ALESSANDREIA MARTA DE OLIVEIRA

DIFF SEMÂNTICO DE DOCUMENTOS XML

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor. Área de Concentração: Engenharia de Sistemas e Informação.

Orientadora: VANESSA BRAGANHOLO MURTA
Co-Orientador: LEONARDO GRESTA PAULINO MURTA

Niterói
2016

ALESSANDREIA MARTA DE OLIVEIRA

DIFF SEMÂNTICO DE DOCUMENTOS XML

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor. Área de Concentração: Engenharia de Sistemas e Informação.

Aprovada em dezembro de 2016.

BANCA EXAMINADORA

_____

Profa. Vanessa Braganholo Murta – Orientadora – UFF

_____

Prof. Leonardo Gresta Paulino Murta – Co-Orientador – UFF

_____

Prof. Renata de Matos Galante – UFRGS

_____

Prof. Jonice de Oliveira Sampaio – UFRJ

_____

Prof. Alexandre Plastino de Carvalho – UFF

_____

Prof. Daniel Cardoso Moraes de Oliveira – UFF

Niterói

2016

RESUMO

Documentos XML têm sido utilizados cada vez mais para permitir a troca de dados entre sistemas. Um problema relacionado é que os documentos XML evoluem ao longo do tempo e identificar e compreender estas mudanças torna-se fundamental. Contudo, abordagens existentes, relacionadas à compreensão de mudanças (*diff*) em documentos XML, têm seu foco somente em mudanças de cunho sintático. Isto significa que estas abordagens realizam comparações entre os documentos XML baseadas nas suas estruturas, sem considerar a semântica associada. Em documentos XML grandes, que passaram por muitas modificações entre suas versões, poucas mudanças semânticas podem englobar muitas mudanças sintáticas. Essa característica tem potencial para contribuir com a facilidade de compreensão de mudanças em documentos XML. Diante disto, esta tese apresenta a abordagem XChange, que visa prover suporte ao *diff* de documentos XML considerando a semântica das modificações efetuadas. Para tal, as modificações sintáticas granulares em atributos e elementos são analisadas por meio de regras de enriquecimento semântico. Essa análise identifica conjuntos de modificações sintáticas de elementos correspondentes que compõem modificações semânticas. Diferente das abordagens existentes, o XChange propõe a utilização das mudanças sintáticas entre versões de um documento XML como meio para inferir a razão real das mudanças e apoiar o processo de *diff* semântico. De forma a garantir o funcionamento correto do *diff* semântico, é importante avaliar a qualidade dos casamentos de elementos correspondentes efetuados. Os resultados experimentais mostram que o XChange foi capaz de fornecer resultados com precisão equivalente, com apenas uma fração do tempo necessário, quando comparado ao estado da arte, sendo, portanto, uma abordagem mais eficiente. Além disso, os resultados mostram que o XChange é mais eficaz e eficiente na compreensão das mudanças entre versões de documentos XML, quando também comparado ao estado da arte.

Palavras-chave: *Diff* Semântico, Casamento, Similaridade, Evolução de documentos XML.

ABSTRACT

XML documents are increasingly being used to allow data interchange among systems. A related problem is that XML documents evolve over time, so identifying and understanding these changes becomes fundamental. However, existing approaches related to understanding changes (diff) in XML documents are focused only on syntactic changes. This means that these approaches compare XML documents based on their structures, without considering the associated semantics. For large XML documents, which have undergone many changes from a version to the next, fewer semantic changes might encompass many syntactic changes. This feature has the potential to ease the understanding of changes in XML documents. Thus, this proposal presents the XChange approach to provide support for diff of XML documents considering the semantic of the changes. For such, the granular syntactic changes in attributes and elements are analyzed by means of inference rules. This analysis identifies sets of syntactic changes that can be combined into semantic changes. Thus, differently from existing approaches, XChange proposes the use of syntactic changes in versions of an XML document as a means to infer the real reason for the change and support the process of semantic diff. In order to ensure the correct functioning of the semantic diff, it is important to evaluate the quality of the matching among elements. The experimental results show that XChange was able to provide equivalent precision results, with only a fraction of the time needed by a state-of-the-art approach, being a more efficient approach. Furthermore, the results show that XChange presents higher efficacy and efficiency in understanding changes between versions of XML documents, when compared to a state-of-the-art approach.

Keywords: Semantic diff, Match, Similarity, Evolution of XML documents.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

1.1 MOTIVAÇÃO

Diversos sistemas se apoiam cada vez mais na linguagem XML - *eXtensible Markup Language* (BRAY *et al.*, 2008) para representar dados semiestruturados e, consequentemente, uma grande quantidade de documentos XML é gerada. Muitas indústrias e comunidades científicas adotaram documentos XML como um padrão para representação, armazenamento e troca de dados. Dentre elas, é possível citar aplicações em *Web Science* (GETOV, 2008), na área de saúde (ARGÜELLO *et al.*, 2009; THUY; LEE; LEE, 2013), no poder legislativo (HALLO CARRASCO; MARTÍNEZ-GONZÁLEZ; DE LA FUENTE REDONDO, 2013) e na indústria de varejo (MORO; BRAGANHOLO, 2009). Além disso podem ser citados a plataforma Lattes (CNPQ, 2017), o Portal Brasileiro de Dados Abertos, que contém dados dados publicados pelos órgãos do governo relacionados a saúde suplementar, sistema de transporte, segurança pública, indicadores de educação, gastos governamentais, processo eleitoral, entre outros (PORTAL BRASILEIRO DE DADOS ABERTOS, 2017), e a Wikipedia, que periodicamente gera arquivos de *dumps* dos dados gerenciados por ela: artigos, imagens, categorias, restrições e outros metadados (WIKIMEDIA, 2017).

Um problema relacionado é que, assim como os dados armazenados em bancos de dados estruturados, os dados semiestruturados evoluem ao longo do tempo, em função, por exemplo, de alterações de cunho técnico. Se por um lado o uso de estrutura hierárquica e marcadores definidos pelo usuário (BRAY *et al.*, 2008) permite flexibilidade na representação dos dados, por outro lado dificulta o acompanhamento de sua evolução, principalmente em grandes repositórios.

Para exemplificar o problema, considere o documento XML referente ao cadastro de funcionários da prefeitura de Baltimore (MAYOR'S OFFICE OF INFORMATION TECHNOLOGY, 2016) que é utilizado no decorrer desta tese. Esse documento XML contém as seguintes informações: nome do funcionário (<*name*>), cargo (<*jobtitle*>), código da agência (<*agencyid*>), nome da agência (<*agency*>), data de admissão (<*hiredate*>), salário anual (<*annualsalary*>) e salário bruto (<*grosspay*>). O documento XML está organizado em três níveis de profundidade representados por <*governement*>, <*employee*> e pelas informações de cada funcionário. A Figura 1.1 apresenta um pequeno fragmento desse documento com quatro funcionários na versão *v1* e as informações relacionadas a eles.

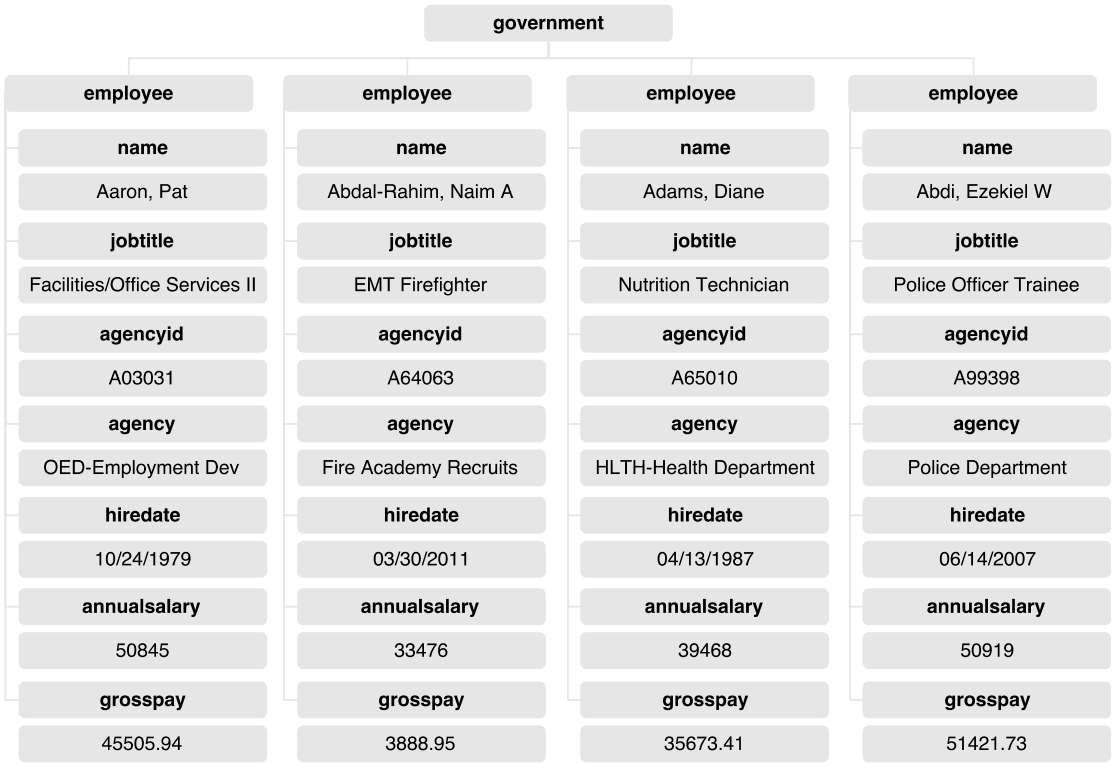

Figura 1.1: Documento XML do cadastro de funcionários na versão *v1*

Já a Figura 1.2 apresenta cinco funcionários na versão *v2* desse fragmento do documento, que é uma revisão da versão *v1*.

Figura 1.2: Documento XML do cadastro de funcionários na versão *v2* com marcação de diferenças em comparação com *v1* (verde representando inclusão, amarelo representando alteração e vermelho representando remoção)

Como pode ser observado, existem três funcionários que estão presentes em *v1* e *v2*, destacados em amarelo. Pela análise de *v1* e *v2*, estes funcionários apresentam mudanças em alguns dos seus elementos. O funcionário *Aaron, Pat*, por exemplo, teve uma alteração em seu nome bem como em seu salário anual e salário bruto (mudanças destacadas em amarelo). Pode-se observar ainda que existe um funcionário destacado em verde, presente somente em *v2*, o que indica que este funcionário foi inserido em *v2*. Existe também um funcionário destacado em vermelho em *v2*, o que indica que ele foi excluído em *v2*. A versão *v2* é, portanto, consequência de uma demanda relacionada à evolução dos dados dos funcionários envolvidos.

Essas mudanças são facilmente identificadas após uma análise das duas versões deste pequeno fragmento de documento XML, mesmo sem a notação de cores apresentada na Figura 1.2. No entanto, no caso de uma empresa com um número de funcionários considerável (por exemplo, o documento original da prefeitura de Baltimore – versão *v1* contém 13966 funcionários), esta tarefa não seria trivial. Para acompanhar as mudanças em documentos XML, é necessária uma forma de detectar exatamente quais são as diferenças entre duas versões do documento, ou seja, o que foi modificado de uma versão para a outra em termos das estruturas que compõem um documento XML (elementos e atributos).

Na literatura este problema já vem sendo estudado há algum tempo (CHAWATHE; GARCIA-MOLINA, 1997; LIM; NG, 2001; MARIAN *et al.*, 2001; COBENA; ABITEBOUL; MARIAN, 2002; XU *et al.*, 2002; JACOB; SACHDE; CHAKRAVARTHY, 2003, 2005; WANG; DEWITT; CAI, 2003; LINDHOLM, 2004; ZHAO; BHOWMICK; MADRIA, 2004; AL-EKRAM; ADMA; BAYSAL, 2005; RUSU; RAHAYU; TANIAR, 2006; SANTOS; HARA, 2007; SONG; BHOWMICK; DEWEY, JR., 2007; RÖNNAU; PHILIPP; BORGHOFF, 2009; THAO; MUNSON, 2010; SUNDARAM; MADRIA, 2012). Diferentemente das abordagens tradicionais de *diff* textual, que consideram linhas dos arquivos texto como elementos atômicos na comparação, essas abordagens são cientes da sintaxe dos documentos XML. Desta forma, elas são capazes de comparar os elementos e os atributos de um documento XML, mesmo que eles estejam todos em uma mesma linha do arquivo.

Para efetuar a comparação e a identificação dos elementos correspondentes entre as versões, algumas destas abordagens utilizam cálculo de similaridade (DORNELES *et al.*, 2009), enquanto outras usam chaves de contexto (BUNEMAN *et al.*, 2002, 2003), que podem ser expressas no esquema do documento (FALLSIDE; WALMSLEY, 2004). O XyDiff (COBENA; ABITEBOUL; MARIAN, 2002), por exemplo, usa o conceito de chave para a identificação dos elementos correspondentes. Um problema relacionado é que as chaves não

são mantidas em todas as situações. Dependendo de como os documentos XML são gerenciados, não há garantia de que os valores permanecem os mesmos entre as versões. Além disso, os esquemas não são obrigatórios, por isso a maioria dos documentos XML não os utiliza (MAAROUF; CHUNG, 2008; VYHNANOVSKÁ; MLÝNKOVÁ, 2010; GRIJZENHOUT; MARX, 2013).

Outro problema associado é que se as versões do documento XML sofrem muitas modificações, torna-se difícil a compreensão desta evolução. Ou seja, saber a razão por trás das ações de inserção, remoção e atualização dos elementos e atributos do documento não é trivial. Vale mencionar também que várias modificações granulares podem se referir em conjunto a uma modificação maior no nível semântico. Por exemplo, após uma análise da Figura 1.1 e da Figura 1.2, verifica-se que quatro modificações sintáticas referentes ao funcionário *Abdal-Rahim, Naim A* (linha 2 da Tabela 1.1) têm como propósito permitir a *promoção e transferência* desse funcionário. As abordagens relacionadas a *diff* de documentos XML existentes, inclusive, o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002) e o X-Diff (WANG; DEWITT; CAI, 2003), que são as duas abordagens da literatura mais citadas e que são base para diversas outras propostas, analisam apenas os aspectos sintáticos, não levando em conta a semântica de modificações, que são dependentes do domínio de conhecimento. Diante disso, esta intenção de mudança não seria identificada por tais abordagens.

Tabela 1.1: Elementos alterados de *v1* para *v2*

	Funcionário	Modificações
1	Aaron, Patricia G	*name, annualsalary, grosspay*
2	Abdal-Rahum, Naim A	*agencyid, agency, annualsalary, grosspay*
3	Abdi, Ezekiel W	*annualsalary, grosspay*

De fato, existem aplicações onde a informação sintática não é suficiente, ou seja, situações onde não basta detectar os elementos ou os atributos que mudaram, mas também é necessário inferir a razão das modificações. No cenário da Figura 1.1 e da Figura 1.2 pode ser necessário verificar quando novos funcionários foram contratados (presentes somente em *v2*) ou ainda o salário de um funcionário em uma determinada data, o que estaria explícito no nível sintático. Contudo, o usuário pode estar interessado em verificar quando ocorreu a promoção de determinados funcionários. Se essa informação não existir explicitamente no documento, a sua percepção é dificultada se forem utilizadas abordagens de detecção de mudanças puramente sintáticas (MENS, 2002). Em outras palavras, um conjunto de modificações na estrutura do documento pode corresponder a uma modificação semântica, que deveria ser automa-

ticamente identificada (*diff* semântico), o que não ocorre com as abordagens mencionadas com foco no *diff* sintático.

1.2 OBJETIVOS

Diante dos problemas mencionados, o objetivo desta tese é apresentar o XChange, uma abordagem de *diff* semântico de documentos XML. Para tanto, a sintaxe das versões do documento XML é considerada e os elementos correspondentes nas duas versões são identificados. Neste processo de identificação, duas estratégias podem ser utilizadas: casamento por chave e casamento por similaridade. A partir da identificação dos elementos correspondentes, é possível apontar as diferenças sintáticas. Essas diferenças sintáticas são então incorporadas em uma base de conhecimento e enriquecidas com regras semânticas pertencentes ao domínio de conhecimento do documento XML. Finalmente, a partir de inferências na base de conhecimento, é possível obter a diferença semântica entre as duas versões do documento XML.

Em suma, o propósito do XChange consiste em apoiar a compreensão da evolução, no que diz respeito ao *diff* semântico, viabilizando a identificação da razão real das modificações em documentos XML. Para avaliar se o XChange atingiu os seus objetivos, foram realizados dois estudos experimentais (E1 e E2) visando responder às seguintes questões de pesquisa (QP):

E1-QP1. O método baseado em cálculo de similaridade do XChange é mais eficaz na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

E1-QP2. O método baseado em cálculo de similaridade do XChange é mais eficiente na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

E2-QP1. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficaz do que a identificação de alterações sintáticas, utilizada pelo X-Diff?

E2-QP2. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficiente do que a identificação de alterações sintáticas, utilizada pelo X-Diff?

1.3 METODOLOGIA

Inicialmente foi construído um mapeamento sistemático da literatura, com o objetivo de detectar as principais características presentes em abordagens de *diff* considerando o estado da arte. O estudo identificou que o problema relacionado ao gerenciamento de mudanças em documentos XML já vem sendo estudado há algum tempo. Porém, o foco dessas abordagens está no *diff* sintático entre as versões de um documento XML. Contudo, ao analisar o resultado de um *diff* se deseja inferir a razão das modificações, o que nos motivou estudar mecanismos de *diff* semântico para XML.

Após a análise da literatura, foi especificada a abordagem XChange. A partir de um mecanismo de inferência e de um conjunto de regras definidas pelo usuário especialista, o XChange enriquece o *diff* sintático, possibilitando deduzir a intenção do usuário ao efetuar as mudanças (resultados).

A abordagem XChange foi implementada utilizando a linguagem Java, visando criar um ambiente integrado onde o usuário pode monitorar as mudanças em diferentes versões de um documento XML. A biblioteca tuProlog (DENTI; OMICINI; RICCI, 2001) foi incorporada para apoiar a inferência. Foi implementado também um apoio semiautomático para a definição de regras de enriquecimento semântico baseado em *itemsets* frequentes, através do algoritmo Apriori (AGRAWAL; SRIKANT, 1994). Além disso, foi definida uma interface gráfica que apresenta as regras de enriquecimento semântico geradas pelo apoio semiautomático, para que o especialista possa finalizar a definição destas regras.

Após a construção da ferramenta, foi realizado um estudo experimental com o intuito de avaliar a eficácia e a eficiência do XChange no que diz respeito ao casamento de elementos correspondentes entre versões de um documento XML, estratégia essa que apoia a construção do *diff*. No critério eficácia, os resultados do XChange não possuem diferença estatisticamente significativa quando comparados aos resultados obtidos pelo X-Diff (WANG; DEWITT; CAI, 2003). Os resultados mostraram ainda que a estratégia utilizada pelo XChange, quando comparada ao X-Diff (WANG; DEWITT; CAI, 2003), é mais eficiente (quase 45 vezes mais), efetuando a identificação correta de um número maior de elementos correspondentes por unidade de tempo.

Além disso, foi realizado um segundo estudo experimental visando caracterizar o apoio existente à compreensão da evolução de documentos XML através da análise do *diff* gerado pelo XChange, em comparação com os resultados obtidos pelo X-Diff (WANG; DEWITT; CAI, 2003). Foi possível observar que o XChange é mais eficaz e mais eficiente

que o X-Diff no que se refere a compreensão das mudanças, razão original de concepção do XChange.

1.4 CONTRIBUIÇÕES

Diante do exposto, pode-se mencionar como contribuições desta tese:

- a definição da abordagem XChange para apoiar a compreensão da evolução de documentos XML, viabilizando a identificação da razão real das modificações, levando em consideração a semântica associada;
- a identificação de elementos correspondentes em versões de documentos XML a partir de duas estratégias: casamento por chave e casamento por similaridade;
- a construção de uma ferramenta para efetuar o *diff* semântico de documentos XML, que inclui um apoio semiautomático para a definição de regras de enriquecimento semântico baseado em *itemsets* frequentes e uma interface gráfica para finalizar a definição e/ou criar novas regras a partir de uma seleção de opções em alto nível.

O XChange, proposta principal dessa tese, apresentou, a partir de avaliações experimentais, os seguintes resultados:

- eficácia equivalente ao estado da arte na identificação de elementos correspondentes;
- cerca de 45 vezes mais eficiência que o estado da arte na identificação de elementos correspondentes;
- maior eficácia e eficiência que o estado da arte na compreensão das mudanças entre versões de documentos XML.

1.5 ORGANIZAÇÃO

Este documento está organizado em outros cinco capítulos, além desta introdução. O Capítulo 2 apresenta conceitos relacionados a documentos XML e alguns fundamentos de *diff*. Além disso, mostra os resultados do mapeamento sistemático relacionado a *diff* de documentos XML. O Capítulo 3 apresenta o XChange. Inicialmente, uma visão geral da abordagem é apresentada. As seções seguintes descrevem as etapas da abordagem que tem como objetivo permitir ao usuário identificar e compreender as mudanças semânticas ao analisar versões de um documento XML. O Capítulo 4 descreve o estudo experimental relacionado ao casamento de elementos correspondentes, onde a eficácia e a eficiência do método utilizado pelo XChange são analisadas e comparadas com o X-Diff (WANG; DEWITT; CAI, 2003) e

com o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002). O Capítulo 5 descreve o estudo experimental relacionado à compreensão da evolução de documentos XML, onde a eficácia e a eficiência do *diff* semântico do XChange são analisadas e comparadas com os resultados obtidos pelo *diff* sintático do X-Diff (WANG; DEWITT; CAI, 2003). São apresentados os resultados também em termos de eficácia e eficiência. Finalmente, o Capítulo 6 discute as conclusões para este trabalho e as sugestões de trabalhos futuros.

CAPÍTULO 2 – EVOLUÇÃO DE DOCUMENTOS XML

2.1 INTRODUÇÃO

Documentos XML são cada vez mais utilizados para a representação de dados na *Web* (BRAY *et al.*, 2008). Além disso, este tipo de documento tornou-se padrão para integração de sistemas e comunicação entre aplicações diferentes (MORO; BRAGANHOLO, 2009). É comum que estes documentos sofram muitas mudanças ao longo do tempo, tornando necessária a definição de um processo de compreensão da evolução dos mesmos. Gerenciar manualmente as mudanças ocorridas, além de ser um processo custoso, é propício a erros.

Diante disso, abordagens que apoiem a compreensão da evolução destes documentos através de ferramentas específicas de detecção de diferenças (*diff*) tornam-se necessárias. Tais iniciativas são abordadas neste capítulo que está organizado como segue. Na Seção 2.2 uma visão geral relacionada a documentos XML é apresentada. A Seção 2.3 apresenta alguns conceitos importantes relacionados a *diff*. A Seção 2.4 aponta as abordagens relacionadas a *diff* de documentos XML. Na Seção 2.5, algumas discussões sobre as abordagens relacionadas e as considerações finais do capítulo são apresentadas.

2.2 FUNDAMENTOS DE XML

XML (*eXtensible Markup Language*) é uma linguagem de marcação de dados, derivada da SGML (*Standard Generalized Markup Language*), que surgiu em 1996. É uma recomendação do W3C (*World Wide Web Consortium*) para especificação de dados semiestruturados. Seu foco é a descrição do conteúdo e não sua forma de apresentação. No restante desta seção são apresentados alguns conceitos relacionados a XML. Tais conceitos são baseados nos trabalhos de Fallside e Walmsley (2004), Bray *et al.* (2008), Moro e Braganholo (2009). Já os exemplos utilizados são baseados no cenário de cadastro de funcionários da prefeitura de Baltimore apresentado na Seção 1.1.

Um documento XML é formado por uma sequência de elementos que englobam valores texto e subelementos, além de atributos. Para exemplificar, a Figura 2.1 mostra um fragmento do documento XML relacionado à prefeitura de Baltimore. Os elementos são representados com marcações (*tags*), como, por exemplo, <*employee*> (linha 2). Ao contrário da HTML (*HyperText Markup Language*), XML não apresenta um conjunto limitado e predefinido de *tags* a serem utilizadas. As *tags* podem ser definidas de acordo com o significado do

dado que se quer representar. Para o cargo, pode-se usar, por exemplo, a *tag* <*jobtitle*>, como mostra a linha 4. Cada marcação < > e </ > delimita um elemento XML.

```
1  <government>
2   <employee>
3    <name>Bond,Filishia M</name>
4    <jobtitle>PARALEGAL</jobtitle>
5    <agencyid phonenumber="(925)465-1961">A06019</agencyid>
6    <agency>Housing and Community</agency>
7    <hiredate>2001-06-25T00:00:00</hiredate>
8    <annualsalary>50364</annualsalary>
9    <grosspay>44941.01</grosspay>
10  </employee>
11  <employee>
12   <name>Bailowitz,Anne</name>
13   <jobtitle>EXECUTIVE LEVEL I</jobtitle>
14   <agencyid phonenumber="(925)467-5188">A65527</agencyid>
15   <agency>HLTH-Health Dept</agency>
16   <hiredate>2001-02-26T00:00:00</hiredate>
17   <annualsalary>119000</annualsalary>
18   <grosspay>103290.62</grosspay>
19  </employee>
20  ...
21  </government>
```

Figura 2.1: Fragmento do documento XML *government.xml*

Um documento XML pode ser representado também por uma estrutura em árvore, onde os nós correspondem a elementos, atributos ou valores texto, e as arestas representam relações de pai/filho. A Figura 2.2 mostra um trecho do documento *government.xml* da Figura 2.1 no formato de árvore. As árvores XML podem ser divididas em ordenadas e não ordenadas. Em ambos os tipos, a relação pai-filho é significativa, mas em árvores ordenadas, a ordem dos nós irmãos também é relevante (PETERS, 2005).

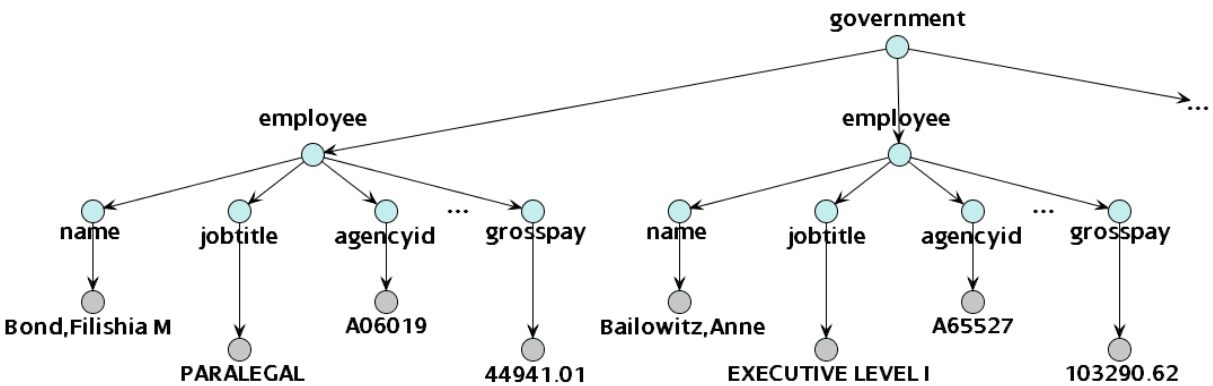

Figura 2.2: Fragmento do documento XML *government.xml* no formato de árvore

Para que possa ser manipulado como uma árvore, um documento XML precisa ser bem formado. Isto significa que um documento XML precisa atender a algumas regras, que são descritas a seguir e exemplificadas na Figura 2.1:

- raiz é única (<*government*>, linha 1 da Figura 2.1);
- todas as *tags* são fechadas;
- os elementos são bem aninhados, ou seja, a *tag* do elemento pai tem que ser fechada depois que todas as *tags* de seus filhos forem fechadas;
- os nomes de elementos são sensíveis a maiúsculas e minúsculas;
- os atributos não se repetem em um mesmo elemento. O elemento *agencyid* da Figura 2.1 (linha 5), por exemplo, é composto por um atributo denominado *phonenumber*. Não é permitido adicionar mais de um atributo *phonenumber* no mesmo elemento. Vale mencionar também que atributos aparecem dentro da marca inicial de um elemento e possuem um valor informado entre aspas (<*agencyid phonenumber="(925)465-1961">A06019</agencyid*>).

Além disso, os elementos nos documentos XML são ordenados. Se os dois elementos *employee* do documento XML da Figura 2.3.a mudarem de posição, como mostra a Figura 2.3.b, os dois documentos são considerados diferentes do ponto de vista sintático. Do ponto de vista semântico, neste domínio de conhecimento, os documentos são iguais, pois a mudança de posição não altera o significado do documento. Já os atributos do documento XML não são ordenados.

Como um dos principais propósitos de XML é a troca de informações entre aplicações, é necessário definir um vocabulário usando uma linguagem de esquema para XML adequada. Esse vocabulário é definido em um esquema que pode ser associado ao documento XML e estabelece o conjunto de *tags* que podem aparecer em um documento XML, e como elas podem aparecer. Um documento XML que, além de ser bem formado, segue as regras do esquema a que está associado, é chamado documento válido. As principais iniciativas para definir um vocabulário usando uma linguagem de esquema para XML são: a DTD (*Document Type Definition*) (BRAY *et al.*, 2008) e a XML *Schema* (FALLSIDE; WALMSLEY, 2004). A DTD define regras de formação dos elementos bem como quais elementos e atributos são válidos e em que contexto. Sua sintaxe é baseada em SGML. Diferente da DTD, a XML *Schema* é uma linguagem baseada no formato XML para definição de esquemas em documentos XML, o que significa que um esquema em XML *Schema* é um documento XML. Todas as declarações de tipos e de elementos são feitas usando a sintaxe XML.

```
1  <government>
2   <employee>
3    <name>Bond,Filishia M</name>
4    <jobtitle>PARALEGAL</jobtitle>
5    <agencyid>A06019</agencyid>
6    <agency>Housing and Community</agency>
7    ...
8   </employee>
9   <employee>
10   <name>Bailowitz,Anne</name>
11   <jobtitle>EXECUTIVE LEVEL I</jobtitle>
12   <agencyid>A65527</agencyid>
13   <agency>HLTH-Health Dept</agency>
14   ...
15  </employee>
16 ...
17 </government>
```

(a)

```
1  <government>
2   <employee>
3    <name>Bailowitz,Anne</name>
4    <jobtitle>EXECUTIVE LEVEL I</jobtitle>
5    <agencyid>A65527</agencyid>
6    <agency>HLTH-Health Dept</agency>
7    ...
8   </employee>
9   <employee>
10   <name>Bond,Filishia M</name>
11   <jobtitle>PARALEGAL</jobtitle>
12   <agencyid>A06019</agencyid>
13   <agency>Housing and Community</agency>
14   ...
15  </employee>
16 ...
17 </government>
```

(b)

Figura 2.3: Documentos XML diferentes em função da ordenação dos elementos

Há também APIs (*Application Programming Interface*) para processamento de documentos XML por meio de programas de computador. As duas principais são DOM – *Document Object Model* (HORS *et al.*, 2004) e SAX - *Simple API for XML* (XML-DEV COMMUNITY, 2002).

A API DOM é um padrão para processamento de dados XML baseado em um modelo em árvore. Foi desenvolvida pelo W3C que define uma interface para a construção e tratamento de instâncias de documentos. O *parser* constrói na memória um objeto representando a árvore XML (objeto DOM). Quando um documento é carregado na memória, suas estruturas podem ser lidas e manipuladas através do objeto DOM. O programador pode então percorrer a árvore da forma que desejar. Os mesmos conceitos de árvores, como nós, ascendentes e descendentes, profundidade da árvore, podem ser aplicados para o documento. Contudo, como o

uso do objeto DOM requer a leitura de toda estrutura XML em uma árvore na memória, pode acarretar um alto consumo de recursos.

A API SAX funciona baseada em eventos. Ela é menos flexível, porém mais eficiente. Processadores baseados nessa API funcionam disparando eventos para a aplicação a cada vez que encontram algo significativo no documento. Desta forma, a ordem em que o documento é processado é fixa, ou seja, ele sempre é processado da raiz até o fim, e cada elemento é processado uma única vez. Isso é diferente do que ocorre quando se usa a API DOM, onde se pode fazer o caminhamento como e quantas vezes for necessário, uma vez que a estrutura do documento está em memória. Por outro lado, SAX é recomendado para o caso de documentos XML maiores, bem como nos cenários onde desempenho e disponibilidade são importantes.

2.3 FUNDAMENTOS DE *DIFF*

Esta seção apresenta alguns conceitos relacionados a *diff*, baseados no trabalho de Leon (2000). Inicialmente, tem-se que uma versão representa o estado de um Item de Configuração (IC) em um determinado momento. O termo IC pode significar, por exemplo, arquivos, diretórios, entidades e relacionamentos. No contexto desta tese, um IC é um documento XML.

Uma configuração é uma versão de um objeto complexo. Ela é composta das versões das suas partes. Por exemplo, uma configuração de um sistema é composta de versões de documentos de requisitos, da arquitetura do software, do código-fonte, dos documentos XML, etc.

De acordo com o tipo de evolução que os ICs sofrem, as versões são classificadas em revisões e variantes. As versões sequenciais que evoluem ao longo do tempo são chamadas de revisões. Elas são criadas quando defeitos são corrigidos ou quando são adicionadas novas funcionalidades. As versões paralelas, ou alternativas, que coexistem, são chamadas de variantes. Enquanto as novas versões sucedem as versões mais antigas, as variantes não substituem umas às outras. Ao invés disso, elas são usadas concorrentemente em configurações alternativas. Por exemplo, variantes de estruturas de dados podem diferir em relação à eficiência ou consumo de memória, ou ainda serem destinadas a diferentes sistemas operacionais.

Considerando a existência de versões diferentes de um IC, é necessária uma forma de detectar exatamente quais são as diferenças entre elas (*diff*), ou seja, o que foi modificado de uma versão para a outra. Em alguns casos, como quando uma determinada ferramenta controla diretamente cada alteração executada, é possível que essa ferramenta armazene exatamente cada passo que foi executado para a alteração. Estas informações poderiam, portanto, ser uti-

lizadas para reconstruir uma das versões do IC a partir da outra. Entretanto, geralmente essas informações não estão disponíveis, sendo necessário detectar as diferenças entre as versões *a posteriori*. A partir dessa detecção é gerada uma representação da diferença conhecida como *delta* ou *edit script*.

Deltas representam as diferenças entre as versões e geralmente são compostos de operações que levam a construção de uma versão a partir da outra. De acordo com o objetivo, diferentes aspectos ganham ou perdem importância na representação dos *deltas*. Para aplicações que visam gerenciar as versões de um IC, o *delta* deve permitir a perfeita reconstrução de uma versão a partir de outra, de forma eficiente e eficaz. Já quando o objetivo é descrever o histórico de certos elementos ao longo do tempo, é necessário que a representação permita identificar unicamente os elementos, bem como perceber quando eles são movidos de um ponto a outro no IC, para permitir que se acompanhe todo o tempo de vida do elemento ao longo de diversas alterações. Além disso, se o objetivo é monitorar modificações, é preciso que se consiga detectar mudanças em determinados elementos e disparar eventos de acordo.

As operações representadas pelos *deltas* geralmente seguem os modelos definidos por Tai (1979) e Selkow (1977). Existem as operações padrão de inserção (I), remoção (R) e atualização (A). Algumas abordagens utilizam também as operações de movimentação (M) e cópia (C), descritas a seguir:

- inserção: um elemento foi inserido em alguma parte do documento. Uma operação é classificada como inserção quando um elemento da versão posterior não possui nenhum correspondente na versão anterior;
- remoção: consiste na exclusão de um elemento do documento. Uma operação é classificada como remoção quando um elemento da versão anterior não possui nenhum correspondente na versão posterior;
- atualização: o elemento sofreu uma atualização em seu conteúdo. Uma operação é classificada como atualização quando o conteúdo do elemento da versão anterior é diferente do conteúdo do elemento correspondente na versão posterior;
- movimentação: um elemento foi movido para outra parte do documento. Uma operação é classificada como movimentação quando a posição do elemento na versão anterior é diferente da posição de seu correspondente na versão posterior;
- cópia: um elemento foi copiado para outra parte do documento. Uma operação é classificada como cópia quando um elemento da versão anterior possui dois ou mais correspondentes na versão posterior.

A seguinte classificação é recorrentemente adotada para *diff* (MENS, 2002): *diff* físico (ou textual), *diff* sintático e *diff* semântico. O *diff* físico consiste na análise textual do IC (quando este é um arquivo texto) considerando como elemento atômico as suas linhas. O *diff* sintático está relacionado às modificações estruturais no IC, ou seja, o interesse está nas comparações baseadas nos elementos sintáticos que formam o IC (por exemplo, construtos da linguagem, no caso de código), sem considerar a semântica associada. Já o *diff* semântico (JACKSON; LADD, 1994), que faz parte desta tese, produz *deltas* semânticos e se propõe a compreender o objetivo da mudança.

2.4 *DIFF* DE DOCUMENTOS XML: ABORDAGENS EXISTENTES

Na literatura, o problema de comparação de documentos XML já vem sendo estudado há algum tempo e algumas abordagens vem sendo propostas. Tais abordagens têm seu foco no *diff* sintático, que está relacionado às modificações sintáticas nos documentos. Sendo assim, no cenário do exemplo de cadastro de funcionários, essas abordagens conseguem detectar, por exemplo, que o valor do salário de um funcionário mudou. No entanto, elas não apresentam o significado desta mudança. As abordagens apresentadas nas próximas seções foram identificadas a partir de um mapeamento sistemático da literatura descrito na Seção 2.4.1.

2.4.1 MAPEAMENTO SISTEMÁTICO DA LITERATURA

O mapeamento sistemático da literatura é um meio de descobrir, avaliar e interpretar as pesquisas disponíveis e relevantes sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse (KITCHENHAM, B., 2004). O procedimento deste mapeamento foi realizado com base em uma seleção e catalogação preliminar de artigos a partir da busca nas bibliotecas digitais Compendex[1], IEEEXplore[2] e Scopus[3].

Para definir a expressão de busca a ser utilizada nas bibliotecas digitais, foi realizado um processo de teste e refinamento baseado no método PICO (PAI *et al.*, 2004). O acrônimo PICO é usado para identificar as quatro partes críticas da expressão de pesquisa, que são população (P), intervenção (I), comparação (C) e resultado (O) (PAI *et al.*, 2004; BALDASSARRE *et al.*, 2007; KITCHENHAM, B. A.; MENDES; TRAVASSOS, 2007; MAGDALENO; WERNER; ARAUJO, 2012). A Tabela 2.1 mostra os elementos PICO deste mapeamento.

[1] http://www.engineeringvillage2.org
[2] http://ieeexplore.ieee.org
[3] http://www.scopus.com

Tabela 2.1: Elementos PICO

(P)	Publicações relacionadas a documentos XML
(I)	Publicações relacionadas a *diff*
(C)	Não há nenhum, porque o objetivo é caracterizar as abordagens
(O)	Abordagens, técnicas, algoritmos, métodos e metodologias de detecção de diferenças de documentos XML

Na etapa seguinte, foi instanciada a população e a intervenção definindo expressões de busca para cada uma delas. Como a comparação não é aplicável a este estudo, ela foi ignorada. As palavras-chave foram combinadas usando o operador OR dentro de cada um desses dois elementos da estrutura. Também foi instanciada a saída correspondente aos resultados esperados, conforme mostrado na Tabela 2.2.

Tabela 2.2: Instanciação do PICO

(P)	*XML OR semistructured data OR RDF OR JSON*
(I)	*change detection OR diff OR change management OR change control OR issue tracking OR bug tracking OR change-based systems OR configuration management*
(O)	*techniques OR approaches OR methods OR methodologies OR processes OR support tools OR algorithms*

As expressões de busca utilizadas nas bibliotecas digitais selecionadas foram construídas com base na instanciação do PICO e nas características de cada uma, como mostra a Tabela 2.3.

