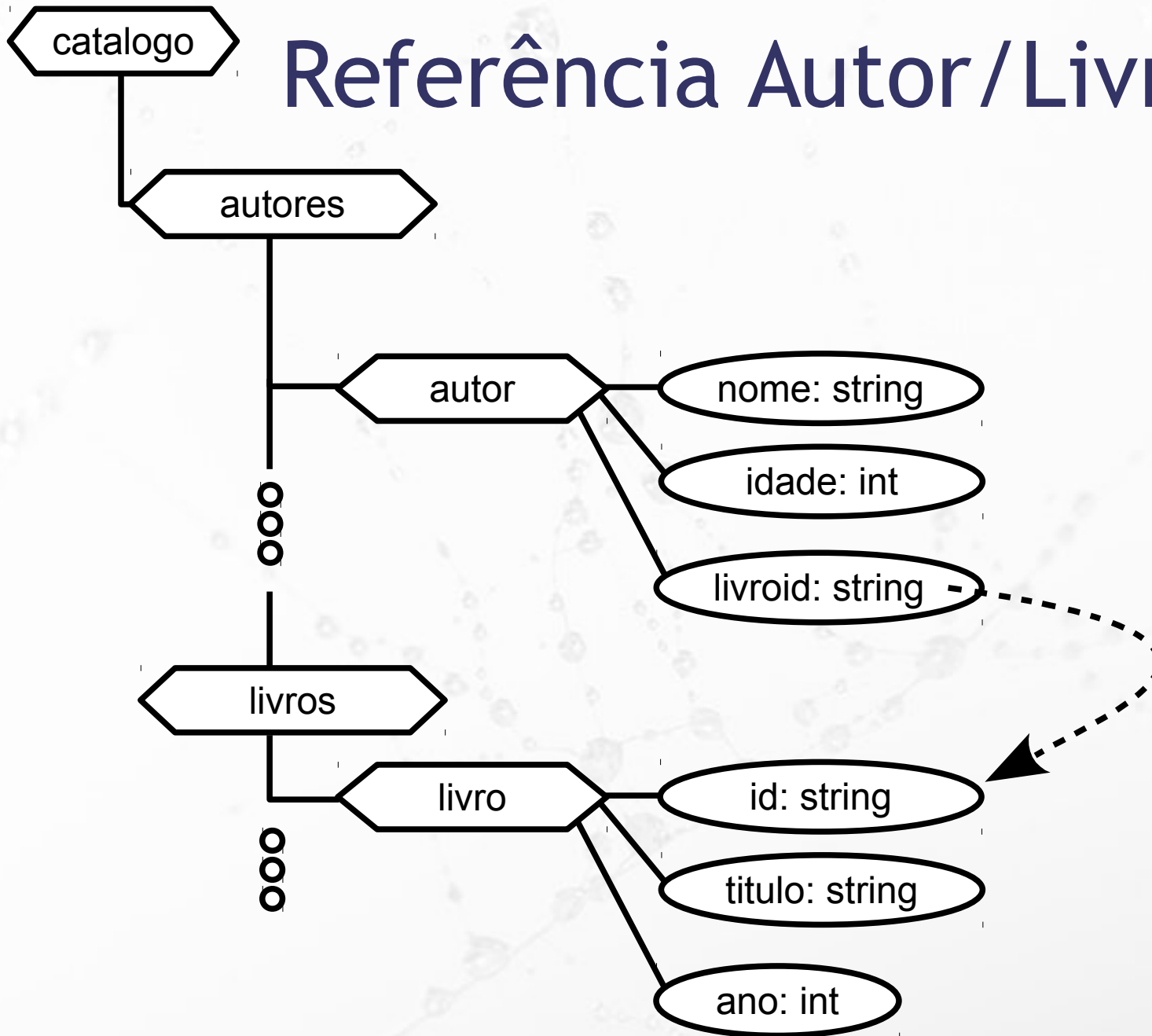
livro

titulo: "Vida sentimental dos dinossauros"
ano: 2013



Referências Cruzadas

Referência Autor/Livro



Insert com “_id” manual bloco livros

```
{  
  "_id": "livro-como-entender",  
  "titulo": "Como entender seu dino",  
  "ano": 2015  
}
```

livro-como-entender
titulo: "Como entender seu dino" ano: 2015

Referência Autor/Livro

```
<catalogo>
```

```
<livros>
```

```
<livro id="livro-como-entender"  
      titulo="Como entender seu dino"  
      ano="2015" />
```

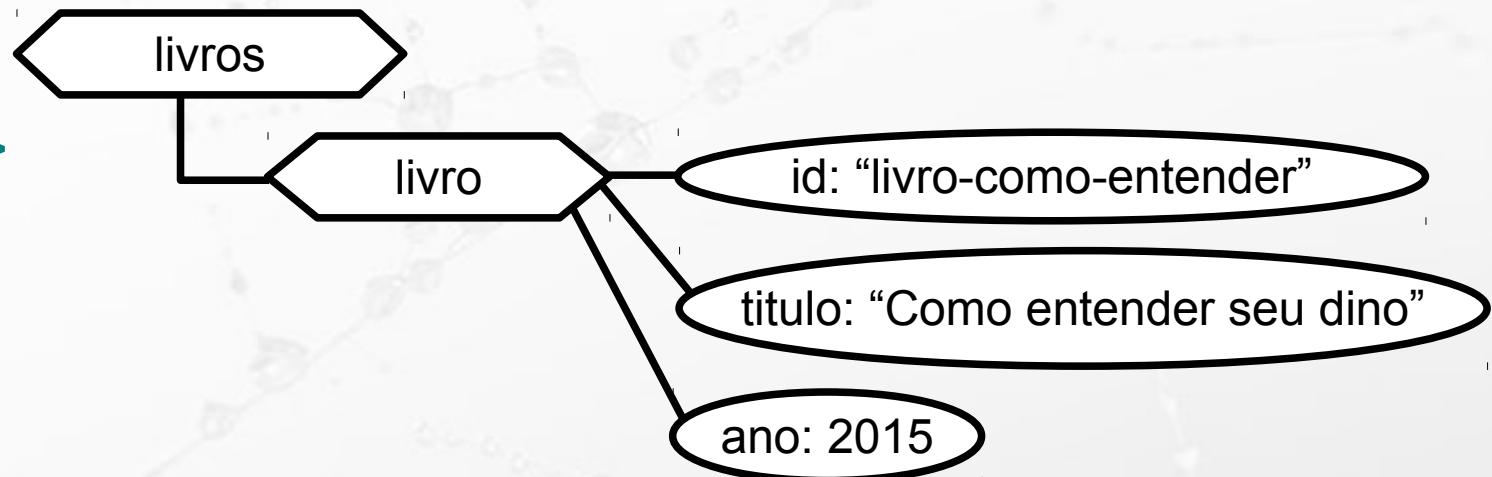
```
...
```

```
</livros>
```

```
...
```

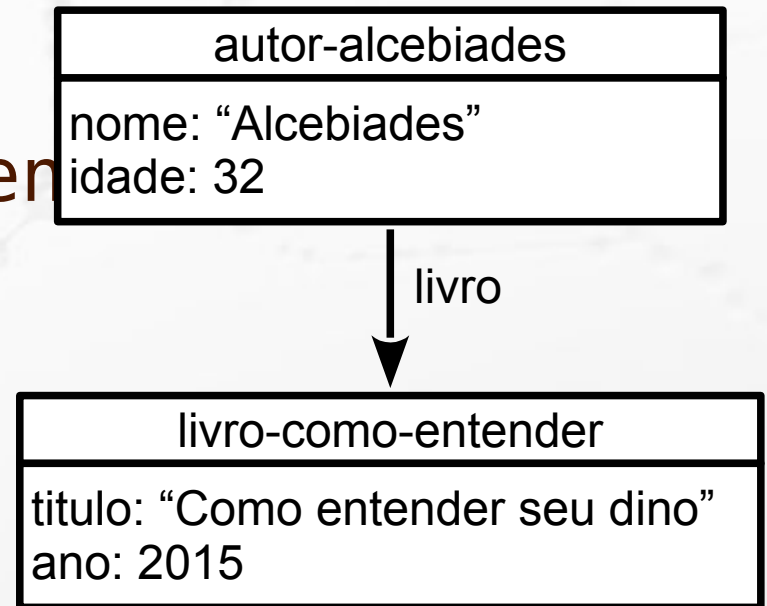
```
</livro>
```

livro-como-entender
titulo: "Como entender seu dino" ano: 2015

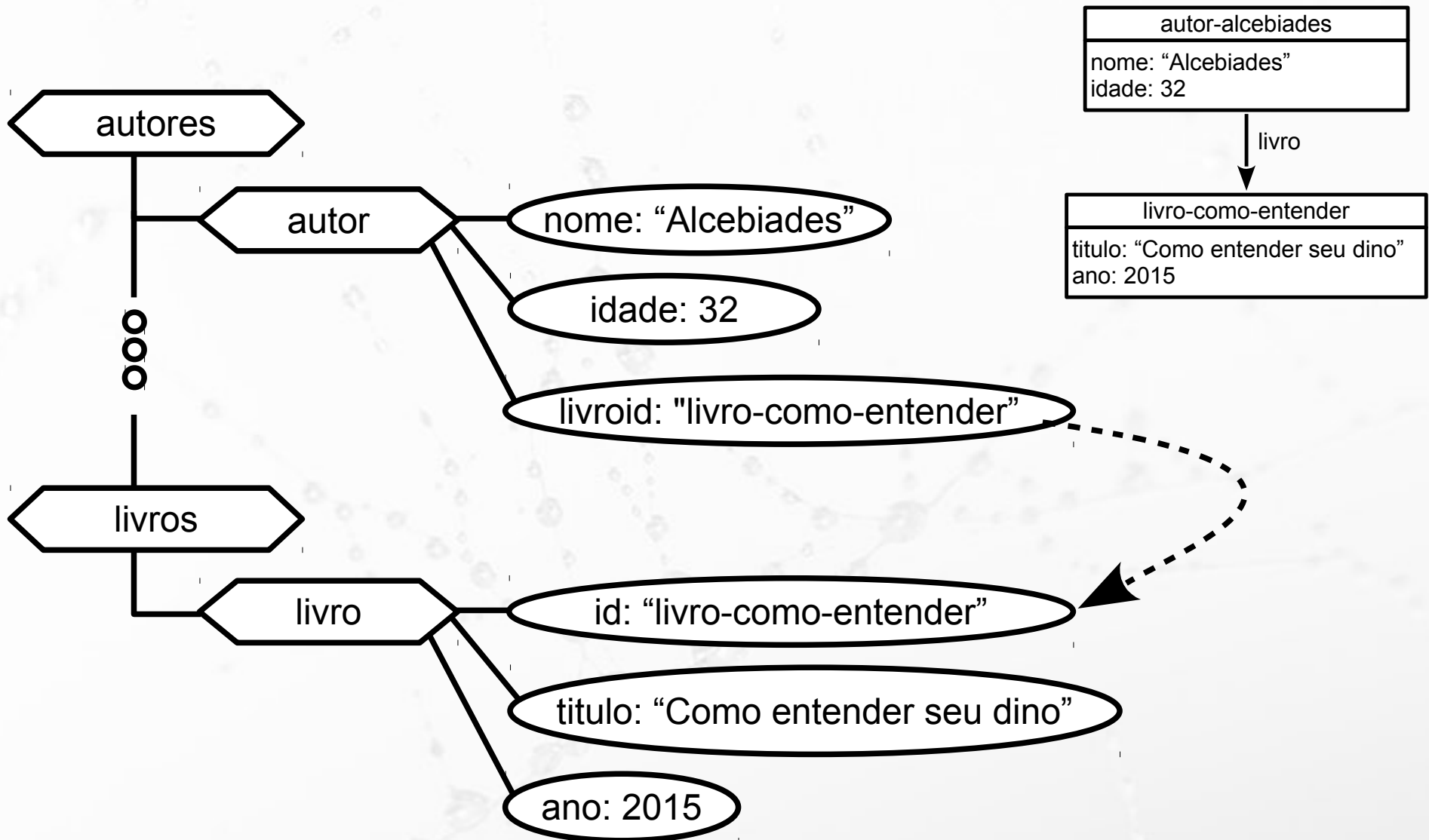


Insert referência a livro bloco autores

```
{  
  "_id": "autor-alcebiades",  
  "nome": "Alcebiades",  
  "idade": 32,  
  "livro": "livro-como-entender"  
}
```

