Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

# Eficiência e eficácia na identificação de elementos correspondentes em documentos XML

**Carlos Roberto Carvalho Oliveira**

JUIZ DE FORA
NOVEMBRO, 2018

# Eficiência e eficácia na identificação de elementos correspondentes em documentos XML

Carlos Roberto Carvalho Oliveira

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Alessandreia Marta de Oliveira

JUIZ DE FORA

NOVEMBRO, 2018

# Eficiência e eficácia na identificação de elementos correspondentes em documentos XML

Carlos Roberto Carvalho Oliveira

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alessandreia Marta de Oliveira
Doutora em Computação (UFF)

Jairo Francisco de Souza
Doutor em Informática (PUC - RJ)

Victor Stroele de Andrade Menezes
Doutor em Engenharia de Sistemas e Computação (UFRJ)

JUIZ DE FORA
13 DE NOVEMBRO, 2018

*Aos meus amigos e irmãos.*

*Aos pais, pelo apoio e sustento.*

# Resumo

Devido ao grande poder de representação e troca de informações, os documentos XML são cada vez mais utilizados. A medida que estes documentos vão sofrendo alterações, torna-se cada vez mais complexo controlar as mudanças, especialmente em documentos muito grandes. Existem algumas abordagens que visam identificar as mudanças presentes em diferentes versões destes documentos, porém, a grande maioria possui seu foco somente na mudança sintática, não levando em conta a semântica associada a tal modificação. Tais diferenças recebem o nome de *diff*. Diante disto, este trabalho apresenta uma caracterização dessas abordagens de *diff* de documentos XML, tomando como base o casamento de elementos correspondentes. Essa caracterização permite avaliar a eficiência (tempo) e a eficácia (corretude). O trabalho apresenta também o XMeasure, desenvolvido para realizar a automatização dessa caracterização, obtendo os resultados das medidas utilizadas nesses campos de estudos, apoiando em métricas utilizadas em Estatística e Recuperação de Informação. Os resultados mostraram que o XChange é mais eficiente e, no critério eficácia, seus resultados são tão bons quanto os resultados comparados.

**Palavras-chave:** Evolução de documentos XML, casamentos de elementos correspondentes.

# Abstract

Due to the great power of representation and exchange of information, XML documents are increasingly used. As these documents change, it becomes increasingly complex to control the changes, especially in very large documents.There are some approaches that aim to identify the changes present in different versions of these documents, however, the great majority has its focus only on the syntactic change, not taking into account the semantics associated to such change. Such differences are called the diff. Therefore, this work presents a characterization of these diff approaches of XML documents, based on the matching of corresponding elements. This characterization allows to evaluate the efficiency (time) and the effectiveness (corretude). The work also presents XMeasure, developed to perform the automation of this characterization, obtaining the results of the measures used in these fields of studies, supporting in metrics used in Statistics and Information Retrieval. The results showed that XChange is more efficient and, in the efficiency criterion, its results are as good as the results compared.

**Keywords:** Evolution of XML documents, matching of corresponding elements.

# Agradecimentos

Agradeço a Deus pelas inúmeras oportunidades e o sucesso no curso. Aos meus parentes e amigos, pelo incentivo e presença.

Aos meus pais, José e Maria, por todo amor, incentivo e confiança nas escolhas feitas. À minha irmã Lidiane, pela cumplicidade em todos os momentos.

À professora Alessandreia pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Agradeço aos amigos do Grupo de Educação Tutorial da Computação (GET-Comp - UFJF), em especial ao Matheus Marques e ao João Victor pelo apoio durante o desenvolvimento deste trabalho.

Aos meus amigos que sempre estiveram comigo, diretamente ou indiretamente, principalmente ao Marcus Antônio e Tamires Mariane que desde o começo apoiaram e estiveram em todas as situações. Ao Júnior Ribeiro, que foi um grande amigo e sempre esteve me apoiando e orando para que pudesse realizar mais esse sonho.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

*"Ebenézer! Até aqui o Senhor nos ajudou".*

*1Samuel 7:12*

# Conteúdo

# Lista de Figuras

# Lista de Tabelas

# 1 Introdução

Os documentos XML (*eXtensible Markup Language*) estão cada vez mais presentes em nosso cotidiano, sendo utilizados em amplas aplicações de transmissão e processamento de dados, mostrando-se eficiente na descrição e definição dos dados (KHAN, 2016).

Tais documentos são constituídos de alguns elementos que variam de acordo com a necessidade do cenário tratado. A flexibilidade e a portabilidade são características que vêm fazendo com que, nos últimos anos, documentos XML sejam utilizados como um padrão para representação, intercâmbio e manipulação de dados em aplicações nas mais diversas áreas. Por exemplo, aplicativos de escritório como o Microsoft Office armazenam seus dados em uma série de documentos XML compactados; ferramentas CASE armazenam seus modelos UML[1] em um documento XML seguindo o esquema XMI[2]; IDEs armazenam seus metadados e scripts como documentos XML; experimentos científicos representam seu fluxo de trabalho em XML (OLIVEIRA et al., 2018).

## 1.1 Motivação

Devido a essa flexibilidade, tais documentos evoluem ao longo do tempo, seja por mudanças no conteúdo textual (valores dos campos sofrem mudanças) ou estrutural (seja devido a maior organização dos dados ou inserção de novos campos). Como exemplo, suponha o sistema que realiza o gerenciamento dos funcionários da Universidade da Califórnia em Berkeley[3]. Nesse contexto, os dados sofrem alterações ao decorrer do tempo, em consequência de admissões, promoções, aumento de salários, demissões, etc. A base é composta pelo nome do funcionário (*employee_name*), cargo (*title*), salário bruto (*grosspay*), salário regular (*regularpay*), salário de horas extras (*overtimepay*) e outros pagamentos (*otherpay*). A Figura 1.1 apresenta três funcionários da primeira versão disponível e as informações relacionadas a eles.

[1]UML - Unified Modeling Language (http://www.omg.org/spec/UML)
[2]XMI - XML Metadata Interchange (http://www.omg.org/spec/XMI)
[3]https://ucannualwage.ucop.edu/wage/

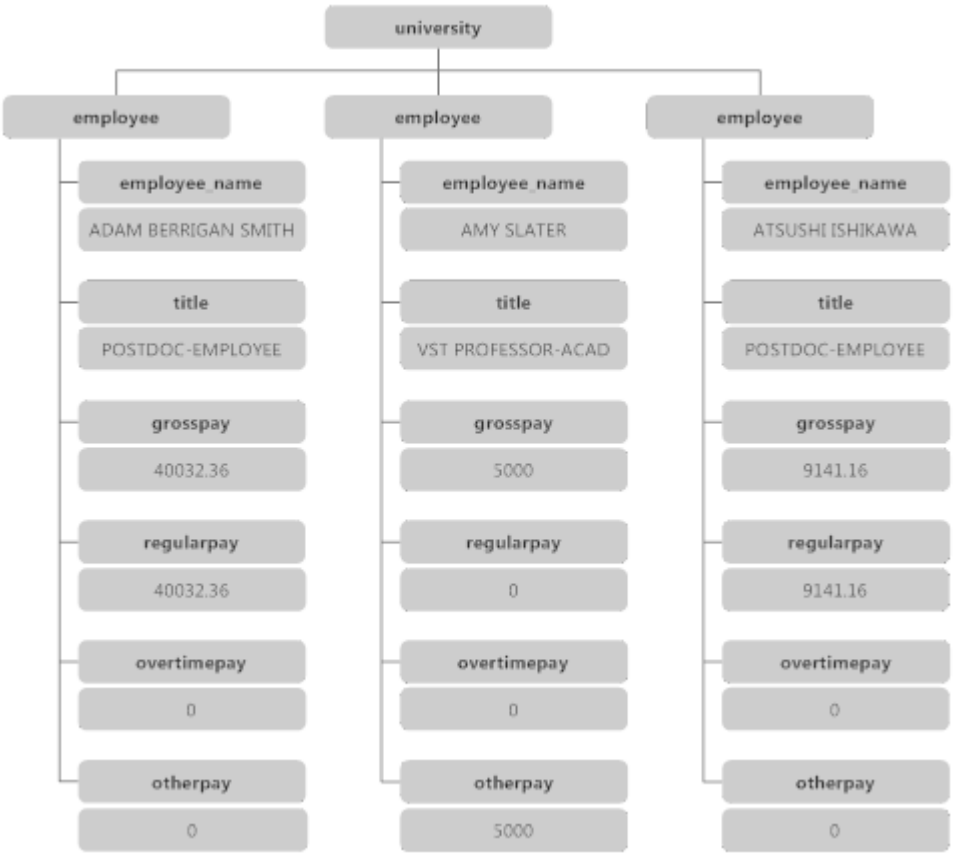


Figura 1.1: Detecção de diferenças em um documento XML - versão 1

A Figura 1.2 apresenta 4 funcionários de uma segunda versão disponível e as informações relacionadas. Como pode ser observado, existem 2 funcionários que estão presentes na primeira e na segunda versão, destacados em laranja e sinalizados com o símbolo de "%" em suas *tags*. Pode-se observar que o funcionário Amy Slater, teve uma alteração no cargo em que o mesmo ocupava dentro da instituição (destacado de laranja e sinalizado com o símbolo de "%" na *tag title*). Pode-se observar ainda que a funcionária Ashley Nicole W., destacada em verde e sinalizada com o símbolo de "+" em suas *tags*, está presente somente na segunda versão, bem como o funcionário Atsushi Ishikawa, destacado em vermelho e sinalizado com o símbolo de "-" em suas *tags* está presente somente na primeira versão. Os valores na cor cinza e que não possuem nenhum símbolo são os dados que não sofreram alterações.

Tais mudanças textuais são facilmente identificadas após uma análise das duas versões deste pequeno fragmento de documento XML. No entanto, gerenciar manualmente essas mudanças ocorridas de uma versão para outra é um processo custoso, principalmente

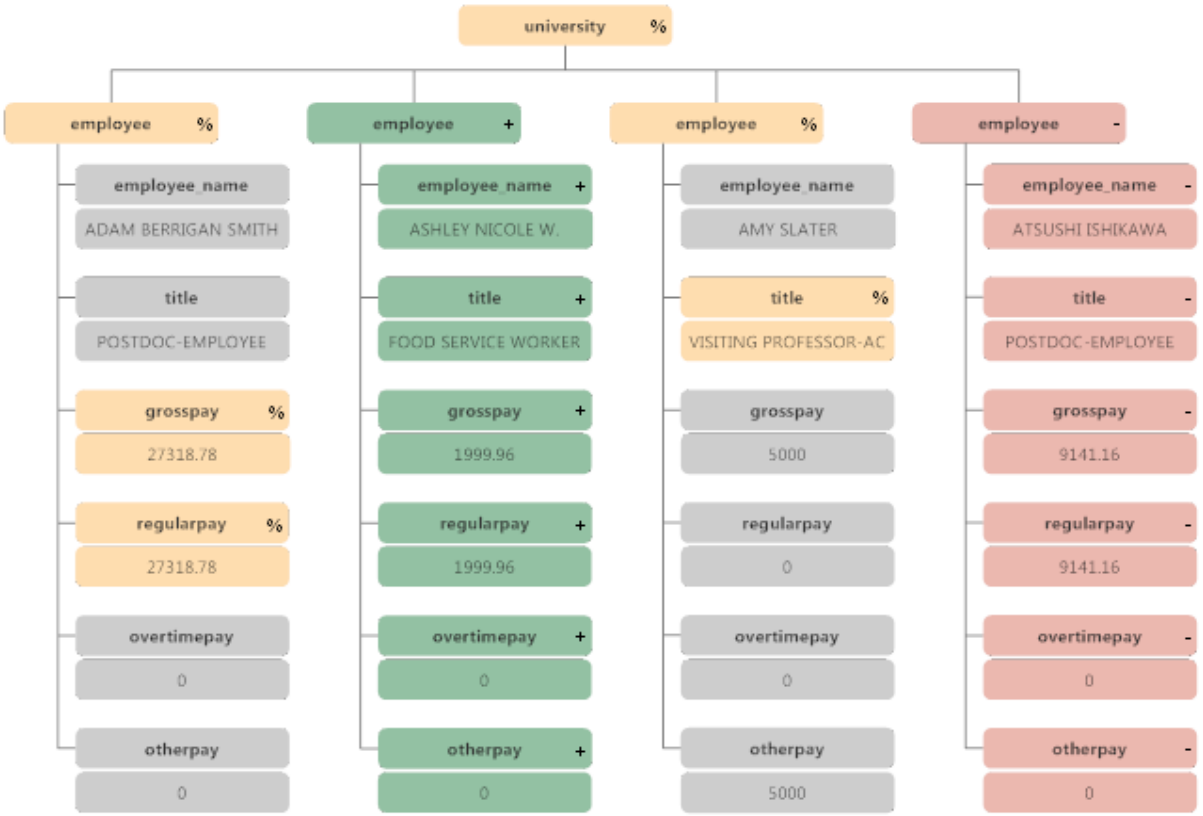


Figura 1.2: Detecção de diferenças em um documento XML - versão 2

em documentos XML maiores, além de estar propício a erros.

Em consequência deste fato, torna-se cada vez mais necessário realizar o controle de mudanças, através de ferramentas específicas de detecção de diferenças entre documentos XML, tais como X-Diff (WANG; DEWITT; CAI, 2003), XyDiff (COBÉNA; ABITEBOUL; MARIAN, 2002) e XChange (OLIVEIRA, 2016). Porém, devido a esse grande poder de representação, os documentos XML se encontram em formas distintas, seja no conteúdo textual ou na estrutura, uma vez que isso vária de acordo com o cenário.

## 1.2 Objetivos

Diante disso, o objetivo desse trabalho é estender a proposta de Oliveira (2016), no que diz respeito a utilização de mais base de dados reais, para compor a avaliação experimental e avaliar a eficiência e eficácia de algumas abordagens de detecção de diferenças (*diff*) de documentos XML e casamento de elementos correspondentes, XChange (OLI-

VEIRA; MURTA; BRAGANHOLO, 2014) e X-Diff (WANG; DEWITT; CAI, 2003), em um conjunto de documentos XML.

## 1.3 Metodologia

Existem diversos trabalhos na literatura que visam gerenciar as mudanças em documentos XML, sendo que o foco é o *diff* sintático, porém existem casos em que a semântica por trás das mudanças também é importante, ou seja, inferir a razão das modificações. Após identificar as abordagens pioneiras no assunto, juntamente com o XChange, que é a abordagem proposta para a comparação com os trabalhos já existentes, tornou-se necessário obter qual abordagem apresentava melhores resultados em termos de eficiência e eficácia.

De tal necessidade concebeu-se o XMeasure (OLIVEIRA, 2014), uma ferramenta para apoiar tal comparativo através de métricas utilizando variáveis capazes de calcular a corretude e a duração das tarefas relacionadas ao estudo.

A fim de validar o estudo, um levantamento de bases reais foi necessária para que pudessem ser utilizadas na avaliação experimental com as abordagens, com o propósito de não favorecer ou desmerecer uma determinada abordagem. Foram realizadas avaliações experimentais com intuito de avaliar a eficiência e eficácia no que diz respeito ao casamento de elementos correspondentes entre versões de um documento XML.

Foram utilizadas duas bases de dados, que variam a quantidade de informações, quantidade de *tags* por elementos entre outros pontos, a fim de verificar os pontos apresentados no trabalho citado.

Os resultados obtidos nas análises serão apresentados em termos de algumas medidas que são utilizadas em estatística e em recuperação de informação, tais como: verdadeiros positivos, falsos positivos, falsos negativos, precisão, cobertura e *F-Measure* (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 correção e integridade. No contexto deste trabalho, a precisão indica quão correto estão os casamentos identificados pela abordagem. Por outro lado, a cobertura indica o quão completa a relação de casamentos correspon-

dentes encontradas pela abordagem está, quando comparada aos resultados esperados. Finalmente, a eficiência é apresentada em termos do número de casamentos corretos por segundo.

De modo a automatizar essa caracterização, a ferramenta XMeasure (OLIVEIRA, 2014) vem sendo adaptada, de forma a obter os resultados para as medidas citadas anteriormente. Tal ferramenta também conta com o recurso de exportar esses dados para o Excel, fazendo com que os dados sejam melhor visualizados através de planilhas e gráficos.

Tais resultados nos proporcionarão a possibilidade de mediar a eficiência e eficácia das abordagens para os cenários tratados na avaliação experimental.

## 1.4 Questões de Pesquisa

As questões de pesquisas levantadas neste trabalho estão descritas a seguir:

1. Qual abordagem é mais eficaz no que diz respeito ao casamento de elementos correspondentes entre as versões do documento XML?

2. Qual abordagem é mais eficiente no que diz respeito ao casamento de elementos correspondentes entre as versões do documento XML?

## 1.5 Organização

Este documento está organizado em outros cinco capítulos, além desta introdução. O Capítulo 2 apresenta alguns conceitos relacionados a documentos XML e alguns fundamentos de *diff*. O Capítulo 3 fornece uma revisão da literatura relacionada a *diff* de documentos XML. O Capítulo 4 descreve o pré-processamento necessário para realizar a avaliação experimental, além de introduzir a ferramenta desenvolvida para apoiar e fornecer os resultados, o XMeasure. O Capítulo 5 apresenta a descrição das bases utilizadas no estudo e a avaliação experimental relacionada ao casamento de elementos correspondentes, onde o XChange e o X-Diff foram comparados quanto à eficiência e à eficácia dos métodos utilizados. Finalmente, o Capítulo 6 discute as conclusões para este trabalho e as sugestões de trabalhos futuros.

# 2 Fundamentação Teórica

Gerenciar mudanças é fundamental para gerir o histórico de um projeto, seja para auxiliar no desenvolvimento colaborativo, bem como para desfazer, analisar ou recuperar versões estáveis. Para tal fim, existem inúmeras ferramentas onde muitas possuem licença *open source* (código aberto). Porém, no contexto do gerenciamento de mudanças em documentos XML, as principais abordagens disponíveis ainda apresentam muitas limitações (MURTA, 2006).

De forma a compreender as mudanças realizadas ao longo do processo de evolução de documentos XML e o funcionamento das abordagens, torna-se necessário introduzir conceitos essenciais que serão utilizados durante todo o texto.

## 2.1 Documentos XML

XML (*eXtensible Markup Language*) é um padrão especificado pelo órgão responsável pela padronização de iniciativas ligadas à *Web*, o W3C(*World Wide Web Consortium*). É um preceito para a marcação de dados na *Web*, com foco na descrição do conteúdo, visando criar documentos organizados de forma hierárquica (BRAY et al., 2018).