Tabela 2.3: Expressões de buscas utilizadas nas bibliotecas digitais

Compendex	*((xml OR rdf OR json OR "semistructured data") AND ("change detection" OR diff OR "change management" OR "change control" OR "issue tracking" OR "bug tracking" OR "change-based systems" OR "configuration management"))*
IEEEXplore	*(((xml OR rdf OR json OR "semistructured data") AND ("change detection" OR diff OR "change management" OR "change control" OR "issue tracking" OR "bug tracking" OR "change-based systems" OR "configuration management")) wn KY and ({english} wn LA) and (72* wn CL))*
Scopus	*TITLE-ABS-KEY ((xml OR rdf OR json OR "semistructured data") AND ("change detection" OR diff OR "change management" OR "change control" OR "issue tracking" OR "bug tracking" OR "change-based systems" OR "configuration management")) AND (LIMIT-TO (SUBJAREA,"COMP") OR LIMIT-TO (SUBJAREA,"ENGI"))*

Na terceira etapa, determinou-se quais seriam considerados artigos de controle para ajustar a expressão de pesquisa. Uma revisão da literatura convencional anterior obteve os artigos de controle mostrados na Tabela 2.4. Eles são úteis para fornecer uma compreensão inicial da área bem como para definir palavras-chave de busca.

Tabela 2.4: Artigos de controle

1	Cobena, G., Abiteboul, S., Marian, A. Detecting changes in XML documents. In: ICDE, pages 41–52, San Jose, California, Feb. 2002.
2	Marian, A., Abiteboul, S., Cobena, G., Mignet, L. Change-centric management of versions in an XML warehouse. In: VLDB, pages 581–590, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
3	Wang, Y., DeWitt, D. J., Cai, J. X-diff: an effective change detection algorithm for XML documents. In: ICDE, pages 519–530, 2003.

Foi necessário também descrever os critérios de exclusão (PETERSEN *et al.*, 2008), com o objetivo de verificar se um artigo é um potencial candidato a ser selecionado ou a ser excluído do mapeamento sistemático. Neste mapeamento, foram usados os critérios de exclusão apresentados na Tabela 2.5.

Tabela 2.5: Critérios de exclusão

EC1	Publicações que não contêm informações sobre o *diff* de documentos XML
EC2	Publicações que contêm informações sobre a evolução de esquemas de documentos XML
EC3	Publicações que não estão disponíveis para *download* em bibliotecas digitais ou por qualquer outro meio sem custo para o pesquisador
EC4	As publicações que, apesar de terem sido devolvidas da expressão de pesquisa, não estão escritas em inglês
EC5	Publicações que descrevem procedimentos, tutoriais ou análogos

Finalizada a definição dos critérios de exclusão, a execução do protocolo do mapeamento teve início e retornou 442 publicações que foram avaliadas em quatro etapas, descritas a seguir. Na primeira etapa, foram identificadas 172 publicações duplicadas (encontradas em 2 ou 3 bibliotecas digitais) que foram eliminadas. Isto é uma consequência da sobreposição entre os documentos indexados em diferentes bibliotecas digitais. Na segunda etapa (Seleção de Publicações Relevantes - 1º Filtro), aplicou-se o primeiro filtro sobre as 270 publicações resultantes da etapa anterior. Foram lidos os títulos e resumos de todas as publicações e analisados os critérios de exclusão descritos na Tabela 2.5. Foram excluídos os trabalhos com títulos e resumos que indicassem claramente que estavam fora do escopo deste mapeamento sistemático. Muitos dos artigos não eram relevantes para a pesquisa. Assim, 198 publicações foram excluídas de acordo com os critérios estabelecidos. Na terceira etapa (Seleção de Publicações Relevantes - 2º filtro), foram lidas as seções de introdução e conclusão das 72 publicações resultantes. Os mesmos critérios de exclusão foram utilizados. Nesta etapa foram excluídas 56 publicações. Na quarta etapa (Seleção de Publicações Relevantes - 3º Filtro), foram lidas as 16 publicações resultantes da etapa anterior, do início ao fim. Nesta etapa, um dos

trabalhos foi excluído. Para finalizar, uma busca manual também foi efetuada com o objetivo de complementar os resultados encontrados nas bibliotecas digitais, a partir da pesquisa de artigos que citavam os artigos levantados no mapeamento bem como dos artigos citados por estes selecionados. Duas publicações foram incluídas como relevantes a partir da busca manual.

Todas as publicações retornadas neste processo foram analisadas com base nos critérios de exclusão e, ao final, 17 publicações (14 abordagens distintas) foram identificadas como relevantes no contexto da pesquisa. Estas iniciativas foram classificadas em três contextos e são apresentadas a seguir: detecção de mudanças em páginas *Web* frequentemente acessadas, mineração de mudanças em documentos XML e *diff* de documentos XML.

2.4.2 DETECÇÃO DE MUDANÇAS EM PÁGINAS *WEB* FREQUENTEMENTE ACESSADAS

Existem ferramentas com o intuito de detectar mudanças em páginas *Web* escritas em XML e HTML e que são frequentemente acessadas. WebVigiL (CHAMAKURA *et al.*, 2005) é um sistema de monitoramento de mudanças para páginas *Web* escritas em XML e HTML, de acordo com um perfil de usuário. O módulo de detecção de alterações desta abordagem é composto por dois algoritmos: CH-Diff e CX-Diff. CH-Diff é um algoritmo para detecção de mudanças em documentos HTML. CX-Diff (JACOB; SACHDE; CHAKRAVARTHY, 2003, 2005), por outro lado, é um algoritmo específico para detectar mudanças de conteúdo de *tags* de documentos XML. CX-Diff foi implementado em Java e usa DOM. Ele detecta alterações de forma customizada. O usuário pode especificar, por exemplo, a página a ser monitorada, o tipo de mudança que deve ser verificado e como as alterações serão notificadas. Esta solicitação é utilizada na detecção e na apresentação das mudanças.

O algoritmo heurístico CDA - *Change-Discovery Algorithm* (LIM; NG, 2004) tem como objetivo descobrir mudanças entre dois documentos XML ou HTML, hierarquicamente estruturados e representados por uma árvore ordenada. A proposta é uma extensão da abordagem descrita por LIM; NG (2001), que lida com a detecção de diferenças de documentos HTML.

2.4.3 *DIFF* DE DOCUMENTOS XML

Existem abordagens relacionadas a *diff* de documentos XML. Estas abordagens se baseiam numa análise estrutural dos documentos. Sua estratégia principal é encontrar fragmentos de dados que são correspondentes em ambas as versões de um documento XML e efetuar

o casamento (*match*) entre esses fragmentos. Depois disso, eles se concentram em identificar a ordem correta de operações que transforma uma versão do documento XML na outra, de uma forma independente do domínio. Em outras palavras, eles calculam o *delta*. O número de operações que forma um *delta* é denominado distância. A distância entre dois documentos XML depende do algoritmo de *diff* utilizado. Algumas abordagens tentam identificar qual é o menor conjunto de operações que transforma uma versão do documento XML na outra (*edit script* mínimo), o que facilita a compreensão das mudanças ocorridas. Em outras abordagens esta distância não é a menor possível, pois algumas delas não retornam o resultado ótimo. Isto porque, em determinadas aplicações, o desempenho é mais importante que a qualidade do resultado. O problema de calcular a distância também pode ser encontrado em diversos outros trabalhos relacionados à estrutura de árvores (TAI, 1979; ZHANG; SHASHA, 1989; CHAWATHE; GARCIA-MOLINA, 1997). O MH-Diff (CHAWATHE; GARCIA-MOLINA, 1997) é uma abordagem de detecção de diferenças que influenciou algumas das propostas descritas nesta seção. Ele não trabalha especificamente com documentos XML, mas com uma estrutura de árvores própria. Neste mapeamento, estas publicações não foram classificadas como relevantes justamente por não tratarem especificamente de documentos XML. A seguir são descritas as técnicas para detectar as diferenças entre as versões de um documento XML reveladas a partir do mapeamento.

O XyDiff (COBENA; ABITEBOUL; MARIAN, 2002) detecta as diferenças entre as versões de um documento XML a partir de uma abordagem baseada em árvores ordenadas, ou seja, a ordem dos filhos é relevante na detecção de diferenças. O XyDiff usa o formato XyDelta (MARIAN *et al.*, 2001), um único arquivo XML contendo todas as diferenças detectadas. Utilizando *hashes*, o XyDiff remove da comparação as subárvores idênticas, reduzindo assim a quantidade de dados para comparar, o que proporciona melhor desempenho do algoritmo quando comparado a outros. Vale ressaltar que o XyDiff considera as operações padrão em *diff*, bem como a operação de movimentação de subárvores. O algoritmo tem complexidade de tempo linear, mas não necessariamente encontra o *delta* mínimo. O XyDiff apresenta o *delta* a partir de uma lista de operações e essa saída tem algumas vantagens. Para o caso de uma base versionada, um dos objetivos é armazenar apenas algumas versões e o *delta*, de tal forma que seja possível construir uma versão não armazenada a partir de uma versão armazenada e o *delta* que representa a sua diferença para as demais versões não armazenadas. Isso proporciona economia de espaço de armazenamento. A saída neste formato também torna mais fácil o mapeamento para outro formato. Por outro lado, para o usuário, fica mais difícil

de identificar as diferenças entre as versões. O processo de detecção das diferenças e posterior geração de *delta* está dividido nas etapas mencionadas a seguir e ilustradas na Figura 2.4:

1) busca pelo casamento único entre os nós, por meio de informação do identificador associado (ID);
2) atribuição de assinaturas e pesos aos nós e ordenação de subárvores através de uma fila de prioridade;
3) realização de casamentos priorizando os nós e as subárvores de maior peso;
4) otimização dos casamentos a partir da procura por nós casados cujos pais possuam a mesma assinatura;
5) identificação das operações de inserção, atualização, remoção e movimentação e geração do *delta*.

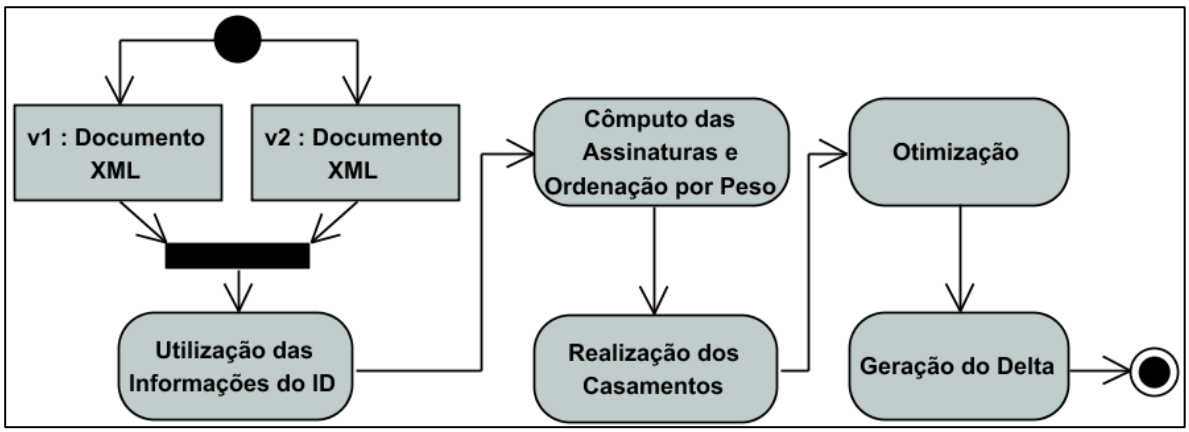

Figura 2.4: Algoritmo XyDiff

A abordagem XKeyMatch (SANTOS; HARA, 2007) é baseada na análise estrutural do documento XML e na utilização de chaves para apoiar o *diff*. O XKeyMatch usa o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002) para apoiar a detecção de diferenças. A leitura e a transformação dos documentos de entrada em árvores XML é feita pelo XyDiff. Estas árvores, juntamente com as chaves para XML definidas pelo usuário, são passadas ao XKeyMatch para efetuar o casamento dos elementos correspondentes nas duas versões do documento XML. As chaves garantem que as informações casadas correspondem a entidades equivalentes. Terminadas as tarefas do XKeyMatch, os resultados são passados ao XyDiff, que prossegue com suas ações para detectar as diferenças entre as versões. Como o XyDiff já recebe a informação relacionada às entidades equivalentes, tem-se uma redução no número de comparações e no processamento necessário. O XKeyMatch está dividido nas etapas mencionadas a seguir e ilustradas na Figura 2.5:

1) realização de um pré-processamento das versões do documento XML utilizando chaves, com o intuito de identificar e casar as entidades que possuam o mesmo valor para estes elementos;
2) detecção das diferenças entre as versões com base nos casamentos identificados na fase anterior;
3) geração do *delta*.

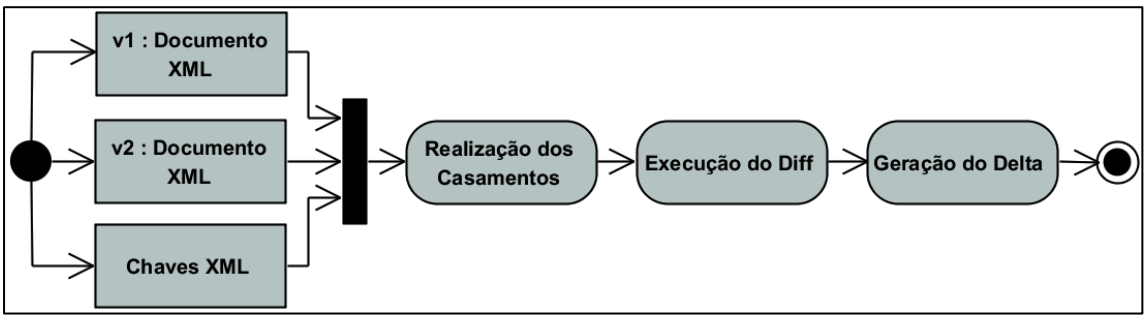

Figura 2.5: Algoritmo XKeyMatch

O X-Diff (WANG; DEWITT; CAI, 2003) é um algoritmo que usa árvores não ordenadas para detectar as diferenças entre as versões de um documento XML. Ele se concentra em garantir o *delta* mínimo. O algoritmo detecta o mapeamento mínimo entre os filhos de duas subárvores, reduzindo o problema a um problema de fluxo máximo com custo mínimo. De acordo com os autores, o algoritmo apresenta bom desempenho quando lida com documentos menores. Já quando se trata de grandes documentos XML, seu tempo de execução é longo. O X-Diff encontra um *delta* mínimo em tempo quadrático. Diferente do XyDiff, o X-Diff não utiliza uma lista de operações para representar o *delta*. O X-Diff utiliza a própria versão do documento XML para registrar as diferenças. Ele anexa a este documento os elementos XML que expressam as operações realizadas sobre ele ou que gerem a outra versão. Isso facilita a identificação das diferenças entre as versões por parte do usuário. Um problema com esta representação é o tamanho do *delta*. Além disso, essa estratégia de representação de diferenças torna difícil a geração da versão original a partir da alterada ou a composição de um *delta* a partir de duas ou mais saídas. O X-Diff, considera apenas as operações consideradas padrão. Apesar de ser uma desvantagem por não conseguir representar exatamente todas as operações (por exemplo, movimentação), um número menor de operações permite reduzir a complexidade do algoritmo de geração do *delta*. O processo de detecção das diferenças e posterior geração de *delta* está dividido nas etapas mencionadas a seguir e ilustradas na Figura 2.6:

1) realização de um pré-processamento que inclui a leitura e a análise dos documentos XML para transformá-los em estruturas de árvore e cômputo das assinaturas a partir de uma função de *hash*;
2) casamento de custo mínimo entre os elementos dos documentos XML, a partir dos valores *hash*;
3) otimização do processo a partir de um limiar (*threshold*) para selecionar o melhor candidato entre os coletados, evitando a comparação com todos os candidatos possíveis (etapa opcional);
4) o reconhecimento das operações padrão e geração do *delta*.

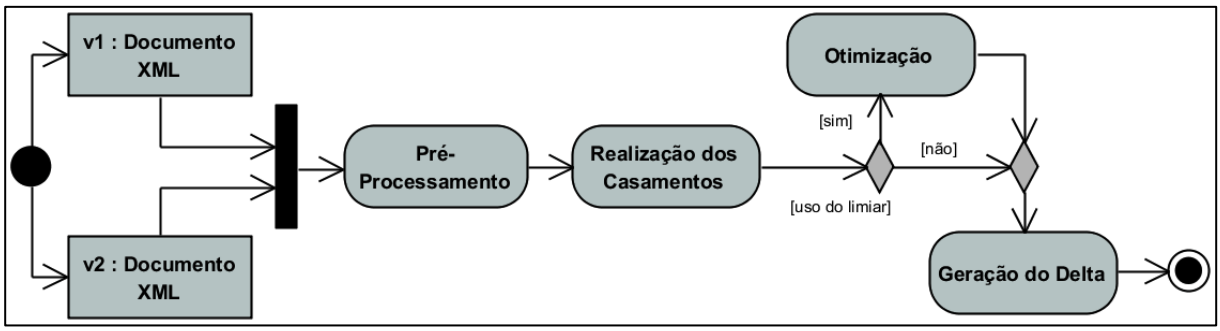

Figura 2.6: Algoritmo X-Diff

O BIODIFF (SONG; BHOWMICK; DEWEY, JR., 2007) é uma extensão do algoritmo X-Diff (WANG; DEWITT; CAI, 2003). É usado no contexto de Biologia Molecular Computacional para detectar mudanças exatas entre duas versões de um documento XML que representam anotações biológicas, com foco em estudar a relação entre as diferentes espécies de seres vivos. Assim como o X-Diff, o BIODIFF possui complexidade quadrática e usa árvores não ordenadas, bem como as operações padrão em diff. Como pode ser observado, o processo de detecção de diferenças descrito a seguir e apresentado na Figura 2.7 é bem semelhante ao X-Diff:

1) identificação e classificação dos elementos XML em diferentes tipos com base em sua estrutura a partir da DTD associado;
2) leitura e análise dos documentos XML para transformá-los em estruturas de árvore e cômputo das assinaturas a partir de uma função de *hash* (*parsing* e *hashing)*;
3) casamento entre os elementos dos documentos XML, a partir da classificação efetuada anteriormente;
4) geração do *delta*.

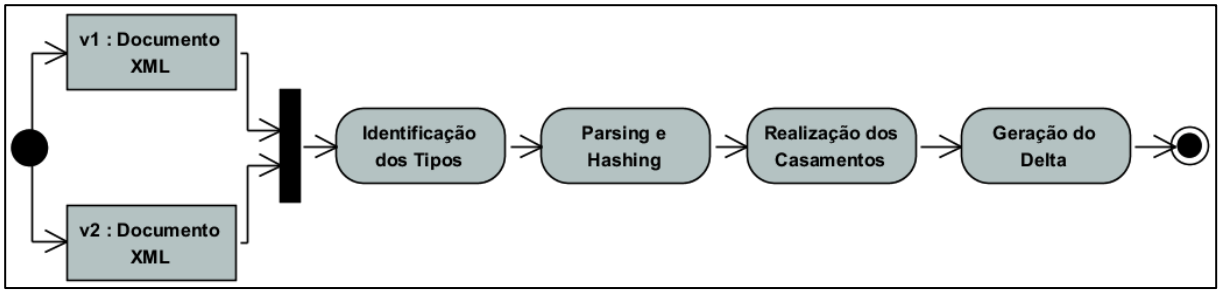

Figura 2.7: Algoritmo BIODIFF

O XRel_Change_SQL (SUNDARAM; MADRIA, 2012) detecta as diferenças entre as versões de um documento XML armazenadas em um banco de dados relacional. A abordagem explora consultas SQL ao invés de utilizar representações e cálculos baseados em DOM. Assim como o X-Diff, é baseado em um modelo de árvore não ordenada. Por outro lado, considera as operações padrão e a movimentação em subárvores, como o XyDiff. O XRel_Change_SQL possui as etapas definidas a seguir e ilustradas na Figura 2.8:

1) casamento das subárvores correspondentes a partir de uma abordagem baseada em similaridade;
2) detecção das movimentações realizadas nas subárvores;
3) detecção das inserções e remoções de nós;
4) identificação das operações padrão sobre os nós folha nas subárvores relacionadas.

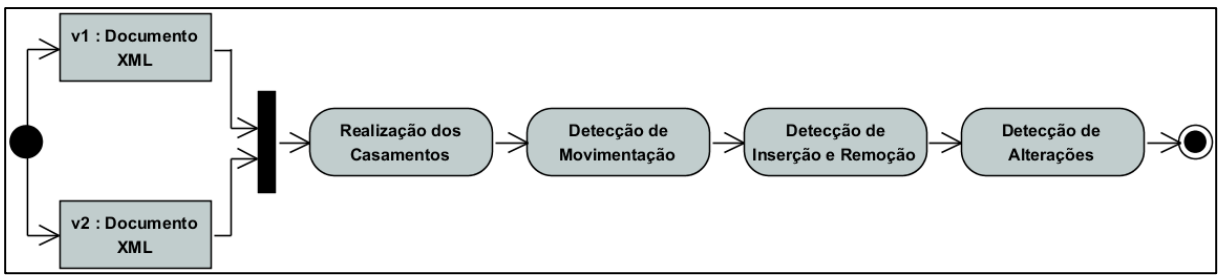

Figura 2.8: Algoritmo XRel_Change_SQL

O DiffX (AL-EKRAM; ADMA; BAYSAL, 2005) é uma abordagem de *diff* de documentos XML que leva em consideração árvores ordenadas e as operações padrão e a movimentação em documentos XML. O DiffX está dividido de acordo com as etapas definidas a seguir e ilustradas na Figura 2.9:

1) mapeamento de fragmentos de árvore isolados, a partir de uma estratégia baseada em similaridade
2) identificação dos casamentos mais amplos de fragmentos entre as versões

3) geração a sequência de operações de edição do mapeamento (*delta*).

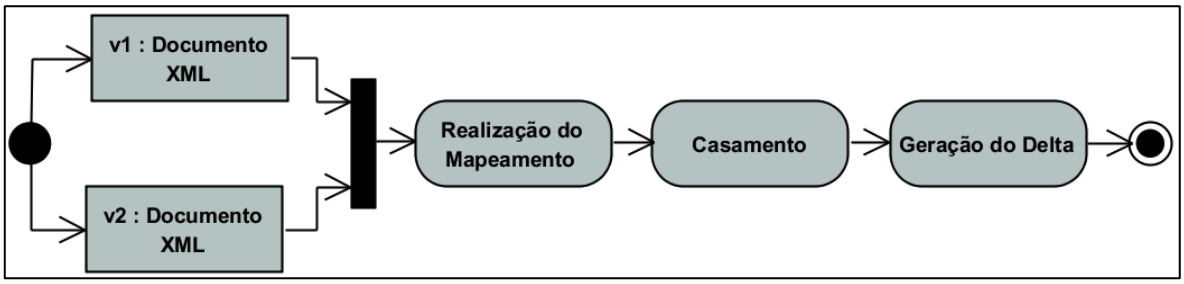

Figura 2.9: Algoritmo DiffX

O KF-Diff+ (XU *et al.*, 2002) é uma abordagem de *diff* de documentos XML que usa árvores ordenadas e não ordenadas e que considera as operações padrão e a movimentação em documentos XML. O KF-Diff+ está dividido de acordo com as etapas definidas a seguir e ilustradas na Figura 2.10:

1) leitura e análise dos documentos XML para transformá-los em estruturas de árvore;
2) realização do casamento de custo mínimo;
3) geração do delta.

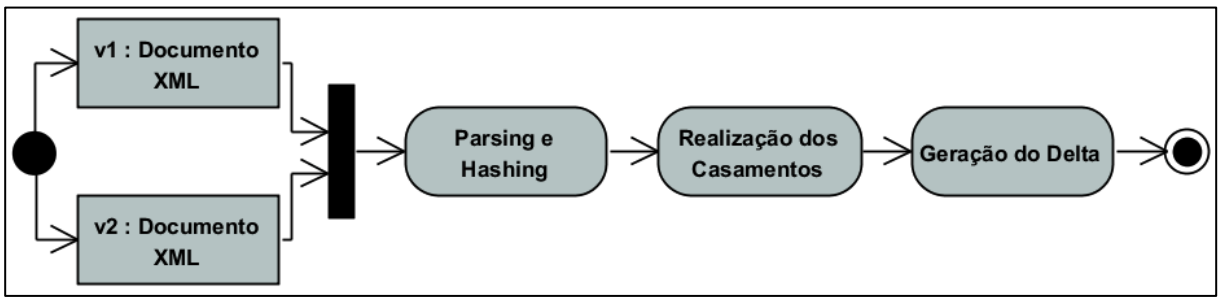

Figura 2.10: Algoritmo KF-Diff+

As abordagens a seguir lidam com o *diff* e o *merge* de documentos XML. O 3DM (LINDHOLM, 2004) se baseia em árvores ordenadas e suporta as operações de inserção, remoção, atualização, movimentação e cópia. As operações de movimentação e cópia podem ter um impacto grande no tamanho do *delta* mínimo, reduzindo-os consideravelmente. Na detecção de alterações, cada uma das versões é comparada com a versão original de forma independente. É necessário casar os nós da versão original com os nós das versões alteradas para descobrir quais foram alterados, inseridos ou removidos. Neste processo de correspondência

entre os nós, o 3DM tenta primeiro utilizar os identificadores definidos na DTD do documento. Quando o documento não possui IDs definidos na DTD, o 3DM utiliza uma função que determina a similaridade entre dois nós, levando em conta a similaridade entre os conteúdos (no caso de elementos que contêm textos), o nome dos elementos, atributos, filhos, dentre outras características. São diferenciados os nós que foram casados apenas no conteúdo (ou seja, a estrutura dele mudou), nós que foram casados apenas na estrutura (o conteúdo mudou) ou que foram completamente casados. Desta maneira o algoritmo pode tratar de forma diferenciada cada um dos casos, gerando resultados mais adequados. A partir dos casamentos dos elementos correspondentes, é possível obter o *delta*.

A abordagem Molhado (THAO; MUNSON, 2010) suporta as operações de inserção, remoção, atualização e movimentação. Além disso, a abordagem fornece uma interface gráfica para visualização das diferenças e resolução de conflitos. O Molhado trabalha com o modelo de árvores ordenadas. No entanto, de acordo com a semântica de documentos XML, ele ignora a ordem dos atributos e, com isso, trata todas as permutações de atributos como irrelevantes. Esta abordagem pressupõe que existem identificadores únicos para todos os nós presentes no documento base. Todas as alterações são representadas pelos *deltas*. Esta abordagem requer que os IDs sejam colocados sobre os elementos do documento base antes de ser compartilhado. Além disso, qualquer elemento nos documentos derivados que não têm um ID deve ser introduzido pelo editor como um novo nó (e receber um novo ID). Assim, o algoritmo não funciona com qualquer documento XML. Como consequência, para a utilização desta abordagem, também é necessário que os editores utilizados preservem os IDs durante a edição. A vantagem, de acordo com os autores, é que esta abordagem é capaz de representar com precisão mudanças radicais nos elementos, o que é muito difícil nos modelos baseados em *hash*.

O DOCTREEDIFF (RÖNNAU; PHILIPP; BORGHOFF, 2009) considera que um documento XML é uma árvore ordenada, com a maior parte do conteúdo armazenado nos nós folha. A abordagem foi desenvolvida no intuito de que as entradas sigam o seguinte padrão: conteúdo concentrado nas folhas; alterações de estrutura realizadas nos níveis mais altos da árvore; e muitos nós não-folha iguais devido a marcação idêntica do documento. O algoritmo considera as operações padrão, além da movimentação. Com o objetivo de alcançar eficiência do algoritmo na transformação dos documentos e *deltas* de qualidade, foi introduzido um modelo que leva em conta os nós vizinhos. Neste modelo de *delta*, a inversão de uma operação é possível, permitindo a reconstrução da versão anterior a partir da nova. A ideia chave do algo-

ritmo para detectar as diferenças, efetuar os casamentos e gerar o *delta* é baseada no algoritmo LCS - *Longest Common Subsequence* (MAIER, 1978) e em funções de *hash*.

2.4.4 MINERAÇÃO DE MUDANÇAS EM DOCUMENTOS XML

Existem trabalhos relacionados à mineração de mudanças em documentos XML. Estas propostas, descritas a seguir, estão mais concentradas na descoberta de regras de associação ou de *itemsets* frequentes.

O trabalho desenvolvido por Rusu *et al*. (2006) tem como objetivo a mineração de regras de associação a partir de documentos XML. O trabalho foca na mineração de documentos XML dinâmicos, ou seja, que podem mudar sua estrutura ou conteúdo ao longo do tempo, como o registro dos funcionários de uma empresa ou de produtos de um supermercado. A abordagem propõe a construção de um algoritmo genérico para a extração das regras de associação em documentos XML, baseado no Apriori (AGRAWAL; SRIKANT, 1994). A proposta usa o X-Diff (WANG; DEWITT; CAI, 2003) para gerar o *delta* consolidado com o histórico de mudanças que ocorreram nas versões do documento XML. As regras extraídas podem trazer informações de como as mudanças ocorridas em diferentes partes dos documentos estão relacionadas, ou seja, se existem relações entre as modificações, remoções ou inserções de um elemento com outro.

Outro trabalho nesta linha aborda o problema da descoberta de estruturas que mudam frequentemente de acordo com determinados padrões, tendo em conta a sua natureza dinâmica (ZHAO; BHOWMICK; MADRIA, 2004). A proposta se concentra em mineração da estrutura dinâmica baseada em padrões de versões não ordenadas de documentos XML. Uma modificação do algoritmo X-Diff (WANG; DEWITT; CAI, 2003) é usada no processo de mineração para apoiar o processo de *diff* e um algoritmo para identificar as estruturas que mudam frequentemente é proposto. Um modelo para guardar o histórico de mudanças estruturais é proposto também. Este modelo é uma extensão do modelo DOM com algumas propriedades de histórico, tornando possível comprimir o histórico de mudanças do documento XML. A descoberta destes padrões de mudança, a partir da análise das versões de um determinado documento XML, pode, por exemplo, prever tendências em comércio eletrônico.

2.5 DISCUSSÕES E CONSIDERAÇÕES FINAIS

A Tabela 2.6 lista algumas das características mencionadas sobre as abordagens de *diff* apresentadas neste capítulo. As abordagens são listadas nesta tabela na mesma ordem em que são brevemente descritas neste capítulo. Para tanto, utilizou-se o nome da abordagem seguido

da referência relacionada. Para as duas últimas abordagens, utilizou-se apenas a referência correspondente, uma que não possuem um nome associado. Sempre que não foi possível determinar a resposta a algum critério com pesquisa na literatura, o mesmo foi marcado com "?".

Tabela 2.6: Características dos algoritmos de *diff* de documentos XML

Abordagem	Árvore Ordenada	*Edit Script* Mínimo	Operações	Implementação	Citações
CX-Diff (JACOB; SACHDE; CHAKRAVARTHY, 2003, p., 2005; CHAMAKURA *et al.*, 2005)	sim	?	?	Java	11
CDA (LIM; NG, 2004)	sim	?	?	Java	3
XyDiff (MARIAN et al., 2001; COBENA; ABITEBOUL; MARIAN, 2002)	sim	não	I, A, R, M	C++	565
XKeyMatch (SANTOS; HARA, 2007)	sim	não	I, A, R, M	C++	6
X-Diff (WANG; DEWITT; CAI, 2003)	não	sim	I, A, R	C++	464
BioDiff (SONG; BHOWMICK; DEWEY, JR., 2007)	não	sim	I, A, R	Java	2
XRel_Change_SQL (SUNDARAM; MADRIA, 2012)	não	sim	I, A, R, M	?	6
DiffX (AL-EKRAM; ADMA; BAYSAL, 2005)	sim	?	I, A, R, M	?	69
KF-Diff (XU *et al.*, 2002)	ambas	sim	I, A, R, M	C, C++	15
3DM (LINDHOLM, 2004)	sim	?	I, A, R, M, C	?	114
Molhado (THAO; MUNSON, 2010)	sim	?	I, A, R, M	?	14
DOCTREEDIFF (RÖNNAU; PHILIPP; BORGHOFF, 2009)	sim	?	I, A, R, M	?	35
(RUSU; RAHAYU; TANIAR, 2006)	?	?	?	VB	26
(ZHAO; BHOWMICK; MADRIA, 2004)	não	?	?	?	9

A maioria das abordagens aproveita-se do formato em árvore dos documentos XML para realizar as comparações. Alguns algoritmos se baseiam em árvores não ordenadas, como o X-Diff (WANG; DEWITT; CAI, 2003), enquanto outros se baseiam em árvores ordenadas, como o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002), o CX-Diff (JACOB; SACHDE;

CHAKRAVARTHY, 2003, 2005), o CDA (LIM; NG, 2004) e o 3DM (LINDHOLM, 2004), entre outros. Como mencionado anteriormente, os elementos são ordenados em um documento XML (BRAY *et al.*, 2008) e, a rigor, não seria correto modelar um arquivo XML como uma árvore não ordenada. No entanto, dependendo da aplicação, a ordem pode não ser relevante, e tais algoritmos podem ser valorosos. Uma simples troca de ordem entre elementos cuja ordem na prática não é relevante, como no cenário de cadastro de funcionários de uma empresa, seria vista como uma movimentação por algoritmos que utilizam modelos de árvore ordenada, ou até mesmo como uma remoção seguida de uma inserção por algoritmos que não suportam movimentações. Já um algoritmo baseado em árvores não ordenadas ignoraria a mudança, obtendo um *delta* menor e sem complexidade desnecessária. Por outro lado, existem aplicações onde a ordem é importante e, portanto, seria mais adequada a utilização de uma abordagem que lide com árvores ordenadas, como, por exemplo, no cenário de redação de um livro composto por várias seções (a ordem das seções é relevante).

Sobre as abordagens de mineração de mudanças em documentos XML (ZHAO; BHOWMICK; MADRIA, 2004; RUSU; RAHAYU; TANIAR, 2006), ambas usam o X-Diff (WANG; DEWITT; CAI, 2003) para gerar o *delta* consolidado com o histórico de mudanças. A abordagem proposta por ZHAO; BHOWMICK; MADRIA (2004) usa árvores não ordenadas. Ela também usa DOM para criar uma estrutura de árvore na memória que contém todos os dados relatados no documento XML e que permite a navegação pelos seus itens.

Como mencionado anteriormente, sobre as abordagens de *diff* de documentos XML, algumas tentam encontrar o *edit script* mínimo, o que facilita a compreensão do que realmente aconteceu neste processo de alterações, como é o caso do X-Diff (WANG; DEWITT; CAI, 2003). Outras abordagens, como por exemplo o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002), devido ao seu contexto de armazenamento de grandes volumes de dados, prioriza a eficiência em termos de memória e velocidade, mesmo que isso implique em não alcançar o *edit scrip*t mínimo. As operações padrão de inserção (I), atualização (A) e remoção (R) estão presentes em todas as abordagens. Já a movimentação (M) é utilizada pelas abordagens XyDiff (COBENA; ABITEBOUL; MARIAN, 2002), XRel_Change_SQL (SUNDARAM; MADRIA, 2012), XKeyMatch (SANTOS; HARA, 2007), diffX (ALEKRAM; ADMA; BAYSAL, 2005), KF-Diff+ (XU *et al.*, 2002), 3DM (LINDHOLM, 2004), Molhado (THAO; MUNSON, 2010) e DOCTREEDIFF (RÖNNAU; PHILIPP; BORGHOFF, 2009). A operação de cópia (C) é utilizada pelo 3DM (LINDHOLM, 2004). As operações de atualização, movimentação e cópia podem ser substituídas por uma combinação de operações de inserção e remoção. Mesmo assim, elas são utilizadas em alguns algoritmos uma vez que o

seu uso diminui o tamanho de suas saídas, que sempre consiste em uma lista das alterações necessárias para se chegar a uma versão do documento XML a partir de outra. Além disso, o uso de movimentação e cópia no lugar de inserção e remoção já é um passo, ainda que tímido, no sentido de enriquecer semanticamente o resultado obtido. Uma desvantagem de usar outras operações além da inserção e remoção é que normalmente o algoritmo não alcança um *edit script* mínimo (COBENA; ABITEBOUL; MARIAN, 2002).

Outro ponto que pode ser observado diz respeito à influência do X-Diff e do XyDiff nas demais abordagens e na quantidade de citações em artigos. O BIODIFF (SONG; BHOWMICK; DEWEY, JR., 2007) é uma extensão do X-Diff usada na Biologia Molecular Computacional para detectar mudanças em anotações biológicas, ou seja, não trabalha com qualquer documento XML. A abordagem XKeyMatch (SANTOS; HARA, 2007) utiliza o XyDiff para detectar as diferenças entre as versões do documentos XML. A Figura 2.11 mostra este relacionamento entre as abordagens obtidas durante o mapeamento sistemático e publicadas entre 2001 e 2012. Como nem todas tem um nome associado, utilizou-se a referência relacionada para a identificação da abordagem.

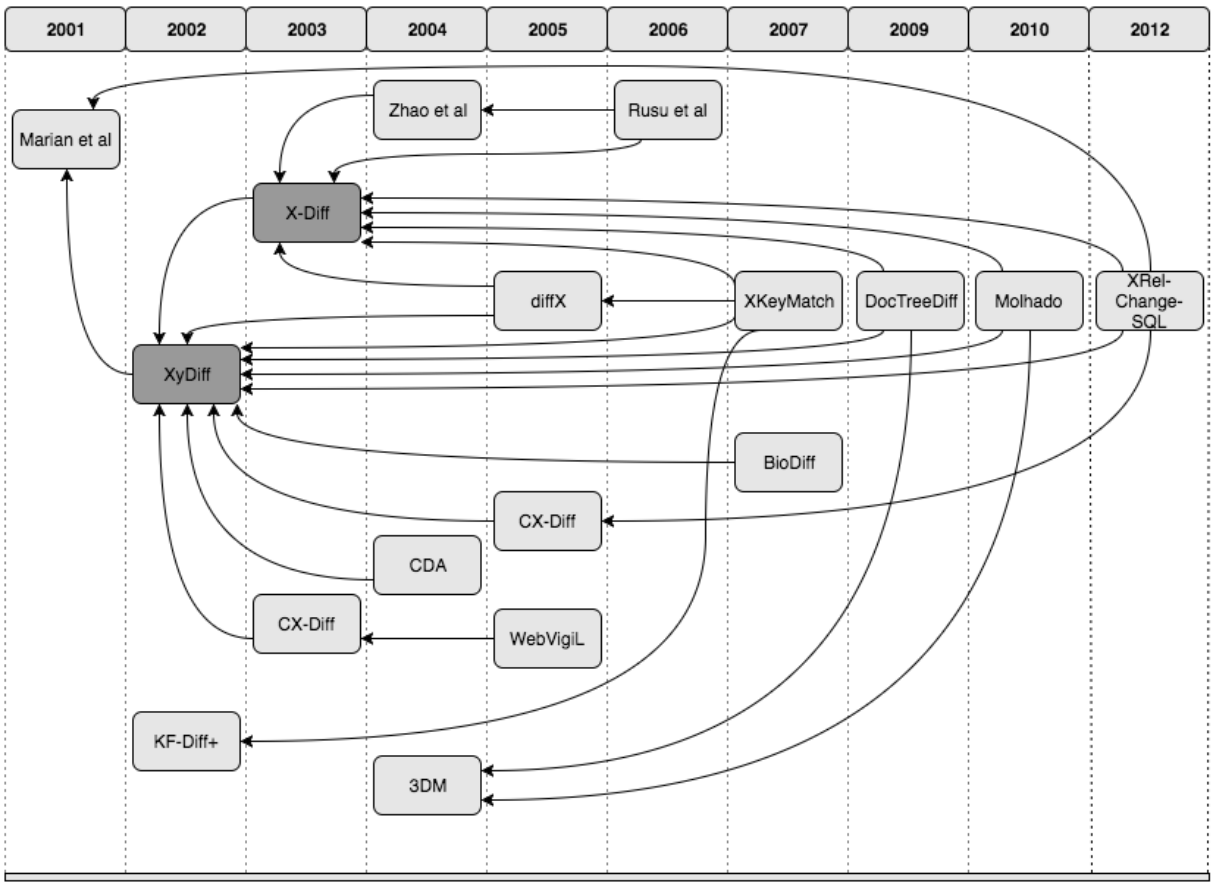

Figura 2.11: Relacionamento entre as abordagens de *diff* sintático

Os trabalhos de *diff*, especificamente de documentos XML, mais citados estão destacados em cinza. O XyDiff (COBENA; ABITEBOUL; MARIAN, 2002) e o X-Diff (WANG; DEWITT; CAI, 2003) foram mencionadas sete e quatro vezes, respectivamente. Isto mostra claramente a influência dessas abordagens sobre as demais. Vale ressaltar que parte das abordagens de detecção de mudanças em páginas *Web* foram influenciadas pelo XyDiff. Entre as abordagens relacionadas a *diff* sintático, a influência do XyDiff também é evidente, sendo referenciado em quatro delas. Já entre as abordagens relacionadas à mineração de mudanças em documentos XML, a influência maior foi do X-Diff, referenciado pelas duas abordagens descritas. A influências destes dois trabalhos também pode ser observada na Tabela 2.6, na coluna citações. Uma informação relevante, que não consta nesta tabela, é que o X-Diff e o XyDiff têm implementação disponível na Web, assim como o 3DM. Para as demais abordagens relacionadas não foram encontradas as respectivas implementações.

