UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Renata Junges Padilha

UM PROCESSO PARA CASAMENTO DE ESQUEMAS DE DOCUMENTOS JSON BASEADO NA ESTRUTURA E NAS INSTÂNCIAS

Santa Maria, RS
2020

Renata Junges Padilha

UM PROCESSO PARA CASAMENTO DE ESQUEMAS DE DOCUMENTOS JSON BASEADO NA ESTRUTURA E NAS INSTÂNCIAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Orientadora: Prof. Dr. Deise de Brum Saccol

Santa Maria, RS

2020

Junges Padilha, Renata

Um processo para casamento de esquemas de documentos JSON baseado na estrutura e nas instâncias / por Renata Junges Padilha. – 2020.

121 f.: il.; 30 cm.

Orientadora: Deise de Brum Saccol

Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Pós-Graduação em Ciência da Computação , RS, 2020.

1. JSON. 2. Casamento de esquemas. 3. Instâncias. 4. Hierarquia. I. de Brum Saccol, Deise. II.Um processo para casamento de esquemas de documentos JSON baseado na estrutura e nas instâncias.

E-mail: rjpadilha@inf.ufsm.br

Renata Junges Padilha

UM PROCESSO PARA CASAMENTO DE ESQUEMAS DE DOCUMENTOS JSON BASEADO NA ESTRUTURA E NAS INSTÂNCIAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Aprovado em 10 de fevereiro de 2020:

Deise de Brum Saccol, Dr. (UFSM)
(Presidente/Orientadora)

Eduardo Kessler Piveta, Dr. (UFSM)

Mirkos Ortiz Martins, Dr. (UFN)

Santa Maria, RS

2020

DEDICATÓRIA

Dedico esta dissertação ao meu pai e minha mãe... Obrigada por sempre me apoiarem e me incentivarem a continuar, independente das dificuldades. Obrigada por todo amor, carinho e confiança depositados em mim. Amo vocês mais do que tudo!

AGRADECIMENTOS

Durante esses dois anos de caminhada no Mestrado, foram diversas pessoas que me apoiaram e me incentivaram de alguma maneira, agradeço à minha família, amigos, professores e principalmente à Deus por me proporcionar força e coragem para superar todas as dificuldades.

Pai e mãe, obrigada por todo o apoio que sempre me deram durante toda a minha vida, me incentivando em todos os passos que tomei. Nada seria possível se eu não tivesse vocês em minha vida. Cada obstáculo que superei foi pensando em vocês. Não tenho palavras para agradecer tudo que fizeram por mim. Todo amor, carinho, conselhos e apoio foram essenciais para atingir meus objetivos. Amo vocês e muito obrigada do fundo do meu coração.

Ao meu marido, Eraldo, obrigada pelo companheirismo que sempre me proporcionou. Me incentivando a sempre continuar e me auxiliando a superar as dificuldades. Obrigada por tudo que fez por mim, pois todas as palavras de incentivo, carinho e a força que me deu foram essenciais nesta caminhada. Amo você.

Às minhas irmãs, Laura e Camila, e à minha vó Maria Lúcia, obrigada por todo incentivo e por não me deixarem desanimar com as dificuldades encontradas. Cada conselho foi de grande importância para mim. Amo vocês.

Aos meus professores do Mestrado, o meu muito obrigada. Cada explicação, conselho e aprendizado foram essenciais no meu desenvolvimento acadêmico. Agradeço principalmente à minha orientadora Deise pelos ensinamentos, conselhos e reciprocidade em todos os momentos. Nunca esquecerei que, independente das dificuldades, sempre me ajudou e me apoiou. Muito obrigada professora.

“Feliz aquele que transfere o que sabe e aprende o que ensina”

(CORA CORALINA)

RESUMO

UM PROCESSO PARA CASAMENTO DE ESQUEMAS DE DOCUMENTOS JSON BASEADO NA ESTRUTURA E NAS INSTÂNCIAS

AUTORA: RENATA JUNGES PADILHA
ORIENTADORA: DEISE DE BRUM SACCOL

Historicamente, o casamento de esquemas é algo muito estudado, mas que apresenta, até os dias de hoje, muitas dificuldades decorrentes de inúmeros conflitos e problemáticas. Abordagens voltadas para bancos de dados NoSQL (Not Only SQL) ainda são pouco estudadas, tendo em vista que estes apresentam esquemas implícitos em sua construção. A crescente utilização de documentos JSON (JavaScript Object Notation) mostra a importância de estudos que possam contribuir com a manipulação deste tipo de documento. Esta dissertação tem como objetivo especificar um processo para o casamento de esquemas em documentos JSON. São utilizadas técnicas de similaridade textuais (sintática, semântica e extração de radicais), algoritmo *diff*, análise da estrutura hierárquica dos elementos, além de levar em consideração as instâncias contidas nos documentos. Estas técnicas são aplicadas de forma combinada a fim de determinar se elementos de documentos JSON são equivalentes. O estudo de caso relatado nesta pesquisa mostra uma precisão e revocação de 67,05% e 82,60%, respectivamente.

Palavras-chave: JSON. Casamento de esquemas. Instâncias. Hierarquia.

ABSTRACT

A SCHEMA MATCHING PROCESS FOR JSON DOCUMENTS BASED ON STRUCTURE AND INSTANCES

AUTHOR: RENATA JUNGES PADILHA
ADVISOR: DEISE DE BRUM SACCOL

Historically, schema matching has been studied a lot, but it still presents many difficulties resulting from countless conflicts and problems. Approaches for NoSQL databases (Not Only SQL) are still poorly studied, given that they have implicit schemes in their construction. The increasing use of JSON (JavaScript Object Notation) documents shows the importance of studies that can contribute to the manipulation of this type of document. This dissertation aims to specify a process to match schemas for JSON documents. The process uses textual similarity techniques (syntactic, semantics and radical extraction), diff algorithm, analysis of the hierarchical element structure, and analysis of the instances contained in the documents. . These techniques are applied in a combined manner to determine whether elements of JSON documents are equivalent. The case study reported in this research shows accuracy and recall of 67.05% and 82.60%, respectively.

Keywords: JSON. Schema matching. Instances. Hierarchy.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE LISTAGENS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AHP	Analytic Hierarchy Process
CSV	Comma Separated Values
ISSN	International Standard Serial Number
JSON	JavaScript Object Notation
LCS	Least Common Subsumer
NoSQL	Not Only Structured Query Language
SGBD	Sistema de Gerenciamento de Banco de Dados
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

O aumento exponencial na quantidade e na variedade das fontes de dados e informações pode ser facilmente notado no decorrer dos anos. A disseminação de aplicativos móveis também contribuiu para este cenário, devido a facilidade com que a população tem acesso às informações. Dados de todos os tipos e armazenados de variadas formas são manipulados constantemente por diversas pessoas e organizações de diferentes lugares.

Com o advento dos bancos de dados NoSQL (SADALAGE; FOWLER, 2012), a manipulação de grandes volumes e variedades de dados se tornou algo eficiente, se comparado aos bancos de dados relacionais. Uma das características dos bancos de dados NoSQL é a ausência de esquemas, possibilitando que trabalhem bem com a heterogeneidade dos dados. Os variados modelos de dados existentes no paradigma NoSQL apresentam algumas características em comum. De acordo com o modelo de dados, os bancos de dados NoSQL podem ser classificados em: chave/valor, documentos, família de colunas e grafos. Conforme DB-Engines Ranking de junho de 2019 (RANKING, 2019), o banco de dados NoSQL mais utilizado é o MongoDB, que apresenta como principais formatos de documentos XML (*Extensible Markup Language*) e JSON (*JavaScript Object Notation*).

Conforme (PIERRE BOURHIS JUAN L. REUTTER; VRGOC, 2017) JSON é um formato leve que consiste em uma coleção de pares chave/valor. Surgiu no ano de 2001 e é, de acordo com (INTERNATIONAL, 2017), um formato baseado em texto que possibilita a troca de dados independente da linguagem. Por não possuir um esquema definido, apresenta uma estrutura implícita, sendo que diversas pesquisas realizaram estudos voltados para a extração desses esquemas.

Outro ponto relevante é a necessidade de se realizar a integração de diferentes esquemas oriundos de documentos JSON. A integração de esquemas, de uma forma geral, é discutida há vários anos, tendo em vista a sua importância perante a necessidade de acessar de uma forma única diversos dados espalhados em diferentes lugares, muitas vezes em bancos de dados distribuídos geograficamente.

Uma das principais etapas presentes na integração de esquemas é o casamento de esquemas. Este tem o objetivo de determinar as equivalências entre os esquemas em análise. Diversos estudos utilizam diferentes abordagens, muitas vezes de maneira combinada, para definir as equivalências entre os elementos dos esquemas.

Tendo como base estudos prévios sobre integração de esquemas e casamento de esquemas voltados tanto para esquemas relacionais quanto não relacionais, esta dissertação objetiva descrever um processo para casamento de esquemas para documentos JSON. São aplicados diferentes métodos, tais como: algoritmo *diff*, técnicas de similaridade (baseadas na extração de radicais, similaridade sintática e de sinônimos), instâncias contidas nos documentos e estrutura hierárquica dos elementos (esta última usando a definição dos caminhos para definir a equivalência entre eles).

A Listagem 1.1 ilustra um exemplo de dois documentos JSON relacionados ao mesmo domínio de aplicação (informações sobre produtos).

```
1  Documento 1                                  |Documento 2
2  {                                            | {
3    "product":                                 |   "aggregator":
4    { "_id": "5968dd23fc13ae0",                |   {"items":
5      "product_name":                          |       { "id": "62",
6          "Dextromathorphan HBr",              |         "name":
7      "supplier": "Schmitt-Weissnat",          |           "Mountain J. ashei",
8      "quantity": 261,                         |         "taxRate": "510",
9      "unit_cost": "$10.47"                    |         "manufacturer":
10 }                                            |               "Schmitt-Weissnat"}
11                                              |   }
12                                              | }
```

Listagem 1.1 – Exemplo de dois documentos JSON que possuem similaridades

Com a utilização do processo proposto nesta dissertação, a estrutura hierárquica é primeiramente analisada, sendo possível determinar que os elementos ancestrais *product* (linha 3 - documento 1) e *items* (linha 4 - documento 2) são equivalentes. Sendo assim, seus elementos descendentes podem ser testados através de diferentes abordagens (baseadas nos esquemas e nas instâncias). Por meio da utilização de diferentes medidas de similaridades, os elementos *product_name* (linha 5 - documento 1) e *name* (linha 6 - documento 2) são analisados pelos nomes dos campos. Já os elementos *supplier* (linha 7 - documento 1) e *manufacturer* (linha 9 - documento 2) são analisados pelas instâncias. Sendo assim, o casamento dos esquemas define que os pares de elementos *product_name* e *name*; *supplier* e *manufacturer* são equivalentes.

O processo como um todo é composto por fases e atividades/subatividades, além da definição de artefatos de entrada e de saída. Cada uma das fases realiza diferentes tarefas no casamento dos esquemas, como o pré-processamento dos documentos, a análise das similaridades textuais dos elementos, verificação das equivalências das instâncias e a abordagem da estrutura hierárquica dos elementos contidos nos documentos JSON.

Por meio da aplicação deste processo em um estudo de caso realizado nesta dissertação,

foi possível determinar o quanto os elementos em análise são equivalentes e se os mesmos podem ser casados. Dentre as medidas de avaliação estão a revocação e precisão, que mostraram 67,05% e 82,60%, respectivamente, para o caso estudado.

1.1 OBJETIVOS

O objetivo geral é especificar um processo para casamento de esquemas para documentos JSON, um dos tipos de fontes de dados mais utilizadas em banco de dados NoSQL orientado a documentos. A importância de realizar o casamento de esquemas (uma das principais etapas da integração de esquemas) é definir quais elementos são equivalentes oriundos de diversas fontes de dados heterogêneas. Para alcançar este objetivo principal são aplicados, principalmente, métodos voltados para medir a similaridade entre elementos presentes em documentos JSON.

1.1.1 Objetivos específicos

De forma a especificar o processo para o casamento de esquemas de documentos JSON, esta dissertação tem os seguintes objetivos específicos:

- Determinar as fases que compõem o processo para casamento de esquemas como um todo, detalhando atividades e artefatos de entrada e de saída;
- Especificar um processo que determina as equivalências entre os elementos dos esquemas, com base na estrutura hierárquica, análises de similaridades dos campos e instâncias;
- Gerar o casamento dos esquemas, determinando os elementos equivalentes;
- Avaliar os resultados obtidos no estudo de caso por meio da análise das medidas de revocação e precisão.

1.2 ORGANIZAÇÃO DO TEXTO

A organização da dissertação está dividida da seguinte maneira: o Capítulo 2 descreve a fundamentação teórica, realizando uma revisão bibliográfica de conceitos sobre integração de esquemas, heterogeneidade dos dados, fases de uma integração de esquemas, casamento de esquemas, medidas de similaridades e sistemas de casamento de esquemas. Um levantamento dos principais trabalhos relacionados à esta dissertação pode ser visto também no Capítulo 2.

As descrições gerais do processo para casamento de esquemas proposto, assim como o detalhamento de cada fase, podem ser vistas no Capítulo 3. Por meio deste detalhamento é possível obter um melhor entendimento de como funciona o processo. Algoritmos também são utilizados para auxiliar nos objetivos propostos por cada atividade.

O Capítulo 4 relata um estudo de caso realizado com documentos JSON oriundos de bibliotecas digitais. O objetivo é demonstrar como funciona todas as fases do processo de um modo prático. As atividades que foram implementadas na linguagem Java, podem ser também vistas no Capítulo 4. Além disso, o Capítulo 4 relata a validação do processo, por meio dos resultados obtidos. A conclusão e trabalhos futuros são descritos no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

Com o objetivo de contextualizar os temas importantes referentes à temática desta pesquisa, o presente capítulo descreve a fundamentação teórica relacionada ao casamento de esquemas. Diferentes abordagens são explanadas como:

- A integração de esquemas (Seção 2.1), destacando as heterogeneidades existentes entre os dados e as fases tradicionalmente encontradas no processo de integrar esquemas.
- O casamento de esquemas (*schema matching*) (Seção 2.2) descreve as diferentes classificações consagradas na literatura para realizar a correspondência entre diferentes esquemas, assim como os componentes de um sistema de casamento de esquemas tradicional.
- O casamento de *strings* (Seção 2.3) é descrito com o objetivo de mostrar sua importância no momento de realizar o casamento de esquemas, assim como as medidas de similaridade.
- O método AHP (Seção 2.4) é descrito brevemente, com o intuito de explicar os pesos utilizados em cada técnica de medida de similaridade presente no processo proposto.
- Os trabalhos relacionados (Seção 2.5), que trouxeram contribuições de diferentes pesquisas para esta dissertação, sendo tratados temas relacionados à extração de esquemas em documentos JSON, assim como o casamento de esquemas para XML.

2.1 INTEGRAÇÃO DE ESQUEMAS

Na história da computação, a integração de esquemas é um assunto estudado há muitos anos, devido às necessidades de se unificar diferentes dados, muitas vezes distribuídos em lugares diferentes. Por meio da integração de esquemas é possível unificar esquemas heterogêneos em um esquema único. As técnicas utilizadas têm o objetivo de identificar correspondências entre diferentes esquemas. Desta forma, as aplicações ou usuários submetem consultas por meio de um esquema mediado e o mapeamento semântico entre este e as fontes de dados é realizado.

Várias pesquisas foram exploradas para modelos convencionais de bancos de dados (ELMASRI; NAVATHE, 2011). No entanto, a popularização do paradigma NoSQL e documentos JSON ainda mantém como relevante o problema da integração de esquemas.

Esta seção realiza uma revisão bibliográfica de abordagens utilizadas no processo de integração, como os diferentes tipos de heterogeneidade dos dados, os conflitos presentes na integração, as fases da integração de esquemas e outros temas importantes que trazem uma elucidação referente a integração de esquemas.

2.1.1 Heterogeneidade dos dados

Antes de se falar em integração de esquemas, é importante frisar as diferenças que existem entre as fontes de dados, visto que se tornam problemas no momento de integrar diferentes esquemas. Para (DIAS, 2006), a heterogeneidade já é bastante explorada pela comunidade de banco de dados distribuídos e ela pode ser subdivida em:

Heterogeneidade entre SGBDs (Sistemas de Gerenciamento de Banco de Dados): está ligada ao fato dos diferentes modelos de dados que uma empresa pode possuir, levando em consideração os requisitos que devem ser atendidos. Essa heterogeneidade se dá pelo fato de que cada SGBD possui seu modelo de dados para estruturar seu banco de dados da forma mais adequada.

Heterogeneidade estrutural: leva em consideração as diferenças que existem nos esquemas dos dados, ou seja, diferenças estruturais devido a forma como os dados são modelados, como por exemplo, os elementos que são representados de formas diferentes, mas dizem respeito ao mesmo objeto do mundo real. Outros exemplos incluem: um elemento é tratado como entidade por um esquema e como atributo pelo outro; diferentes chaves para um mesmo elemento; relacionamentos diferentes entre elementos; um elemento possui um atributo simples, mas no outro esquema é representado por um conjunto de atributos (atributo composto), entre outros.

Heterogeneidade semântica: está diretamente relacionada ao modo como as informações são interpretadas em diferentes situações, ou seja, relaciona-se ao significado dos dados. A análise realizada e atribuída aos dados pode ser diferente devido às percepções das pessoas envolvidas na modelagem de um esquema de banco de dados. Por exemplo, dois atributos com nomes iguais (por exemplo “custo_curso”), mas com finalidades diferentes – um deles se refere ao custo de um curso, e o outro leva em consideração taxas adicionais. Esse tipo de cenário pode acarretar equívocos se tratados como elementos iguais no momento de integrar (IPPOLITO, 2012).

Heterogeneidade sintática: é tratada de uma forma mais detalhada por (IPPOLITO,

2012), trazendo algumas subdivisões quanto ao tipo de heterogeneidade presente nos esquemas de bancos de dados. O conceito de sintático está relacionado à nomenclatura presente nos nomes dentro de um esquema de bancos de dados.

Esta subdivisão, citada anteriormente, pode ser descrita como: sinonímia (nomes diferentes com significados iguais); homonímia (nomes iguais com significados diferentes); hiperonímia (relação hierárquica de significado que uma palavra superior estabelece com uma palavra inferior); e hiponímia (representa uma parte de um todo). Um exemplo simples de hiperonímia e hiponímia é que *país* é hiperônimo de *Brasil* e *Brasil* é hipônimo de *país*.

2.1.2 Fases da integração de esquemas

Integrar esquemas é necessário em qualquer tipo de integração, ou seja, não importa qual método será utilizado para integrar dados, a integração de esquemas precisa ser realizada. Por meio da integração de esquemas é possível unificar esquemas heterogêneos em um esquema único. As técnicas utilizadas têm o objetivo de identificar correspondências entre diferentes esquemas.

As aplicações ou usuários submetem consultas por meio de um esquema mediado e o mapeamento semântico entre este e as fontes de dados é realizado por meio do detalhamento das fontes de dados. Tradicionalmente, o processo de integração de esquemas é dividido em quatro fases: pré-integração, comparação de esquemas, unificação de esquemas e *merging* e reestruturação.

Fase 1: Pré-integração: De acordo com (C. BATINI; NAVATHE, 1986) a fase de pré-integração analisa os esquemas antes da integração, cuidando da organização, como quais esquemas serão integrados e ordem de integração. Informações adicionais, que são importantes para a integração, também são relatadas nesta fase. Um exemplo da aplicação da política de integração, que realiza a organização da integração, é determinar quais esquemas devem ter preferência sobre outros.

Quatro estratégias para escolher a ordem de integração entre os esquemas, além da quantidade de esquemas em cada fase, são descritas por (C. BATINI; NAVATHE, 1986). A Figura 1 ilustra as estratégias, sendo que os esquemas são representados pelas folhas da árvore e o esquema integrado é representado pelo nó raiz. As estratégias são divididas em duas: *binárias* e *n-árias*.

- **Binária:** dois esquemas podem ser integrados por vez, sendo que podem variar em *ladder*

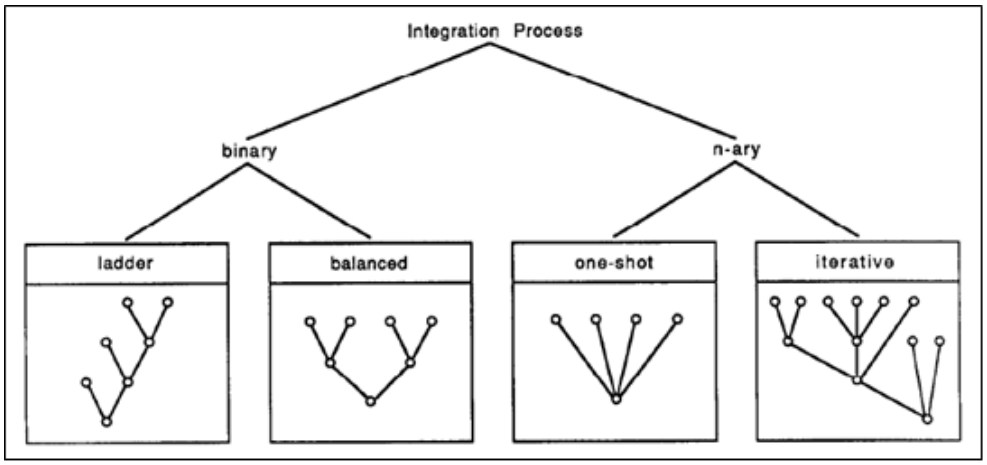
(em forma de passo a passo) e *balanced* (de forma balanceada e/ou equilibrada).