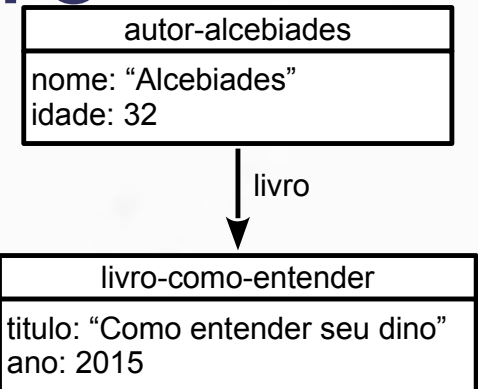


Referência Autor/Livro

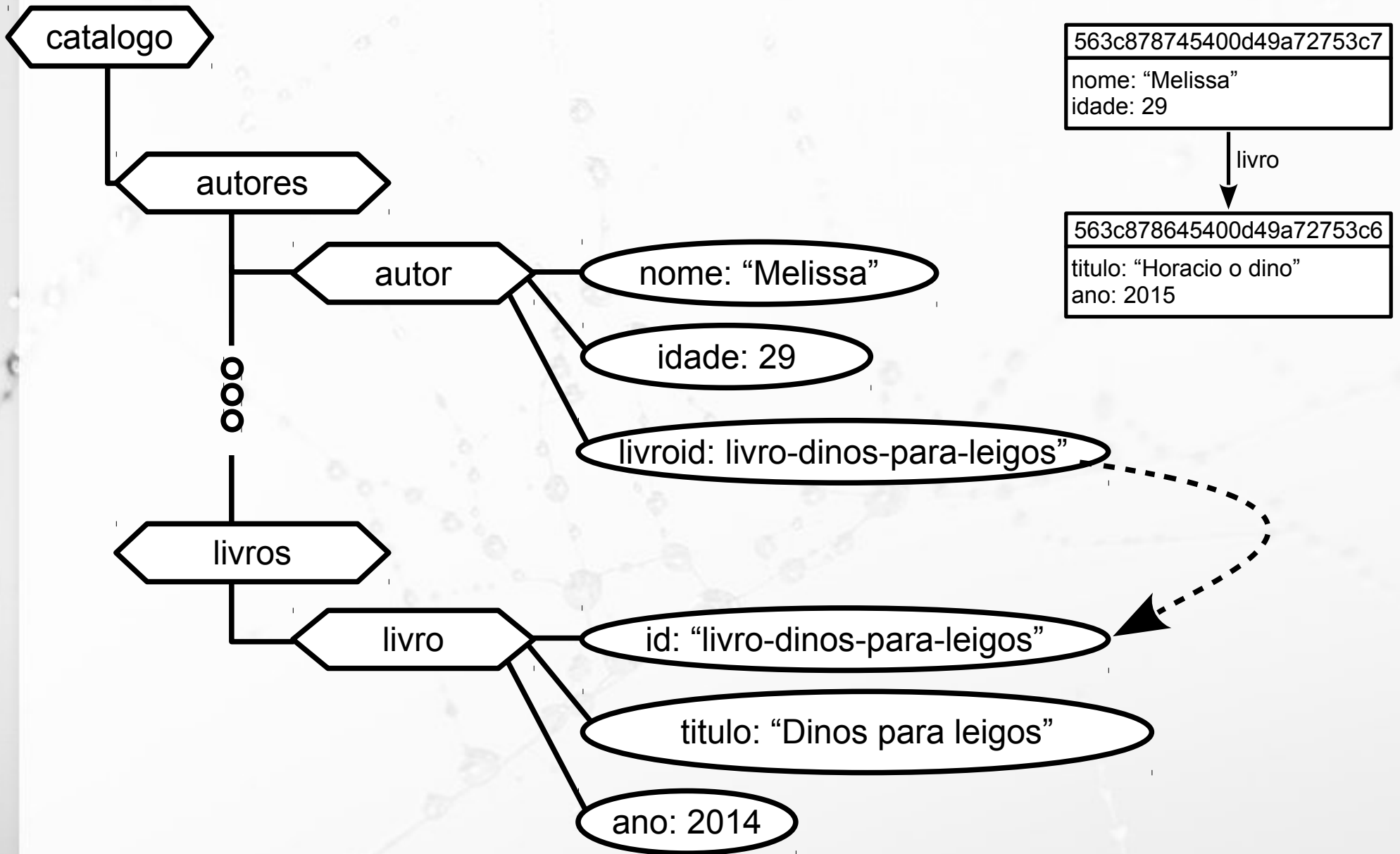


Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Alcebiades"
        idade="32"
        livroid="livro-como-entender" />
  ...
</autores>
...
<livros>
  <livro id="livro-como-entender"
        titulo="Como entender seu dino"
        ano="2015" />
  ...
</livros>
</catalogo>
```

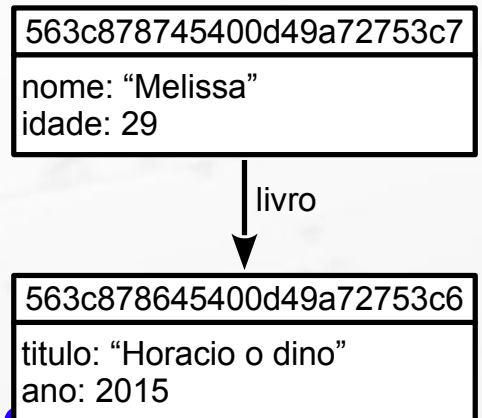


Referência Autor/Livro



Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Melissa"
        idade="29"
        livroid="livro-dinos-para-
leigos" />
  ...
</autores>
  ...
<livros>
  <livro id="livro-dinos-para-leigos"
        titulo="Dinos para leigo"
        ano="2014" />
  ...
</livros>
</catalogo>
```



Exercício 2

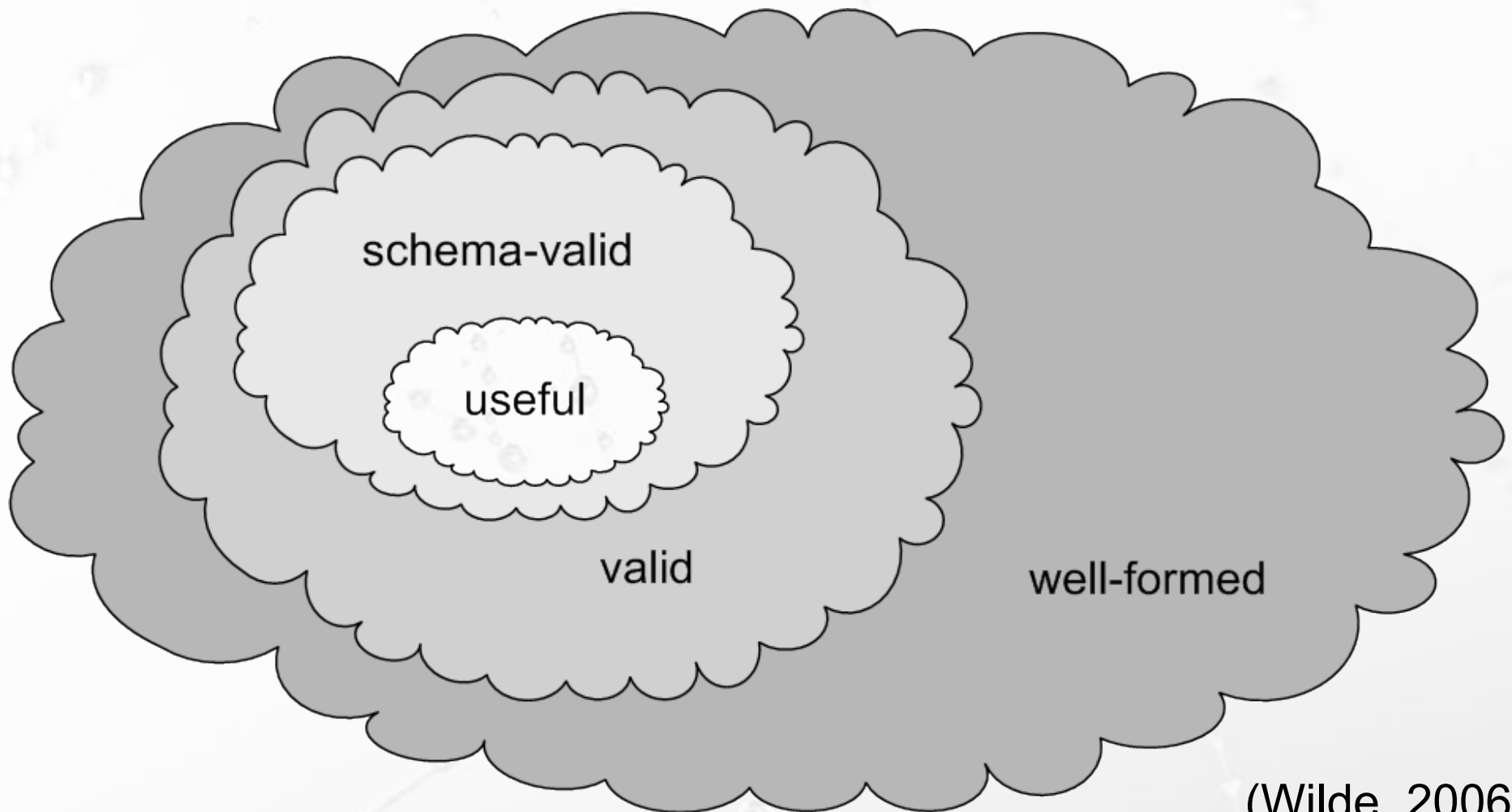
Representar:

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

Validação de Documentos

- Documento bem formado:
 - atende às regras de construção XML
- Documento válido:
 - bem formado
 - atende a um esquema
 - DTD
 - XML Schema

Validação de Documentos



(Wilde, 2006)

DTD

- O documento XML pode se basear em uma gramática definida através de uma DTD (*Document Type Definition*).

```
<!ELEMENT documento (topico+)>  
<!ELEMENT topico (titulo, subtopico*)>  
<!ELEMENT titulo (#PCDATA)>  
<!ELEMENT subtopico (titulo, #PCDATA)>
```

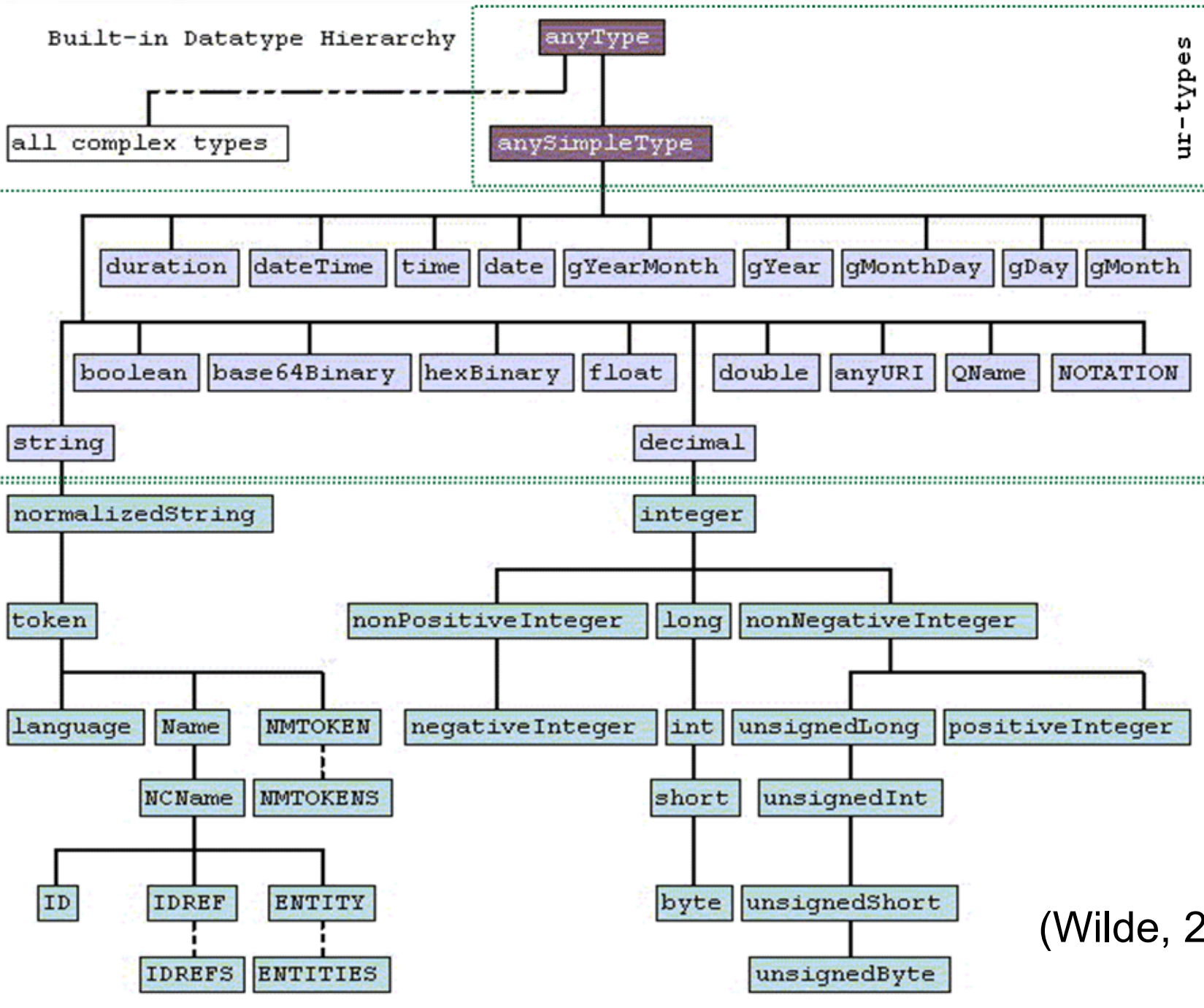
XML Schema

- Padrão para definição de esquemas XML
- Mais poderoso

XML Schema

Tipos de Dados

Built-in Datatype Hierarchy



ur-types

primitive types

derived types

— derived by restriction

- - - derived by list

- · - · - derived by extension or

ur types

built-in primitive types

built-in derived types

(Wilde, 2006)

Invoice Schema



Green Leaf Design
 111 Main Street
 Pleasanton, CA 99999
 555.555.4444
info@greenleafdesign.com

INVOICE

Invoice : 00016
 Date: 03/11/2009
 Due Date: 04/10/2009
 Balance Due \$: 1,937.01
 Customer PO#:

Invoice number: 00016
 Issue Date: 03/11/2009

Bill To:
 Aqua Hotel and Resort
 1040 Boulevard
 Anytown, CA 94558
 555-555-2222 (Phone)
 555-555-2225 (Fax)

Ship To:
 Aqua Hotel and Resort
 1040 Boulevard
 Anytown, CA 94558
 555-555-2222 (Phone)

Shipment Details:
 Carrier: USPS
 Method: Overnight Priority
 Tracking#: w126a5s4321sas
 Ship Date: 03/02/2009

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System Includes logo, layout for letterhead, 2nd sheet, A10 envelope, and business card	3,500.00	project	1	3,500.00	8.25%
Design Services - Additional Concepts includes thumbnail sketches for one additional concept	125.00	each	1	125.00	8.25%

Pre-tax Total: 3,625.00
 Tax: 299.06
 Shipping: 12.95

Total: 3,937.01
 Payments: -2,000.00

Balance (\$): 1,937.01

Notes
 Thanks for the work! Art hard copies and CD mailed to printer per instructions.