Por fim, é possível notar que o foco dessas abordagens está principalmente voltado para a detecção de mudanças sintáticas entre as versões de um documento XML. Algumas abordagens consideram que a detecção semântica está relacionada às operações de transformação, ou seja, dada uma versão *v1* de um documento XML, o objetivo é descobrir qual é a ordem correta das operações para se obter a versão *v2*. Outras abordagens identificam movimentações e cópias de elementos ao invés de simples ações de remoção e inserção de elementos. No entanto, nenhuma delas é capaz de elevar o nível de abstração do *delta* gerado e informar, segundo os termos que pertencem ao domínio de conhecimento do documento XML, o significado das diferenças entre as duas versões.

CAPÍTULO 3 – *DIFF* SEMÂNTICO

3.1 INTRODUÇÃO

Este capítulo apresenta o XChange, uma abordagem para apoiar a compreensão da evolução de documentos XML baseada em inferência. O objetivo do XChange é permitir ao usuário identificar e compreender as mudanças semânticas ao analisar versões de um documento XML. Diferente das abordagens existentes, o XChange usa as mudanças sintáticas das versões e um conjunto de regras para inferir a razão das mudanças e apoiar o *diff* semântico.

A Figura 3.1 apresenta uma visão geral do XChange. Para que o XChange possa efetuar o *diff* semântico entre duas versões sequenciais (não necessariamente consecutivas) de um documento XML, ele precisa ser configurado para o domínio de conhecimento do documento XML. Depois de configurado, ele pode ser utilizado diversas vezes para comparar versões de diferentes documentos XML que pertencem àquele domínio. Nesta figura, há dois papéis que interagem com o XChange: o especialista de domínio (chamado de especialista deste ponto em diante), que faz a configuração do XChange para um dado domínio de conhecimento e o usuário final (chamado de usuário deste ponto em diante), que compara as versões dos documentos pertencentes àquele domínio.

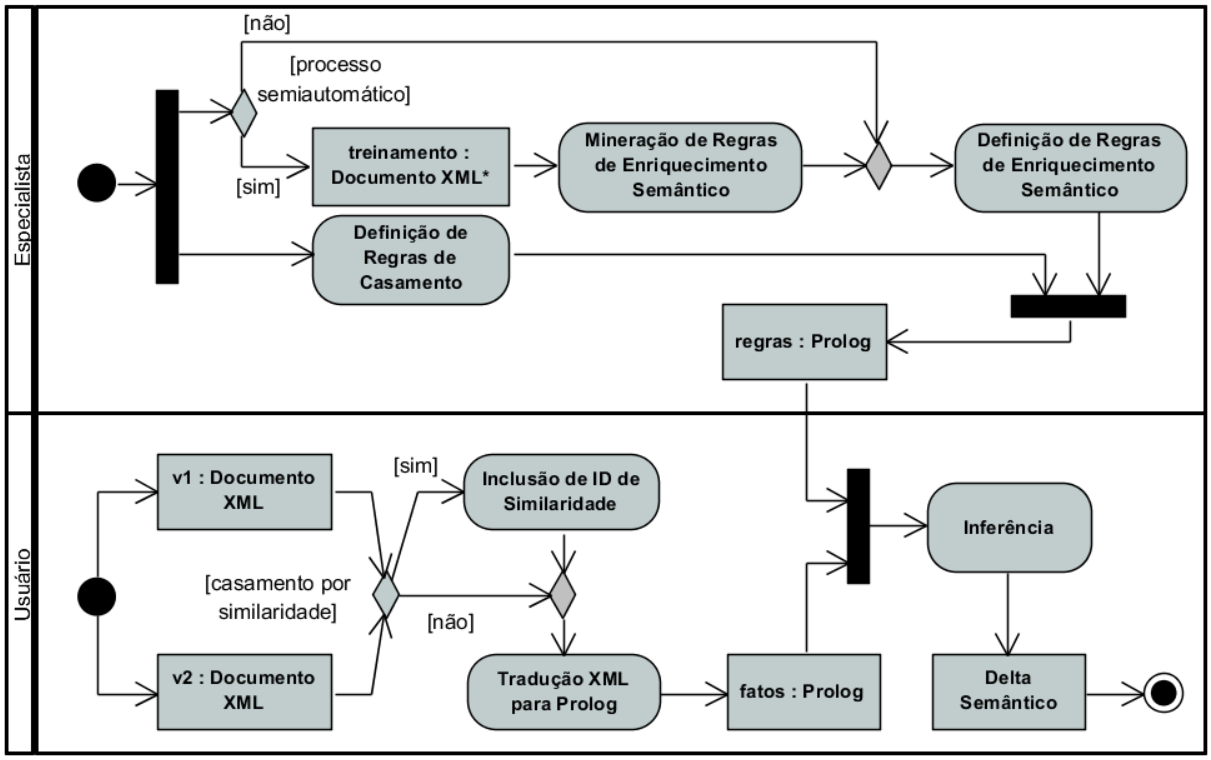

Figura 3.1: Diagrama de atividades UML apresentando a visão geral do XChange

O especialista deve executar duas atividades principais para configurar o XChange: a definição de regras de casamento e a definição de regras de enriquecimento semântico. A **Definição de Regras de Casamento** visa identificar os elementos correspondentes nas duas versões sequenciais. Existem diferentes formas de identificar elementos correspondentes em documentos XML. Nesta tese são levadas em consideração duas delas: casamento por chave e casamento por similaridade. Uma regra de casamento por chave poderia, por exemplo, indicar que elementos de duas versões de um documento XML são correspondentes quando seus subelementos de nome "CPF" têm valores iguais. Por outro lado, uma regra de casamento por similaridade faz uso de IDs artificiais, obtidos a partir da análise de similaridade entre os elementos das versões do documento XML, para indicar a correspondência, conforme discutido mais a frente.

A **Definição de Regras de Enriquecimento Semântico** pode ser feita de forma manual ou através de um processo semiautomático. É possível definir regras manuais simples, equivalentes a um *diff* sintático, que indicam que uma modificação em algum elemento foi efetuada (por exemplo, no contexto da Prefeitura de Baltimore, uma mudança de salário). Por outro lado, regras manuais mais elaboradas também podem ser definidas, agrupando várias modificações. Por exemplo, uma regra pode indicar que um funcionário foi promovido a partir da informação sintática de que ele teve seu salário aumentado e o nome do seu cargo alterado. Como o processo manual de definição de regras pode se tornar complexo, mesmo para um especialista, há a possibilidade de fazer uso de um processo semiautomático. Esse processo inicia pela **Mineração de Regras de Enriquecimento Semântico** a partir de um conjunto de documentos XML. O objetivo dessa mineração é descobrir elementos do documento XML que mudam em conjunto com frequência, ao se analisar versões sequenciais. Por exemplo, se o aumento de salário ocorre frequentemente com a alteração de cargo, o processo de mineração é capaz de identificar essa ocorrência conjunta, e o especialista é necessário somente para nomear essa regra, dando um significado para essa mudança conjunta frequente. Estas regras de enriquecimento, juntamente com as regras de casamento, são utilizadas no momento da inferência para produzir o *diff* semântico.

Após a configuração das regras para um dado domínio, um usuário pode fornecer duas versões de um documento XML pertencentes àquele domínio para serem comparadas. Se o casamento dos elementos correspondentes for efetuado por similaridade, é feita a **Inclusão de ID de Similaridade** nas versões do documento XML. Esse ID de similaridade, inserido artificialmente nos elementos dos documentos XML, indica quais elementos são correspondentes por terem grau de similaridade, considerando atributos, subelementos e conteúdo, superior a

um limiar previamente configurado. Em seguida é feita a **Tradução XML para Prolog**, que transforma cada elemento das duas versões do documento XML em fatos Prolog (LIMA *et al.*, 2012).

Por fim, o processamento do *diff* é efetuado via **Inferência**. Uma base de conhecimento é construída a partir dos fatos e das regras gerados nos passos anteriores. Essa base de conhecimento é submetida a consultas usando as cabeças de cada uma das regras de enriquecimento semântico, que fornecem como resposta um *delta* semântico contendo os elementos do documento XML que se encaixaram nas situações modeladas pelas regras. Em outras palavras, este *delta* corresponde à razão da evolução do documento XML, de uma versão anterior para uma posterior. Por exemplo, o *delta* pode informar, entre outras coisas, quais funcionários foram promovidos ao se analisar duas versões sequenciais de um documento XML.

Este capítulo está organizado como segue. A Seção 3.2 define as regras de identificação de elementos correspondentes, classificadas em casamento por chave e casamento por similaridade. A Seção 3.3 define as regras de enriquecimento semântico. A Seção 3.4 descreve o processo semiautomático de mineração de regras de enriquecimento semântico. A Seção 3.5 apresenta o processo de inclusão de um ID de similaridade, utilizado na estratégia de casamento por similaridade. A Seção 3.6 apresenta a tradução das versões do documento XML em fatos Prolog. Para finalizar, a Seção 3.7 descreve o processo de inferência utilizado no *diff* semântico, enquanto a Seção 3.8 apresenta as considerações finais deste capítulo.

3.2 DEFINIÇÃO DE REGRAS DE CASAMENTO

Como mencionado anteriormente, existem diferentes formas de identificar elementos correspondentes em documentos XML. As duas estratégias de casamento (*match*) usadas nesta tese, casamento por chave e casamento por similaridade, são descritas a seguir.

Uma forma simples de se estabelecer a correspondência entre elementos é a adoção de uma chave. Para exemplificar, a regra *match*, apresentada na Figura 3.2, no contexto da Prefeitura de Baltimore, utiliza o elemento <*name*> como um identificador (chave) para estabelecer a correspondência entre os elementos das versões deste documento. Esta estratégia garante a unicidade do elemento <*employee*> e pode ser usada também em outros cenários/domínios (nesse caso, com chave diferente, dependendo do domínio).

```
1 match(EMPLOYEEBefore, EMPLOYEEAfter, NAME):
2   employee(before,EMPLOYEEBefore), employee(after,EMPLOYEEAfter),
3   name(EMPLOYEEBefore, _, NAME), name(EMPLOYEEAfter, _, NAME).
```

Figura 3.2: Regra de casamento por chave

Como descrito, na estratégia de casamento por chave, o usuário utiliza a regra *match* para identificar elementos correspondentes entre duas versões, *v1* e *v2* por exemplo. Esta regra usa um atributo-chave (<*name*> no exemplo) que é um identificador definido pelo especialista. Contudo, dependendo da forma como os documentos XML são gerenciados, não há garantia de que o valor permanece o mesmo entre as versões sequenciais. Por exemplo, pode acontecer um erro de digitação no valor de <*name*> em *v1* que é corrigida em *v2*. Outro problema relacionado é que a maioria dos documentos XML não tem um elemento identificador e nem um esquema associado (MAAROUF; CHUNG, 2008; VYHNANOVSKÁ; MLÝNKOVÁ, 2010; GRIJZENHOUT; MARX, 2013). Neste contexto, pode-se utilizar o casamento por similaridade, a partir do uso de IDs artificiais, detalhado na Seção 3.5. Desta forma, pode-se usar a regra *match* da Figura 3.2, com o elemento identificador gerado artificialmente como chave, como mostra a Figura 3.3, para estabelecer os elementos correspondentes entre duas versões do documento XML.

```
1 match(EMPLOYEEBefore, EMPLOYEEAfter, XID):-
2   employee(before,EMPLOYEEBefore),employee(after,EMPLOYEEAfter),
3   xchangeid(EMPLOYEEBefore, _, XID),xchangeid(EMPLOYEEAfter, _, XID).
```

Figura 3.3: Regra de casamento por similaridade com o uso de identificador artificial

3.3 DEFINIÇÃO DE REGRAS DE ENRIQUECIMENTO SEMÂNTICO

As regras de enriquecimento semântico são definidas uma vez para cada domínio, podendo ser utilizadas posteriormente a cada execução do *diff*. O especialista é fundamental neste processo de configuração das regras, que pode ser feito manualmente como descrito a seguir ou a partir de um processo semiautomático descrito na próxima seção.

Um especialista com conhecimento no domínio pode criar regras que permitam a inferência sobre fatos Prolog e apoiem o *diff* semântico. Para exemplificar, a Figura 3.4 apresenta algumas regras definidas no contexto da Prefeitura de Baltimore. A regra *salary_increased* (linhas 1 a 6) identifica os funcionários que receberam um aumento de salário (<*annualsalary*>). A regra *transferred* (linhas 7 a 12) identifica os funcionários que mudaram de agência (*<agencyid>)* enquanto a regra *fired* (linhas 13 a 18) mostra os funcionários demitidos. A regra *promoted* (linhas 19 a 27) identifica os funcionários que receberam aumento de salário (<*annualsalary*>) e, além disso, mudaram de função (<*jobtitle*>), o que significa que eles foram promovidos. Para finalizar, a regra *promoted_transferred* (linhas 28 a 39) identifica os funcionários promovidos e transferidos.

Como pode ser observado, é possível definir regras equivalentes a um *diff* sintático, que indicam que uma modificação em algum elemento foi efetuada, como é o caso da regra *salary_increased*. Por outro lado, regras mais elaboradas também podem ser definidas, agrupando várias modificações, como é o caso da regra *promoted*. Como essas regras utilizam inferência e buscam compreender o significado (semântica) destas mudanças, pode-se dizer que elas produzem um *diff* semântico.

```
salary_increased(NAME):-
   match(EMPLOYEEBefore, EMPLOYEEAfter, XID),
   name(EMPLOYEEBefore, _, NAME),
   annualsalary(EMPLOYEEBefore, _, ANNUALSALARYBefore),
   annualsalary(EMPLOYEEAfter, _, ANNUALSALARYAfter),
   ANNUALSALARYBefore<ANNUALSALARYAfter.

transferred(NAME):-
   match(EMPLOYEEBefore, EMPLOYEEAfter, XID),
   name(EMPLOYEEBefore, _, NAME),
   agencyid(EMPLOYEEBefore, _,AGENCYIDBefore),
   agencyid(EMPLOYEEAfter, _,AGENCYIDAfter),
   AGENCYIDBefore\=AGENCYIDAfter.

fired(NAME):-
   employee(before,EMPLOYEEBefore),
   xchangeid(EMPLOYEEBefore, _, XID),
   name(EMPLOYEEBefore, _, NAME),
   not((employee(after,EMPLOYEEAfter),
   xchangeid(EMPLOYEEAfter, _, XID))).

promoted(NAME):-
   match(EMPLOYEEBefore, EMPLOYEEAfter, XID),
   name(EMPLOYEEBefore, _, NAME),
   jobtitle(EMPLOYEEBefore, _, JOBTITLEBefore),
   jobtitle(EMPLOYEEAfter, _, JOBTITLEAfter),
   JOBTITLEBefore\=JOBTITLEAfter,
   annualsalary(EMPLOYEEBefore, _, ANNUALSALARYBefore),
   annualsalary(EMPLOYEEAfter, _, ANNUALSALARYAfter),
   ANNUALSALARYBefore<ANNUALSALARYAfter.

promoted_transferred(NAME):-
   match(EMPLOYEEBefore, EMPLOYEEAfter, XID),
   name(EMPLOYEEBefore, _, NAME),
   jobtitle(EMPLOYEEBefore, _, JOBTITLEBefore),
   jobtitle(EMPLOYEEAfter, _, JOBTITLEAfter),
   JOBTITLEBefore\=JOBTITLEAfter,
   agencyid(EMPLOYEEBefore, _,AGENCYIDBefore),
   agencyid(EMPLOYEEAfter, _,AGENCYIDAfter),
   AGENCYIDBefore\=AGENCYIDAfter,
   annualsalary(EMPLOYEEBefore, _, ANNUALSALARYBefore),
   annualsalary(EMPLOYEEAfter, _, ANNUALSALARYAfter),
   ANNUALSALARYBefore<ANNUALSALARYAfter.
```

Figura 3.4: Exemplos de regras de enriquecimento semântico

Embora os Itens de Configuração sujeitos ao *diff* semântico sejam documentos XML, os especialistas precisam elaborar as regras de enriquecimento semântico em Prolog, uma vez que os documentos XML são traduzidos para esta linguagem. Ainda que muitos dos usuários sejam de áreas tecnológicas, nem todos podem estar familiarizados com Prolog. Diante disso, foi desenvolvida uma interface gráfica para apoiar a definição das regras de enriquecimento semântico pelo especialista.

A Figura 3.5 apresenta a definição da regra de enriquecimento semântico *promoted_transferred* utilizando um formulário para configuração das regras. Inicialmente é necessário informar o nome da regra (*Rule Name*), a saída (*Output*) e as condições a serem satisfeitas (*Conditions*). Um conjunto inicial, envolvendo os elementos e a relação entre eles, é fornecido para a definição das condições. Neste exemplo, o nome da regra é *promoted_transferred* e a saída esperada envolve o nome do funcionário (<*name*>). Sobre as condições, neste exemplo, o cargo do funcionário nas duas versões deve ser diferente *(<jobtitle>)* bem como a agência em que ele trabalha (<*agencyid*>). Além disso, o salário anual deve ser maior na segunda versão analisada (<*annualsalary*>). Os demais elementos que compõem a regra e se referem à identificação de elemento correspondente são inseridos automaticamente após a definição da regra de enriquecimento semântico. De posse da especificação das regras a partir da interface, o XChange define a regra em Prolog, como já apresentado na Figura 3.4.

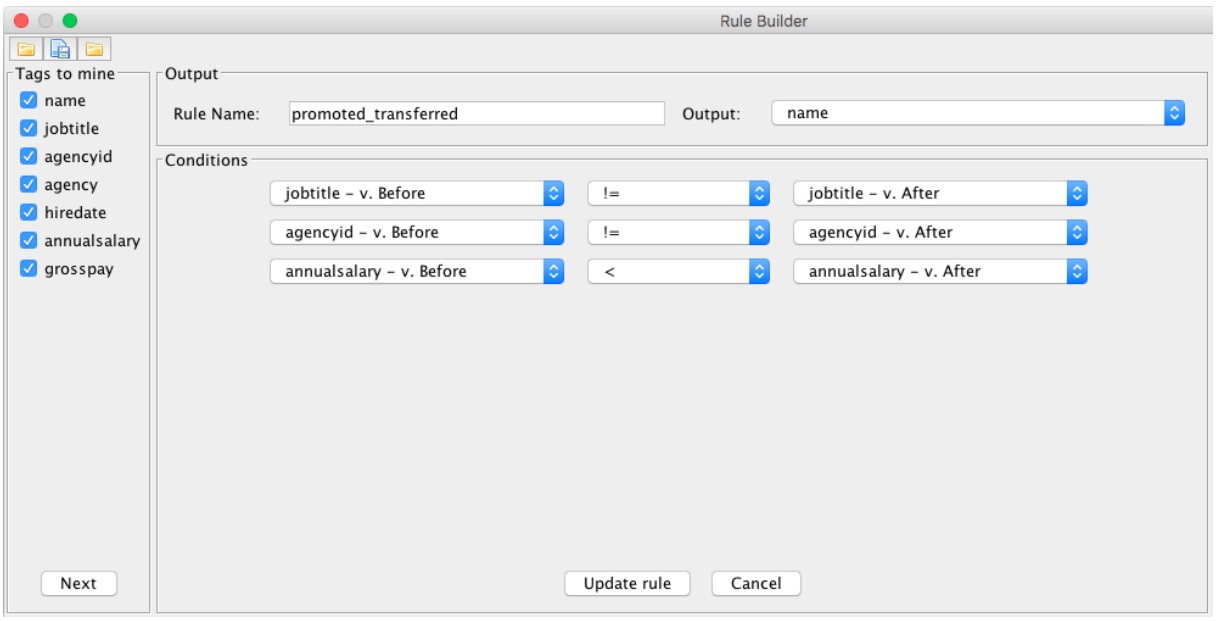

Figura 3.5: Interface de apoio a definição das regras de enriquecimento semântico

3.4 MINERAÇÃO DE REGRAS DE ENRIQUECIMENTO SEMÂNTICO

Como mencionado anteriormente, o especialista define as regras que apoiam a inferência. Esta tarefa se concentra no início do processo, mas em algumas situações, o especialista perceberá a necessidade de definir novas regras e o fará em outras fases. Ainda assim, esta é uma tarefa complexa e demorada. Diante disso, o XChange fornece um apoio semiautomático para construção das regras de enriquecimento semântico baseado em *itemsets* frequentes (AGRAWAL; SRIKANT, 1994).

A tarefa de mineração de regras de associação (AGRAWAL; IMIELIŃSKI; SWAMI, 1993; AGRAWAL; SRIKANT, 1994) tem como objetivo encontrar relacionamentos ou padrões frequentes entre conjuntos de dados. Um exemplo de aplicação de regras de associação que se tornou muito conhecido está inserido no cenário de compras em um supermercado. A ideia é descobrir quais produtos os clientes costumam comprar em conjunto quando vão ao supermercado. A partir daí, pode-se descobrir informações úteis no que diz respeito ao planejamento de vendas a fim de criar promoções, organizar os produtos nas prateleiras e gerar propagandas, por exemplo.

Fazendo uma analogia com o cenário de compras em um supermercado, no cenário de cadastro de funcionários uma regra de associação poderia ajudar a descobrir quais elementos são alterados em conjunto. A Tabela 3.1 mostra, na primeira coluna, alguns exemplos de produtos vendidos no supermercado. Analogamente, na segunda coluna são apresentados alguns elementos do documento XML do cenário de cadastro de funcionários que podem sofrer alterações entre as versões sequenciais.

Tabela 3.1: Exemplo de itens vendidos no supermercado e de elementos que podem ser alterados no cadastro de funcionários

Produto	***Employee***
cerveja	*name*
leite	*jobtitle*
pão	*agencyid*
fralda	*agency*
arroz	*hiredate*
feijão	*annualsalary*
macarrão	*grosspay*

Cada compra realizada por um cliente é tratada como uma transação. Cada transação possui os produtos adquiridos pelo cliente. O termo *itemset* é utilizado para o conjunto de produtos comprados por um cliente (os produtos de uma transação). Um *itemset* com *x* ele-

mentos é denominado *x-itemset*. No cenário de cadastro de funcionários, uma transação corresponde às alterações nos elementos de um funcionário efetuadas em conjunto. A Tabela 3.2.a mostra alguns exemplos de transações realizadas pelos clientes no supermercado enquanto algumas transações efetuadas pelos usuários no contexto de cadastro de funcionários da prefeitura de Baltimore são mostradas na Tabela 3.2.b.

Tabela 3.2: Transações no cenário de compras (a) e cadastro de funcionários (b)

(a)

Trans.	Compras
1	cerveja, pão, leite, fralda
2	leite, fralda, cerveja
3	pão, arroz, feijão, leite
4	cerveja, fralda, leite
5	leite, pão, cerveja, arroz

(b)

Trans.	Baltimore
1	*jobtitle*
2	*jobtitle, annualsalary, grosspay*
3	*jobtitle, agencyid, agency*
4	*agencyid, jobtitle, grosspay, annualsalary*
5	*jobtitle, agencyid, agency, grosspay*

Caso o usuário defina que *itemsets* que aparecem em pelo menos 70% de todas as transações sejam considerados frequentes, apenas os *itemsets* cerveja, leite (com 2 elementos); cerveja (1 elemento) e leite (1 elemento) seriam considerados no cenário de compras em um supermercado. No cadastro de funcionários, apenas o *itemset jobtitle* seria considerado. A partir desse resultado, o usuário decide se aumenta ou diminui o limiar que ele considera aceitável para considerar como frequentes outros resultados. No contexto de cadastro de funcionários, considerando que o salário de muitos funcionários muda de uma versão para outra, é interessante descobrir quais outras mudanças são comuns quando isso acontece. Por exemplo, se o valor do limiar fosse reduzido para 60%, o *itemset jobtitle, grosspay* seria considerado, pois muitos funcionários que têm seus salários modificados têm sua função alterada também (o que acontece em 60% dos casos). Essa frequência define a métrica denominada suporte. O suporte é a porcentagem de transações onde o *itemset* aparece (BRIN *et al.*, 1997).

O objetivo do apoio semiautomático no XChange é descobrir elementos do documento XML que mudam em conjunto com frequência, ao se analisar versões sequenciais, e permitir ao especialista construir as regras com base nas sugestões apresentadas. Neste processo, a técnica de regras de associação (AGRAWAL; IMIELIŃSKI; SWAMI, 1993) foi escolhida para apoiar a identificação dos *itemsets* frequentes. O algoritmo selecionado foi o Apriori (AGRAWAL; SRIKANT, 1994) e a ferramenta utilizada como biblioteca de mineração foi a Weka (HALL *et al.*, 2009). O processo ocorre como descrito a seguir.

Os dados de entrada (treinamento) para a mineração são constituídos de versões sequenciais de um documento XML. O algoritmo de *diff* sintático do XChange é aplicado para

cada par de versões do documento XML, visando gerar *deltas*, que informam quais elementos sofreram alterações de uma versão para outra. O interesse está na identificação dos elementos alterados e na forma como eles foram alterados (por exemplo, se um determinado elemento teve seu valor aumentado ou reduzido). Neste caso, um único *delta* consolidado é gerado a partir do *diff* destas versões. Em seguida, é feito um pré-processamento da saída (*delta* consolidado) para que esta possa ser usada pelo algoritmo escolhido.

Após o pré-processamento, o algoritmo Apriori retorna os *itemsets* frequentes identificados, como exemplificado na Tabela 3.3. Nesta tabela o valor associado *y* (*yes*) indica que um elemento de um determinado funcionário foi alterado de uma versão para outra. Os valores associados *u* (*up)* e *d* (*down*) indicam que um elemento foi alterado para um valor maior ou menor, respectivamente, de uma versão para outra. A linha 3 por exemplo, indica que é comum a alteração de *jobtitle* em conjunto com *annualsalary*, o que indica por exemplo que alguns funcionários foram promovidos (regra *promoted* definida na Figura 3.4). Na coluna *Suporte* pode-se observar o número de vezes que o *itemset* foi identificado nas transações.

Tabela 3.3: Exemplos de *itemsets* frequentes encontrados na mineração

	Elementos alterados	Suporte
1	*agencyid=y annualsalary=u*	173
2	*jobtitle=y annualsalary=u grosspay=u*	182
3	*jobtitle=y annualsalary=u*	189
4	*agencyid=y grosspay=u*	190
5	*jobtitle=y grosspay=u*	195
6	*jobtitle=y*	209
7	*agency=y grosspay=d*	252
8	*agencyid=y agency=y*	257
9	*grosspay=d*	269
10	*agencyid=y*	292
11	*agency=y annualsalary=u grosspay=u*	796
12	*agency=y annualsalary=u*	931
13	*agency=y grosspay=u*	1009
14	*annualsalary=u grosspay=u*	1125
15	*annualsalary=u*	1269
16	*agency=y*	1322
17	*grosspay=u*	1383

Após este passo, as sugestões são então repassadas para o especialista a partir da interface gráfica apresentada anteriormente. O especialista analisa este conjunto de *itemsets* frequentes e verifica se estas sugestões têm algum significado relevante, que deva ser capturado através de uma regra de enriquecimento semântico. Neste caso, o especialista

nomeia as sugestões, dando um significado para esta mudança conjunta como ilustrado na Figura 3.6 com a regra *promoted* (Figura 3.4).

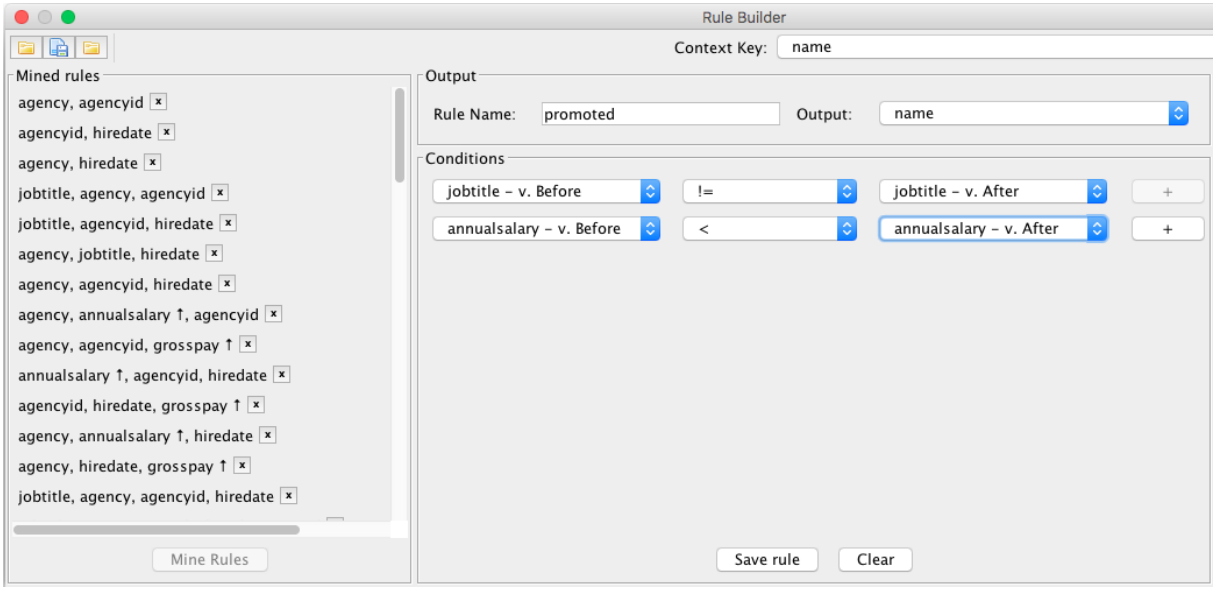

Figura 3.6: Sugestões para a definição de regras de enriquecimento semântico a partir da identificação dos *itemsets* frequentes

Definidas as regras de enriquecimento semântico, de forma manual ou a partir deste apoio semiautomático, estas podem ser usadas no *diff* semântico de qualquer par de versão de um documento XML deste domínio de conhecimento.

3.5 INCLUSÃO DE ID DE SIMILARIDADE

Como mencionado anteriormente, o casamento dos elementos correspondentes pode ser efetuado por similaridade, evitando a necessidade de uso de atributos chave. Para apoiar a identificação das correspondências, a abordagem Phoenix (PINTO; CAMPELLO, 2012; CAMPELLO *et al.*, 2014; OLIVEIRA *et al.*, 2016) foi utilizada. Vale mencionar que outro algoritmo de similaridade poderia ser utilizado nesta tarefa.

O Phoenix foi criado em 2012, no contexto de um trabalho de conclusão de curso de graduação, para abordar o problema de comparação de documentos XML utilizando o conceito de similaridade (PINTO; CAMPELLO, 2012). Em 2013, o Phoenix foi escolhido como algoritmo de similaridade para apoiar a identificação de elementos correspondentes no XChange. Desde então, o Phoenix foi aprimorado no contexto desta tese, no que diz respeito aos pesos configuráveis e ao cálculo de similaridade entre documentos XML baseada em ti-

pos de dados, que serão descritos posteriormente, bem como na solução de problemas relacionados a estouro de memória e reescrita do código.

O Phoenix calcula a similaridade entre documentos em um intervalo que varia de 0 (0% – totalmente dissimilares) a 1 (100% – documentos iguais). Este valor é calculado recursivamente através da comparação dois a dois dos elementos com mesmo pai nas árvores das duas versões do documento XML em análise. Para cada par de elementos, o Phoenix considera a similaridade entre seus nomes, seus atributos, seus conteúdos textuais e seus subelementos, como mostra a Figura 3.7 (CAMPELLO *et al.*, 2014).

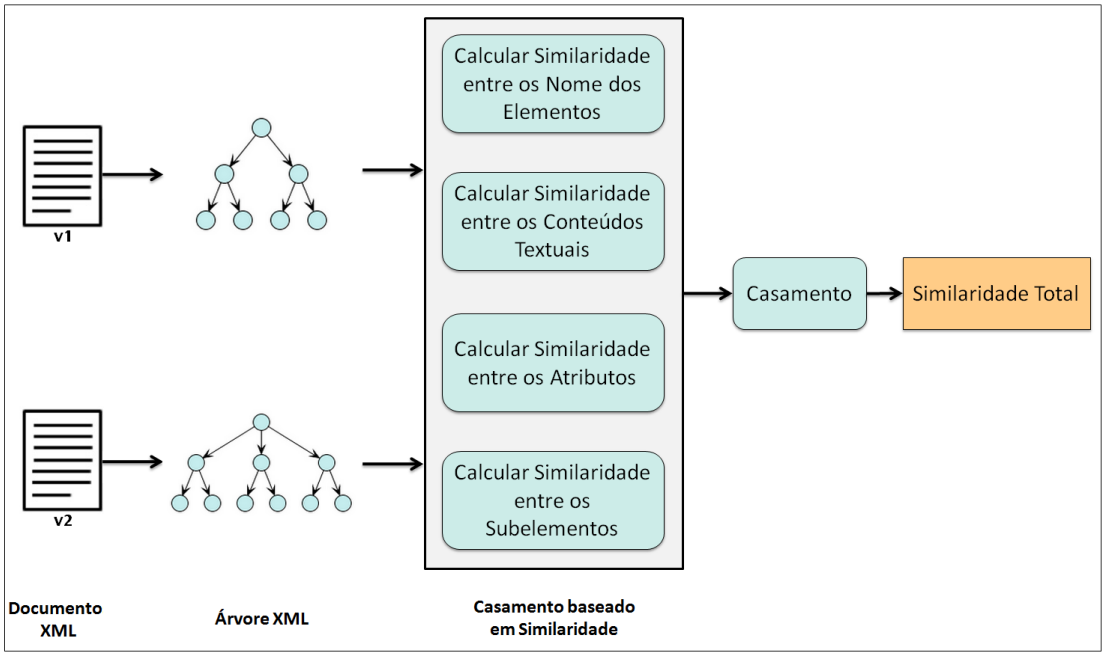

Figura 3.7: Abordagem Phoenix de cálculo de similaridade considerando nomes, atributos, conteúdos textuais e subelementos – adaptada de Campello *et al. (2014)*

3.5.1 ALGORITMO DE SIMILARIDADE

A comparação de dois elementos XML e_1 e e_2 de duas versões *v1* e *v2* é baseada em quatro características, como mencionado anteriormente. A similaridade de cada característica é calculada independentemente e o seu valor representa um componente que contribui para a similaridade geral dos elementos e_1 e e_2 . O componente de similaridade referente a cada uma das quatro características é um número variando entre 0 e 1. A similaridade total de e_1 e e_2 é calculada utilizando uma média ponderada dos componentes de similaridade. O Phoenix utiliza um algoritmo específico para calcular cada componente de similaridade, como mostrado a seguir (CAMPELLO *et al.*, 2014).

Para a similaridade entre os nomes dos elementos, *NS(e_1, e_2)*, adota-se o seguinte procedimento. Os nomes dos elementos são *strings* que representam o tipo do elemento. No contexto desta tese, os nomes devem ser iguais, ou seja, assume-se que as versões do documento XML possuem a mesma estrutura, mesmo que o documento não tenha esquema associado, uma vez que a evolução de esquema não está no escopo deste trabalho. Assim, os nomes de e_1 e e_2 são comparados para verificar a igualdade, e o cálculo de similaridade é interrompido se eles forem diferentes.

No cálculo da similaridade entre os conteúdos dos elementos, *CS(e_1, e_2)*, por padrão, o conteúdo é tratado como *string* e o algoritmo LCS (MAIER, 1978) é utilizado. A similaridade resultante é o tamanho da LCS entre as duas *strings* dividido pela média do tamanho delas. Entretanto, uma parametrização definida na extensão do Phoenix realizada no contexto dessa tese (OLIVEIRA *et al.*, 2016) permite que o algoritmo reconheça o tipo de dado do conteúdo de e_1 e e_2 (*booleano*, número ou datas, inferidos com base no valor) e aplique um algoritmo diferente para cada caso:

- para *booleanos*, a similaridade é 100% se os valores forem iguais, ou 0% caso contrário;
- para números, aplica-se a fórmula descrita na Equação (1 para obter a similaridade do conteúdo. Nessa fórmula, *max(x,y)* e *min(x,y)* representam o valor máximo e mínimo entre *x* e *y*, respectivamente, e *abs(x)* representa o valor absoluto de *x*. De acordo com essa fórmula, a similaridade é inversamente proporcional à diferença entre os números;

$$CS(e_1, e_2) = \begin{cases} 1, & se\ e_1 = e_2 \\ 1 - \dfrac{abs(e_1 - e_2)}{max\big(abs(e_1), abs(e_2)\big)}, & se\ min(e_1, e_2) \geq 0\ ou\ max(e_1, e_2) \leq 0 \\ 0, & se\ min(e_1, e_2) < 0\ e\ max(e_1, e_2) > 0 \end{cases} \quad \textbf{(1)}$$

- para datas, os valores são convertidos em *timestamp* (número em segundos desde 1º de janeiro de 1970, 00:00:00 GMT). Uma vez que os valores tornam-se números após a conversão, a mesma fórmula da Equação (1 é usada para calcular a similaridade de datas;
- se o conteúdo não pode ser interpretado como *booleano*, número, ou data, ele é considerado uma *string* e o algoritmo LCS é utilizado.

No cálculo da similaridade entre os atributos, *AS(e_1, e_2)*, são usados os seguintes passos:

1. extrai-se o conjunto completo de nomes de atributos usados em ambos os elementos e_1 e e_2;
2. para cada nome de atributo identificado, seu valor é comparado em ambos os elementos. Esta comparação usa a mesma abordagem da similaridade de conteúdo: considera o atributo como *string* ou tenta interpretá-lo como *booleano*, número ou data. Se o atributo não está presente em um dos elementos, sua similaridade é definida como 0%. Vale ressaltar que os valores são comparados somente quando o nome do atributo é o mesmo nos dois elementos;
3. soma-se todas as similaridades calculadas nos passos anteriores e divide-se pelo número de atributos no conjunto completo.

Para calcular o componente de similaridade entre os subelementos, $SES(e_1,e_2)$, outro procedimento é usado:

1. cada subelemento de e_1 é comparado com cada subelemento de e_2, usando recursivamente os mesmos algoritmos de similaridade, e os resultados são computados em uma matriz denominada *matriz de similaridade*;
2. a matriz de similaridade é então fornecida ao algoritmo Húngaro (KUHN, 1955), que indica o casamento ótimo global entre os subelementos. Este passo é realizado, no máximo, uma vez por chamada recursiva;
3. a partir do melhor casamento obtido na etapa anterior, o componente de similaridade de subelementos é calculado dividindo a soma de todas as similaridades pela ordem da matriz. Similaridades que estão abaixo de um limiar previamente definido são zeradas, uma vez que este valor representa o grau mínimo de similaridade entre dois elementos para serem considerados correspondentes. Decidiu-se aplicar o limiar durante esta etapa e não no primeiro passo para fornecer ao algoritmo Húngaro a maior quantidade de informações possível.

