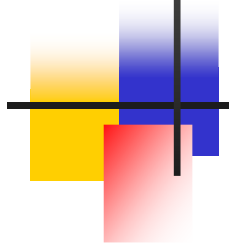




# **UTILIZANDO ANÁLISE DE MUTANTES PARA REALIZAR O TESTE DE DOCUMENTOS *XML SCHEMA***

**Ledyvânia Franzotte**

**Orientadora: Sílvia Regina Vergílio**



# Roteiro

---

- Introdução
- XML e Tecnologias
  - XML, DTD, XML Schema e JDOM
- Testes de Software
  - Trabalhos Relacionados
- Operadores de Mutação
- Ferramenta XTM
- Experimentos
- Conclusões Trabalhos Futuros



# Introdução

---

- XML (eXtensible Markup Language)
- Validações: DTD x XML Schema
- Atividade de Teste
  - Critério Análise de Mutantes
- Testes de aplicações Web
  - Web Service
  - XML Schema



# Motivação

---


- Grande utilização de documentos XML Schema
- Eficácia do Critério Análise de Mutantes
- Defeitos existentes em documentos XML devido a documentos XML Schema incorretos



# Objetivo

---

- Explorar o uso do critério Análise de Mutantes no teste de documentos XML Schema.
- Conjunto de Operadores de Mutação
- Ferramenta XTM



# XML e Tecnologias

---

- XML
  - Linguagem de marcação para troca de informação
  - Documento definido como uma hierarquia
  - Documento válido e bem-formatado

# XML e Tecnologias

```
<?xml version="1.0" encoding="UTF-8"?> ← opcional  
<!DOCTYPE dissertacoes_mestrado SYSTEM "dissertacoes.dtd"> ← DTD para validação  
  
<dissertacoes_mestrado> ← nó raiz  
<tese> todas as tags abertas estão devidamente fechadas  
  <aluno>Ledyvania Franzotte</aluno> ← elemento  
  <orient>Silvia Regina Vergilio</orient>  
  <titulo>Mutações para Documentos XML Schema</titulo>  
  <area>Engenharia de Software</area>  
</tese>  
</dissertacoes_mestrado> ← tag fechando o nó raiz
```



# XML e Tecnologias

---


- DTD (*Document Type Definition*)
  - Define os tipos de elementos de um documento XML
    - *Element, Attlist, Entity e Notation*
  - Gramática EBNF (*Extended Backus Naur Form*)





# XML e Tecnologias

```
<!ELEMENT dissertacoes_mestrado (tese+) > ← composição do nó raiz  
      (1 ou mais elemento tese)  
<!ELEMENT tese (aluno, orient, titulo, area) > ← composição do nó  
<!ELEMENT aluno (#PCDATA) >  
<!ELEMENT orient (#PCDATA) >      descrição do tipo de cada nó  
<!ELEMENT titulo (#PCDATA) >  
<!ELEMENT area (#PCDATA) >
```



# XML e Tecnologias

---

- XML Schema
  - Metadados – documento XML que descreve outro documento XML
  - Possui os tipos primitivos de dados
    - Cadeias, números, booleanos e datas
  - Descreve outros tipos
  - Possui uma estrutura mais rica que a DTD