  - *Ladder*: a cada etapa um novo esquema é integrado com um resultado intermediário, ou seja, esquemas intermediários são criados para integrar com esquemas subsequentes;
  - *Balanced*: inicialmente os esquemas são divididos em pares e seguem sendo integrados de maneira simétrica, isto é, cada esquema é integrado um ao outro, criando um esquema intermediário para integração com outros esquemas intermediários;

- **N-árias:** integra N esquemas a cada etapa, sendo mais de dois a cada iteração ($N > 2$). Podem ser *one-shot* (um único passo) e *iterative* (de forma iterativa).

  - *One-shot*: de forma geral integra todos os esquemas em uma única vez, gerando o esquema conceitual global depois de uma iteração.
  - *Iterative*: integra os esquemas de maneira iterativa, sendo que o número de esquemas pode variar dependendo da preferência do integrador.

Figura 1 – Estratégias de integração de esquemas



Fonte: (C. BATINI; NAVATHE, 1986)

Dentre estas estratégias, a n-ária *one-shot* foi adotada para realizar o casamento dos esquemas. Conforme levantamento realizado por (C. BATINI; NAVATHE, 1986), a estratégia n-ária *one-shot* possui como vantagens a diminuição no número de análises adicionais nos esquemas, pois como não possui esquemas intermediários, a análise é realiza em uma única vez. Além disso, aumenta a abrangência das análises dos elementos.

Fase 2: Comparação de esquemas: A fase de comparação de esquemas tem como objetivo analisar e comparar os diferentes esquemas para identificar equivalências, assim como conflitos decorrentes de esquemas heterogêneos. Essas identificações devem ser bem definidas para que a fase seguinte (unificação de esquemas) possa unificar os conflitos encontrados. Para (C. BATINI; NAVATHE, 1986) os conflitos entre conceitos que podem ser encontrados são divididos em conflitos estruturais e conflitos de nomenclatura. O primeiro leva em consideração as diferentes maneiras de se representar os conceitos, e o segundo está relacionado aos conflitos dos nomes nos diferentes esquemas.

Os conflitos de nomenclatura podem ser divididos em:

- **Conflitos de homonímia**: um mesmo nome para designar diferentes conceitos em diferentes esquemas;
- **Conflitos de sinonímia**: mesmo conceito pode ser descrito com diferentes nomes em diferentes esquemas;

Em se tratando de conflitos estruturais, estes podem ser divididos em:

- **Conflitos de tipo**: ocorre quando um mesmo conceito é definido de maneiras diferentes em determinados esquemas. Um exemplo é representar um conceito como entidade em um esquema e no outro como atributo.
- **Conflitos de dependência**: diferentes dependências entre conceitos de diferentes esquemas podem ocorrer. Por exemplo, em um esquema a relação entre entidades pode ser *1:n* e em outra *m:n*.
- **Conflitos de chave identificadora**: diferentes chaves são atribuídas ao mesmo conceito em diferentes esquemas.
- **Conflitos comportamentais**: diferentes regras de comportamento dos dados podem ser diferentes em cada esquema. Em um esquema, por exemplo, uma entidade quando passa a não possuir mais dados é deletada, mas em outro não é aplicado esta regra.

A fase de comparação de esquemas é uma das fases mais importantes no processo de integração de esquemas, pois é nesta fase que se identificam os conflitos existentes, independentemente do tipo. Analisar os dados semanticamente por meio de análises de propriedade das entidades, atributos e relacionamentos de um esquema, faz parte desta fase.

O casamento de esquemas é realizado nesta fase tendo como objetivo realizar o mapeamento entre os conceitos que possuem equivalência entre esquemas diferentes. Esta similaridade pode ser analisada por meio da estrutura das fontes de dados, combinação de métodos, entre outros (RAHM; BERNSTEIN, 2001).

Fase 3: Unificação de esquemas: A fase de unificação tem como objetivo resolver os conflitos encontrados na fase de comparação de esquemas. Conflitos de equivalência, compatibilidade e incompatibilidade são tratados para que a fase de *merging* e reestruturação seja executada. Dentre as atividades realizadas nessa fase estão: transformar tipos, reestruturar elementos e renomear nomes. Embora existam métodos automáticos para resolver os conflitos, nem todos conseguem ser solucionados, sendo necessária a intervenção de um profissional em algumas vezes.

Os conflitos de equivalência, compatibilidade e incompatibilidade estão relacionados a ideia de conceitos comuns, ou seja, um mesmo conceito pode ter diferentes representações em diferentes esquemas. Os conflitos de equivalência dizem respeito aos conceitos que não possuem inconsistência entre os esquemas, sem diferenças de entendimento, mas a representação é feita por construtores diferentes.

Os conflitos de equivalência podem ser: (a) comportamental, (b) mapeamento e (c) transformacional. Em (a) dado uma consulta aos dados, um conceito é dito equivalente - comportamental se a busca pelo conceito em uma fonte é também encontrada na outra. Para (b) se as instâncias de um conceito possuírem um mapeamento um-para-um em outro conceito, este é chamado de conceito mapeamento-equivalente. E em (c) para um conceito ser equivalente transformacional, ele tem que ser obtido a partir de outro conceito, sem perder suas equivalências, por meio do emprego de transformações.

Outro tipo de conflito, denominado de conflito de compatibilidade, leva em consideração a percepção dos projetistas, tendo em vista que dois conceitos podem não serem idênticos e nem equivalentes, mas são compatíveis na percepção deles. Já os conceitos incompatíveis apresentam inconsistências em seus elementos, ou seja, possuem diferenças entre eles.

Fase 4: *Merging* e reestruturação: A integração de esquemas, propriamente dita, é realizada na fase de *merging* e reestruturação. Reestruturar esquemas intermediários é um dos objetivos, tendo como finalidade satisfazer os seguintes critérios: completude e correção, minimalidade e compreensibilidade (C. BATINI; NAVATHE, 1986).

A completude e correção estão relacionados ao fato de que o esquema global deve con-

ter, de forma coerente, todos os conceitos presentes nos esquemas locais de entrada. A minimalidade remete a ideia de mínimo, ou seja, um mesmo conceito contido em vários esquemas locais, deve ser representado uma única vez no esquema global. E o último critério é a compreensibilidade, onde tanto para o projetista quanto para o usuário final deve ser possível se ter um entendimento adequado e/ou facilitado do esquema integrado.

2.2 CASAMENTO DE ESQUEMAS

As correspondências semânticas entre os esquemas especificam como os elementos de um esquema fonte e o esquema mediado correspondem semanticamente um com outro. Por exemplo, o atributo *"name"* em um esquema corresponde ao atributo *"title"* em outro. Criar esse tipo de correspondência não é nada simples na aplicação da integração de dados, devido a inúmeros fatores como o entendimento semântico dos esquemas, que muitas vezes é distribuído entre muitas pessoas e especialistas.

O casamento de esquemas (*schema matching*) é uma operação fundamental para encontrar correspondências entre os esquemas que se deseja integrar. Geralmente o desenvolvimento do casamento de esquemas é realizado manualmente, acarretando inconsistências, além limitar o processo e ser um tanto trabalhoso para o especialista. Pesquisas têm proposto técnicas para realizar o casamento de esquemas de maneira semiautomática.

2.2.1 Visão geral

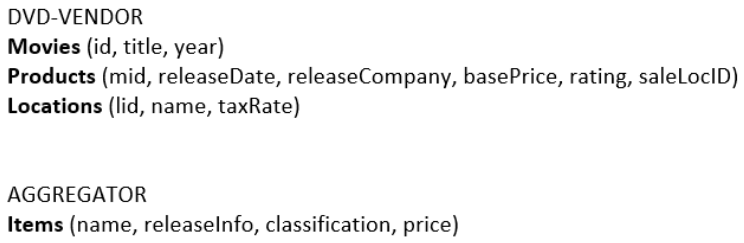
A importância do casamento de esquemas não é limitada à apenas a integração de esquemas, mas tem um propósito mais amplo, sendo utilizado em diferentes domínios de aplicação. Pode-se citar *Data warehouses*, onde operações de correspondência são úteis para projetar transformações de dados do formato de origem no formato adequado.

Outro domínio é o *E-commerce*, que utiliza o casamento de esquemas com o intuito de traduzir mensagens. A troca de mensagens entre parceiros comerciais para transações comerciais podem ser de diferentes formatos ou até mesmo podem usar esquemas de mensagens diferentes.

Para elucidar como funciona a correspondência semântica, considere o exemplo da Figura 2. No primeiro esquema as tabelas **Movies**, **Products** e **Locations** descrevem, respectivamente, os detalhes dos filmes, os produtos de DVD e os locais de venda. Já no segundo esquema

(*AGGREGATOR*) é utilizado para um *site* de compras, então a tabela **Items** descreve os detalhes que são importantes para a aplicação, como o nome do filme e o preço. (ANHAI DOAN; IVES, 2012)

Figura 2 – Exemplo de dois esquemas de banco de dados



Fonte: (ANHAI DOAN; IVES, 2012)

As correspondências descrevem as relações entre os elementos de dois esquemas, que podem ser do tipo: *um-para-um*, *um-para-muitos*, *muitos-para-um* e *muitos-para-muitos*. A correspondência *um-para-um* relaciona um elemento de um esquema com outro elemento de um segundo esquema, conforme o exemplo a seguir.

Movies.title $\approx$ **Items**.name

Movies.year $\approx$ **Items**.year

Products.rating $\approx$ **Items**.classification

Na correspondência *um-para-muitos* é relacionado um elemento de um esquema com vários elementos de um segundo esquema. Já as correspondências *muitos-para-um* realiza a relação ao contrário, ou seja, vários elementos de um esquema com um elemento de um segundo esquema. E a correspondência *muitos-para-muitos* relaciona múltiplos elementos de um esquema com múltiplos elementos de outro esquema.

Quando se trata de casamento de esquemas existem inúmeros desafios que dificultam o processo devido às diversas heterogeneidades encontradas nos esquemas. Entre os diversos desafios pode-se encontrar:

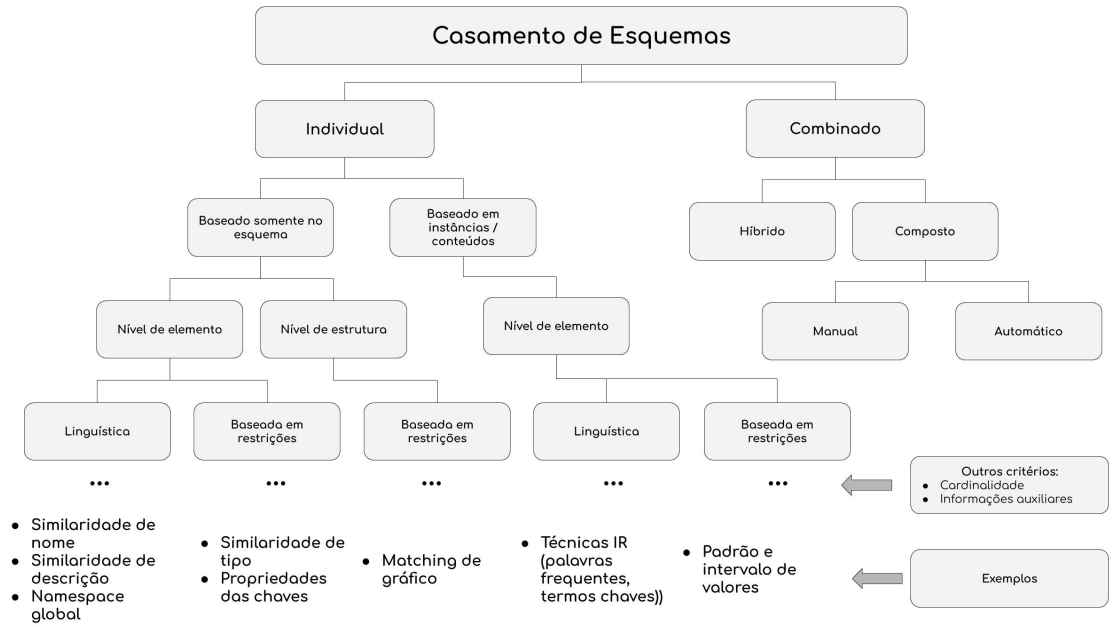
- Nomes de tabelas e atributos podem ser escritos de maneiras diferentes, mas dizem respeito ao mesmo conceito;
- Vários atributos de um esquema correspondem à um único atributo no outro esquema;
- Organização estrutural dos esquemas são diferentes;

- Níveis e objetivos diferentes em cada esquema.

Isso ocorre principalmente pelo motivo de que diferentes pessoas criam esquemas voltados para propósitos diferentes. Além disso, cada um possui uma percepção distinta sobre um mesmo conceito. É difícil resolver essas heterogeneidades semânticas, pois muitas vezes a intenção do esquema não é totalmente claro, as especificações e dicas, contidas nos esquemas, podem não ser confiáveis e também a decisão de escolher as correspondências é, muitas vezes, conflitante entre os especialistas.

Conforme (RAHM; BERNSTEIN, 2001) a abordagem de casamento de esquemas pode ser classificada de diversas maneiras, conforme a Figura 3. Dependendo do domínio de aplicação ou tipo de esquema, o casamento de esquemas pode ser implementado de maneiras diferentes: *matchers individuais* e a *combinação de matchers*.

Figura 3 – Classificação da abordagem de casamento de esquemas (*schema matching*)



Fonte: (RAHM; BERNSTEIN, 2001) - Adaptado

- ***Matcher individual***: o casamento pode ser baseado somente nas informações contidas nos *esquemas* ou também nas *instâncias*, que leva em consideração o conteúdo dos dados. A diferença entre *nível de elemento* e *nível de estrutura* diz respeito, respectivamente, a elementos de esquemas individuais (como atributos), e as estruturas mais complexas (combinação de elementos).

O método baseado nos nomes e elementos textuais nos esquemas é classificado como *baseado em linguística*, e o *baseado em restrições* está associado à chaves e relações entre elementos. Alguns critérios adicionais como *cardinalidade matching* e *informações auxiliares* podem ser descritas como: correspondência entre elementos de esquemas diferentes (casos *1:1, 1:n, n:1, n:m*); e uso de informações adicionais como dicionários e entrada de usuários.

- ***Combinação de matchers***: o casamento pode ser: *híbrido* ou *composto*. O método *híbrido* manipula vários critérios de correspondência ao mesmo tempo, ou seja, combina dois ou mais métodos e os processa simultaneamente, sendo mais eficiente do que *matcher individual*. Já o *composto* executa um método para depois executar outro, de uma forma sequencial, combinando os resultados no final.

Algumas ferramentas para realizar o casamento de esquemas são encontradas como em (MASSMANN, 2001), (PETER MORK LEN SELIGMAN; WOLF, 2006) e (J. MADHAVAN; RAHM, 2001). Elas não são totalmente automáticas, devido às dificuldades encontradas relacionadas à inúmeros fatores, como os desafios quanto à semântica.

A realização de um casamento de esquemas, na maioria das vezes, se torna algo difícil por diversas razões, não só em questões de heterogeneidade de dados, mas também pelo fato de as estruturas dos bancos de dados serem distintas e de diferentes projetistas modelarem os bancos de dados conforme suas percepções.

O processo para casamento de esquemas de diferentes bancos de dados é essencial e é a condição para a realização da integração dos esquemas e consequentemente a integração dos dados, pois só a partir de um esquema único é possível realizar a modelagem dos dados de uma forma unificada.

Abordagens importantes do casamento de esquemas que são necessárias ao entendimento desta dissertação, como a consideração de instâncias e a combinação de *matchers* são descritas a seguir.

- Abordagens a nível de instâncias: Quando se trata de dados semiestruturados, como XML ou JSON, as informações contidas nos esquemas, muitas vezes não permitem ter um conhecimento sobre o que se trata determinados elementos. Os dados no nível da instância podem fornecer informações importantes sobre o conteúdo e o significado dos elementos do esquema.

Considerar a correspondência a nível da instância pode ser usado não apenas quando não se tem informações do esquema, mas também pode ser útil para descobrir interpretações equivocadas das informações dos esquemas. Para elementos textuais, aplicações da abordagem a nível de instância podem ser utilizadas por meio da extração de palavras-chave, frequência e combinação de palavras. (RAHM; BERNSTEIN, 2001)

O principal benefício da avaliação de instâncias é uma caracterização precisa do conteúdo real dos elementos do esquema. Várias abordagens foram propostas para executar uma correspondência ou classificação de instância, como as baseadas em regras, redes neurais e técnicas de aprendizado de máquina (*machine learning*). (RAHM; BERNSTEIN, 2001)

- Abordagens com combinação de *matchers*: Casamentos de esquemas que utilizem apenas um tipo de abordagem provavelmente consiga menos candidatos à correspondências do que um que utilize diversos tipos de abordagens. Como visto anteriormente, a combinação de *matchers* pode ser *híbrido* ou *composto*, sendo que este combina os resultados de correspondências executados independentemente, e aquele que utiliza vários critérios de correspondência ao mesmo tempo.

  Conforme (RAHM; BERNSTEIN, 2001) os *matchers* híbridos utilizam diferentes abordagens de correspondências de esquemas para determinar candidatos correspondentes. Para isto pode-se usar, por exemplo, dicionário de sinônimos com compatibilidade de tipos de dados, acarretando melhores resultados do que a execução separada destas abordagens. Outro exemplo são as correspondências a nível de estrutura, que podem ser melhorados pela combinação de correspondências de nomes.

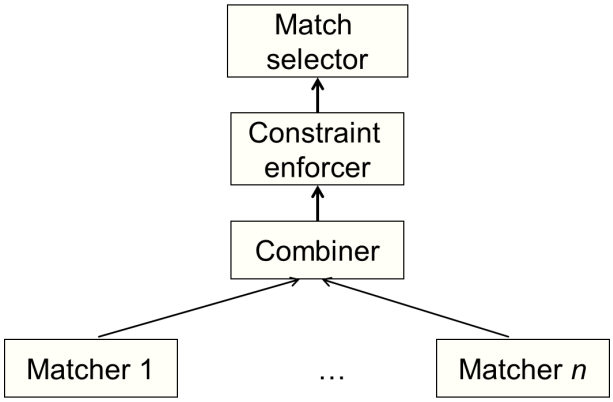
  Outra abordagem que combina diferentes maneiras de correspondências de elementos são os *matchers* compostos, que foi escolhido para esta dissertação. A diferença entre o composto e o híbrido é que o composto realiza as correspondências de forma autônoma (podendo incluir híbridos) e junta os resultados obtidos. A escolha das abordagens de correspondências a serem utilizadas, assim como a ordem de execução, podem ser selecionadas conforme critérios pré-estabelecidos, tornando o *matcher* composto mais flexível que o híbrido.

2.2.2 Sistemas de Casamento de Esquemas

Heurísticas diferentes podem fazer diferentes correspondências entre os esquemas, como o conjunto de heurísticas que examinam as similaridades entre os nomes dos elementos dos esquemas e aqueles que levam em consideração os valores dos dados. Além disso existe a combinação dessas heurísticas para melhorar a precisão do casamento de esquemas.

De acordo com (ANHAI DOAN; IVES, 2012) a arquitetura de casamento de esquemas é motivada pela investigação das variadas heurísticas que existem para maximizar a precisão das correspondências dos esquemas. A Figura 4 ilustra os componentes de um típico casamento de esquemas. Cada um dos componentes é descrito a seguir.

Figura 4 – Componentes de um típico sistema de casamento de esquemas



Fonte: (ANHAI DOAN; IVES, 2012)

- ***Matchers* (esquemas → matriz de similaridade):** Cada *matcher* é baseado em um conjunto de heurísticas, sendo que podem explorar diferentes tipos de correspondências (nomes dos elementos e/ou valores dos dados, por exemplo). Um *matcher* considera dois esquemas S e T e gera uma matriz de similaridade, atribuindo um número entre 0 (zero) e 1 (um) para cada par de elementos s de S e t de T.

- ***Combiner* (matriz χ ... matriz χ → matriz):** Um combinador (*combiner*) mescla as matrizes de similaridades geradas pelos *matchers*, gerando uma única matriz. Entre as operações que podem ser feitas pelos combinadores é a média, máximo, mínimo ou soma de pesos de *scores* de similaridades. Tipos mais complexos utilizam outras técnicas, como *machine learning*.

- ***Constraint enforcer* (matriz χ restrições $\rightarrow$ matriz):** Por meio da aplicação de restrições aos candidatos *matches*, o *constraint enforcer* (aplicador de restrições) transforma a matriz de similaridade, gerada pelo *combiner*, em outra matriz que representa melhor as similaridades verdadeiras.

- ***Match selector* (matriz $\rightarrow$ *matches*):** Este componente gera as correspondências (*matches*) entre os elementos vindos da matriz de similaridade gerada no *constraint enforcer*. Para isto, utiliza um limiar onde todos os pares de elementos são testados e os que obtiverem um *score* maior que um determinado limiar pode ser considerado correspondentes.