Finalmente, os quatro componentes de similaridade são combinados usando uma média ponderada, que resulta na similaridade entre os dois elementos XML e_1 e e_2. Para cada componente de similaridade, há um peso configurável (w_n para nome, w_c para conteúdo, w_a para atributo e w_s para subelemento). A fórmula geral da similaridade é dada pela Equação (2.

$$Similaridade(e_1, e_2) = \frac{w_n * NS(e_1, e_2) + w_c * CS(e_1, e_2) + w_a * AS(e_1, e_2) + w_s * SES(e_1, e_2)}{w_n + w_c + w_a + w_s} \textbf{(2)}$$

3.5.2 EXEMPLO DE UTILIZAÇÃO

A Figura 3.8 apresenta duas versões *v1* e *v2* de um documento XML no contexto da prefeitura de Baltimore. Neste exemplo simplificado, somente os subelementos <*name*> e <*grosspay*> de <*employee*> foram mantidos. Além disso, os elementos <*hiredate*> e <*agencyid*> foram transformados em atributos para melhor ilustrar a técnica. Por fim, o atributo *agencyid* foi adicionado somente em *v2*. Neste exemplo, considerou-se o limiar de similaridade em 70%, todos os componentes de similaridade com o mesmo peso (25%) e permitiu-se que o algoritmo detectasse o tipo de dados na comparação de conteúdo e atributos.

A fim de comparar *v1* e *v2*, o algoritmo de similaridade começa pela comparação do elemento raiz dos documentos. Ambos têm o mesmo nome (<*employee*>), como a abordagem requer. Assim, o resultado da similaridade de nome é 100%. Nenhum dos elementos raiz tem conteúdo. Assim, a similaridade de conteúdo é de 100% também. Estas comparações vazias são chamadas de similaridades triviais na abordagem, e a forma como as similaridades triviais são tratadas (ou valendo 100% ou sendo ignoradas) pode ser configurada.

<employee hiredate="10/24/1979"> <name>Aaron,Pat</name> <grosspay>45505.94</grosspay> </employee>	<employee hiredate="10/24/1979" **agencyid="A03031"**> <name>Aaron,Pat**ricia G**</name> <grosspay>**52247.39**</grosspay> </employee>

Figura 3.8: Versões do documento XML, *v1* à esquerda e *v2* à direita

O próximo passo consiste em calcular a similaridade entre os atributos. Para isso, o Phoenix considera todos os atributos dos elementos raiz e cria um conjunto de atributos. Neste exemplo, este conjunto tem os atributos *hiredate* e *agencyid*. Cada atributo a partir deste conjunto é então avaliado e os seus valores são verificados para identificar se são *booleano*s, números ou datas. O primeiro atributo, presente em *v1* e *v2*, é *hiredate*. O valor do atributo *hiredate* em *v1* e *v2* é interpretado como data com sucesso. Assim, a similaridade é calculada convertendo-a para *timestamp* e aplicando a fórmula de similaridade de número. Dado que o atributo *hiredate* de *v1* e de *v2* contém exatamente a mesma *string*, o número convertido é igual para ambos e a similaridade para este atributo resulta em 100%. O segundo atributo é *agencyid*, que existe somente em *v2*. Neste caso a similaridade é 0%. A similaridade final é calculada com base na média simples de todas as similaridades calculadas a partir do conjunto de atributos. Neste exemplo, tem-se 50% de similaridade para este componente.

O próximo passo é calcular a similaridade entre os subelementos. Os subelementos de *v1* e *v2* são S_1 = {<*name*>*Aaron,Pat*</*name*>, <*grosspay*>*45505.94*</*grosspay*>} e S_2 =

{<*name*>*Aaron,Patricia G*</*name*>, <*grosspay*>*52247.39*</*grosspay*>} respectivamente. O Phoenix compara cada subelemento de S_1 com cada um dos subelementos de S_2, armazenando o resultado em uma matriz de similaridade (Tabela 3.4). A semelhança entre <*name*> e <*grosspay*> é 0%, porque os nomes são diferentes e, portanto, neste caso, nenhum outro elemento é calculado. Para <*name*>*Aaron, Pat*</ *name*> e <*name*>*Aaron, Patricia G*</ *name*>, o nome é igual, então a similaridade do componente nome é de 100%. O seu conteúdo é diferente, e a similaridade de conteúdo, neste caso, é calculada com base no LCS, resultando no valor de 72% $(9/((9+16)/2))$. Ambos não têm atributos, de modo que a similaridade do componente atributo é de 100%. Eles também não têm subelementos, o que resulta em uma similaridade do componente subelemento de 100%.

Tabela 3.4: Matriz de similaridade de subelementos

		S_2	
		<name>	***<grosspay>***
S_1	***<name>***	0,930	0,000
	<grosspay>	0,000	0,968

A similaridade total, neste caso, é 93% $(100\% + 72\% + 100\% + 100\%/4)$. Na comparação dos elementos <*grosspay*>*45505.94*</*grosspay*> e <*grosspay*>*52247.39* </*grosspay*>, a abordagem é capaz de detectar que o conteúdo é um número e aplicar o cálculo específico para a similaridade de número, resultando em uma similaridade de 96,8% quando todos os componentes de similaridade são considerados. O algoritmo Húngaro processa a matriz de similaridade e produz o melhor casamento entre os elementos. Neste exemplo, é fácil identificar que <*name*> de *v1* e <*name*> de *v2* são elementos correspondentes. O mesmo ocorre para <*grosspay*>. Depois de sumarizar as similaridades dos elementos casados e dividi-los pela ordem da matriz, tem-se que a similaridade final de subelementos é de 94,9%, neste exemplo. Finalmente, considerando-se todas as similaridades parciais mencionadas, pode-se calcular a similaridade geral dos elementos raiz de *v1* e *v2* e, consequentemente, a similaridade geral de *v1* e *v2*, que é 86,2% $((100\% + 100\% + 50\% + 94.9\%)/4)$.

3.5.3 PARAMETRIZAÇÃO

O limiar de similaridade ajusta a sensibilidade do método. Como tal, indica o grau de similaridade mínimo necessário para considerar dois elementos XML similares. Se a similaridade calculada for menor do que o limiar, os elementos são tratados como dissimilares e o Phoenix não efetua o casamento. No exemplo do item anterior, se o limiar fosse alterado para 94%, a similaridade global diminuiria de 86,2% para 74,6%. Isso acontece porque durante os

cálculos de similaridade entre os subelementos de <*employee*> há similaridades que se enquadram abaixo desse valor (Tabela 3.4), afetando assim a similaridade de subelementos de <*employee*> (Equação (2) e, consequentemente, o grau de similaridade entre as duas versões.

Embora a distribuição de peso igual entre os componentes de similaridade possa parecer razoável, há vários cenários que não apresentam bons resultados com essa estratégia. Por exemplo, se as versões do documento XML que está sendo comparado têm a mesma estrutura, pode-se diminuir ou eliminar o peso da similaridade de nome dos elementos, enfatizando as diferenças de conteúdo, atributos e subelementos. Por este motivo, os pesos da fórmula de similaridade (Equação (2) são configuráveis. No exemplo anterior apresentado na Figura 3.8), se ao invés de usar uma distribuição com pesos iguais, tivesse sido considerado $w_n = 0,0$, $w_c = 0,35$, $w_a = 0,4$ and $w_s = 0,25$, a similaridade total teria diminuído de 86,2% para 78,2%.

Outra situação que pode alterar o cálculo de similaridade total é a presença de similaridades triviais. No exemplo anterior, a similaridade dos elementos raiz é aumentada em 25% em relação à similaridade de conteúdo porque ambos os elementos não têm conteúdo textual. Isto pode não ser adequado em determinadas situações, especialmente quando se quer focar nas diferenças. Em função disso, um parâmetro para ignorar as similaridades triviais foi criado. Se ativado, o algoritmo descarta os componentes de similaridade identificados como triviais. Vale ressaltar que o peso dos componentes de similaridade descartados também fica de fora da fórmula. No exemplo, caso a abordagem fosse configurada para ignorar as similaridades triviais, a similaridade geral diminuiria de 86,2% para 64,8%. Como <*employee*> não tem conteúdo, o resultado de $CS(e_1,e_2)$ seria mantido fora do cálculo da similaridade de elemento, bem como o seu peso w_c. No entanto, como existem subelementos e atributos, os resultados de $SES(e_1,e_2)$ e de $AS(e_1,e_2)$ seriam considerados, bem como os seus pesos.

Como já mencionado, os cálculos de similaridade de conteúdo e de atributo podem ser ajustados para detectar o tipo de dados (*booleano*, número ou data), ou considerá-lo sempre como uma *string* e usar o algoritmo LCS para o cálculo da similaridade. Isto pode ser feito utilizando um parâmetro configurável chamado similaridade de tipo de dados. No exemplo, se essa detecção fosse desativada e todos os valores fossem considerados como *strings*, a similaridade geral resultante diminuiria de 86,2% para 84,7%.

3.5.4 USO DO PHOENIX NO XCHANGE

Apresentado o cálculo de similaridade, pode-se exemplificar a sua utilização no contexto da Prefeitura de Baltimore. Dadas as versões *v1* (Figura 3.9) e *v2* (Figura 3.10), o Phoenix calcula a similaridade entre os elementos usando o cálculo mostrado anteriormente.

```
<government>
 <employee>
  <name>Berube,Leslie A</name>
  <jobtitle>COORDINATOR</jobtitle>
  <agencyid>A50701</agencyid>
  <agency>DPW-Water </agency>
  <hiredate>1978-06-26</hiredate>
  <annualsalary>50981</annualsalary>
  <grosspay>48956.35</grosspay>
 </employee>
 <employee>
  <name>Bond,Filishia M</name>
  <jobtitle>PARALEGAL</jobtitle>
  <agencyid>A06019</agencyid>
  <agency>Housing Com</agency>
  <hiredate>2001-06-25</hiredate>
  <annualsalary>50364</annualsalary>
  <grosspay>44941.01</grosspay>
 </employee>
 <employee>
  <name>Bailowitz,Anne</name>
  <jobtitle>EXECUTIVE</jobtitle>
  <agencyid>A65527</agencyid>
  <agency>HLTH-Health Dept</agency>
  <hiredate>2001-02-26T00:00:00</hiredate>
  <annualsalary>119000</annualsalary>
  <grosspay>103290.62</grosspay>
 </employee>
</government>
```

Figura 3.9: Versão *v1.xml*

```
<government>
 <employee>
  <name>Blow,Teresa L</name>
  <jobtitle>MOTOR DRIVER</jobtitle>
  <agencyid>B49330</agencyid>
  <agency>TRANS-Highways</agency>
  <hiredate>2004-06-14</hiredate>
  <annualsalary>30742</annualsalary>
  <grosspay>31222.54</grosspay>
 </employee>
 <employee>
  <name>Berube,Leslie A</name>
  <jobtitle>ASSISTANT</jobtitle>
  <agencyid>A49101</agencyid>
  <agency>TRANS-Highways </agency>
  <hiredate>1978-06-26T00:00:00</hiredate>
  <annualsalary>55811</annualsalary>
  <grosspay>56025.54</grosspay>
 </employee>
 <employee>
  <name>Barnes,Ikea T</name>
  <jobtitle>AIDE BLUE CHIP</jobtitle>
  <agencyid>W02235</agencyid>
  <agency>Youth Summer </agency>
  <hiredate>2010-06-03T00:00:00</hiredate>
  <annualsalary>11310</annualsalary>
  <grosspay>1051.25</grosspay>
 </employee>
 <employee>
  <name>Bond,Filishia M</name>
  <jobtitle>EXECUTIVE</jobtitle>
  <agencyid>A06019</agencyid>
  <agency>Housing Com</agency>
  <hiredate>2001-06-25</hiredate>
  <annualsalary>52912</annualsalary>
  <grosspay>53047.47</grosspay>
 </employee>
</government>
```

Figura 3.10: Versão *v2.xml*

Com base neste cálculo, os elementos correspondentes são encontrados e *v1* e *v2* são modificadas, recebendo elementos ID (chamados *xchangeid*) com a função de identificadores artificiais. Na linha 3 da Figura 3.11, por exemplo, foi inserido o elemento *xchangeid* com valor 1, nas duas versões sequenciais, indicando que existe um funcionário em *v1* que também está presente em *v2*. O *xchangeid* 2 foi inserido também em *v1* e *v2* (linha 13). Já o *xchangeid* 3 foi inserido somente em *v1* enquanto os *xchangeids* com valor 4 e 5 foram inseridos somente em *v2*, o que indica que tais empregados estão presentes apenas em uma das versões sequenciais.

```
1  <government>
2   <employee>
3    <xchangeid>1</xchangeid>
4    <name>Berube,Leslie A</name>
5    <jobtitle>COORDINATOR </jobtitle>
6    <agencyid>A50701</agencyid>
7    <agency>DPW-Water </agency>
8    <hiredate>1978-06-26</hiredate>
9    <annualsalary>50981 </annualsalary>
10   <grosspay>48956.35</grosspay>
11  </employee>
12  <employee>
13   <xchangeid>2</xchangeid>
14   <name>Bond,Filishia M</name>
15   <jobtitle>PARALEGAL</jobtitle>
16   <agencyid>A06019</agencyid>
17   <agency>Housing Com</agency>
18   <hiredate>2001-06-25</hiredate>
19   <annualsalary>50364</annualsalary>
20   <grosspay>44941.01</grosspay>
21  </employee>
21  <employee>
22   <xchangeid>3</xchangeid>
23   <name>Bailowitz,Anne</name>
24   <jobtitle>EXECUTIVE</jobtitle>
25   <agencyid>A65527</agencyid>
26   <agency>HLTH-Health Dept</agency>
27   <hiredate>2001-02-26T00:00:00
28 </hiredate>
29   <annualsalary>119000 </annualsalary>
30   <grosspay>103290.62</grosspay>
31  </employee>
32 </government>
33
```

(a) *v1.xml*

```
<government>
 <employee>
  <xchangeid>1</xchangeid>
  <name>Berube,Leslie A</name>
  <jobtitle>ASSISTANT</jobtitle>
  <agencyid>A49101</agencyid>
  <agency>TRANS-Highways </agency>
  <hiredate>1978-06-26</hiredate>
  <annualsalary>55811 </annualsalary>
  <grosspay>56025.54</grosspay>
 </employee>
 <employee>
  <xchangeid>2</xchangeid>
  <name>Bond,Filishia M</name>
  <jobtitle>EXECUTIVE</jobtitle>
  <agencyid>A06019</agencyid>
  <agency>Housing Com</agency>
  <hiredate>2001-06-25</hiredate>
  <annualsalary>52912</annualsalary>
  <grosspay>53047.47</grosspay>
 </employee>
 <employee>
  <xchangeid>4</xchangeid>
  <name>Blow,Teresa L</name>
   ...
  <grosspay>31222.54</grosspay>
 </employee>
 <employee>
  <xchangeid>5</xchangeid>
  <name>Barnes,Ikea T</name>
  ...
   <grosspay>1051.25</grosspay>
 </employee>
</government>
```

(b) *v2.xml*

Figura 3.11: Versões *v1* e *v2* após o cálculo de similaridade com a inserção do identificador artificial

3.6 TRADUÇÃO XML PARA PROLOG

Para viabilizar a realização das inferências é necessário traduzir os dados contidos no documento XML para uma linguagem que forneça esta capacidade, tal como Datalog (HUANG, SHAN SHAN; GREEN; LOO, 2011), RuleML (BOLEY, 2003) ou Prolog (BRATKO, 2001). Datalog é uma linguagem de consulta não procedural baseada em Prolog, que surgiu da combinação de programação em lógica com banco de dados. O fato de Datalog

não permitir termos complexos como argumento do predicado e possuir restrições no uso de negação e recursividade, tornou-a incompatível com o XChange. A linguagem RuleML é o resultado de um esforço para fornecer um padrão de definição de regras na *Web* e descreve tanto as informações como os seus relacionamentos, tornando possível a realização de inferência. Por RuleML ser uma linguagem de marcação voltada para representação de regras de inferência com foco na *Web*, e suas implementações atuais utilizarem uma máquina Datalog como mecanismo de inferência, sua adoção foi descartada. Consequentemente optou-se por Prolog para estabelecer relações a partir de documentos XML. De fato, a linguagem Prolog já foi utilizada com sucesso para realizar consultas com inferência a documentos XML (LIMA *et al.*, 2012; SANTOS, 2015; MACHADO, 2016).

Diante disso, o XChange estende o método de tradução original (LIMA *et al.*, 2012) para utilizar duas versões de um documento XML (neste caso, *before* e *after*) como entrada e gerar os fatos correspondentes. O processo de tradução gera fatos Prolog a partir de um único documento XML, transformando elementos em predicados e seus conteúdos em constantes. Para exemplificar, no cenário de cadastro de funcionários da Prefeitura de Baltimore, o processo de tradução dos trechos das versões *v1* (Figura 3.9) e *v2* (Figura 3.10) é mostrado na Figura 3.12. Este processo é realizado em três passos descritos a seguir: tradução da raiz, dos elementos complexos e dos elementos simples sem atributos.

O primeiro passo traduz a raiz de *v1* <*government*> (linha 1 da Figura 3.9) em um fato com o nome do elemento e argumento único igual a um identificador gerado para estabelecer o vínculo com seus elementos filhos (*government(before)*), como mostra a Figura 3.12.a na linha 1. Para relacionar predicados distintos, são criadas constantes Prolog que atuam como identificadores que preservam a ligação pai/filho entre os elementos XML correspondentes.

O segundo passo é a tradução dos elementos complexos, ou seja, aqueles que tem outros elementos como filho, como por exemplo, <*employee*>, na Figura 3.9, linha 2. Um novo identificador é criado para relacionar os filhos ao pai. O resultado é a geração de um fato com o nome do elemento e dois argumentos: o identificador do pai e um novo identificador gerado para referenciá-lo, como por exemplo, *employee(before, 2)* na Figura 3.12.a, linha 2.

O terceiro passo traduz os elementos simples sem atributos. Um exemplo é mostrado, na linha 3 da Figura 3.9 (<*name*>*Berube,Leslie A*</*name*>). O resultado é um fato com nome igual ao nome do elemento e argumentos iguais ao identificador do elemento pai, o identificador do elemento corrente e o conteúdo do elemento corrente, como mostra a linha 3 da Figura 3.12.a (*name(2, 3, 'Berube,Leslie A')*).

```
1  government(before).
2  employee(before, 2).
3  name(2, 3, 'Berube,Leslie A').
4  jobtitle(2, 5, 'COORDINATOR').
5  agencyid(2, 7, 'A50701').
6  agency(2, 9, 'DPW-Water ').
7  hiredate(2, 11, '1978-06-26').
8  annualsalary(2, 13, 50981.0).
9  grosspay(2, 15, 48956.35).
10 employee(before, 17).
11 name(17, 18, 'Bond,Filishia M').
12 jobtitle(17, 20, 'PARALEGAL').
13 agencyid(17, 22, 'A06019').
14 agency(17, 24, 'Housing Com').
15 hiredate(17, 26, '2001-06-25').
16 annualsalary(17, 28, 50364.0).
17 grosspay(17, 30, 44941.01).
18 employee(before, 32).
19 name(32, 33, 'Bailowitz,Anne').
20 jobtitle(32, 35, 'EXECUTIVE').
21 agencyid(32, 37, 'A65527').
22 agency(32, 39, 'HLTH-Health Dept').
23 hiredate(32, 41, '2001-02-26T00:00:00').
24 annualsalary(32, 43, 119000.0).
25 grosspay(32, 45, 103290.62).
```

(a) *v1.pl*

```
government(after).
employee(after, 48).
name(48, 49, 'Blow,Teresa L').
jobtitle(48, 51, 'MOTOR DRIVER').
agencyid(48, 53, 'B49330').
agency(48, 55, 'TRANS-Highways').
hiredate(48, 57, '2004-06-14').
annualsalary(48, 59, 30742.0).
grosspay(48, 61, 31222.54).
employee(after, 63).
name(63, 64, 'Berube,Leslie A').
jobtitle(63, 66, 'ASSISTANT').
agencyid(63, 68, 'A49101').
agency(63, 70, 'TRANS-Highways ').
hiredate(63, 72, '1978-06-
26T00:00:00').
annualsalary(63, 74, 55811.0).
grosspay(63, 76, 56025.54).
employee(after, 78).
name(78, 79, 'Barnes,Ikea T').
jobtitle(78, 81, 'AIDE BLUE CHIP').
agencyid(78, 83, 'W02235').
agency(78, 85, 'Youth Summer ').
hiredate(78, 87, '2010-06-
03T00:00:00').
annualsalary(78, 89, 11310.0).
grosspay(78, 91, 1051.25).
employee(after, 93).
name(93, 94, 'Bond,Filishia M').
jobtitle(93, 96, 'EXECUTIVE').
agencyid(93, 98, 'A06019').
agency(93, 100, 'Housing Com').
hiredate(93, 102, '2001-06-25').
annualsalary(93, 104, 52912.0).
grosspay(93, 106, 53047.47).
```

(b) *v2.pl*

Figura 3.12: Fatos Prolog gerados a partir das versões *v1* e *v2*

3.7 INFERÊNCIA

Para finalizar, o processamento do *diff* semântico é efetuado via **Inferência**. Uma base de conhecimento é construída a partir dos fatos e das regras de casamento e de enriquecimento semântico gerados nos passos anteriores. Essa base de conhecimento é submetida a consultas usando as cabeças de cada uma das regras, que fornecem como resposta um *delta* semântico contendo os elementos do documento XML que se encaixaram nas situações modeladas pelas regras. Em outras palavras, este *delta* corresponde à razão da evolução do documento XML, de uma versão anterior para uma posterior.

Para exemplificar, a Figura 3.13 mostra um fragmento do resultado da aplicação das regras definidas na Figura 3.4 sobre os fatos gerados e apresentados na Figura 3.12.

```
<diff-set from="v1 - Wed Nov 04 16:51:53" to="v2 - Fri Nov 04 16:51:52">
 <diff name="salary_increased">
  <description>
   <change attr="annualsalary" type="increased"/>
  </description>
  <delta count="2" annualsalary="7378">
   <employee name="Berube,Leslie A">
    <annualsalary before="50981" after="55811" delta="4830"/>
   </employee>
   <employee name="Bond,Filishia M">
    <annualsalary before="50364" after="52912" delta="2548"/>
   </employee>
  </delta>
 </diff>
 <diff name="fired">
  <description>
   <change attr="name" type="deleted"/>
  </description>
  <delta count="1" annualsalary="-119000" grosspay="-103290.62">
   <employee name="Bailowitz,Anne">
    <jobtitle>EXECUTIVE</jobtitle>
    <agencyid>A65527</agencyid>
    <agency>HLTH-Health Dept</agency>
    <hiredate>2001-02-26T00:00:00</hiredate>
    <annualsalary>119000</annualsalary>
    <grosspay>103290.62</grosspay>
   </employee>
  </delta>
 </diff>
 <diff name="promoted">
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="annualsalary" type="increased"/>
  </description>
  <delta count="2" annualsalary="7378">
   <employee name="Berube,Leslie A">
    <jobtitle before="COORDINATOR" after="ASSISTANT"/>
    <annualsalary before="50981" after="55811" delta="4830"/>
   </employee>
   <employee name="Bond,Filishia M">
    <jobtitle before="PARALEGAL" after="EXECUTIVE"/>
    <annualsalary before="50364" after="52912" delta="2548"/>
   </employee>
  </delta>
 </diff>
 <diff name="promoted_transferred">
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agencyid" type="different"/>
   <change attr="annualsalary" type="increased"/>
  </description>
  <delta count="1" annualsalary="4830">
   <employee name="Berube,Leslie A">
    <jobtitle before="COORDINATOR" after="ASSISTANT"/>
    <agencyid before="A50701" after="A49101"/>
    <annualsalary before="50981" after="55811" delta="4830"/>
   </employee>
  </delta>
 </diff>
 ...
</diff-set>
```

Figura 3.13: *Delta* semântico

Como se pode observar, *Berube,Leslie A* (linha 7) e *Bond,Filishia M* (linha 10) receberam um aumento (linha 2). Além disso, os mesmos dois funcionários (linhas 36 e 40) foram promovidos (linha 30) mas apenas o funcionário *Berube,Leslie A* (linha 53) foi promovido e transferido (linha 46). Com isso, é possível identificar o motivo da evolução de um documento XML. Além disso, o *delta* fornece algumas sumarizações ao usuário. A linha 6, por exemplo, mostra o total de funcionários identificados pela regra (*count = 2*), bem como o somatório dos aumentos concedidos (*annualsalary= "7378"*). Já na linha 8 é possível visualizar o salário do funcionário nas duas versões analisadas e a diferença entre estes valores.

3.8 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o XChange, desenvolvido para viabilizar a identificação da razão real das modificações em versões de documentos XML, tomando como base a análise das modificações sintáticas granulares em atributos e elementos. Para isto, o XChange utiliza um mecanismo de inferência baseado emProlog. É importante mencionar que o XChange pode efetuar a análise de versões sequenciais não necessariamente consecutivas. Um problema relacionado é que pode ocorrer perda semântica neste caso. Para exemplificar, considere a situação de um empregado que foi demitido e depois novamente contratado. Ao se utilizar versões não consecutivas, não é possível acompanhar a real evolução deste empregado.

O XChange trabalha na identificação dos elementos correspondentes de duas formas: usando casamento por chave ou casamento por similaridade. Na abordagem de casamento por chave, o usuário deve indicar um atributo-chave. Dependendo da forma como os documentos XML são gerenciados, não há garantia de que o valor de atributos-chave permanece o mesmo entre as versões (por exemplo, pode acontecer um erro de digitação no valor do atributo-chave que é corrigido numa versão). Para atenuar esse problema, uma abordagem baseada em análise de similaridade foi contemplada.

Além disso, o XChange não exige que o usuário seja especialista em Prolog, uma vez que usa uma interface que gera as regras Prolog a partir de uma seleção de opções em um alto nível de abstração. Por fim, as regras geradas são válidas para todos os documentos do mesmo domínio, o que faz com que essa etapa ocorra uma única vez para um dado domínio.

Outra funcionalidade do XChange é o apoio semiautomático para a construção de regras de enriquecimento semântico baseado na mineração de elementos alterados em conjunto com frequência. Através do Apriori, algumas regras de enriquecimento semântico são identificadas e informadas ao especialista de domínio para que sejam validadas e nomeadas.

CAPÍTULO 4 – ESTUDO EXPERIMENTAL I: CASAMENTO DE ELEMENTOS CORRESPONDENTES

4.1 INTRODUÇÃO

O *diff* semântico proposto pelo XChange é baseado no casamento de elementos correspondentes. Algumas abordagens tradicionais comparam versões de documentos XML usando um identificador para fazer a correspondência de elementos. Em algumas situações isto não é viável. Há casos em que não é possível definir um identificador. De fato, a maioria dos documentos XML não possui um esquema associado e nem um elemento identificador (MAAROUF; CHUNG, 2008; VYHNANOVSKÁ; MLÝNKOVÁ, 2010; GRIJZENHOUT; MARX, 2013). Em outras situações, não há garantia de que os valores-chave continuarão os mesmos em todas as versões. O XChange oferece uma alternativa ao casamento de elementos correspondentes que é aplicável mesmo nessas situações, baseada no cálculo de similaridade proposto pelo Phoenix (OLIVEIRA *et al.*, 2016). Por outro lado, casamentos incorretos impactam o resultado do *diff* semântico. Sendo assim é preciso avaliar a qualidade dos casamentos efetuados pelo Phoenix, de forma a garantir o funcionamento correto do *diff* semântico do XChange. Diante disso, este capítulo avalia a eficácia e a eficiência alcançadas pela abordagem Phoenix, no que diz respeito à identificação dos elementos correspondentes de versões de um documento XML e compara os resultados com duas abordagens da literatura que foram usadas como *baseline*. O X-Diff (WANG; DEWITT; CAI, 2003) e o XyDiff (COBENA; ABITEBOUL; MARIAN, 2002) foram escolhidos devido à sua importância na área, sendo as abordagens de *diff* de XML mais citadas: 455 e 564 citações, respectivamente. Estas abordagens serviram de base para vários trabalhos, inclusive alguns mencionados no Capítulo 2. Além disso, outro fator que importante é que estes trabalhos têm implementação disponível na Web.

Neste estudo experimental, a eficácia é calculada em termos da *F-Measure*, uma métrica tradicional na área de Recuperação de Informação (BAEZA-YATES; RIBEIRO-NETO, 1999). A *F-Measure* é a média harmônica entre a precisão e a cobertura, com o objetivo de encontrar o melhor compromisso entre a correção e a completude do resultado. No contexto desta tese, a precisão indica quão corretos estão os casamentos identificados pela abordagem. Por outro lado, a cobertura indica o quão completa a relação de elementos correspondentes encontrados pela abordagem está, quando comparada aos resultados esperados. Finalmente, a

eficiência é calculada em termos do número de casamentos corretos por segundo. Neste estudo, o objetivo consiste, portanto, em responder às seguintes questões de pesquisa:

E1-QP1. O método baseado em cálculo de similaridade do XChange é mais eficaz na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

E1-QP2. O método baseado em cálculo de similaridade do XChange é mais eficiente na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

Diante disso, este capítulo está organizado da seguinte forma. Na Seção 4.2, o documento XML utilizado no estudo experimental é apresentado. A Seção 4.3 descreve a análise de sensibilidade para calibrar a abordagem Phoenix e definir o limiar de similaridade base para este domínio. A Seção 4.4 mostra o processo utilizado no estudo experimental. A Seção 4.5 avalia a eficácia e a eficiência obtida pelas abordagens utilizadas no estudo experimental. Finalmente, a Seção 0 apresenta as ameaças à validade do estudo experimental enquanto a Seção 4.7 apresenta as considerações finais deste capítulo.

4.2 DESCRIÇÃO DO DOCUMENTO XML

Durante o estudo, foi utilizado o documento XML contendo informações sobre os salários dos funcionários da prefeitura de Baltimore (MAYOR'S OFFICE OF INFORMATION TECHNOLOGY, 2016) apresentado na Seção 1.1. Este documento é considerado representativo (MIGNET; BARBOSA; VELTRI, 2003; BARBOSA; MIGNET; VELTRI, 2005), pois possui 3 níveis de profundidade e não contém atributos. Neste estudo foram utilizadas 5 versões deste documento, uma para cada ano: 2011, 2012, 2013, 2014 e 2015. Por conveniência, as versões foram denominadas *v1* a *v5*, respectivamente. A Tabela 4.1 mostra o número de empregados (#empregados) e o tamanho em KBytes (tamanho (KB)) de cada versão. Além disso, mostra a data de início e de fim, que define o período considerado em cada versão.

Tabela 4.1: Caracterização do documento XML da prefeitura de Baltimore

Versão	#empregados	tamanho (KB)	data inicial	data final
v1	13.966	3.806	01/07/2010	24/05/2011
v2	15.514	4.231	01/07/2011	30/06/2012
v3	18.372	5.117	01/07/2012	30/06/2013
v4	18.318	4.968	01/07/2013	30/06/2014
v5	13.495	3.827	01/07/2014	30/06/2015

A fim de maximizar o número de cenários neste estudo, cada versão foi dividida em 15 fragmentos, numerados de 0 a 14. A Tabela 4.2 mostra o número de empregados (colunas *#emp*) e o tamanho em KBytes (colunas *tam*) de cada fragmento. O elemento <*name*> foi utilizado como parâmetro para a fragmentação horizontal (ANDRADE *et al.*, 2006) para manter os funcionários nos mesmos fragmentos em todas as versões. O fragmento 0 contém os funcionários cujos nomes começam com a letra *A*. O fragmento 1 contém os funcionários com nomes começando com a letra *B* até *Bo* e assim por diante (coluna *critério*). A Figura 1.1 e a Figura 1.2, apresentadas anteriormente, mostram como a informação está estruturada em um fragmento.

Tabela 4.2: Características dos fragmentos (tamanho em Kb)

frag	critério	v1		v2		v3		v4		v5	
		#emp	tam	#emp	tam	#emp	tam	#emp	tam	#emp	tam
0	A	446	121	513	140	647	180	628	170	471	133
1	B até Bo	781	213	876	238	1.054	293	1.038	281	775	219
2	Bp até Bz	600	163	705	192	896	249	868	234	595	168
3	C	1.049	286	1.157	316	1.410	393	1.360	369	1.021	290
4	D até E	968	263	1.057	287	1.229	341	1.191	323	916	259
5	F até G	1.268	345	1.375	375	1.678	467	1.634	443	1.226	348
6	H até I	1.067	290	1.178	321	1.386	385	1.383	375	1.019	289
7	J	729	198	839	228	1.042	289	1.031	278	664	188
8	K até L	939	255	1.016	276	1.152	320	1.143	310	882	249
9	M	1.281	349	1.457	397	1.666	464	1.691	458	1.249	354
10	N até P	962	262	1.067	290	1.216	339	1.238	336	961	272
11	Q até R	717	195	794	216	921	256	970	262	715	202
12	S	1.290	351	1.372	375	1.616	451	1.634	444	1.201	341
13	T até V	710	193	788	214	928	258	962	260	701	198
14	W até Z	1.159	316	1.320	360	1.531	426	1.547	419	1.099	312

A Figura 4.1 ilustra como o conteúdo de cada fragmento evolui ao longo do tempo. Embora não haja nenhum esquema associado a este documento XML, uma análise manual mostrou que o nome do funcionário (<*name*>*)* é o elemento identificador. Ele não muda entre as versões e é único - por isso foi usado para gerar os resultados esperados (*i.e.*, gabarito) no que diz respeito ao casamento de elementos correspondentes. Cada gráfico na Figura 4.1 representa a comparação entre duas versões sequenciais (*v1 x v2*, *v2 x v3*, *v3 x v4* e *v4 x v5*) de cada fragmento (eixo *x*). A coluna *#atualizações* mostra o número de empregados presentes em ambas as versões, com modificações em algum de seus elementos. A coluna *#remoções* mostra o número de empregados presentes apenas na versão anterior (empregados demitidos). Por outro lado, a coluna *#inserções* mostra o número de funcionários que estão presentes apenas na versão mais recente (funcionários contratados). Embora haja algumas variações, todos os fragmentos possuem uma distribuição similar com *#atualizações* > *#inserções* > *#remo-*

ções nos dois primeiros *diffs* (*v1 x v2*, *v2 x v3*), *#atualizações* > (*#inserções* = *#remoções*) no terceiro *diff* (*v3 x v4*) e #atualizações > *#remoções* > *#inserções* no quarto *diff* (*v4 x v5*).

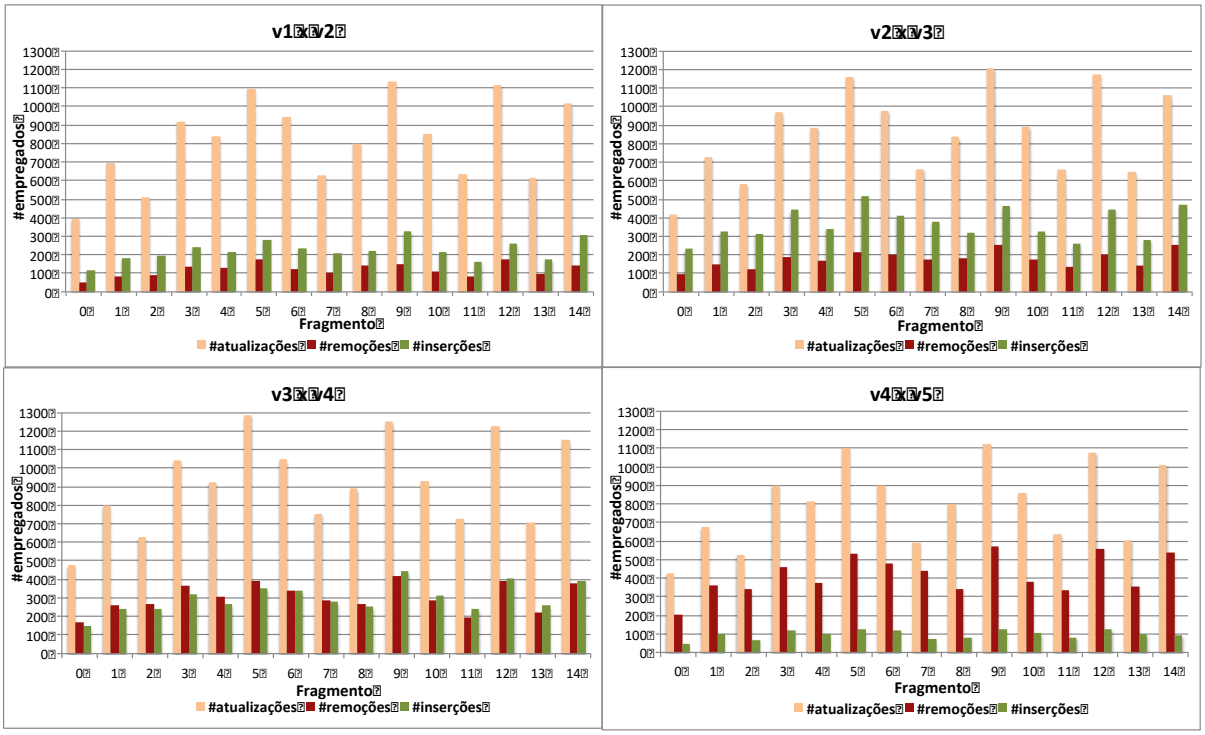

Figura 4.1: Características do documento XML com identificação do total de mudanças na comparação de duas versões sequenciais (verde representando inserção, amarelo representando atualização e vermelho representando remoção)

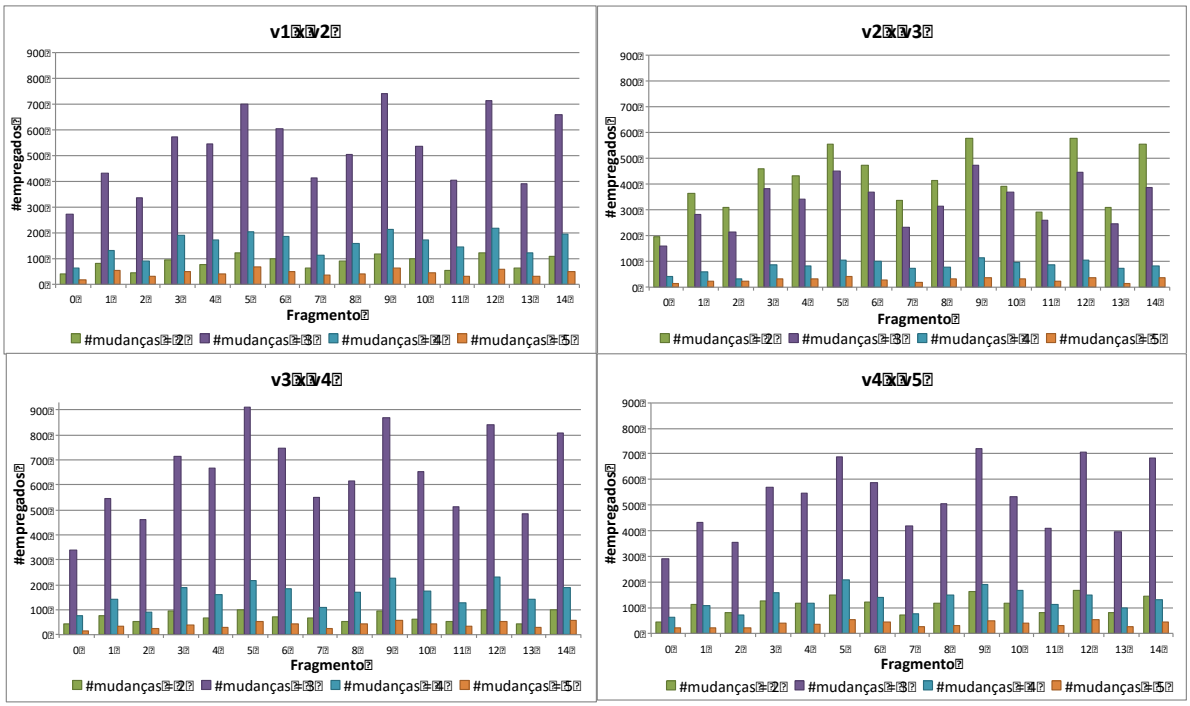

Figura 4.2: Evolução do documento XML com identificação do número de mudanças nos subelementos ao se analisar elementos correspondentes

A Figura 4.2 mostra a distribuição do número de mudanças entre os elementos correspondentes. Todos os funcionários sofreram alterações em pelo menos um dos seus subelementos, ao comparar duas versões sequenciais (*v1 x v2*, *v2 x v3*, *v3 x v4* e *v4 x v5*). O eixo *y* indica quantos funcionários *(#empregados*) têm alterações para 2, 3, 4, ou 5 subelementos. Poucos funcionários têm alterações em 1 ou 6 dos seus subelementos e por isso não são mostrados. Como pode ser observado, a maioria dos funcionários tem mudanças em três de seus subelementos entre duas versões sequenciais, com uma exceção no segundo *diff* (*v2 x v3*), onde a maioria dos funcionários tem alterações em dois dos seus subelementos.