# XML e Tecnologias


```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="dissertacoes_mestrado"> ← nó raiz
    <xs:complexType> ← composição do nó
      <xs:sequence>
        <xs:element name="tese" minOccurs="1"> ← 1 ou mais elementos
          <xs:complexType>
            <xs:sequence> ← nós serão sequenciais
              <xs:element name="aluno" type="xs:string" minOccurs="1"/>
              <xs:element name="orient" type="xs:string" minOccurs="1"/>
              <xs:element name="titulo" type="xs:string" minOccurs="1"/>
              <xs:element name="area" type="xs:string" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



# XML e Tecnologias

---

- API para Manipulação
  - SAX
    - Documento não é carregado para memória
    - Baseado em eventos
    - Simples de usar
  - DOM
    - Documento é interpretado como uma árvore e carregado para a memória
    - W3C -> especificação



# XML e Tecnologias

---

- JDOM
  - Extensão DOM
  - Java Puro e *Open Source*
  - Mais intuitiva que DOM
  - Possui algumas diferenças da interface DOM
    - Exceções mais específicas



# Teste de Software

---

- “A atividade de teste de software é um elemento crítico da garantia de qualidade de software...” (Pressman)
- Fases do teste: Planejar, Projetar, Executar e Avaliar
- Técnicas de projeto
  - Baseada em erros, Funcional e Estrutural



# Teste de Software

---