2.2.2.1 Matchers

Dado dois esquemas *S* e *T*, o *matcher* gera uma matriz de similaridade que atribui para cada par de elementos *s* de *S* e *t* de *T* um número entre 0 e 1, para determinar se *s* e *t* correspondem. Outras informações adicionais podem ser utilizadas como instâncias de dados e descrições textuais.

São inúmeras as técnicas para determinar as correspondências entre os elementos de diferentes esquemas, cada uma se baseando em conjuntos heurísticos distintos. Entre os mais básicos pode-se citar os *matchers* baseados em nomes e os *matchers* baseados em instâncias. Estes são descritos a seguir.

- ***Matchers* baseados em nomes:** Este tipo de correspondência entre nomes é bem comum. Quando não há conflitos semânticos, este tipo de *matcher* tem boas contribuições, mas nem sempre os nomes são escritos de maneira igual ou semelhante, assim como seus significados podem ser divergentes. As técnicas de casamento de *strings* (Seção 2.3) podem ser usadas na correspondência de nomes, como a distância de edição, medida de *Jaccard* ou medida de *Jaro & Winkler* entre outras.

  Alguns tipos de normalização nos nomes dos elementos antes de aplicar as técnicas são importantes, como dividir os nomes de acordo com certos delimitadores, expandir abreviaturas e acrônimos, assim como remover artigos e preposições.

- ***Matchers* baseados em instâncias:** Utilizar os valores dos dados (quando os mesmos estão disponíveis) é uma boa opção, pois geralmente apresenta o significado dos elementos do esquema. Existem diferentes técnicas para corresponder elementos baseando-se

nas instâncias dos dados: (a) criação de reorganizadores; (b) medição da sobreposição de valores; e (c) classificadores.

(a) Por meio da **criação de reorganizadores**, que aplicam dicionários ou regras, é possível organizar os valores de dados em diferentes conjuntos de atributos.

(b) A técnica de **medição da sobreposição de valores**, utilizada nesta dissertação, se baseia nos valores que aparecem nos dois elementos dos esquemas em comparação. É comum a utilização da medida de *Jaccard* para este propósito. Além disso, é importante frisar que esta técnica se aplica a domínios limitados, como por exemplo classificações de filmes, nomes de filmes ou nomes de países.

(c) Geralmente as técnicas de **classificadores** utilizadas são *Naive Bayes*, árvore de decisão, regras de aprendizagem, entre outros. Este tipo de técnica constrói classificadores em um esquema e utiliza para classificar os elementos do outro esquema. Esses classificadores são treinados a partir das instâncias com exemplos positivos e negativos, que, quando aplicados, produzem um número no intervalo de 0 e 1, determinando se as instâncias de um correspondem às instâncias do outro esquema.

2.2.2.2 Combinar Matchs

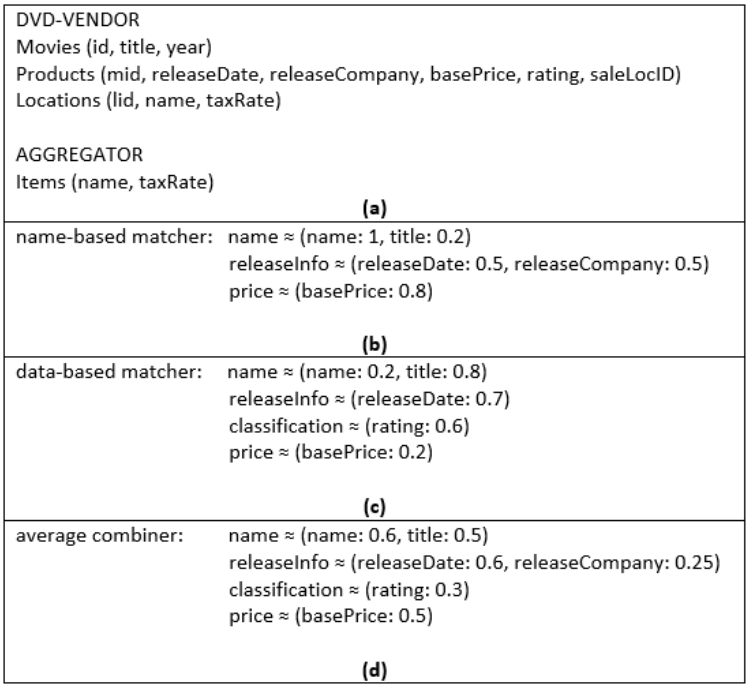
Um combinador (*combiner*) mescla as matrizes de similaridades geradas anteriormente pelos *matchers* em uma única matriz. Combinadores podem ser de média, mínimo e máximo calculando os *scores* entre os elementos.

O combinador de média é utilizado quando não há confiança nas correspondências geradas nos *matchers*. Os combinadores máximo e mínimo funcionam da seguinte maneira: calculam o *score* entre os elementos para ser o mínimo (ou o máximo) dos *scores* produzidos pelos *matchers*. Quando há um sinal forte de que uma correspondência está correta, ou seja, possui um *score* alto, é mais aconselhável usar o combinador máximo.

A Figura 5 ilustra um exemplo de dois esquemas em (a) relacionados à filmes. Para realizar o casamento entre eles pode-se aplicar diferentes *matchers* como os baseados em nomes para normaliza os nomes dos elementos no esquema. De uma forma compacta, em (b) e (c) são mostradas as matrizes de similaridades produzidas pelos *matchers* correspondentes (baseado em nome e baseado em dados, respectivamente).

O uso de *matchers* pode ser aplicado de diferentes maneiras, como as medidas de similaridades baseadas em conjunto, observando se os nomes ou os dados são avaliados. Nos baseados em dados pode ser aplicado medidas de sobreposição de dados com a medida de *Jaccard*, por exemplo.

Figura 5 – Exemplo da aplicação dos componentes de um casamento de esquemas. (a) Dois esquemas; (b)-(c) Uso de diferentes *matchers* gerando diferentes matrizes de similaridades; (d) Matriz de similaridade combinada



Fonte: (ANHAI DOAN; IVES, 2012)

É possível observar que valores diferentes foram obtidos para cada um dos *matchers*, pois um é aplicado sobre os nomes e o outro sobre os valores dos dados. O primeiro grupo de correspondência baseada em dados *name ≈ (name: 0.2, title: 0.8)* obteve valores diferentes dos baseados em nomes *name ≈ (name: 1, title: 0.2)*.

Tendo em vista que o atributo *name* do esquema (*AGGREGATOR)* se refere à títulos de DVD, o mesmo possui poucos valores em comum com *name* do esquema *DVD-VENDOR* que diz respeito aos nomes de locais de venda. Sendo assim, a similaridade entre *name* e *title* é maior no *matcher* baseado em dados, pois os mesmos compartilham os mesmos dados. Em (d) é ilustrado a matriz de similaridade aplicada pelo combinador de média, que mescla as matrizes baseadas em nomes e as baseadas em dados.

2.2.2.3 Aplicar restrições

Geralmente o especialista tem conhecimento sobre o domínio que está realizando o casamento dos esquemas, e as restrições de integridade de domínio são utilizadas para retirar certas combinações de *matchs* (oriundas do combinador) que satisfaçam as restrições do domínio. Aplicar um tipo de restrição para determinar se os elementos dos esquemas correspondem é uma tarefa desafiadora, por questões de decisões de quais restrições utilizar, grande número de combinações para analisar, entre outros.

Existem restrições de integridade de domínio que são mais difíceis e outras mais a amenas. As difíceis são aquelas que devem ser aplicadas, ou seja, não podem ser ignoradas, enquanto que as outras restrições são de natureza mais heurística, podendo ser ignoradas. Um custo pode ser aplicado para cada restrição, sendo que para as restrições difíceis o custo é ∞ e para as outras pode ser qualquer número positivo.

Quando se trata de pesquisar espaço de combinações *matchs*, existem diversos algoritmos que podem aplicar restrições nas matrizes de similaridades geradas pelo *combinador*. A seguir dois algoritmos são destacados. O primeiro é denominado de *A* Search* que assegura encontrar uma solução ideal para as correspondências, mas possui um custo computacional muito elevado. Este algoritmo possui como entrada as restrições de domínio pré-estabelecidas e a matriz de similaridade adquirida no *combinador*. Por meio da busca por correspondências possíveis, o algoritmo retorna correspondências com custos mais baixos. Seu objetivo é pesquisar um estado ideal dentro de um conjunto de estados.

O segundo algoritmo dissemina restrições localmente, ou seja, começando pelos elementos de um esquema para os seus vizinhos até atingir um ponto fixo. A presente dissertação não faz uso de domínios de restrições, sendo assim, mais detalhes sobre a aplicação de restrições podem ser vistos em (ANHAI DOAN; IVES, 2012).

2.2.2.4 Selecionar os Matchs

O último componente de um sistema de casamento de esquemas possui como entrada a matriz de similaridade gerada pelo *combinador* e melhorada pelo *aplicador de restrições*. Seu objetivo é produzir *matchs* a partir da matriz de similaridade.

Para que isto ocorra, um limiar é aplicado. Todos os pares de elementos do esquema são testados e aqueles que possuem *scores* iguais ou superiores ao limiar são considerados

correspondentes. Por exemplo, a matriz de similaridade:

name ≈ *(title: 0.5)*

releaseInfo ≈ *(releaseDate: 0.6)*

classification ≈ *(rating: 0.3)*

price ≈ *(basePrice: 0.5)*

Se um limiar de *0.5* for estabelecido, então os *matchs* produzidos são: *name* ≈ *(title: 0.5)*, *releaseInfo* ≈ *(releaseDate: 0.6)* e *price* ≈ *(basePrice: 0.5)*. Existem estratégias que geram uma seleção de melhores *matchs* para que o especialista possa decidir entre os dois melhores conjuntos de *matchs*, por exemplo.

2.3 CASAMENTO DE *STRINGS*

De acordo com (ANHAI DOAN; IVES, 2012) o casamento de *strings* tem um papel importante nas tarefas de integração de dados, como o casamento de esquemas, casamento de dados e extração de informações. O casamento de *strings* tenta encontrar *strings* que correspondam à alguma entidade do mundo real.

Esta seção descreve diferentes formas de encontrar equivalências entre *strings*. O estudo prévio sobre algumas medidas de similaridade é relevante para se obter conhecimento das técnicas utilizadas nesta pesquisa.

2.3.1 Medidas de similaridade

As técnicas utilizadas nesta dissertação dizem respeito às similaridades encontradas entre palavras considerando o radical, caracteres e conhecimento. Medidas de similaridades textuais podem ser descritas como uma mineração de texto, onde encontrar equivalências entre diferentes palavras é o início para descobrir similaridades em sentenças, parágrafos e documentos. Existem diferentes abordagens para medir as similaridades textuais como: (a) baseadas em sequência; (b) baseado em conjuntos; (c) híbrido; (d) fonética; (e) baseado em conhecimento; e (f) baseado em radicais.

De acordo com (DIDIK DWI PRASETYA; HIRASHIMA, 2018) abordagens semânticas e léxicas são essenciais na descoberta de similaridades textuais. A semelhança lexical trata das similaridades encontradas em sequências de caracteres quando são comparadas. Um *score*, ou pontuação, é atribuído conforme o grau de similaridade encontrada entre as sequências,

determinando palavras lexicalmente idênticas (valor 1), ou não idênticas (valor 0).

Já a abordagem semântica leva em consideração o significado do contexto entre textos e documentos. Muitas vezes palavras que possuem similaridade lexical, ou seja, são escritos de maneiras bem parecidas, podem não serem equivalentes devido ao seu significado. Um exemplo é que o par de palavras *book* e *cook*, que possuem similaridade lexical alta, mas não são semanticamente semelhantes.

(a) Baseadas em sequência: A medida de similaridade baseada em sequência considera as *strings* como uma sequência de caracteres, onde é calculado a transformação de uma *string* em outra. Para obter esse tipo de similaridade é levado em consideração a distância de edição entre duas cadeias de caracteres. Essa distância de edição inclui as inserções, exclusões e substituições entre os caracteres. Ou seja, é realizado o cálculo para determinar se duas sequências de caracteres são similares, tendo como critério se o número mínimo de operações de distância de edição (por exemplo, operações necessárias para transformar um caractere em outro) for menor que um limiar específico.

Entre as vantagens de se utilizar este tipo de medida de similaridade é que são bons quando existem, por exemplo, erros de digitação e ortografia nas palavras. Mas por outro lado, não é útil para identificar termos organizados de maneiras diferentes (*data analyzing* e *analyzing data*, por exemplo).

Exemplos de abordagens das medidas de similaridades baseado em sequência: distância de edição; medida de *Needleman-Wunsch*; medida de distância afim; medida de *Smith-Waterman*; medida de *Jaro*; medida de *Jaro Winkler*. Nesta pesquisa foi utilizado a medida de *Jaro Winkler* (WINKLER, 1999), que é uma extensão da medida de *Jaro* (JARO, 1989) (JARO, 1995).

A medida de *Jaro Winkler* é explanada para justificar o seu uso e descrever como ele foi utilizado. As outras abordagens, citadas anteriormente, não serão detalhadas, tendo em vista que as mesmas não foram aplicadas no estudo.

Como a medida de *Jaro Winkler* é uma extensão da medida de *Jaro*, é essencial descrevê-la. O seu funcionamento se baseia principalmente em comparar pequenas *strings*. Considera que caracteres comuns são aqueles idênticos e que são posicionados um perto do outro. O *score* de *Jaro* é calculado da seguinte maneira (Equação 2.1):

$$jaro(x, y) = 1/3[c/|x| + c/|y| + (c - t/2)/c] \tag{2.1}$$

Dado um exemplo *x = jon* e *y=john*, o número de caracteres comuns é *c = 3*. Como não há transposição, ou seja, a sequência de caracteres em *x* é a mesma em *y*, sendo assim *t = 0*. Aplicando o cálculo: *jaro(x,y) = 1/3(3/3 + 3/4 + 3/2) = 0.917*. Este resultado mostra a diferença encontrada para o mesmo exemplo com a medida de distância de edição, que obteve 0.75.

A medida de *Jaro Winkler* leva em consideração as inserções, exclusões e transposições, calculando o número de caracteres comuns das palavras. O *Winkler* é um melhoramento do algoritmo de *Jaro*, que aumenta a medida de correspondências entre nomes, combinando os caracteres iniciais. Os valores gerados pela aplicação do algoritmo representam o quanto duas *strings* são similares, ficando no intervalo de 0 (zero) a 1 (um), representando maior similaridade quando se aproxima de 1.

Para realizar a correspondência, *Jaro Winkler* analisa cada caractere do prefixo da primeira *string* correspondendo ao primeiro caractere da segunda. O cálculo pode ser visto na Equação 2.2. Dois parâmetros são utilizados: *PL* e *PW*, que correspondem respectivamente ao comprimento do maior prefixo comum e o peso dado ao prefixo.

$$jaro - winkler(x, y) = (1 - PL * PW) * jaro(x, y) + PL * PW \tag{2.2}$$

A medida trabalha bem com *strings* curtas, além de nomes abreviados, sendo útil para as *strings* testadas nesta dissertação.

(b) Baseadas em conjuntos: Os tipos de medidas de similaridades baseadas em conjuntos consideram as *strings* como um conjunto ou múltiplos conjuntos de *tokens*, usando isto para calcular *scores* de similaridades. Existem algumas maneiras de dividir as *strings* em *tokens*, geralmente por delimitadores entre as palavras.

É comum que *stop words* (ou seja, palavras que podem ser consideradas irrelevantes, como *e, o, de*) não sejam consideradas. Um exemplo da maneira como as *strings* são divididas em conjunto de *tokens* é: dada a *string* (*david smith*), o conjunto de *tokens* que pode ser gerado é {*david*, *smith*}. A divisão em *q-grams* também pode ser explorada, onde é levado em consideração *substrings* de comprimento *q* presentes nas *strings*. Levando em consideração o mesmo exemplo anterior, o conjunto de todos os *3-grams* de *david smith* é {##d, #da, dav, avi,...ith, h##, th#}.

Existem diversas medidas de similaridades que consideram um ou múltiplos conjuntos de *tokens* como a medida de sobreposição, medida de *Jaccard* e medida de TF/IDF. A medida

de *Jaccard* é explorada neste estudo, pois é a mais comum para medir a sobreposição de dados. A medida de *Jaccard* é dada pela Equação 2.3. (ANHAI DOAN; IVES, 2012)

$$J(x, y) = \frac{|Bx \cap By|}{|Bx \cup By|} \tag{2.3}$$

Para exemplificar como funciona, dada as *strings x = dave* e *y = dav*, os valores de *Bx* e *By* são: *Bx = {#d, da, av, ve, e#}* e *By = {#d, da, av, v#}*. Então a medida de *Jaccard* é dada por $J(x, y) = \frac{3}{6}$, ou seja, são três elementos em comum nos dois conjuntos e seis é o número da união entre eles.

(c) Híbrida: As medidas de similaridades híbridas reúnem diversas medidas vistas anteriormente (*baseada em sequência* e *baseada em conjuntos*), assim como as medidas de similaridades que não podem ser categorizadas em outras. Com esta combinação, o objetivo deste tipo de medida de similaridade é melhorar a métrica.

Alguns exemplos de medidas são: medida generalizada de *Jaccard*, medida de *Soft TF/IDF* e medida de *Monge-Elkan*. Estas medidas não são descritas com mais detalhes em virtude da não utilização das mesmas. Mais detalhes podem ser vistos em (ANHAI DOAN; IVES, 2012).

(d) Fonética: Existe uma grande diferença das medidas de similaridades vistas até o momento e a fonética. Enquanto as outras se baseiam na parte *"visual"* das *strings*, a fonética leva em consideração o *"som"*. Correspondências de nomes são bem eficientes, pois geralmente são escritos de maneiras diferentes, mas são pronunciados de maneiras iguais. Um exemplo descrito por (ANHAI DOAN; IVES, 2012) são as palavras *Meyer, Meier e Mire* que possuem o mesmo som.

A medida mais comum utilizada é a *Soundex*, que usa códigos baseados no som de cada letra para traduzir uma *string* em uma forma regular de no máximo quatro caracteres, preservando a primeira letra. Valores são atribuídos aos nomes de maneira que nomes com sons semelhantes obtenham o mesmo valor, sendo que esses valores são conhecidos como codificações *soundex*.

(e) Baseada em conhecimento: Este tipo de medida de similaridade leva em consideração informações de redes semânticas para identificar o quanto duas palavras são similares. A abordagem semântica usa uma representação do conhecimento, como a interconexão dos fatos e os significados das palavras. Além disso inclui taxonomia e ontologia.

Existem diversas ontologias como *SENSUS1, Cyc2, UMLS3* e *SNOMED4*, mas a mais

popular é o *WordNet* (MILLER, 1995). Por ser amplamente utilizado em diferentes pesquisas como em (FARKHUND IQBAL BENJAMIN C. M. FUNG; MARRINGTON, 2019) e bem consagrado na literatura, esta dissertação fez uso do *WordNet*. Basicamente o *WordNet* é um grande banco de dados lexical em inglês, que organiza substantivos, verbos, advérbios e adjetivos em conjuntos de sinônimos (*synsets*).

O *WordNet* pode ser interpretado como uma taxonomia, pois as palavras são organizadas de maneira hierárquica usando conceitos de hiponímia e hiperonímia (relação todo-parte), e o que faz a ligação dos *synsets* são as relações conceitual-semântica e lexical. Medidas de similaridades diferentes podem ser obtidas pelo *WordNet*, como as medidas *Resnik* (res), *Lin* (lin), *Jiang Conrath* (jcn) e *Wu & Palmer* (wup).

Para esta pesquisa foi utilizada a medida de *Wu & Palmer* (WU; PALMER, 1994), tendo em vista seu bom desempenho para determinar a similaridade de palavras (DIDIK DWI PRASETYA; HIRASHIMA, 2018) (TINGTING WEI YONGHE LU; BAO, 2014). O algoritmo utiliza o menor comprimento de caminho entre os conceitos, ou seja, leva em conta a posição dos conceitos na taxonomia em relação à posição do conceito comum mais específico. Esta medida retorna uma pontuação que indica como os sentidos de duas palavras são semelhantes. Os resultados que podem ser obtidos na medida de *Wu & Palmer* estão no intervalo de 0 (zero) a 1 (um), sendo que quanto mais perto de 1 maior é a similaridade entre os elementos.

A Equação 2.4 ilustra como é realizado o cálculo de *Wu & Palmer*, tendo em vista que o mesmo leva em consideração o LCS (*least common subsumer)*. O LCS é o ancestral comum mais específico de dois conceitos encontrados em uma determinada ontologia. Semanticamente, representa a semelhança do par de conceitos. Para medir a similaridade entre dois elementos, primeiro ele encontra o nó de LCS que conecta os elementos e então calcula a distância:

$$\delta Wu\_Palmer(Cp, Cq) = \frac{2d}{Lp + Lq + 2d} \tag{2.4}$$

Onde *d* é a profundidade do LCS a partir da raiz, *Lp* é o comprimento do caminho entre *Cp* e LCS, e *Lq* é o comprimento do caminho entre *Cq* e LCS.

(f) Baseada em radicais: Medir a similaridade por meio da análise do radical das palavras diz respeito à técnica de *stemming* (extrator de radicais). Esse tipo de técnica é uma maneira popular de reduzir o tamanho das palavras em tarefas de linguagem natural, combinando palavras relacionadas.