4.3 ANÁLISE DE SENSIBILIDADE

Para viabilizar a parametrização do Phoenix, módulo usado pelo XChange para o casamento por similaridade, foi necessária a execução de uma análise de sensibilidade. O objetivo da análise de sensibilidade é identificar o limiar de similaridade que maximiza a *F-Measure* e definir o limiar base para este domínio. Vale mencionar que esta calibragem deve ser feita uma única vez para cada domínio. Nesta análise, foram utilizados os fragmentos 0 de todas as versões como conjunto de treinamento. Comparou-se cada versão consecutiva utilizando o Phoenix, variando o limiar de similaridade entre 0 e 1, com incrementos de 0,01. Foi calculada a eficácia do Phoenix em cada execução, o que permitiu determinar os melhores valores do limiar para cada comparação. Também foi possível verificar se o limiar de similaridade que obteve melhor eficácia na primeira execução (*v1 x v2*) é o mesmo para os demais pares de versões (*v2 x v3*, *v3 x v4* e *v4 x v5*).

A Figura 4.3 mostra a curva *F-Measure* para as quatro comparações. O maior valor em cada comparação é destacado. Pode-se observar que duas comparações obtiveram a maior *F-Measure* definido pelo mesmo limiar enquanto as outras duas comparações alcançaram valores mais altos com limiares diferentes. Portanto, pode-se concluir que não existe um limiar de similaridade único que maximiza a *F-Measure*, embora todos estejam entre 0,50 e 0,64.

A fim de adotar um único limiar base para o experimento, uma vez que o limiar que maximiza a *F-Measure* nas quatro comparações não é o mesmo, os valores da *F-Measure* obtidos nas 4 execuções foram somados, resultando em uma nova curva. A Figura 4.4 mostra parte dessa curva com as 20 maiores sumarizações da *F-Measure*. Como resultado, o limiar com a maior soma foi 0,55. Portanto, este valor de limiar de similaridade foi adotado nas demais execuções deste estudo experimental. Vale ressaltar que, como visto na Figura 4.3, os

valores da *F-Measure* obtidos com o limiar de 0,55 estão muito perto do maior valor da *F-Measure* alcançado em cada comparação.

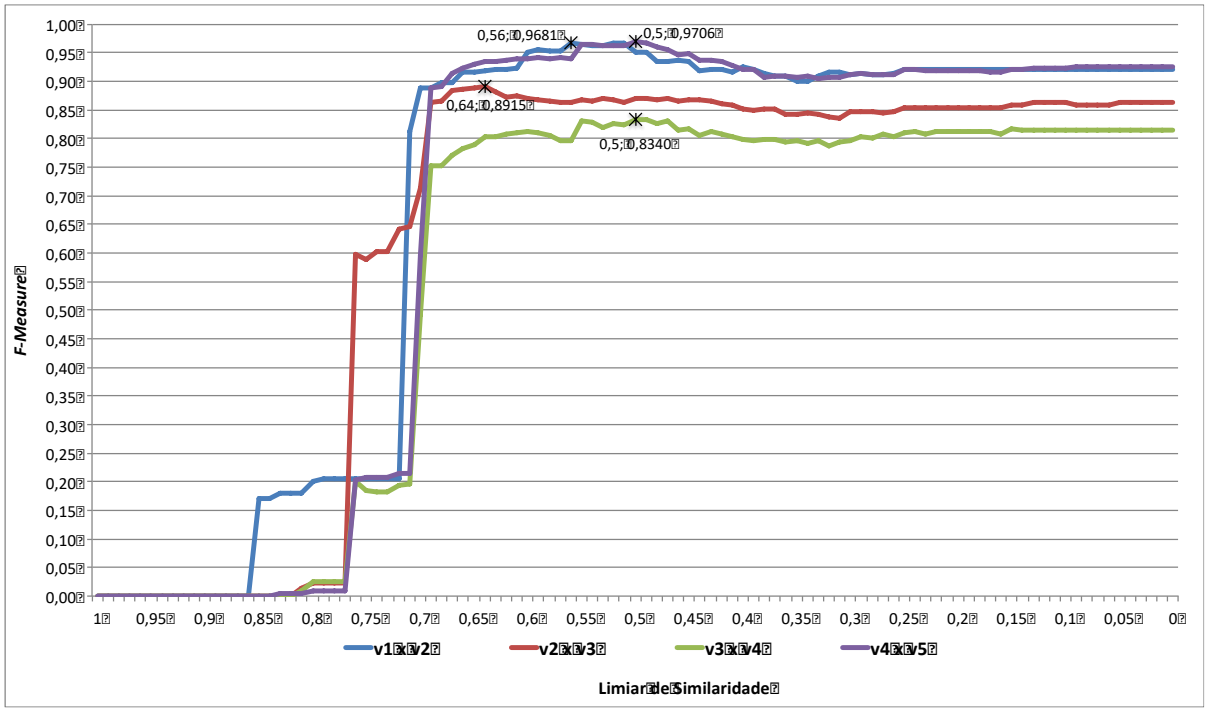

Figura 4.3: *F-Measure* de acordo com a variação do limiar de similaridade

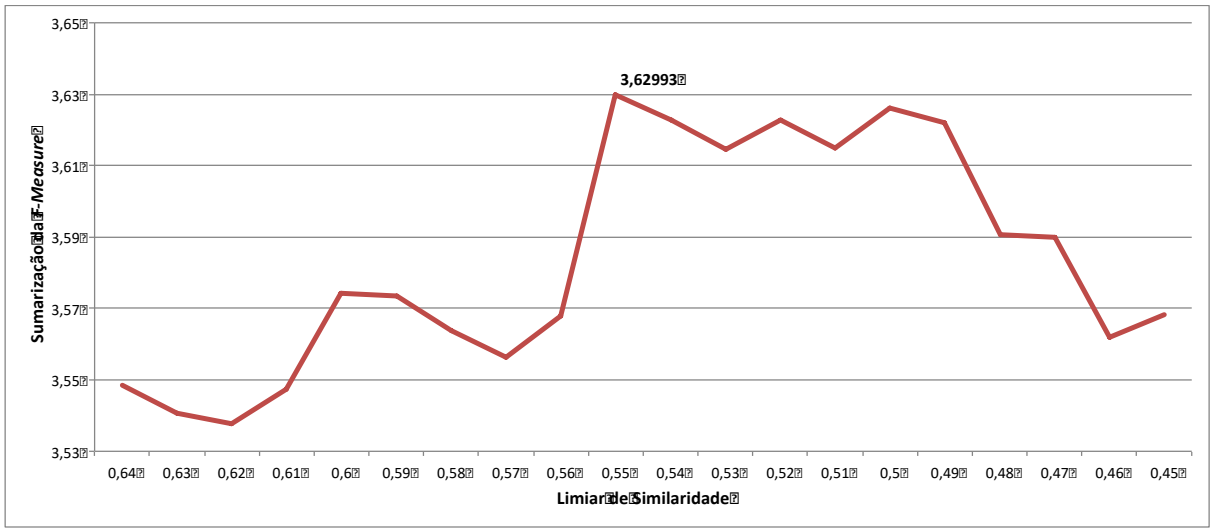

Figura 4.4: *F-Measure* acumulado de acordo com a variação do limiar de similaridade

4.4 PROCESSO DO ESTUDO EXPERIMENTAL

No estudo experimental, foram utilizados os fragmentos 1 a 14 das 5 versões do documento XML apresentado na Seção 4.2. O fragmento 0 foi excluído, uma vez que foi utili-

zado para definir o limiar de similaridade base a ser utilizado pelo Phoenix neste domínio. Este estudo experimental centra-se na eficácia e eficiência alcançadas pelo Phoenix, módulo de casamento por similaridade do XChange, em comparação com o X-Diff e o XyDiff. Como representado na Figura 4.5, para cada par de versões sequenciais (*v1 x v2*, *v2 x v3*, *v3 x v4*, *v4 x v5*) processadas pelas 3 abordagens, tem-se o número de verdadeiros positivos (casamentos corretos), falsos positivos (casamentos incorretos) e falsos negativos (casamentos não identificados) com base no gabarito (resultados esperados). Conforme mencionado anteriormente, como <*name*> é elemento identificador deste documento XML, ele foi utilizado para gerar o gabarito. Depois disso, foram calculadas as métricas precisão, cobertura e *F-Measure*. Também foi calculado o tempo de execução para apoiar a análise de eficiência. A eficiência é apresentada em termos de verdadeiros positivos por tempo de execução (ou seja, casamentos corretos por segundo – CCPS). É importante destacar que, uma vez que não há esquema associado ao documento, nenhuma das abordagens estava ciente de que <*name*> é o identificador, de modo que nenhuma delas foi capaz de usar esta informação para efetuar o casamento dos elementos correspondentes.

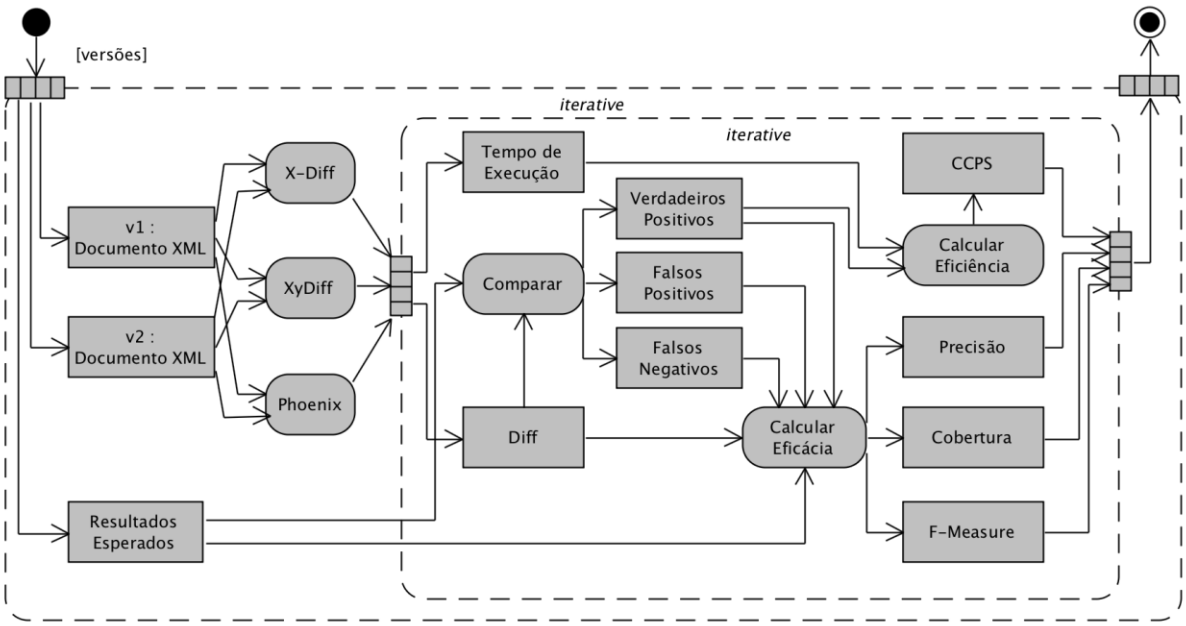

Figura 4.5: Processo do estudo experimental

Todos os experimentos foram realizados em um computador com processor Intel Core i7-4770, 3.40 GHz, com 24GB de memória RAM e sistema operacional Windows 10.

4.5 AVALIAÇÃO DA EFICÁCIA E DA EFICIÊNCIA

Com o valor do limiar de similaridade base do Phoenix definido (0,55), pode-se comparar a eficácia e a eficiência entre as abordagens Phoenix, X-Diff e XyDiff. Os testes desta seção foram realizados no ambiente de software livre R[4], utilizado para análises estatísticas e construção de gráficos, através da IDE RStudio[5]. Foram executadas comparações envolvendo todas as versões sequenciais de cada fragmento (1 a 14), totalizando 56 cenários (14 fragmentos x 4 pares de versões sequenciais). Para o X-Diff e o XyDiff, foram utilizados os parâmetros recomendados pelos respectivos autores.

O XyDiff apresenta baixa precisão ($\mu = 0{,}0015$, $\sigma = 0{,}0016$) e baixa cobertura ($\mu = 0{,}0019$, $\sigma = 0{,}0020$), devido à sua estratégia posicional para combinar elementos. Uma vez que ele assume que os elementos correspondentes estão na mesma posição em ambas as versões do documento XML, os elementos excluídos e inseridos levam o XyDiff a fazer casamentos errados. No entanto, na presença de um esquema e de um identificador, o XyDiff pode alcançar alta precisão e cobertura. Como o contexto utilizado não considera a existência de um esquema associado ao documento XML nem de um identificador, optou-se por retirar o XyDiff do estudo experimental.

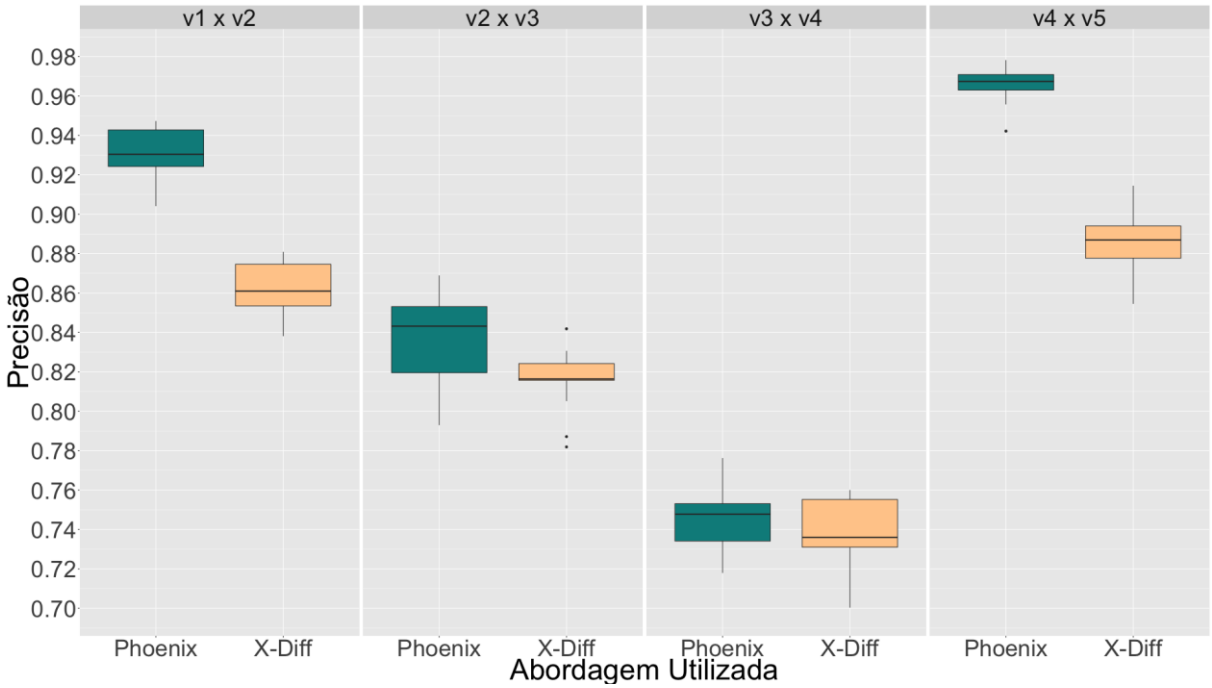

Figura 4.6: Precisão: Resultados obtidos

[4] https://www.r-project.org/
[5] https://www.rstudio.com/

A Figura 4.6 apresenta os resultados para a métrica precisão. Pode-se notar que o Phoenix tem a melhor mediana em todos os cenários. Em todos os cenários a precisão assumiu valores acima de 70% para as duas abordagens. Além disso, a precisão no terceiro *diff* (*v3 x v4*) apresenta os menores valores e ao mesmo tempo os mais próximos, quando comparado aos demais *diff*s. Como mencionado anteriormente, a Figura 4.1 ilustrou como o conteúdo de cada fragmento evolui e, para este terceiro *diff*, o numero de inserções é similar ao de remoções (*#inserções = #remoções)*, o que pode ter contribuído para este resultado.

A Figura 4.7 apresenta os resultados para a métrica cobertura obtidos pelo Phoenix e pelo X-Diff. O X-Diff tem a melhor mediana em todos os cenários. Também observa-se valores de cobertura elevados para ambas as abordagens em todos os cenários (acima de 85%).

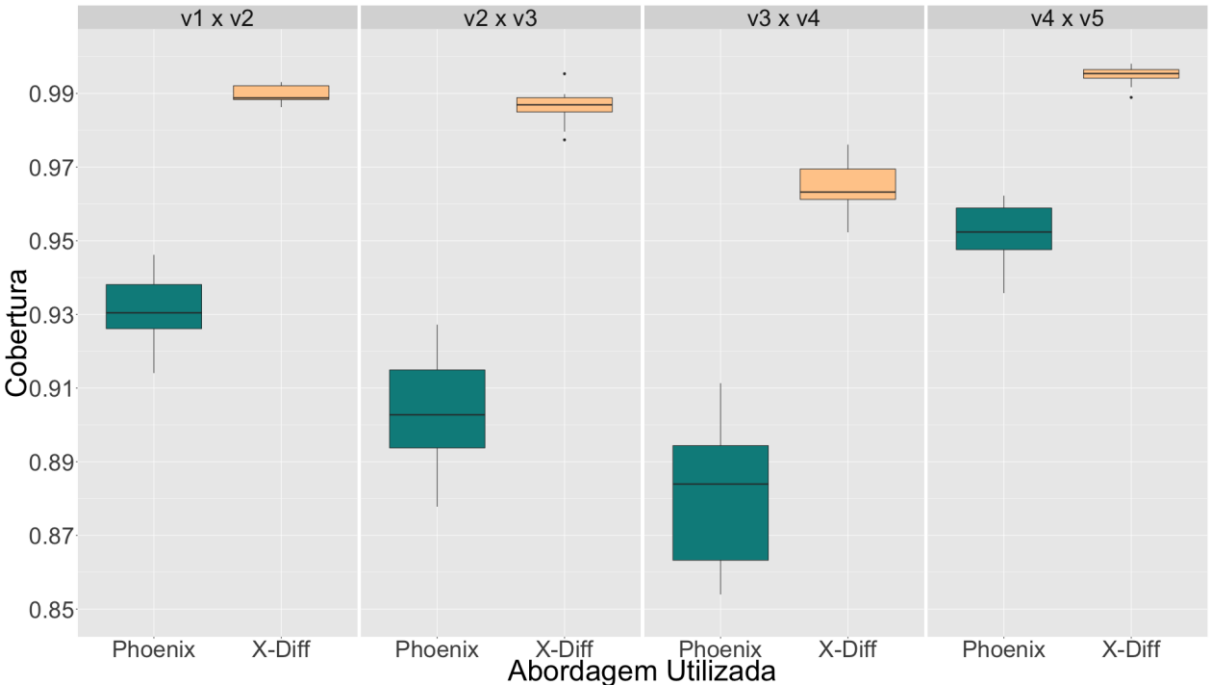

Figura 4.7: Cobertura: Resultados obtidos

Como é possível observar na Figura 4.6 e na Figura 4.7, as duas abordagens alcançaram valores elevados de precisão e cobertura, mesmo sem a existência de um identificador. O Phoenix tem maior precisão do que o X-Diff. Por outro lado, o X-Diff tem maior cobertura do que o Phoenix. Visando identificar qual abordagem tem a melhor eficácia, a Figura 4.8 apresenta os resultados obtidos para a *F-Measure*. O Phoenix tem a maior mediana em dois cenários e o X-Diff nos outros dois. Ao analisar os 14 fragmentos dos 4 pares de versões sequenciais, observou-se que o Phoenix obteve melhores resultados em 26 comparações e o X-Diff

em 30. Este resultado mostra valores elevados da *F-Measure* tanto para o Phoenix (μ = 0,8897, σ = 0,0641) quanto para o X-Diff (μ = 0,8964, σ = 0,0407), mas com uma ligeira vantagem para o X-Diff.

Foi realizada a análise do pressuposto de normalidade dos resultados para decisão entre o uso de testes paramétricos ou não-paramétricos na verificação de diferença significativa deste resultado. Para tanto foi utilizado o teste de Shapiro-Wilk (CONOVER, 1999). Foi utilizado um intervalo de confiança de 95%, ou seja, *α-value* = 0,05 (ROYSTON, 1995). A suposição de normalidade foi violada para o resultado da variável eficácia tanto para o X-Diff (*p-value* = 0,000408) como para o Phoenix (*p-value* = 0,0003016). O valor encontrado para o *p-value* é maior que zero mas é muito pequeno sendo possível verificar que *p-value* < *α-value.* Desta forma, esta amostra não possui distribuição normal.

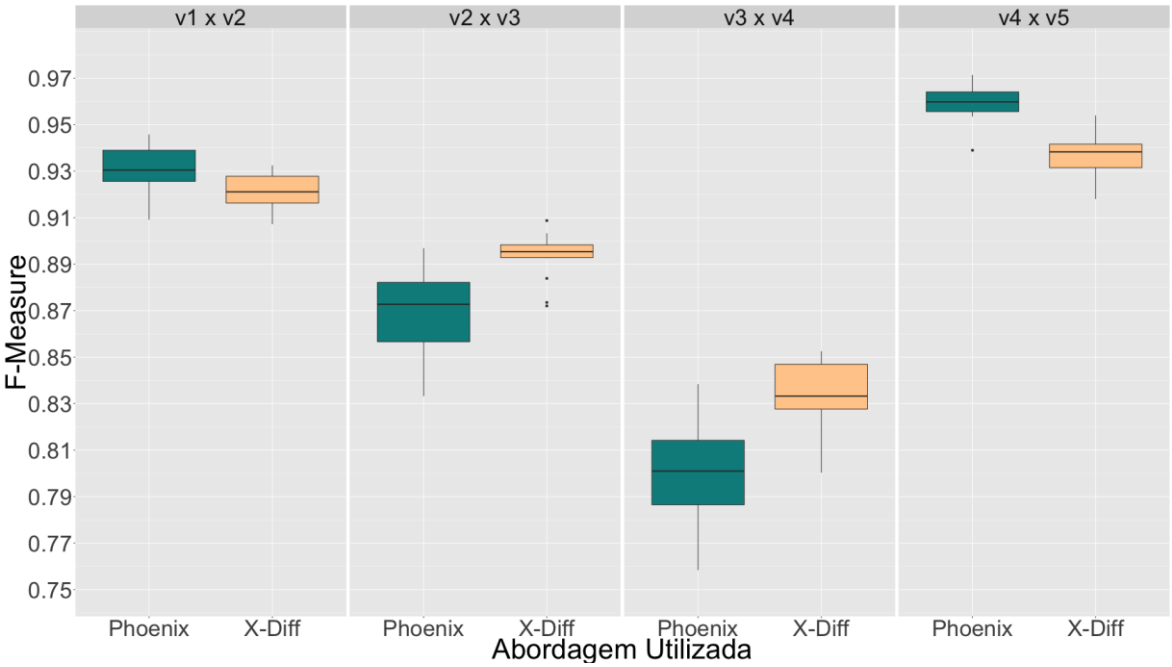

Figura 4.8: *F-Measure*: Resultados obtidos

Diante disso, o teste não-paramétrico Mann-Whitney para duas amostras independentes (WILCOXON, 1945) foi utilizado para análise estatística dos dados. O teste de Mann-Whitney foi executado e mostrou que neste cenário a diferença não é estatisticamente significativa (*p-value* = 0,3911).

Como o Phoenix e o X-Diff apresentaram resultados onde a diferença não é estatisticamente significativa em termos de eficácia, uma análise complementar contrastou a eficiência de ambas as abordagens. A Figura 4.9 mostra os casamentos corretos por segundo (CCPS) de cada abordagem. O Phoenix efetuou mais casamentos corretos por segundo em todos os

cenários. Neste caso, a diferença entre o Phoenix e o X-Diff é estatisticamente significativa (*p-value* = 2,2 x 10^{-16}). Este resultado é consequência da grande diferença do tempo de execução do Phoenix (μ = 78,89, σ = 38,10) e do X-Diff (μ = 3508,42, σ = 2281,54). O Phoenix é quase 45 vezes mais rápido do que o X-Diff.

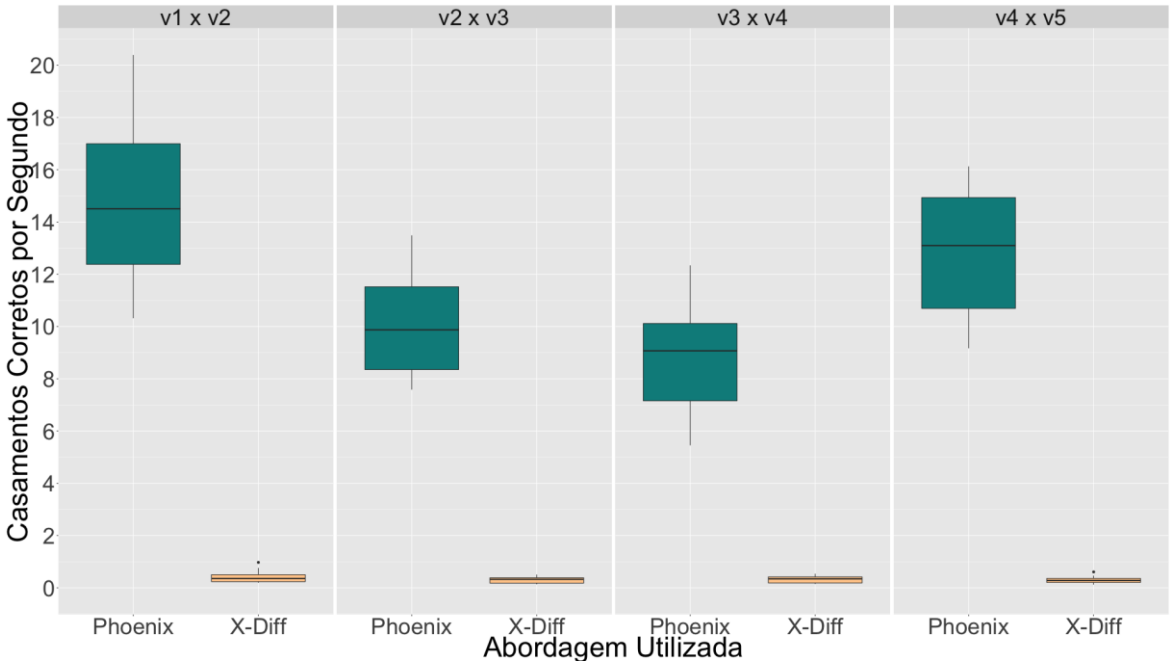

Figura 4.9: Casamentos corretos por segundo

Em síntese, o Phoenix e o X-Diff apresentaram valores elevados para a *F-Measure*. O X-Diff obteve melhores resultados na maioria dos fragmentos. No entanto, o Phoenix alcançou resultados sem diferença estatisticamente significativa em termos de eficácia em apenas uma fração do tempo de execução do X-Diff. Com isso em mente, pode-se responder às questões de pesquisa levantadas na Seção 4.1.

E1-QP1. O método baseado em cálculo de similaridade do XChange é mais eficaz na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

Resposta: Pode-se observar que, para o documento XML analisado, tanto o Phoenix quanto o X-Diff apresentaram alta eficácia na identificação de elementos correspondentes. O X-Diff mostrou uma ligeira vantagem - 30 vitórias contra 26 do Phoenix num total de 56 comparações. No entanto, esta diferença não é estatisticamente significativa (*p-value* = 0,3911).

E1-QP2. O método baseado em cálculo de similaridade do XChange é mais eficiente na identificação de elementos correspondentes entre versões de um documento XML, quando comparado às abordagens da literatura X-Diff e XyDiff?

Resposta: Pode-se observar que, para o documento XML analisado, o Phoenix apresentou maior eficiência no casamento de elementos correspondentes, sendo quase 45 vezes mais rápido do que o X-Diff. Além disso apresentou melhores resultados em todos os 56 fragmentos. Neste caso, a diferença é estatisticamente significativa (*p-value* = 2,2 x 10^{-16}). Como mencionado anteriormente, de acordo com os autores, o X-Diff apresenta bom desempenho quando lida com documentos menores e seu foco é encontrar o *delta* mínimo.

4.6 AMEAÇAS À VALIDADE

Apesar do cuidado tomado para reduzir as ameaças à validade para este estudo, alguns fatores não controlados podem ter influenciado os resultados observados.

O estudo experimental envolveu apenas um documento XML. O documento contendo os dados dos funcionários da Prefeitura de Baltimore tem três níveis, não contém atributos, e seus fragmentos variam de 121 a 467 KBytes. De acordo com (MIGNET; BARBOSA; VELTRI, 2003), os documentos XML têm uma média de 4 níveis, e, portanto, acredita-se que o documento utilizado seja representativo.

Foi utilizado um documento XML contendo dados reais. Essa escolha leva a situações que ocorrem em um ambiente real, incluindo inconsistências nos dados. Por exemplo, o elemento <*name*> foi utilizado como identificador na geração dos resultados esperados. No entanto, problemas como nomes duplicados ou alterações no nome do empregado poderiam produzir falsos positivos e falsos negativos. Para atenuar esta ameaça, uma inspeção manual foi feita para garantir a utilização do elemento <*name*> como identificador.

Foi utilizado um fragmento do documento XML como conjunto de treinamento para calibrar a abordagem Phoenix. Isto foi feito para descobrir o limiar de similaridade que maximiza a *F-Measure* no conjunto de treinamento *(i. e.*, o fragmento 0), enquanto que para as outras abordagens foram utilizados os valores padrão propostos por seus autores. Para atenuar esta ameaça, o fragmento 0 foi descartado, e a análise da eficácia e da eficiência foi realizada com os demais fragmentos (1 a 14).

4.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou um estudo comparativo entre o módulo de casamento por similaridade do XChange e duas abordagens de *diff* sintático de XML do estado da arte: X-Diff e XyDiff. O XyDiff apresenta um baixo índice de casamentos corretos de elementos correspondentes por se basear em casamento posicional quando não há identificador disponível. Por outro lado o Phoenix obteve resultados sem diferença estatística significativa quanto comparado ao X-Diff no casamento de elementos correspondentes, com apenas uma fração do tempo necessário pelo X-Diff. A partir deste resultado, conclui-se que o Phoenix é uma alternativa apropriada para apoiar o casamento por similaridade do XChange. Resta então comparar o XChange com o X-Diff no que diz respeito à compreensão da evolução dos documentos XML, diretamente influenciada pelo casamento correto dos elementos correspondentes. Essa comparação é feita no próximo capítulo.

CAPÍTULO 5 ESTUDO EXPERIMENTAL II: COMPREENSÃO DA EVOLUÇÃO DE DOCUMENTOS XML

5.1 INTRODUÇÃO

Neste estudo experimental, pretende-se caracterizar o apoio existente à compreensão da evolução de documentos XML através da análise do *diff* gerado por duas abordagens diferentes: o XChange, e o X-Diff (WANG; DEWITT; CAI, 2003). O X-Diff foi escolhido pois obteve alta eficácia na avaliação do Capítulo 4, no que diz respeito ao casamento de elementos correspondentes. Vale mencionar também que esta escolha se deu em função de não terem sido encontradas abordagens com foco no *diff* semântico como o XChange. Além disso, as abordagens de *diff* sintático também tem como objetivo a compreensão das mudanças. O propósito aqui é comparar os resultados obtidos com respeito à eficácia e à eficiência na análise e na compreensão da evolução de documentos XML a partir de um estudo experimental com usuários com experiência em manipulação de documentos XML. A eficácia é calculada em termos do número de acertos obtidos em cada tarefa conforme discutido mais a frente. Já a eficiência é calculada em termos do total de acertos por segundo (verdadeiros positivos por tempo de execução). Neste estudo experimental, o objetivo consiste, portanto, em responder às seguintes questões de pesquisa:

E2-QP1. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficaz do que a identificação de alterações sintáticas, utilizada pelo X-Diff?

E2-QP2. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficiente do que a identificação de alterações sintáticas, utilizada pelo X-Diff?

Este capítulo está organizado como segue. A Seção 5.2 apresenta uma visão geral do planejamento do estudo experimental, enquanto a Seção 5.3 apresenta o modo como ele foi executado. A Seção 5.4 apresenta uma caracterização dos participantes envolvidos no estudo. A análise estatística é apresentada na Seção 5.5, detalhando os testes realizados, seus respectivos resultados e conclusões sobre os dados obtidos. A Seção 5.6 descreve as ameaças à validade relacionadas a este estudo. Finalmente a Seção 5.7 apresenta as considerações finais deste capítulo.

5.2 DEFINIÇÃO E PLANEJAMENTO

Este estudo experimental tem como objetivo analisar o *diff* de versões de um documento XML com o propósito de avaliar a eficácia e a eficiência na análise e na compreensão, do ponto de vista de usuários com experiência em manipulação de documentos XML, no contexto da evolução de documentos XML.

De forma a facilitar a imersão do participante no contexto do estudo, foi criada uma situação fictícia, na qual o participante (usuário com experiência em manipulação de documentos XML) havia sido contratado pela prefeitura da cidade de Baltimore, sendo alocado inicialmente para realizar algumas tarefas no contexto do cadastro de funcionários da respectiva prefeitura a partir das versões *v1* (Apêndice A) e *v2* (Apêndice B) e dos *deltas* resultantes (Apêndice C e Apêndice D).

Vale mencionar que para a geração do *delta* resultante do XChange foi utilizado o apoio para a definição de regras de enriquecimento semântico apresentado na Seção 3.4. O fragmento 0 de cada uma das versões do documento XML da Prefeitura de Baltimore, descrito na Seção 4.2 foi utilizado na mineração. Para evitar qualquer tipo de influência, todos os 17 *itemsets* frequentes apresentados foram utilizados da forma que foram gerados, ou seja, não foram previamente analisados especialista.

Tabela 5.1: *Itemsets* frequentes apresentados pela mineração

	***Itemsets* Frequentes**	**Suporte**
1	*jobtitle=y*	209
2	*agencyid=y*	292
3	*agency=y*	1322
4	*annualsalary=u*	1269
5	*grosspay=u*	1383
6	*grosspay=d*	269
7	*jobtitle=y annualsalary=u*	189
8	*jobtitle=y grosspay=u*	195
9	*agencyid=y agency=y*	257
10	*agencyid=y annualsalary=u*	173
11	*agencyid=y grosspay=u*	190
12	*agency=y annualsalary=u*	931
13	*agency=y grosspay=u*	1009
14	*agency=y grosspay=d*	252
15	*annualsalary=u grosspay=u*	1125
16	*jobtitle=y annualsalary=u grosspay=u*	182
17	*agency=y annualsalary=u grosspay=u*	796

Um treinamento relacionado a *diff* de documentos XML, de aproximadamente 30 minutos, foi elaborado para apresentar o tema aos participantes. Além disso, foi preparada uma

contextualização do estudo experimental a partir da realização de uma tarefa exemplo, similar às que seriam executadas pelos participantes durante o estudo experimental.

Foram elaboradas três tarefas diferentes para serem usadas em cada uma das duas etapas do estudo experimental. As seis tarefas abordam diferentes tipos de operações usando conceitos da linguagem de consulta SQL (ELMASRI; NAVATHE, 2010), uma vez que deseja-se utilizar o *diff* para ajudar a responder perguntas (consultas). A Tabela 5.2 apresenta a classificação utilizada, e, para exemplificar, uma tarefa associada a cada tipo.

Tabela 5.2: Classificação das tarefas

	Enumeração	**Agregação**
Existencial	Quais funcionários foram demitidos?	Qual o impacto financeiro das contratações e demissões com base no salário anual?
Mudança	Quem foi promovido, ou seja, teve aumento de salário bruto e mudou de cargo?	Quantos funcionários foram transferidos, ou seja, mudaram de agência?
Conteúdo	Qual o funcionário teve o maior aumento de salário anual?	Qual o impacto financeiro dos aumentos do salário bruto?

A Tabela 5.2 serviu de guia para elaborar as questões do experimento. Seu objetivo é ajudar a cruzar os tipos possíveis de questões que se pode fazer sobre duas versões de documentos XML. As colunas da tabela tratam dos dois tipos possíveis de projeção de resultados de uma consulta (cláusula SELECT). Em uma consulta qualquer, pode-se enumerar os resultados, ou então agregá-los (*max*, *min*, *count*, *sum*). Já as linhas da Tabela 5.2 contemplam tipos de seleção que podem ser aplicados a uma consulta. A primeira delas é a existencial, que compara dois conjuntos selecionando os resultados que aparecem no primeiro conjunto, mas não aparecem no segundo (operador *minus*). A segunda linha contempla mudanças ocorridas em uma mesma instância do resultado. Para isso, é feita uma junção (dos elementos correspondentes), e seleção (cláusula WHERE) testando valores diferentes para os mesmos elementos (por exemplo, $e1.salario \neq e2.salario$, onde *e1* corresponde a um empregado na primeira versão, e *e2* é o elemento correspondente a *e1* na segunda versão). Finalmente, a terceira linha da tabela trata de restrições sobre o conjunto de resultados. Da mesma forma que para a segunda linha, é feita a junção dos elementos correspondentes, e então é aplicada uma restrição do tipo HAVING para filtrar apenas os resultados desejados (por exemplo, o funcionário que teve o maior aumento de salário anual).

Para a **E2-QP1**, a quantidade de acertos na realização das tarefas foi utilizada. Já para apoiar a análise da **E2-QP2**, foi registrada também a duração para realizar cada tarefa do es-

tudo experimental, em cada etapa. Estas são as variáveis dependentes deste estudo, como mostra a Tabela 5.3. A variável independente se refere ao *diff* de documentos XML e dois tratamentos foram utilizados: o *diff* sintático usado pelo X-Diff e o *diff* semântico usado pelo XChange. Os participantes do estudo experimental foram alunos e ex-alunos dos diversos cursos oferecidos pelo Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora (UFJF), com experiência em manipulação de documentos XML. Vale mencionar que houve um monitoramento sobre as atividades dos participantes durante o estudo.

Tabela 5.3: Planejamento do estudo experimental

Variáveis Dependentes	acerto, duração
Variável Independente	*diff* de documentos XML
Tratamentos	*diff* sintático usado pelo X-Diff *diff* semântico usado pelo XChange
Seleção de Contexto	*dataset real*
Dataset	Baltimore

Antes da execução do estudo, um projeto piloto com a mesma estrutura descrita neste planejamento foi realizado com dois participantes. A proposta era detectar possíveis problemas no material planejado para o estudo bem como na sua execução, permitindo que este material fosse aprimorado antes de sua utilização. Além disso um ambiente foi preparado com cuidado para isolar as tarefas e fornecer somente os dados e ferramentas necessários ao participante em cada etapa do experimento.