- Critério Análise de Mutantes
  - Baseada na Hipótese do Programador Competente e Efeito do Acoplamento.
  - Erros introduzidos no programa
  - Mutantes Mortos e Mutantes Equivalentes
  - *Score* de Mutação
  - Proteum -> linguagem C



# Trabalhos Relacionados

---

- Testes de componentes Web
  - Lee e Offutt (2001)
    - 2 operadores
      - LenOf e MemberOf
    - Altera o conteúdo do documento XML
    - Objetivo: Testar a comunicação entre as aplicações.





# Trabalhos Relacionados

---

- Testes de componentes Web
  - Offutt e Xu (2004)
    - Teste de Componente de Software via Web Service usando perturbação de dados (publicado em 2004)
    - Dados: Altera os valores dos dados -> limites especificados no XML Schema
    - Interação: RPC e Comunicação de dados
    - RPC: 4 operadores numéricos e 2 para comandos SQL
      - *Divide(n)*, *Multiply(n)*, *Negative(n)* e *Absolute(n)*
      - *Exchange(n1, n2)* e *Unauthorized(str)*



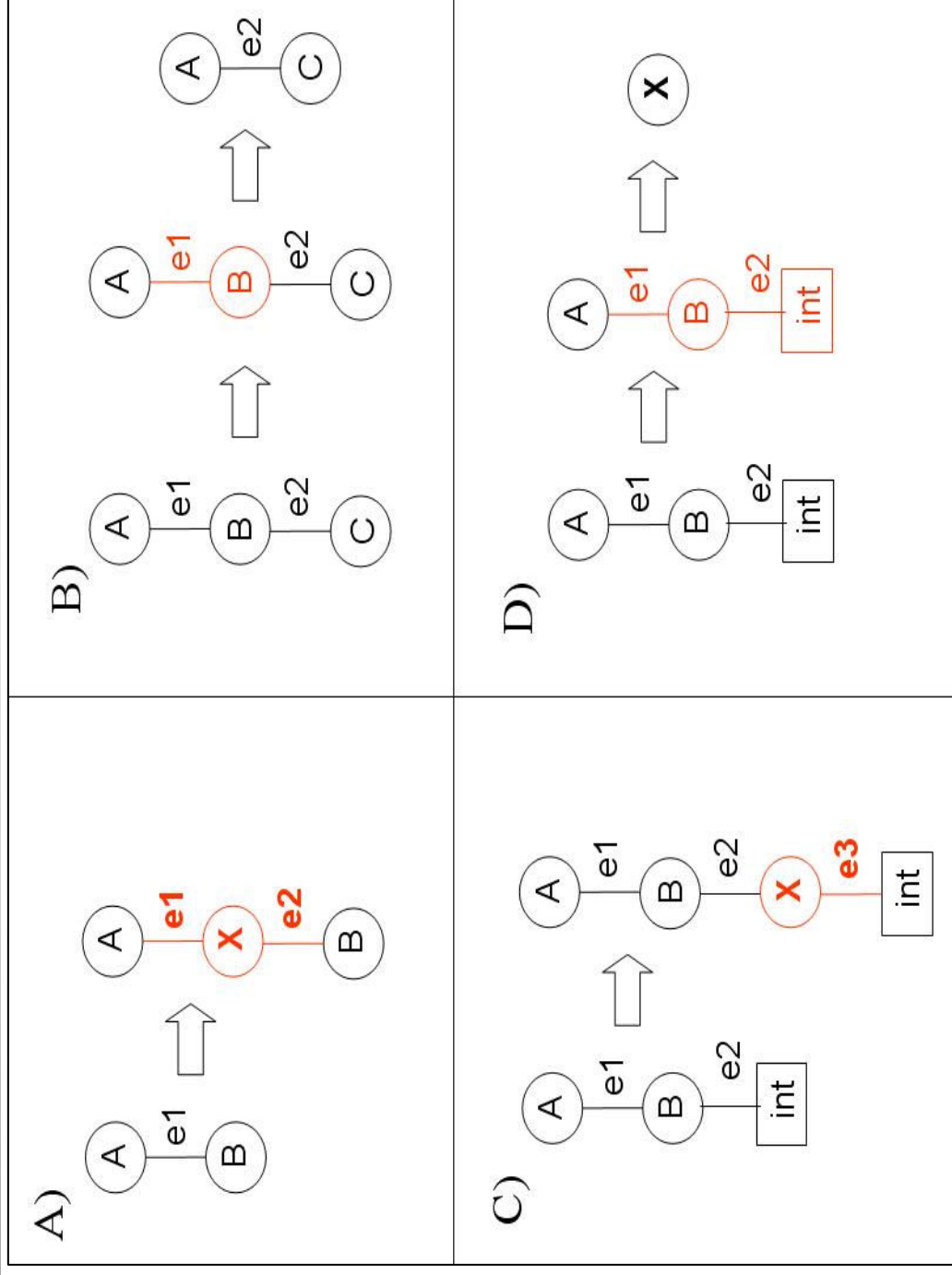
# Trabalhos Relacionados

---

- Testes de componentes Web
  - Xu et al (2005)
    - Operadores de mutação para perturbação de esquemas – documento XML Schema
    - Criar mensagens XML incorretas
    - 7 operadores
      - 4 primitivos: InsertN, DeleteN, InsertND, DeleteND
      - 3 não-primitivos: InsertT, DeleteT, ChangeE

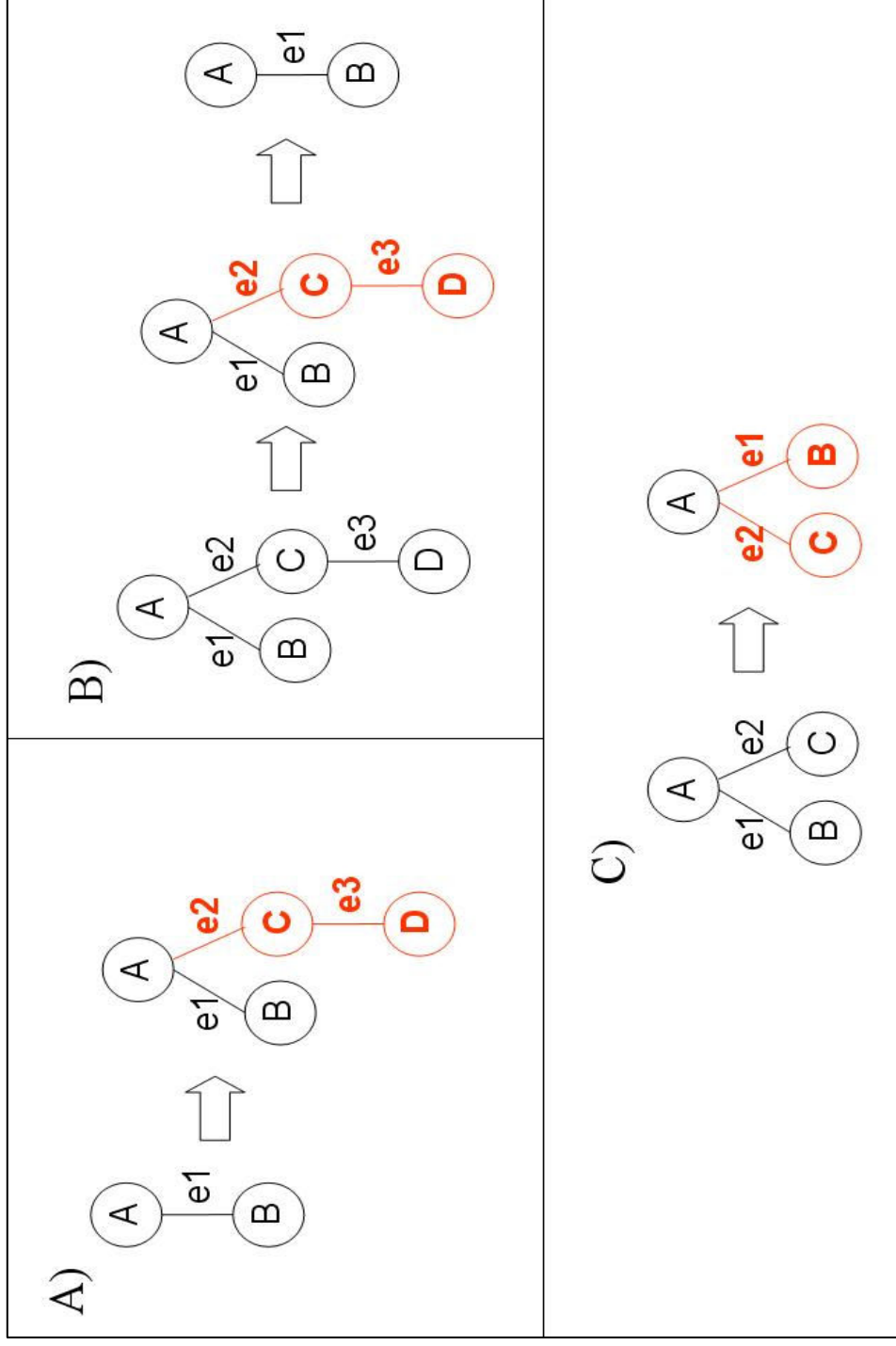