Conforme (SCHOFIELD; MIMNO, 2016), o *stemming* converte palavras com o mesmo

radical ou raiz em uma única palavra. Por exemplo as palavras *"creative"* e *"creator"* são reduzidas ao seu radical *"create"*. O benefício da utilização de *stemmers* é o melhoramento na percepção de pequenas diferenças morfológicas encontradas nas palavras. *Stemmers* baseados em regras, como o bem-conceituado *stemmer* de Porter (PORTER, 1980), aplicam métodos organizados por um conjunto de regras que convertem um afixo em outro.

O *stemmer* de Porter é aplicado nesta dissertação tendo em vista sua consolidação na literatura e sua popular utilização. Ele usa cinco fases de regras e condições que correspondem aos padrões de sequências de vogal e consoante. Para aplicar o *stemmer* de Porter foi utilizado o algoritmo de (PORTER, 2006), que realiza a remoção dos prefixos e sufixos das palavras da língua inglesa, para obter o radical, ou seja, reduzir as palavras à sua base (*stem*).

2.4 PROCESSO ANALÍTICO HIERÁRQUICO

Diante das medidas de similaridades apresentadas na Seção 2.3, o método de tomada de decisão AHP (Processo Analítico Hierárquico) é utilizado para determinar os pesos que cada uma das medidas de similaridades utilizadas possui no processo para casamento de esquemas. Por meio de uma hierarquia de relevância, que pode ser vista em (SAATY, 2008), foram atribuídos os pesos 1, 2 e 3 para as técnicas de extrator de radicais, similaridade de conhecimento e similaridade de caracteres, respectivamente.

Para demonstrar o motivo da utilização da hierarquia de relevância apresentada, considera-se os valores de importância considerados em (SAATY, 2008) na Tabela 1.

Tabela 1 – Valores dos níveis de relevância do método AHP

1	Importância igual
2	Levemente mais importante
3	Importância moderada
4	Mais moderado
5	Forte importância
6	Mais forte
7	Importância muito forte ou demonstrada
8	Muito, muito forte
9	Extrema importância

Fonte: (SAATY, 2008)

A ordem de importância das técnicas apresentadas, considerando a tabela anterior, pode ser definida como: a análise do conhecimento é levemente mais importante que o radical (valor

2); a análise de caracteres possui importância moderada em comparação à análise de conhecimento (valor 3); e a análise de caracteres é mais moderado do que o radical (valor 4). A organização dos critérios é dada por uma matriz (Figura 6), onde *x* é a análise de conhecimento, *y* representa os radicais e *z* é a análise de caracteres.

Figura 6 – Critérios de importância em uma matriz

$$M = \begin{array}{c} \begin{matrix} x & y & z \end{matrix} \\ \begin{bmatrix} 1 & 2 & 1/3 \\ 1/2 & 1 & 1/4 \\ 3 & 4 & 1 \end{bmatrix} \end{array} \begin{matrix} x \\ y \\ z \end{matrix}$$

Fonte: (SAATY, 2008) (Adaptado)

Para gerar os pesos de cada técnica, é calculado a matriz quadrada da matriz representada na Figura 6. Depois é realizado a divisão da soma de todas as linhas pela soma de cada linha. A partir disso, um vetor que normaliza é gerado para que seja multiplicado pela matriz apresentada na Figura 6. Um *ranking* é montado (com os resultados obtidos), estabelecendo quais critérios são mais importantes. Foi possível observar que a análise de caracteres apresenta maior pontuação de relevância. Mais detalhes podem ser vistos em (SAATY, 2008).

2.5 TRABALHOS RELACIONADOS

Considerando os estudos analisados, foi observada a falta de pesquisas voltadas para o casamento de esquemas de documentos JSON, assim como trabalhos que levassem em consideração a estrutura dos mesmos e as instâncias presentes nos documentos JSON. Diversos estudos voltados para a realização da extração e/ou descoberta de esquemas de documentos JSON foram analisados, mas destacou-se (MACHADO, 2017) e (MEIKE KLETTKE; SCHERZINGER, 2015). Quanto ao casamento de esquemas, foi observado os estudos voltados para documentos XML, como em (LAURI MUKKALA JUKKA ARVO; KNUUTILA, 2017) e (W. E. DJEDDI; YAHIA, 2015).

Um processo para a extração de esquemas em fontes de dados JSON pode ser visto em (MACHADO, 2017). Por meio da análise dos campos que representam a mesma informação, mas são escritos de maneiras diferentes, o processo faz uma análise de similaridades de caracteres, conhecimento e radicais. As técnicas de similaridades textuais utilizadas são: *WordNet* com a medida de *Lin*; distância de *Levenshtein*; e extrator de radicais Porter. O objetivo é extrair o

esquema implícito presente nas fontes de dados, gerando um esquema conceitual. É importante destacar que não leva em consideração a estrutura dos documentos JSON e nem as instâncias contidas neles, ou seja, apenas os nomes dos campos quanto à sua grafia.

Em (MEIKE KLETTKE; SCHERZINGER, 2015) o objetivo é também a extração de esquemas em documentos JSON. O algoritmo proposto utiliza um conjunto de medidas de similaridade que determina o grau de heterogeneidade dos dados JSON, além de revelar *outliers* estruturais nos dados. Os *outliers* estruturais são padrões que ocorrem apenas em poucos conjuntos de dados, sendo que às vezes são advindos de erros. Um esquema JSON é gerado a partir de um conjunto de dados disponíveis.

Os erros ou *outliers* que ocorrem frequentemente em documentos JSON têm como motivo que na maioria dos sistemas de banco de dados NoSQL não é verificado nenhuma restrição estrutural. Isso ocorre pelo motivo dos dados serem coletados por longos períodos de tempo, além de diferentes aplicativos e usuários modificarem os dados.

A pesquisa de (LAURI MUKKALA JUKKA ARVO; KNUUTILA, 2017) descreve uma ferramenta para casamento de esquemas XML, chamada *TRC-Matcher*. A proposta é de um algoritmo *matcher* híbrido, que necessita de nenhuma ou pequena colaboração do usuário. Ele utiliza métodos baseados em: dicionário de sinônimos *WordNet* e abreviações, distância de *Jaro Winkler* e perfil de conteúdo. É importante destacar que o TRC-Matcher não leva em consideração a estrutura para casar os elementos. Além disso, a pesquisa descreve uma versão aprimorada do TRC-Matcher que utiliza *machine learning* (aprendizado de máquina) com algoritmos *matching* voltados para trabalhar com dados de testes comerciais.

O sistema *matching* denomidado XMap (W. E. DJEDDI; YAHIA, 2015) utiliza três camadas para calcular as correspondências entre elementos: terminológica, estrutural e de alinhamento. A camada terminológica calcula as semelhanças entre os nomes de entidades dentro das ontologias por meio da combinação das semelhanças linguísticas com os elementos semânticos. A camada estrutural calcula a similaridade levando em consideração a posição do elemento na ontologia e as restrições do elemento. E ao final, a camada de alinhamento fornece a matriz de similaridade final entre os conceitos. XMap utiliza *WordNet*, selecionando um conjunto de sinônimos com o maior potencial de correspondência de elementos com significados semelhantes.

A Tabela 2 sumariza os trabalhos discutidos voltados para casamento e extração de esquemas JSON e XML. Algumas comparações foram realizadas a fim de demonstrar quais mé-

todos e/ou técnicas foram analisadas neste estudo.

Tabela 2 – Trabalhos relacionados

Autores	**Finalidade**	**Métodos e/ou técnicas utilizadas**	**Formato de dados suportado**
[Lauri Mukkala and Knuutila, 2017]	Casamento de esquemas	Dicionário de sinônimos WordNet; Medida de Jaro Winkler; Perfil de conteúdo	XML
[W. E. Djeddi and Yahia, 2015]	Casamento de esquemas	Dicionário de sinônimos WordNet; Similaridade estrutural	XML
[Machado, 2017]	Extração de esquemas	Dicionário de sinônimos WordNet; Medida de Levenshtein; Extrator de radicais Porter	JSON
[Meike Klettke and Scherzinger]	Extração de esquemas	Algoritmo com conjunto de medidas de similaridade; Detecta outliers em documentos	JSON
Proposta	Casamento de esquemas	Considera a estrutura com: Dicionário de sinônimos WordNet; Medida de Jaro Winkler; Extrator de radicais Porter; Sobreposição de dados (medida de Jaccard)	JSON

Fonte: Autor

Os estudos citados anteriormente foram brevemente sintetizados, mas foi possível observar que fazem uso de diferentes métodos e algoritmos para realizar a correspondência de esquemas, utilizando documentos XML ou JSON como entrada. Até o presente momento, incluindo os estudos em análise, não foram relatados experimentos especificamente com documentos JSON voltados para casamento de esquemas, apenas para a extração. Sendo assim, acredita-se que especificar um processo para realizar o casamento de esquemas de documentos JSON é uma contribuição relevante na comunidade de banco de dados, uma vez que o casamento dos diversos elementos presentes nos documentos torna sua posterior integração mais eficaz, permitindo o acesso integrado por aplicações.

2.6 CONSIDERAÇÕES FINAIS

Apresentar os principais temas relacionados à integração de esquemas e casamento de esquemas é fundamental para compreender o estudo apresentado nesta dissertação. Por meio da fundamentação teórica é possível ter uma base dos principais conteúdos utilizados no processo

para casamento de esquemas. A integração de esquemas (Seção 2.1) é apresentada por meio da explanação das fases tradicionalmente encontradas na literatura. Uma das atividades principais é o casamento de esquemas, que é o foco principal desta dissertação.

As principais abordagens quanto ao casamento de esquemas podem ser visualizadas na Seção 2.2. Os componentes de um típico sistema de casamento de esquemas podem ser observados por meio de exemplos.

As medidas de similaridades são descritas (Seção 2.3), tendo em vista que as mesmas são aplicadas para obter as equivalências entre os elementos. Inúmeras foram apresentadas, mas as principais são as medidas de: *Jaro Winkler*, *Jaccard*, *Wu & Palmer* e extrator de radicais.

Na Seção 2.4 foi explanado o método de tomada de decisão AHP, tendo em vista a sua importância para definir os pesos adotados em cada uma das medidas de similaridades utilizadas nesta dissertação. Já na Seção 2.5 é possível ter conhecimento dos principais trabalhos relacionados que tiveram grandes contribuições no processo desta dissertação.

3 PROCESSO PARA CASAMENTO DE ESQUEMAS DE DOCUMENTOS JSON

A partir de diferentes esquemas JSON de um determinado domínio, este trabalho realiza um processo para casamento de esquemas. Entende-se por processo a execução de todas as etapas para realizar o casamento dos esquemas. Como o casamento de esquemas funciona e técnicas de medição de similaridades, apresentados na Seção 2.2 e 2.3, são importantes para se ter uma orientação de como proceder no processo, tendo em vista que a literatura é bem difundida na área de integração de dados como um todo.

3.1 VISÃO GERAL

O casamento de esquemas, uma sub etapa essencial na integração de esquemas, foi definido para determinar a equivalência dos elementos presentes nos documentos JSON. A determinação da equivalência nos esquemas é medida com base em técnicas de similaridade (voltadas para os campos e as instâncias) e equivalência de ancestrais dos elementos.

Conforme apresentado na Seção 2.2, o casamento de esquemas pode aplicar inúmeros tipos de comparações dos elementos. Alguns exemplos são métodos baseados em linguística, baseados no conteúdo e significado dos elementos e uso de informações auxiliares, como a utilização de dicionários de sinônimos.

Esta dissertação realiza todo o processo para casamento de esquemas, determinando quais elementos são equivalentes e se os mesmos podem ser unificados. É importante ressaltar que são aplicados não apenas técnicas que determinem a similaridade dos elementos com base em sua grafia ou significado, mas também técnicas que levam em consideração a estrutura dos documentos JSON, assim como as instâncias contidas nos documentos, ou seja, o valor dos dados.

A Listagem 3.1 ilustra um exemplo de dois documentos JSON, onde os elementos *year* (linhas 11 e 10) possuem o mesmo nome (grafia igual), mas seus significados são diferentes, ou seja, no primeiro documento representa o ano de premiação e no segundo se refere ao histórico escolar. Sendo assim, não podem ser casados.

```
1  Documento 1                          |Documento 2
2  {                                    | {
3    "_id": 1,                          |   "_id": 1,
4    "name" : { "first" : "John",       |   "name" : { "first" : "Marc",
5      "last" : "Backus" },             |       "last" : "Ludk" },
```

```
6   "contribs" : [ "Fortran", "ALGOL",|   "school_history" : [
7   "Backus-Naur Form", "FP" ],       |   {
8   "awards" : [                      |       "institution":
9     {                               |           "The Pennington School",
10      "award" : "W.W. McDowell",    |       "year": 1990
11      "year" : 1967,                |   }
12      "by" : "IEEE Computer Society"|   ]
13    }, {                            | }
14      "award" : "Draper Prize",     |
15      "year" : 1993,                |
16      "by" :                        |
17        "National Academy           |
18        of Engineering"             |
19    }                               |
20  ]                                 |
21 }                                  |
```

Listagem 3.1 – Exemplo de dois documentos JSON

Uma maneira possível para resolver este tipo de incompatibilidade é levar em consideração os elementos ancestrais no momento de corresponder os elementos. No exemplo da Listagem 3.1, os elementos *awards* (linha 8) e *school_history* (linha 6) devem ser comparados para determinar se seus sub elementos (descendentes) podem ser casados.

Outra questão que pode ocorrer no momento de realizar a comparação entre diferentes elementos pode ser observada na Listagem 3.2. Os elementos da linha 5 (*address* e *detailed_address*) possuem a mesma finalidade, mas a forma de mostrá-las é diferente. Em *address* é possível visualizar todo o conteúdo em apenas um sub elemento e em *detailed_address* os sub elementos estão separados.

```
1  Documento 1                        | Documento 2
2  {                                  | {
3      "_id" : 1,                     |   "_id" : 1,
4      "name" : { "first" : "John",   |   "name" : { "first" : "Marc",
5          "last" : "Backus" },       |       "last" : "Ludk" },
6      "address" : [                  |   "detailed_address" : [
7          {                          |    {
8              "name" : "10732        |    'street': "Delancey St",
9                  Research Ave,      |    'number' : 1052,
10                 Austin, TX, USA"   |    "city" : "New York",
11         }                          |    "state" : "NY",
12     ]                              |    "country" : "USA"
13 }                                  |    }
14                                    |   ]
15                                    | }
```

Listagem 3.2 – Comparação entre elementos de dois documentos JSON

Estes e outros conflitos são importantes de serem observados no momento de realizar o casamento de esquemas. A fim de ilustrar de uma forma simplificada como o processo para casamento é realizado, as seções a seguir descrevem detalhadamente as atividades e subatividades

de cada fase.

3.2 PROCESSO PARA CASAMENTO DE ESQUEMAS

O processo para casamento de esquemas explanado neste estudo é ilustrado na Figura 7. A separação em quatro fases é essencial para detalhar as atividades e subatividades pertencentes a cada momento.

A 1ª fase realiza um pré-processamento dos documentos JSON, extraindo informações para a 2ª e 3ª fase. Por meio da entrada de uma coleção de documentos JSON pertencentes a um mesmo domínio, os mesmos são testados através do algoritmo *diff*. Dependendo das diferenças encontradas em seus elementos, podem ser considerados idênticos, ou seja, podem ser casados sem nenhum processamento, ou se não foram idênticos passam por todo o processo para realizar o casamento dos esquemas.

Uma análise do esquema é realizada na 2ª fase. Uma primeira verificação determina os elementos ancestrais e se os mesmos possuem similaridades. Diversas técnicas são aplicadas para determinar o quanto os elementos são similares. A partir dos elementos ancestrais são extraídos os seus elementos descendentes, que também são medidos quanto a sua similaridade, gerando candidatos ao casamento dos esquemas.

Na 3ª fase são analisados todos os elementos descendentes que não foram considerados candidatos ao casamento de esquemas na fase anterior, ou seja, aqueles que não obtiveram bons *scores* de similaridade. Por meio da extração das instâncias contidas nesses elementos, é realizado a sobreposição de valores dos dados para avaliar o quanto são equivalentes.

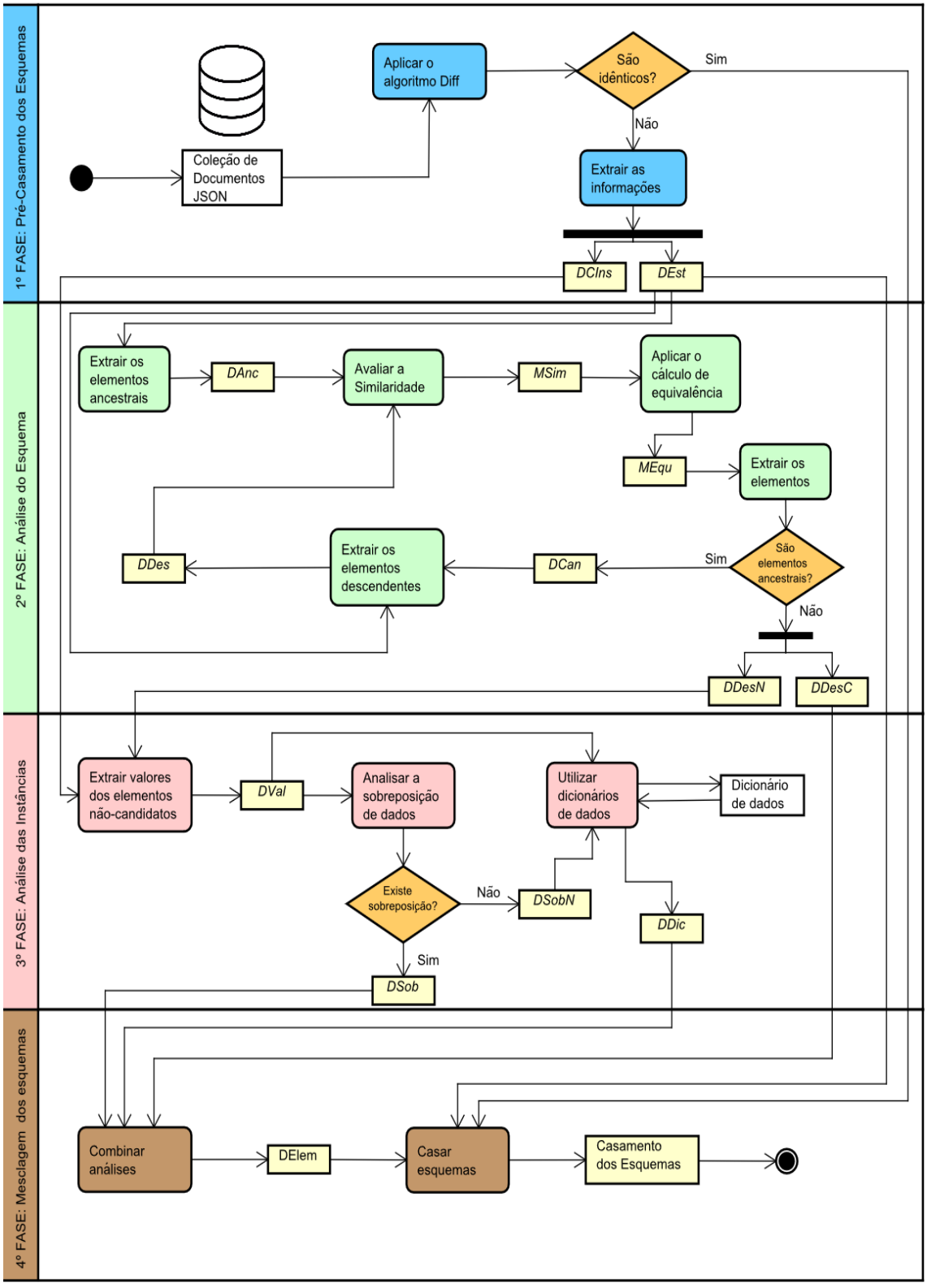
Com base nos elementos equivalentes encontrados na 2ª e 3ª fase, a 4ª fase consolida os esquemas, determinando quais são os elementos equivalentes e a quais documentos JSON de origem os mesmos pertencem. É importante destacar que as análises foram realizadas com documentos JSON de uma mesma coleção, onde os esquemas foram todos comparados de uma única vez (estratégia n-ária *one-shot*).

A fim de melhorar o entendimento das expressões e conceitos utilizados ao longo do texto são descritas algumas definições.

Definição 1. Estrutura. A estrutura de um documento JSON diz respeito à hierarquia dos campos, ou seja, são os elementos ancestrais e descendentes.

Por exemplo, na Listagem 3.3 o elemento *Person* (linha 3) é um ancestral acima dos elementos descendentes *Last* e *First* (linhas 5 e 6), enquanto que o elemento *NameList* (linha 1)

Figura 7 – Processo para casamento de esquemas



Fonte: Autor

está a dois níveis ancestrais de *Last* e *First*.

```
{"NameList": [
           {
             "Person": [
                       {
                  "Last": ["Hu"],
                  "First": ["Yang"]
                }
                  ]
                }
       ]
}
```

Listagem 3.3 – Exemplo de estrutura em um documento JSON

Definição 2. Esquema. Entende-se por esquema todos os campos e a estrutura presente nos documentos JSON.