Terms and Conditions
 Balance Due Net 30 days.

Simple and Complex Types

- Simple Types
 - **Não pode conter elementos e atributos agregados**
- Complex Types
 - **Pode conter elementos e atributos agregados**
- “In XML Schema, there is a basic difference between complex types which allow elements in their content and may carry attributes, and simple types which cannot have element content and cannot carry attributes.”

Associando Elementos e Atributos a Tipos

```
<attribute name="invoiceNumber" type="ID">  
</attribute>
```

```
<element name="issueDate" type="date">  
</element>
```


Tipo Simples

- Derivado de tipos simples existentes
- String com restrições de tamanho:

```
<simpleType name="InvoiceNumberType">  
  <restriction base="ID">  
    <length value="7"></length>  
  </restriction>  
</simpleType>
```


Tipo Simples

- Derivado de tipos simples existentes
- String com restrições de tamanho:

```
<simpleType name="DescriptionType">  
  <restriction base="string">  
    <minLength value="10"></minLength>  
    <maxLength value="200"></maxLength>  
  </restriction>  
</simpleType>
```

Invoice Schema



Green Leaf Design
 111 Main Street
 Pleasanton, CA 99999
 555.555.4444
info@greenleafdesign.com

INVOICE

Invoice : 00016
 Date: 03/11/2009
 Due Date: 04/10/2009
 Balance Due \$: 1,937.01
 Customer PO#:

Invoice number: 00016
 Issue Date: 03/11/2009

Bill To:
 Aqua Hotel and Resort
 1040 Boulevard
 Anytown, CA 94558
 555-555-2222 (Phone)
 555-555-2225 (Fax)

Ship To:
 Aqua Hotel and Resort
 1040 Boulevard
 Anytown, CA 94558
 555-555-2222 (Phone)

Shipment Details:
 Carrier: USPS
 Method: Overnight Priority
 Tracking#: w126a5s4321sas
 Ship Date: 03/02/2009

Item	Price (\$)	Unit	Qty	Total (\$)	Tax
Design Services - Business System Includes logo, layout for letterhead, 2nd sheet, A10 envelope, and business card	3,500.00	project	1	3,500.00	8.25%
Design Services - Additional Concepts includes thumbnail sketches for one additional concept	125.00	each	1	125.00	8.25%

Pre-tax Total: 3,625.00

Tax: 299.06

Shipping: 12.95

Total: 3,937.01

Payments: -2,000.00

Balance (\$): 1,937.01

Notes

Thanks for the work! Art hard copies and CD mailed to printer per instructions.

Terms and Conditions

Balance Due Net 30 days.

Tipo Composto

```
<complexType name="InvoiceType1">  
  <sequence>  
    <element name="issueDate" type="date"></element>  
  </sequence>  
  <attribute name="invoiceNumber" type="ID"></attribute>  
</complexType>
```

Tipo Composto

```
<complexType name="InvoiceType">  
  <attribute name="invoiceNumber" type="tns:InvoiceNumberType"  
use="required"></attribute>  
  <sequence>  
    <element name="issueDate" type="date"></element>  
    <element ref="tns:client"></element>  
  </sequence>  
</complexType>
```

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>
```

Query

- **XPath**

- Especifica expressões na forma de caminhos que atendem padrões para alcançar nós específicos (elementos ou atributos)

- **XQuery**

- Queries para XML (usam XPath)

XPath

/	no começo → nó raiz entre nós → separador hierárquico
//	no começo → nó que começa em qualquer nível hierárquico entre nós → nó em qualquer nível hierarquicamente abaixo
@atr	indica um atributo
*	qualquer elemento
@*	qualquer atributo
text()	conteúdo textual de um nó

Xpath Exemplos

<http://www.online-toolz.com/tools/xpath-editor.php>

```
<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>
```


Xpath

Exemplos

/fichario/individuo

```
<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>
```

Xpath

Exemplos

/fichario/individuo

```
<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>
```

```
<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>
```

Xpath

Exemplos

//individuo

```
<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>
```

Xpath

Exemplos

//individuo

```
<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>
```

```
<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>
```

Xpath

Exemplos

```
//individuo/@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>
```

Xpath

Exemplos

```
//individuo/@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="Asdrubal da Silva"

nome="Quincas Borba"

nome="Doriana Margarina"

Xpath

Exemplos

/fichario/*/idade

```
<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>
```

Xpath

Exemplos

/fichario/*/idade

```
<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>
```

```
<idade>15</idade>
```

```
-----
<idade>33</idade>
```

```
-----
<idade>42</idade>
```


Xpath

Exemplos

/fichario/*/idade

```
<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>
```

Xpath Exemplos

/fichario/*/idade/text()

```
<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>
```

15

33

42

XPath

Predicados

[]	predicado – filtra o conjunto de nós
[elem]	filtra nós com elemento
[@atr]	filtra nós com atributo
[n]	filtra enésimo nó
[last()]	filtra último nó
[@atr=val] [@atr>val] [elem=val] [elem>val]	filtra nós que atendem a condição

Xpath

Exemplos

```
//individuo[2]
```

```
<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>
```

Xpath

Exemplos

```
//individuo[2]
```

```
<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>
```

```
<individuo nome="Quincas Borba">  
<idade>33</idade>  
<genero>masculino</genero>  
</individuo>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]
```

```
<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>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]
```

```
<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>
```

```
<individuo nome="Quincas Borba">  
<idade>33</idade>  
<genero>masculino</genero>  
</individuo>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade
```

```
<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>
```


Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade
```

```
<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>
```

```
<idade>33</idade>
```

Xpath Exemplos

```
//individuo[@nome="Quincas Borba"]/idade/text()
```

```
<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>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade/text()
```

```
<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>
```

33

Xpath

Exemplos

```
//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>
```

Xpath Exemplos

```
//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

- Construa uma 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"
```

XQuery

FOR <variable bindings to individual nodes (elements)>

LET <variable bindings to collections of nodes (elements)>

WHERE <qualifier conditions>

RETURN <query result specification>

(Elmasri, 2011)

XQuery Exemplos

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{$message}</livro>
```


XQuery Exemplos

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{$message}</livro>
```

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

XML no resultado

- `{}` → para indicar valores de retorno dentro do XML
- `data()` → extrai o conteúdo literal do elemento ou atributo

XQuery

Label parte do XML externo

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return data($message)
```

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

XQuery

Label parte do XML externo

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>Título: {$message}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

XQuery

Label dentro da { }

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{data('Título:'),  
$message}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

- vírgula concatena com espaços intermediários.

XQuery concat()

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{concat('Título: ',  
$message)}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

- `concat()` concatena sem espaços intermediários.

XQuery XML Base

<http://www.ic.unicamp.br/~santanch/teaching/db/xml/fichario.xml>

```
<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>
```

XQuery XML Base

`http://www.ic.unicamp.br/~santanch/teaching/db/xml/fichario.xml`

- Para simplificar vou chamá-lo de

`icunicamp:fichario.xml`

XQuery

let/return

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario
```

```
<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>
```

XQuery

let/return

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
return $fichariodoc/fichario
```

```
<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>
```

```
<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>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario/individuo/idade
```

```
<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>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario/individuo/idade
```

```
<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>
```

```
<idade>15</idade>  
<idade>33</idade>  
<idade>42</idade>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc//individuo[idade>20][genero="masculino"]
```

```
<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>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc//individuo[idade>20][genero="masculino"]
```

```
<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>
```

```
<individuo nome="Quincas Borba">  
  <idade>33</idade>  
  <genero>masculino</genero>  
</individuo>
```

XQuery count()

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return count($fichariodoc//individuo)
```

```
<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>
```

XQuery count()

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return count($fichariodoc//individuo)
```

```
<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>
```

3

XQuery

count() e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
return count($fichariodoc//individuo[idade>20]  
[genero="masculino"])
```

```
<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>
```

XQuery count() e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
return count($fichariodoc//individuo[idade>20]  
[genero="masculino"])
```

```
<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>
```

1

XQuery for

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return $i/idade
```

```
<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>
```

XQuery for

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return $i/idade
```

```
<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>
```

```
<idade>15</idade>  
<idade>33</idade>  
<idade>42</idade>
```

XQuery where

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

```
<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>
```

XQuery where

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

```
<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>
```

```
<individuo nome="Quincas Borba">  
  <idade>33</idade>  
  <genero>masculino</genero>  
</individuo>  
<individuo nome="Doriana Margarina">  
  <idade>42</idade>  
  <genero>feminino</genero>  
</individuo>
```

XQuery

XML no resultado

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

```
<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>
```

XQuery

XML no resultado

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

```
<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 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 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)}
```

XQuery

XML no resultado

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

```
<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>
```

```
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```

XQuery

XML no resultado

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

```
<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>
```

```
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```

XQuery

Xquery dentro de outro XQuery

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

```
<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>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<classificacao>
  <maior>Quincas Borba</maior>
  <maior>Doriana Margarina</maior>
</classificacao>
```

Exercício para Casa 1

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

XQuery order by

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

```
<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>
```

XQuery order by

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

```
<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>
```

```
<maior>Doriana Margarina</maior>  
<maior>Quincas Borba</maior>
```


XQuery if

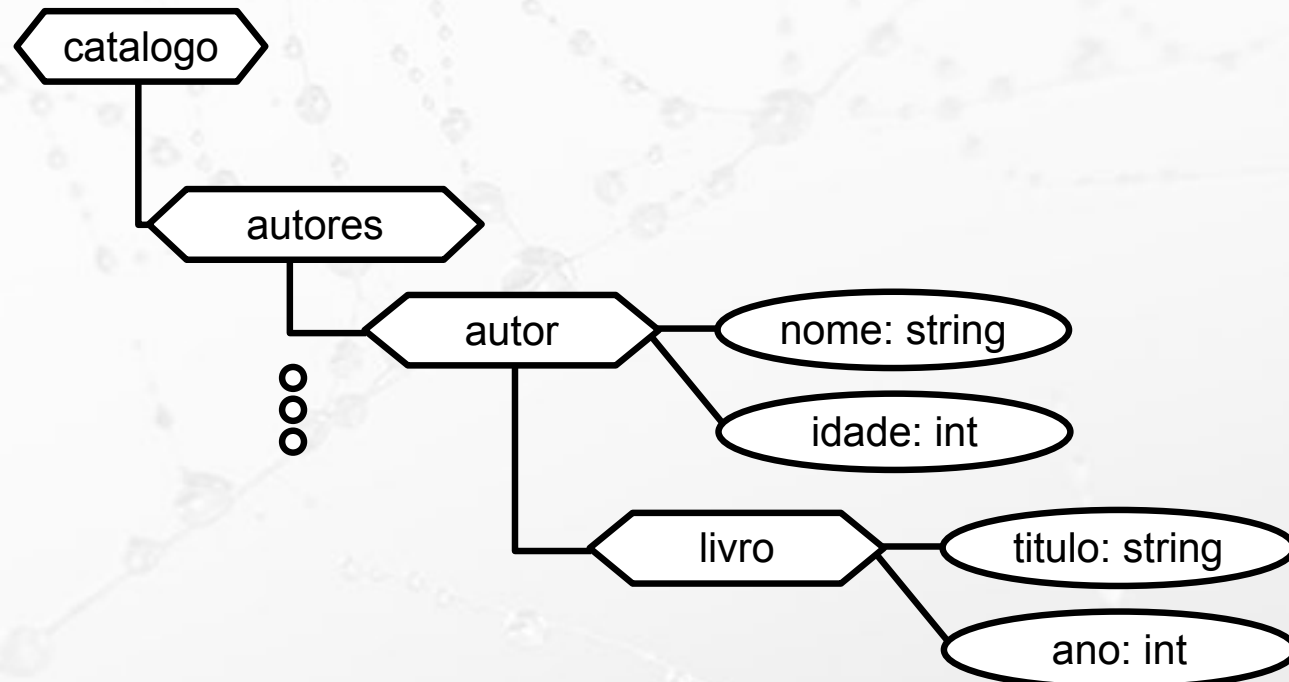
```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return if ($i[idade>=18])  
  then <maior>{data($i/@nome)}</maior>  
  else <menor>{data($i/@nome)}</menor>
```

```
<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>
```

```
<menor>Asdrubal da Silva</menor>  
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```


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



Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Doriana">
```

```
    <livro titulo="Horacio o dino"  
ano="2015" />
```

```
  </autor>
```

```
  ...
```

```
</autores>
```