Tomando como base Moro e Braganholo (2009), a seguir são apresentados alguns conceitos relacionados a documentos XML. Um documento XML é constituído de alguns componentes que variam de acordo com a necessidade do cenário tratado, diferentemente do HTML (*HyperText Markup Language*) que possui *tags* predefinidas para serem utilizadas. Os elementos de um documento XML são definidos pelo usuário do domínio e descrevem o conteúdo e a estrutura do mesmo, sendo representados por marcas chamadas de *tags*, demilitadas por uma marca inicial e por uma marca final, onde tudo o que estiver entre essas marcações diz respeito ao conteúdo do elemento em questão. Há também a possibilidade de usar atributos e estes ficam localizados dentro da primeira marcação de um elemento.

De forma a visualizar graficamente tais componentes, a Figura 2.1 apresenta um

pequeno fragmento de um documento XML relacionado a informações de funcionários em uma empresa. Como se pode observar, a primeira linha de um documento XML se caracteriza por uma instrução padrão, chamada instrução de processamento, trazendo informações de forma explícita em um documento destinadas a alguma aplicação. Sintaticamente uma instrução de processamento inicia-se com <? e termina com ? >. Pode ser composta por vários parâmetros que ajudam na caracterização do documento analisado, por exemplo, o parâmetro "*version*" que indica a versão da linguagem (parâmetro obrigatório) e outros dois parâmetros opcionais que podem estar presentes descritos a seguir. O tipo de codificação de caracteres utilizada no documento, dada pelo parâmetro "*encoding*", e o "*standalone*", que informa se o documento utiliza-se de declarações externas ou não. Caso a opção "*yes*" (*default*) esteja presente, significa que o documento não faz uso de nenhuma declaração externa, já o caso oposto, "*no*", indica que um conjunto de declarações definidas externamente afetam a interpretação do conteúdo do documento, por exemplo, a utilização de Definição de Tipo de Documento (*Document Type Definition* - DTD) externa (BRAY et al., 2018).

```
1  <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2  <empresa>
3      <funcionario>
4          <nome>Beltrano da Silva</nome>
5          <cargo>Gerente de Vendas</cargo>
6          <salario>3560</salario>
7          <data_contratacao>20-06-2003</data_contratacao>
8      </funcionario>
9      <funcionario>
10         <nome>Ciclano Costa</nome>
11         <cargo>Analista de Projetos</cargo>
12         <salario>2920</salario>
13         <data_contratacao>01-12-2007</data_contratacao>
14     </funcionario>
15     ...
16 </empresa>
```

Figura 2.1: Componentes de um documento XML

Dando continuidade, o documento XML também é formado por um elemento principal, no exemplo "empresa". Tal elemento é importante para manter a organização e o entendimento desse documento. É um tipo de elemento composto por conter outros sub-elementos.

Outro item é o sub-elemento, no exemplo da Figura 2.1 representadas pelas *tags*

funcionário, que são filhos da *tag* principal empresa. O sub-elemento também se caracteriza como um elemento composto.

As *tags* nome, cargo, salário e data_contratacao que são características do sub-elemento funcionário, são definidas como tipos de elementos textuais, por receberem somente textos.

Além desses tipos de elementos mencionados acima, existem também os elementos classificados como misto, isto é, aqueles que além de possuírem texto também são constituídos de sub-elementos; e o tipo de elemento vazio, caracterizado pela ausência de conteúdo, podendo ser definido de duas formas, <exemplo></exemplo> ou <exemplo/>.

Uma característica de um documento XML é que o mesmo pode ser representado graficamente através de uma estrutura de árvores, sendo seus elementos, atributos e textos, os nós e as relações pai/filho, as arestas. Para tal, é necessário que o documento XML respeite algumas regras, podendo dizer assim que o mesmo é bem formado. Tais regras são descritas a seguir:

- deve existir uma única raiz;
- todas as *tags* devem ser fechadas;
- não existe interseção de elementos, isto é, os mesmo são bem alinhados, respeitando a ordem, sendo fechados na ordem inversa das que foram abertos, ou seja, a *tag* do elemento pai é fechada depois que todas as *tags* de seus filhos forem fechadas;
- atributos não se repetem no mesmo elemento, ou seja, não é permitido inserir mais de um atributo no mesmo elemento que possua o mesmo nome;
- maiúsculas e minúsculas são relevantes nas definições dos nomes dos elementos (*Case-sensitive*, que significa que algo faz distinção entre letras maiúsculas e minusculas)

O fragmento apresentado na Figura 2.1 se enquadra nas regras descritas anteriormente. Portanto o mesmo pode ser visualizado na Figura 2.2, onde os tipos de elementos compostos estão destacados na cor azul, e em suas folhas, estão os tipos de elementos textuais, simbolizados na cor cinza.

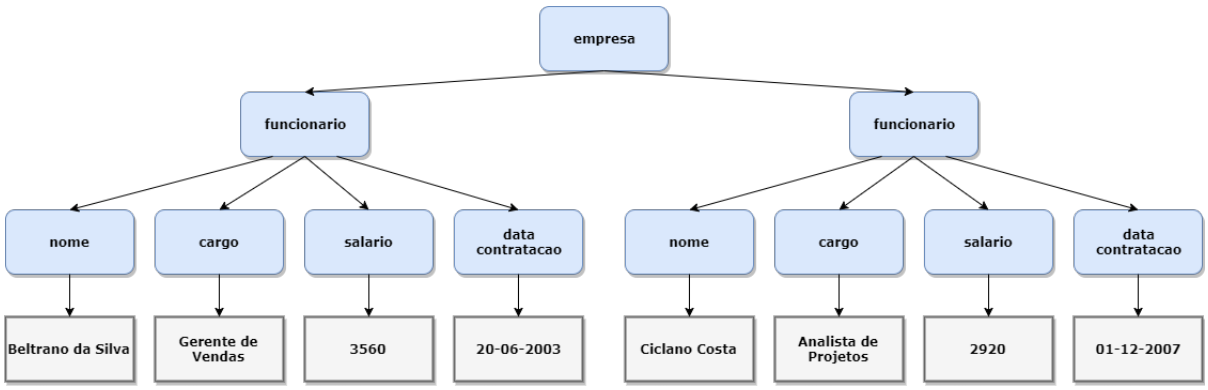


Figura 2.2: Fragmento do documento XML no formato de árvore

## 2.2 Controle de Versão

Ao longo do tempo, durante o ciclo de vida de um software, o mesmo recebe constantes modificações, sejam elas por alterações providas de erros, questões de segurança, novas necessidades dos usuários, mudança das regras de negócio, refatoração de código, entre outros. O mesmo acontece com os documentos XML, ao decorrer do tempo, eles sofrem alterações, sejam elas de estruturas, elementos e valores, tudo depende de como o cenário se modifica.

Pensando no cenário de uma empresa, pode-se identificar várias alterações possíveis, por exemplo, pessoas são admitidas; demitidas; sofrem alterações de salário anualmente baseado em alguma porcentagem; são transferidas para outra localidade, trocando assim seu endereço, salário, etc.; recebem promoção, alterando sua função, cargo, salário; dentre outras possibilidades.

Um grande desafio está em gerenciar as mudanças sofridas à medida que estes documentos vão evoluindo em sua utilização, uma vez que possuem características diferentes de um arquivo redigido em algum editor de texto, por exemplo. Muitos métodos se baseiam em verificar as alterações entre uma versão e outra através da comparação linha a linha, sendo tratados como um arquivo de texto puro, negligenciando todo o conhecimento acerca do formato dos dados (MURTA, 2006), porém para documentos XML, isso não é interessante dado que através de sua estrutura hierárquica pode-se obter resultados mais expressivos e com valor semântico diante das informações obtidas por esse gerenciamento de mudanças.

As informações descobertas são uma preciosa fonte de conteúdo para gerar um histórico de um documento, seja para analisá-lo, refazê-lo, recuperar versões estáveis, etc. Entretanto, em documentos XML são necessários métodos especializados em extrair conhecimento, o que não ocorre nos sistemas que utilizam métodos firmados em pesquisa textual (GARCIA, 2012).

Recebe o nome de *diff* (LEON, 2000), as diferenças existentes entre versões de um documento XML, isto é, aquilo que foi modificado de uma versão para outra. A partir da detecção dessas diferenças, um documento *delta* ou *edit script* é gerado, representando todas as diferenças entre as versões compostas pelo conjunto de operações necessárias para transformar uma versão na outra. Através de uma versão de um determinado documento e um *delta* é possível de forma eficiente e eficaz chegar em uma outra versão.

Para o caso de identificar elementos correspondentes, é necessário que a representação consiga identificar unicamente os elementos do documento, bem como distinguir quando eles são movidos de um ponto para outro nas versões, permitindo assim acompanhar todo o tempo de vida do elemento ao longo de diversas alterações (OLIVEIRA, 2016).

Segundo Lindholm (2001) as operações que compõe o *diff* podem ser classificadas em cinco tipos:

- inserção: consiste na inserção de um nó em alguma parte do documento. Caracteriza-se uma operação de inserção quando um determinado nó está presente na versão mais recente e não presente na versão anterior;
- exclusão: um nó foi deletado do documento. Uma operação é classificada como deleção quando um nó existe somente na versão anterior, não possuindo nenhum correspondente na versão mais atualizada;
- atualização: significa que um nó teve seu conteúdo alterado. Caracteriza-se por um nó que teve seu conteúdo modificado no seu elemento correspondente na outra versão, existindo assim uma diferença de seus conteúdos;
- movimentação: um nó foi movido para outra parte do documento. É classificada como movimentação, quando a posição do nó na versão anterior é diferente da

posição de seu correspondente na versão mais atualizada;

- cópia: um nó foi copiado para outra parte do documento. Uma operação é classificada como cópia quando um nó da versão anterior possui dois ou mais correspondentes na versão posterior.

## 2.3 Técnicas de Casamentos de Elementos Correspondentes

Como mencionado anteriormente, identificar e casar (*match*) os elementos correspondentes nas duas versões não é uma tarefa trivial, devido a maior parte dos documentos não possuírem previamente um ID definido na DTD, o qual permitiria seu casamento. Já existem na literatura várias abordagens para realizar o *diff* de documentos XML, sendo assim propostas várias formas de identificação desses elementos.

Tais abordagens normalmente buscam obter proveito do ID definido da DTD do documento (OLIVEIRA, 2016). Porém, quando isso não é possível, alguns deles possuem técnicas heurísticas para determinar chaves que identificam unicamente cada elemento no documento. A correspondência de seus elementos se dá com a utilização de árvores e são encontrados através de seus valores identificadores, definidos na transformação desses documentos para a estrutura de árvore, recebendo cada elemento um identificador de acordo com a heurística empregada.

Outro método presente na literatura é intitulado como técnica de similaridade (OLIVEIRA, 2016). Muita das vezes não é possível definir um atributo chave que identifique unicamente um determinado elemento em todo o documento. Além disso, ao fixar um elemento chave para o documento, o mesmo pode apresentar falhas no casamento dependendo da geração da versão, por exemplo, no contexto da base de funcionários, se na hora da geração de uma versão o atributo identificador sair com algum carácter errado, devido a um erro de digitação ou outra anomalia que possa ocorrer. Desta forma, técnicas baseadas em chave de contexto não identificariam tal correspondência.

Em meio a isso, os documentos XML são comparados através de algoritmos de similaridade que levam em consideração várias métricas que avaliam o quão semelhante

os elementos são, e posteriormente realizam o casamento dos elementos equivalentes.

## 2.4 Considerações Finais

Este capítulo apresentou alguns conceitos e características dos documentos XML, passando por sua formação, nomenclatura e definições de seus componentes. Apresentou também as regras que devem ser obedecidas para que esses documentos XML sejam representados em árvores. O capítulo abordou também o conceito de *diff* e suas operações, além de descrever as técnicas para realização de casamentos em elementos correspondentes. Compreender os conceitos de documentos XML, *diff*, e técnicas de casamento são fundamentais para compreensão da avaliação experimental, fornecendo assim uma base teórica para o entendimento da abordagem proposta.

# 3 Algoritmos de *Diff* e Casamentos de Elemento Correspondentes

Existem diversas abordagens de *diff* de documentos XML, tais como o X-Diff (WANG; DEWITT; CAI, 2003), XyDiff (COBÉNA; ABITEBOUL; MARIAN, 2002), XKeyDiff (SANTOS; HARA, 2004), BioDIFF (SONG et al., 2007), DeltaXML[4], XKeyMatch (SANTOS, 2006), XRel (SUNDARAM; MADRIA, 2012) e XChange (OLIVEIRA, 2016).

Para compor esse capítulo foram selecionados alguns trabalhos encontrados na literatura, em especial o X-Diff e o XyDiff por serem pioneiros nesta linha, possuírem mais menções e que serviram de base para outras abordagens, e o XChange que é o alvo deste trabalho, para validação das hipóteses na comparação de casamento de elementos correspondentes, os quais iremos descrever suas principais funcionalidades. Também é apresentada uma comparação entre essas abordagens expondo suas principais características. O capítulo é encerrado com as considerações finais.

## 3.1 X-Diff

O X-Diff (WANG; DEWITT; CAI, 2003) é uma abordagem eficiente para calcular a diferença ideal entre duas versões de um documento XML. Esta abordagem considera características do domínio desses documentos e introduz vários conceitos de chave, como a assinatura do nó, e XHash. A abordagem é uma combinação destas técnicas com o padrão de correção árvore a árvore, que consiste numa abordagem de identificação de alterações entre documentos no formato de árvore.

Sua principal característica é o uso de árvores não ordenadas, que somente avalia o relacionamento pai-filho, não levando em conta a ordenação entre os irmãos (COBÉNA; ABDESSALEM; HINNACH, 2004). Os autores defendem que, por ter um resultado mais preciso das alterações, este modelo se adequa melhor à maioria das aplicações de banco de

[4]https://www.deltaxml.com/

dados. No entanto, por usar árvores não ordenadas, o algoritmo não é capaz de detectar operações de movimentação, suportando apenas as operações básicas de inserção, remoção e alteração.

O X-Diff é uma abordagem específica para documentos XML, e encontra o mapeamento mínimo entre os filhos de duas subárvores reduzindo-se a um problema de fluxo máximo com custo mínimo (PETERS, 2005). O X-Diff encontra um *delta* mínimo com complexidade de tempo e memória quadráticos. Os autores propõem algumas heurísticas que melhoram o desempenho, mas com isso, não garante o resultado mínimo. Esta abordagem suporta apenas comparação entre duas versões.

O fluxo básico do X-Diff pode ser visualizado a partir da Figura 3.1 e está dividido nas etapas descritas a seguir.

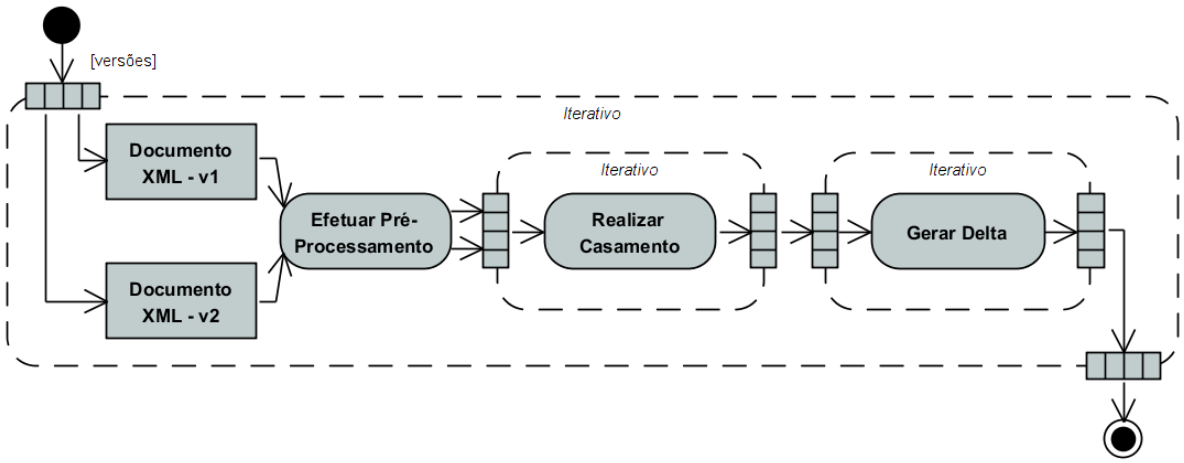


Figura 3.1: Fluxo X-Diff (OLIVEIRA, 2016)

A primeira etapa consiste em um pré-processamento. Nesta etapa ocorre a transformação da entrada. Dados os documentos XML de ingresso, o algoritmo os converte em estruturas de árvores, através do *parsing*, i.e., leitura e análise dos documentos, que são denominadas XTrees.

O próximo passo é realizar a computação das assinaturas. Ao realizar o *parsing*, é utilizada uma função especial de *hash* para calcular essas assinaturas em ambas as árvores, identificando cada nó do documento XML, e é intitulada como XHash. Tal assinatura representa toda a sub-árvore enraizada no nó, e desconsidera a ordem entre irmãos.

O objetivo da segunda etapa é criar um mapeamento de custo mínimo entre

os documentos XML representados pelas árvores. Primeiro o algoritmo elimina as sub-árvores equivalentes entre os dois nós raiz, comparando os valores XHash de nós filhos de segundo nível. Sub-árvores com valores XHash idênticos podem ser consideradas como equivalentes. Este passo reduz de forma eficaz o tamanho da árvore, evitando cálculos desnecessários nas fases subsequentes do algoritmo.

Em seguida é realizada a comparação das distâncias para cada folha dos nós da outra árvore XML. Os valores dessas comparações ficam armazenados em uma tabela de distância que posteriormente é utilizada para avaliar os melhores casamentos.

Ao terminar o processo de comparação, executa-se o mesmo procedimento com os nós pais. Em cada ascensão de nível na árvore, é feita a comparação dos valores *hash* a fim de diminuir o espaço de busca. Após realizar estes procedimentos, a avaliação dos resultados armazenados na tabela começa com os nós raízes, partindo para seus descendentes. O algoritmo realiza uma avaliação *top-down* nas estruturas de árvore, analisando os melhores candidatos coletados e casando-os com os respectivos nós.

No terceiro e último passo, o *delta script* mínimo é gerado a partir do mapeamento de custo mínimo, e contém as operações de inserção, remoção e atualização para realizar a transformação de uma versão na outra. Tal processo se dá de maneira recursiva iniciando da raiz até as folhas.