Definição 3. Instância. Os valores dos dados contidos nos documentos JSON são as instâncias. As instâncias são analisadas para obter as similaridades dos campos. Na Listagem 3.3 os campos *Last* e *First* possuem as instâncias *Hu* e *Yang* respectivamente.

Definição 4. Documentos. Durante o processo são gerados documentos de texto (*.txt*) contendo diferentes informações. Os documentos são:

- ***DCIns*** - contém todos os campos extraídos dos documentos JSON com suas instâncias, ou seja, os valores dos dados obtidos;
- ***DEst*** - são os caminhos pertencentes a cada campo dentro dos documentos JSON, ou seja, descreve a estrutura;
- ***DAnc*** - são os elementos ancestrais que estão contidos nos documentos, independentemente do nível;
- ***MSim*** - são as matrizes de similaridades extraídas de cada técnica aplicada, gravadas com a extensão *.csv*.
- ***MEqu*** - contém a matriz de equivalência com os elementos considerados equivalentes e não equivalentes, gravada com a extensão *.csv*.
- ***DCan*** - são os elementos ancestrais equivalentes, considerados candidatos ao casamento de esquemas;
- ***DDes*** - são os elementos descendentes extraídos dos ancestrais equivalentes;

- ***DDesN*** - os elementos descendentes que não obtiveram equivalências estão contidos neste documento;
- ***DDesC*** - são os elementos descendentes considerados equivalentes;
- ***DVal*** - são os valores, ou seja, instâncias obtidas dos campos dos descendentes;
- ***DSob*** - são os nomes dos campos que obtiverem sobreposição nos seus valores;
- ***DSobN*** - contém os nomes dos campos que não obtiverem sobreposição;
- ***DDic*** - são os nomes dos campos considerados equivalentes em decorrência do uso de um dicionário de dados;
- ***DElem*** - são todos os elementos (campos) que foram considerados equivalentes;

As seções seguintes fazem um levantamento detalhado de cada fase do processo, mostrando as atividades realizadas em cada uma, assim como suas subatividades. As entradas, saídas, algoritmos e exemplos de cada fase são relatadas a fim de melhorar o entendimento de como funciona o processo como um todo. A organização da apresentação do processo é a seguinte: cada seção descreve uma fase com informações de entrada, atividades, saída e um exemplo, além disso as subseções exploram as atividades com mais detalhes, apresentando a entrada, o algoritmo que descreve a atividade, saída e um exemplo.

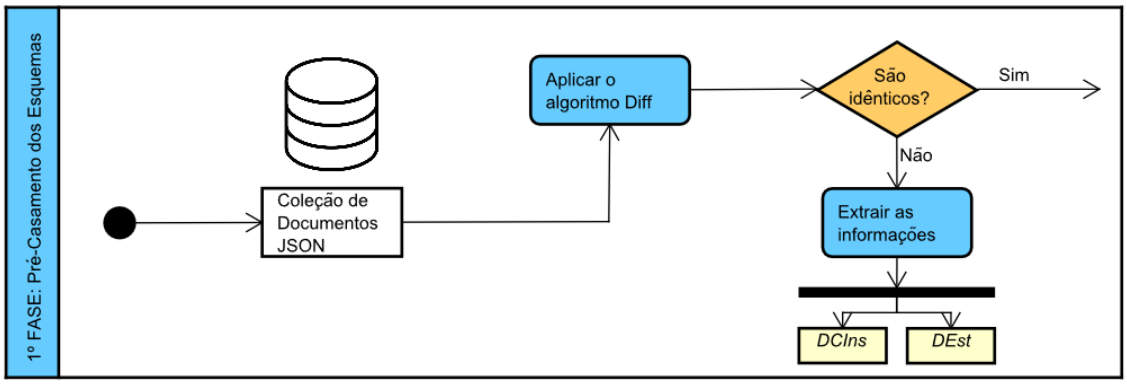
3.3 PRÉ-CASAMENTO DOS ESQUEMAS - 1ª FASE

O objetivo desta primeira fase, mostrada na Figura 8, é realizar um pré-processamento nos documentos JSON. Algumas informações (nomes dos campos, as instâncias e a estrutura hierárquica) são extraídas para serem utilizadas nas fases posteriores (2ª e 3ª fase). O início do processo se dá pela entrada de uma coleção de documentos JSON. Os documentos são todos comparados em uma única iteração, ou seja, utiliza a estratégia n-ária *one-shot* (que pode ser vista na Subseção 2.1.2).

Os documentos JSON que foram testados são do mesmo domínio de aplicação. Foi escolhido o domínio de publicações científicas, onde os arquivos de entrada são referências exportadas de artigos e demais trabalhos científicos de bibliotecas digitais.

Entrada: Uma coleção de documentos JSON pertencentes ao mesmo domínio.

Figura 8 – Pré-casamento dos esquemas



Fonte: Autor

Atividade Aplicar o algoritmo diff: O algoritmo é aplicado para realizar a comparação entre os documentos, determinando se são idênticos ou não.

Atividade Extrair as informações: Esta atividade percorre todo o documento, extraindo informações essenciais para serem utilizadas nas fases posteriores (2ª e 3ª fase).

Saída: Os artefatos de saída são o *DCIns* e *DEst*. As informações extraídas dos documentos são: os campos, instâncias e os caminhos que cada campo pertence.

Exemplo: Dados dois documentos JSON das bases de dados de *BibSinomy* (Documento 0) e *PubMed* (Documento 1), a Listagem 3.4 e 3.5 ilustra alguns trechos dos documentos.

```
{
  "Sources": {
    "@SelectedStyle": "",
    "Source": [
      {
        "SourceType": ["JournalArticle"],
        "Tag": ["hu2019synoptic"],
        "Title": [
          "A synoptic assessment of the summer extreme rainfall over the
   middle reaches of Yangtze River in CMIP5 models"
        ],
        "Year": ["2019"],
        "Author": [
          {
            "Author": [
              {
                "NameList": [
                  {
                    "Person": [
                      {
                        "Last": ["Hu"],
                        "First": ["Yang"]
                      },
                      {
                        "Last": ["Deng"],
```

```
                        "First": ["Yi"]
                      }
  ...
```

Listagem 3.4 – Documento 0

```
{
  "PubmedArticleSet": {
    "PubmedArticle": [
      {
        "MedlineCitation": [
          {
            "@Status": "PubMed-not-MEDLINE",
            "@Owner": "NLM",
            "PMID": [
              {
                "@Version": 1,
                "$": "28578772"
              }
            ],
            "DateCompleted": [
              {
                "Year": ["2017"],
                "Month": ["11"],
                "Day": ["01"]
              }],
            "Article": [
              {
                "@PubModel": "Print",
                "Journal": [
                  {
                    "ISSN": [
                      {
                        "@IssnType": "Electronic",
                        "$": "1531-6564"}],
                "ArticleTitle": [{  }],
                "AuthorList": [ {
                    "@CompleteYN": "Y",
                    "Author": [
                      {
                        "@ValidYN": "Y",
                        "LastName": [ "Luther"],
                        "ForeName": [ "Gaurav Aman"] }]}],
                "PublicationTypeList": [
                {

  ...
```

Listagem 3.5 – Documento 1

A primeira atividade é aplicar o algoritmo *diff*. Para fins de testes foi utilizada a ferramenta *online* disponível em JSONDiff[1], cujo resultado demonstrou que esses dois documentos não são idênticos. Caso eles fossem idênticos, os documentos passariam diretamente para a 4ª fase, onde seria realizada a consolidação dos elementos.

[1] http://www.jsondiff.com/

As próximas atividades realizam a extração de informações dos documentos em análise para serem utilizadas nas fases seguintes. A Listagem 3.6 ilustra o documento *DEst* e a Listagem 3.7 o documento *DCIns*.

```
[
Doc: 0, $['Sources'],
Doc: 0, $['Sources']['@SelectedStyle'],
Doc: 0, $['Sources']['Source'],
Doc: 0, $['Sources']['Source'][0],
Doc: 0, $['Sources']['Source'][1],
Doc: 0, $['Sources']['Source'][0]['SourceType'],
Doc: 0, $['Sources']['Source'][0]['Tag'],
Doc: 0, $['Sources']['Source'][0]['Title'],
Doc: 0, $['Sources']['Source'][0]['Year'],
Doc: 0, $['Sources']['Source'][0]['Author'],
...
Doc: 1, $['PubmedArticleSet'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]
    ['MedlineCitation'][0]['Article'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]
    ['MedlineCitation'][0]['Article'][0]['AuthorList'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]
    ['MedlineCitation'][0]['Article'][0]['ArticleTitle'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]
    ['MedlineCitation'][0]['Article'][0]['PublicationTypeList'],
...
```

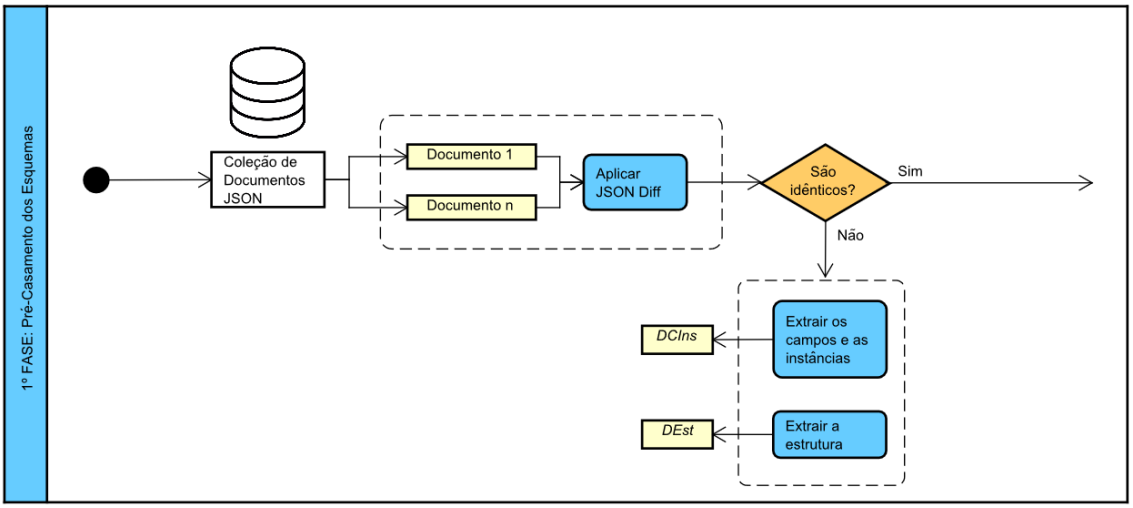
Listagem 3.6 – Documento *DEst*

```
[
Doc: 0, Campo: @SelectedStyle, Valor: ,
Doc: 0, Campo: SourceType, Valor: JournalArticle2,
Doc: 0, Campo: Tag, Valor: hu2019synoptic,
Doc: 0, Campo: Title, Valor: A synoptic assessment of the summer extreme
    rainfall over the middle reaches of Yangtze River in CMIP5 models,
Doc: 0, Campo: Year, Valor: 2019,
Doc: 0, Campo: Last, Valor: Hu,
Doc: 0, Campo: First, Valor: Yang,
...
Doc: 1, Campo: @Status, Valor: PubMed-not-MEDLINE,
Doc: 1, Campo: @Owner, Valor: NLM,
Doc: 1, Campo: $, Valor: 28578772,
Doc: 1, Campo: Year, Valor: 2017,
Doc: 1, Campo: Month, Valor: 11,
Doc: 1, Campo: LastName, Valor: Luther,
Doc: 1, Campo: ForeName, Valor: Gaurav Aman,
...]
```

Listagem 3.7 – Documento *DCIns*

A atividade *Extrair as informações* dos documentos é dividida em duas subatividades, conforme é observado na Figura 9. Estas subatividades e a atividade de *Aplicar o algoritmo diff* são descritas com mais detalhes a seguir.

Figura 9 – Detalhamento da 1ª Fase



Fonte: Autor

3.3.1 Aplicar *JSON Diff*

O objetivo desta atividade é determinar se os documentos JSON são idênticos ou não, auxiliando na redução do processamento, no caso de serem idênticos. Nesta atividade os documentos são analisados de dois em dois, analisando todos com todos. Em um primeiro momento é aplicado o algoritmo *diff* que determina se os esquemas em comparação são iguais. Se todos são idênticos, então são passados diretamente para a última fase, não necessitando passar por todas as fases do processo. Mas se não forem idênticos a atividade de *Extrair as informações* é realizada.

O algoritmo *diff* ou *JSON Diff* é uma ferramenta de comparação semântica para documentos JSON. Por meio da comparação dos arquivos, é gerado as diferenças existentes entre eles. É uma boa opção quando se quer trabalhar com documentos grandes devido às dificuldades encontradas para visualizar as diferenças entre eles.

Entrada: Coleção de documentos JSON

Atividade: Descrita pelo Algoritmo 1

Saída: Coleção de documentos JSON processados

O Algoritmo 1 descreve como a atividade *Aplicar JSON Diff* funciona, contendo a entrada e a saída.

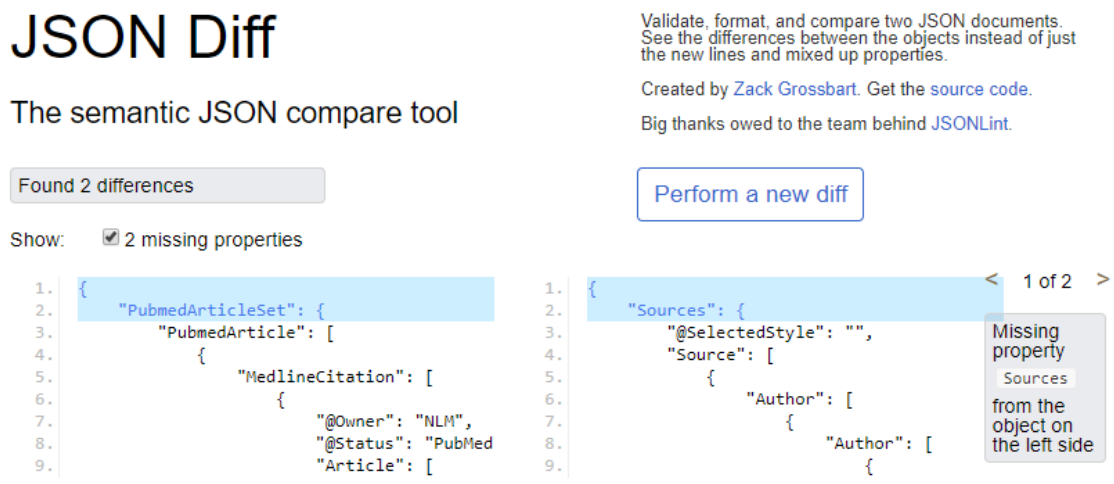
Algoritmo 1: Atividade Aplicar JSON Diff

Entrada: Coleção de documentos JSON
Saída: Coleção de documentos JSON processados

para *cada n documentos da coleção* **faça**
    Aplicar algoritmo Diff;
    se *todos são idênticos* **então**
        Atividade Casar esquemas;
    fim
    senão
        Atividade Extrair as informações;
    fim
fim

Exemplo: Para demostrar como a atividade *Aplicar JSON Diff* funciona considera-se as Listagem 3.4 e 3.5. Por meio da aplicação do algoritmo na ferramenta *online* disponível em JSONDiff[2] foi possível observar (Figura 10) que os documentos não são idênticos.

Figura 10 – Aplicação do *JSON Diff*



Fonte: Autor

Sendo assim, os mesmos necessitam passar por todo o processo. Se os arquivos fossem idênticos, ou seja, possuíssem a mesma semântica iriam diretamente para a 4ª fase, sem a necessidade de processamento.

[2] http://www.jsondiff.com/

3.3.2 Extrair os campos e as instâncias

Esta subatividade extrai os nomes dos campos contidos nos documentos JSON, além de suas instâncias, para a continuidade de todo o processo. Para isto é utilizado a API Java JSR 353, que processa documentos JSON por meio de um *parser*. Este divide o documento em eventos que podem ser analisados como *JSONObject* ou *JSONArray*. Os eventos utilizados são os que contém os campos e as instâncias, resultando no documento denominado *DCIns*.

Quanto aos valores dos dados extraídos, ou seja, as instâncias de cada campo contido nos documentos JSON, podem ser *strings* ou números. Outra importante informação armazenada em *DCIns* é a designação de qual documento pertence os campos e as instâncias (Doc 0 ou Doc 1, por exemplo).

Entrada: Coleção de documentos JSON processados

Atividade: Descrita pelo Algoritmo 2

Saída: Documento *DCIns*

O funcionamento da subatividade *Extrair os campos e as instâncias* pode ser vista no Algoritmo 2 com mais detalhes. É possível notar que os eventos determinam que tipo de informação irá ser considerada. Os nomes tratados neste algoritmo dizem respeito aos nomes dos campos, assim como os valores se referem às instâncias encontradas logo após os campos. Os campos que não possuem instâncias não são considerados neste algoritmo. Os eventos presentes na API Java JSR 353 são denominados *KEY_NAME* (relacionado aos campos) e *VALUE_STRING* (relacionado às instâncias).

Algoritmo 2: Subatividade *Extrair os campos e as instâncias*

```
Entrada: Coleção de documentos JSON processados
Saída: Documento dos campos e das instâncias DCIns
para cada documento faça
    para cada evento faça
        se evento for nome então
            se próximo evento for valor então
                Grava nome e valor
            fim
        fim
    fim
    Grava arquivo de texto DCIns
fim
```

Exemplo: Os campos e as instâncias dos documentos contidos nas Listagem 3.4 e 3.5 podem ser visualizados a seguir na Listagem 3.8.

```
[
Doc: 0, Campo: @SelectedStyle, Valor: ,
Doc: 0, Campo: SourceType, Valor: JournalArticle2,
Doc: 0, Campo: Tag, Valor: hu2019synoptic,
Doc: 0, Campo: Title, Valor: A synoptic assessment of the summer extreme
    rainfall over the middle reaches of Yangtze River in CMIP5 models,
Doc: 0, Campo: Year, Valor: 2019,
Doc: 0, Campo: Last, Valor: Hu,
Doc: 0, Campo: First, Valor: Yang,
Doc: 0, Campo: Last, Valor: Deng,
Doc: 0, Campo: First, Valor: Yi,
...
Doc: 1, Campo: @Status, Valor: PubMed-not-MEDLINE,
Doc: 1, Campo: @Owner, Valor: NLM,
Doc: 1, Campo: $, Valor: 28578772,
Doc: 1, Campo: Year, Valor: 2017,
Doc: 1, Campo: Month, Valor: 11,
Doc: 1, Campo: Day, Valor: 01,
Doc: 1, Campo: @PubModel, Valor: Print,
Doc: 1, Campo: @IssnType, Valor: Electronic,
Doc: 1, Campo: $, Valor: 1531-6564
Doc: 1, Campo: @CompleteYN, Valor: Y,
Doc: 1, Campo: @ValidYN, Valor: Y,
Doc: 1, Campo: LastName, Valor: Luther,
Doc: 1, Campo: ForeName, Valor: Gaurav Aman,
...]
```

Listagem 3.8 – Documento *DCIns* dos Documentos 0 e 1

Os campos são extraídos sem os delimitadores de objetos e *arrays*, ou seja, apenas o nome dos campos propriamente ditos. Por meio da utilização da API Java JSR 353 é possível escolher apenas o evento *nome* para extrair os campos.

Quanto às instâncias são extraídos os valores pertencentes aos campos correspondentes, para determinar futuramente a qual campo a instância pertence. É possível observar que na Listagem 3.8 as instâncias pertencem a diferentes campos, determinadas por *Valor*.

É importante notar que os campos que não possuem instâncias não são considerados. Por exemplo, o campo *Sources* que pode ser visto na Listagem 3.4, não possui instâncias, sendo assim não aparece no documento *DCIns*.

3.3.3 Extrair a estrutura

Tendo como objetivo extrair a estrutura hierárquica dos documentos JSON, esta atividade analisa os caminhos dos campos presentes nos documentos JSON. A estrutura presente nos documentos é essencial para determinar quais elementos são ancestrais e descendentes. Para esta subatividade foi utilizada a linguagem *JSONPath*. Por meio desta é possível visualizar o caminho que os campos estão inseridos dentro de um documento JSON. O resultado da

utilização do *JSONPath* são expressões de caminho de cada campo do documento em análise. Mediante estas informações é viável visualizar a estrutura hierárquica dos documentos JSON.

As expressões *JSONPath* se referem a uma estrutura presente nos documentos JSON, do mesmo modo que as expressões do *XPath* são usadas para documentos XML. A Tabela 3 ilustra as diferenças entre o *JSONPath* e *XPath*, este é bem utilizado para manipulação de documentos XML.