# Trabalhos Relacionados

## Operadores Primitivos



# Trabalhos Relacionados

## Operadores Primitivos





# Trabalhos Relacionados

---

- Testes de documentos XML Schema
  - Li e Miller (2005)
    - 18 operadores
    - Alteram o valor ou algum atributo
    - Não diferencia os mutantes, não aplica o critério.



# Trabalhos Relacionados

---

- Testes de documentos XML Schema
  - Emer et al (2005)
    - XQUERY
    - Revelar defeitos com base na especificação
    - Definição de 3 classes de defeitos: Elemento, Atributo e Restrições de dados
    - Gera Documentos XML mutantes
    - Consultas (queries) para cada documento XML mutante



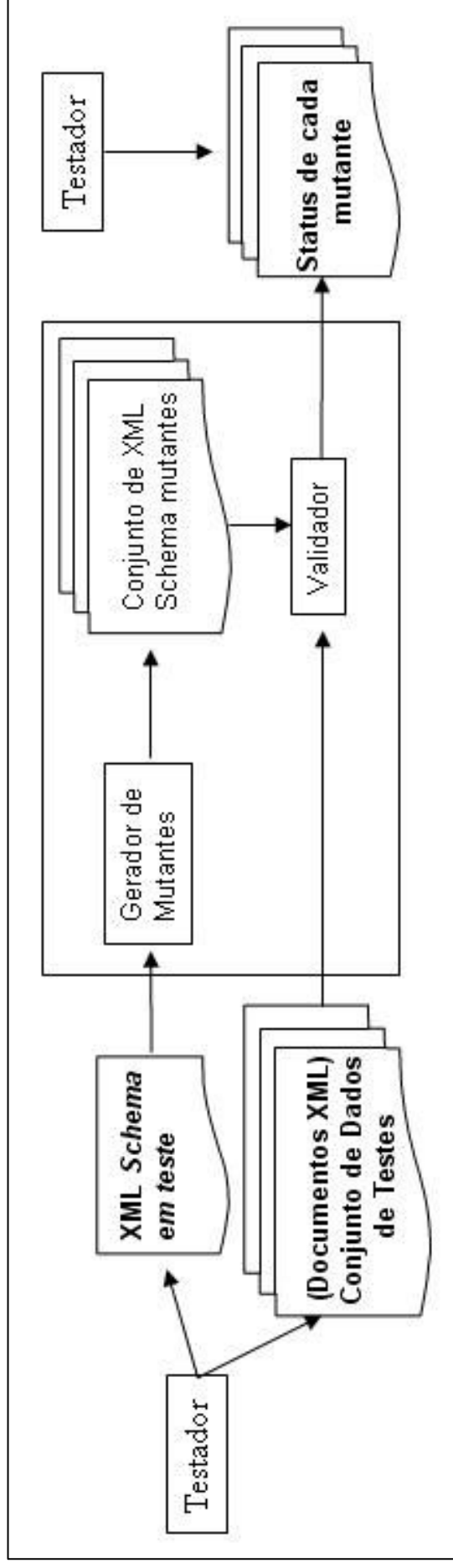
# Operadores de Mutação

---

- 4 passos foram necessários:
  - Estudar os principais erros cometidos ao se projetar esquemas
  - Definir operadores de mutação
  - Definir um processo de aplicação dos operadores
  - Implementar uma ferramenta de suporte
- Rica estrutura e vocabulário do documento XML Schema
- Objetivo: Validar a semântica do documento XML

# Operadores de Mutação

## Processo de Teste







# Operadores de Mutação

---

- 2 conjuntos de operadores
  - Manipulação de Dados
    - 7 operadores
  - Manipulação da Estrutura
    - 3 operadores



# Operadores de Mutação

---

- Manipulação de Dados
  - **Group\_Order (GO)**: Altera a ordem na qual os elementos devem estar no grupo

*Exemplo:*