Os autores também comentam sobre o desempenho da abordagem. Como já mencionado, o algoritmo possui tempo polinomial, porém em alguns casos isso pode não suprir as necessidades dos usuários. A motivação é acelerar o processo do X-Diff sem reduzir significativamente a qualidade dos resultados, ou seja, em alguns casos o usuário pode sacrificar algum grau de precisão em troca de um melhor tempo de resposta. Tal otimização tem como fundamento evitar a comparação entre todos os candidatos possíveis. Para isso é utilizado um limiar para selecionar o melhor candidato entre os coletados.

Tal limiar deve ser escolhido com cuidado, pois se o mesmo for elevado demais consequentemente levara à incompatibilidade de elementos, porém caso ele seja muito baixo pode-se não obter casamentos corretos prejudicando assim a precisão. A solução apresentada pelos autores é fazer uso de amostragem para calcular esse limiar sempre que houver mais de dois elementos com casamentos correspondentes atualizados nos docu-

mentos. Seguindo a linha de pensamento dos mesmos, quando se deseja verificar a diferença entre duas versões de um documento, as duas versões não serão significativamente diferente, portanto, ao calcular a distância de edição dos pares que tiveram elementos correspondentes, um pequeno número de nós são aleatoriamente selecionados do primeiro documento. Para cada nó da amostra é calculada a distância de edição entre este nó e todos os candidatos do outro documento, afim de obter o menor valor (melhor correspondência). Posteriormente, o limiar é definido como a média das distâncias de edição dos elementos tomados para amostra (WANG; DEWITT; CAI, 2003).

## 3.2 XyDiff

O XyDiff (COBÉNA; ABITEBOUL; MARIAN, 2002) é uma abordagem de *diff* de documentos XML, concebido para verificar alterações entre versões de documentos XML em um projeto que investiga data *warehouses* dinâmicos. Devido ao seu contexto, armazenamento de grandes volumes de dados, é uma abordagem que prioriza a eficiência em termos de espaço de memória e velocidade, mesmo à custa de alguma perda de qualidade. Desta forma, esta abordagem não gera um *delta* mínimo, ou seja, um conjunto mínimo de operações através das quais pode-se transformar uma versão na outra (MURTA, 2006). Entretanto, experiências feitas por seus autores, mostram que o resultado do XyDiff é correto, e na maioria das vezes é bastante próximo da solução ótima.

Este algoritmo leva em consideração as operações de inclusão, remoção, alteração e movimentação. Sendo esta última a mais custosa, pois um nó movido pode estar em um contexto totalmente diferente do anterior (pais, irmãos e filhos diferentes), dificultando sua localização. Utiliza recursos próprios do formato XML, como os IDs definidos na DTD e o modelo de árvore ordenada, onde a ordem dos filhos é relevante na detecção de alterações.

O XyDiff possui complexidade no tempo de O(nlog(n)), onde n é o número de nós (PETERS, 2005). Esta ordem logarítmica-linear é atingida somente em raros casos, como quando existem muitas alterações no arquivo (pior caso). Isto significa que o algoritmo, na maioria dos casos, pode ser executado em ordem de grandeza inferior ao descrito. A complexidade de espaço é linear em relação ao tamanho do documento.

Similar ao realizado com o X-Diff, o fluxo básico de execução do XyDiff é exibido na Figura 3.2 e está dividido nas etapas descritas a seguir:

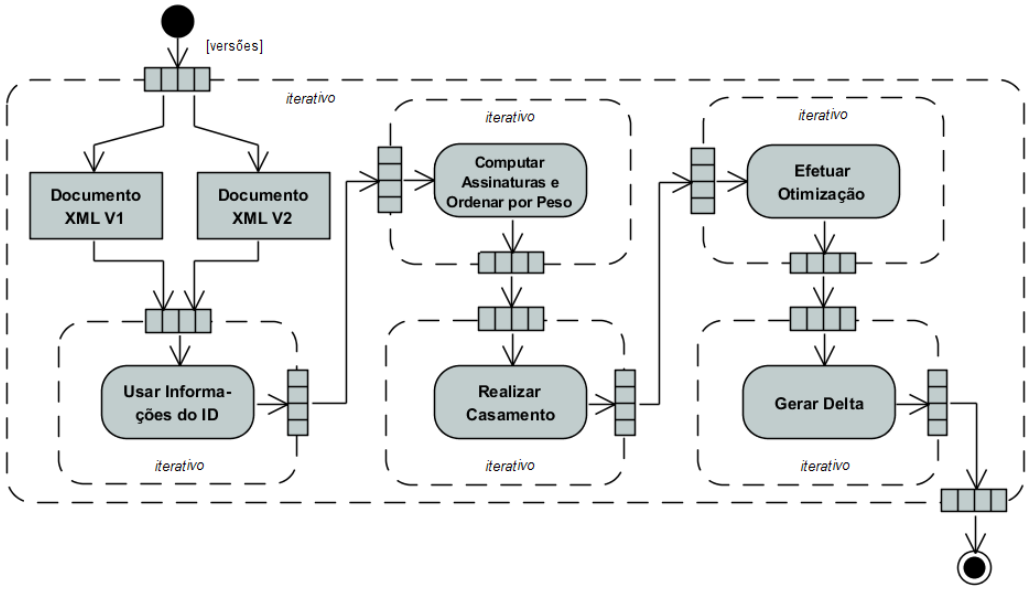


Figura 3.2: Fluxo XyDiff (OLIVEIRA, 2016)

Na DTD alguns nós recebem um ID. Essa etapa consiste na utilização desses atributos ID. A existência de um ID em um determinado nó, nos garante que ele somente realizará casamento correto se também existir esse identificador na outra versão do documento XML. Assim quanto mais nós de ID o documento possuir, mais rápido ele executará, uma vez que o casamento é obtidos de maneira direta sem a necessidade de nenhuma outra técnica.

A segunda etapa compreende na computação das assinaturas e ordenação das subárvores pelo peso. Através de uma função de *hash*, cada nó recebe um valor que representa a assinatura e que é obtido levando em consideração o conteúdo e as assinaturas dos seus filhos. Tal assinatura representa unicamente o conteúdo da subárvore toda, enraizada neste nó. O peso é obtido através do cálculo do tamanho do conteúdo do nó e dos pesos dos nós filhos.

Fazendo uso de uma fila de prioridade, o algoritmo consegue obter qual é a próxima subárvore para qual se deve encontrar a correspondência. Tal fila é construída levando em consideração o peso, e sendo a primeira posição o nó raiz do documento, por possuir maior peso.

O próximo passo consiste em obter as correspondências a partir da fila de prio-

ridade. A primeira árvore do documento modificado da fila de prioridades é removida. O algoritmo tenta obter a correspondência dessa árvore com uma lista de candidatos que são obtidos na versão antiga do documento que possuem a mesma assinatura. Caso não haja nenhuma correspondência e o nó seja um elemento, seus filhos são adicionados a fila.

Para encontrar o melhor candidato em caso de muitos elementos, o selecionado é aquele cujo pai já foi correspondido, e equivale ao pai do elemento a ser casado. Caso nenhum concorrente seja aceito, o algoritmo verifica um nível acima, a quantidade de níveis a serem analisados depende do peso do nó. Caso contrário, enquanto possuir a mesma assinatura, o casamento é propagado para seus antepassados.

Dando continuidade, o próximo item consiste na utilização de uma estrutura criada para propagar os casamentos entre os nós, com objetivo de otimização. Nesse processamento, a fim de melhorar a qualidade do *delta* gerado é evitada a detecção de inserções e deleções desnecessárias. É realizada uma busca *botton-up* e outra *top-down* na árvore tentando casar os nós da versão original com os da nova versão, identificando assinaturas semelhantes e seus pais sejam equivalentes.

A etapa final se concretiza no cálculo do *delta*. Inicialmente é selecionado no novo documento XML os nós que não obtiveram correspondências na nova versão. Estes são marcados como inseridos e os nós presentes na versão antiga sem correspondentes na nova versão são marcados como removidos. Como alterados se enquadram os nós que foram casados porém possuem conteúdos diferentes. Os nós que obtiveram correspondência mas possuem pais diferentes em cada versão, e nós casados que possuem posições diferentes em relação a seus irmãos, são assinalados como movimentados. Por fim, ocorre uma reorganização das operações, é produzido o *delta* no formato XML.

## 3.3 XChange

O XChange (OLIVEIRA, 2016) é uma abordagem que usa inferência para apoiar a compreensão de mudanças entre versões de um documento XML. Diferentemente do X-Diff e do XyDiff, o XChange utiliza-se das informações explicitas associadas a cada versão para deduzir conhecimentos implícitos sobre as mudanças, ou seja, além de encontrar as mudanças de forma sintática, o mesmo também propõe verificar a semântica associada a

cada mudança.

A abordagem foi desenvolvida inicialmente na linguagem de programação Java. Estão presentes no XChange duas formas de identificar elementos correspondentes em documentos XML, a primeira delas denomina-se chave de contexto e a segunda, similaridade.

**Chave de Contexto:** baseia-se na informação de um atributo que identifica unicamente um determinado elemento, por exemplo, em uma base de funcionários pode-se dizer que o campo Nome identifica unicamente um determinado elemento, isto é, não existe mais de um funcionário com o mesmo nome;

**Similaridade:** nem sempre é possível determinar uma chave de contexto para o cenário analisado, ou garantir que o atributo chave permaneça o mesmo em todas as versões do documento (por exemplo, caso haja um erro de digitação no campo nome de uma versão para a outra). Em meio a isso a técnica de similaridade leva em consideração alguns itens presentes no documento XML, tais como a similaridade entre seus nomes, seus atributos, conteúdos textuais e subelementos, para encontrar os casamentos de elementos correspondentes entre as versões.

A Figura 3.3 apresenta uma visão geral do XChange, dividido em três etapas.

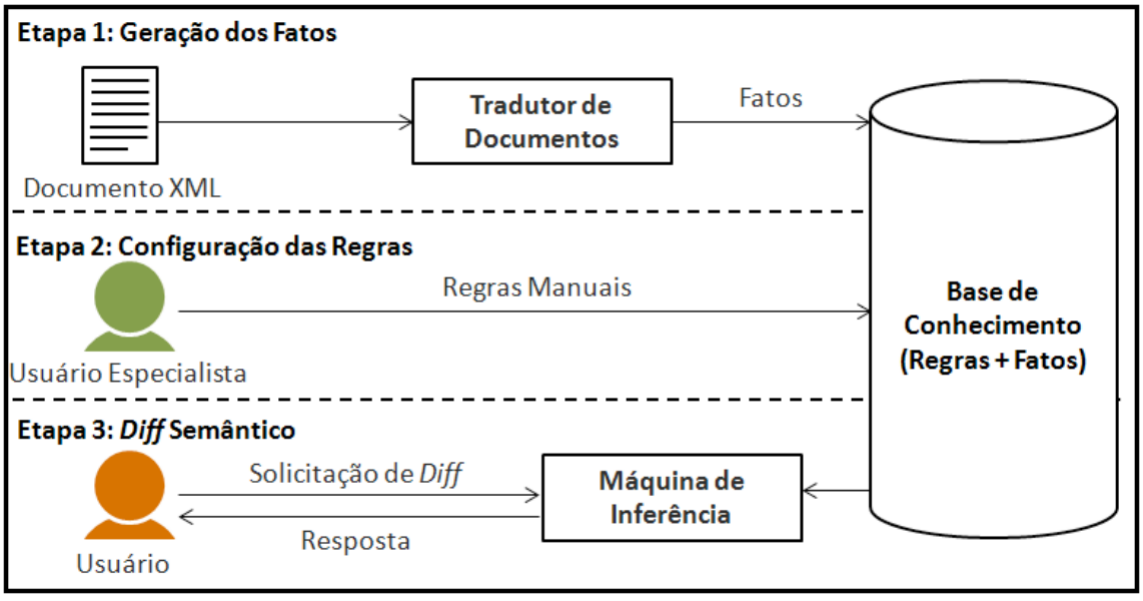


Figura 3.3: Visão geral XChange (OLIVEIRA, 2016)

Inicialmente, na primeira etapa, tem-se o conjunto de dados de entrada que são as versões do documento XML em estudo. Para realizar inferências lógicas sobre as informações contidas no conjunto de entrada, estes são transformados para fatos Prolog (LIMA et al., 2012). A saída da primeira etapa compõe a base de conhecimento, que recebe todas as informações do conjunto de entrada convertidos para fatos Prolog.

A próxima etapa, intitulada Configuração das Regras se caracteriza pela definição de regras relevantes para o domínio. Geralmente estas regras são criadas pelo especialista do cenário em questão. O XChange fornece a possibilidade de inserir essas regras manualmente, mas também proporciona a facilidade de selecionar uma determinada regra através da mineração de dados, onde as alterações são pré-analisadas e fornecem as informações das *tags* que mais sofrem alterações em conjunto, por exemplo, o que pode ajudar o especialista não deixando assim uma regra importante pra o mesmo, passar despercebida.

Tanto para a criação de regras e/ou a seleção das mesmas o XChange utiliza-se de uma interface gráfica de modo a facilitar tais procedimentos. Ao fim da identificação das regras do domínio as mesmas são convertidas para Prolog, e podem ser salvas a fim de serem carregadas futuramente para analisar outras versões, possibilitando assim agilizar o processo sem a repetição desses passos. A saída da segunda etapa se agrega aos fatos Prolog do conjunto de entrada já presente na base de conhecimento.

A etapa final se dá pela realização do *diff*. A máquina de inferência opera sobre os fatos e as regras armazenadas na base de conhecimento, produzindo assim o resultado semântico associado a cada regra fornecida pelo especialista. Ao final da inferência os resultados são apresentados ao usuário, podendo o mesmo repetir o processo quando desejar, uma vez que já possui as informações necessárias na base de conhecimento.

## 3.4 Demais Abordagens

A abordagem DeltaXML[5] é uma ferramenta comercial para detecção de mudanças em documentos XML. Diferentemente das demais abordagens citadas nesse trabalho, a abordagem admite ambos os tipos de árvores (ordenadas e não ordenadas) e suporta arquivos de entrada de até 50 Megabytes. O DeltaXML usa um algoritmo que roda em tempo

[5]https://www.deltaxml.com/

linear e suporta as operações de inserção, remoção e atualização. Segundo Cobéna, Abdessalem e Hinnach (2004), o DeltaXML é classificado como um algoritmo rápido, pois apesar de não rodar em tempo linear para o pior caso, é linear para o caso médio, e os resultados são próximos ao resultado ótimo.

O BioDiff (SONG et al., 2007) foi concebido para fins específicos no campo da Biologia Molecular Computacional. É capaz de detectar diferenças em dados que representam genomas, para fins de estudos das relações existentes entre diferentes espécies de seres vivos. Comporta árvores não ordenadas e possui complexidade quadrática. As operações suportadas por essa abordagem são inserção, remoção e atualização.

XKeyMatch (SANTOS, 2006) é uma abordagem com base no XyDiff. Faz uso de um conjunto de chaves XML para realizar o casamento entre os nós das árvores. Tais chaves XML buscam dar um significado semântico para a abordagem, de modo que as chaves possam apontar casamentos que, eventualmente, não seriam encontrados apenas realizando a análise sintática dos documentos.

A abordagem X-Rel (SUNDARAM; MADRIA, 2012) trabalha com banco de dados relacionais tendo como entrada documentos XML desordenados. Diferentemente das demais abordagens, ela utiliza banco de dados relacional e linguagem Structured Query Language (SQL). Tal proposta proporciona melhor eficiência para documentos de grande escala. Por utilizar banco de dados não realiza a conversão dos documentos XML para árvores, sendo assim reduz o consumo de memória, por evitar o carregamento de ambas as árvores para a memória.

O XKeyDiff (SANTOS; HARA, 2004) é uma abordagem de *diff* para XML que leva em consideração além da estrutura sintática, a estrutura semântica. Ele é resultado da combinacão de técnicas de chaves para XML e da abordagem XyDiff mencionado anteriormente. Foi implementado na linguagem C++, como um módulo do algoritmo XyDiff, utilizando e estendendo as estruturas de dados contidas nele. No mesmo é especificada a chave que identifica unicamente os elementos das versões dos documentos XML. A abordagem realiza o casamento correto dos elementos nas duas versões e este casamento é então propagado aos seus descendentes pela abordagem XyDiff. Desta forma, o *delta* resultante do algoritmo conterá apenas as alteração correspondentes ao elemento analisado.

## 3.5 Comparativo das abordagens

Esta seção apresenta um estudo comparativo sobre as abordagens mencionadas nesse capítulo.A Tabela 3.1 exibe em suas linhas as abordagens e em suas colunas, itens chaves para a comparação entre eles. A seguir, explora-se cada um dos itens presentes na tabela. As células compostas pelo símbolo "-" indicam que a informação da coluna não se aplica a abordagem, já as células com o símbolo "?" indicam que a resposta não foi encontrada com base na literatura.

Tabela 3.1: Comparativo entre as abordagens

| | **Memória** | **Árvore** | **Complexidade** | **Operações** | **Comercial** |
|---|---|---|---|---|---|
| X-Diff | quadrática | não ordenada | O($n^2$) | I, R, A | não |
| XyDiff | linear | ordenada | O(nlogn) | I, R, A, M | não |
| XChange | ? | ordenada | O($n^4$) | I, R, A, M | não |
| XKeyMatch | ? | ordenada | ? | I, R, A, M | não |
| XKeyDiff | ? | ordenada | ? | I, R, A, M | não |
| X-Rel | ? | - | ? | ? | não |
| DeltaXML | linear | ambas | ? | I, R, A | sim |
| BIODIFF | quadrática | não ordenada | O($n^2$) | I, R, A | não |

Memória: nos cenários reais e práticos, a memória disponível é limitada. O consumo de memória de um determinado algoritmo é de suma importância, pois nem sempre a velocidade e a eficiência são válidas para um cenário, causando assim uma limitação para o algoritmo.

Árvore: dois tipos de árvores são levados em consideração, as árvores ordenadas e as árvores não ordenadas. Na primeira, a ordem entre os irmãos é relevante, já na segunda, a ordem entre os irmãos não é considerada, porém em ambos os tipos a ordem entre pai e filho é sempre relevante.

Complexidade: a complexidade de um determinado algoritmo deve ser levada em consideração. Muitas das vezes o fato de uma determinada abordagem resolver um dado problema nem sempre quer dizer que o mesmo será aceitável. É necessário identificar qual

a função que determina o tempo gasto no seu processamento (ordem de um algoritmo). Também é importante ressaltar a velocidade em que a ordem do algoritmo cresce, isto é, se o tempo de execução cresce linearmente, quadraticamente, exponencialmente ou fatorialmente.