5.3 EXECUÇÃO DO ESTUDO

Inicialmente o convite para participação do estudo foi enviado para aproximadamente 200 alunos e ex-alunos da Universidade Federal de Juiz de Fora (UFJF), dos diversos cursos oferecidos pelo Departamento de Ciência da Computação. Destes convidados, 111 preencheram um questionário de caracterização disponibilizado *on-line* (Apêndice E). Muitos participantes, mesmo tendo se disponibilizado a participar, não compareceram ou tiveram problemas na data combinada e justificaram a ausência. Efetivamente, 60 convidados participaram do estudo experimental, divididos em 5 sessões, em função da disponibilidade de horário.

No início da sessão do estudo experimental, os participantes preencheram um formulário de consentimento em participar do estudo (Apêndice F). Num segundo momento, os participantes receberam o treinamento relacionado a *diff* de documentos XML. Em seguida participaram da contextualização do estudo experimental mencionada na seção anterior.

Em cada uma das 5 sessões realizadas, a execução do estudo experimental foi dividida em duas etapas e os participantes em dois grupos. Vale mencionar que as respostas do questionário de caracterização serviram de base para efetuar a divisão dos participantes em grupos mais homogêneos, em cada sessão realizada. A divisão foi feita de forma aleatória em cada nível de formação acadêmica, seguida do grau de experiência em XML, quando necessário. Foi definido um quadrado latino de dois tratamentos (XChange e X-Diff). As tarefas foram divididas em duas etapas: a etapa 1 contendo as tarefas 1, 2 e 3 mostradas no formulário do Apêndice G e a etapa 2 contendo as tarefas 4, 5 e 6 mostradas no formulário do Apêndice H. O primeiro grupo executou a etapa 1, e suas respectivas tarefas, utilizando o XChange como abordagem apoio; enquanto o segundo grupo utilizou o X-Diff para executar as mesmas tarefas. Em um segundo momento, os participantes receberam um novo conjunto de tarefas (etapa 2), que foi executado com a abordagem diferente da utilizada na primeira etapa. Os participantes não foram comunicados da existência e do objetivo desta divisão de tarefas em grupos, de forma a não influenciar a execução das tarefas.

Como mencionado na seção anterior, os participantes receberam a versão *v1* (Apêndice A) e a versão *v2* (Apêndice B) para serem usadas para consulta, caso necessário, bem como o *delta* resultante a partir da utilização do X-Diff (Apêndice C) e do XChange (Apêndice D) para utilizarem na solução das tarefas na etapa correspondente. Os participantes foram orientados a sempre que possível utilizar somente o *delta* na resolução das tarefas.

Para finalizar, os participantes responderam um questionário de encerramento (Apêndice I) para a obtenção de informações acerca do estudo, incluindo a percepção e considerações dos participantes sobre a aplicação das abordagens XChange e X-Diff.

5.4 CARACTERIZAÇÃO DOS PARTICIPANTES

Os participantes do estudo, selecionados por conveniência, são alunos e ex-alunos da Universidade Federal de Juiz de Fora no nível de graduação (em sua maioria), mestrado ou doutorado, como mostra a Figura 5.1. Todos os participantes já cursaram as disciplinas de Banco de Dados e de Engenharia de Software. No semestre em que o estudo foi aplicado, eles não eram alunos de nenhuma disciplina sob a responsabilidade dos pesquisadores responsáveis pelo estudo e não houve nenhum tipo de compensação para os participantes. Como um grande número de participantes contribuiu com o estudo ($\geq$ 30) foi possível realizar uma análise estatística mais aprimorada (JURISTO; MORENO, 2001).

A Figura 5.2 apresenta a experiência dos participantes em projeto de software. A maioria dos participantes é composta por graduandos ou graduados e, em função disso, tem pouca

experiência. Comprovando esta informação, pode-se observar que um pequeno grupo de participantes possui mais de 6 anos de experiência na indústria enquanto um grupo significativo possui experiência de nenhum a 2 anos em projetos pessoais e acadêmicos.

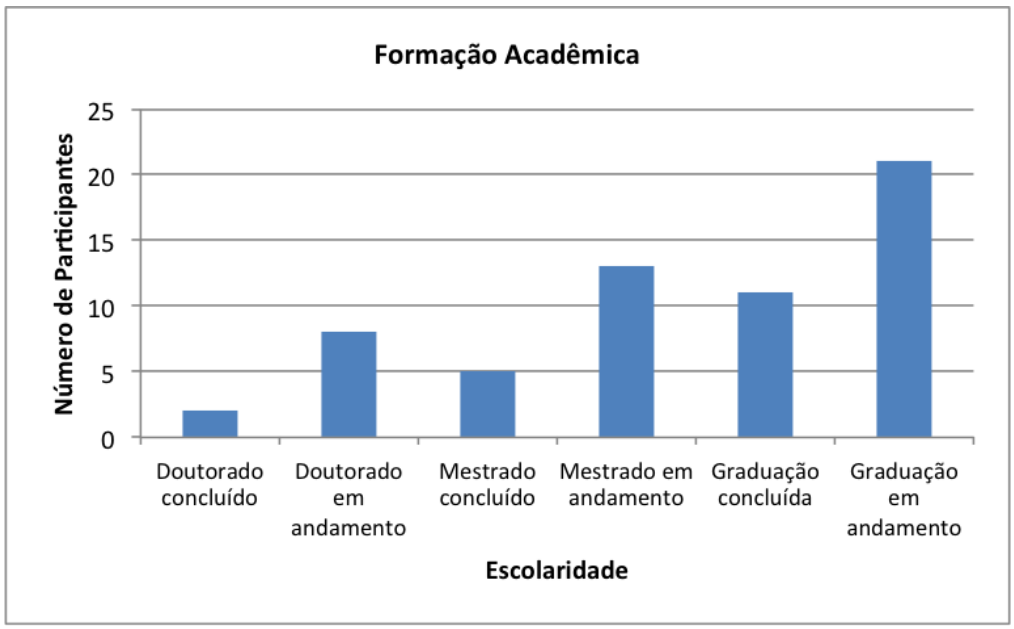

Figura 5.1: Formação acadêmica dos participantes envolvidos

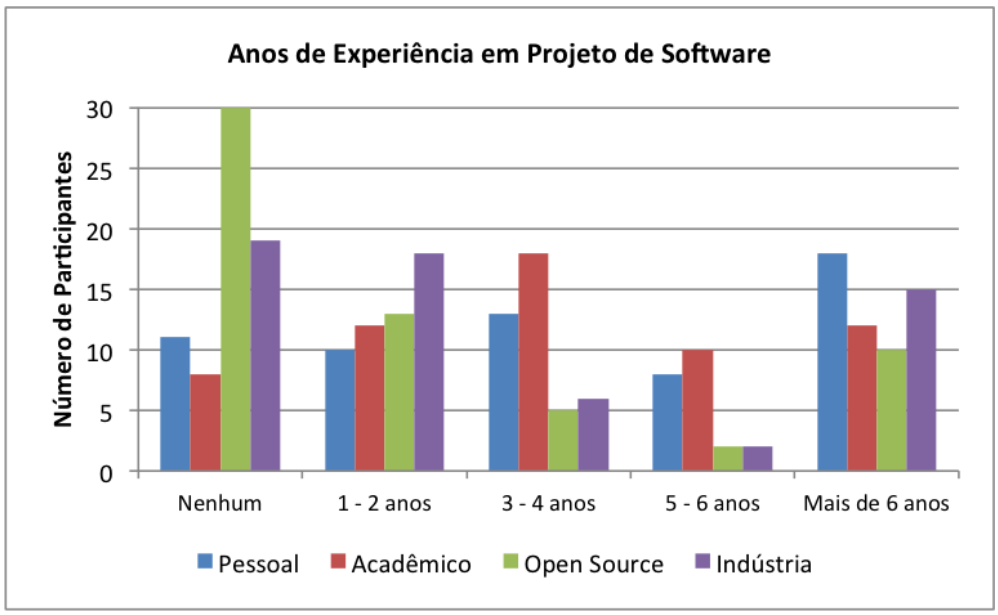

Figura 5.2: Anos de experiência em projeto de software dos participantes envolvidos

A Figura 5.3 mostra que a maioria dos participantes tem experiência na indústria relacionada a Banco de Dados e a Controle de Versão. Além disso, cerca de 25% tem experiência na indústria com XML e *diff* de arquivos. Como o número de participantes cursando a gradu-

ação ou com graduação completa representa a maioria dos participantes, tem-se que muitos deles têm experiência em XML e *diff* apenas em livros ou sala de aula nos temas em questão.

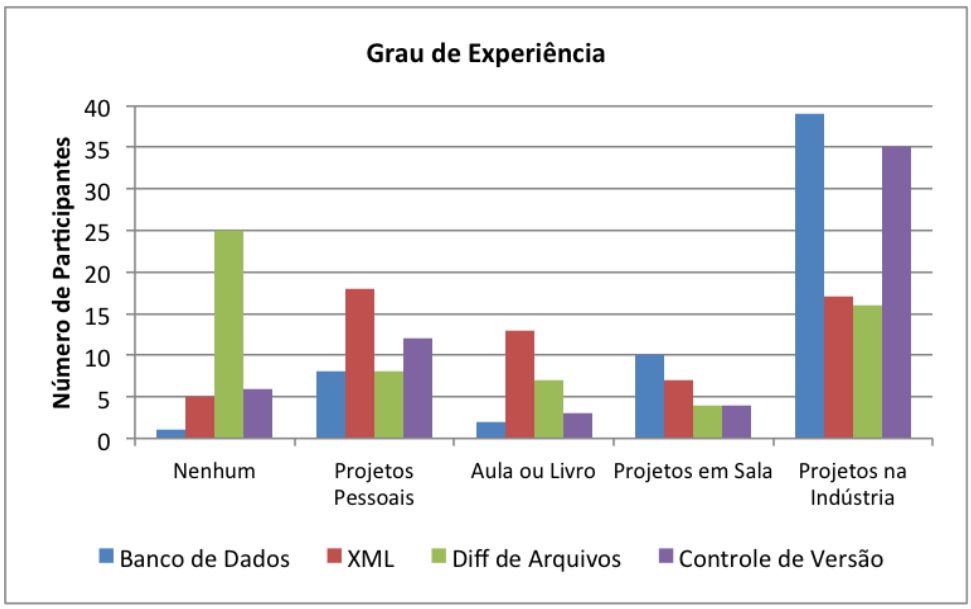

Figura 5.3: Grau de experiência por área de conhecimento dos participantes envolvidos

Para finalizar, a Tabela 5.4 mostra a alocação dos participantes nas sessões realizadas. Dos 60 participantes, 29 utilizaram o XChange como abordagem apoio na etapa 1 do estudo experimental e o X-Diff, na etapa 2. Por consequência, 31 participantes utilizaram o X-Diff como abordagem apoio na etapa 1 e o XChange, na etapa 2. Em cada sessão, procurou-se manter o mesmo número de participantes utilizando cada abordagem, como pode ser observado. No entanto, na sessão 4, dois participantes não registraram o tempo gasto em parte das tarefas realizadas e em função disso foram descartados. Além disso, em função de um equívoco na distribuição dos formulários, um participante recebeu apenas duas das três tarefas de uma etapa, sendo por isso também descartado.

Tabela 5.4: Alocação dos participantes no estudo experimental

Sessão	Etapa 1	Etapa 2	Identificadores dos Participantes
1	X-Diff	XChange	1 2 3 4 5 6 7 8
	XChange	X-Diff	9 10 11 12 13 14 15 16 17
2	X-Diff	XChange	18 19 20 21 22 23 24 25 26
	XChange	X-Diff	27 28 29 30 31 32 33 34 35 36
3	X-Diff	XChange	37 38 39 40
	XChange	X-Diff	41 42 43 44 45
4	X-Diff	XChange	46 47 48 49 50
	XChange	X-Diff	51
5	X-Diff	XChange	52 53 54 55 56
	XChange	X-Diff	57 58 59 60

5.5 AVALIAÇÃO DA EFICÁCIA E DA EFICIÊNCIA

A avaliação apresentada nesta seção foi realizada a partir do ambiente de software livre R, através da IDE RStudio, assim como os testes do Capítulo 4. Inicialmente, foi realizada a análise do pressuposto de normalidade dos resultados. Para tanto foi utilizado o teste de Shapiro-Wilk (CONOVER, 1999) e um intervalo de confiança de 95% (ROYSTON, 1995).

A suposição de normalidade foi violada para os resultados da variável acerto em todas as tarefas nas duas abordagens, como pode ser visto na segunda e terceira colunas da Tabela 5.5. O valor encontrado para o *p-value* nas tarefas é maior que zero mas é muito pequeno e, nos testes efetuados, é possível verificar que *p-value* < *α-value.* A suposição de normalidade foi violada para os resultados da variável duração em todas as tarefas nas duas abordagens, com exceção das tarefas 2, 5 e 6 utilizando o X-Diff, destacadas em cinza na quinta coluna da Tabela 5.5. Desta forma, esta amostra não possui distribuição normal.

Tabela 5.5: *P-value* para o teste de normalidade das variáveis Acerto e Duração

Tarefa	Acerto		Duração	
	XChange	X-Diff	XChange	X-Diff
1	0,000000007759000	0,0000000009976	0,0000002248	0,001374
2	0,000000286000000	0,0000001440000	0,0000002800	0,087650
3	0,000000014090000	0,0000000085660	0,0000142700	0,008617
4	0,000000000004636	0,0000000176400	0,0086170000	0,002378
5	0,000000000004636	0,0000000003103	0,0000024940	0,754000
6	0,000000000103600	0,0000000176400	0,0000482500	0,092790

Diante disso, o teste não-paramétrico Mann-Whitney para duas amostras independentes (WILCOXON, 1945) foi utilizado para análise estatística dos dados. Os resultados relacionados às variáveis acerto e duração podem ser visualizados na Tabela 5.6, considerando o seguinte padrão: * nos casos onde *p-value* < *0,05*; ** para *p-value* < *0,01* e *** nos casos onde *p-value* < *0,001*. Nestes casos, *p-value* assumiu valores inferiores ao de *α-value*, o que indica que existe diferença estatisticamente significativa entre os escores de execução destas tarefas usando o XChange e o X-Diff. Para as demais tarefas não existe diferença significativa. Além disso, foi utilizada a medida de tamanho de efeito *Delta de Cliff* - |*d*| (CLIFF, 1996). Essa é uma medida não-paramétrica que permite quantificar a magnitude da diferença entre dois grupos que não atendem aos pressupostos da normalidade e varia de -1 a +1. Um *Delta de Cliff* de -1 ou +1 indica uma ausência de sobreposição, enquanto que um *Delta de Cliff* de 0.0 indica que os grupos estão completamente sobrepostos. Além disso, a sua descrição possibilita complementar a interpretação do *p-value* associado ao correspondente teste utilizado. A

interpretação desses escores considera: $|d| < 0,147$ diferença negligenciável, $|d| < 0,33$ diferença pequena, $|d| < 0,474$ diferença média, caso contrário, considera-se que é grande (ROMANO *et al.*, 2006). Na Tabela 5.6, em cinza, foram destacados os valores onde a diferença a partir do *Delta de Cliff* é média ou grande.

Tabela 5.6: *Delta de Cliff* e *p-value* para as variáveis Acerto e Duração

Tarefa	Acerto		Duração	
	Delta de Cliff	***P-value***	***Delta de Cliff***	***P-value***
1	-0,045606		0,010011	
2	0,351502	**	-0,692992	***
3	0,036707		0,219132	
4	0,131113	*	0,089490	
5	0,002081		-0,764828	***
6	0,554631	***	-0,911551	***

O gráfico de barras empilhadas apresentado na Figura 5.4 tem como objetivo avaliar o número de acertos obtidos em cada tarefa a partir do XChange e do X-Diff. Para as tarefas 2 e 3, considerou-se também meio certo, conforme gabarito elaborado.

A partir deste gráfico pode-se também confirmar o resultado obtido no teste de Mann-Whitney e *Delta de Cliff* mostrados na Tabela 5.6. Ao verificar a Figura 5.4, percebe-se que a abordagem XChange obtém um número maior de acertos na maior parte das tarefas. Somente na tarefa 1, o total de acertos dos participantes que utilizaram o X-Diff foi maior que o dos que usaram o XChange.

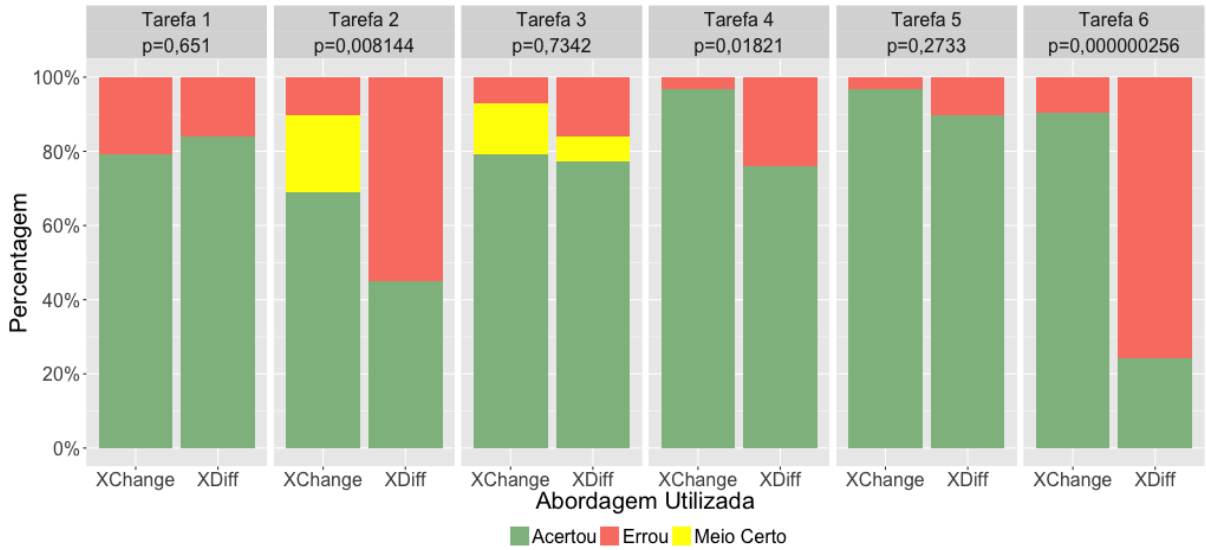

Figura 5.4: Análise da variável acerto

Ainda sobre a corretude das tarefas realizadas, como pode ser visto na Figura 5.5 os participantes obtiveram um número maior de acertos utilizando o XChange. Na etapa 1, mais de 80% dos participantes que utilizaram o XChange obtiveram número de acertos maior ou igual a 2 de um total de 3. Para aqueles que usaram o X-Diff, menos de 70% alcançaram o mesmo resultado. Uma diferença maior ainda foi observada na etapa 2, onde todos os participantes, que num segundo momento utilizaram o XChange, obtiveram número de acertos maior ou igual a 2. Para aqueles que utilizaram o X-Diff na etapa 2, o resultado foi similar ao dos participantes que utilizaram esta abordagem na etapa 1 (menos de 70% obtiveram número de acertos maior ou igual a 2).

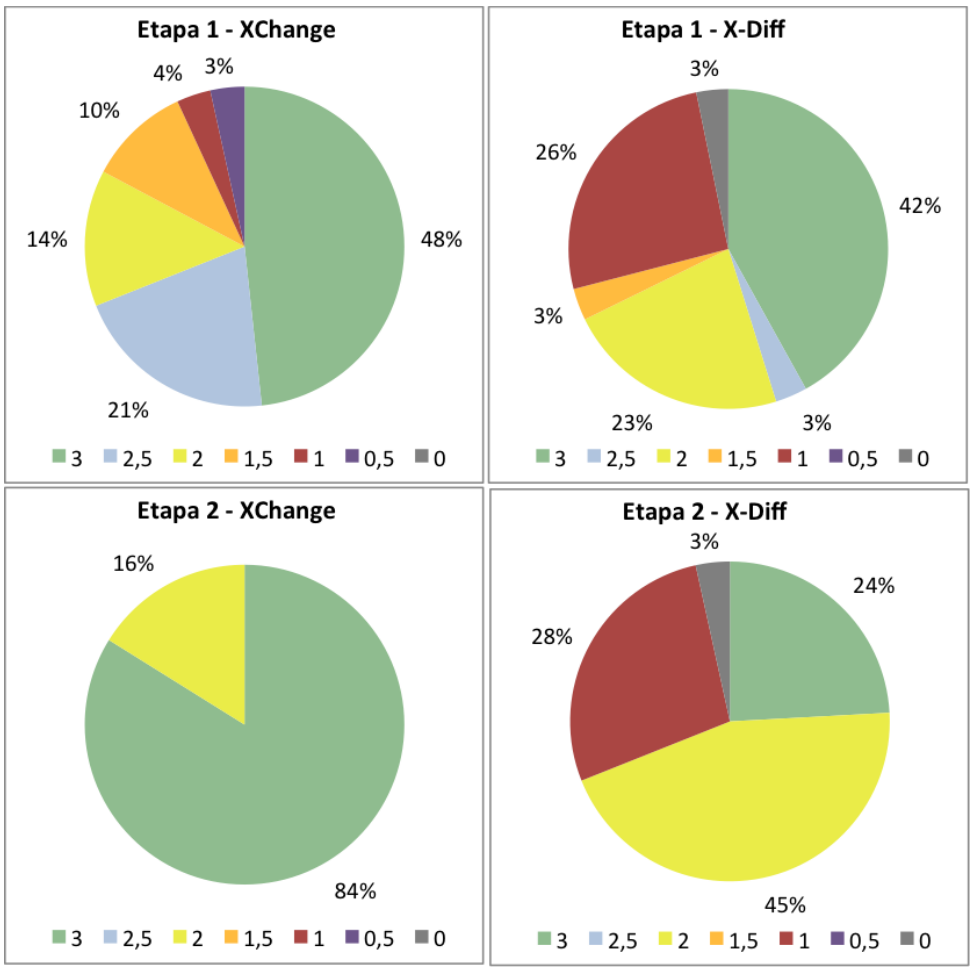

Figura 5.5: Distribuição de acertos pelo total de participantes em cada etapa

Finalizando a avaliação do total de acertos, na Figura 5.6 foram analisadas as respostas dos participantes quanto ao grau de dificuldade de execução das tarefas (1 - fácil; 5 - difícil). Alguns participantes não informaram o grau de dificuldade e isso foi registrado como NI (Não Informado). De maneira geral, as mesmas tarefas, quando executadas a partir do X-Diff, fo-

ram classificadas com grau de dificuldade maior para mais de 80% dos participantes. A tarefa 6 foi considerada a mais difícil pelos participantes que usaram o X-Diff, seguida das tarefas 2 e 5. No caso do XChange, a tarefa 5 foi classificada com grau de dificuldade maior.

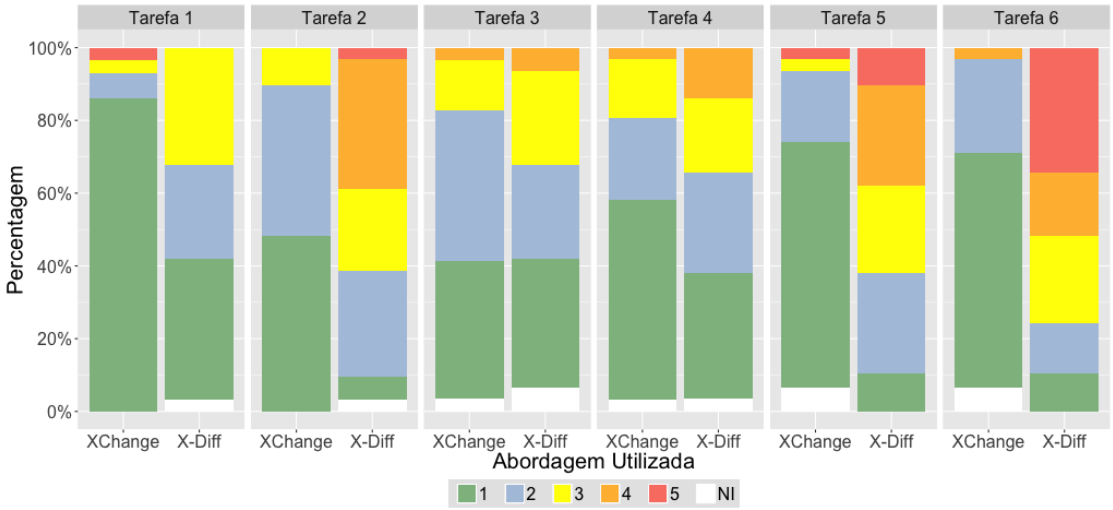

Figura 5.6: Grau de dificuldade de execução das tarefas

Desta forma pode-se concluir que a identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficaz, com base na quantidade de acertos, do que a identificação de alterações sintáticas, utilizada pelo X-Diff.

Já os *boxplots* apresentados na Figura 5.7 têm como objetivo resumir as distribuições do XChange e do X-Diff no que se refere à variável duração. Como pode ser observado, os *boxplots* também auxiliam na exibição da diferença de duração para as tarefas 2, 5 e 6. Nestas tarefas, os participantes que utilizaram o XChange efetuaram as tarefas em um tempo menor que os participantes que utilizaram o X-Diff.

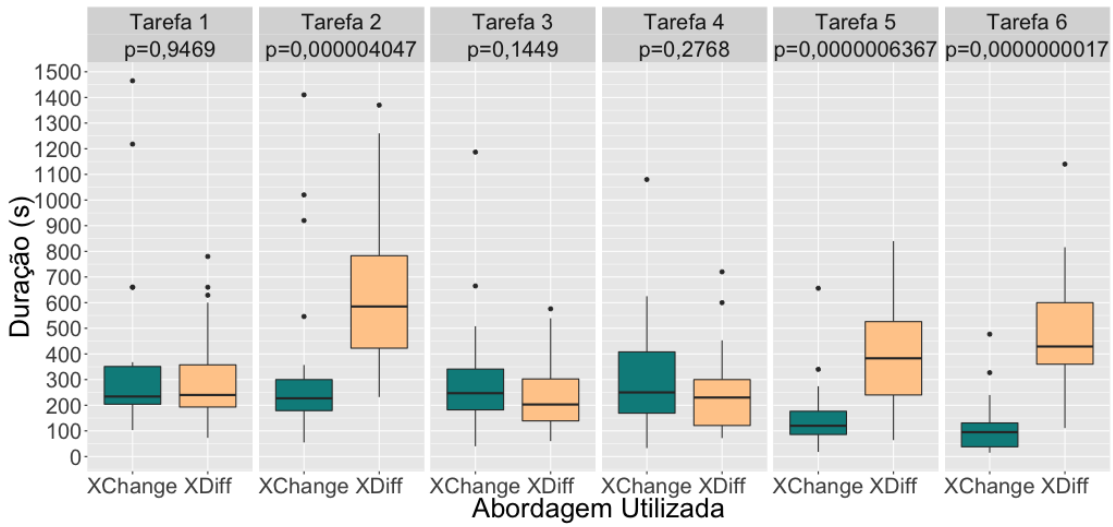

Figura 5.7: Análise da variável duração

É possível notar ainda na Figura 5.7 que a abordagem XChange apresenta um maior número de *outliers*. *Outliers* podem ser gerados por diversas situações, como, por exemplo, dados defeituosos e procedimentos incorretos (BARNETT; LEWIS, 1994). Como não há suspeita que os *outliers* identificados neste estudo experimental tenham sido decorrentes de medições incorretas ou outras falhas no processo experimental, não há razão para a sua remoção.

Efetuada a análise com base na variável duração e na variável acerto, é possível contrastar a eficiência de ambas abordagens, que leva em consideração o total de acertos por segundo (verdadeiros positivos por tempo de execução) mostrado na Figura 5.8. O X-Diff obteve mais acertos por segundo para as tarefas 1, 3 e 4 enquanto o XChange obteve o mais acertos por segundo para as tarefas 2, 5 e 6.

Os resultados para esta amostra, relacionados ao total de acertos por segundo, podem ser visualizados na Tabela 5.7 onde pode ser observado que *p-value* < *α-value* para as tarefas 2, 5 e 6. Portanto, existe diferença estatisticamente significativa entre os escores de acertos destas tarefas usando o XChange e o X-Diff. Para as demais tarefas não existe diferença significativa. A Tabela 5.7, mostra também, destacados em cinza, os valores para o *Delta de Cliff* onde a diferença é grande. De acordo com os participantes, a sumarização oferecida pelo *delta* do XChange contribuiu para o melhor resultado nas tarefas 2, 5 e 6, mesmo com o tamanho do *delta* significativamente maior quando comparado ao do X-Diff.

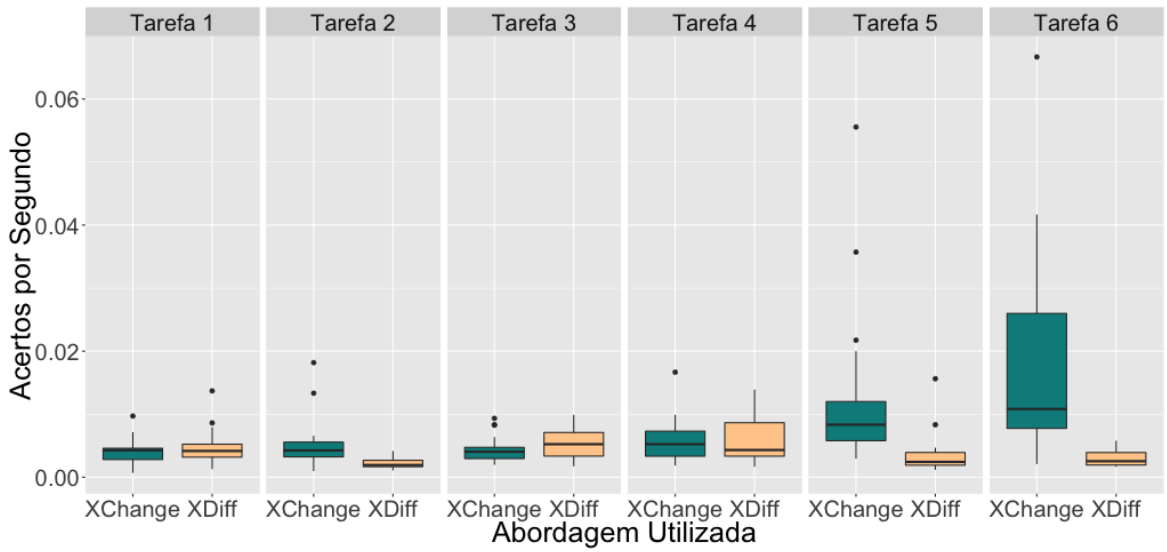

Figura 5.8: Total de acertos por segundo

Assim conclui-se que a identificação de alterações semânticas, usada pelo XChange, torna a compreensão da evolução de documentos XML mais eficiente, com base na quantidade de acertos por segundo, do que a identificação de alterações sintáticas, usada pelo X-Diff.

Tabela 5.7: *Delta de Cliff* e *p-value* para o total de acertos por segundo

Tarefa	*Delta de Cliff*	*P-value*
1	-0,9120879	
2	-0, 9122711	***
3	-0,9353400	
4	-0,9207570	
5	-0,8225885	***
6	-0,9489796	***

Em síntese, através da análise estatística realizada sobre os valores do estudo executado, pode-se expressar as seguintes conclusões.

E2-QP1. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficaz do que a identificação de alterações sintáticas, utilizada pelo X-Diff?
Resposta: Sim. Os participantes que executaram as tarefas utilizando o XChange obtiveram maior número de acertos em todas as tarefas, com exceção da tarefa 1. As tarefas 1, 3 e 5 alcançaram resultados sem diferença estatisticamente significativa. Nas demais tarefas foi comprovada a existência de diferença estatisticamente significativa entre os escores alcançados. Sendo assim, o XChange se mostrou mais eficaz na compreensão da evolução de documentos XML, quando comparado ao X-Diff. Complementando os resultados, os participantes consideraram que o *delta* resultante da abordagem XChange facilita a consulta, em função do formato e da sumarização. Mencionaram que, como no X-Diff era necessário percorrer o documento todo, estavam mais sujeitos a erros ou resoluções incompletas.

E2-QP2. A identificação de alterações semânticas, utilizada pelo XChange, torna a compreensão da evolução de documentos XML mais eficiente do que a identificação de alterações sintáticas, utilizada pelo X-Diff?
Resposta: Sim. A execução das tarefas pelos participantes usando o XChange gerou mais acertos por segundo, com diferença significativa para as tarefas 2, 5 e 6. Nas demais tarefas, o X-Diff apresentou valores levemente maiores. No entanto, esta diferença não é estatisticamente significativa. Desta forma o XChange se mostrou mais eficiente que o X-Diff na compreensão da evolução de documentos XML. Complementando a análise, os participantes mencionaram que o tamanho do *delta* resultante do X-Diff foi um ponto positivo. Por outro lado, em função do formato deste *delta*, os participantes tinham que percorrer o documento todo para descobrir as respostas desejadas. A sumarização e o formato do *delta* utilizados pelo XChange contribuíram para que as respostas corretas fossem encontradas em um tempo menor, sem a necessidade de percorrer todo o documento.

5.6 AMEAÇAS À VALIDADE

Durante o planejamento deste estudo, buscou-se evitar ameaças que pudessem impactar ou limitar a validade dos resultados obtidos (WOHLIN *et al.,* 2000). No entanto, não é possível garantir que tais ameaças não tenham afetado os resultados. Desta forma, as ameaças identificadas no contexto deste estudo são descritas a seguir.

O estudo não foi executado em um único dia por todos os participantes, em função da disponibilidade dos mesmos. Isto pode ter influenciado os resultados, já que não é possível confirmar que as circunstâncias eram as mesmas nas ocasiões em que cada participante participou do estudo experimental. Contudo, o mesmo roteiro e o mesmo ambiente computacional foi utilizado com a intenção de minimizar esta ameaça.

Conforme descrito na seção de planejamento, a execução do estudo consistiu de duas etapas. Embora o estudo tenha sido projetado para evitar o efeito de aprendizado dos participantes (fornecendo tarefas diferentes a cada etapa a partir do quadrado latino descrito na Seção 5.3), não é possível confirmar que este efeito tenha sido totalmente eliminado.

Para a geração do *diff* a partir do XChange, uma possibilidade é utilizar as regras definidas manualmente pelo especialista de domínio. Outra opção é efetuar uma seleção a partir das regras sugeridas pelo apoio automático baseado em mineração. Com o intuito de não haver interferência por parte do especialista neste processo de seleção das regras, foram utilizadas todas as regras de enriquecimento semântico sugeridas pelo apoio automático. Vale mencionar que o suporte mínimo utilizado na identificação dos *itemsets* frequentes que deram origem a estas regras foi 0,03. Consequentemente, os resultados do XChange podem ter sido afetados negativamente pela ausência do passo semiautomático onde o especialista refina e nomeia as regras geradas pela mineração.

O entendimento dos participantes sobre as questões dos formulários é diretamente influenciado pela forma como as questões foram elaboradas. A análise dos instrumentos utilizados (inclusive os formulários) a partir de um estudo piloto visou reduzir esta interferência.

Como não se trata de um estudo de observação, em função do número razoável de participantes, assumiu-se que os participantes seguiram as instruções e a ordem das atividades. Para minimizar esta ameaça, somente depois que o participante concluiu a primeira etapa, é que foi apresentada a segunda etapa do estudo. Adicionalmente, não é possível confirmar se a duração informada pelos participantes durante as tarefas está correta.

Os participantes não possuem a mesma habilidade para resolução de problemas. Para minimizar este efeito, as respostas do questionário de caracterização serviram de base para efetuar a divisão dos participantes em grupos mais homogêneos em cada sessão realizada.

Os participantes deste estudo são, na sua maioria, alunos de graduação, o que limita a representatividade das pessoas que poderiam se beneficiar da abordagem; no entanto, parte dos alunos cursou ou está cursando mestrado e doutorado ou possui experiência na indústria, diminuindo tal ameaça.

Foi utilizado um único *dataset* durante o estudo, o que limita a generalização dos resultados. Estudos experimentais com outros *datasets* devem ser executados.

O tamanho da amostra é limitado, o que não é ideal do ponto de vista estatístico. Desta forma, os resultados do estudo não são conclusivos: somente fornecem indícios. No entanto, o número de participantes superior a 30 que é considerado alto (JURISTO; MORENO, 2001), o que aumenta a validade estatística das conclusões obtidas.

Há uma hierarquia entre os participantes do estudo (ex-alunos) e o responsável pela execução (ex-professor). A fim de minimizar esta ameaça, nenhum participante estava cursando alguma disciplina sob a responsabilidade dos pesquisadores responsáveis por esse estudo no semestre em que foi realizado. No entanto, não foi possível confirmar se os participantes executaram o estudo da mesma forma como o teriam executado em outro cenário.

Por fim, o agrupamento das tarefas por tipo auxilia a análise dos dados. No entanto, embora algumas destas tarefas possam ter grau de dificuldade maior do que o de outras tarefas, o mesmo peso foi atribuído a todas as tarefas. Isto pode influenciar os resultados. Devido à subjetividade na avaliação do grau de dificuldade (o que introduziria viés na análise dos dados), optou-se por manter esta configuração.

5.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou como motivação principal a resposta às questões de pesquisa relacionadas à eficiência e eficácia do XChange no que diz respeito a compreensão da evolução de documentos XML em comparação com o X-Diff. Após a avaliação, o XChange se mostrou mais eficaz e mais eficiente em comparação como X-Diff. Os participantes deste estudo experimental indicaram que o arquivo resultante que apresenta o *diff* das versões do documento XML para o X-Diff é muito menor que o do XChange. Por outro lado, para que as tarefas fossem concluídas com sucesso utilizando o X-Diff, foi necessário percorrer todo o documento de *diff*, pois as respostas não estavam sumarizadas como no XChange. Segundo os participantes, em função disso, a consulta é mais demorada, menos intuitiva e mais sujeita a

erros a partir do X-Diff. Por outro lado, a sumarização oferecida pelo XChange torna o documento resultante muito grande e não mostra todas as informações do documento original. Já o X-Diff apresenta o *diff* a partir da mesma estrutura do documento XML original e mostra o documento completo.

CAPÍTULO 6 – CONCLUSÃO

Para controlar versões de documentos XML, é necessária uma forma de detectar exatamente quais são as diferenças entre elas, ou seja, o que foi modificado de uma versão para a outra em termos das estruturas que compõem um documento XML (elementos e atributos). Existem algumas abordagens dedicadas à detecção de diferenças em documentos XML que foram apresentadas nesta tese, tais como X-Diff (WANG; DEWITT; CAI, 2003), XyDiff (COBENA; ABITEBOUL; MARIAN, 2002), XRel_Change_SQL (SUNDARAM; MADRIA, 2012), XKeyMatch (SANTOS; HARA, 2007), KF-Diff+ (XU *et al.*, 2002), DiffX (AL-EKRAM; ADMA; BAYSAL, 2005), 3DM (LINDHOLM, 2004), Molhado (THAO; MUNSON, 2010), DOCTREEDIFF (RÖNNAU; PHILIPP; BORGHOFF, 2009), entre outras. No entanto, apesar destas abordagens levarem em consideração a estrutura do documento XML, elas se prendem à sintaxe, não possibilitando a identificação da semântica das modificações (i.e., razão por trás das ações de adição, remoção e alteração dos elementos e atributos do documento). Existem diversas aplicações onde essa informação sintática não é suficiente, ou seja, situações onde não basta detectar os elementos ou atributos que mudaram, mas que também é necessário inferir a razão das modificações.

Diante disso, esta tese apresentou a XChange, uma abordagem que processa os dados contidos em duas versões de um documento XML com o objetivo de compreender a razão das modificações, possibilitando a obtenção de conhecimento semântico a partir de informações explícitas nas versões ou deduzidas automaticamente através de regras de inferência. Enquanto uma abordagem puramente sintática apresentaria, por exemplo, um conjunto de elementos adicionados ou removidos, a XChange visa extrair desse conjunto o significado de alto nível da mudança, facilitando a sua compreensão.