```
<group order= ONE >
```

*Mutações possíveis:*

```
<group order = SEQ>
```

```
<group order= CHOICE>
```

```
<group order= MANY>
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **Required (REQ)**: Altera o tipo da obrigatoriedade dos atributos

*Exemplo:*

```
<attribute type="bar"
dt:type="int" required="YES" />
```

*Mutação possível:*

```
<attribute type="bar"
dt:type="int" required="NO" />
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **DataTypes (DT)**: Altera o tipo dos elementos/atributos

*Exemplo:*

```
<attribute type name="bar"
dt:type="INT" required="yes" />
```

*Mutações possíveis:*

```
<attribute type name = "bar"
dt:type="STRING" required="yes" />;
<attribute type name="bar"
dt:type="FLOAT" required="yes" />
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **LengthOf (LO)**: Altera o tamanho do nome dos elementos.

*Exemplo:*

```
<element type="A">
```

*Mutação possíveis:*

```
<element type="AXXX">
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **ChangeSingPlural (CSP)**: Altera o tamanho do elemento adicionando ou removendo o caracter `s` no final da string.

*Exemplo 1:*

```
<element name="A" >
```

*Mutação possível:*

```
<element name="AS" >
```

*Exemplo 1:*

```
<element type="AS" >
```

*Mutação possíveis:*

```
<element type="A" >
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **ChangeTag (CTP):** Altera as *tags* dos nós.

*Exemplo:*

```
<attribute type name="bar"
dt:type="int" required="yes" />
```

*Mutação possível:*

```
<ELEMENT type name="bar" dt:type="int"
required="yes" />
```



# Operadores de Mutação

---

- Manipulação de Dados
  - **SizeOccurs (SO)**: Altera os tamanhos das ocorrências Máximas e/ou Mínimas tanto dos tipos fixos (*strings*) quanto dos tipos criados pelo usuário.

*Exemplo:*

```
<element type="B" minOccurs="1">
```

*Mutação possível:*

```
<element type="B" minOccurs="0">
```





# Operadores de Mutação

- Manipulação da Estrutura
  - InsertTree

```
<group order="one">  
  <element type="X">  
  <element type="Y">  
  <ELEMENT TYPE="TESTE">  
  
</group>  
<group order="seq">  
  <ELEMENT TYPE="B" MINOCCURS="1">  
  <ELEMENT TYPE="C">  
  <ELEMENT TYPE="A">  
  
</group>
```