Tabela 3 – Comparação entre *XPath* e *JSONPath*

XPath	**JSONPath**	**Descrição**
/	$	Objeto / elemento raiz
.	@	Objeto / elemento atual
/	. or []	Operador filho
..	n/a	Operador pai
//	..	Descida recursiva
*	*	Todos os objetos / elementos, independentemente de seus nomes
@	n/a	Acesso de atributo. As estruturas JSON não têm atributos
[]	[]	Operador subscript. O XPath o utiliza para iterar sobre coleções de elementos e para predicados. Em Javascript e JSON, é o operador de matriz nativo
|	[,]	O operador Union no XPath resulta em uma combinação de conjuntos de nós. JSONPath permite nomes alternativos ou índices de matriz como um conjunto
n/a	[start:end:step]	Operador de divisão de matriz
[]	?()	Aplica uma expressão de filtro (script)
n/a	()	Expressão de script, usando o mecanismo de script subjacente
()	n/a	Agrupamento no XPath

Fonte: Adaptado de (GöSSNER, 2007)

Entrada: Coleção de documentos JSON processados

Atividade: Descrita pelo Algoritmo 3

Saída: Documento *DEst*

O Algoritmo 3 é detalhado a seguir, a fim de ilustrar como a subatividade *Extrair a estrutura* funciona.

Algoritmo 3: Subatividade *Extrair a estrutura*

Entrada: Coleção de documentos JSON processados
Saída: Documento dos caminhos *DEst*

para *cada documento* **faça**
  para *cada caminho desde a raiz $* **faça**
    Grava caminho
  fim
  Grava arquivo de texto *DEst*
fim

Exemplo: Os documentos contidos nas Listagem 3.4 e 3.5 são utilizados neste exemplo. Por meio da utilização do *JSONPath* é possível realizar a extração dos caminhos contidos nos documentos, ou seja, é utilizado a opção de listar todo o caminho desde a raiz, representada pelo símbolo $.

A Listagem 3.9 ilustra trechos do documento *DEst* gerado. É possível visualizar que o caminho que os campos percorrem dentro do documento JSON podem ser explorados pelo *JSONPath*, ou seja, sua estrutura hierárquica. O trecho entre as linhas 2 e 3 descreve o caminho percorrido até o elemento folha *Last*, e as linhas 4 e 5 ilustra o caminho até o elemento *Day*.

```
1 [
2 Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
3         ['NameList'][0]['Person'][0]['Last'],
4 Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
5         ['DateCompleted'][0]['Day']
6  ...]
```

Listagem 3.9 – Trecho do documento *DEst* com os caminhos dos elementos *Last* e *Day*

3.4 ANÁLISE DO ESQUEMA - 2ª FASE

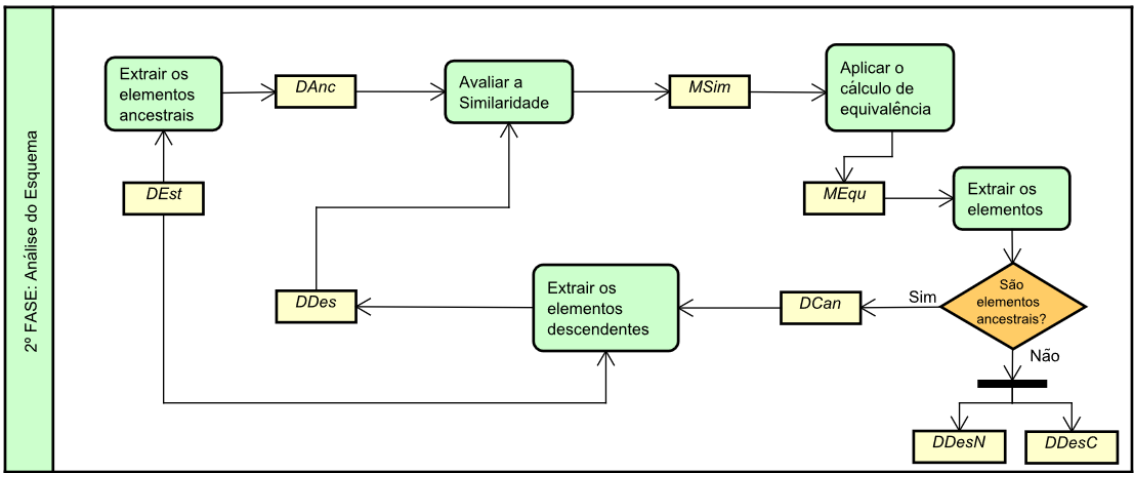
A aplicação de medidas de similaridades é executada nesta fase, tendo como objetivo encontrar elementos equivalentes. Primeiramente os elementos ancestrais são analisados por meio das medidas de similaridades. Depois os elementos descentes, extraídos dos ancestrais considerados equivalentes, são analisados com as medidas de similaridade. Nem todos os descendentes são processados, isto por que apenas aqueles elementos ancestrais que obtiverem equivalências extraem seus descendentes. Objetivando a comparação entre elementos de diferentes esquemas, três técnicas são aplicadas para medir a similaridade: extração de radicais, análise de caracteres e análise de conhecimento.

Quando se trata de medir a similaridade semântica e lexical, o principal objetivo é determinar, por meio de um valor, o quanto duas palavras são equivalentes entre si. Neste trabalho,

a combinação das análises de similaridade por cada técnica contribui para determinar a equivalência das palavras. Para auxiliar na definição de pesos é empregada a técnica de apoio à tomada de decisões AHP (descrita no Capítulo 2).

Como mencionado anteriormente, a 2ª fase realiza uma análise do esquema, medindo as similaridades dos ancestrais primeiramente, para então analisar seus descendentes. A Figura 11 ilustra a 2ª Fase de um modo geral, com seus artefatos de entrada e saída, além das atividades aplicadas. Alguns detalhamentos são observados a seguir.

Figura 11 – Análise do esquema



Fonte: Autor

Entrada: O *DEst* gerado na fase anterior é utilizado para realizar a extração dos elementos ancestrais.

Atividade Extrair os elementos ancestrais: Por meio da obtenção dos caminhos dos campos dentro dos documentos, esta atividade realiza a extração dos elementos ancestrais, ou seja, os elementos não folhas. Os elementos extraídos são guardados em um documento de texto para serem utilizados na próxima atividade.

Atividade Avaliar a similaridade: Esta atividade aplica três técnicas para avaliar o quanto duas palavras são equivalentes. As comparações são realizadas de modo que as palavras sejam testadas "todas com todas". Leva em consideração inúmeros fatores, como seu radical, grafia e significado. O resultado são três matrizes provenientes de cada técnica aplicada.

Atividade Aplicar o cálculo de equivalência: Por meio das matrizes resultantes da atividade anterior, são executados cálculos e testes a fim de determinar o grau de similaridade entre os elementos. É importante observar que para cada técnica utilizada são atribuídos pesos

diferentes, possibilitando que algumas sejam mais relevantes do que outras.

Atividade Extrair os elementos: A extração das palavras resultante da *Matriz de equivalência* podem ser armazenados em distintos documentos de textos, dependendo se são elementos ancestrais ou descendentes. Apenas os ancestrais que obtiveram equivalências são processados, e quando se trata de descendentes são extraídos os elementos que são considerados equivalentes e os que não são considerados.

Atividade Extrair os elementos descendentes: A partir dos elementos ancestrais considerados equivalentes, esta atividade realiza a extração dos elementos descendentes destes ancestrais. Para isto utiliza os documentos dos ancestrais equivalentes e da estrutura dos documentos JSON. Como resultado são extraídos os descendentes, cujos seus ancestrais são considerados candidatos.

Saída: Os artefatos de saída desta fase são os documentos contendo os elementos descendentes que possuem equivalências, ou seja, são candidatos (*DDesC*) e os que não possuem equivalências (*DDesN*).

Exemplo: A partir do documento com a estrutura dos documentos JSON (*DEst*), a atividade *Extrair os elementos ancestrais* gera um documento contendo os ancestrais presentes nos documentos JSON, como pode ser visto alguns trechos na Listagem 3.10.

```
1 Sources;
2 Source;
3 Author;
4 PubmedArticleSet;
5 PubmedArticle;
6 MedlineCitation;
7 PMID;
8 ...
```

Listagem 3.10 – Documento com os elementos ancestrais *DAnc*

A seguinte atividade (*Avaliar a similaridade*) aplica três técnicas de similaridade, comparando as palavras todas com todas. Os radicais são extraídos por meio do algoritmo de (PORTER, 2006), a similaridade de conhecimento utiliza o *WordNet* com a medida de *Wu & Palmer* e a técnica de análise de similaridade de *Jaro Winkler* calcula o número de caracteres comuns das palavras. Três matrizes são geradas com os valores de cada técnica aplicada. Para demonstrar, a Figura 12 ilustra um trecho da matriz resultante da aplicação da medida de *Wu & Palmer*.

Depois de geradas as três matrizes decorrentes das técnicas de similaridade, a atividade de *Aplicar o cálculo de equivalência* é executada tendo como base os casos previstos na Tabela 4. As técnicas são denominadas na tabela como extração de radicais (a), similaridade de caracteres (b) e similaridade de conhecimento (c).

Figura 12 – Trecho da Matriz de similaridade (*MSim* da análise de similaridade de conhecimento)

	Sou	Sou	Aut	Aut	Nar	Per	Pub	Pub	Me	PMI	Dat	Dat	Artic	Jou	ISS	Jou	Pub	Pagi	ELo
Sources	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Source	0.0	1.0	0.4	0.4	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.57	0.2	0.0	0.0	0.0	0.28	0.0
Author	0.0	0.0	1.0	1.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.58	0.2	0.0	0.0	0.0	0.23	0.0
Author	0.0	0.0	0.0	1.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.58	0.2	0.0	0.0	0.0	0.23	0.0
NameList	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Person	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.66	0.2	0.0	0.0	0.0	0.26	0.0
PubmedArticleSet	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubmedArticle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MedlineCitation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PMID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateCompleted	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateRevised	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Article	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.5	0.0	0.0	0.0	0.42	0.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.42	0.0
ISSN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
JournalIssue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
PubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

Fonte: Autor

Considerando que [*i*] e [*j*] se referem respectivamente à linha e coluna das matrizes, o par de elementos *Source* e *Article* obtiveram os valores: Radical[*i*][*j*] = 0, Jaro[*i*][*j*] = 0.64 e WuP[*i*][*j*] = 0.57. Como mais de duas técnicas obtiveram valores no intervalo de 0 a 1, aplica-se a média ponderada (*MPond*), atribuindo à *Matriz de equivalência* MEqu[*i*][*j*] = 1, pois o valor de *MPond* foi de 0,51 (maior que 0,5).

Tabela 4 – Casos que podem ocorrer dependendo do valor extraído da matriz de cada técnica (*MSim*)

Casos	**Valor**	**Cálculo de equivalência**	**Resultado**
(a) OU (b) OU (c)	1	Equivalente	1
(a) E (b) E (c)	0 ou ausência de valor	Não equivalente	0
(a) OU (b) OU (c)	Intervalo 0 <x <1	Valor >0,6	1
		Valor <= 0,6	0
Nenhum dos casos		Média ponderada >0,5	1
		Média ponderada <= 0,5	0

Fonte: Autor

Mais detalhes sobre como é realizado o cálculo de equivalência são mostrados nas subseções seguintes. A Matriz de equivalência (*MEqu*) é então gerada com os valores 0 ou 1, determinando os pares de palavras consideradas equivalentes e não equivalentes. A partir desta matriz, a atividade *Extrair os elementos* determina os pares de palavras que são equivalentes e quais não são. Uma verificação é realizada para determinar se são elementos ancestrais ou

descendentes. Se forem ancestrais, apenas os elementos equivalentes são armazenados no documento dos ancestrais candidatos (*DCan*). A Listagem 3.11 ilustra alguns pares de ancestrais considerados equivalentes neste exemplo.

```
0 -> Sources - Source
1 -> Sources - Article
2 -> Sources - Journal
3 -> Sources - JournalIssue
4 -> Sources - ArticleIdList
5 -> Source - Article
6 -> Source - ArticleId
7 -> Author - Author
8 -> Author - Person
9 -> Author - AuthorList
10 -> Author - Author
11 -> Author - Person
12 -> Author - AuthorList
13 -> Author - Author
14 -> NameList - DateRevised
15 -> NameList - AuthorList
16 -> Person - Author
...
```

Listagem 3.11 – Documento com os elementos ancestrais candidatos *DCan*

Os elementos descendentes provenientes dos ancestrais candidatos (*DCan*) são extraídos na atividade *Extrair os elementos descendentes*. Cada par de ancestrais são analisados, com o auxílio do documento da estrutura dos documentos (*DEst*), e seus descendentes obtidos (documento *DDes*). A Listagem 3.12 ilustra os elementos descendentes dos ancestrais (*Source* e *Article*; *Author* e *AuthorList*) da Listagem 3.11 (linhas 6 e 10).

```
SourceType; Tag; Title; Year; Author; StandardNumber1; JournalName; Month;
   URL; BIBTEXAbstract; BIBTEXKeyWords; StandardNumber2; @PubModel; Journal
   ; ArticleTitle; Pagination; ELocationID; AuthorList; Language;
   PublicationTypeList;
Author; @CompleteYN; Author;
```

Listagem 3.12 – Documento com os elementos descendentes *DDes*

A partir dos elementos descendentes, o processo se encaminha para a verificação das similaridades presentes nestes elementos. As atividades *Avaliar a similaridade*, *Aplicar o cálculo de equivalência* e *Extrair os elementos* são novamente utilizadas, mas com os elementos descendentes. A saída da atividade *Extrair os elementos* verifica que são elementos descendentes. Assim, dois documentos de elementos descentes candidatos e não candidatos são gerados. As Listagens 3.13 e 3.14 ilustram alguns pares de elementos descendentes candidatos e não candidatos.

```
[
...
Par de descendente: 5===SourceType - ArticleTitle
```

```
Par de descendente: 5===SourceType - PublicationTypeList
Par de descendente: 5===Tag - Language
Par de descendente: 5===Title - Journal
Par de descendente: 5===Title - ArticleTitle
Par de descendente: 5===Year - Month
Par de descendente: 9===Author - Author
...]
```

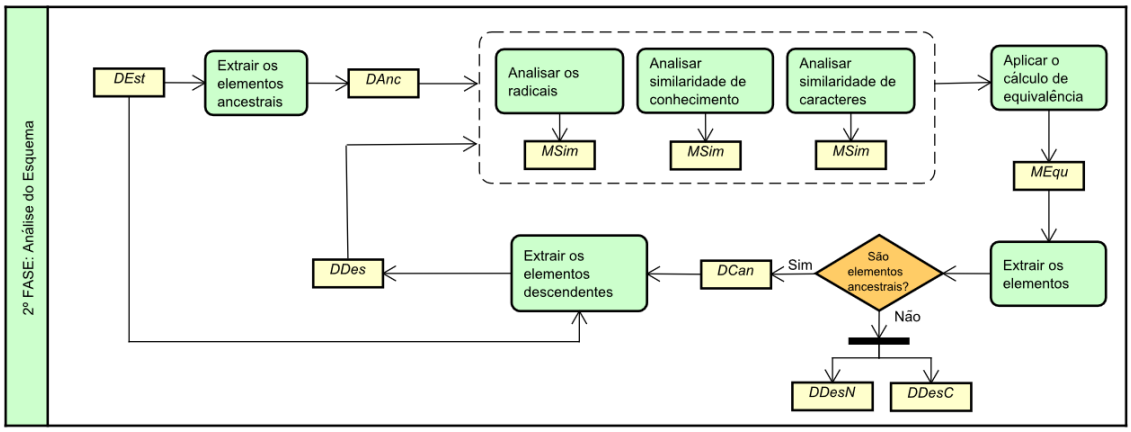
Listagem 3.13 – Documento com os elementos descendentes candidatos *DDesC*

```
[
Par de descendente nao equ.: 5===SourceType - Tag
Par de descendente nao equ.: 5===SourceType - Title
Par de descendente nao equ.: 5===SourceType - Year
Par de descendente nao equ.: 5===SourceType - Author
Par de descendente nao equ.: 5===Tag - Title
Par de descendente nao equ.: 5===Tag - Year
Par de descendente nao equ.: 5===Tag - Author
Par de descendente nao equ.: 5===Tag - StandardNumber1
Par de descendente nao equ.: 5===Tag - JournalName
Par de descendente nao equ.: 5===Title - Year
Par de descendente nao equ.: 5===Title - Author
Par de descendente nao equ.: 5===Title - StandardNumber1
Par de descendente nao equ.: 5===Title - JournalName
Par de descendente nao equ.: 5===Title - Month
...
```

Listagem 3.14 – Documento com os elementos descendentes não candidatos *DDesN*

As atividades descritas anteriormente são detalhadamente descritas nas subseções seguintes. Algumas atividades são divididas em diferentes subatividades, conforme é ilustrado na Figura 13. Destaca-se a divisão da atividade de *Avaliar a similaridade* que é composta pelas subatividades que representam as técnicas utilizadas.

Figura 13 – Detalhamento da 2ª fase



Fonte: Autor

3.4.1 Extrair os elementos ancestrais

Esta atividade tem como objetivo determinar quais são os elementos ancestrais presentes nos documentos JSON. Para atingir este objetivo o documento denominado *DEst* é utilizado. Tendo como base este documento é possível determinar os elementos ancestrais. Esta atividade foi realizada manualmente, considerando que elementos que são ancestrais são aqueles que possuem descendentes.

Entrada: Documento *DEst*

Atividade: Descrita pelo Algoritmo 4

Saída: Documento *DAnc*

Com o propósito de demonstrar como a atividade *Extrair os elementos ancestrais* funciona, o Algoritmo 4 é ilustrado a seguir.

Algoritmo 4: Atividade *Extrair os elementos ancestrais*

Entrada: Documento da estrutura *DEst*
Saída: Documento dos elementos ancestrais *DAnc*

repita
  para *cada campo de descendente* **faça**
    Grava o primeiro campo ancestral
  fim
  Grava arquivo de texto *DAnc*
até *documento terminar*;

Exemplo: O documento *DEst*, contendo as estruturas, pode ser observado na Listagem 3.6. A partir deste trecho, a Listagem 3.15 ilustra o resultado da aplicação desta atividade, ou seja, extrai os ancestrais da Listagem 3.6.

```
Sources;
Source;
Author;
NameList;
Person;
PubmedArticleSet;
PubmedArticle;
MedlineCitation;
PMID;
DateCompleted;
Article;
Journal;
ISSN;
AuthorList;
Author;
```

Listagem 3.15 – Documento com os elementos ancestrais *DAnc* retirados do documento *DEst*

3.4.2 Analisar os radicais

Esta subatividade está presente na atividade *Avaliar a similaridade*, tendo como objetivo explorar os radicais presentes nos elementos, tanto os ancestrais quanto os descendentes. Os documentos utilizados são dos ancestrais (*DAnc*) e dos descendentes (*DDEs*), dependendo em qual estágio está o processo.

A análise do radical das palavras diz respeito à técnica de *Stemming* (extrator de radicais), que é executada com o algoritmo de (PORTER, 2006). Este algoritmo realiza a remoção dos prefixos e sufixos das palavras da língua inglesa, para obter o radical, ou seja, reduzir as palavras à sua base (*stem*). Depois de aplicar o algoritmo de Porter, são realizadas comparações para ver quais radicais são equivalentes. No caso de total similaridade, o valor gerado para aquele par de palavras é 1 (um), mas se não há nenhuma similaridade, o valor gerado é 0 (zero).

Entrada: Documento *DAnc* ou *DDes*

Atividade: Descrita pelo Algoritmo 5

Saída: Matriz *MSim*

O Algoritmo 5 é explanado a seguir para demonstrar o funcionamento da subatividade *Analisar os radicais*.

Algoritmo 5: Subatividade *Analisar os radicais*

Entrada: Documento *DAnc* ou *DDes*
Saída: Matriz *MSim*

repita
  Executa o extrator de radicais
  para *cada radical* **faça**
    Compara este com todos os outros
    se *RadicalA é igual RadicalB* **então**
      MSim = 1
    fim
    senão
      MSim = 0
    fim
  fim
  Grava matriz *MSim*
até *documento terminar*;

Exemplo: O par de palavras ancestrais *Source* e *Article* possuem os descendentes ilustrados anteriormente na Listagem 3.12 (linha 1). Os radicais extraídos dessas palavras, que estão contidas no documento *DDes*, pode ser vista, a seguir, na Listagem 3.16.

```
1 [sourcetyp, tag, titl, year, author, standardnumb, journalnam, month, url,
     bibtexabstract, bibtexkeyword, standardnumb, pubmodel, journal,
```

```
articletitl, pagin, elocationid, authorlist, languag,
publicationtypelist]
```

Listagem 3.16 – Radicais extraídos para análise

A partir dos radicais extraídos é realizado uma verificação, testando todos com todos. Dependendo se as palavras são iguais ou não, um valor é atribuído na linha e coluna verificado. A Figura 14 ilustra um trecho da matriz *MSim* dos radicais da Listagem 3.16. Pode-se notar que apenas a diagonal principal obteve valor 1, tendo em vista que na diagonal principal são testados eles com eles mesmos.

Figura 14 – Matriz *MSim* da análise dos radicais

	sou	tag	titl	yea	autl	star	jou	mo	url	bib	bib	star	pub	jou	art	pag	elo	aut	lang	pul
sourcetyp	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
tag	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
titl	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
year	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
author	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
standardnumb	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
journalnam	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
month	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
url	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bibtexabstract	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bibtexkeyword	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
standardnumb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
pubmodel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
articletitl	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
pagin	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
elocationid	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
authorlist	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
languag	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
publicationtype	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

3.4.3 Analisar similaridade de conhecimento

Assim como a subatividade anterior, esta também compõe a atividade *Avaliar a similaridade*. Diferente da análise dos radicais, esta subatividade, denominada *Analisar similaridade de conhecimento*, leva em consideração a semântica das palavras para poder medir sua similaridade, ou seja, analisa o seu significado.