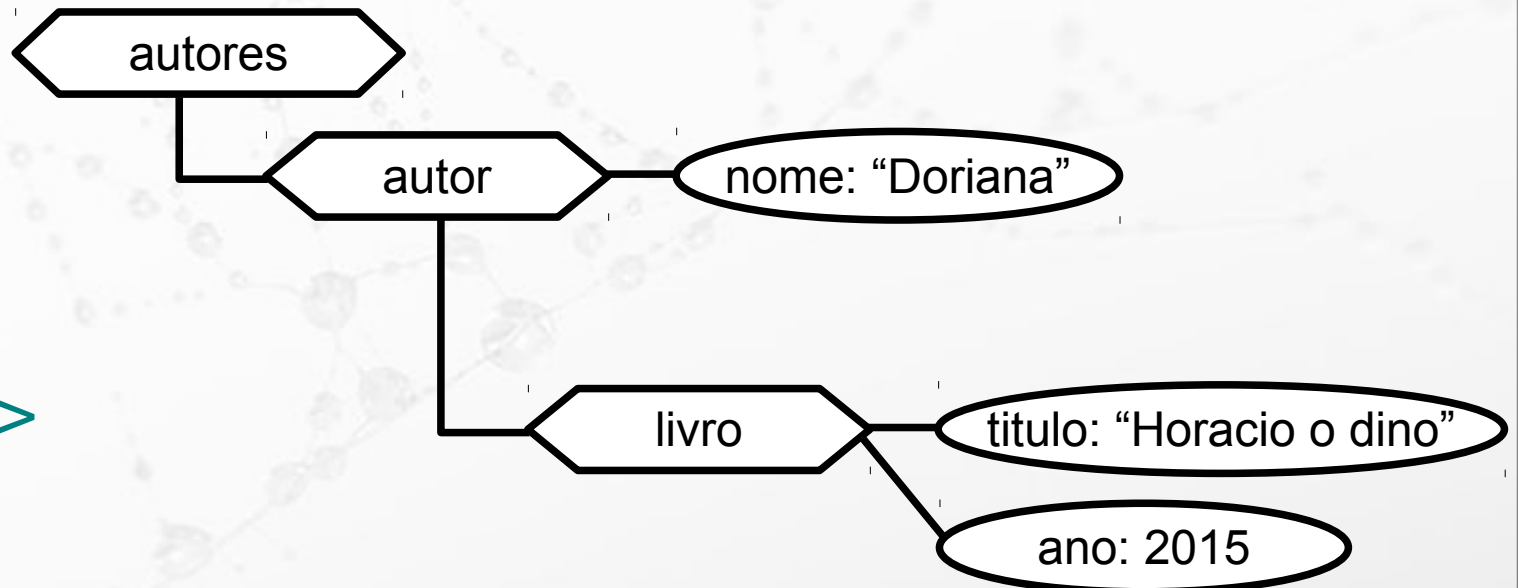
```
...
```

```
</catalogo>
```

563c871c45400d49a72753c4
nome: "Doriana"



titulo: "Horacio o dino" ano: 2015



Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Asdrubal" idade="25"
```

```
    <livro titulo="Vida sentimental dos dinossauros"  
ano="2013" />
```

```
  </autor>
```

```
...
```

```
</autores>
```

```
...
```

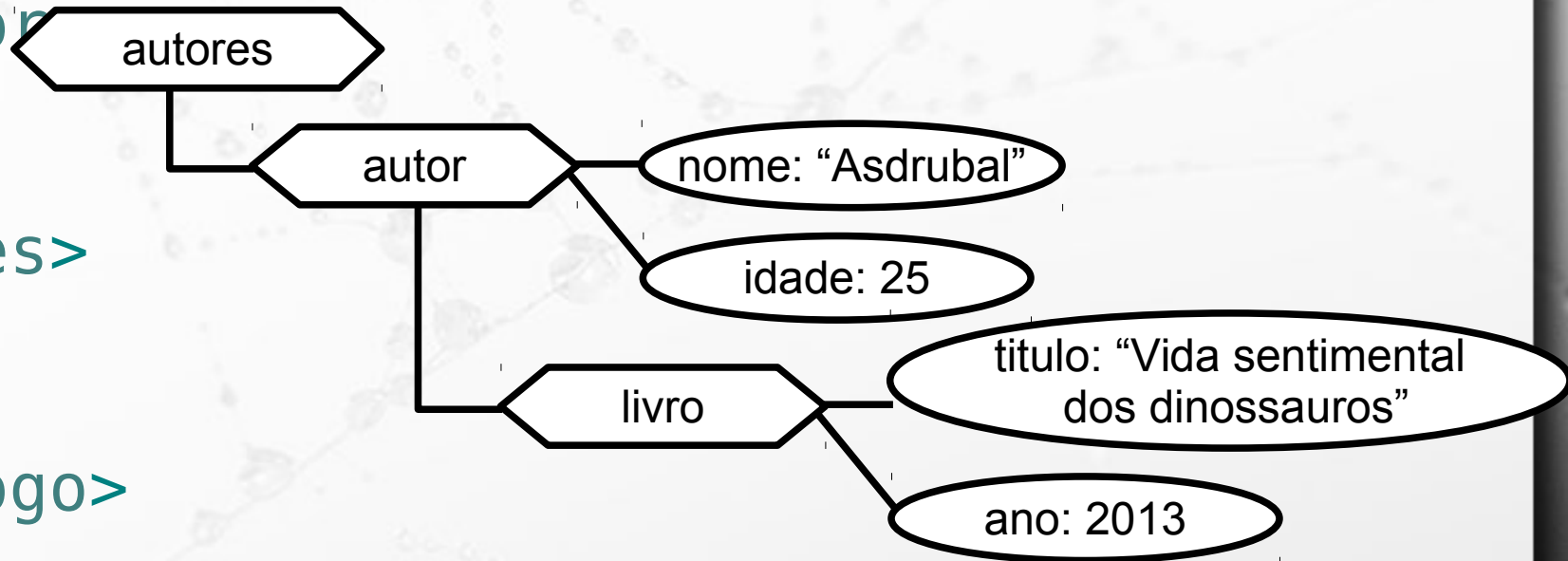
```
</catalogo>
```

563c873745400d49a72753c5

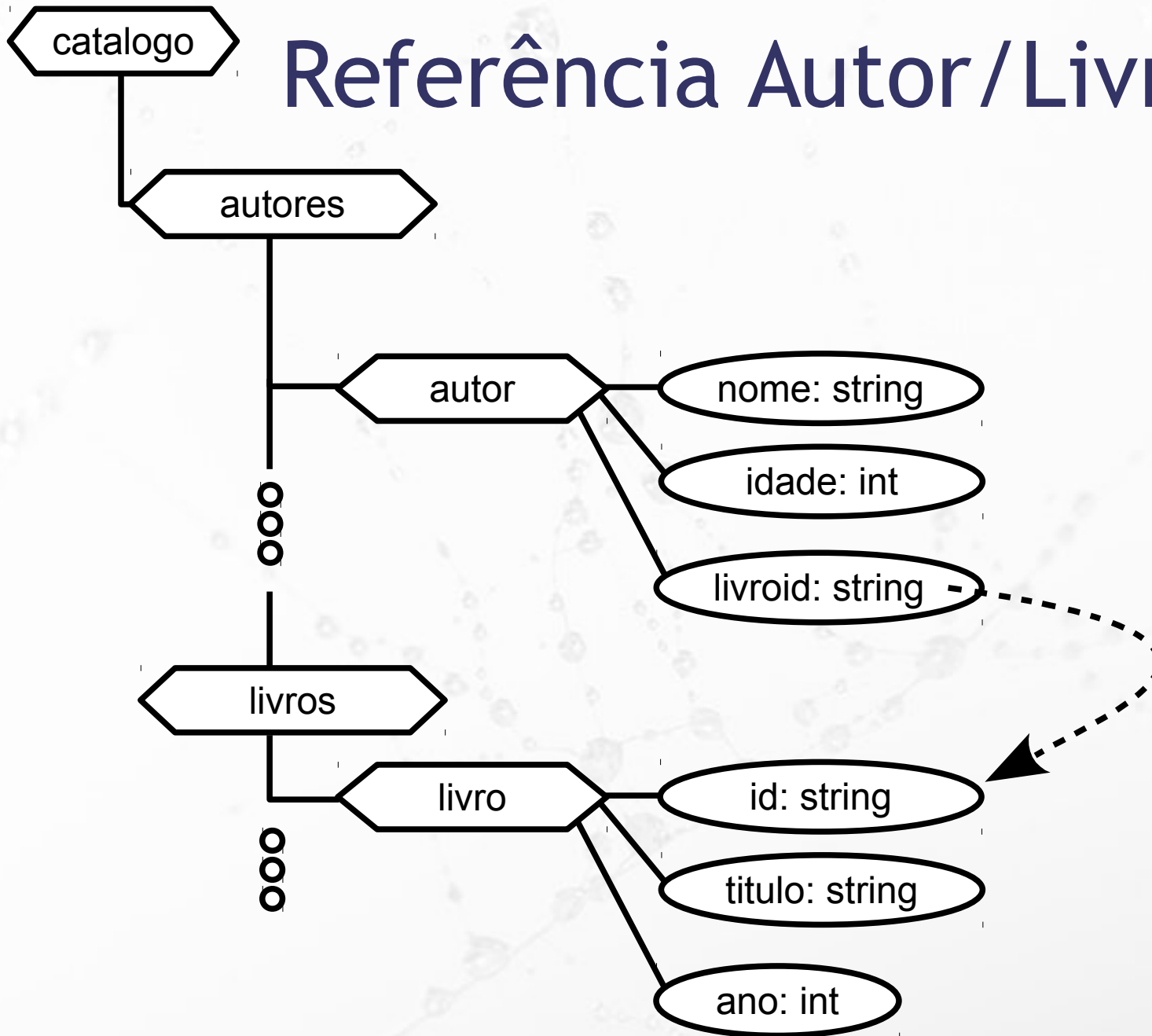
nome: "Asdrubal"
idade: 25

livro

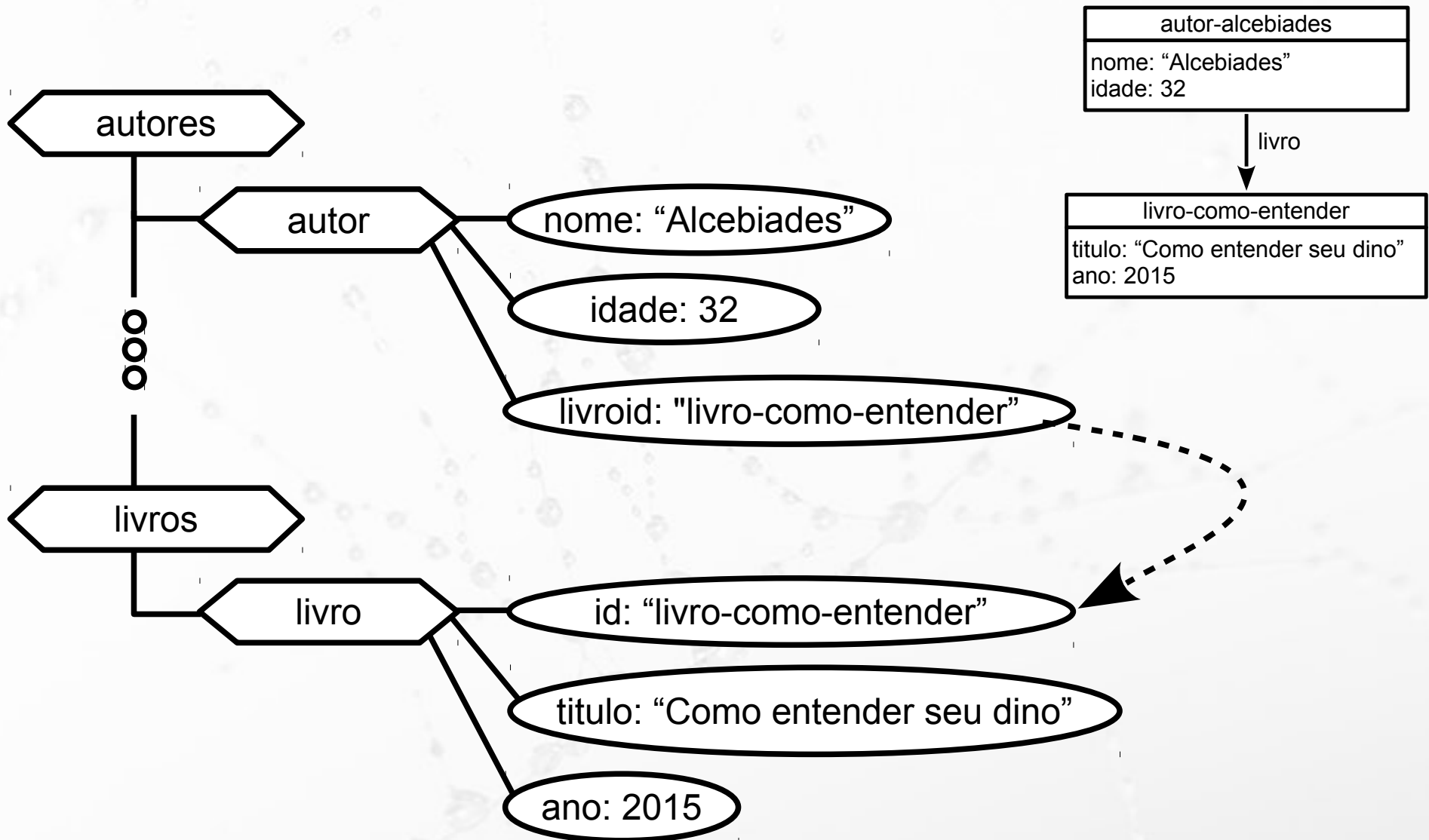
titulo: "Vida sentimental dos dinossauros"
ano: 2013



Referência Autor/Livro

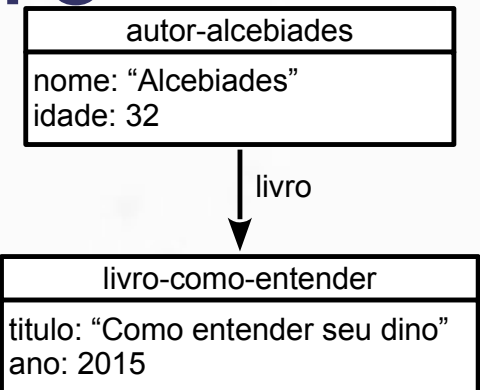


Referência Autor/Livro

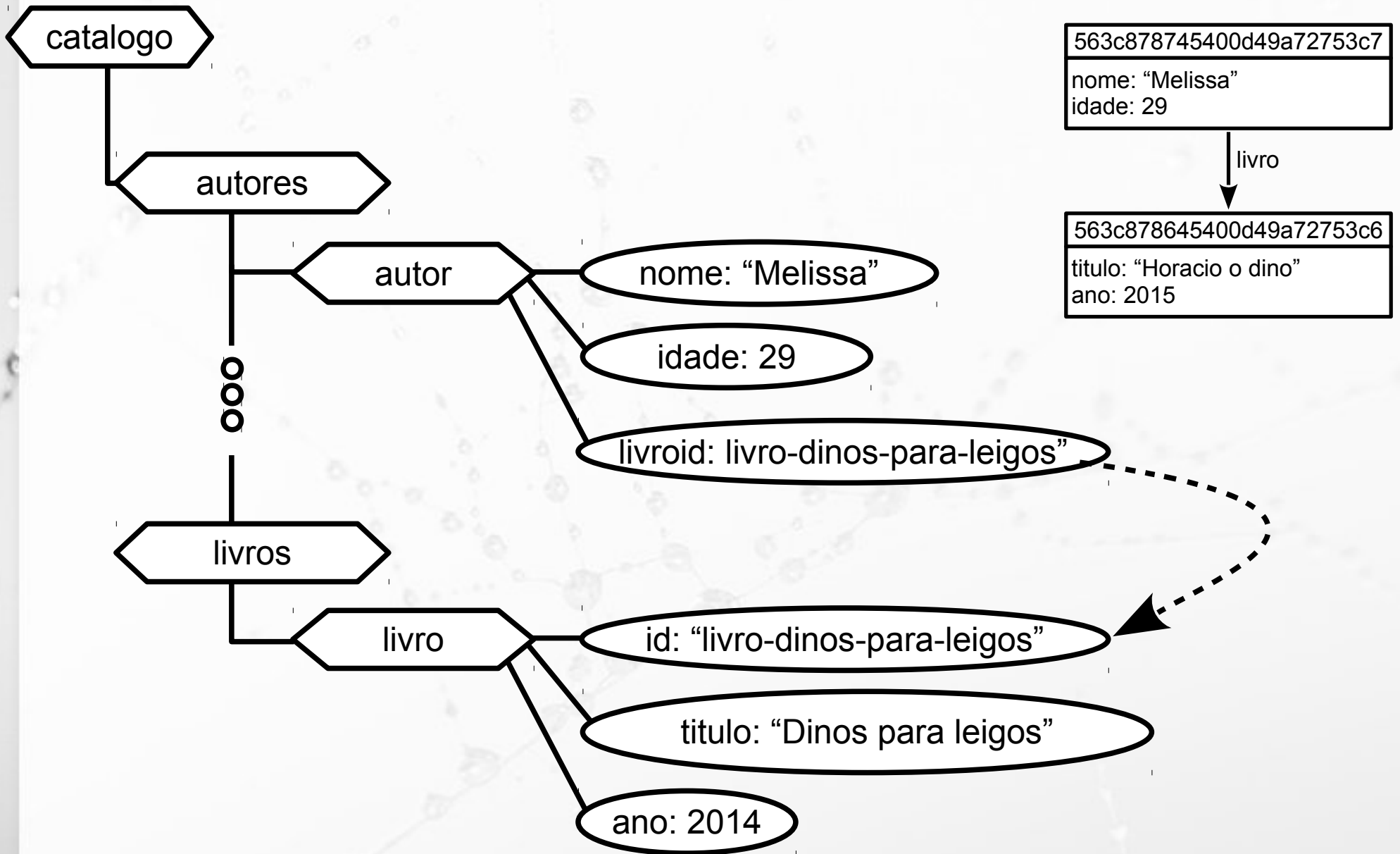


Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Alcebiades"
        idade="32"
        livroid="livro-como-entender" />
  ...
</autores>
  ...
<livros>
  <livro id="livro-como-entender"
        titulo="Como entender seu dino"
        ano="2015" />
  ...
</livros>
</catalogo>
```