Operações: existem cinco tipos de operações de edição possíveis para uma árvore (inserção, remoção, atualização, movimentação e cópia). Visando manter a eficiência, as abordagens muitas das vezes abrem mão de cobrir as cinco operações e tomam somente um subconjunto dessas operações, o que por outro lado pode deixar o *diff* muito extenso, dificultando assim sua análise.

Comercial: existem abordagens de caráter comercial e abordagens com princípios *open source*.

## 3.6 Considerações Finais

Neste capítulo foi possível analisar alguns trabalhos presentes na literatura que tem como fundamento detectar mudanças entre versões de documentos XML, encontrando os elementos correspondentes. Das abordagens pioneiras existentes, X-Diff e XyDiff, foram utilizados o X-Diff para a avaliação experimental e o XChange, visto que em Oliveira (2016), o XyDiff foi retirado da avaliação experimental por realizar casamentos incorretos em grande número quando não é fornecido um esquema associado ao cenário e um identificador, devido a sua estratégia baseada na posição para combinar elementos.

# 4 Pré-Processamento para Avaliação

De forma a verificar a eficiência e eficácia no comparativo das abordagens, este presente capítulo apresenta a ferramenta desenvolvida para obter as respostas desejadas e as técnicas empregadas. Expõe-se também o procedimento realizado na etapa de pré-processamento dos resultados gerados pelas abordagens X-Diff e XChange.

## 4.1 XMeasure

O XMeasure foi desenvolvido para auxiliar a caracterização das abordagens de *diff* de documentos XML e casamentos de elementos correspondentes. A ferramenta automatiza o processo de análise dos resultados das abordagens e as exibe em termos de algumas medidas que são utilizadas em estatística e em recuperação de informação, tais como: verdadeiros positivos, falsos positivos, falsos negativos, precisão, cobertura e *F-Measure* (BAEZA-YATES; RIBEIRO-NETO, 1999).

A ferramenta foi desenvolvida na linguagem de programação Java e conta com um recurso para exportação dos dados para Excel, fazendo com que os resultados sejam melhores visualizados através de planilhas e gráficos. Tais resultados auxiliam à mensurar a eficiência e a eficácia.

De forma a melhor compreender as métricas utilizadas, estas são descritas a seguir e a Figura 4.1 apresenta tais conteúdo de forma visual:

**Verdadeiros Positivos (VP):** são todos os elementos que estão presentes no arquivo de similaridade e no gabarito, ou seja, os casamentos que são classificados como corretos;

**Falsos Positivos (FP):** se dá pelos elementos que são considerados como casamentos verdadeiros pela similaridade, porém tais elementos não estão presentes no gabarito, isto é, casamentos que são realizados porém classificados como incorretos;

**Falsos Negativos (FN):** são os elementos que estão corretos, ou seja, estão presentes

no gabarito, porém não estão presentes nos arquivos de similaridade, isto é, são casamentos não identificados com base no gabarito (resultados esperados);

**Precisão (P):** é a fração de elementos recuperados que são relevantes. Em nosso contexto, significa o quão correto estão os casamentos identificados pela abordagem. A precisão pode ser obtida da seguinte maneira:

$$P = \frac{ElementosGabarito \cap ElementosSimilaridade}{ElementosSimilaridade} \quad \text{ou} \quad P = \frac{VP}{VP + FP} \quad (4.1)$$

**Cobertura (C):** se dá pela parte de elementos relevantes que são recuperados. No contexto deste trabalho, indica o quão completa a relação de casamentos correspondentes pela abordagem está, quando comparada ao resultados esperados. A cobertura pode ser obtida da seguinte forma:

$$C = \frac{ElementosGabarito \cap ElementosSimilaridade}{ElementosGabarito} \quad \text{ou} \quad C = \frac{VP}{VP + FN} \quad (4.2)$$

***F-Measure*:** É a média harmônica entre a precisão e a cobertura, sendo possível obter o melhor compromisso entre a correção e integridade. A *F-Measure* pode ser obtida por:

$$F\text{-}Measure = \frac{2}{\frac{1}{precisao} + \frac{1}{cobertura}} \quad (4.3)$$

Tais métricas foram implementadas na ferramenta de forma que a mesma retorna-se de forma automática todos os itens necessários para realizar a caracterização dos casamentos de elementos correspondentes de documentos XML. A Figura 4.2 exibe a tela inicial da ferramenta.

Ao iniciar a ferramenta, o usuário seleciona os documentos que compõe a análise. Primeiramente carrega-se o gabarito, ou seja, um documento contendo as informações de todos os casamentos que deveriam ocorrer, que provém da abordagem de chave de contexto pertencente ao XChange, posteriormente, carrega-se o(s) documento(s) chamado(s) de similaridade, neste ponto carrega-se o(s) documento(s) da abordagem que queremos obter as métricas (XChange, X-Diff ou XyDiff).

Depois de carregar os arquivos na ferramenta, logo abaixo existe a opção de

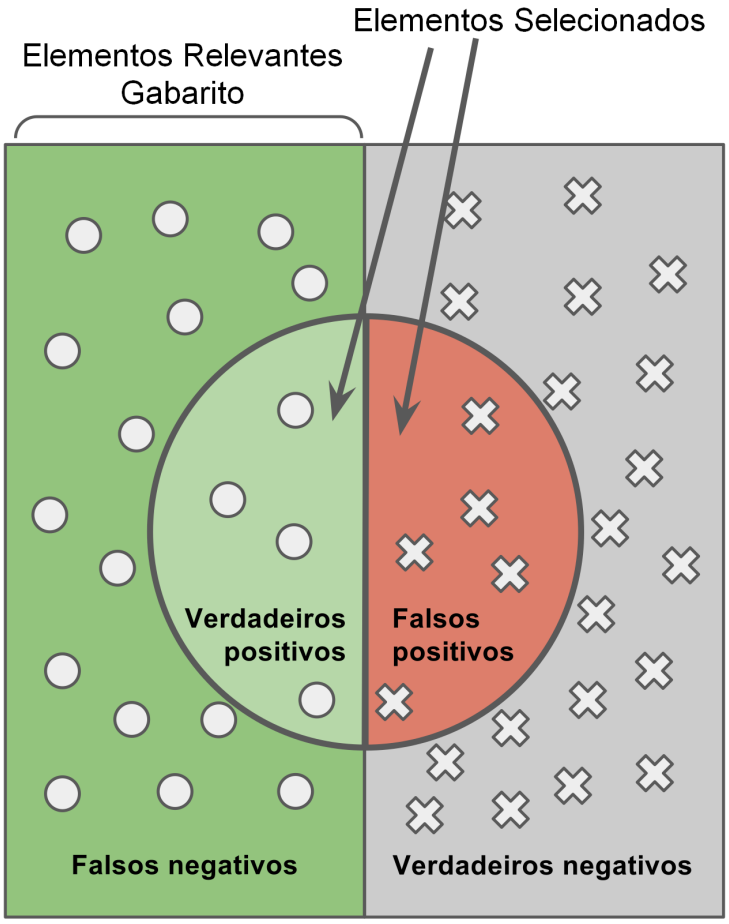


Figura 4.1: Métricas

escolha da abordagem que está sendo processada. Tal escolha torna-se necessária pelos padrões utilizados nas abordagens do comparativo, por exemplo, o limiar de similaridade 1.0 no XChange indica similaridade total dos componentes de cada elemento, já no X-Diff o limiar que representa o mesmo é o 0.0.

O terceiro botão (Comparar) da esquerda para a direita que aparece desabilitado na Figura 4.2, é responsável por iniciar o comparativo desses documentos e, posteriormente, apresentar os cálculos obtidos na caixa de texto em branco exibida na figura. Este botão é habilitado depois de os documentos serem carregados e após a seleção da abordagem a ser processada.

O último botão da ferramenta tem a funcionalidade de exportar esses dados para o Excel, fazendo com que os dados concebidos sejam melhores visualizados através de planilhas e gráficos.

O diagrama da Figura 4.3 expõe o funcionamento da ferramenta XMeasure, sendo

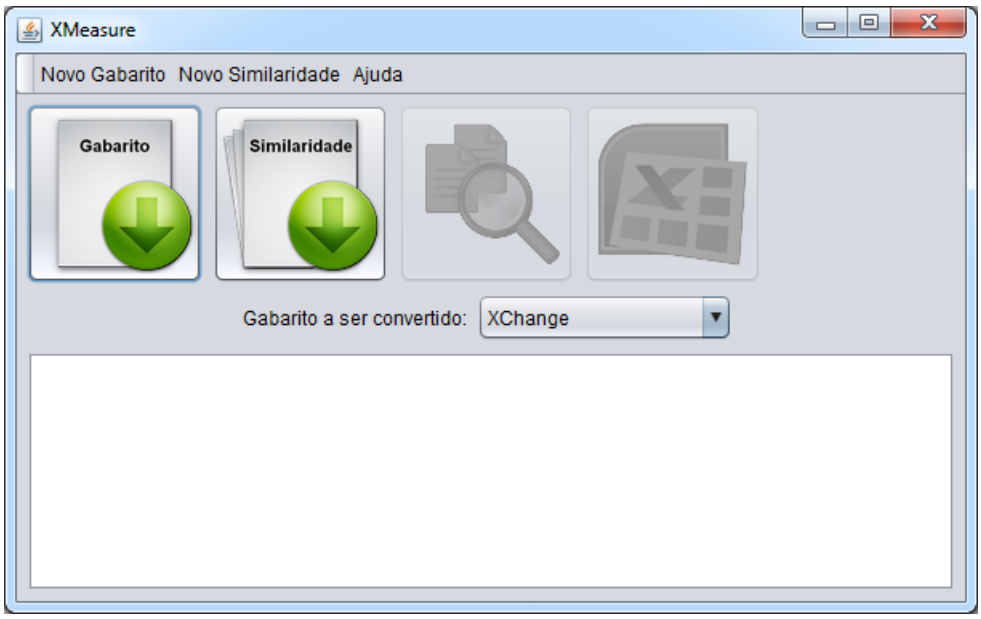


Figura 4.2: Interface inicial da ferramenta XMeasure

descrito através das ações que o usuário deve executar, apresentando o fluxo de execução da ferramenta e pode ser descrito a partir dos seguintes passos:

1. Abrir o Gabarito;
2. Abrir o(s) documento(s) a serem comparados;
3. Escolher a abordagem em análise;
4. Gerar os resultados;
5. Exportar para o Excel.

## 4.2 Abordagens na Avaliação Experimental

As próximas seções descrevem as implementações e processos realizados a partir dos resultados obtidos em cada abordagem para incluir em nossa avaliação experimental.

### 4.2.1 X-Diff

Para a utilização do X-Diff na avaliação experimental foi criado um algoritmo em Java para atender as necessidades requeridas, de tal forma que a abordagem nativa não fosse alterada.

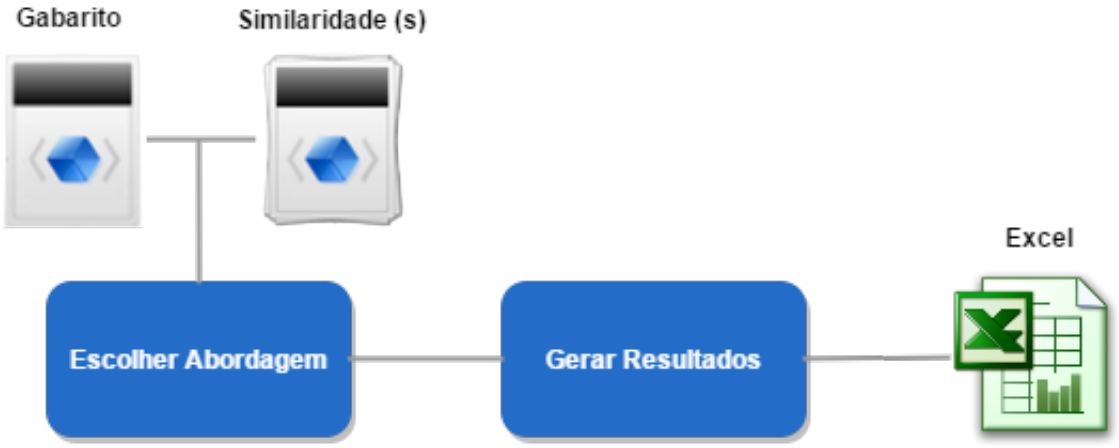


Figura 4.3: Fluxo de execução do XMeasure

Através de uma versão do X-Diff disponível em Java, foi desenvolvido um algoritmo para executá-lo e analisar sua saída. O início do procedimento se dá com a chamada do algoritmo do X-Diff para o par de versões desejado. Após executá-lo se inicia o algoritmo desenvolvido, que é responsável por ler a saída obtida pelo X-Diff e gerar o resultado esperado para a avaliação experimental no XMeasure.

De maneira mais detalhada, ao término da execução do X-Diff, o algoritmo realiza a leitura de sua saída. A Figura 4.4 exibe um pequeno exemplo de uma saída do algoritmo.

Após a leitura, o algoritmo identifica as *tags* presentes no documento XML e as exibe em uma caixa de diálogo pra que o usuário possa escolher qual é a chave de contexto do documento em questão, como pode ser visto na Figura 4.5. Nesse exemplo a chave de contexto é a *tag "name"*.

Como mencionado anteriormente, o X-Diff realiza 3 operações básicas (inserção, remoção e atualização). Consequentemente na saída da abordagem pode-se encontrar as seguintes *tags* "*INSERT*", "*DELETE*" e/ou "*UPDATE*".

A seguir o algoritmo desenvolvido realiza uma varredura na saída do X-Diff de forma a encontrar as *tags* mencionadas. Caso exista no funcionário analisado a *tag* "*UPDATE*", é necessário verificar se essa *tag* se encontra na *tag* escolhida como chave de contexto, em caso afirmativo, o funcionário é classificado como um casamento errado. Caso contrário, o mesmo é classificado como casamento correto, por não possuir uma atualização na *tag* que identifica unicamente o elemento no documento. Caso haja a *tag* "*INSERT*" ou "*DELETE*", os funcionários não são classificados como corretos ou errados,

```
<government>
    <employee>
        <name>Aaron,Patricia G</name>
        <jobtitle>Facilities/Office Services II</jobtitle>
        <agencyid>A03031</agencyid>
        <agency>OED-Employment Dev <?UPDATE FROM "OED-Employment Dev"?></agency>
        <hiredate>10/24/1979</hiredate>
        <annualsalary>51862<?UPDATE FROM "50845"?></annualsalary>
        <grosspay>52247.39<?UPDATE FROM "45505.94"?></grosspay>
    </employee>
        <employee>
        <name>Anderson,Caitlyn M<?UPDATE FROM "Adams,Diane"?></name>
        <jobtitle>POLICE OFFICER TRAINEE<?UPDATE FROM "NUTRITION TECHNICIAN"?></jobtitle>
        <agencyid>A99416<?UPDATE FROM "A65010"?></agencyid>
        <agency>Police Department <?UPDATE FROM "HLTH-Health Department"?></agency>
        <hiredate>04/17/2012<?UPDATE FROM "04/13/1987"?></hiredate>
        <annualsalary>43136<?UPDATE FROM "39468"?></annualsalary>
        <grosspay>6967.94<?UPDATE FROM "35673.41"?></grosspay>
        </employee>
    <employee>
        <name>Abdi,Ezekiel W</name>
        <jobtitle>POLICE OFFICER</jobtitle>
        <agencyid>A99398</agencyid>
        <agency>Police Department <?UPDATE FROM "Police Department"?></agency>
        <hiredate>06/14/2007</hiredate>
        <annualsalary>58244<?UPDATE FROM "50919"?></annualsalary>
        <grosspay>62669.25<?UPDATE FROM "51421.73"?></grosspay>
    </employee>
</government>
```

Figura 4.4: Exemplo da saída X-Diff

por se tratarem da admissão ou demissão de um funcionário.

Outra possibilidade é um elemento não possuir nenhuma das *tags* mencionadas, isso ocorre quando o elemento não apresenta mudanças de uma versão para a outra, consequentemente o mesmo é incluído como casamento correto.

O último passo se caracteriza em salvar esse novo resultado para ser utilizado no XMeasure. A Figura 4.6 exibe o novo resultado para o exemplo em questão, onde os funcionários que obtiveram casamentos corretos são exibidos primeiro, separados por uma linha em branco dos funcionários que foram classificados como casamentos errados.

### 4.2.2 XChange

Nenhum pré-processamento foi necessário ser efetuado na saída do XChange, uma vez que o alvo em Oliveira (2016) era comparar a presente abordagem com as demais, o mesmo será realizado aqui neste trabalho.

Para a avaliação experimental, deseja-se obter qual abordagem fornece a melhor

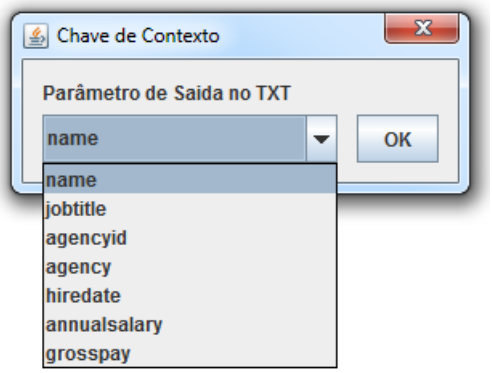


Figura 4.5: Caixa de diálogo X-Diff

```
Aaron,Patricia_G
Abdi,Ezekiel_W

Anderson,Caitlyn M
```

Figura 4.6: Nova saída X-Diff

eficiência e eficácia, de forma a encontrar os casamentos de elementos correspondentes. Com isso o XChange foi tomado em duas frentes:

**Arquivo de Gabarito:** nessa fase, os documentos XML são inferidos na abordagem, de forma a verificar os elementos correspondentes entre as versões. Para tal, foi criada uma regra em Prolog intitulada "*match*", que identifica os elementos que obtiveram o casamento corretamente, ou seja, aqueles atributos identificados unicamente que estão presentes em ambas as versões. Ao final, o resultado obtido através da abordagem recebe o nome de gabarito, por retratar exatamente os elementos que são correspondentes entre as versões.

**Arquivos de Similaridade:** tais arquivos são gerados através do XChange fazendo-se o uso da abordagem de similaridade. Ao contrário da abordagem de chave de contexto, que conta com um atributo identificador único em todo o documento, a abordagem de similaridade utiliza-se de outras técnicas para identificar os casamentos. Com isso, na inferência dos resultados, foi necessária a criação de duas regras, "*match*" e "*no_match*". A primeira, exerce o mesmo funcionamento da descrita para o arquivo de gabarito, ou seja, obtém os elementos correspondentes que tiveram casamento correto; já a segunda, se caracteriza pelos elementos que obtiveram

casamento, porém, esses casamentos foram classificados como incorretos, chegando a essa conclusão, partindo do ponto que para ambas as regras sabemos qual é o atributo identificador único (levado em consideração na chave de contexto), ou seja, o XChange utilizando a abordagem de similaridade, casou os elementos, porém por serem elementos diferentes (chave de contexto distintas) os mesmos são classificados como casamentos incorretos.