# Operadores de Mutação

- Manipulação da Estrutura
  - RemoveTree

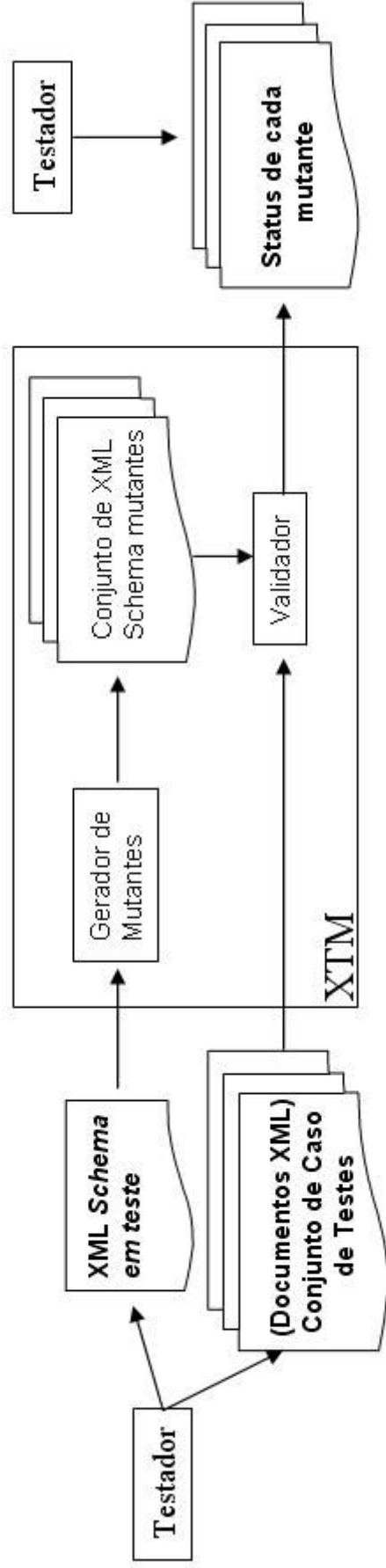
```
<group order="one">  
  <#ELEMENT TYPE="A" #>  
  <element type="Y">  
</group>  
<group order="seq">  
  <ELEMENT TYPE="X">  
  <element type="B" minOccurs="1">  
  <element type="C">  
</group>
```

# Operadores de Mutação

- Manipulação da Estrutura
  - SubTree\_Exchange

Exemplo 1: Elementos invertidos entre 2 nós distintos	Exemplo 2: Elementos invertidos dentro do próprio nó cuja order é SEQ
<pre>&lt;group order="one"&gt; &lt;ELEMENT TYPE="A"&gt; &lt;element type="Y"&gt; &lt;/group&gt; &lt;group order="seq"&gt; &lt;ELEMENT TYPE="X"&gt; &lt;element type="B" minOccurs="1"&gt; &lt;element type="C"&gt; &lt;/group&gt;</pre>	<pre>&lt;group order="one"&gt; &lt;element type="X"&gt; &lt;element type="Y"&gt; &lt;/group&gt; &lt;group order="seq"&gt; &lt;ELEMENT TYPE="B" MINOCCURS="1"&gt; &lt;ELEMENT TYPE="C"&gt; &lt;ELEMENT TYPE="A"&gt; &lt;/group&gt;</pre>

# Ferramenta XTM





# Ferramenta XTM

```
C:\WINNT\system32\CMD.exe
C:\XTM>java -cp . XTM aluno aluno.xml

    XTM - Tool for XML Schema Testing based on Mutation

XML Schema: ..... \schema\aluno.xsd
Documento XML: ..... \schema\aluno.xml

XML Schema is alive: .. \schema\aluno.xsd
Validade do documento XML: ..... True

*****
Altera Dados XML Schema
*****

XML Schema is alive: .. \schema\aluno12.xsd
XML Schema is alive: .. \schema\aluno112.xsd
XML Schema is alive: .. \schema\aluno118.xsd
XML Schema is alive: .. \schema\aluno122.xsd
XML Schema is alive: .. \schema\aluno123.xsd
XML Schema is alive: .. \schema\aluno130.xsd
XML Schema is alive: .. \schema\aluno136.xsd
XML Schema is alive: .. \schema\aluno140.xsd
XML Schema is alive: .. \schema\aluno141.xsd
XML Schema is alive: .. \schema\aluno155.xsd

Quantidade de Mutantes Criados: ..... 51
Quantidade de Mutantes vivos: ..... 10

C:\XTM>
```



# Experimentos

---

- 3 sistemas – 6 esquemas
  - Matrículas, Biblioteca: os mesmos utilizados por Emer et al
  - Hospitalar: aplicação real
- 2 experimentos foram conduzidos
  - Utilização de validadores para identificar os defeitos inseridos – o mesmo realizado por Li e Miller
  - Utilização de casos de testes para identificar os defeitos existentes com relação as especificações dos esquemas



# Experimento 1

---

- O mesmo experimento realizado por Li e Miller
- Passos
  - Gerar mutantes utilizando todos os operadores implementados pela XTM.
  - Validar os esquemas mutantes utilizando as ferramentas:
    - Validador da W3C
    - *XMLSpy*
    - *Stylus Studio 6*