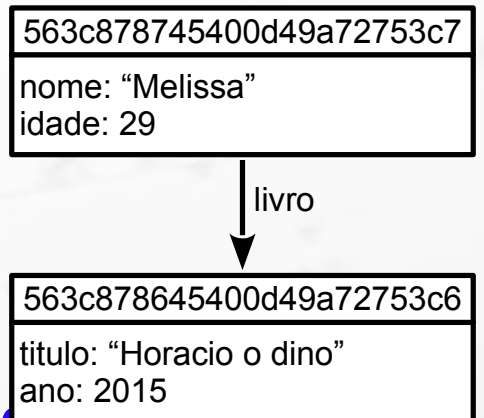


Referência Autor/Livro



Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Melissa"
        idade="29"
        livroid="livro-dinos-para-
leigos" />
  ...
</autores>
...
<livros>
  <livro id="livro-dinos-para-leigos"
        titulo="Dinos para leigo"
        ano="2014" />
  ...
</livros>
</catalogo>
```



XQuery Join

- Retorne o nome dos autores e o título de seus livros:

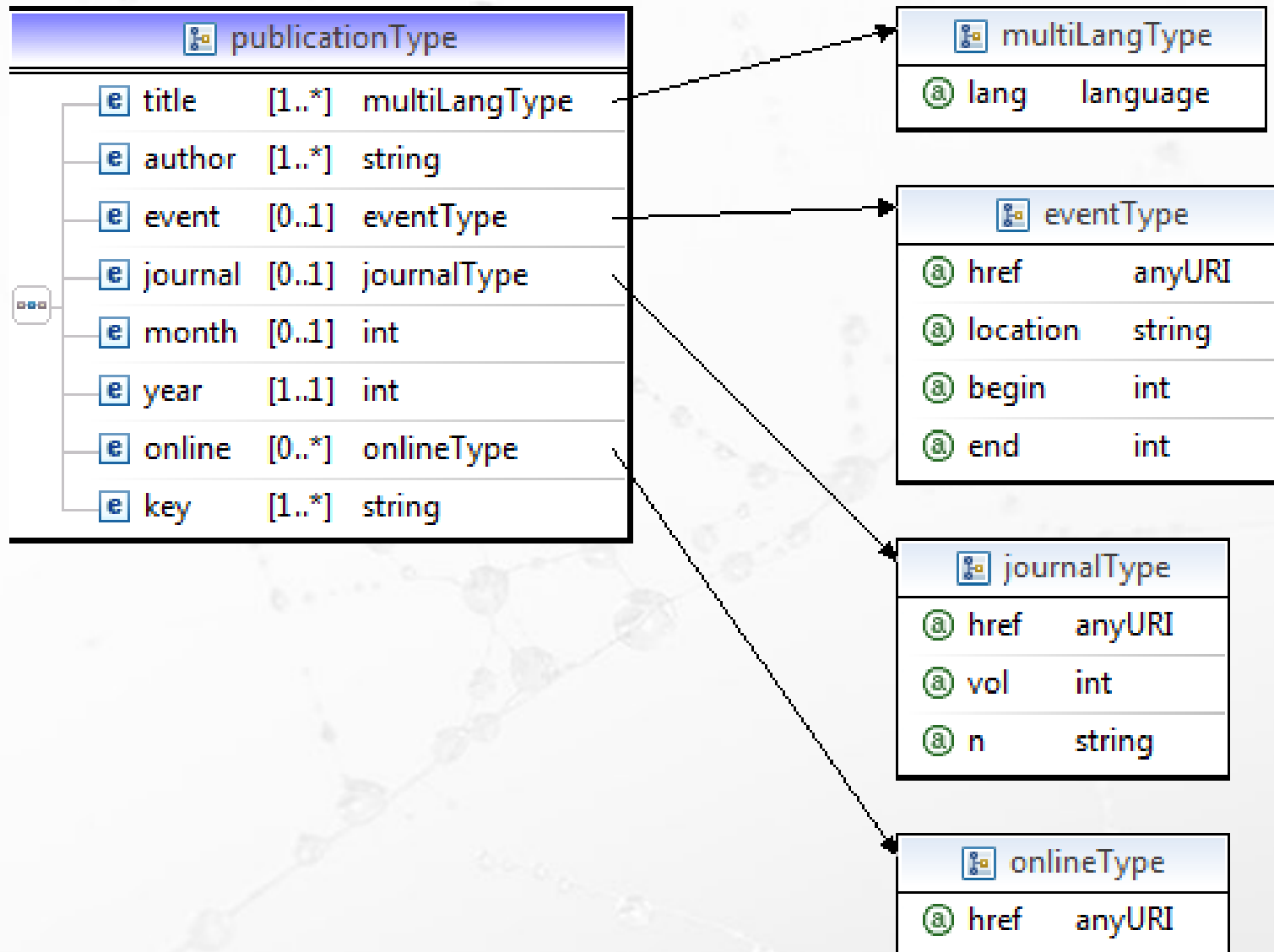
```
let $autorlivro :=  
doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/autor-  
livro-referencia.xml')  
  
for $a in ($autorlivro/catalogo/autores/autor),  
    $l in ($autorlivro/catalogo/livros/livro)  
where $a/@livroid = $l/@id  
  
return {concat('Autor: ', data($a/@nome), ', Livro: ',  
data($l/@titulo))}
```

XQuery group by

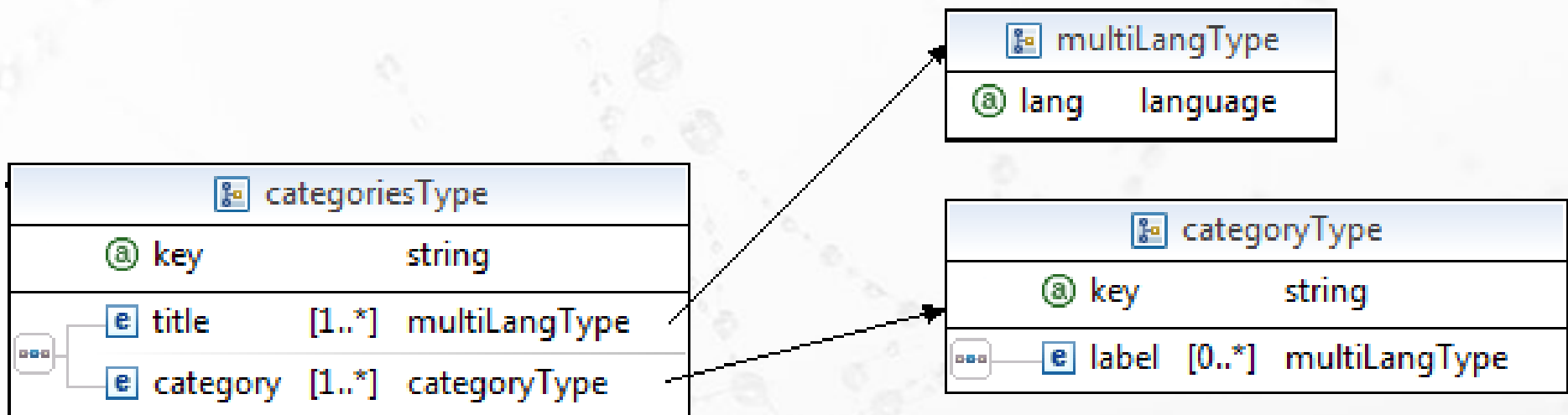
- Retorne o número de livros por ano:

```
let $autorlivro :=  
doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/autor-livro-referencia.xml')  
  
for $l in ($autorlivro/catalogo/livros/livro)  
group by $l/@ano  
return {'Livros -- ano: ', 'quantidade: ', count($l)}
```

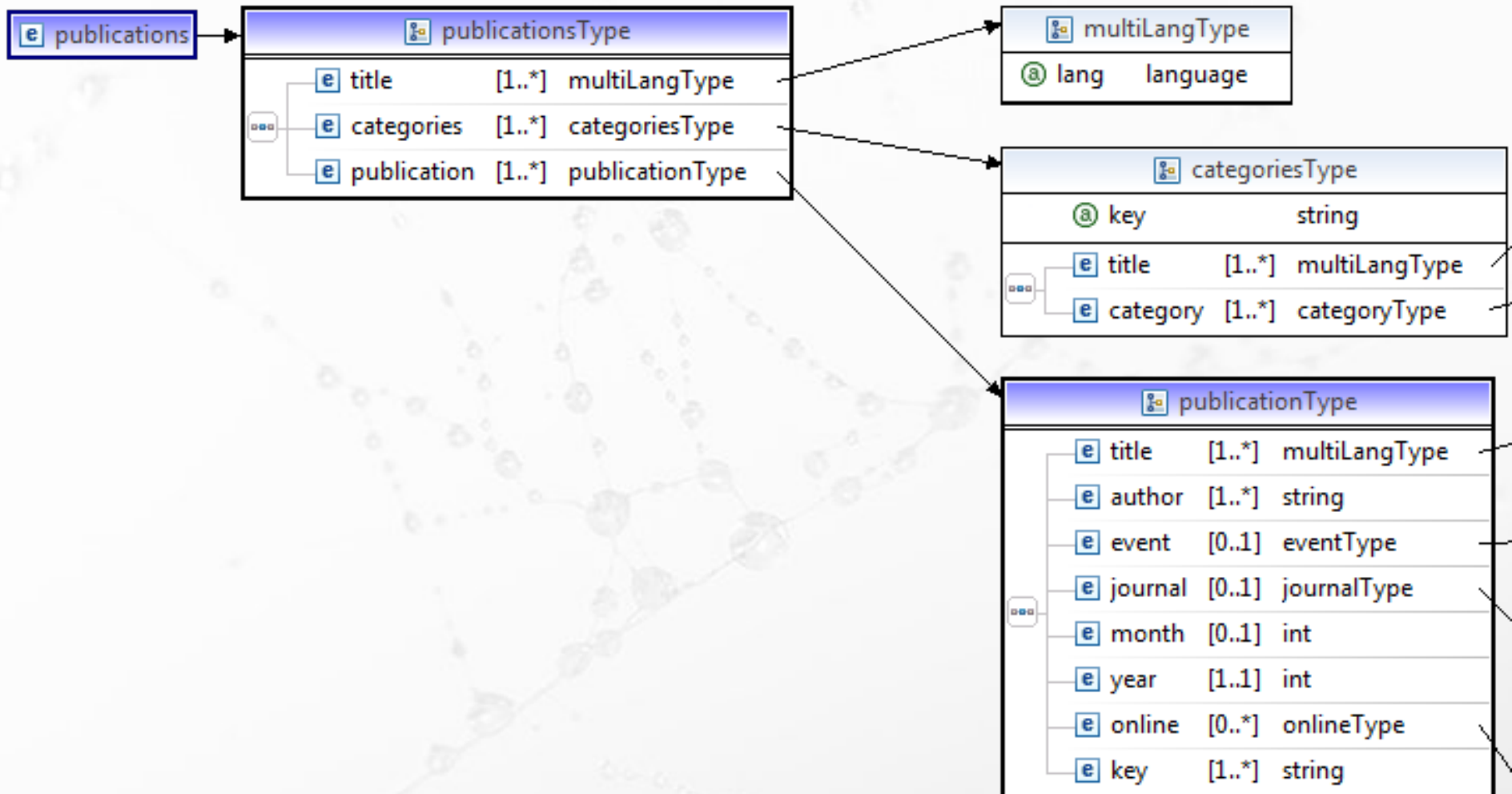
Publication Type



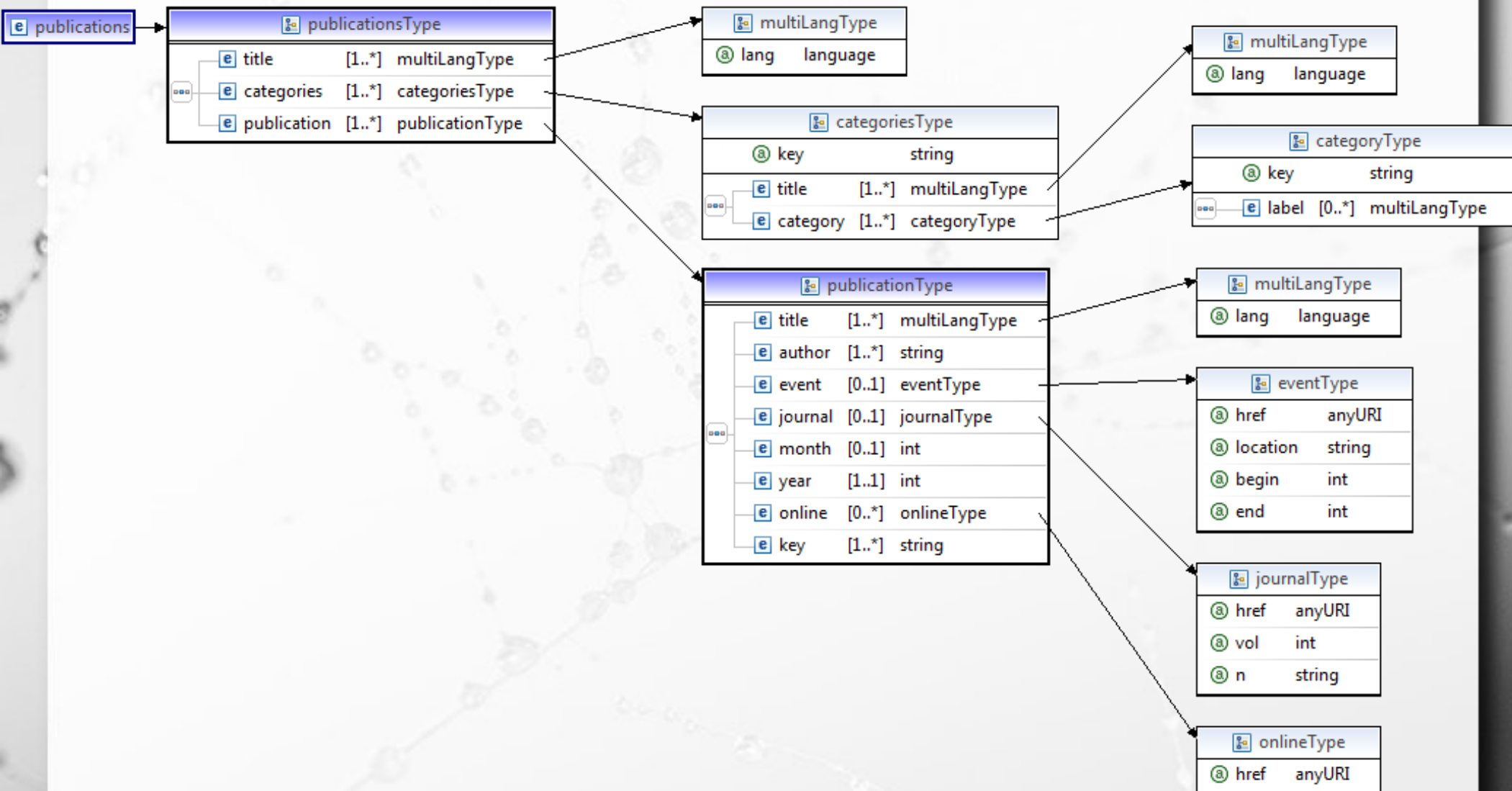
CategoryType and CategoriesType



publicationsType



Everything together



Exercício 9

- Retorne quantas publicações são posteriores ao ano de 2011

Exercício 9

- 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

- 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

count

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

Join

- Retorne as publicações associadas à categoria cujo <label> em inglês seja 'e-Science Domain'. A associação entre o label e a key da categoria deve ser feita na consulta.

```
for $c in ($publicadoc//categories/category),  
    $p in ($publicadoc//publication)  
where $c/label[@lang='en-US'] = 'e-Science  
Domain' and  
    $p/key = $c/@key  
return $p
```

Join

- Retorne as publicações associadas à categoria cujo <label> em inglês seja 'e-Science Domain'. A associação entre o label e a key da categoria deve ser feita na consulta.

```
for $c in ($publicadoc//categories/category),  
    $p in ($publicadoc//publication)  
where $c/label[@lang='en-US'] = 'e-Science  
Domain' and  
    $p/key = $c/@key  
return $p
```

```
LET $d := doc(www.company.com/info.xml)
FOR $x IN $d/company/project[projectNumber = 5]/projectWorker,
    $y IN $d/company/employee
WHERE $x/hours gt 20.0 AND $y.ssn = $x.ssn
RETURN <res> $y/employeeName/firstName, $y/employeeName/lastName,
    $x/hours </res>
```

-
1. FOR \$x IN
doc(www.company.com/info.xml)
//employee [employeeSalary gt 70000]/employeeName
RETURN <res> \$x/firstName, \$x/lastName </res>
 2. FOR \$x IN
doc(www.company.com/info.xml)/company/employee
WHERE \$x/employeeSalary gt 70000
RETURN <res> \$x/employeeName/firstName, \$x/employeeName/lastName </res>
 3. FOR \$x IN
doc(www.company.com/info.xml)/company/project[projectNumber = 5]/projectWorker,
\$y IN doc(www.company.com/info.xml)/company/employee
WHERE \$x/hours gt 20.0 AND \$y.ssn = \$x.ssn
RETURN <res> \$y/employeeName/firstName, \$y/employeeName/lastName, \$x/hours </res>

Figure 12.7

Some examples of XQuery queries on XML documents that follow the XML schema file *company* in Figure 12.5.

(Elmasri, 2011)