A Figura 4.7 exibe um pequeno exemplo da saída do XChange com a utilização das regras empregadas na abordagem de similaridade.

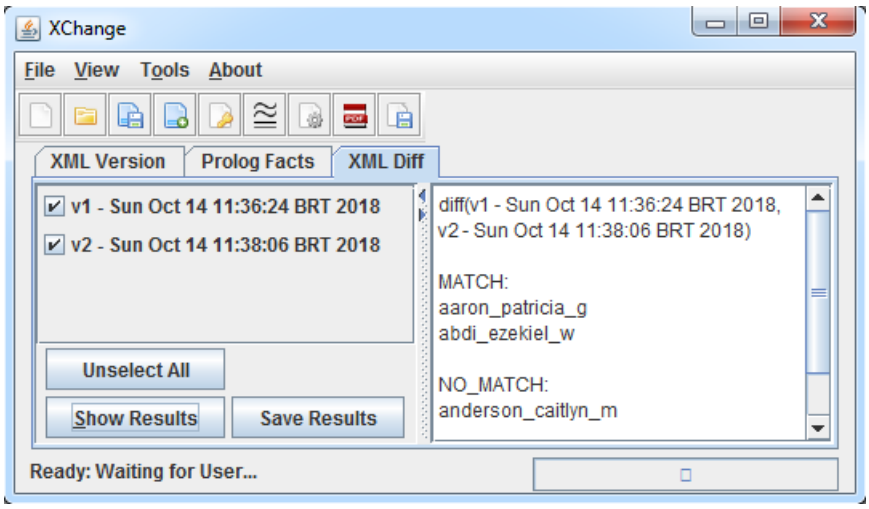


Figura 4.7: Exemplo da Saída XChange

Portanto, diferentemente do X-Diff que para ser comparado com o arquivo gabarito é necessário executar um pré-processamento em sua saída, o XChange não realiza nenhum esforço a mais, bastando executar apenas o fluxo normal com as regras mencionadas acima.

## 4.3 Considerações Finais

Tais conceitos de pré-processamento, tornam-se necessários para o entendimento da avaliação experimental. Este capítulo descreveu como os arquivos foram gerados para realizar a avaliação experimental para as duas abordagem, X-Diff e XChange, além de introduzir a ferramenta XMeasure desenvolvida para realizar a caracterização dos elementos correspondentes, bem como suas métricas utilizadas.

# 5 Avaliação Experimental

Esse capítulo tem por objetivo exibir os resultados obtidos a partir da identificação de elementos correspondentes em documentos XML. Através do casamentos de elementos correspondentes gerados pelas abordagens X-Diff e XChange, o estudo deseja avaliar a eficiência e eficácia alcançadas por ambas abordagens. O estudo tem foco em responder as mesmas perguntas levantadas em Oliveira (2016)

QP1. Qual abordagem é mais eficaz no que diz respeito ao casamento de elementos correspondentes entre versões de um documento XML?

QP2. Qual abordagem é mais eficiente no que diz respeito ao casamento de elementos correspondentes entre versões de um documento XML?

Tais questões podem ser solucionadas a partir das métricas utilizadas na área de Recuperação de Informação, como mencionadas no capítulo anterior. A eficácia é apresentada em termos da *F-Measure*. *A F-Measure* é a média harmônica entre a precisão e a cobertura, com o objetivo de encontrar o melhor compromisso entre correção e integridade. No contexto deste trabalho, a precisão indica quão correto estão os casamentos identificados pela abordagem. Por outro lado, a cobertura indica o quão completa a relação de casamentos correspondentes encontradas pela abordagem está, quando comparada aos resultados esperados. Finalmente, a eficiência é apresentada em termos do número de casamentos corretos por segundo.

Em meio a isso, o capítulo está organizado da seguinte forma. Na Seção 5.1, são apresentados os documentos XML das bases de dados selecionadas. A Seção 5.2 descreve a análise de sensibilidade para calibrar a abordagem de similaridade do XChange e definir o melhor limiar de similaridade para cada domínio. A Seção 5.3 mostra o processo utilizado na avaliação experimental. A Seção 5.4 descreve a execução da avaliação experimental, comparando o XChange com o X-Diff em cada base de dados. Finalmente, a Seção 5.5 apresenta as ameaças à validade da avaliação experimental enquanto a Seção 5.6 apresenta as considerações finais deste capítulo.

## 5.1 Descrição das Bases

Durante a avaliação experimental, foram utilizados documentos XML pertencentes a dois cenários diferentes. Todos os documentos são considerados representativos (MIGNET; BARBOSA; VELTRI, 2003), pois possuem 3 níveis de profundidade e não contém atributos:

**Condado de Montgomery (CM):** o Condado de Montgomery é um dos 23 condados do estado norte americano de Maryland. O documento XML provém do escritório de recursos humanos e é composto por informação salarial anual, incluindo remuneração bruta e pagamento de horas extras para todos os funcionários ativos e permanentes do condado. Tais informações são publicadas anualmente. Foram utilizadas as versões disponíveis, sendo a primeira delas do ano de 2014[6], a segunda de 2015[7] e a última pertencente ao ano de 2016[8].

**Universidade da Califórnia (UC):** de forma a manter a transparência dos dados e por responsabilidade pública, a UC divulga publicamente os dados de pagamentos dos empregados, sendo eles com carreira na universidade, temporários e estudantes. A universidade possui vários polos em localidades distintas. Para compor esta avaliação experimental, foram extraídos os dados da Universidade da Califórnia em Berkeley [9], sendo visíveis as informações desde o ano de 2010 até o ano de 2016, ou seja, 7 versões do documento XML.

Nesses estudos foram utilizadas as versões disponíveis de cada cenário, sendo renomeadas por conveniência utilizando-se da letra "v" seguida do número de forma incremental, por exemplo, a base de dados do CM possui 3 versões: 2014, 2015 e 2016, logo as mesmas foram nomeadas v1, v2 e v3. O mesmo foi efetuado para a UC. A Tabela 5.1 e a Tabela 5.2 exibe a quantidade de elementos em cada cenário e o tamanho em KBytes de cada versão.

A fim de maximizar o número de documentos XML neste estudo, cada documento XML original dos dois cenários encontrados foi dividido em fragmentos menores,

[6]https://data.montgomerycountymd.gov/Human-Resources/Employee-Salaries-2014/54rh-89p8
[7]https://data.montgomerycountymd.gov/Human-Resources/Employee-Salaries-2015/6rqk-pdub
[8]https://data.montgomerycountymd.gov/Human-Resources/Employee-Salaries-2016/xj3h-s2i7
[9]https://ucannualwage.ucop.edu/wage/

Tabela 5.1: Caracterização do documento XML do Condado de Montgomery

| versão | #elementos | tamanho (KB) |
|---|---|---|
| v1 | 9074 | 4962 |
| v2 | 9062 | 4954 |
| v3 | 9106 | 4981 |

Tabela 5.2: Caracterização do documento XML da Universidade da Califórnia - Berkeley

| versão | #elementos | tamanho (KB) |
|---|---|---|
| v1 | 31326 | 7713 |
| v2 | 32331 | 7942 |
| v3 | 32513 | 7565 |
| v4 | 34245 | 7794 |
| v5 | 35369 | 8020 |
| v6 | 35373 | 8023 |
| v7 | 35540 | 8061 |

de forma que a base CM resultou em 8 fragmentos, numerados de 0 a 7; a base UC após a fragmentação ocasionou em 11 novos fragmentos, numerados de 0 a 10. A Tabela 5.3 e as Tabelas 5.4 e 5.5 mostram o número de elementos (colunas #emp) e o tamanho em KBytes (colunas tam) de cada fragmento. Em cada base de dados, uma *tag* dita como chave de contexto para o cenário, foi utilizada como parâmetro para a fragmentação horizontal (ANDRADE et al., 2006) para manter os mesmos elementos nos fragmentos em todas as revisões. A fragmentação visa também equilibrar a #emp. Os elementos pertencentes a um determinado fragmento foram selecionados de forma a obedecer o que pode ser observado na coluna critério das tabelas, por exemplo, o fragmento 0 da base do CM contém os funcionários com nomes começando com as letras A e B, o fragmento 1 é composto pelos funcionários que iniciam seus nomes com as letras C, D e E, e assim por diante (coluna critério).

| **frag** | **critério** | v1 | | v2 | | v3 | |
|---|---|---|---|---|---|---|---|
| | | #emp | tam | #emp | tam | #emp | tam |
| 0 | AB | 1140 | 625 | 1152 | 631 | 1170 | 641 |
| 1 | CDE | 1354 | 742 | 1343 | 735 | 1349 | 739 |
| 2 | FGH | 1430 | 782 | 1402 | 766 | 1423 | 778 |
| 3 | IJKL | 1152 | 629 | 1143 | 624 | 1150 | 628 |
| 4 | MNO | 1198 | 656 | 1205 | 659 | 1215 | 666 |
| 5 | PQR | 929 | 508 | 932 | 509 | 942 | 515 |
| 6 | ST | 1131 | 619 | 1134 | 621 | 1124 | 615 |
| 7 | UVXWYZ | 740 | 406 | 751 | 412 | 773 | 403 |

Tabela 5.3: Características dos fragmentos da base do CM (tamanho em Kb)

| frag | critério | v1 | | v2 | | v3 | |
|---|---|---|---|---|---|---|---|
| | | #emp | tam | #emp | tam | #emp | tam |
| 0 | A | 1369 | 336 | 1382 | 338 | 1401 | 333 |
| 1 | BC | 1748 | 432 | 1764 | 433 | 1774 | 424 |
| 2 | DE | 1899 | 468 | 1904 | 465 | 1919 | 455 |
| 3 | FGHI | 1338 | 328 | 1337 | 326 | 1321 | 313 |
| 4 | J | 1961 | 482 | 1954 | 477 | 1924 | 457 |
| 5 | KL | 1823 | 450 | 1832 | 448 | 1862 | 443 |
| 6 | M | 1769 | 437 | 1779 | 236 | 1772 | 422 |
| 7 | NOPQ | 1135 | 279 | 1123 | 275 | 1103 | 263 |
| 8 | R | 1063 | 263 | 1083 | 266 | 1058 | 252 |
| 9 | S | 1445 | 335 | 1462 | 357 | 1413 | 336 |
| 10 | TUVWXYZ | 1578 | 386 | 1548 | 377 | 1555 | 369 |

Tabela 5.4: Características dos fragmentos da base da UC (tamanho em Kb)

| frag | critério | v4 | | v5 | | v6 | | v7 | |
|---|---|---|---|---|---|---|---|---|---|
| | | #emp | tam | #emp | tam | #emp | tam | #emp | tam |
| 0 | A | 1493 | 347 | 1579 | 365 | 1612 | 373 | 1628 | 376 |
| 1 | BC | 1861 | 435 | 1901 | 441 | 1941 | 451 | 1889 | 439 |
| 2 | DE | 1984 | 461 | 2011 | 464 | 2030 | 469 | 2031 | 269 |
| 3 | FGHI | 1341 | 312 | 1342 | 310 | 1353 | 313 | 1358 | 314 |
| 4 | J | 2038 | 474 | 2087 | 481 | 2102 | 485 | 2109 | 487 |
| 5 | KL | 1923 | 447 | 1960 | 453 | 2037 | 470 | 1938 | 448 |
| 6 | M | 1810 | 422 | 1822 | 422 | 1839 | 426 | 1807 | 419 |
| 7 | NOPQ | 1152 | 268 | 1193 | 276 | 1170 | 271 | 1156 | 267 |
| 8 | R | 1078 | 252 | 1109 | 257 | 1073 | 248 | 1089 | 252 |
| 9 | S | 1479 | 344 | 1489 | 344 | 1506 | 348 | 1510 | 349 |
| 10 | TUVWXYZ | 1625 | 377 | 1637 | 378 | 1616 | 373 | 1579 | 364 |

Tabela 5.5: Características dos fragmentos da base da UC (tamanho em Kb)

A Figura 5.1 e a Figura 5.2 ilustram como os conteúdos de cada fragmento evolui ao longo do tempo. Embora não haja nenhum esquema associado a estes documentos XML, uma análise manual mostrou que em todas as bases existe uma *tag* que identifica unicamente um elemento. Tal *tag* não sofre alterações entre as versões, e é única - sendo assim utilizada para gerar os resultados esperados no que diz respeito ao casamento de elementos correspondentes. Cada gráfico na Figura 5.1 e da Figura 5.2 representa a comparação entre duas versões consecutivas de cada fragmento. A coluna #atualizações exibe o número de elementos presentes em ambas as versões, com ou sem modificações em alguma de suas *tags*. A coluna #remoções mostra o número de elementos presentes apenas na revisão anterior. Por outro lado, a coluna #inserções mostra o número de elementos que estão presentes apenas na versão mais recente. Embora haja algumas variações, todos os fragmentos possuem uma distribuição similar, por exemplo, na base do CM a quantidade de atualizações é sempre maior que a quantidade de remoções e inserções (que praticamente se igualam); já no cenário da UC o mesmo ocorrido na base do CM se repete, exceto na comparação "v4 x v5" onde houve superioridade da quantidade de remoções e inserções se comparados a quantidade de atualizações.

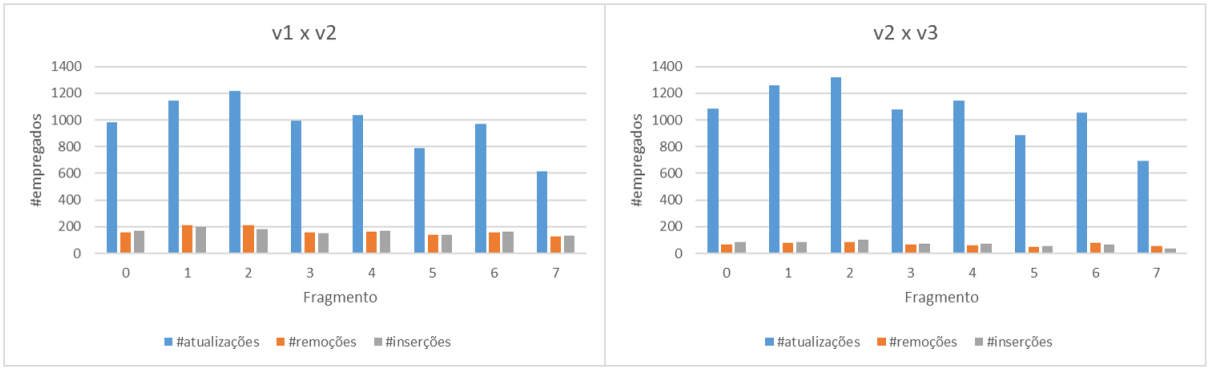


Figura 5.1: Características do documento XML do Condado de Montgomeyy

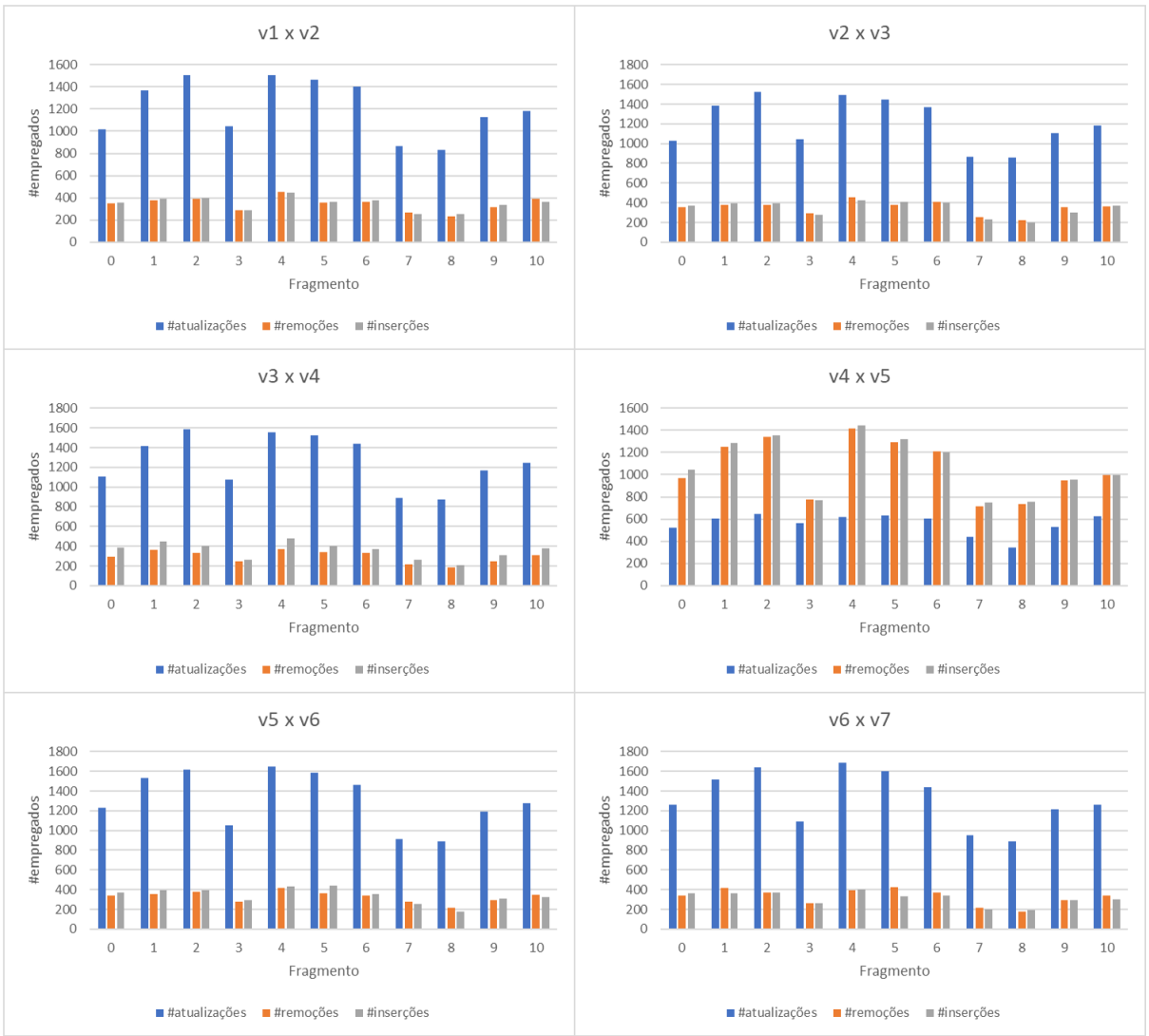


Figura 5.2: Características do documento XML da Universidade da Califórnia

A Figura 5.3 e a Figura 5.4 mostram a distribuição do número de mudanças entre os elementos correspondentes. Todos os elementos sofrem alterações em pelo menos um dos seus subelementos, ao comparar duas versões consecutivas. O eixo y indica quantos elementos têm alterações para as quantidades informadas na legenda horizontal dos gráficos. Em específico, para a base do CM, poucos funcionários têm alterações em 1, 5, 6 e 7 dos seus subelementos e, por isso, não são mostrados (a quantidade de elementos que sofre alterações nesses valores compreendem-se na faixa de 0 a 14 elementos). Como pode ser observado, a maioria dos funcionários têm mudanças em dois de seus subelementos entre duas versões consecutivas.