6.1 RESULTADOS

Conforme apresentado no Capítulo 2, o controle de mudanças em documentos XML bem como a compreensão desta evolução ainda é um problema em aberto, que precisa de amadurecimento em relação a diversas funcionalidades necessárias para sua ampla utilização. Esta tese apresenta alguns resultados para aumentar o apoio existente na área, fornecendo soluções com aspectos inovadores, a saber:

- estabelece uma forma para apoiar a compreensão da evolução de documentos XML, levando em consideração as características destes documentos e a semântica associada. A partir do estudo experimental realizado com usuários, o

XChange se mostrou mais eficiente e mais eficaz na compreensão de mudanças de documentos XML, quando comparado ao X-Diff (WANG; DEWITT; CAI, 2003);

- identifica as diferenças no nível sintático e semântico, possibilitando detectar as diferenças entre as versões do documentos XML e compreender a razão real das modificações;
- estabelece uma forma eficiente para o casamento de elementos correspondentes entre versões de documentos XML. Duas abordagens foram propostas: casamento por chave e casamento por similaridade. Para o primeiro caso, dependendo de como os documentos XML são gerenciados, não há garantia de que o valor da chave permaneça o mesmo entre duas versões (por exemplo, se houver algum erro de digitação em *v1* que foi corrigido em *v2*). Outro ponto é que muitos documentos XML não têm um identificador associado. Por outro lado, a abordagem de cálculo de similaridade evita a necessidade de uso de atributos chave, mas em alguns casos efetua casamentos incorretos (por exemplo, em situações onde dois elementos têm informações associadas muito semelhantes). O XChange apresentou resultados sem diferença estatística significativa quando comparado ao X-Diff no que diz respeito a eficácia do método utilizado. A precisão assumiu valores acima de 70% e a cobertura superior a 85%, para as duas abordagens, em todos os cenários. Além disso, o XChange se mostrou cerca de 45 vezes mais rápido na identificação correta de elementos correspondentes a partir da abordagem baseada em similaridade, quando comparado ao X-Diff;
- fornece uma interface gráfica que permite a definição de regras de enriquecimento semântico a partir de uma seleção de opções em alto nível de abstração, não exigindo que o usuário seja especialista em Prolog;
- fornece um apoio semiautomático para a construção de regras de enriquecimento semântico baseado em *itemsets* frequentes. Através do algoritmo Apriori (AGRAWAL; SRIKANT, 1994), alguns *itemsets* frequentes são identificados e informados ao especialista para que sejam validados. Vale notar que a avaliação apresentada no Capítulo 5 fez uso desse mecanismo para gerar as regras de enriquecimento semântico, sem a intervenção do especialista, e os resultados obtidos tanto em termos de eficácia quanto em termos de eficiência foram superiores ao X-Diff.

Nesta pesquisa os seguintes resultados de publicação foram alcançados:

- OLIVEIRA, A.; OLIVEIRA, A. M.; BRAGANHOLO, V.; MURTA, L. Gerenciando Alterações em Documentos XML. REVISTA DE INFORMÁTICA TEÓRICA E APLICADA (RITA), 2010. v. 17;
- OLIVEIRA, A.; MURTA, L.; BRAGANHOLO, V. Uso de inferência na compreensão das modificações em documentos semiestruturados. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBD), São Paulo, SP: SBC, 2012. 65–72 p;
- MARTINS, G.; LARCHER, J.; OLIVEIRA, A.; MURTA, L.; BRAGANHOLO, V. XChange: Compreensão de Mudanças em Documentos XML. In: SESSÃO DE DEMOS DO SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBD), Recife, PE: SBC, 2013. 31–36 p;
- OLIVEIRA, A.; MURTA, L.; BRAGANHOLO, V. Towards Semantic Diff of XML Documents. In: SYMPOSIUM ON APPLIED COMPUTING (SAC), Gyeongju, Korea: ACM, 2014. 833–838 p.

Além disso, uma publicação relacionada aos resultados obtidos no Capítulo 4 foi submetida e duas publicações em fase de preparação envolvendo o mapeamento sistemático do Capítulo 2 está em fase de preparação:

- OLIVEIRA, A.; TESSAROLLI, G.; GHIOTTO, G.; PINTO, B.; CAMPELLO, F.; MARQUES, M.; OLIVEIRA, C.; RODRIGUES; I.; KALINOWSKI, M.; SOUZA, U.; MURTA, L.; BRAGANHOLO, V. An Efficient Similarity-based Approach for Comparing XML Documents. INFORMATION SYSTEMS. submitted. october. 2016;
- OLIVEIRA, A.; MURTA, L.; BRAGANHOLO, V., A Systematic Mapping of Semistructured Data Change Management. em desenvolvimento.
- OLIVEIRA, A.; MURTA, L.; BRAGANHOLO, V., XChange: Semantic Diff of XML Documents. em desenvolvimento.

Esta pesquisa também resultou nas seguintes orientações de projetos de iniciação científica e educação tutorial:

- GAZZOLA, P. O. L., GARCIA, P. A., Gerência de Configuração Aplicada a Documentos XML, Iniciação Científica, BIC/UFJF, 01/08/2011 a 31/07/2012;

- LARCHER JUNIOR, C. H. N., MARTINS, G. G., Uso de Inferência no Controle das Mudanças em Documentos XML, Iniciação Científica, BIC/UFJF, 01/09/2012 a 31/07/2013;
- SOUZA FILHO, F. M., Consulta e Compreensão de Mudanças em Documentos XML, Iniciação Científica, BIC/UFJF, 01/08/2013 a 31/07/2014;
- OLIVEIRA, C. R. C., Compreensão da Evolução de Documentos XML, GET-Comp/UFJF, 01/08/2012 a 01/02/2016.
- MARQUES, M. O. C., Compreensão da Evolução de Documentos XML, GET-Comp/UFJF, 01/06/2014 a 30/06/2016.

6.2 LIMITAÇÕES

É importante mencionar algumas limitações observadas no decorrer do estudo realizado por este trabalho. A primeira está relacionada ao uso do cálculo de similaridade na identificação de elementos correspondentes. Esta estratégia pode realizar casamentos incorretos, ainda que em pequeno número, como pode ser observado no estudo experimental descrito no Capítulo 4. Esses casamentos incorretos podem comprometer os resultados relacionados ao *diff* semântico. Além disso, apesar dos bons resultados relacionados à eficiência do XChange no mesmo estudo experimental, quando comparados ao X-Diff, o tempo de processamento ainda é considerável e pode comprometer aplicações online com documentos grandes.

Outra limitação deste trabalho é que a qualidade do *diff* semântico está relacionada à qualidade das regras de enriquecimento semântico fornecidas pelo apoio semiautomático e/ou definidas manualmente pelo especialista. Inicialmente, o apoio semiautomático é dependente de um conjunto de treinamento que pode não ser significativo ou não representar bem a evolução do documento XML em análise. Além disso, dependendo da configuração utilizada na definição das regras de enriquecimento ou até da experiência do especialista, o conjunto de sugestões fornecido pelo apoio semiautomático pode gerar um *diff* semântico pouco representativo. Além disso, o número de versões usado na identificação de itens frequentes pode afetar os resultados. É importante considerar também o aspecto temporal e a sazonalidade de alguns tipos de mudança.

6.3 TRABALHOS FUTUROS

É possível descrever novos focos de pesquisas a partir da abordagem apresentada nesta tese. Novos experimentos, modificações internas, melhorias relacionadas ao desempenho do *diff* semântico, alteração do algoritmo de similaridade e inclusão do *diff* entre variantes e

da detecção de conflitos e do *merge* semântico são as principais propostas para realização de trabalhos futuros.

Em relação aos estudos experimentais realizados, o primeiro deles, apresentado no Capítulo 4 envolveu apenas um *dataset* real. Outro ponto importante é que para este *dataset*, assim como na maioria dos casos (MAAROUF; CHUNG, 2008; VYHNANOVSKÁ; MLÝNKOVÁ, 2010; GRIJZENHOUT; MARX, 2013), não existia um esquema associado. Utilizar outros *datasets* reais inclusive com esquemas associados e diferentes distribuições relacionadas ao número de atributos e de níveis de elementos pode ser interessante para comparar os resultados obtidos nesta tese com as novas execuções. Além disso, uma sugestão é a realização novos estudos envolvendo o especialista e verificando a sua contribuição na definição das regras de enriquecimento semântico.

Como mencionado na Seção 2.2, o DTD e o XML *Schema* podem ser utilizados para descrever a estrutura de documentos XML. Entretanto, dados e estrutura de documentos XML tendem a evoluir com o passar do tempo. Estas modificações podem ser causadas por diversos fatores, tais como, correção de erros de projeto, expansão do escopo da aplicação, alterações no domínio (SU *et al.*, 2001). Na literatura, o problema de evolução de esquemas vem sendo estudado há algum tempo (BOUCHOU *et al.*, 2004; GUERRINI; MESITI; ROSSI, 2005; LEONARDI *et al.*, 2007; GUERRINI; MESITI, 2008). Uma possibilidade é estender o XChange, que leva em consideração somente a evolução dos dados, para que seja possível utilizar versões com esquemas diferentes no *diff* semântico.

Outra proposta está relacionada à paralelização do algoritmo baseado em similaridade para o aumento do desempenho do processo que apoia a identificação de elementos correspondentes entre versões de um documento XML. Uma possibilidade é paralelizar as etapas do cálculo de similaridade de atributos e de subelementos, uma vez que a comparação ocorre entre os elementos de uma versão e todos os elementos da outra versão.

Outra análise interessante diz respeito ao limiar de similaridade adequado para o estudo experimental. Nesta tese, num primeiro momento foi executada uma calibragem para este conjunto de dados utilizado. Um trabalho futuro consiste em verificar a possibilidade de definir um limiar, ou uma faixa de valores, para ser utilizado em qualquer conjunto de dados.

Com o surgimento da MDE - *Model Driven Engineering* (SCHMIDT, 2006) que permitiu que os desenvolvedores usassem abstrações de alto nível durante o desenvolvimento de sistemas, surgiu a necessidade de acompanhar a evolução dos modelos. Esta necessidade se justifica porque modelos possuem uma estrutura mais complexa do que o código por exemplo. Além de verificar a sintaxe, modelos requerem que durante o controle de sua evolução

também a sua semântica seja levada em consideração. De fato, na literatura existem pesquisas relacionadas a *diff*, detecção de conflitos e *merge* voltado para modelos, tanto no nível sintático, como semântico (CICCHETTI; RUSCIO; PIERANTONIO, 2008; KERSTIN ALTMANNINGER; MARTINA SEIDL; MANUEL WIMMER, 2009; ALTMANNINGER; SCHWINGER; KOTSIS, 2010; GERTH *et al.*, 2013). Diante disso, uma sugestão de trabalho futuro consiste na extensão do *diff* semântico do XChange para apoiar a compreensão da evolução dos modelos e até de outros itens de configuração como por exemplo, dos códigos gerados durante o processo de desenvolvimento de software.

Outro trabalho futuro, já em andamento na UFJF, está relacionado ao *diff* entre variantes e o *merge* semântico de documentos XML. Em algumas situações, quando usuários alteram em paralelo um mesmo documento XML, as modificações em um mesmo ponto ou em pontos diferentes do documento podem gerar inconsistências. Na literatura, este problema está sendo estudado e algumas abordagens vem sendo propostas (FONTAINE, 2002; LINDHOLM, 2004; RÖNNAU; PHILIPP; BORGHOFF, 2009). Para exemplificar, considere a versão *base.xml*, no cenário da prefeitura de Baltimore, apresentada na Figura **6.1**.

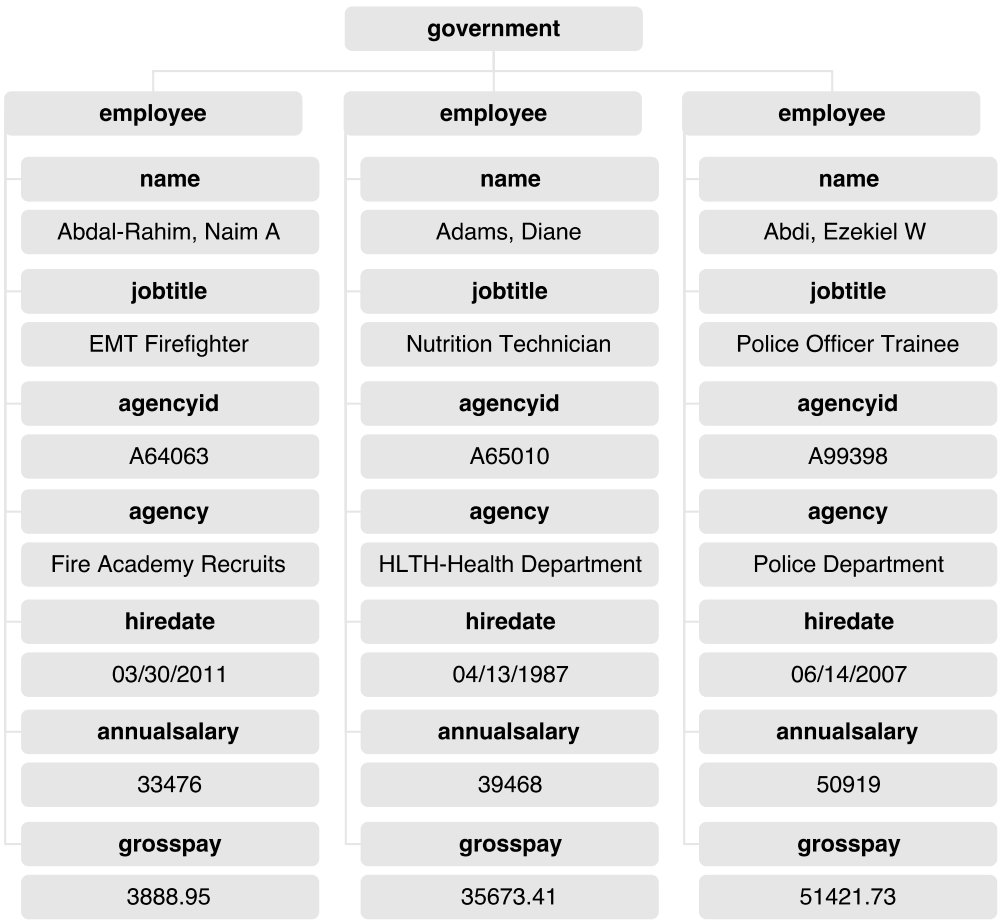

Figura 6.1**: Documento XML do cadastro de funcionários na versão *base.xml***

Suponha que um usuário fez algumas alterações na versão *v1.xml* que estava trabalhando, e entre elas, excluiu o funcionário *Adams, Diane*, destacado em vermelho na Figura 6.2, o que significa que este funcionário foi demitido.

Figura 6.2: Versão *v1.xml* com marcação de diferenças em comparação com a versão *base.xml* (amarelo representando alteração e vermelho representando remoção)

Por outro lado, um segundo usuário, trabalhando no mesmo documento, na versão *v2.xml* fez, entre outras ações, uma alteração relativa à promoção (aumento de salário e mudança de cargo) do mesmo funcionário (*Adams, Diane)*, como mostra a Figura 6.3. Estas alterações geram um conflito, inclusive no nível semântico, pois um funcionário não pode ser demitido e promovido ao mesmo tempo. A interdependência entre os trabalhos de diferentes usuários tornam o processo de consolidação de mudanças mais custoso. Alterações executadas por um usuário podem, de forma sutil, comprometer as alterações executadas por outros, levando a um processo de *merge* difícil e propenso a erros. A proposta de *diff* entre variantes e *merge* semântico é apoiar a identificação de cenários como este, levando em consideração a

semântica associada e contribuindo para a compreensão do que foi modificado (*diff*) bem como para a conciliação destas mudanças (*merge*).

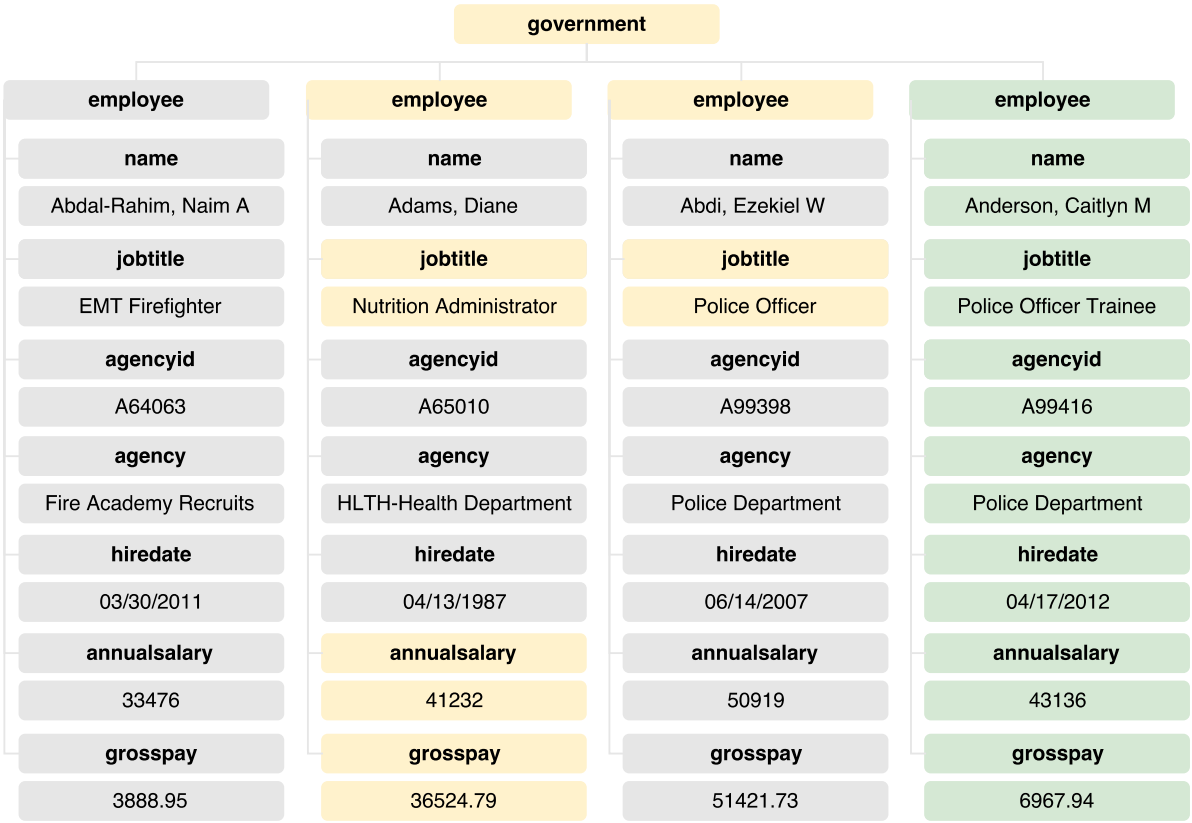

Figura 6.3: Versão *v2.xml* com marcação de diferenças em comparação com a versão *base.xml* (verde representando inclusão, amarelo representando alteração)

A Figura 6.4 apresenta a proposta de *diff* entre variantes e *merge* semântico do XChange. Os dados de entrada são constituídos por 3 versões de um documento XML (versão *base*, *v1* e *v2*), que são pré-processadas e transformadas em um conjunto de fatos Prolog, e por um conjunto de regras definidas no início do processo para detectar os conflitos semânticos no processo de inferência a partir da identificação das mudanças entre as versões. Na etapa de *merge*, os conflitos semânticos são então exibidos e para cada conflito gerado, deve-se informar qual das versões será mantida. Suponha que neste caso, o que realmente aconteceu foi a demissão do referido funcionário. Diante disso, a alteração efetuada na versão *v1.xml* será considerada no *merge* destas versões para gerar a versão *modificada.* Após resolver todos os conflitos semânticos, uma lista com os conflitos sintáticos ainda existentes seria exibida para que o usuário possa escolher qual das versões será mantida e gerar assim a versão modificada do documento XML.

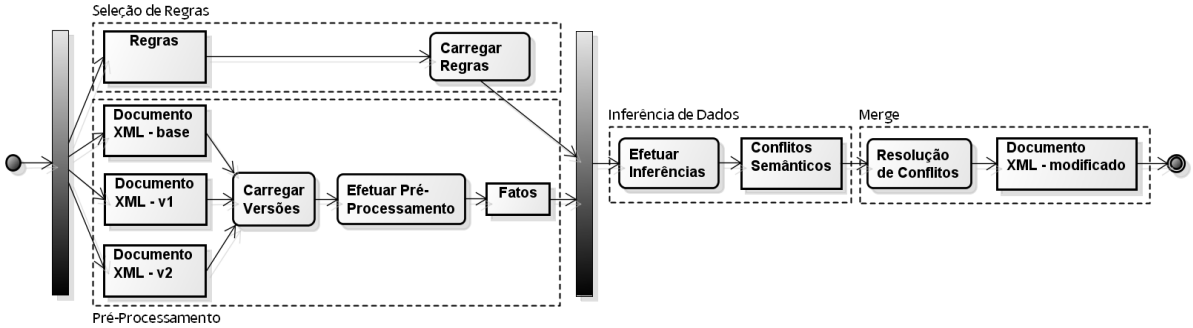

Figura 6.4: Proposta de *merge* semântico do XChange

Estas opções indicadas como trabalhos futuros apresentam o mesmo objetivo de possibilitar uma compreensão mais eficaz e eficiente da diferença entre as versões de documentos XML. Além disso, propõem tratar de forma semântica não somente o *diff* como também o *merge* de documentos XML e de outros itens de configuração, tais como modelos.

REFERÊNCIAS

AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining Association Rules Between Sets of Items in Large Databases. SIGMOD '93, New York, NY, USA: ACM, 1993. 207–216 p.

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES (VLDB), Santiago de Chile, Chile: Jorge B. Bocca and Matthias Jarke and Carlo Zaniolo, 1994. 487–499 p.

AL-EKRAM, R.; ADMA, A.; BAYSAL, O. diffX: An Algorithm to Detect Changes in Multi-version XML Documents. CASCON '05, Toranto, Ontario, Canada: IBM Press, 2005. 1–11 p.

ALTMANNINGER, K.; SCHWINGER, W.; KOTSIS, G. Semantics for Accurate Conflict Detection in SMoVer: Specification, Detection and Presentation by Example. *Int. J. Enterp. Inf. Syst.*, jan. 2010. v. 6.

ANDRADE, A.; RUBERG, G.; BAIÃO, F.; BRAGANHOLO, V.; MATTOSO, M. Efficiently Processing XML Queries over Fragmented Repositories with PartiX. In: INTERNATIONAL WORKSHOP ON DATABASE TECHNOLOGIES FOR HANDLING XML INFORMATION ON THE WEB (DATAX), Munich, Germany: [s.n.], 2006. 150–163 p.

ARGÜELLO, M.; DES, J.; FERNANDEZ-PRIETO, M. J.; PEREZ, R.; PANIAGUA, H. Executing Medical Guidelines on the Web: Towards Next Generation Healthcare. Em: MSC, T. A. B.; MSC, R. E. BS.; AMBA, M. P. D., MBA,, MBCS (Org.). . *Applications and Innovations in Intelligent Systems XVI*. [S.l.]: Springer London, 2009. p. 197–210. Disponível em: <http://link.springer.com/chapter/10.1007/978-1-84882-215-3_15>. Acesso em 17 ago. 2014.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern information retrieval*. New York, NY, USA: [s.n.], 1999. v. 463.

BALDASSARRE, M. T.; CAIVANO, D.; KITCHENHAM, B.; VISAGGIO, G. Systematic review of statistical process control: an experience report. EASE'07, [S.l.]: British Computer Society, 2007. 94–102 p.

BARBOSA, D.; MIGNET, L.; VELTRI, P. Studying the XML Web: Gathering Statistics from an XML Sample. *World Wide Web*, dez. 2005. v. 8.

BARNETT, V.; LEWIS, T. *Outliers in Statistical Data*. 3. ed. USA: Wiley, 1994.

BOLEY, H. The rule markup language: RDF-XML data model, XML schema hierarchy, and XSL transformations. In: INTERNATIONAL CONFERENCE ON WEB KNOWLEDGE MANAGEMENT AND DECISION SUPPORT (INAP), Berlin, Heidelberg: Springer-Verlag, 2003. 5–22 p.

BOUCHOU, B.; DUARTE, D.; ALVES, M. H. F.; LAURENT, D.; MUSICANTE, M. A. Schema Evolution for XML: A Consistency-Preserving Approach. Em: FIALA, J.; KOUBEK, V.; KRATOCHVÍL, J. (Org.). . *Mathematical Foundations of Computer Science 2004*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2004. p. 876–

888. Disponível em: <http://link.springer.com/chapter/10.1007/978-3-540-28629-5_69>. Acesso em 13 out. 2016.

BRATKO, I. *Prolog programming for artificial intelligence*. Harlow, England; New York: Addison Wesley, 2001.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F. *Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation.* Disponível em: <http://www.w3.org/TR/xml/>. Acesso em 9 mar. 2014.

BRIN, S.; MOTWANI, R.; ULLMAN, J. D.; TSUR, S. Dynamic Itemset Counting and Implication Rules for Market Basket Data. SIGMOD '97, New York, NY, USA: ACM, 1997. 255–264 p.

BUNEMAN, P.; DAVIDSON, S.; FAN, W.; HARA, C.; TAN, W.-C. Keys for XML. *Computer Networks*, Agosto 2002. v. 39.

BUNEMAN, P.; DAVIDSON, S.; FAN, W.; HARA, C.; TAN, W.-C. Reasoning about keys for XML. *Information Systems*, dez. 2003. v. 28.

CAMPELLO, F.; PINTO, B.; TESSAROLLI, G.; OLIVEIRA, A.; OLIVEIRA, C.; JUNIOR, M. O.; MURTA, L.; BRAGANHOLO, V. A similarity-based approach to match elements across versions of XML documents. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBD), Curitiba, PR, Brasil: SBC, 2014.

CHAMAKURA, S.; SACHDE, A.; CHAKRAVARTHY, S.; ARORA, A. WebVigiL: Monitoring Multiple Web Pages and Presentation of XML Pages. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE) - WORKSHOPS, Tokyo, Japan: IEEE Computer Society, 2005. 1276 p.

CHAWATHE, S. S.; GARCIA-MOLINA, H. Meaningful Change Detection in Structured Data. New York, USA: ACM, 1997. 26–37 p.

CICCHETTI, A.; RUSCIO, D. D.; PIERANTONIO, A. Managing Model Conflicts in Distributed Development. Em: CZARNECKI, K.; OBER, I.; BRUEL, J.-M.; UHL, A.; VÖLTER, M. (Org.). . *Model Driven Engineering Languages and Systems*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2008. p. 311–325. Disponível em: <http://link.springer.com/chapter/10.1007/978-3-540-87875-9_23>. Acesso em 24 nov. 2016.

CLIFF, N. *Ordinal Methods for Behavioral Data Analysis*. Mahwah, N.J: Psychology Press, 1996.

CNPQ. *CNPQ*. Disponível em: <http://lattes.cnpq.br/>. Acesso em 15 ago. 2012.

COBENA, G.; ABITEBOUL, S.; MARIAN, A. Detecting changes in XML documents. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), San Jose, California, USA: IEEE Computer Society, 2002. 41–52 p.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to Algorithms*. Third Edition ed. [S.l.]: The MIT Press, 2009.

DENTI, E.; OMICINI, A.; RICCI, A. tuProlog: A Light-weight Prolog for Internet Applications and Infrastructures. *Practical Aspects of Declarative Languages*, 2001.

DORNELES, C. F.; NUNES, M. F.; HEUSER, C. A.; MOREIRA, V. P.; SILVA, A. S. DA; MOURA, (EDLENO S. DE). A strategy for allowing meaningful and comparable scores in approximate matching. *Information Systems*, 2009. v. 34.

ELMASRI, R.; NAVATHE, S. *Fundamentals of Database Systems*. 6. ed. [S.l.]: Addison-Wesley, 2010.

FALLSIDE, D. C.; WALMSLEY, P. *XML Schema Part 0: Primer Second Edition. W3C Recommendation.* Disponível em: <http://www.w3.org/TR/xmlschema-0/>.

FONTAINE, R. L. Merging XML files: A new approach providing intelligent merge of XML data sets. In: XML EUROPE CONFERENCE, , 2002.

GERTH, C.; KÜSTER, J. M.; LUCKEY, M.; ENGELS, G. Detection and resolution of conflicting change operations in version management of process models. *Software & Systems Modeling*, 1 jul. 2013. v. 12.

GETOV, V. e-Science: The Added Value for Modern Discovery. *Computer*, v. 41, n. 11, p. 30–31, 2008.

GRIJZENHOUT, S.; MARX, M. The quality of the XML Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, mar. 2013. v. 19.

GUERRINI, G.; MESITI, M. X-Evolution: A Comprehensive Approach for XML Schema Evolution. [S.l.]: IEEE, set. 2008. 251–255 p.

GUERRINI, G.; MESITI, M.; ROSSI, D. Impact of XML schema evolution on valid documents., 2005.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 2009. v. 11.

HALLO CARRASCO, M.; MARTÍNEZ-GONZÁLEZ, M. M.; DE LA FUENTE REDONDO, P. Data Models for Version Management of Legislative Documents. *Journal of Information Science*, Agosto 2013. v. 39.

HORS, A. L.; HÉGARET, P. L.; WOOD, L.; NICOL, G.; ROBIE, J.; CHAMPION, M.; BYRNE, S. *Document Object Model (DOM) Level 3 Core Specification*. Disponível em: <http://www.w3.org/TR/DOM-Level-3-Core/>. Acesso em 16 fev. 2014.

HUANG, S. S.; GREEN, T. J.; LOO, B. T. Datalog and Emerging Applications: An Interactive Tutorial. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA (SIGMOD), New York, NY, USA: ACM, 2011. 1213–1216 p.

HUANG, S. S.; GREEN, T. J.; LOO, B. T. Datalog and emerging applications: An interactive tutorial., 2011.

JACKSON, D.; LADD, D. A. Semantic Diff: a tool for summarizing the effects of modifications. In: INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE (ICSM), Victoria, BC, Canada,: IEEE Computer Society, 1994. 243–252 p.

JACOB, J.; SACHDE, A.; CHAKRAVARTHY, S. CX-DIFF: A Change Detection Algorithm for XML Content and Change Presentation Issues for WebVigiL. *Conceptual Modeling for Novel Application Domains*. Lecture Notes in Computer Science. [S.l.]: Jeusfeld, ManfredA. and Pastor, Óscar, 2003. v. 2814. p. 273–284.

JACOB, J.; SACHDE, A.; CHAKRAVARTHY, S. CX-DIFF: A Change Detection Algorithm for XML Content and Change Visualization for WebVigiL. *Data & Knowledge Engineering*, 2005. v. 52.

JURISTO, N.; MORENO, A. M. *Basics of Software Engineering Experimentation*. [S.l.]: Kluwer Academic Publishers, 2001.

KERSTIN ALTMANNINGER; MARTINA SEIDL; MANUEL WIMMER. A survey on model versioning approaches. *International Journal of Web Information Systems*, 28 ago. 2009. v. 5.

KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. , TR/SE-0401.Technical Report, nº TR/SE-0401. Australia: Department of Computer Science, Keele University and National ICT, 2004.

KITCHENHAM, B. A.; MENDES, E.; TRAVASSOS, G. H. Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *IEEE Transactions on Software Engineering*, maio 2007. v. 33.

KUHN, H. W. The Hungarian Method for the Assignment Problem. *Naval Research Logistics*, 1955. v. 2.

LEON, A. *A Guide to Software Configuration Management*. Norwood, MA: Artech House Publishers, 2000.

LEONARDI, E.; HOAI, T. T.; BHOWMICK, S. S.; MADRIA, S. DTD-Diff: A Change Detection Algorithm for DTDs. *Data Knowl. Eng.*, maio 2007. v. 61.

LIM, S.; NG, Y.-K. An Automated Change Detection Algorithm for HTML Documents Based on Semantic Hierarchies. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), Heidelberg, Germany: IEEE Computer Society, 2001. 303–312 p.

LIM, S.; NG, Y.-K. Change Discovery of Hierarchically Structured, Order-Sensitive Data in HTML/XML Documents. In: SYMPOSIUM ON APPLICATIONS AND THE INTERNET (SAINT), Tokyo, Japan: IEEE Computer Society, 2004. 178 p.

LIMA, D.; DELGADO, C.; MURTA, L.; BRAGANHOLO, V. Towards Querying Implicit Knowledge in XML Documents. *Journal of Information and Data Management (JIDM)*, 2012. v. 3.

LINDHOLM, T. A Three-way Merge for XML Documents. In: ACM SYMPOSIUM ON DOCUMENT ENGINEERING (DOCENG), Milwaukee, Wisconsin, USA: ACM, 2004. 1–10 p.

MAAROUF, M. Y.; CHUNG, S. M. XML Integrated Environment for Service-Oriented Data Management. In: 2008 20TH IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, , nov. 2008. 361–368 p. v. 2.

MACHADO, L. C. ***Tradução de Consultas XPath para Predicados Prolog***. Niterói, RJ - Brasil: Universidade Federal Fluminense, 2016

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M. DE. Reconciling software development models: A quasi-systematic review. *J. Syst. Softw.*, Fevereiro 2012. v. 85.

MAIER, D. The Complexity of Some Problems on Subsequences and Supersequences. *J. ACM*, abr. 1978. v. 25.

MARIAN, A.; ABITEBOUL, S.; COBENA, G.; MIGNET, L. Change-centric management of versions in an XML warehouse. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES (VLDB), Roma, Italy: Morgan Kaufmann Publishers Inc., 2001. 581–590 p.

MAYOR'S OFFICE OF INFORMATION TECHNOLOGY. *OpenBaltimore*. Disponível em: <https://data.baltimorecity.gov/>. Acesso em 14 out. 2016.

MENS, T. A State-of-the-Art Survey on Software Merging. *IEEE Transactions on Software Engineering*, TSE, 2002. v. 28.

MIGNET, L.; BARBOSA, D.; VELTRI, P. The XML Web: A First Study. WWW '03, New York, NY, USA: ACM, 2003. 500–510 p.

MORO; BRAGANHOLO, V. Desmistificando XML: da Pesquisa à Prática Industrial. *Atualizações em Informática 2009*. [S.l.]: PUC-Rio, 2009. p. 231–278.

OLIVEIRA, A. *et al.* An Efficient Similarity-based Approach for Comparing XML Documents. *Submitted - Information Systems*, out. 2016.

PAI, M.; MCCULLOCH, M.; GORMAN, J. D.; PAI, N.; ENANORIA, W.; KENNEDY, G.; THARYAN, P.; COLFORD, J. M. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *The National Medical Journal of India*, abr. 2004. v. 17.

PETERS, L. Change Detection in XML Trees: a Survey. *Twente Student Conference on IT (TSCONTI)*, 2005.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. Swinton, UK, UK: British Computer Society, 2008. 68–77 p.

PINTO, B. F.; CAMPELLO, F. ***Cálculo de similaridade de documentos XML***. Niterói, RJ - Brasil: Ciência da Computação, IC-UFF, 2012

PORTAL BRASILEIRO DE DADOS ABERTOS. *Dados Abertos*. Disponível em: <http://dados.gov.br/>. Acesso em 2 jan. 2017.

ROMANO, J.; KROMREY, J. D.; CORAGGIO, J.; SKOWRONEK, J. Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys., 2006. 1–33 p.

RÖNNAU, S.; PHILIPP, G.; BORGHOFF, U. M. Efficient Change Control of XML Documents. In: ACM SYMPOSIUM ON DOCUMENT ENGINEERING (DOCENG), Munich, Germany: ACM, 2009. 3–12 p.

ROYSTON, P. Remark {AS R94}: A Remark on Algorithm {AS 181}: The {W}-test for Normality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1995. v. 44.

RUSU, L. I.; RAHAYU, W.; TANIAR, D. Mining Changes from Versions of Dynamic XML Documents. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY FROM XML DOCUMENTS (KDXD), Berlin, Heidelberg: Springer-Verlag, setembro 2006. 3–12 p. v. 3915.

SANTOS. ***Prolog Versus XQuery Processors: A Performance Evaluation Of XML Queries Processing Methods***. Niterói, RJ - Brasil: Universidade Federal Fluminense, 2015

SANTOS, R. C.; HARA, C. S. A Semantical Change Detection Algorithm for XML. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING & KNOWLEDGE ENGINEERING (SEKE), Boston, Massachusetts, USA: Knowledge Systems Institute Graduate School, 2007. 438–443 p.

SCHMIDT, D. C. Guest Editor's Introduction: Model-Driven Engineering. *Computer*, fev. 2006. v. 39.

SELKOW, S. M. The tree-to-tree editing problem. *Information processing letters*, 1977. v. 6.

SONG, Y.; BHOWMICK, S. S.; DEWEY, JR., C. F. BioDIFF: an effective fast change detection algorithm for biological annotations. In: INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS (DASFAA), Berlin, Heidelberg: Springer-Verlag, 2007. 275–287 p.

SU, H.; KRAMER, D.; CHEN, L.; CLAYPOOL, K.; RUNDENSTEINER, E. A. XEM: managing the evolution of XML documents. In: ELEVENTH INTERNATIONAL WORKSHOP ON RESEARCH ISSUES IN DATA ENGINEERING, 2001. PROCEEDINGS, , 2001. 103–110 p.

SUNDARAM, S.; MADRIA, S. K. A change detection system for unordered XML data using a relational model. *Data Knowledge Engineering*, 2012. v. 72.

TAI, K.-C. The Tree-to-Tree Correction Problem. *Journal of the ACM (JACM)*, 1979. v. 26.

THAO, C.; MUNSON, E. V. Using Versioned Tree Data Structure, Change Detection and Node Identity for Three-way XML Merging. In: ACM SYMPOSIUM ON DOCUMENT ENGINEERING (DOCENG), Manchester, UK: ACM, 2010. 77–86 p.

THUY, P. T. T.; LEE, Y.-K.; LEE, S. Semantic and structural similarities between XML Schemas for integration of ubiquitous healthcare data. *Personal and Ubiquitous Computing*, 1 out. 2013. v. 17.

VYHNANOVSKÁ, J.; MLÝNKOVÁ, I. Interactive inference of XML schemas. In: 2010 FOURTH INTERNATIONAL CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE (RCIS), , maio 2010. 191–202 p.

WANG, Y.; DEWITT, D. J.; CAI, J.-Y. X-Diff: an effective change detection algorithm for XML documents. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), Bangalore, India: IEEE Computer Society, 2003. 519–530 p.

WIKIMEDIA. *Wikimedia*. Disponível em: <https://dumps.wikimedia.org/>. Acesso em 2 jan. 2017.

WILCOXON, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, dez. 1945. v. 1.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

XML-DEV COMMUNITY. *SAX 2.0.1*. Disponível em: <http://www.saxproject.org/>. Acesso em 11 jun. 2016.

XU, H.; WU, Q.; WANG, H.; YANG, G.; JIA, Y. KF-Diff+: Highly Efficient Change Detection Algorithm for XML Documents. Em: MEERSMAN, R.; TARI, Z. (Org.). . *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2002. p. 1273–1286. Disponível em: <http://link.springer.com/chapter/10.1007/3-540-36124-3_80>. Acesso em 18 nov. 2016.

ZHANG, K.; SHASHA, D. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 1989. v. 18.

ZHAO, Q.; BHOWMICK, S. S.; MADRIA, S. Discovering Pattern-Based Dynamic Structures from Versions of Unordered XML Documents. In: INTERNATIONAL CONFERENCE DATA WAREHOUSING AND KNOWLEDGE DISCOVERY (DAWAK), Lecture Notes in Computer Science, Zaragoza, Spain: Springer Berlin Heidelberg, 2004. 77–86 p.