```
LET $d := doc(www.company.com/info.xml)
FOR $x IN $d/company/project[projectNumber = 5]/projectWorker,
    $y IN $d/company/employee
WHERE $x/hours gt 20.0 AND $y.ssn = $x.ssn
RETURN <res> $y/employeeName/firstName, $y/employeeName/lastName,
    $x/hours </res>
```

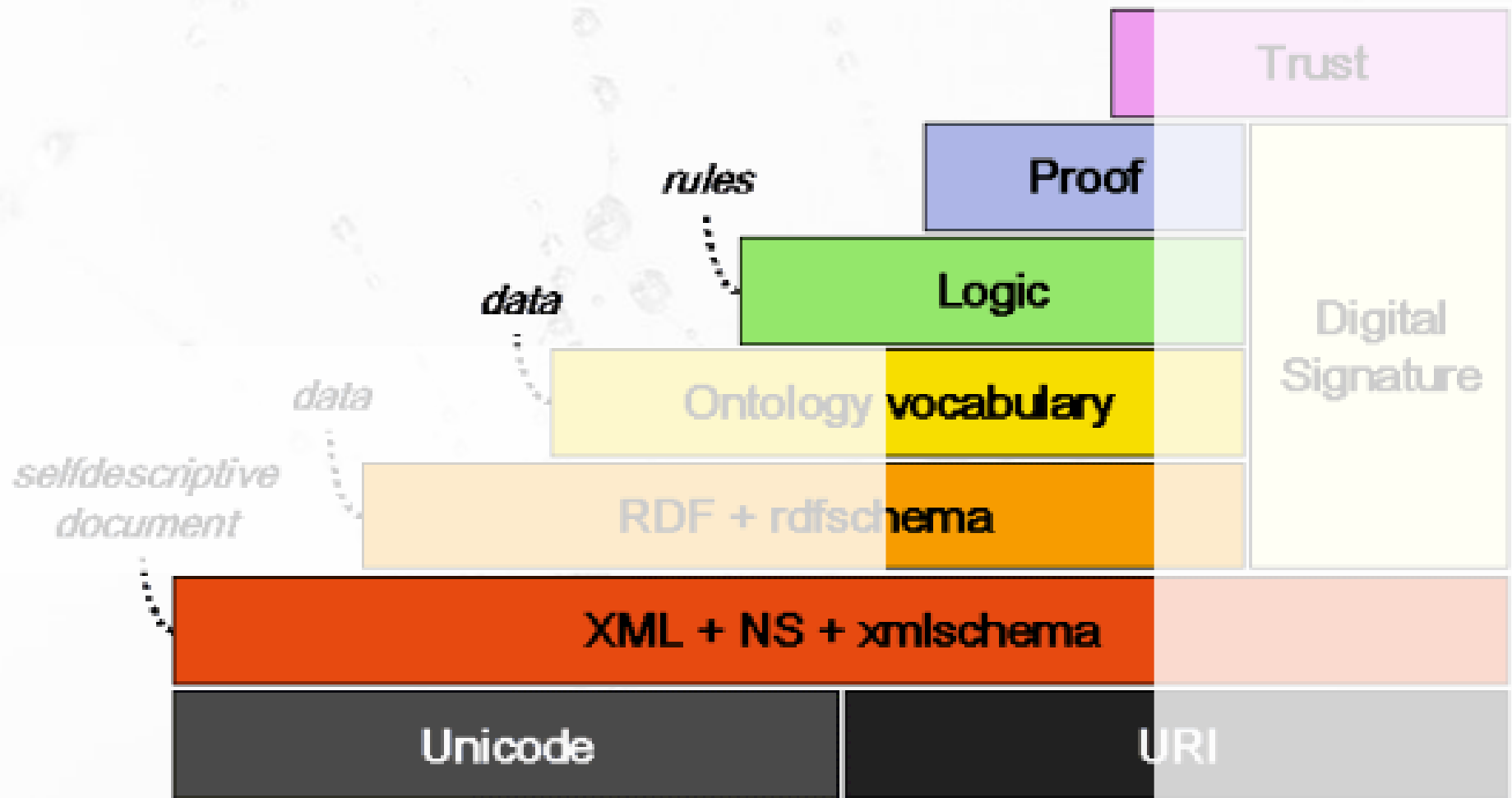
-
1. FOR \$x IN
doc(www.company.com/info.xml)
//employee [employeeSalary gt 70000]/employeeName
RETURN <res> \$x/firstName, \$x/lastName </res>
 2. FOR \$x IN
doc(www.company.com/info.xml)/company/employee
WHERE \$x/employeeSalary gt 70000
RETURN <res> \$x/employeeName/firstName, \$x/employeeName/lastName </res>
 3. FOR \$x IN
doc(www.company.com/info.xml)/company/project[projectNumber = 5]/projectWorker,
\$y IN doc(www.company.com/info.xml)/company/employee
WHERE \$x/hours gt 20.0 AND \$y.ssn = \$x.ssn
RETURN <res> \$y/employeeName/firstName, \$y/employeeName/lastName, \$x/hours </res>

Figure 12.7

Some examples of XQuery queries on XML documents that follow the XML schema file *company* in Figure 12.5.

(Elmasri, 2011)

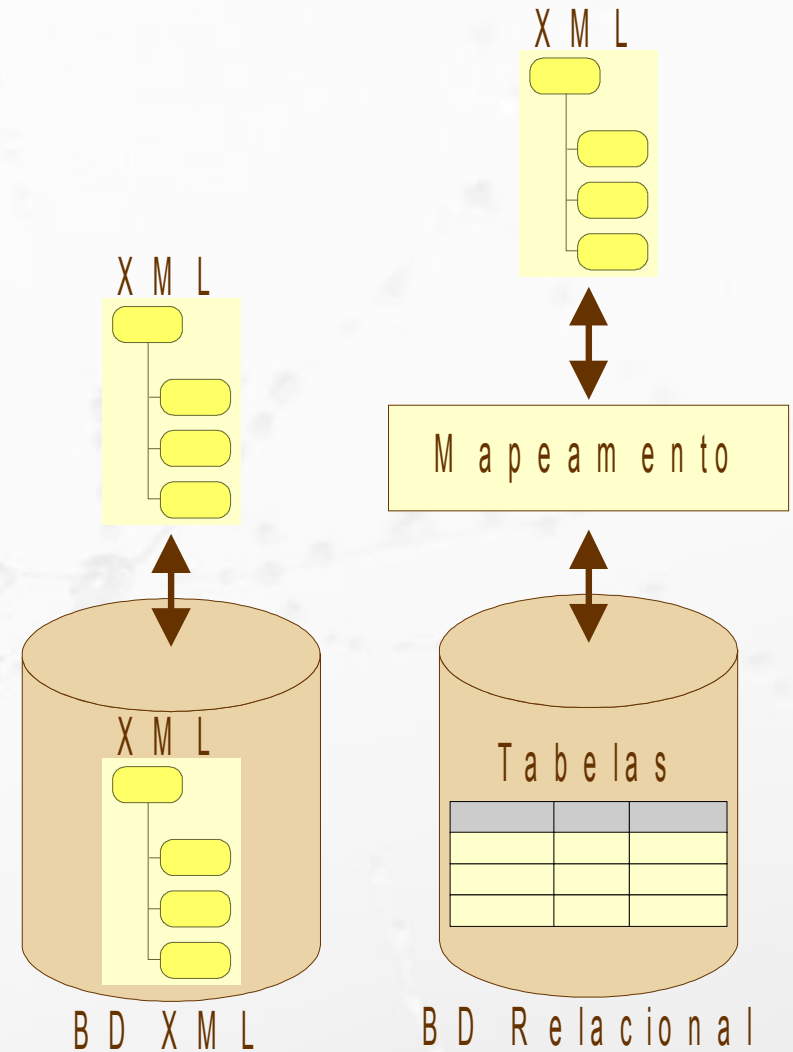
Web Semântica



XML e Bancos de Dados

Banco de dados & XML

- Dois tipos:
 - SGBD XML nativo;
 - SGBD relacional que mapeia dados XML para sua estrutura interna e vice-versa.

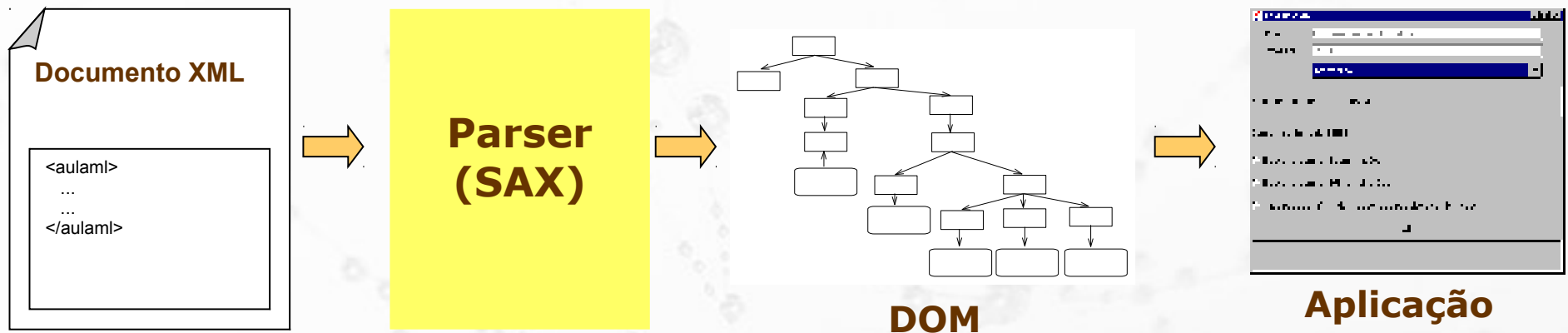


Banco de dados & XML

Aspectos conceituais

- Padrão para:
 - intercâmbio de dados × armazenamento.
- Modelo:
 - documentos × dados (registros).

Parser XML

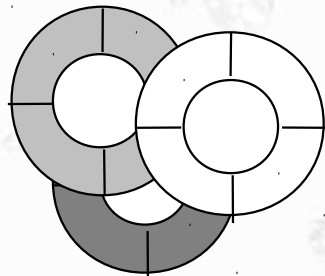


Introdução

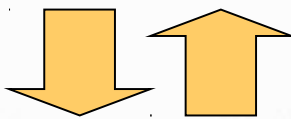
Diversas tecnologias têm sido criadas para o processamento de documentos XML.

Aplicação

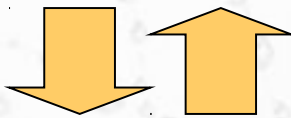
Server Pages



Classes



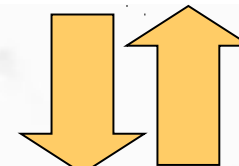
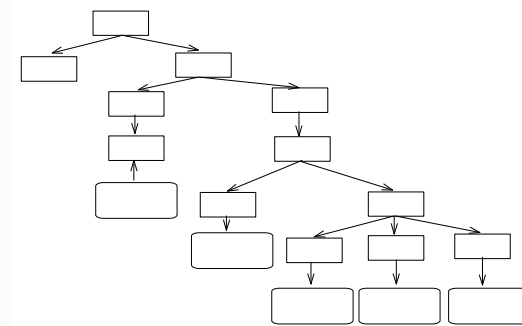
Data-Binding



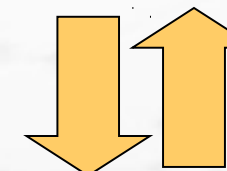
Eventos



SAX



DOM



```
<aulaml>
<curso>
...
</curso>
<quadro>
<texto>
...
</texto>
<teste>
...
</teste>
</quadro>
</aulaml>
```

```
<aulaml>
<curso>
...
</curso>
<quadro>
<texto>
...
</texto>
<teste>
...
</teste>
</quadro>
</aulaml>
```

```
<aulaml>
<curso>
...
</curso>
<quadro>
<texto>
...
</texto>
<teste>
...
</teste>
</quadro>
</aulaml>
```

```
<aulaml>
<curso>
...
</curso>
<quadro>
<texto>
...
</texto>
<teste>
...
</teste>
</quadro>
</aulaml>
```

XML

Introdução

Dentre estas tecnologias duas se destacaram e se tornaram referência:

- SAX - Simple API for XML
- DOM - Document Object Model

SAX

- API baseada em eventos.
- Se tornou a mais estável API XML largamente utilizada [DOD01].
- Iniciou como uma solução para acesso a documentos XML por programas Java.
- Hoje tem sido portada para outras linguagens de programação, tal como: C++, Pascal, Perl, Python, etc.

SAX - Estudo de Caso

```
<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>
```

SAX - Estudo de Caso

===== Início do Documento =====

Início de elemento: DOCUMENTO

Início de elemento: individuo

(atributos): nome=Asdrubal da Silva;

Início de elemento: IDADE

Texto: 15

Final de elemento : IDADE

Início de elemento: GENERO

Texto: masculino

Final de elemento : GENERO

Final de elemento : individuo

Início de elemento: individuo

(atributos): nome=Quincas Borba;

Início de elemento: IDADE

Texto: 33

Final de elemento : IDADE

Início de elemento: GENERO

Texto: masculino

Final de elemento : GENERO

Final de elemento : individuo

Início de elemento: individuo

(atributos): nome=Doriana Margarina;

Início de elemento: IDADE

Texto: 42

Final de elemento : IDADE

Início de elemento: GENERO

Texto: feminino

Final de elemento : GENERO

Final de elemento : individuo

Final de elemento : DOCUMENTO

===== Final do Documento =====

Eventos de conteúdo

```
public class SAXBasico extends
    org.xml.sax.helpers.DefaultHandler
{

    public void startDocument () ...

    public void startElement (...) ...

    public void characters (...) ...

    public void endElement (...) ...

    public void endDocument () ...