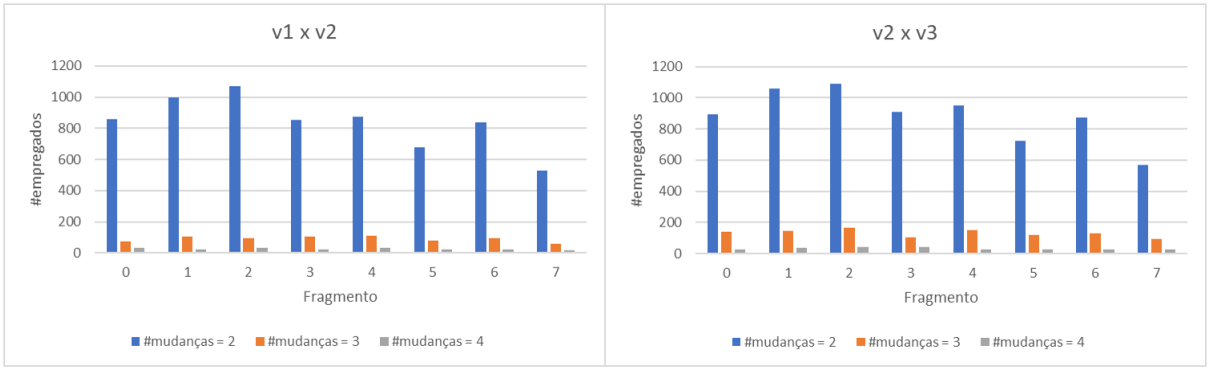


Figura 5.3: Evolução do documento XML do Condado de Montgomey

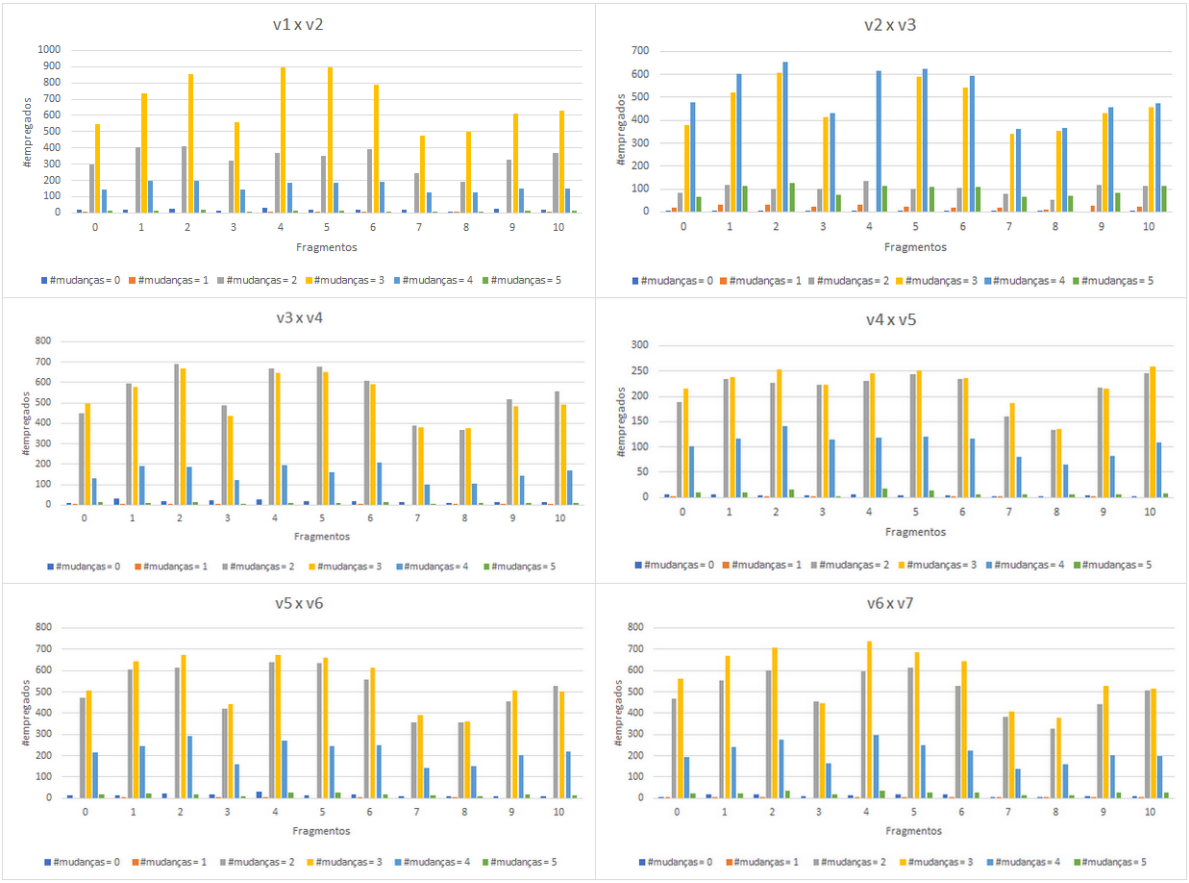


Figura 5.4: Evolução do documento XML da Universidade da Califórnia

De forma a facilitar a recuperação e a reutilização destes fragmentos dos documentos XML utilizados nesta avaliação experimental, os mesmos foram disponibilizados no repositório *XMLDatasets*[10] pertencente ao GitHub.

## 5.2 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 os dois domínios em estudo. Para isso, foi utilizado o fragmento 0 de todas as revisões. Comparou-se cada versão consecutiva utilizando a abordagem de similaridade do XChange, variando o limiar de similaridade entre 0 e 1, com incrementos de 0,01. Foi calculada a eficácia da abordagem em cada execução, o que permitiu determinar os melhores valores do limiar para cada comparação. Também foi possível verificar se os valores-limite que obtiveram melhor eficácia na primeira execução

[10]https://github.com/getcomp-dev/XMLDatasets

são os mesmos para os demais pares das versões. Durante todas as execuções, foram utilizados os valores padrão estabelecidos pela abordagem de similaridade, utilizando-se do parâmetro de similaridade de tipos ativada.

A Figura 5.5 e a Figura 5.6 mostram a curva *F-Measure* para as todas as comparações. O maior valor para *F-Measure* em cada comparação é destacado. Pode-se observar para a base do CM que as duas comparações obtiveram a maior *F-Measure* com limiares de similaridade diferentes; já a base da UC em quatro comparações obtiveram a maior *F-Measure* com mesmo limiar de similaridade, sendo em duas 0,65 e as outras duas comparações com 0,66 enquanto as outras comparações obtiveram valores menores para *F-Measure*, que além de distintos apresentaram valores maiores para os limiares. Portanto, podem-se concluir que não existe um limiar de similaridade único que maximiza *F-Measure*, embora estejam entre 0,70 e 0,80 para a base CM, 0,65 e 0,79 para UC.

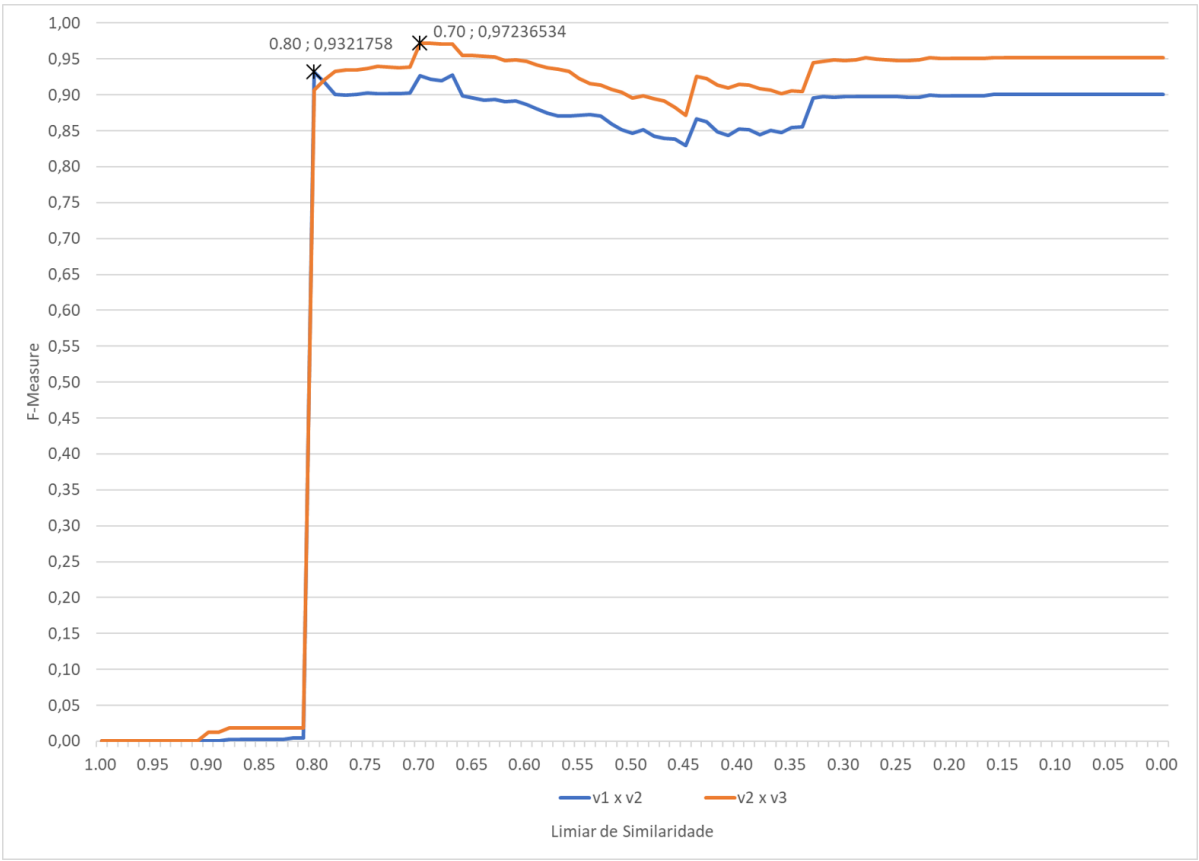


Figura 5.5: *F-Measure* de acordo com a variação do limiar de similaridade - Condado de Montgomery

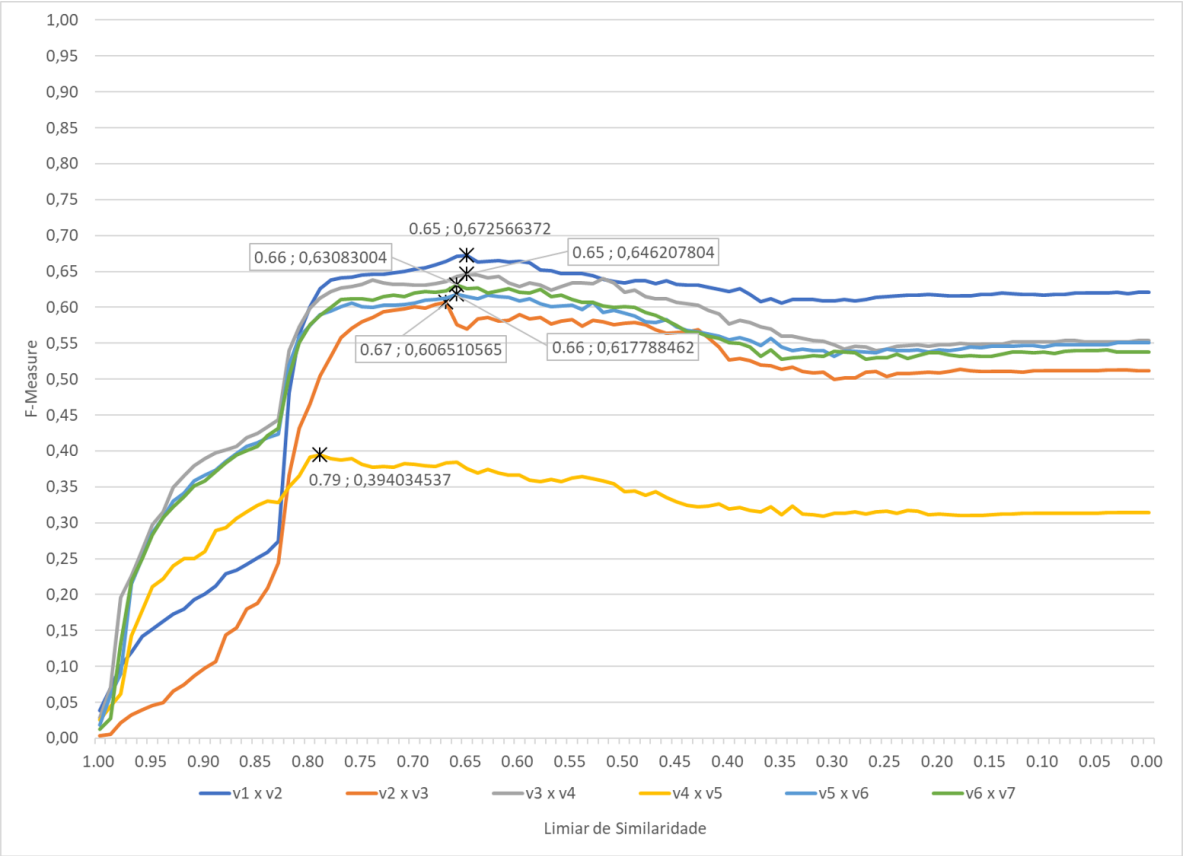


Figura 5.6: *F-Measure* de acordo com a variação do limiar de similaridade - Universidade da Califórnia Berkeley

A fim de adotar um único limiar base para a avaliação experimental como em Oliveira (2016), uma vez que o limiar que maximiza a *F-Measure* em todas as comparações não é o mesmo, os valores da *F-Measure* obtidos para cada limiar nas execuções foram somados, e posteriormente divididos pela quantidade de fragmentos existentes, de forma a se obter a média dos melhores valores obtidos para a *F-Measure*. Com isso tem-se:

**Condado de Montgomery (CM):**

$$Limiar = \frac{0,800 + 0,700}{2} = 0,750 \tag{5.1}$$

**Universidade da Califórnia (UC):**

$$Limiar = \frac{0,65 + 0,67 + 0,65 + 0,79 + 0,66 + 0,66}{6} = 0,68 \tag{5.2}$$

## 5.3 Processo da Avaliação Experimental

Na avaliação experimental, foram utilizados os fragmentos gerados a partir das versões do documento XML de cada cenário apresentado na Seção 5.1. Os Fragmentos 0 de todos as versões foram excluídos, uma vez que estes foram utilizados apenas para definir o limiar de similaridade base para cada domínio a ser utilizado pela abordagem de similaridade do XChange. Esta avaliação experimental centra-se na eficácia e eficiência alcançadas pelo XChange, em comparação com o X-Diff. Como representado na Figura 5.7, para cada par de versões consecutivas processadas nas 2 abordagens (X-Diff e XChange), 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, obtidos pela abordagem de chave de contexto do XChange). Como para cada cenário foi reconhecido um identificador único para os elementos destes documentos XML, eles foram utilizados para gerar os resultados esperados. Depois disso, foram calculadas as métricas de 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 à esses documentos das bases, nenhuma das abordagens estava ciente de que existia uma *tag* que se caracterizava como identificador nesses documentos XML, de modo que nenhuma delas foi capaz de usar esta informação para efetuar o casamento dos elementos correspondentes.

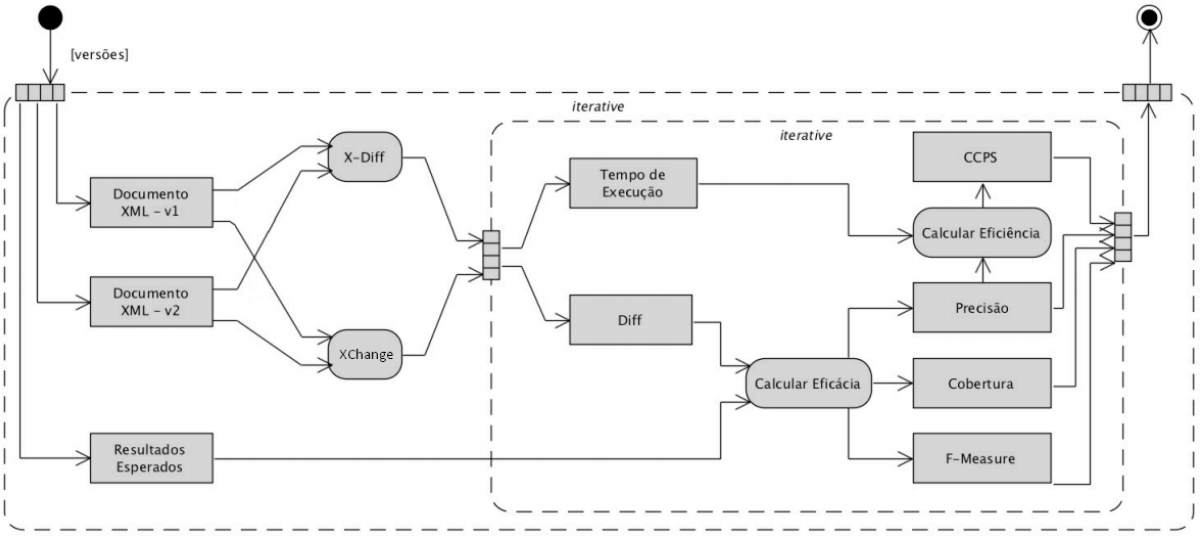


Figura 5.7: Processo da avaliação experimental (OLIVEIRA, 2016)

## 5.4 Avaliação da Eficiência e Eficácia

Com os valores dos limiares de similaridade base do XChange definidos (0,75 para CM e 0,68 para UC), pode-se comparar a eficácia e a eficiência entre as abordagens XChange e X-Diff. Foram executadas comparações envolvendo todas as versões consecutivas de cada fragmento (1 a 7 para a base do CM e de 1 a 10 para a base da UC) sendo utilizados os parâmetros recomendados pelo autor do X-Diff.