# Experimento 1

Oper	XML Schema							Total
	Aluno	Disciplina	Obras	Usuário	Sia	Sih		
SNC	1	1	1	1	1	1	6	
QCR	1	1	1	1	1	1	6	
AOC	0	0	0	0	0	0	0	
QIR	0	0	0	0	0	0	0	
CCR	4	4	6	4	16	16	50	
UCR	0	0	0	0	0	0	0	
SLC	0	0	0	0	0	0	0	
RAR	0	0	0	0	3	3	6	
NCC	0	0	0	0	0	0	0	
DTC	0	0	0	0	1	1	2	





# Experimento 1

Oper	XML Schema							Total
	Aluno	Disciplina	Obras	Usuário	Sia	Sih	Total	
GO	0	0	0	0	0	0	0	0
CSP	10	10	11	8	36	34	109	109
REQ	0	0	0	0	0	0	0	0
DT	12	12	10	8	52	48	142	142
SO	23	20	32	25	80	75	255	255
LO	10	10	11	8	35	36	110	110
IT	22	28	28	20	80	83	261	261
STE	28	28	28	15	77	96	271	271
RT	4	15	15	12	54	57	157	157
<b>Total</b>	<b>115</b>	<b>129</b>	<b>143</b>	<b>102</b>	<b>436</b>	<b>451</b>	<b>1376</b>	<b>1376</b>



# Experimento 1

<b>XML Schema</b>	<b>% Média de Mutantes Válidos</b>					
	<b>Dados</b>	<b>SBT</b>	<b>IT</b>	<b>RT</b>	<b>TOTAL</b>	
Aluno	60.10	100	32,14	100	68,24	
Disciplina	68.39	100	14,29	100	72,83	
Obras	68.12	100	14,29	100	71,97	
Usuário	67.23	100	15	100	69,15	
Sia	65.57	100	12,50	96,3	69,62	
Sih	59.43	100	14,46	96,49	74,62	
Total	64.80	100	17,11	98,8	71,02	



# Experimento 1

---

- Operador InsertTree gera muitos mutantes não válidos
- Operadores Propostos x Operadores propostos por Li e Miller
- Documentos complexos -> maior abrangência da aplicação dos operadores



# Experimento 2

---

- Objetivos do experimento
  - Aplicação do critério Análise de Mutantes
  - Quantidade de dados de testes necessários
  - Número de esquemas equivalentes gerados
  - Número de defeitos encontrados
  - *Score* de Mutação



# Experimento 2

---

- Passos
  - Conjunto inicial de dados de teste que deverão ser validados pelo esquema mutante
  - Documentos XML adicionais são gerados, cajo necessário
  - Calculado *score* de mutação



# Experimento 2

<b>XML Schema</b>	<b>Qtd mutantes válidos</b>	<b>Qtd Dados de Testes</b>	<b>Qtd Defeitos Revelados /Existentes</b>
<b>Aluno</b>	60	3	10/10
<b>Disciplina</b>	79	4	8/8
<b>Obras</b>	95	3	4/4
<b>Usuário</b>	66	3	9/9
<b>Sia</b>	250	4	0/0
<b>Sih</b>	268	4	1/1

*Score de Mutação = 1*  
*Mutantes Equivalentes = 0*



# Experimento 2

---

- Todos os defeitos conhecidos (naturais e semeados) foram identificados
- Mais eficiente do que apenas o uso de validadores
- Trabalho de Emer et al
- Quase todos os defeitos foram revelados usando os operadores propostos
- Um defeito foi revelado usando o operador RAR definido por Li e Miller



# Conclusão

---

- Processo de Teste usando o critério Análise de Mutantes
- Conjunto de Operadores definidos com base nos erros mais comuns
- Implementação da ferramenta XTM
- Defeitos semânticos revelados
- Geração de Mutantes - > custoso





# Trabalhos Futuros

---

- Aperfeiçoamento da ferramenta
- Redefinição de alguns operadores
  - O operador InsertTree deve considerar o contexto da inserção
- Definição de novos operadores
- Condução de novos experimentos

Obrigada