}
```

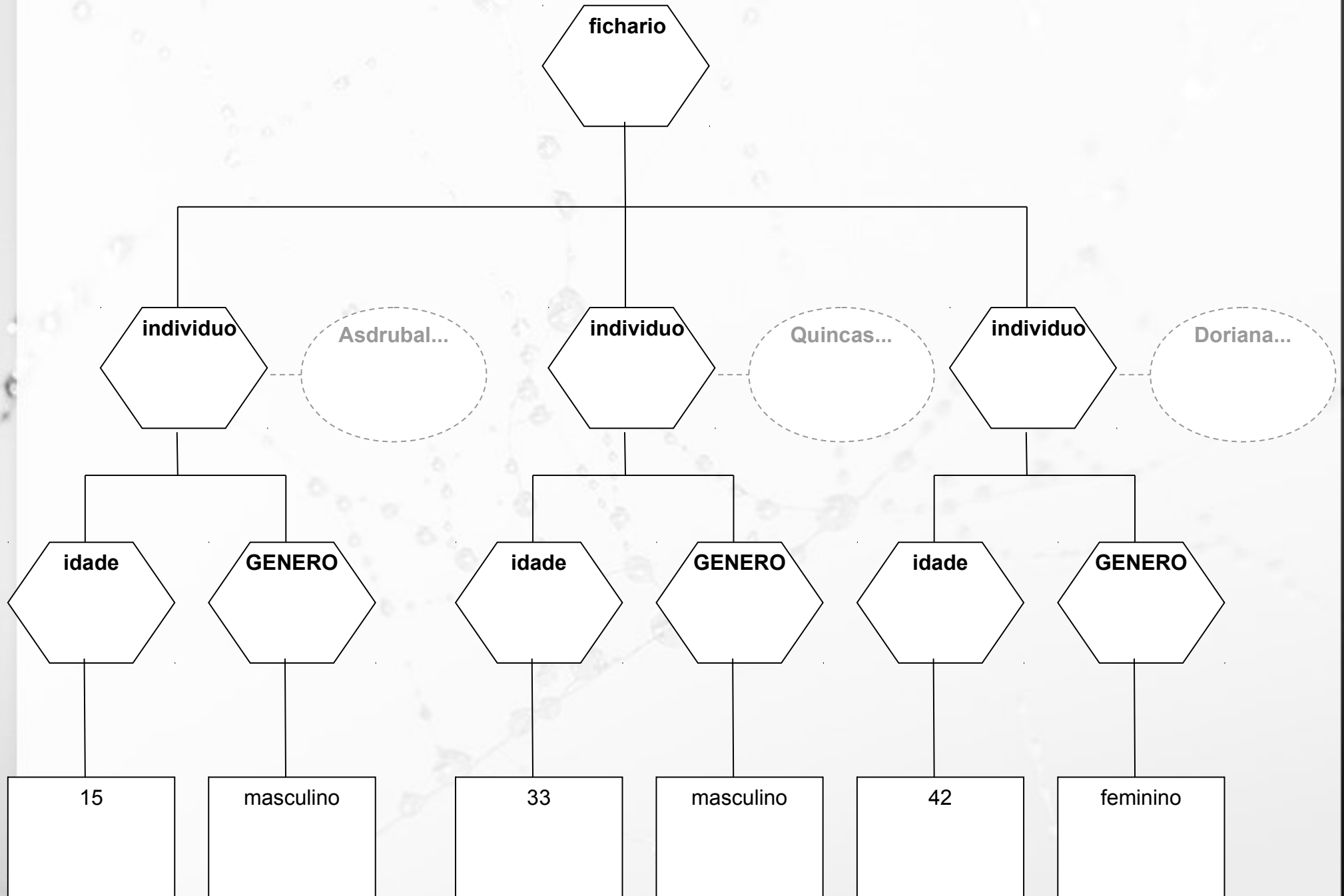
Eventos de conteúdo

Método	Acionado quando o <i>parser</i> encontra
<code>startDocument</code>	início do documento
<code>startElement</code>	início de um elemento
<code>characters</code>	conteúdo texto
<code>endElement</code>	final de um elemento
<code>endDocument</code>	final do documento

DOM

- DOM define uma API para documentos XML e HTML.
- Ele acrescenta ao padrão destas linguagens toda a funcionalidade e flexibilidade que um programa precisa para acessar e manipular documentos.
- Definido em IDL, ECMAScript e Java.

Document Object Model



DOM Level

- **Level 0** - define funcionalidades equivalentes ao Netscape Navigator 3.0 e o Microsoft Internet Explorer 3.0.
- **Level 1** - especifica recursos para navegação e manipulação de estrutura e conteúdo de documentos XML e HTML.
- **Level 2** - estende alguns recursos do *Level 1* e acrescenta suporte a: *Cascading Style Sheets*, Eventos, etc.
- **Level 3** - estende alguns recursos do *Level 2* e acrescenta suporte a: esquemas abstratos (DTD, XML Schema, etc.), recursos de leitura e gravação, etc.

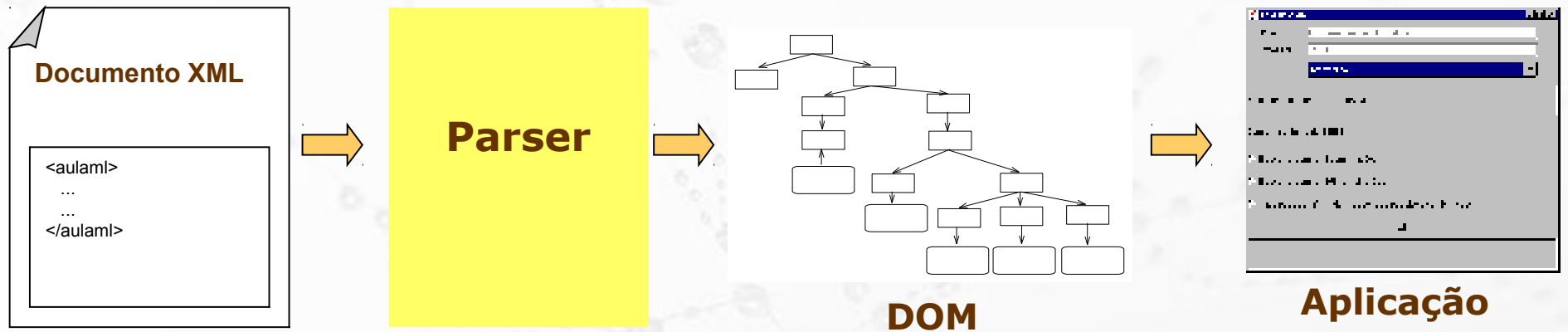
DOM Core & DOM HTML

- O *DOM Level 1* é dividido em duas partes: DOM Core e DOM HTML.
- **DOM Core** - define o conjunto de funcionalidades básicas para documentos XML.
- **DOM HTML** - está montado sobre o DOM Core e acrescenta funcionalidades para lidar com HTML.

DOM - Estudo de Caso

```
<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>
```

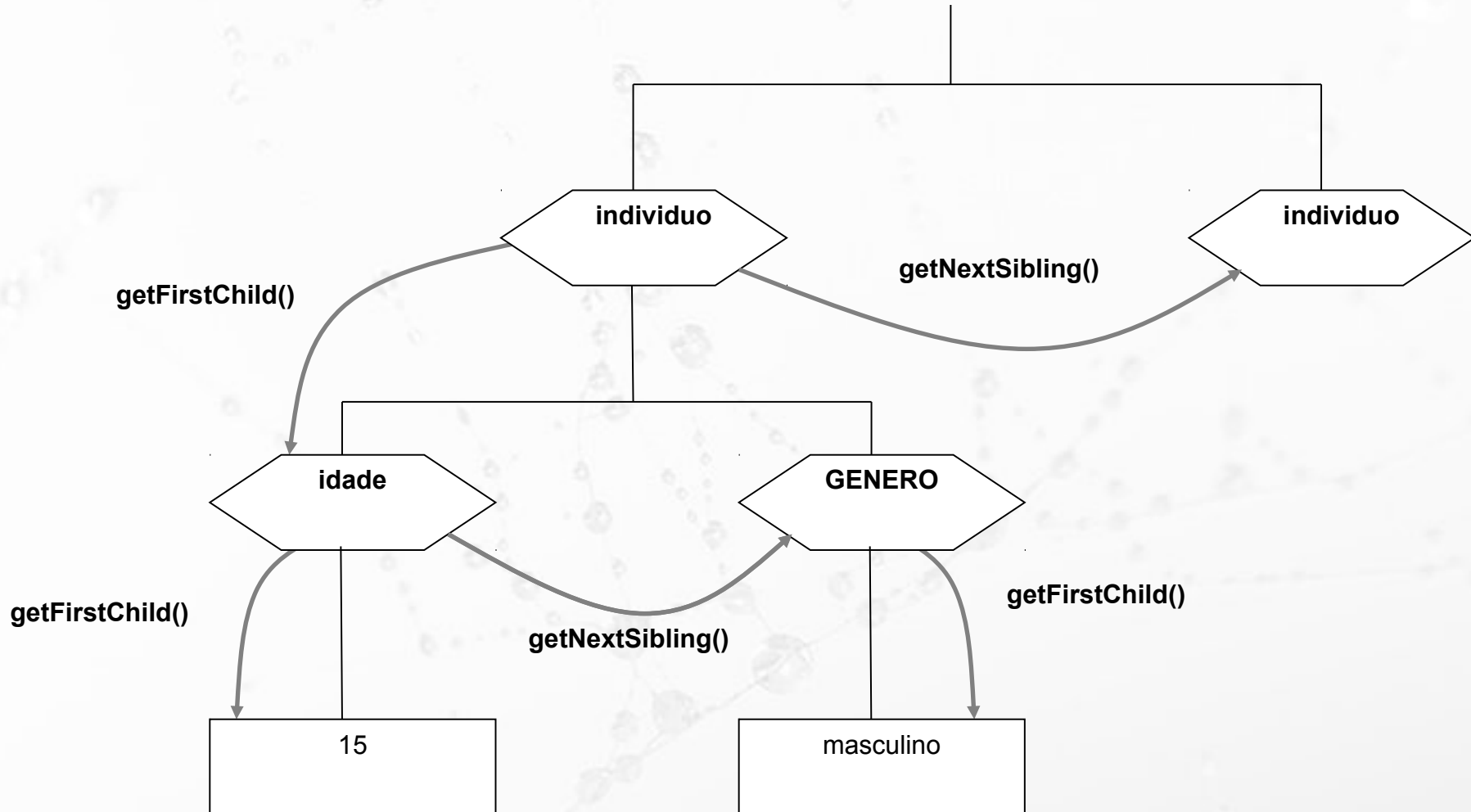
Processo



Interfaces

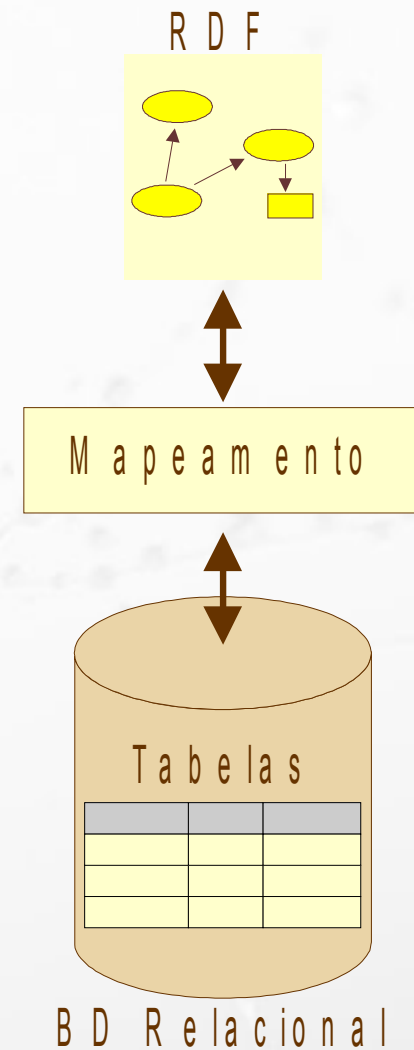
- **Node** - esta interface representa genericamente qualquer nó da árvore.
- **Element** - acrescenta propriedades e métodos específicos de um nó do tipo elemento.
- **Document** - interface do nó raiz da árvore que representa o documento completo.
- **NodeList** - representa uma lista de nós. Pode representar, por exemplo, a lista de filhos de

Navegar pelo Documento

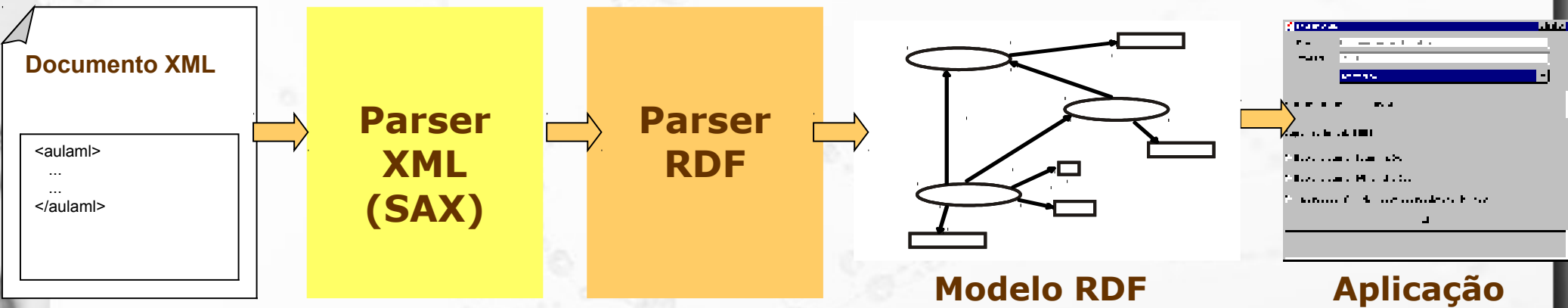


Banco de dados & RDF

- Não associado a BD XML.
- Modelo de dados RDF mais próximo ao relacional que XML.



Parser RDF



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/>