Uma ferramenta bem popular e que foi utilizada é o *WordNet* (MILLER, 1995). Este é um grande banco de dados léxico da língua inglesa que organiza substantivos, verbos, advérbios e adjetivos agrupados em conjuntos de sinônimos (*synsets*). O *WordNet* é uma abordagem

que mede a similaridade baseada no conhecimento, podendo ser categorizada em diferentes medidas.

Para esta dissertação foi utilizada a medida de *Wu & Palmer* (WU; PALMER, 1994), tendo em vista seu bom desempenho para determinar a similaridade de palavras (DIDIK DWI PRASETYA; HIRASHIMA, 2018). O algoritmo utiliza o menor comprimento de caminho entre os conceitos, ou seja, leva em conta a posição dos conceitos na taxonomia em relação à posição do conceito comum mais específico. Para a implementação foi utilizado o *Wordnet Similarity for Java*[3], considerando os resultados obtidos na medida de *Wu & Palmer*, trazendo valores no intervalo de 0 (zero) a 1 (um).

Entrada: Documento *DAnc* ou *DDes*

Atividade: Descrita pelo Algoritmo 6

Saída: Matriz *MSim*

O funcionamento da subatividade *Analisar similaridade de conhecimento* é descrita por meio do Algoritmo 6.

Algoritmo 6: Subatividade *Analisar similaridade de conhecimento*

```
Entrada: Documento DAnc ou DDes
Saída: Matriz MSim
repita
    Carrega palavras do documento no vetor WuPalmer
    para cada linha A da MatrizPalmer faça
        para cada coluna B da MatrizPalmer faça
            Executa distância Wu & Palmer para par de palavras A e B
            Gera valor de similaridade sim
            MSim = sim
        fim
    fim
    Grava matriz MSim
até documento terminar;
```

Exemplo: Por meio da aplicação do *WordNet* com a medida de *Wu & Palmer* sobre os descendentes, que estão contidas no documento *DDes* (Listagem 3.12 - linha 1) é gerado a matriz *MSim* ilustrada na Figura 15 (alguns trechos).

Como as palavras são testadas todas com todas, a diagonal principal têm o valor 1, e os valores abaixo da diagonal principal não são considerados, tendo em vista que os valores se repetem acima da diagonal principal. Valores no intervalo de 0 a 1 podem ser observados, como os valores 0,58 e 0,23, ambos se referindo aos pares testados *URL* e *Journal*; *Author* e *Journal*,

[3] https://github.com/Sciss/ws4j

respectivamente.

Figura 15 – Matriz *MSim* da análise do conhecimento

	Sou	Tag	Titl	Yea	Aut	Sta	Jou	Mor	URL	BIB	BIBT	Sta	@Pu	Jour	Arti	Pagir	ELo	Au	Lang	Pub
SourceType	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tag	0.0	1.0	0.4	0.37	0.22	0.0	0.0	0.37	0.42	0.0	0.0	0.0	0.0	0.5	0.0	0.375	0.0	0.0	0.57	0.0
Title	0.0	0.0	1.0	0.33	0.2	0.0	0.0	0.33	0.47	0.0	0.0	0.0	0.0	0.66	0.0	0.333	0.0	0.0	0.5	0.0
Year	0.0	0.0	0.0	1.0	0.23	0.0	0.0	0.85	0.35	0.0	0.0	0.0	0.0	0.42	0.0	0.571	0.0	0.0	0.5	0.0
Author	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.23	0.2	0.0	0.0	0.0	0.0	0.23	0.0	0.235	0.0	0.0	0.26	0.0
StandardNumber1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalName	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Month	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.35	0.0	0.0	0.0	0.0	0.42	0.0	0.571	0.0	0.0	0.5	0.0
URL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.58	0.0	0.352	0.0	0.0	0.53	0.0
BIBTEXAbstract	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BIBTEXKeyWords	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
StandardNumber2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
@PubModel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.428	0.0	0.0	0.66	0.0
ArticleTitle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.5	0.0
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Language	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

3.4.4 Analisar similaridade de caracteres

Uma terceira subatividade presente na atividade *Avaliar a similaridade* é a análise da similaridade de caracteres, descrita nesta subseção. Esta análise leva em consideração a relação sintática das palavras. A função aplicada foi a de *Jaro Winkler* (WINKLER, 1990) que é uma extensão das técnicas de distância de edição. O algoritmo leva em consideração as inserções, exclusões e transposições, calculando o número de caracteres comuns das palavras. O *Winkler* é um melhoramento do algoritmo de *Jaro*, que aumenta a medida de correspondências entre nomes, combinando os caracteres iniciais.

Mais detalhes sobre o funcionamento de *Jaro Winkler* podem ser vistos na Subseção 2.3.1. Os valores gerados pela aplicação do algoritmo *Jaro Winkler* ficam no intervalo de 0 (zero) a 1 (um). Para poder determinar o grau de similaridade entre os elementos, o resultado de cada par de palavras é armazenado em uma matriz.

Entrada: Documento *DAnc* ou *DDes*

Atividade: Descrita pelo Algoritmo 7

Saída: Matriz *MSim*

Por meio do Algoritmo 7 é possível visualizar como o processo da aplicação da subati-

vidade *Analisar similaridade de caracteres* é realizado.

Algoritmo 7: Subatividade *Analisar similaridade de caracteres*

Entrada: Documento *DAnc* ou *DDes*
Saída: Matriz *MSim*

```
repita
    Carrega palavras do documento no vetor JaroWinkler
    para cada linha A da MatrizJaro faça
        para cada coluna B da MatrizJaro faça
            Executa distância de Jaro Winkler para par de palavras A e B
            Gera valor de similaridade sim
            MSim = sim
        fim
    fim
    Grava matriz MSim
até documento terminar;
```

Exemplo: Assim como no exemplo anterior, foi gerado uma matriz *MSim* (Figura 16) com os valores obtidos por meio da aplicação do *Jaro Winkler* sobre os descendentes, que estão contidas no documento *DDes* (Listagem 3.12 - linha 1). Como pode ser visto na Figura 16, os valores no intervalo de 0 a 1 são gerados na comparação entre os elementos.

Figura 16 – Matriz *MSim* da análise de caracteres

	Sou	Tag	Title	Yea	Aut	Stan	Jour	Mor	UR	BIBT	BIBT	Stan	@Pu	Jour	Arti	Pagi	ELoc	Aut	Lang	Pub
SourceType	1.0	0.0	0.43	0.39	0.48	0.5	0.58	0.43	0.0	0.39	0.39	0.5	0.43	0.57	0.63	0.0	0.46	0.42	0.48	0.61
Tag	0.0	1.0	0.51	0.52	0.0	0.46	0.47	0.0	0.0	0.46	0.46	0.46	0.0	0.0	0.0	0.62	0.47	0.0	0.63	0.46
Title	0.0	0.0	1.0	0.0	0.45	0.42	0.43	0.46	0.0	0.42	0.51	0.42	0.43	0.0	0.62	0.53	0.43	0.43	0.44	0.41
Year	0.0	0.0	0.0	1.0	0.47	0.54	0.39	0.0	0.0	0.0	0.44	0.54	0.0	0.46	0.38	0.45	0.44	0.45	0.45	0.43
Author	0.0	0.0	0.0	0.0	1.0	0.48	0.47	0.57	0.0	0.49	0.49	0.48	0.51	0.53	0.47	0.51	0.33	0.86	0.43	0.55
StandardNumber1	0.0	0.0	0.0	0.0	0.0	1.0	0.62	0.34	0.0	0.36	0.30	0.95	0.45	0.43	0.37	0.38	0.37	0.44	0.58	0.43
JournalName	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.52	0.0	0.38	0.38	0.62	0.43	0.87	0.50	0.41	0.40	0.41	0.62	0.48
Month	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.42	0.0	0.34	0.0	0.56	0.42	0.53	0.52	0.46	0.44	0.41
URL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.47	0.48	0.0
BIBTEXAbstract	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.67	0.36	0.39	0.40	0.37	0.28	0.44	0.43	0.39	0.44
BIBTEXKeyWords	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.30	0.34	0.0	0.26	0.39	0.44	0.50	0.39	0.44
StandardNumber2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.45	0.43	0.37	0.38	0.37	0.44	0.58	0.43
@PubModel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.50	0.29	0.47	0.40	0.47	0.49	0.57
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.48	0.32	0.45	0.46	0.49	0.49
ArticleTitle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.40	0.34	0.57	0.40	0.55
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.51	0.36	0.54	0.54
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.41	0.47	0.56
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.40	0.41
Language	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.44
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

3.4.5 Aplicar o cálculo de equivalência

O objetivo desta atividade é realizar o cálculo de equivalência para gerar uma única matriz (denominada *MEqu*) com os valores de equivalência. A matriz é o resultado da obtenção

dos valores gerados nas matrizes das subatividades *Analisar os radicais*, *Analisar similaridade de conhecimento* e *Analisar similaridade de caracteres*. Para realizar o cálculo de equivalência, ou seja, determinar os valores de similaridade, o método de tomada de decisões AHP (Processo Analítico Hierárquico) é utilizado (Seção 2.4). Este auxilia no processo para definir os pesos adotados para cada técnica de similaridade textual. Por meio de uma hierarquia de relevância, que pode ser vista em (SAATY, 2008), foram atribuídos os pesos 1, 2 e 3 para as técnicas de extrator de radicais, similaridade de conhecimento e similaridade de caracteres, respectivamente.

Algumas observações referentes aos valores obtidos nas matrizes *MSim* são importantes para a determinação das equivalências. Conforme as observações relatadas na Tabela 4, o cálculo da média ponderada é realizado quando mais um valor estiver no intervalo de 0 a 1, levando em consideração os pesos estabelecidos anteriormente. A média ponderada pode ser vista na Equação 3.1.

$$MediaPond = \frac{(pesoA * radical) + (pesoB * conhecimento) + (pesoC * caractere)}{6} \tag{3.1}$$

A definição do ponto de corte é por meio de um limiar, que é o fator que separa os elementos relevantes dos irrelevantes. Existem estudos que realizam definições quanto a isto, como pode ser visto em (SANTOS CARLOS A. HEUSER; WIVES, 2011). Como o intuito desta dissertação não é aprofundar critérios de escolha de pontos de corte, a determinação dos valores escolhidos (*Ponto de corte 1* = 0,6 e *Ponto de corte 2* = 0,5) foram determinados por meio de observações nos testes realizados.

Entrada: Matriz *MSim* de cada técnica de similaridade

Atividade: Descrita pelo Algoritmo 8

Saída: Matriz *MEqu*

Esta atividade tem seu funcionamento descrito por meio do Algoritmo 8.

Algoritmo 8: Atividade *Aplicar o cálculo de equivalência*

Entrada: Matriz *MSim* de cada técnica de similaridade
Saída: Matriz *MEqu*

```
repita
    para cada elemento da matriz faça
        se Radical ou Jaro ou WuP for igual a 1 então
            MEqu=1
        fim
        se Radical e Jaro e WuP for igual a 0 então
            MEqu=0
        fim
        se Apenas uma medida for diferente de 0 então
            se elemento > ponto de corte 1 então
                MEqu=1
            fim
            senão
                MEqu=0
            fim
        fim
        senão
            Aplica média ponderada com pesos definidos pelo método AHP
            se resultado > ponto de corte 2 então
                MEqu=1
            fim
            senão
                MEqu=0
            fim
        fim
    fim
    Grava matriz MEqu
até documento terminar;
```

Exemplo: As matrizes *MSim* ilustradas nas Figuras 14, 15 e Figura 16 representam as diferentes técnicas de similaridade aplicadas à algumas palavras descendentes (oriundas da extração dos pares de ancestrais *Source* e *Article*). Por meio da aplicação do cálculo de equivalência, uma única matriz *MEqu* é gerada para aquele conjunto de elementos, conforme pode ser observado na Figura 17.

Os elementos em destaque são aqueles que obtiveram valor 1, ou seja, considerados equivalentes. Considerando que [*i*] e [*j*] se referem respectivamente à linha e coluna das matrizes, o elemento *MEqu*[*1*][*18*] (par de elementos *Tag* e *Language*) obteve valor 1 pois as matrizes *MSim* de cada técnica aplicada obtiveram: *Radical*[*1*][*18*] = 0; *WuP*[*1*][*18*] = 0,571 e *Jaro*[*1*][*18*] = 0,638. Como mencionado anteriormente, quando mais de uma medida possui valores no intervalo de 0 a 1 é aplicado a média ponderada. Aplicando a Equação 3.1 nos

Figura 17 – Matriz *MEqu*

	Sou	Tag	Titl	Yea	Aut	Sta	Jou	Mo	UR	BIB	BIB	Sta	@P	Jou	Art	Pag	ELo	Aut	Lan	Pub
SourceType	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
Tag	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Title	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
Year	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
StandardNumber1	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalName	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
Month	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
URL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BIBTEXAbstract	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BIBTEXKeyWords	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
StandardNumber2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
@PubModel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
ArticleTitle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Language	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

elementos deste exemplo obtém-se o seguinte cálculo:

$$MediaPond = \frac{(1*0)+(2*0{,}571)+(3*0{,}638)}{6}$$

$$MediaPond = 0,51$$

Observando que o valor obtido é maior que o ponto de corte de 0,5, atribui-se o valor 1, ou seja, o par de elementos é considerado equivalente.

3.4.6 Extrair os elementos

O principal objetivo desta atividade é retirar os elementos contidos nas matrizes de equivalências (*MEqu*). Os elementos podem ser designados para diferentes documentos, dependendo se forem ancestrais ou descendentes. É importante observar que os pares de palavras que possuem valor 1 (um) são considerados equivalentes e os não equivalentes são aqueles com valor 0 (zero).

Os três documentos que podem ser obtidos na saída desta atividade são utilizados para a fase atual (2ª fase) e para as próximas. Uma das saídas são os ancestrais considerados equivalentes, que são utilizados para extraírem seus descendentes. As outras duas saídas são os descendentes classificados como equivalentes e não equivalentes. A partir deste momento os elementos ficam contidos em documentos únicos, não sendo separados pelos seus pares.

Entrada: Matriz *MEqu*

Atividade: Descrita pelo Algoritmo 9

Saída: Documentos *DCan*, *DDesN* e *DDesC*

Por meio do Algoritmo 9, a atividade é detalhada, possibilitando um melhor entendimento sobre seu funcionamento.

Algoritmo 9: Atividade *Extrair os elementos*

```
Entrada: Matriz MEqu
Saída: Documentos DCan, DDesN e DDesC
repita
    Carrega matriz de resultados MEqu
    se matriz MEqu possui elementos ancestrais então
        Carrega vetor de palavras ancestrais
        senão
            Carrega vetor de palavras descendentes
        fim
    fim
    para cada linha A da matriz MEqu faça
        para cada coluna B da matriz MEqu faça
            se MEqu na linha A e coluna B for igual a 1 então
                Equivalentes = vetor na linha A e vetor na coluna B
                se vetor de descendentes então
                    DDesC=Equivalentes
                    senão
                        DCan=Equivalentes
                    fim
                fim
            fim
            senão
                DDesN = vetor na linha A e vetor na coluna B
            fim
        fim
    fim
    Grava documento de texto DDesN
    Grava documento de texto DDesC
    Grava documento de texto DCan
até documento terminar;
```

Exemplo: Como resultado da aplicação da atividade de *Extrair os elementos* sobre as matrizes *MEqu* podem ser gerados três diferentes documentos, como mencionado anteriormente. Tendo como base as matrizes geradas nos exemplos anteriores, a Listagem 3.17 ilustra os elementos descendentes equivalentes, ou seja, são aqueles que obtiveram valor 1 na matriz *MEqu* da Figura 17.

```
1 SourceType - ArticleTitle,
2 SourceType - PublicationTypeList,
```

```
Tag - Language,
Title - Journal,
Title - ArticleTitle,
Year - Month,
Author - AuthorList,
StandardNumber1 - JournalName,
StandardNumber1 - StandardNumber2,
JournalName - StandardNumber2,
JournalName - Journal,
JournalName - Language,
BIBTEXAbstract - BIBTEXKeyWords
```

Listagem 3.17 – Documento dos elementos descendentes equivalentes/candidatos (*DDesC*)

Os outros documentos resultantes desta atividade (elementos descendentes não candidatos e os ancestrais candidatos) podem ser observados nos exemplos citados anteriormente nas Listagens 3.11 e 3.14.

3.4.7 Extrair os elementos descendentes

A extração dos elementos descendentes, onde seus elementos ancestrais foram considerados equivalentes, é o objetivo proposto por esta atividade. Descobrir a similaridade dos elementos ancestrais é essencial para poder determinar seus elementos descendentes. Considerar a hierarquia dos documentos JSON é de suma importância, visto que estes possuem estruturas composta por diversos níveis em um mesmo documento. O documento *DEst* é considerado para extrair os elementos descendentes.

Esta atividade tem por objetivo extrair todos os elementos descendentes pertencentes ao par de ancestrais considerados equivalentes. É importante destacar que foi considerado nesta dissertação apenas um nível de hierarquia, ou seja, os elementos descendentes possuem um nível acima de ancestral (elemento pai).

Entrada: Documentos *DCan* e *DEst*

Atividade: Descrita pelo Algoritmo 10

Saída: Documento *DDes*

Para ilustrar como a atividade de *Extrair os elementos descendentes* funciona para retirar os elementos descendentes dos pares de ancestrais equivalentes, o Algoritmo 10 é descrito a seguir.

Algoritmo 10: Atividade *Extrair os elementos descendentes*

Entrada: Documentos *DCan* e *DEst*
Saída: Documento *DDes*

repita
  Carrega documento *DEst*
  para *cada par de ancestrais do documento DCan* **faça**
    se *ancestral possui descendente conforme documento DEst* **então**
      Grava descendentes
    fim
  fim
  Grava documento de texto *DDes*
até *documento terminar*;

Exemplo: Para demonstrar os elementos extraídos dos ancestrais *Source* e *Article* (já mencionado em exemplos anteriores), a Listagem 3.12 (linha 1) ilustra os descendentes pertencentes a esse par de ancestrais. Alguns trechos dos documentos JSON (Documento 0 e 1 das Listagens 3.4 e 3.5) são ilustrados na Listagem 3.18 e 3.19 para demonstrar a hierarquia presente nos documentos, observando seus elementos descendentes.

```
1  {
2  ...
3  "Source": [
4        {
5          "SourceType": ["JournalArticle"],
6          "Tag": ["hu2019synoptic"],
7          "Title": [ "A synoptic assessment of the summer extreme rainfall
       over the middle reaches of Yangtze River in CMIP5 models"],
8          "Year": ["2019"],
9          "Author": [
10 ...
```

Listagem 3.18 – Trecho do Documento 0 demonstrando os elementos descendentes de *Source*

```
1  ...
2  "Article": [
3        {
4        "@PubModel": "Print",
5        "Journal": [{
6                  ...
7                      }]
8        "ArticleTitle": [{}],
9        "Pagination": [{...}],
10       "ELocationID": [{
11                 ...
12                     }],
13       "AuthorList": [{
14                                    ...
15                      }],
16        "Language": ["eng"],
17        "PublicationTypeList": [{...}]
18 ...
```

Listagem 3.19 – Trecho do Documento 1 demonstrando os elementos descendentes de *Article*

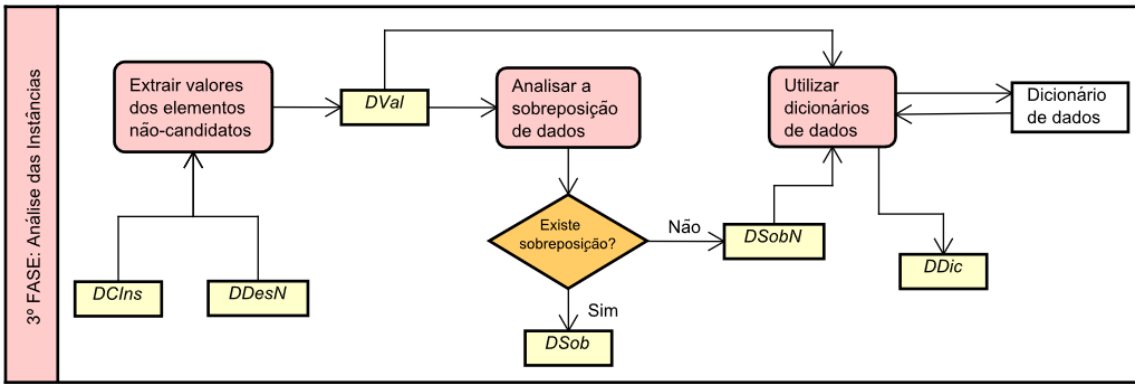
3.5 ANÁLISE DAS INSTÂNCIAS - 3ª FASE

Os elementos descritos, implementados e testados até o momento dizem respeito ao nome dos campos presentes nos documentos JSON. A fase apresentada nesta seção leva em consideração as instâncias presentes nos documentos JSON, ou seja, os valores que os campos podem possuir. A Figura 18 ilustra a 3º Fase de um modo geral, mostrando que esta fase utiliza os documentos *DCIns* e *DDesN* para iniciar as atividades da análise das instâncias dos elementos descendentes que não obtiveram similaridades em seus campos.