A Figura 5.8 e a Figura 5.9 apresentam os resultados das precisões obtidos pelo XChange e pelo X-Diff para o CM e a UC, respectivamente. Primeiramente, para o CM, pode-se notar que o XChange obteve melhor mediana no primeiro cenário e o X-Diff no segundo. Além disso, em todos os cenários a precisão assumiu valores acima de 80% para o XChange e para o X-Diff. Para a UC, temos um empate no comparativo das medianas. A precisão assumiu valores acima de 62% para o XChange e para o X-Diff, exceto no comparativo v4xv5.

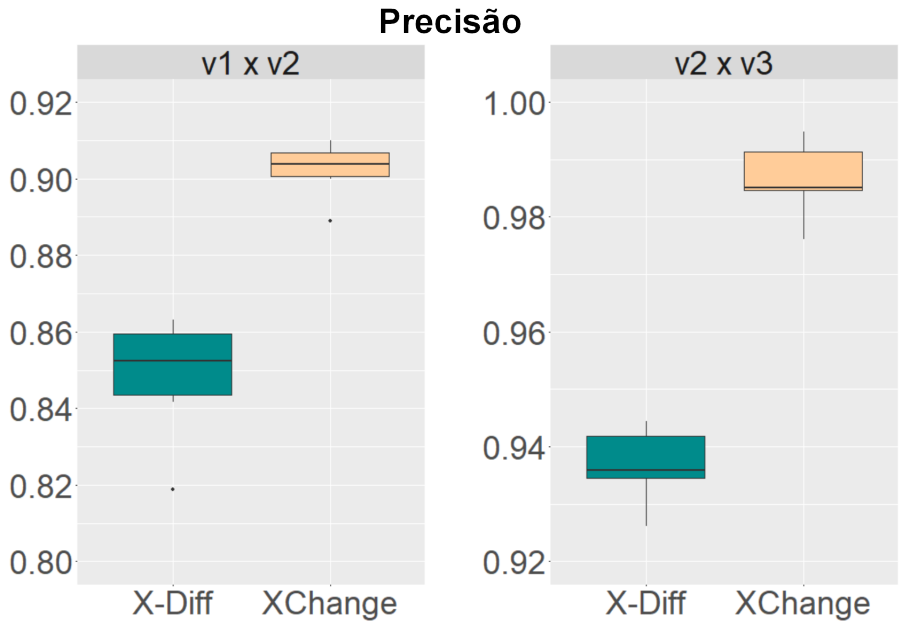


Figura 5.8: Resultados obtidos - Precisão CM

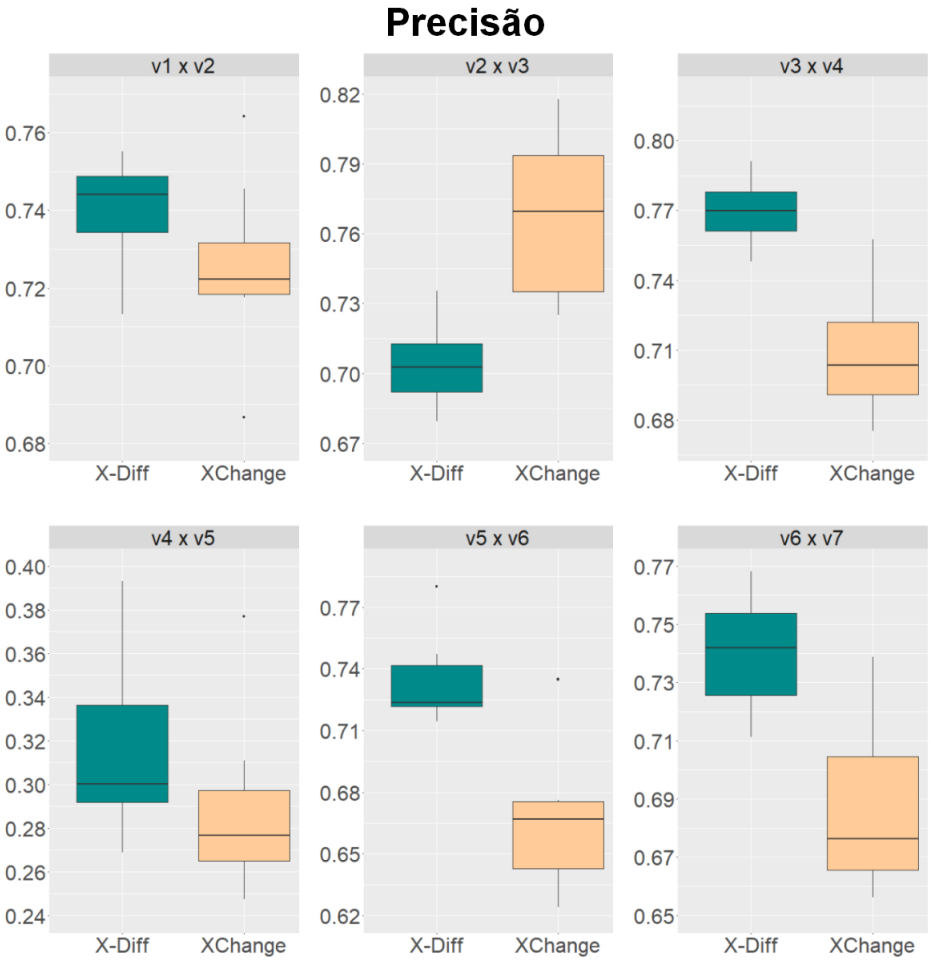


Figura 5.9: Resultados obtidos - Precisão UC

A Figura 5.10 e a Figura 5.11 apresentam os resultados das coberturas obtidos pelo XChange e pelo X-Diff para o CM e a UC, respectivamente. Novamente para o CM houve um empate, onde cada abordagem obteve melhor cobertura em um comparativo. Também observou-se valores de cobertura elevados para ambas as abordagens em todos os comparativos (acima de 85%). Já para a UC, a cobertura assumiu valores acima de 53% para o XChange e para o X-Diff.

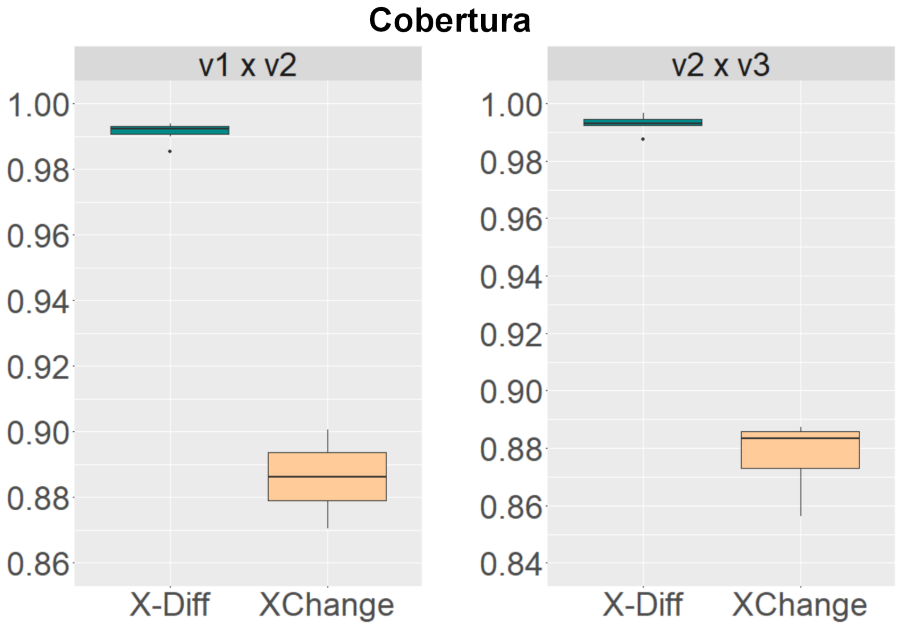


Figura 5.10: Resultados obtidos - Cobertura CM

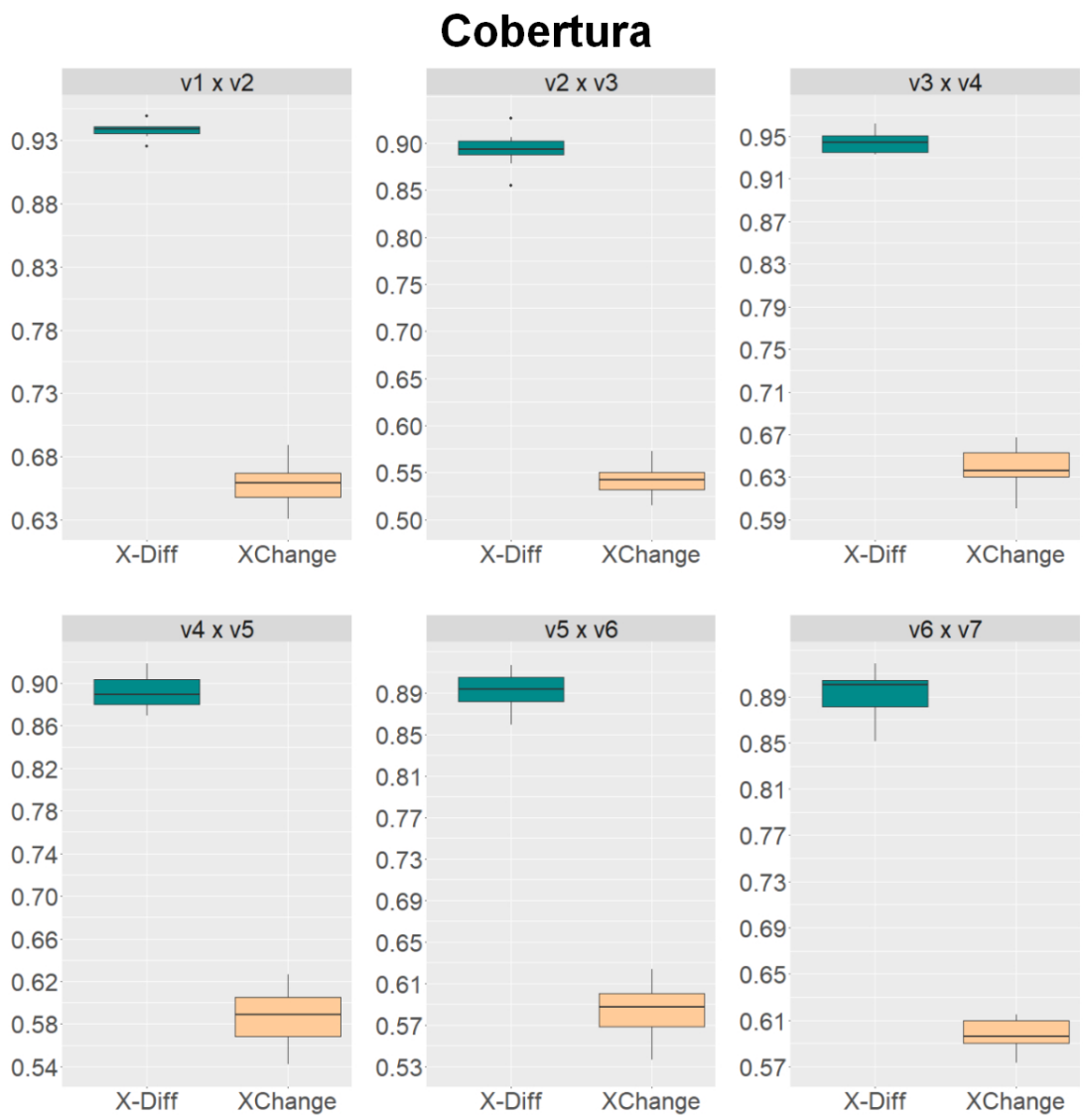


Figura 5.11: Resultados obtidos - Cobertura UC

Como pode-se observar nas Figuras 5.8, 5.9, 5.10 e 5.11, ambas as abordagens alcançaram valores elevados de precisão e cobertura, mesmo sem a existência de um identificador. Houve um empate em meio a essas métricas, pois em determinado cenário o XChange apresentava maior valor, e em outro cenário o X-Diff era superior. Visando identificar qual abordagem tem a melhor eficácia, a Figura 5.12 e a Figura 5.13 apresentam os resultados de *F-Measure* obtidos para o CM e a UC respectivamente. Para o CM, o X-Diff apresentou mediana maior nos dois comparativos. Ao analisar os 7 fragmentos dos 2 pares de versões consecutivas, observou-se que o X-Diff obteve melhor resultados em todas as 14 comparações, sendo levemente superior. Este resultado mostra valores elevados da *F-Measure* tanto para o XChange ($\mu$ = 0,9117, $\alpha$ = 0,0187) quanto para o X-Diff ($\mu$

= 0,9393, $\alpha$ = 0,0259), mas com uma ligeira vantagem para X-Diff. O teste estatístico de Mann-Whitney (WILCOXON, 1945) mostra que esta diferença não é estatisticamente significativa (valor de p = 0.9926). Analisando a UC, o X-Diff apresentou mediana maior em três comparativos e o XChange nos outros 3. Ao analisar os 10 fragmentos dos 6 pares de versões consecutivas, observou-se que o X-Diff obteve melhor resultados em todas as 60 comparações, sendo levemente superior. Este resultado mostra valores elevados da *F-Measure* tanto para o XChange ($\mu$ = 0,6069, $\alpha$ = 0,1050) quanto para o X-Diff ($\mu$ = 0,7573, $\alpha$ = 0,1341), mas com uma ligeira vantagem para X-Diff. O teste estatístico de Mann-Whitney (WILCOXON, 1945) mostra que esta diferença não é estatisticamente significativa (valor de p = 1).

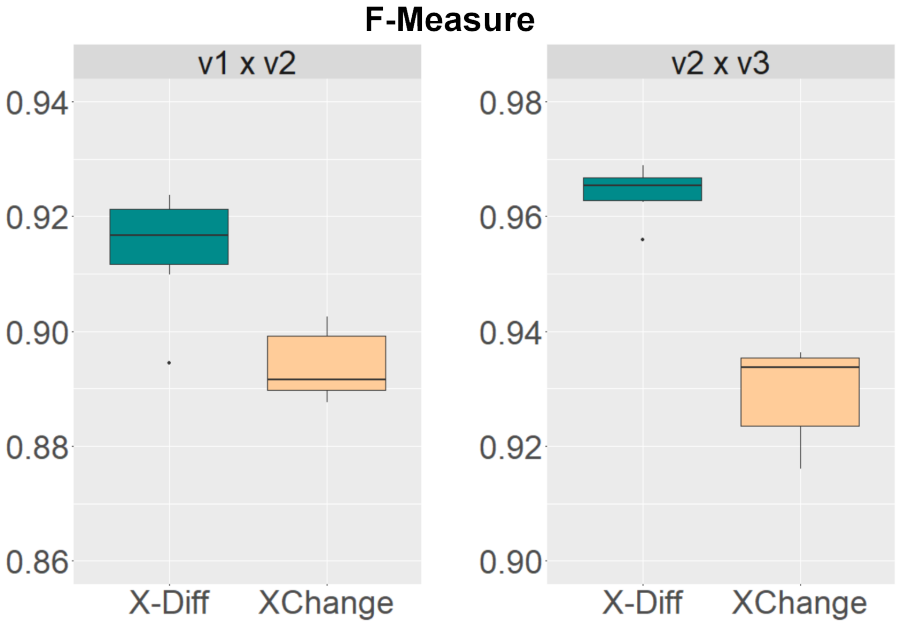


Figura 5.12: Resultados obtidos - *F-Measure* CM

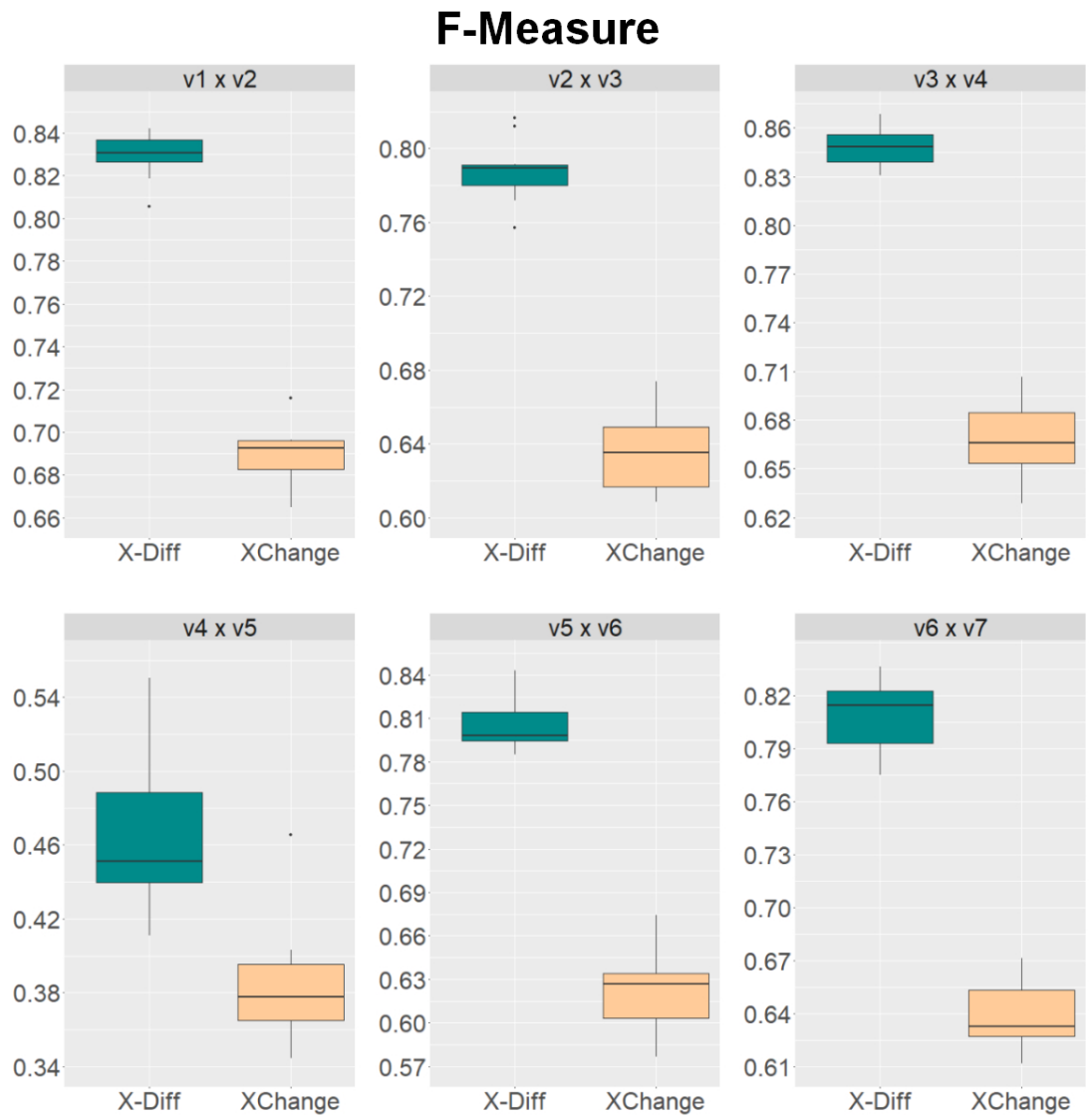


Figura 5.13: Resultados obtidos - *F-Measure* UC

Como o XChange e o X-Diff apresentam resultados equivalentes em termos de eficácia, uma análise complementar contrasta a eficiência de ambas as abordagens. A Figura 5.14 e a Figura 5.15 mostram os casamentos corretos por segundo (CCPS) de cada abordagem para o CM e a UC, respectivamente. Analisando os resultados para o CM, o XChange obteve o melhor resultado em todos as comparações. Neste caso, a diferença entre o XChange e o X-Diff é estatisticamente significativa (valor p $= 6,355 \cdot 10^{-6}$). Esta é uma consequência de uma grande diferença do tempo de execução do XChange ($\mu$ = 495,57, $\alpha$ = 174,73) e do X-Diff ($\mu$ = 1907,07, $\alpha$ = 1115,34). Já para a UC, o XChange também obteve o melhor resultado em todas as comparações. A diferença entre as abordagens é estatisticamente significativa (valor p $= 2,2 \cdot 10^{-16}$). Esta é uma

consequência de uma grande diferença do tempo de execução do XChange ($\mu$ = 350,25, $\alpha$ = 142,19) e do X-Diff ($\mu$ = 12227,78, $\alpha$ = 8980,28). O XChange é quase 4 vezes mais rápido do que X-Diff para o cenário do CM e quase 35 vezes mais rápido que o X-Diff para a UC.