APÊNDICE A - DOCUMENTO XML – *V1*

```
<?xml version="1.0" encoding="UTF-8"?>
<government>
 <employee>
  <name>Abdal-Rahim,Naim A</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64063</agencyid>
  <agency>Fire Academy Recruits</agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>33476</annualsalary>
  <grosspay>33888.95</grosspay>
 </employee>
 <employee>
  <name>Adams,Nicholas B</name>
  <jobtitle>POLICE OFFICER</jobtitle>
  <agencyid>A99416</agencyid>
  <agency>Police Department</agency>
  <hiredate>2010-05-20T00:00:00</hiredate>
  <annualsalary>42391</annualsalary>
  <grosspay>37879.36</grosspay>
 </employee>
 <employee>
  <name>Addison,Rosalind D</name>
  <jobtitle>OFFICE SUPERVISOR</jobtitle>
  <agencyid>A49104</agencyid>
  <agency>TRANS-Highways</agency>
  <hiredate>2004-02-02T00:00:00</hiredate>
  <annualsalary>39210</annualsalary>
  <grosspay>34730.39</grosspay>
 </employee>
 <employee>
  <name>Adetola,Adewale A</name>
  <jobtitle>ACCOUNTANT I</jobtitle>
  <agencyid>A65001</agencyid>
  <agency>HLTH-Health Department</agency>
  <hiredate>2007-07-23T00:00:00</hiredate>
  <annualsalary>45498</annualsalary>
  <grosspay>34131.1</grosspay>
 </employee>
 <employee>
  <name>Albertson,Tyler K</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64063</agencyid>
  <agency>Fire Academy Recruits</agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>33476</annualsalary>
  <grosspay>33862.61</grosspay>
 </employee>
 <employee>
  <name>Alexander,Obray S</name>
  <jobtitle>SOLID WASTE WORKER</jobtitle>
  <agencyid>B70410</agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>1970-08-03T00:00:00</hiredate>
  <annualsalary>33446</annualsalary>
  <grosspay>31781.66</grosspay>
 </employee>
 <employee>
  <name>Anbinder,Robert D</name>
  <jobtitle>CHIEF SOLICITOR</jobtitle>
  <agencyid>A30002</agencyid>
```

```
  <agency>Law Department</agency>
  <hiredate>1994-07-02T00:00:00</hiredate>
  <annualsalary>89800</annualsalary>
  <grosspay>79093.67</grosspay>
 </employee>
 <employee>
  <name>Anderson,Linda L</name>
  <jobtitle>OFFICE ASSISTANT II</jobtitle>
  <agencyid>A65007</agencyid>
  <agency>HLTH-Health Department</agency>
  <hiredate>2008-10-30T00:00:00</hiredate>
  <annualsalary>26388</annualsalary>
  <grosspay>24461.46</grosspay>
 </employee>
 <employee>
  <name>Anderson,Patricia A</name>
  <jobtitle>Facilities/Office Services II</jobtitle>
  <agencyid>A03089</agencyid>
  <agency>OED-Employment Dev</agency>
  <hiredate>1976-12-13T00:00:00</hiredate>
  <annualsalary>46576</annualsalary>
  <grosspay>41353.26</grosspay>
 </employee>
 <employee>
  <name>Arrington,Vera G</name>
  <jobtitle>LABORER</jobtitle>
  <agencyid>B70357</agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>2000-07-03T00:00:00</hiredate>
  <annualsalary>28891</annualsalary>
  <grosspay>25459.87</grosspay>
 </employee>
 <employee>
  <name>Ayers,Geneva B</name>
  <jobtitle>ACCOUNTING ASST I</jobtitle>
  <agencyid>A14006</agencyid>
  <agency>FIN-Collections</agency>
  <hiredate>2005-09-01T00:00:00</hiredate>
  <annualsalary>29326</annualsalary>
  <grosspay>35452.34</grosspay>
 </employee>
</government>
```

APÊNDICE B - DOCUMENTO XML – *V2*

```
<?xml version="1.0" encoding="UTF-8"?>
<government>
 <employee>
  <name>Abdal-Rahim,Naim A</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64215</agencyid>
  <agency>Fire Department</agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>34146</annualsalary>
  <grosspay>35537.88</grosspay>
 </employee>
 <employee>
  <name>Abdul-Rahim,Anees</name>
  <jobtitle>CONTRACT SERV SPEC II</jobtitle>
  <agencyid>A09001</agencyid>
  <agency>Liquor License Board</agency>
  <hiredate>2011-09-12T00:00:00</hiredate>
  <annualsalary>28603</annualsalary>
  <grosspay>2720.96</grosspay>
 </employee>
 <employee>
  <name>Adams,Timothy L</name>
  <jobtitle>SOLID WASTE WORKER</jobtitle>
  <agencyid>B70411</agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>2010-10-19T00:00:00</hiredate>
  <annualsalary>28600</annualsalary>
  <grosspay>23199.86</grosspay>
 </employee>
 <employee>
  <name>Addison,Rosalind D</name>
  <jobtitle>ADMINISTRATIVE COORDINATOR</jobtitle>
  <agencyid>A23100</agencyid>
  <agency>FIN-Admin and Budgets</agency>
  <hiredate>2004-02-02T00:00:00</hiredate>
  <annualsalary>44486</annualsalary>
  <grosspay>41615.95</grosspay>
 </employee>
 <employee>
  <name>Adetola,Adewale A</name>
  <jobtitle>ACCOUNTANT II</jobtitle>
  <agencyid>A17001</agencyid>
  <agency>FIN-Purchasing</agency>
  <hiredate>2007-07-23T00:00:00</hiredate>
  <annualsalary>48900</annualsalary>
  <grosspay>47658.22</grosspay>
 </employee>
 <employee>
  <name>Ahmed,Jamila L</name>
  <jobtitle>POLICE OFFICER</jobtitle>
  <agencyid>A99004</agencyid>
  <agency>Police Department</agency>
  <hiredate>2010-06-28T00:00:00</hiredate>
  <annualsalary>43895</annualsalary>
  <grosspay>46147.95</grosspay>
 </employee>
 <employee>
  <name>Albertson,Tyler K</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64140</agencyid>
```

```
  <agency>Fire Department</agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>34146</annualsalary>
  <grosspay>36806.23</grosspay>
 </employee>
 <employee>
  <name>Anbinder,Robert D</name>
  <jobtitle>EXECUTIVE LEVEL II</jobtitle>
  <agencyid>A30002</agencyid>
  <agency>Law Department</agency>
  <hiredate>1994-07-02T00:00:00</hiredate>
  <annualsalary>93000</annualsalary>
  <grosspay>92356.48</grosspay>
 </employee>
 <employee>
  <name>Anderson,Caitlyn M</name>
  <jobtitle>POLICE OFFICER TRAINEE</jobtitle>
  <agencyid>A99416</agencyid>
  <agency>Police Department</agency>
  <hiredate>2012-04-17T00:00:00</hiredate>
  <annualsalary>43136</annualsalary>
  <grosspay>6967.94</grosspay>
 </employee>
 <employee>
  <name>Anderson,Linda L</name>
  <jobtitle>OFFICE ASSISTANT II</jobtitle>
  <agencyid>A46001</agencyid>
  <agency>M-R Environmental Cntrl</agency>
  <hiredate>2008-10-30T00:00:00</hiredate>
  <annualsalary>26916</annualsalary>
  <grosspay>25935.36</grosspay>
 </employee>
 <employee>
  <name>Anderson,Patricia A</name>
  <jobtitle>Facilities/Office Services II</jobtitle>
  <agencyid>A03089</agencyid>
  <agency>OED-Employment Dev</agency>
  <hiredate>1976-12-13T00:00:00</hiredate>
  <annualsalary>48142</annualsalary>
  <grosspay>48246.29</grosspay>
 </employee>
 <employee>
  <name>Arrington,Vera G</name>
  <jobtitle>LABORER</jobtitle>
  <agencyid>B70357</agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>2000-07-03T00:00:00</hiredate>
  <annualsalary>29515</annualsalary>
  <grosspay>30824.02</grosspay>
 </employee>
 <employee>
  <name>Ayers,Geneva B</name>
  <jobtitle>ACCOUNTING ASST II</jobtitle>
  <agencyid>A17002</agencyid>
  <agency>FIN-Purchasing</agency>
  <hiredate>2005-09-01T00:00:00</hiredate>
  <annualsalary>31741</annualsalary>
  <grosspay>36312.6</grosspay>
 </employee>
</government>
```

APÊNDICE C - *DELTA* RESULTANTE DO X-DIFF

```
<?xml version="1.0" encoding="UTF-8"?>
<government>
 <employee>
  <name>Abdal-Rahim,Naim A</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64215<?UPDATE FROM "A64063"?></agencyid>
  <agency>Fire Department<?UPDATE FROM "Fire Academy Recruits"?></agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>34146<?UPDATE FROM "33476"?></annualsalary>
  <grosspay>35537.88<?UPDATE FROM "33888.95"?></grosspay>
 </employee>
 <employee>
  <name>Ahmed,Jamila L<?UPDATE FROM "Adams,Nicholas B"?></name>
  <jobtitle>POLICE OFFICER</jobtitle>
  <agencyid>A99004<?UPDATE FROM "A99416"?></agencyid>
  <agency>Police Department</agency>
  <hiredate>2010-06-28T00:00:00<?UPDATE FROM "2010-05-20T00:00:00"?></hiredate>
  <annualsalary>43895<?UPDATE FROM "42391"?></annualsalary>
  <grosspay>46147.95<?UPDATE FROM "37879.36"?></grosspay>
 </employee>
 <employee>
  <name>Addison,Rosalind D</name>
  <jobtitle>ADMINISTRATIVE COORDINATOR<?UPDATE FROM "OFFICE
SUPERVISOR"?></jobtitle>
  <agencyid>A23100<?UPDATE FROM "A49104"?></agencyid>
  <agency>FIN-Admin and Budgets<?UPDATE FROM "TRANS-Highways"?></agency>
  <hiredate>2004-02-02T00:00:00</hiredate>
  <annualsalary>44486<?UPDATE FROM "39210"?></annualsalary>
  <grosspay>41615.95<?UPDATE FROM "34730.39"?></grosspay>
 </employee>
 <employee>
  <name>Adetola,Adewale A</name>
  <jobtitle>ACCOUNTANT II<?UPDATE FROM "ACCOUNTANT I"?></jobtitle>
  <agencyid>A17001<?UPDATE FROM "A65001"?></agencyid>
  <agency>FIN-Purchasing<?UPDATE FROM "HLTH-Health Department"?></agency>
  <hiredate>2007-07-23T00:00:00</hiredate>
  <annualsalary>48900<?UPDATE FROM "45498"?></annualsalary>
  <grosspay>47658.22<?UPDATE FROM "34131.1"?></grosspay>
 </employee>
 <employee>
  <name>Albertson,Tyler K</name>
  <jobtitle>EMT Firefighter</jobtitle>
  <agencyid>A64140<?UPDATE FROM "A64063"?></agencyid>
  <agency>Fire Department<?UPDATE FROM "Fire Academy Recruits"?></agency>
  <hiredate>2011-03-30T00:00:00</hiredate>
  <annualsalary>34146<?UPDATE FROM "33476"?></annualsalary>
  <grosspay>36806.23<?UPDATE FROM "33862.61"?></grosspay>
 </employee>
 <employee>
  <name>Adams,Timothy L<?UPDATE FROM "Alexander,Obray S"?></name>
  <jobtitle>SOLID WASTE WORKER</jobtitle>
  <agencyid>B70411<?UPDATE FROM "B70410"?></agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>2010-10-19T00:00:00<?UPDATE FROM "1970-08-03T00:00:00"?></hiredate>
  <annualsalary>28600<?UPDATE FROM "33446"?></annualsalary>
  <grosspay>23199.86<?UPDATE FROM "31781.66"?></grosspay>
 </employee>
 <employee>
  <name>Anbinder,Robert D</name>
  <jobtitle>EXECUTIVE LEVEL II<?UPDATE FROM "CHIEF SOLICITOR"?></jobtitle>
  <agencyid>A30002</agencyid>
  <agency>Law Department</agency>
  <hiredate>1994-07-02T00:00:00</hiredate>
  <annualsalary>93000<?UPDATE FROM "89800"?></annualsalary>
```

```
  <grosspay>92356.48<?UPDATE FROM "79093.67"?></grosspay>
 </employee>
 <employee>
  <name>Anderson,Linda L</name>
  <jobtitle>OFFICE ASSISTANT II</jobtitle>
  <agencyid>A46001<?UPDATE FROM "A65007"?></agencyid>
  <agency>M-R Environmental Cntrl<?UPDATE FROM "HLTH-Health Department"?></agency>
  <hiredate>2008-10-30T00:00:00</hiredate>
  <annualsalary>26916<?UPDATE FROM "26388"?></annualsalary>
  <grosspay>25935.36<?UPDATE FROM "24461.46"?></grosspay>
 </employee>
 <employee>
  <name>Anderson,Patricia A</name>
  <jobtitle>Facilities/Office Services II</jobtitle>
  <agencyid>A03089</agencyid>
  <agency>OED-Employment Dev</agency>
  <hiredate>1976-12-13T00:00:00</hiredate>
  <annualsalary>48142<?UPDATE FROM "46576"?></annualsalary>
  <grosspay>48246.29<?UPDATE FROM "41353.26"?></grosspay>
 </employee>
 <employee>
  <name>Arrington,Vera G</name>
  <jobtitle>LABORER</jobtitle>
  <agencyid>B70357</agencyid>
  <agency>DPW-Solid Waste</agency>
  <hiredate>2000-07-03T00:00:00</hiredate>
  <annualsalary>29515<?UPDATE FROM "28891"?></annualsalary>
  <grosspay>30824.02<?UPDATE FROM "25459.87"?></grosspay>
 </employee>
 <employee>
  <name>Ayers,Geneva B</name>
  <jobtitle>ACCOUNTING ASST II<?UPDATE FROM "ACCOUNTING ASST I"?></jobtitle>
  <agencyid>A17002<?UPDATE FROM "A14006"?></agencyid>
  <agency>FIN-Purchasing<?UPDATE FROM "FIN-Collections"?></agency>
  <hiredate>2005-09-01T00:00:00</hiredate>
  <annualsalary>31741<?UPDATE FROM "29326"?></annualsalary>
  <grosspay>36312.6<?UPDATE FROM "35452.34"?></grosspay>
 </employee>
 <employee>
  <?INSERT employee?>
  <name>Abdul-Rahim,Anees</name>
  <jobtitle>CONTRACT SERV SPEC II</jobtitle>
  <agencyid>A09001</agencyid>
  <agency>Liquor License Board</agency>
  <hiredate>2011-09-12T00:00:00</hiredate>
  <annualsalary>28603</annualsalary>
  <grosspay>2720.96</grosspay>
 </employee>
 <employee>
  <?INSERT employee?>
  <name>Anderson,Caitlyn M</name>
  <jobtitle>POLICE OFFICER TRAINEE</jobtitle>
  <agencyid>A99416</agencyid>
  <agency>Police Department</agency>
  <hiredate>2012-04-17T00:00:00</hiredate>
  <annualsalary>43136</annualsalary>
  <grosspay>6967.94</grosspay>
 </employee>
</government>
```

APÊNDICE D - *DELTA* RESULTANTE DO XCHANGE

```
<?xml version="1.0"?>
<diff-set from="v1 - Mon May 11 16:36:36 2011" to="v2 - Mon Jun 12 16:36:36 2012">
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agency" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="agencyid" type="different"/>
  </description>
  <delta count="3" annualsalary="11093">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <agencyid before="A49104" after="A23100"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <agencyid before="A65001" after="A17001"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="agencyid" type="different"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="3" annualsalary="11093" grosspay="21272.94">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <agencyid before="A49104" after="A23100"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <agencyid before="A65001" after="A17001"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <agencyid before="A14006" after="A17002"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="agencyid" type="different"/>
  </description>
```

```
  <delta count="3" annualsalary="11093">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <agencyid before="A49104" after="A23100"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <agencyid before="A65001" after="A17001"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agency" type="different"/>
   <change attr="agencyid" type="different"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="3" grosspay="21272.94">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <agencyid before="A49104" after="A23100"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <agencyid before="A65001" after="A17001"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <agencyid before="A14006" after="A17002"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agencyid" type="different"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="3" grosspay="21272.94">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agencyid before="A49104" after="A23100"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agencyid before="A65001" after="A17001"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agencyid before="A14006" after="A17002"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
```

```
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agency" type="different"/>
   <change attr="agencyid" type="different"/>
  </description>
  <delta count="3">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <agencyid before="A49104" after="A23100"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <agencyid before="A65001" after="A17001"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agencyid" type="different"/>
  </description>
  <delta count="3">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agencyid before="A49104" after="A23100"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agencyid before="A65001" after="A17001"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agency" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="3" annualsalary="11093" grosspay="21272.94">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
```

```
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="4" annualsalary="14293" grosspay="34535.75">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Anbinder,Robert D">
    <jobtitle before="CHIEF SOLICITOR" after="EXECUTIVE LEVEL II"/>
    <annualsalary before="89800" after="93000" delta="3200"/>
    <grosspay before="79093.67" after="92356.48" delta="13262.81"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="agency" type="different"/>
   <change attr="annualsalary" type="increased"/>
  </description>
  <delta count="3" annualsalary="11093">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="jobtitle" type="different"/>
   <change attr="annualsalary" type="increased"/>
  </description>
  <delta count="4" annualsalary="14293">
   <employee name="Addison,Rosalind D">
    <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
```

```
    </employee>
    <employee name="Adetola,Adewale A">
     <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
     <annualsalary before="45498" after="48900" delta="3402"/>
    </employee>
    <employee name="Anbinder,Robert D">
     <jobtitle before="CHIEF SOLICITOR" after="EXECUTIVE LEVEL II"/>
     <annualsalary before="89800" after="93000" delta="3200"/>
    </employee>
    <employee name="Ayers,Geneva B">
     <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
     <annualsalary before="29326" after="31741" delta="2415"/>
    </employee>
   </delta>
  </diff>
  <diff>
   <description>
    <change attr="agency" type="different"/>
    <change attr="annualsalary" type="increased"/>
    <change attr="agencyid" type="different"/>
    <change attr="grosspay" type="increased"/>
   </description>
   <delta count="6" annualsalary="12961" grosspay="27339.39">
    <employee name="Abdal-Rahim,Naim A">
     <agency before="Fire Academy Recruits" after="Fire Department"/>
     <annualsalary before="33476" after="34146" delta="670"/>
     <agencyid before="A64063" after="A64215"/>
     <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
    </employee>
    <employee name="Addison,Rosalind D">
     <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
     <annualsalary before="39210" after="44486" delta="5276"/>
     <agencyid before="A49104" after="A23100"/>
     <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
    </employee>
    <employee name="Adetola,Adewale A">
     <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
     <annualsalary before="45498" after="48900" delta="3402"/>
     <agencyid before="A65001" after="A17001"/>
     <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
    </employee>
    <employee name="Albertson,Tyler K">
     <agency before="Fire Academy Recruits" after="Fire Department"/>
     <annualsalary before="33476" after="34146" delta="670"/>
     <agencyid before="A64063" after="A64140"/>
     <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
    </employee>
    <employee name="Anderson,Linda L">
     <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
     <annualsalary before="26388" after="26916" delta="528"/>
     <agencyid before="A65007" after="A46001"/>
     <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
    </employee>
    <employee name="Ayers,Geneva B">
     <agency before="FIN-Collections" after="FIN-Purchasing"/>
     <annualsalary before="29326" after="31741" delta="2415"/>
     <agencyid before="A14006" after="A17002"/>
     <grosspay before="35452.34" after="36312.6" delta="860.26"/>
    </employee>
   </delta>
  </diff>
  <diff>
   <description>
    <change attr="jobtitle" type="different"/>
    <change attr="agency" type="different"/>
    <change attr="grosspay" type="increased"/>
   </description>
   <delta count="3" grosspay="21272.94">
```

```
    <employee name="Addison,Rosalind D">
     <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
     <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
     <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
    </employee>
    <employee name="Adetola,Adewale A">
     <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
     <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
     <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
    </employee>
    <employee name="Ayers,Geneva B">
     <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
     <agency before="FIN-Collections" after="FIN-Purchasing"/>
     <grosspay before="35452.34" after="36312.6" delta="860.26"/>
    </employee>
   </delta>
  </diff>
  <diff>
   <description>
    <change attr="jobtitle" type="different"/>
    <change attr="grosspay" type="increased"/>
   </description>
   <delta count="4" grosspay="34535.75">
    <employee name="Addison,Rosalind D">
     <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
     <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
    </employee>
    <employee name="Adetola,Adewale A">
     <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
     <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
    </employee>
    <employee name="Anbinder,Robert D">
     <jobtitle before="CHIEF SOLICITOR" after="EXECUTIVE LEVEL II"/>
     <grosspay before="79093.67" after="92356.48" delta="13262.81"/>
    </employee>
    <employee name="Ayers,Geneva B">
     <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
     <grosspay before="35452.34" after="36312.6" delta="860.26"/>
    </employee>
   </delta>
  </diff>
  <diff>
   <description>
    <change attr="agency" type="different"/>
    <change attr="agencyid" type="different"/>
   </description>
   <delta count="6">
    <employee name="Abdal-Rahim,Naim A">
     <agency before="Fire Academy Recruits" after="Fire Department"/>
     <agencyid before="A64063" after="A64215"/>
    </employee>
    <employee name="Addison,Rosalind D">
     <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
     <agencyid before="A49104" after="A23100"/>
    </employee>
    <employee name="Adetola,Adewale A">
     <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
     <agencyid before="A65001" after="A17001"/>
    </employee>
    <employee name="Albertson,Tyler K">
     <agency before="Fire Academy Recruits" after="Fire Department"/>
     <agencyid before="A64063" after="A64140"/>
    </employee>
    <employee name="Anderson,Linda L">
     <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
     <agencyid before="A65007" after="A46001"/>
    </employee>
    <employee name="Ayers,Geneva B">
```

```
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="annualsalary" type="increased"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="9" annualsalary="18351" grosspay="52859.38">
   <employee name="Abdal-Rahim,Naim A">
    <annualsalary before="33476" after="34146" delta="670"/>
    <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
   </employee>
   <employee name="Addison,Rosalind D">
    <annualsalary before="39210" after="44486" delta="5276"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <annualsalary before="45498" after="48900" delta="3402"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Albertson,Tyler K">
    <annualsalary before="33476" after="34146" delta="670"/>
    <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
   </employee>
   <employee name="Anbinder,Robert D">
    <annualsalary before="89800" after="93000" delta="3200"/>
    <grosspay before="79093.67" after="92356.48" delta="13262.81"/>
   </employee>
   <employee name="Anderson,Linda L">
    <annualsalary before="26388" after="26916" delta="528"/>
    <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
   </employee>
   <employee name="Anderson,Patricia A">
    <annualsalary before="46576" after="48142" delta="1566"/>
    <grosspay before="41353.26" after="48246.29" delta="6893.03"/>
   </employee>
   <employee name="Arrington,Vera G">
    <annualsalary before="28891" after="29515" delta="624"/>
    <grosspay before="25459.87" after="30824.02" delta="5364.15"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <annualsalary before="29326" after="31741" delta="2415"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="annualsalary" type="increased"/>
   <change attr="agencyid" type="different"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="6" annualsalary="12961" grosspay="27339.39">
   <employee name="Abdal-Rahim,Naim A">
    <annualsalary before="33476" after="34146" delta="670"/>
    <agencyid before="A64063" after="A64215"/>
    <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
   </employee>
   <employee name="Addison,Rosalind D">
    <annualsalary before="39210" after="44486" delta="5276"/>
    <agencyid before="A49104" after="A23100"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <annualsalary before="45498" after="48900" delta="3402"/>
```

```
    <agencyid before="A65001" after="A17001"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Albertson,Tyler K">
    <annualsalary before="33476" after="34146" delta="670"/>
    <agencyid before="A64063" after="A64140"/>
    <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
   </employee>
   <employee name="Anderson,Linda L">
    <annualsalary before="26388" after="26916" delta="528"/>
    <agencyid before="A65007" after="A46001"/>
    <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <annualsalary before="29326" after="31741" delta="2415"/>
    <agencyid before="A14006" after="A17002"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="agency" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="agencyid" type="different"/>
  </description>
  <delta count="6" annualsalary="12961">
   <employee name="Abdal-Rahim,Naim A">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
    <annualsalary before="33476" after="34146" delta="670"/>
    <agencyid before="A64063" after="A64215"/>
   </employee>
   <employee name="Addison,Rosalind D">
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <agencyid before="A49104" after="A23100"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <agencyid before="A65001" after="A17001"/>
   </employee>
   <employee name="Albertson,Tyler K">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
    <annualsalary before="33476" after="34146" delta="670"/>
    <agencyid before="A64063" after="A64140"/>
   </employee>
   <employee name="Anderson,Linda L">
    <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
    <annualsalary before="26388" after="26916" delta="528"/>
    <agencyid before="A65007" after="A46001"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <agencyid before="A14006" after="A17002"/>
   </employee>
  </delta>
</diff>
<diff>
  <description>
   <change attr="agency" type="different"/>
   <change attr="agencyid" type="different"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="6" grosspay="27339.39">
   <employee name="Abdal-Rahim,Naim A">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
```

```
    <agencyid before="A64063" after="A64215"/>
    <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
   </employee>
   <employee name="Addison,Rosalind D">
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <agencyid before="A49104" after="A23100"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <agencyid before="A65001" after="A17001"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Albertson,Tyler K">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
    <agencyid before="A64063" after="A64140"/>
    <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
   </employee>
   <employee name="Anderson,Linda L">
    <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
    <agencyid before="A65007" after="A46001"/>
    <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <agencyid before="A14006" after="A17002"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="agency" type="different"/>
   <change attr="annualsalary" type="increased"/>
   <change attr="grosspay" type="increased"/>
  </description>
  <delta count="6" annualsalary="12961" grosspay="27339.39">
   <employee name="Abdal-Rahim,Naim A">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
    <annualsalary before="33476" after="34146" delta="670"/>
    <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
   </employee>
   <employee name="Addison,Rosalind D">
    <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
    <annualsalary before="39210" after="44486" delta="5276"/>
    <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
   </employee>
   <employee name="Adetola,Adewale A">
    <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
    <annualsalary before="45498" after="48900" delta="3402"/>
    <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
   </employee>
   <employee name="Albertson,Tyler K">
    <agency before="Fire Academy Recruits" after="Fire Department"/>
    <annualsalary before="33476" after="34146" delta="670"/>
    <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
   </employee>
   <employee name="Anderson,Linda L">
    <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
    <annualsalary before="26388" after="26916" delta="528"/>
    <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
   </employee>
   <employee name="Ayers,Geneva B">
    <agency before="FIN-Collections" after="FIN-Purchasing"/>
    <annualsalary before="29326" after="31741" delta="2415"/>
    <grosspay before="35452.34" after="36312.6" delta="860.26"/>
   </employee>
  </delta>
```

```
</diff>
<diff>
 <description>
  <change attr="jobtitle" type="different"/>
 </description>
 <delta count="4">
  <employee name="Addison,Rosalind D">
   <jobtitle before="OFFICE SUPERVISOR" after="ADMINISTRATIVE COORDINATOR"/>
  </employee>
  <employee name="Adetola,Adewale A">
   <jobtitle before="ACCOUNTANT I" after="ACCOUNTANT II"/>
  </employee>
  <employee name="Anbinder,Robert D">
   <jobtitle before="CHIEF SOLICITOR" after="EXECUTIVE LEVEL II"/>
  </employee>
  <employee name="Ayers,Geneva B">
   <jobtitle before="ACCOUNTING ASST I" after="ACCOUNTING ASST II"/>
  </employee>
 </delta>
</diff>
<diff>
 <description>
  <change attr="agencyid" type="different"/>
 </description>
 <delta count="6">
  <employee name="Abdal-Rahim,Naim A">
   <agencyid before="A64063" after="A64215"/>
  </employee>
  <employee name="Addison,Rosalind D">
   <agencyid before="A49104" after="A23100"/>
  </employee>
  <employee name="Adetola,Adewale A">
   <agencyid before="A65001" after="A17001"/>
  </employee>
  <employee name="Albertson,Tyler K">
   <agencyid before="A64063" after="A64140"/>
  </employee>
  <employee name="Anderson,Linda L">
   <agencyid before="A65007" after="A46001"/>
  </employee>
  <employee name="Ayers,Geneva B">
   <agencyid before="A14006" after="A17002"/>
  </employee>
 </delta>
</diff>
<diff>
 <description>
  <change attr="agency" type="different"/>
 </description>
 <delta count="6">
  <employee name="Abdal-Rahim,Naim A">
   <agency before="Fire Academy Recruits" after="Fire Department"/>
  </employee>
  <employee name="Addison,Rosalind D">
   <agency before="TRANS-Highways" after="FIN-Admin and Budgets"/>
  </employee>
  <employee name="Adetola,Adewale A">
   <agency before="HLTH-Health Department" after="FIN-Purchasing"/>
  </employee>
  <employee name="Albertson,Tyler K">
   <agency before="Fire Academy Recruits" after="Fire Department"/>
  </employee>
  <employee name="Anderson,Linda L">
   <agency before="HLTH-Health Department" after="M-R Environmental Cntrl"/>
  </employee>
  <employee name="Ayers,Geneva B">
   <agency before="FIN-Collections" after="FIN-Purchasing"/>
  </employee>
```

```
  </delta>
</diff>
<diff>
  <description>
    <change attr="annualsalary" type="different"/>
  </description>
  <delta count="9" annualsalary="18351">
    <employee name="Abdal-Rahim,Naim A">
      <annualsalary before="33476" after="34146" delta="670"/>
    </employee>
    <employee name="Addison,Rosalind D">
      <annualsalary before="39210" after="44486" delta="5276"/>
    </employee>
    <employee name="Adetola,Adewale A">
      <annualsalary before="45498" after="48900" delta="3402"/>
    </employee>
    <employee name="Albertson,Tyler K">
      <annualsalary before="33476" after="34146" delta="670"/>
    </employee>
    <employee name="Anbinder,Robert D">
      <annualsalary before="89800" after="93000" delta="3200"/>
    </employee>
    <employee name="Anderson,Linda L">
      <annualsalary before="26388" after="26916" delta="528"/>
    </employee>
    <employee name="Anderson,Patricia A">
      <annualsalary before="46576" after="48142" delta="1566"/>
    </employee>
    <employee name="Arrington,Vera G">
      <annualsalary before="28891" after="29515" delta="624"/>
    </employee>
    <employee name="Ayers,Geneva B">
      <annualsalary before="29326" after="31741" delta="2415"/>
    </employee>
  </delta>
</diff>
<diff>
  <description>
    <change attr="grosspay" type="different"/>
  </description>
  <delta count="9" grosspay="52859.38">
    <employee name="Abdal-Rahim,Naim A">
      <grosspay before="33888.95" after="35537.88" delta="1648.93"/>
    </employee>
    <employee name="Addison,Rosalind D">
      <grosspay before="34730.39" after="41615.95" delta="6885.56"/>
    </employee>
    <employee name="Adetola,Adewale A">
      <grosspay before="34131.1" after="47658.22" delta="13527.12"/>
    </employee>
    <employee name="Albertson,Tyler K">
      <grosspay before="33862.61" after="36806.23" delta="2943.62"/>
    </employee>
    <employee name="Anbinder,Robert D">
      <grosspay before="79093.67" after="92356.48" delta="13262.81"/>
    </employee>
    <employee name="Anderson,Linda L">
      <grosspay before="24461.46" after="25935.36" delta="1473.9"/>
    </employee>
    <employee name="Anderson,Patricia A">
      <grosspay before="41353.26" after="48246.29" delta="6893.03"/>
    </employee>
    <employee name="Arrington,Vera G">
      <grosspay before="25459.87" after="30824.02" delta="5364.15"/>
    </employee>
    <employee name="Ayers,Geneva B">
      <grosspay before="35452.34" after="36312.6" delta="860.26"/>
    </employee>
```

```
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="name" type="inserted"/>
  </description>
  <delta count="4" annualsalary="144234" grosspay="79036.71">
   <employee name="Abdul-Rahim,Anees">
    <jobtitle>CONTRACT SERV SPEC II</jobtitle>
    <agencyid>A09001</agencyid>
    <agency>Liquor License Board</agency>
    <hiredate>2011-09-12T00:00:00</hiredate>
    <annualsalary>28603</annualsalary>
    <grosspay>2720.96</grosspay>
   </employee>
   <employee name="Adams,Timothy L">
    <jobtitle>SOLID WASTE WORKER</jobtitle>
    <agencyid>B70411</agencyid>
    <agency>DPW-Solid Waste</agency>
    <hiredate>2010-10-19T00:00:00</hiredate>
    <annualsalary>28600</annualsalary>
    <grosspay>23199.86</grosspay>
   </employee>
   <employee name="Ahmed,Jamila L">
    <jobtitle>POLICE OFFICER</jobtitle>
    <agencyid>A99004</agencyid>
    <agency>Police Department</agency>
    <hiredate>2010-06-28T00:00:00</hiredate>
    <annualsalary>43895</annualsalary>
    <grosspay>46147.95</grosspay>
   </employee>
   <employee name="Anderson,Caitlyn M">
    <jobtitle>POLICE OFFICER TRAINEE</jobtitle>
    <agencyid>A99416</agencyid>
    <agency>Police Department</agency>
    <hiredate>2012-04-17T00:00:00</hiredate>
    <annualsalary>43136</annualsalary>
    <grosspay>6967.94</grosspay>
   </employee>
  </delta>
 </diff>
 <diff>
  <description>
   <change attr="name" type="deleted"/>
  </description>
  <delta count="2" annualsalary="-75837" grosspay="-69661.02">
   <employee name="Adams,Nicholas B">
    <jobtitle>POLICE OFFICER</jobtitle>
    <agencyid>A99416</agencyid>
    <agency>Police Department</agency>
    <hiredate>2010-05-20T00:00:00</hiredate>
    <annualsalary>42391</annualsalary>
    <grosspay>37879.36</grosspay>
   </employee>
   <employee name="Alexander,Obray S">
    <jobtitle>SOLID WASTE WORKER</jobtitle>
    <agencyid>B70410</agencyid>
    <agency>DPW-Solid Waste</agency>
    <hiredate>1970-08-03T00:00:00</hiredate>
    <annualsalary>33446</annualsalary>
    <grosspay>31781.66</grosspay>
   </employee>
  </delta>
 </diff>
</diff-set>
```

APÊNDICE E – QUESTIONÁRIO DE CARACTERIZAÇÃO

Este formulário contém algumas perguntas sobre sua experiência acadêmica e profissional.

1. Formação acadêmica

( ) Doutorado concluído

( ) Doutorado em andamento

( ) Mestrado concluído

( ) Mestrado em andamento

( ) Graduação concluída

( ) Graduação em andamento

Ano de ingresso: _____ Ano de conclusão (ou previsão de conclusão): _____

2. Formação Geral

2.1. Quantos anos de experiência você tem em cada tipo de projeto de software?

Tipo de projeto de software	Anos
Pessoal	
Acadêmico	
Open-source	
Indústria	

2.2. Por favor, indique o grau de sua experiência nas áreas a seguir, com base na escala abaixo:

0 = Nenhum

1 = Estudei em aula ou em livro

2 = Pratiquei em projetos em sala de aula

3 = Utilizei em projetos pessoais

4 = Utilizei em projetos na indústria

Área de Conhecimento	Grau de Experiência				
Banco de Dados	0	1	2	3	4
XML (*Extensible Markup Language*)	0	1	2	3	4
Diff de arquivos	0	1	2	3	4
Controle de Versão (GIT, Subversion, etc)	0	1	2	3	4

2.3. Se seu perfil se enquadrar no perfil desejado para o experimento, você aceitaria participar? O experimento tem previsão de duração de 2:30 horas.

( ) Sim

( ) Não

2.4. Caso tenha respondido SIM à questão anterior, por favor, indique quais as suas disponibilidades de horário para participar do experimento:

Disponibilidade
( ) Segunda-feira - noite
( ) Terça-feira - tarde
( ) Quarta-feira – manhã
( ) Quinta-feira - noite
( ) Sexta-feira – tarde
( ) Sábado – manhã

APÊNDICE F – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Condutora do Estudo: Alessandreia Marta de Oliveira (aluna de doutorado)
Pesquisadores Responsáveis: Profa Vanessa Braganholo e Prof. Leonardo Murta
Instituição: Universidade Federal Fluminense (UFF)

Eventualmente realizamos estudos experimentais para caracterizar/avaliar uma determinada abordagem. Estes estudos são conduzidos por alunos de Pós-graduação em Computação da Universidade Federal Fluminense (UFF). Você foi previamente selecionado pelo seu perfil/conhecimento/experiência e está sendo convidado a participar desta pesquisa. Essa pesquisa consiste em avaliar os resultados obtidos por duas abordagens de *diff* de documentos XML.

1) Procedimentos

O estudo será realizado com data e hora marcada com os participantes pré-selecionados. O estudo será executado de forma individual e consiste na análise dos resultados obtidos por duas abordagens de *diff* de documentos XML. Ao final do estudo será solicitado que você responda um questionário de avaliação sobre as abordagens que estão sendo caracterizadas/avaliadas.

2) Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato.

3) Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização do uso dos dados coletados nesse estudo.

4) Confidencialidade da Pesquisa

Seu nome não será identificado de modo algum. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

5) Participação

Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Caso você decida retirar-se do estudo, favor notificar o pesquisador responsável. Você pode solicitar esclarecimento sobre o estudo a qualquer momento.

6) Declaração de Consentimento

Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmo também que recebi uma cópia deste Termo (TCLE), e compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.

Data:

Nome do Participante (letra de forma):

RG do Participante:

Assinatura:

APÊNDICE G – *DIFF* DE DOCUMENTOS XML – ETAPA 1

Condutora do Estudo: Alessandreia Marta de Oliveira (aluna de doutorado)
Pesquisadores Responsáveis: Profa Vanessa Braganholo e Prof. Leonardo Murta
Instituição: Universidade Federal Fluminense (UFF)
Participante: _____

Tarefa 1

QUAIS funcionários foram demitidos?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Tarefa 2

<u>QUAL</u> o impacto financeiro das contratações e demissões com base no salário bruto (*grosspay*)?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Tarefa 3

<u>QUEM</u> foi promovido, ou seja, teve aumento de salário anual (*annualsalary*) e mudou de cargo (*jobtitle*)?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Obrigada!

APÊNDICE H – *DIFF* DE DOCUMENTOS XML – ETAPA 2

Condutora do Estudo: Alessandreia Marta de Oliveira (aluna de doutorado)
Pesquisadores Responsáveis: Profa Vanessa Braganholo e Prof. Leonardo Murta
Instituição: Universidade Federal Fluminense (UFF)
Participante: _____

Tarefa 4

QUANTOS funcionários foram transferidos, ou seja, mudaram de agência *(agencyid, agency)*?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Tarefa 5

QUAL o funcionário teve o maior aumento de salário bruto (*grosspay*)?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Tarefa 6

QUAL o impacto financeiro dos aumentos do salário anual (*annualsalary*)?

Horário de início	_____ : _____ : _____
Resposta	
Horário de fim	_____ : _____ : _____
Explique como chegou a esta conclusão	
Grau de dificuldade na realização da tarefa (1 - fácil; 5 - difícil)	
Comentários adicionais	

Obrigada!

APÊNDICE I - QUESTIONÁRIO DE ENCERRAMENTO

Condutora do Estudo: Alessandreia Marta de Oliveira (aluna de doutorado)
Pesquisadores Responsáveis: Profa Vanessa Braganholo e Prof. Leonardo Murta
Instituição: Universidade Federal Fluminense (UFF)
Participante:
_____

Realização das Tarefas

1. Na sua opinião, quais são os pontos positivos e negativos da abordagem 1?	
Positivos	**Negativos**

2. Na sua opinião, quais são os pontos positivos e negativos da abordagem 2?	
Positivos	**Negativos**

3. Na sua opinião, qual das abordagens apresenta um resultado mais intuitivo? Por que?

4. Comentários adicionais

Obrigada!