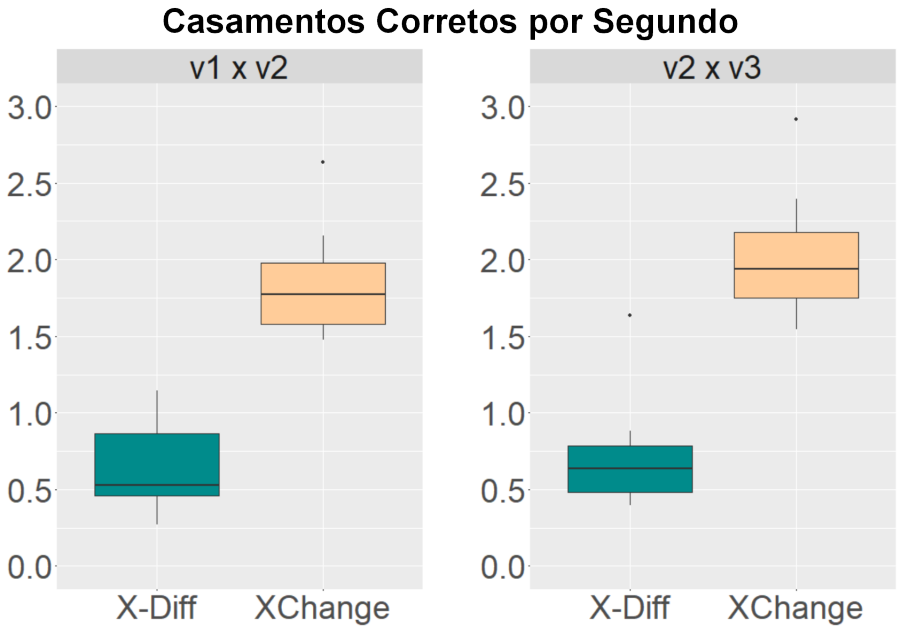


Figura 5.14: Resultados obtidos - CCPS CM

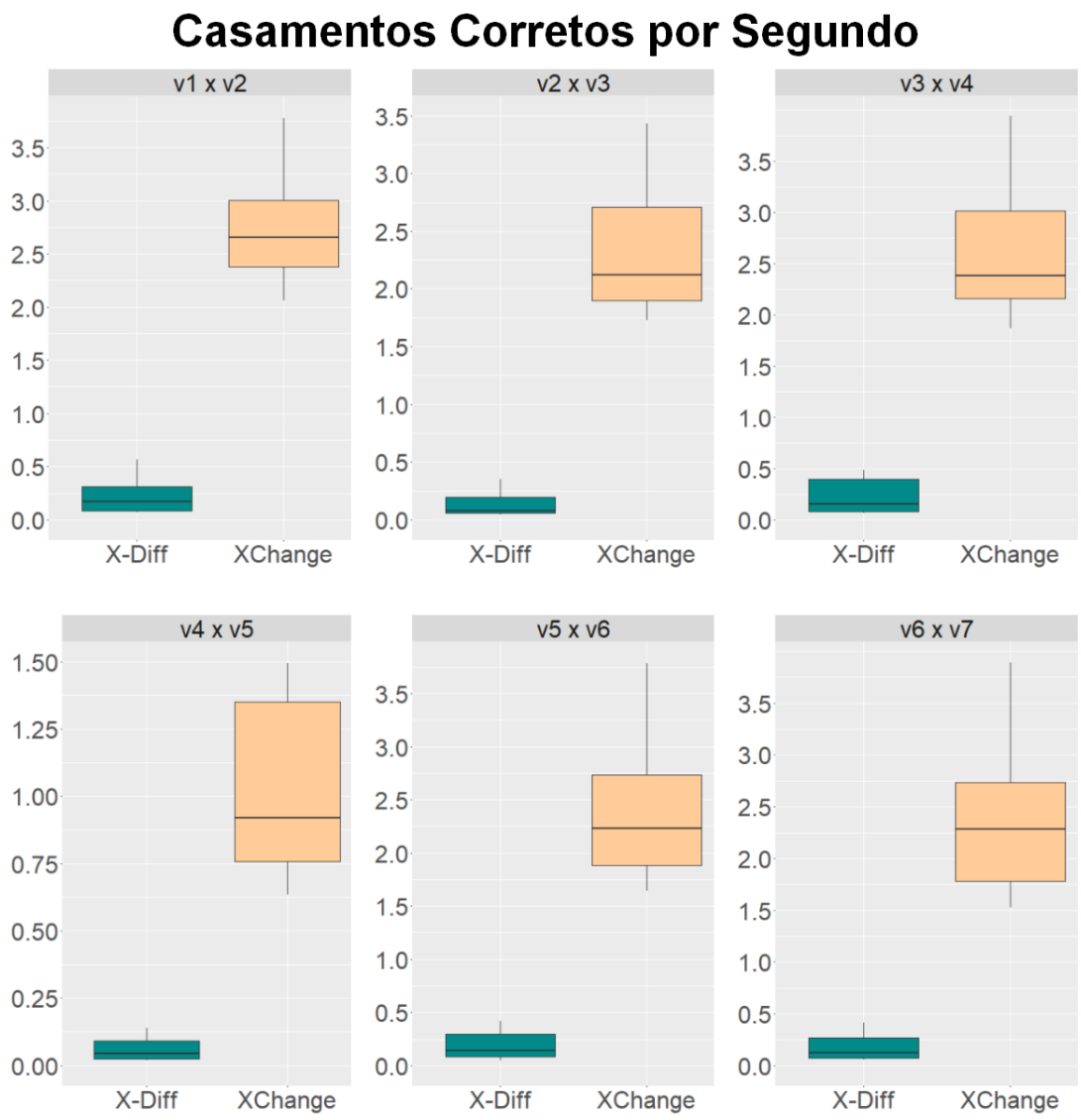


Figura 5.15: Resultados obtidos - CCPS UC

Em síntese, o XChange e o X-Diff apresentaram valores elevados para a *F-Measure*. O X-Diff obteve melhores resultados em todos os fragmentos. No entanto, o XChange alcançou resultados semelhantes em termos de eficácia em apenas uma fração do tempo de execução do X-Diff.

## 5.5 Ameaças à Validade

Embora tenha-se tomado cuidado para reduzir as ameaças à validade para este estudo, alguns fatores não controlados podem ter influenciado os resultados observados.

A avaliação experimental envolveu dois cenários de documentos XML. Os do-

cumentos XML contendo dados em diferentes cenários possuem três níveis, não contém atributos, e seus fragmentos variam de tamanho. De acordo com Mignet, Barbosa e Veltri (2003), os documentos XML têm uma média de 4 níveis, e, portanto, acredita-se que os documentos XML são representativos.

Todos os cenários contêm dados reais. Essa escolha leva a situações que ocorrem em um ambiente real, incluindo inconsistências nos dados. Por exemplo, as *tags* que foram utilizadas como identificadores na geração dos resultados esperados. No entanto, problemas como identificadores duplicados ou alterações nos identificadores poderiam produzir falsos positivos e falsos negativos. Para atenuar esta ameaça, uma inspeção manual foi feita para garantir a utilização de cada *tag* como identificador.

Em todas as base de dados, um fragmento do documento XML foi utilizado como treinamento para calibrar a abordagem de similaridade do XChange. Isto foi feito para descobrir o limiar de similaridade que maximiza a *F-Measure* no conjunto de treinamento (isto é, o fragmento 0), enquanto que para as outras abordagens foram utilizados os valores padrão propostos. Para atenuar esta ameaça, o fragmento 0 foi descartado, e a análise da eficácia e da eficiência foi realizada com os demais fragmentos.

## 5.6 Considerações Finais

Nossa avaliação experimental mostrou que o XChange e o X-Diff alcançaram excelentes resultados para a métrica *F-Measure* no que diz respeito ao casamento de elementos de três cenários contendo documentos XML representativos e abrangendo dados reais. É importante ressaltar que o X-Diff obteve o melhor resultado em todas as comparações. Por outro lado, o XChange obteve maior eficiência em todas as comparações, com um resultado quase 4 vezes mais rápido do que o X-Diff para o CM e quase 35 vezes mais rápido para a UC. Consequentemente, efetuando um maior número de casamentos de elementos correspondentes por segundo.

# 6 Conclusões

Apesar do XChange não ultrapassar o X-Diff em nenhuma comparação ao analisar a métrica *F-Measure*, seus resultados ficaram bem próximos. Para o CM a menor diferença foi de 0.0068 e a maior diferença de 0.0517, consequentemente o valor p = 0.9926 próximo a 1 representa tal proximidade dos resultados, não sendo estatisticamente significativa. O mesmo acontece para a UC, onde a menor diferença foi de 0.0522 e a maior diferença foi de 0.2322, não sendo estatisticamente significativa (p = 1). Ambas as abordagens apresentam alta eficácia no casamento de elementos XML.

No critério eficiência, o XChange se sobressaiu em todas as comparações. Para o CM sendo quase 4 vezes mais rápido que o X-Diff, sendo uma diferença estatisticamente significativamente ($p = 6,355 \cdot 10^{-6}$). Analogamente para a UC, o XChange foi quase 35 vezes mais rápido que o X-Diff, tento também uma diferença estatística significante ($p = 2,2 \cdot 10^{-16}$) .

## 6.1 Resultados

Uma das grandes dificuldades desde presente trabalho esteve relacionado a investigação para rastrear documentos XML reais, uma vez que a utilização de dados fictícios pode influenciar positivamente ou negativamente uma determinada abordagem. Tal complexidade pode ser atribuída a questões de sigilo e segurança dos dados. Porém existem alguns documentos disponíveis, como o caso dos portais de dados abertos dos governos[11]. Outro agravante na busca por base de dados reais para este trabalho, está relacionada a unicidade de um determinado elemento em todo o documento XML, ou seja, o mesmo deve possuir um atributo que o identifica exclusivamente, para que possa haver a análise comparando as abordagens.

Após o levantamento dos documentos XML e todo o recurso utilizado para o processamento nas abordagens, tornou-se necessário o desenvolvimento de uma ferramenta

[11]http://dados.gov.br/dataset

que auxiliasse na comparação dos resultados obtidos através de métricas utilizadas em Estatística e em Recuperação de Informação para verificar a eficiência e eficácia das abordagens em cada cenário e assim analisar qual obteve maior destaque em relação a outra. Como mencionado no Seção 4.1, o XMeasure foi desenvolvido para apoiar tal necessidade, dando a possibilidade de maior visibilidade dos dados em tabelas e gráficos.

Conforme apresentado no Capítulo 5, a avaliação experimental foi desenvolvida em cima de duas bases de dados reais, o que valoriza a avaliação experimental, dando um maior valor do que se os mesmos fossem realizados em cenários fictícios, o que poderia ocasionar melhores resultados para uma determinada abordagem.

Em relação a este trabalho, os resultados parciais foram publicados em:

- CAMPELLO, F.; PINTO, B.; TESSAROLLI, G.; OLIVEIRA, A.; OLIVEIRA, C.; OLIVEIRA, M; MURTA, L; BRAGANHOLO, V. A Similarity-based Approach to Match Elements Across Versions of XML Documents. SBBD. october. 2014;

- 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. 2018;

## 6.2 Trabalhos Futuros

Como trabalhos futuros propõe-se a utilização de outros documentos XML no que diz respeito a avaliação experimental. A avaliação experimental apresentada no Capítulo 5 envolveu apenas dois *datasets* reais, não sendo assim possível generalizar. Outro ponto importante é que para estes documentos XML, assim como na maioria dos casos não existia um esquema associado à elas. Além disso, os *datasets* utilizados foram considerados representativos no que diz respeito a profundidade (MIGNET; BARBOSA; VELTRI, 2003). Utilizar outros documentos XML, inclusive com esquema associado, diferentes distribuições relacionadas ao número de atributos e de níveis de elementos pode ser interessante para comparar os resultados obtidos neste trabalho com as novas execuções. Outra análise interessante diz respeito ao limiar de similaridade adequado para a avaliação

experimental. Neste trabalho, num primeiro momento foi executada uma calibragem para estes conjuntos de dados utilizados. Uma possibilidade é verificar a viabilidade de definir um limiar de similaridade, ou uma faixa de valores, para ser utilizado em qualquer conjunto de dados.

# Bibliografia

BAEZA-YATES, R.; RIBEIRO-NETO, B. Modern information retrieval. In: . [S.l.]: Addison Wesley, 1999.

BRAY, T. et al. Extensible markup language (XML) 1.0 (fifth edition). w3c recommendation. disponível em: <http://www.w3.org/tr/xml/>. acesso em 13 out. 2018.

COBÉNA, G.; ABDESSALEM, T.; HINNACH, Y. A comparative study for XML change detection. In: *Verso report number 221, INRIA 2002 (updated 2004), http://www.deltaxml.com/support/documents/articles-and-papers/is2004.pdf*. [S.l.: s.n.], 2004.

COBÉNA, G.; ABITEBOUL, S.; MARIAN, A. Detecting changes in XML documents. In: *International Conference on Data Engineering*. [S.l.: s.n.], 2002. p. 41–52.

GARCIA, P. A. EDX: Uma abordagem de apoio ao controle de mudanças de documentos XML. *Monografia de Graduação, UFJF*, 2012.

KHAN, A. S. *System and method for automatically generating XML schema for validating XML input documents*. [S.l.]: Google Patents, 2016. US Patent 9,286,275.

LEON, A. *A Guide to software configuration management*. [S.l.]: Artech House, Inc., 2000.

LIMA, D. et al. Towards querying implicit knowledge in XML documents. *Journal of Information and Data Management*, v. 3, n. 1, p. 51, 2012.

LINDHOLM, T. A 3-way merging algorithm for synchronizing ordered trees. *Master's thesis, Helsinki University of Technology*, 2001.

MIGNET, L.; BARBOSA, D.; VELTRI, P. The XML web: a first study. In: ACM. *Proceedings of the 12th international conference on World Wide Web*. [S.l.], 2003. p. 500–510.

MORO, M.; BRAGANHOLO, V. Desmistificando XML: da pesquisa à prática industrial. *Atualização em Informática*, p. 231–278, 2009.

MURTA, L. G. P. *Gerência de configuração no desenvolvimento baseado em componentes*. Dissertação (Mestrado) — COPPE, UFRJ, 2006.

OLIVEIRA, A. et al. An efficient similarity-based approach for comparing XML documents. *Information Systems*, Elsevier, v. 78, p. 40–57, 2018.

OLIVEIRA, A. M. *Diff Semântico de Documentos XML*. Tese (Doutorado) — Instituto de Computação, UFF, Niterói, 2016.

OLIVEIRA, A. M.; MURTA, L. G. P.; BRAGANHOLO, V. Towards semantic diff of XML documents. In: *ACM Symposium on Applied Computing (SAC), 2014, Gyeongju. ACM Symposium on Applied Computing (SAC)*. [S.l.]: New York: ACM, 2014.

OLIVEIRA, C. R. C. *XMeasure: Métricas de Avaliação Aplicadas no Controle de Mudanças de Documentos XML*. Monografia (Trabalho de Conclusão de Curso - Ciências Exatas) — Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, 2014.

PETERS, L. Change detection in XML trees: a survey. *3rd Twente Student Conference on IT*, 2005.

SANTOS, R. C. *XKeyMatch: um algoritmo semântico para detecção de diferenças entre documentos XML*. Dissertação (Mestrado) — UFPR, Curitiba - Brasil, 2006.

SANTOS, R. C.; HARA, C. S. XKeyDiff - um algoritmo semântico para detecção de mudanças entre documentos XML. In: *Rev. Eletrônica Iniciação Científica REIC 4(3)*. [S.l.: s.n.], 2004.

SONG, Y. et al. BioDIFF: an effective fast change detection algorithm for biological annotations. In: SPRINGER. *International Conference on Database Systems for Advanced Applications*. [S.l.], 2007. p. 275–287.

SUNDARAM, S.; MADRIA, S. K. A change detection system for unordered XML data using a relational model. *Data & Knowledge Engineering*, Elsevier, v. 72, p. 257–284, 2012.

WANG, Y.; DEWITT, D. J.; CAI, J.-Y. X-Diff: An effective change detection algorithm for XML documents. In: *International Conference on Data Engineering*. [S.l.: s.n.], 2003.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics bulletin*, JSTOR, v. 1, n. 6, p. 80–83, 1945.