Na fase anterior os elementos descendentes foram analisados quanto à similaridade presente em seus campos, determinando os elementos candidatos ao casamento de esquemas e os que não obtiveram resultados de similaridade, ou seja, não candidatos. Com o objetivo de verificar se existem similaridades nas instâncias dos elementos descendentes não candidatos, esta fase considera os valores desses campos. Isto é relevante pois pode acontecer de os nomes dos campos serem totalmente diferentes, mas seus valores podem estar relacionados ao mesmo conteúdo.

Para realizar esta análise são utilizados dois tipos de verificações: a sobreposição de dados e uso de dicionários de dados. Este só é utilizado quando o *dataset* disponibilizar informações sobre os mesmos. Em um primeiro momento é realizado a análise da sobreposição dos dados e aqueles que não obtiveram bons resultados são passados para a análise utilizando dicionário de dados, se disponível.

Figura 18 – Análise das Instâncias



Fonte: Autor

Entrada: Documentos *DCIns* e *DDesN* gerados nas fases anteriores para auxiliar na extração dos valores dos campos.

Atividade Extrair os valores dos elementos não-candidatos: O objetivo desta atividade é identificar e extrair as instâncias pertencentes aos campos de descendentes considerados não equivalentes na 2ª fase. O resultado é um documento denominado *DVal* contendo as instâncias.

Atividade Analisar a sobreposição de dados: Esta atividade faz uma verificação se os elementos possuem sobreposição de dados, para isto utiliza o documento *DVal* com os valores das instâncias. Os resultados são analisados em relação a possuírem sobreposição ou não.

Atividade Utilizar dicionário de dados: As instâncias que não obtiverem sobreposição nos seus valores podem ser analisados com a utilização de um dicionário de dados. Para isto ocorrer é necessário que o *dataset* possua/disponibilize um dicionário de dados com informações relacionadas aos documentos JSON.

Saída: Os artefatos de saída são dois documentos denominados *DSob* e *DDic* relacionados, respectivamente, aos elementos que possuem sobreposição de dados e os que são equivalentes considerando o dicionário de dados.

Exemplo: Os dois documentos decorrentes da 1ª fase (*DCIns*) e da 2ª fase (*DDesN*) podem ser observados nas Listagens 3.8 e 3.14, respectivamente. Por meio da aplicação e utilização desses documentos, a atividade *Extrair os valores dos elementos não-candidatos* realiza verificações e comparações para extrair as instâncias pertencentes àqueles campos de *DDesN*.

A Listagem 3.20 ilustra um trecho do documento *DVal*, resultado da aplicação da atividade *Extrair os valores dos elementos não-candidatos*. É possível observar que em alguns pares de elementos descendentes não são extraídos os seus valores, isto por que alguns campos não possuem instâncias. Assim como pode ocorrer de alguns campos possuírem mais de uma instância, havendo, assim, mais comparações entre os campos.

```
...
{
Campo: SourceType
Campo: Tag

valor B[0] = hu2019synoptic;
valor A[0] = JournalArticle2;
valor B[0] = hu2019synoptic;
valor A[1] = JournalArticle;
valor B[1] = gonzalez2019contribution;
valor A[0] = JournalArticle2;
valor B[1] = gonzalez2019contribution;
valor A[1] = JournalArticle;
};
{
Campo: Tag
Campo: StandardNumber1

valor B[0] =  ISSN: 1432-0894 DOI: 10.1007/s00382-019-04803-3;
```

```
valor A[0] = hu2019synoptic;
valor B[0] =  ISSN: 1432-0894 DOI: 10.1007/s00382-019-04803-3;
valor A[1] = gonzalez2019contribution;
}
...
```

Listagem 3.20 – Trecho do documento *DVal* com as instâncias extraídas dos campos descendentes

Com as instâncias obtidas, a próxima atividade aplicada é *Analisar a sobreposição de dados*. A Listagem 3.21 mostra alguns elementos que possuem sobreposição, com seus campos e instâncias correspondentes, assim como a medida utilizada para medir a sobreposição de dados.

```
...
Par dos campos: Title - BIBTEX_KeyWords
Par dos valores: A synoptic assessment of the summer extreme rainfall over
   the middle reaches of Yangtze River in CMIP5 models; -
    CMIP, China, circulation, climatology, dynamics, eastasianmonsoon,
   extremes, precip;

JaccardSimilarity 0.75

...
```

Listagem 3.21 – Exemplo de elementos com sobreposição de dados

O documento *DSob* é o resultado da análise dos elementos que obtiveram sobreposição de dados ao longo da execução das verificações. Se não existir sobreposição de dados um dicionário é utilizado, quando o mesmo estiver disponível na aplicação.

O detalhamento das atividades da 3ª fase (Figura 18) pode ser explorada em mais detalhes nas próximas subseções. Por meio de explicações, exemplos e algoritmos é possível entender de uma maneira mais aprofundada como as atividades funcionam.

3.5.1 Extrair valores dos elementos não-candidatos

Esta atividade necessita de documentos que foram geradas em fases anteriores, para realizar seu objetivo, que é verificar e determinar quais são as instâncias presentes nos campos de descendentes que foram considerados não equivalentes nas análises anteriores.

Para atingir este objetivo foi utilizado a API Java JSR 353, para determinar as instâncias de cada campo. Foi necessário também, haver uma comparação entre os elementos presentes no documento *DCIns* e os elementos de *DDesN*.

Entrada: Documento com os campos e valores (*DCIns*) e o documento com os elementos descendentes não equivalentes (*DDesN*)

Atividade: Descrita pelo Algoritmo 11

Saída: Documento com os valores de instâncias *DVal*

O funcionamento da atividade de *Extrair valores dos elementos não-candidatos* é descrita em detalhes pelo Algoritmo 11.

Algoritmo 11: Atividade *Extrair valores dos elementos não-candidatos*

Entrada: Documentos *DCIns* e *DDesN*
Saída: Documento *DVal*

repita
  para *cada elemento não candidato de DDesN* **faça**
    se *elemento está contido em DCIns* **então**
      se *elemento possui valor* **então**
        Grava valor
      fim
    fim
  fim
  Grava documento de texto *DVal*
até *documento terminar*;

Exemplo: Para ilustrar as instâncias retiradas de determinados campos (*SourceType* e *Tag*; *Tag* e *StandarNumber1*; etc), a Listagem 3.20 descreve trechos do documento *DVal*. É importante notar que em alguns casos nem todos os campos possuem valores extraídos, isto por que os mesmos não possuem instâncias em seus documentos JSON originais.

Na Listagem 3.22 é possível visualizar um trecho do documento *DVal* onde os campos possuem mais de uma instância. Sendo assim, todas essas instâncias são testadas. É possível observar que o campo *SourceType* possui dois elementos e *Title* possui três.

```
...
{
Campo: SourceType
Campo: Title

valor B[0] = The Journal of hand surgery;
valor A[0] = JournalArticle2;
valor B[0] = The Journal of hand surgery;
valor A[1] = JournalArticle;
valor B[1] = A synoptic assessment of the summer extreme rainfall over the
    middle reaches of Yangtze River in CMIP5 models;
valor A[0] = JournalArticle2;
valor B[1] = A synoptic assessment of the summer extreme rainfall over the
    middle reaches of Yangtze River in CMIP5 models;
valor A[1] = JournalArticle;
valor B[2] = The contribution of North Atlantic atmospheric circulation
    shifts to future wind speed projections for wind power over Europe;
valor A[0] = JournalArticle2;
valor B[2] = The contribution of North Atlantic atmospheric circulation
    shifts to future wind speed projections for wind power over Europe;
valor A[1] = JournalArticle;
```

```
18 }
19 ...
```

Listagem 3.22 – Exemplo de campos que possuem mais de uma instância

3.5.2 Analisar a sobreposição dos dados

O principal objetivo encontrado na aplicação desta atividade é tentar encontrar similaridades entre as instâncias dos campos que foram considerados não equivalentes na fase anterior. Observar os dados, ou seja, as instâncias (ou valores) é algo importante na obtenção de informações sobre determinados campos.

Uma das técnicas mais populares na aplicação de sobreposição de dados é a medida de *Jaccard*, que é descrita na Subseção 2.3.1. Os pares de instâncias extraídas, conforme seus descendentes, são executadas com a medida de *Jaccard*. Dependendo do valor obtido são considerados sobrepostos ou não. O valor de limiar utilizado foi de 0.55, sendo este escolhido por meio de observações quanto aos casos obtidos com valores maiores ou menores.

Entrada: Documento das instâncias *DVal*

Atividade: Descrita pelo Algoritmo 12

Saída: Documentos *DSob* e *DSobN*

Os detalhes de como funciona a atividade de *Analisar a sobreposição dos dados* é demonstrada a seguir no Algoritmo 12.

Algoritmo 12: Atividade *Analisar a sobreposição dos dados*

Entrada: Documento *DVal*
Saída: Documentos *DSob* e *DSobN*

```
repita
    para cada par de instâncias de DVal faça
        Aplica cálculo de sobreposição Jaccard
        se medida de Jaccard > 0.55 então
            Grava campos com sobreposição de dados nas instâncias
        fim
        senão
            Grava campos sem sobreposição de dados nas instâncias
        fim
    fim
    Grava documento de texto DSob
    Grava documento de texto DSobN
até documento terminar;
```

Exemplo: A Listagem 3.21 mostra parte da execução da atividade *Analisar a sobreposição dos dados*. Essas informações sobre os pares de campos que obtiveram sobreposições de

seus dados são descritas em detalhes para futuras verificações, se necessárias.

Um documento final denominado de *DSob* (Listagem 3.23) é gerado com os pares de campos que obtiveram valores de sobreposição de instâncias maiores que 0,55 na medida de *Jaccard*. Os pares de campos que não obtiveram medidas de sobreposição satisfatórias são armazenados no documento *DSobN* e designados para a atividade que verifica a possibilidade de utilização de um dicionário de dados.

```
[Year - Day,
$a - Year,
ISOAbbreviation - MedlineTA,
$a - @PubStatus,
Day - Year,
SourceType - Tag,
Year - $a,
First - @ValidYN,
Month - Hour]
```

Listagem 3.23 – Documento *DSob* com os campos que suas instâncias obtiveram medidas de sobreposição satisfatórias

3.5.3 Utilizar dicionário de dados

O objetivo desta atividade é utilizar um dicionário de dados para verificar se as instâncias dos campos correspondem aos valores disponíveis no dicionário. O uso de dicionário de dados é um benefício que pode acarretar em uma melhor aproximação dos valores equivalentes. Um pequeno dicionário de dados pode listar todos os possíveis valores para determinados campos.

Com a utilização do documento *DVal* (campos e suas instâncias) e os campos que não obtiveram sobreposição de dados (documento *DSobN*), é verificado se o *dataset* possui algum dicionário de dados para auxiliar na verificação das similaridades. O resultado desta atividade é um documento *DDic* com os elementos considerados equivalentes determinados pela consulta ao dicionário de dados.

Entrada: Documentos *DSobN* e *DVal*

Atividade: Descrita pelo Algoritmo 13

Saída: Documento *DDic* com os elementos considerados similares

Por meio do Algoritmo 13 é possível visualizar em detalhes como funciona a atividade *Utilizar dicionário de dados*.

Algoritmo 13: Atividade *Utilizar dicionário de dados*

Entrada: Documento *DSobN* e *DVal*
Saída: Documento *DDic*

```
repita
    se possui dicionário de dados então
        para cada par de elementos sem sobreposição DSobN faça
            se elementos estão contidos em DVal então
                para cada par de instâncias faça
                    Consulta dicionário
                    Grava campos equivalentes
                fim
            fim
        fim
    fim
    Grava documento de texto DDic
até documento terminar;
```

Exemplo: Para exemplificar como esta atividade funciona, um dicionário de dados foi escolhido para ser utilizado. Um conjunto de dados com as principais cidades dos Estados Unidos foi usado para verificar campos que possuíssem nomes de cidades. A Listagem 3.24 ilustra o dicionário de dados retirado do *site data world*[4].

```
Rank,City,Population (2013),Mayor,Budget,Elections in 2017?
1,"New York, New York","8,405,837",Bill de Blasio (D),"$73,000,000,000",Yes
2,"Los Angeles, California","3,884,307",Eric Garcetti (D),"$8,100,000,000",
   Yes
3,"Chicago, Illinois","2,718,782",Rahm Emanuel (D),"$7,300,000,000",Yes
4,"Houston, Texas","2,195,914",Sylvester Turner (D),"$5,100,000,000",No
5,"Philadelphia, Pennsylvania","1,553,165",James Kenney (D),"$3,950,000,000
   ",Yes
6,"Phoenix, Arizona","1,513,367",Greg Stanton (D),"$3,700,000,000",Yes
7,"San Antonio, Texas","1,409,019",Ivy R. Taylor (D),"$2,400,000,000",Yes
8,"San Diego, California","1,355,896",Kevin Faulconer (R),"$3,200,000,000",
   No
9,"Dallas, Texas","1,257,676",Mike Rawlings (D),"$2,800,000,000",Yes
10,"San Jose, California","998,537",Sam Liccardo (D),"$3,000,000,000",No
11,"Austin, Texas","885,400",Stephen Adler (D),"$3,500,000,000",No
12,"Indianapolis, Indiana","843,393",Joseph Hogsett (D),"$1,030,000,000",No
13,"Jacksonville, Florida","842,583",Lenny Curry (R),"$1,040,000,000",No
14,"San Francisco, California","837,442",Edwin M. Lee (D),"$9,600,000,000",
   No
15,"Columbus, Ohio","822,553",Andrew J. Ginther (D),"$814,000,000",Yes
...
```

Listagem 3.24 – Conjunto de dados das principais cidades para verificações

Suponha que os campos *city* e *location* sejam verificados por esta atividade. Com a utilização do conjunto de dados descrito na Listagem 3.24, é possível determinar que os dois campos possuem instâncias que podem ser localizados no mesmo conjunto de dados, ou seja, se

[4] https://data.world/government/mayors-of-top-100-us-cities

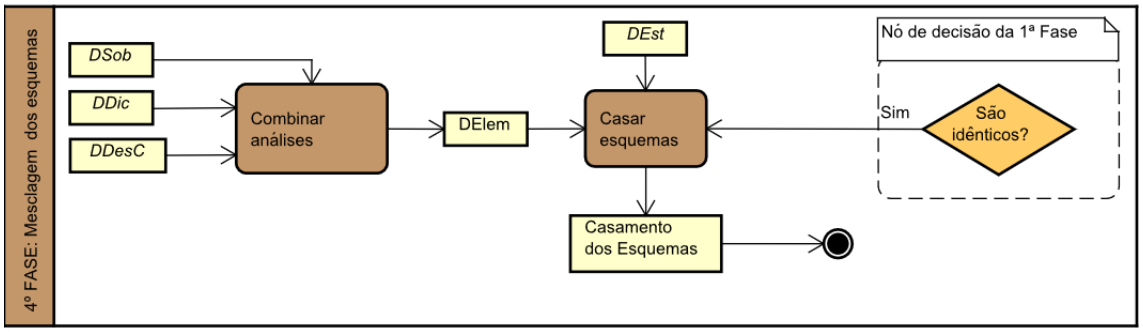
referem à nomes de cidades. O uso desta atividade depende se o *dataset* possui algum dicionário de dados para realizar as verificações.

3.6 MESCLAGEM DOS ESQUEMAS - 4ª FASE

A última fase realiza o casamento dos esquemas propriamente dito. Por meio das análises e verificações executadas nas fases anteriores, são realizadas atividades que mesclam e/ou combinam os elementos, para que o casamento dos esquemas seja concretizado.

Como mencionado anteriormente, o casamento dos esquemas está relacionado à determinação de quais elementos podem ser unificados. Essa determinação é essencial para dar andamento à uma integração de esquemas e consequentemente à uma integração de dados. As análises realizadas na 2ª fase, com a geração dos elementos descendentes equivalentes, e na 3ª fase, por meio da análise das instâncias, a 4ª fase realiza a combinação destes elementos e determina quais documentos de origem os elementos pertencem.

Figura 19 – Mesclagem dos Esquemas



Fonte: Autor

Entrada: Os documentos extraídos nas fases anteriores (*DSob, DDic* e *DDesC*).

Atividade Combinar análises: Com a obtenção dos elementos considerados equivalentes (por diferentes atividades das fases anteriores), esta atividade realiza a combinação dos elementos, ou seja, os elementos são mesclados para serem analisados na atividade seguinte. Essa combinação dos elementos nada mais é do que unificar os elementos considerados equivalentes oriundos de diferentes atividades.

Atividade Casar esquemas: Com os elementos equivalentes é possível determinar quais são os documentos de origem que cada um pertence e determinar o casamento dos esquemas. Para definir a quais documentos de origem os elementos pertencem é utilizado o documento

DEst com a estrutura dos documentos JSON, gerados na 1ª fase. O resultado desta atividade são as informações dos esquemas, ou seja, a determinação de quais elementos que obtiveram equivalências (o casamento dos esquemas).

Saída: O artefato de saída desta fase é o casamento dos esquemas, ou seja, uma descrição dos elementos considerados equivalentes.

Exemplo: Para exemplificar as atividades realizadas nesta fase alguns documentos são utilizados. A Listagem 3.13 contém os elementos descendentes candidatos (*DDesC*), já a Listagem 3.23 descreve quais campos obtiveram equivalências em suas instâncias (*DSob*). Neste exemplo não é considerado o documento *DDic*, decorrente da utilização de dicionário de dados pelo motivo do conjunto de dados testado não possuir.

A atividade de *Combinar análises* mescla as duas Listagens (3.13 e 3.23), como pode ser visto na Listagem 3.25, gerando o documento *DElem*.

```
[
SourceType - ArticleTitle
SourceType - PublicationTypeList
Tag - Language
Title - Journal
Title - ArticleTitle
Year - Month
Author - Author,
...
Year - Day,
ISOAbbreviation - MedlineTA,
Day - Year,
SourceType - Tag,
...
```

Listagem 3.25 – Documento com os elementos equivalentes mesclados *DElem*

A determinação dos documentos de origem dos elementos contidos na Listagem 3.25 são extraídos com o auxílio do documento *DEst*, que pode ser visto na Listagem 3.6. O resultado da atividade *Casar esquemas* é ilustrado na Listagem 3.26, ou seja, o casamento dos esquemas.

```
[
Documento 0 -> SourceType - Documento 1 -> ArticleTitle,
Documento 0 ->SourceType - Documento 1 -> PublicationTypeList,
Documento 0 ->Tag - Documento 1 -> Language,
Documento 0 ->Title - Documento 1 -> Journal,
Documento 0 ->Title - Documento 1 -> ArticleTitle,
Documento 0 ->Year - Documento 0 -> Month,
Documento 0 -> Author - Documento 1 -> Author,
...]
```

Listagem 3.26 – Casamento dos esquemas

As próximas subseções detalham as atividades da 4ª fase (Figura 19) para um maior entendimento das mesmas. Com a utilização de exemplos e algoritmos é possível entender de

uma maneira mais aprofundada como as atividades funcionam.

3.6.1 Combinar análises

O objetivo desta atividade é realizar uma mesclagem dos elementos considerados equivalentes nas fases anteriores. Essa combinação é necessária para utilizar tanto os elementos descendentes equivalentes quanto os elementos que obtiveram equivalência em suas instâncias. Quanto ao documento que contém os elementos que foram testados com um dicionário de dados (*DDic*), em alguns casos, o mesmo não possui conteúdo devido à indisponibilidade de um dicionário de dados no *dataset*.

Realizar esta atividade é importante para dar sequência ao processo e consequentemente finalizar o mesmo. A atividade descrita nesta subseção se torna algo relativamente simples, pois trata-se de uma união entre todos os resultados obtidos no decorrer de todo o processo. A seguir são descritos mais detalhes sobre esta fase.

Entrada: Documentos *DSob*, *DDic* e *DDesC* com os campos de sobreposição, do dicionário de dados e os descendentes equivalentes, respectivamente.

Atividade: Descrita pelo Algoritmo 14.

Saída: Documento *DElem* com os elementos unificados.

O Algoritmo 14 ilustra, em detalhes, como funciona a atividade *Combinar análises*, para uma melhor compreensão e entendimento.

Algoritmo 14: Atividade *Combinar análises*

```
Entrada: Documentos DSob, DDic e DDesC
Saída: Documento DElem
repita
    para cada documento de entrada faça
        se possui elementos então
            Grava par de campos equivalentes
        fim
    fim
    Grava documento de texto DElem
até documento terminar;
```

Exemplo: O funcionamento da atividade *Combinar análises* foi explorada anteriormente na Listagem 3.25, trazendo como resultado o documento *DElem*, contendo a mesclagem dos elementos descendentes considerados equivalentes na saída da 2ª fase e os elementos equivalentes considerando as suas sobreposições das instâncias (saída da 3ª fase). É possível observar que entre as linhas 2 e 8 são os pares de campos do documento *DDesC* e entre as linhas

10 e 13 são os elementos contidos no documento *DSob*. Como mencionado anteriormente, este exemplo não aborda o documento *DDic* resultante da utilização de um dicionário de dados.

3.6.2 Casar esquemas

A última atividade do processo para casamento de esquemas informa os campos considerados equivalentes e à quais documentos JSON de origem cada campo pertence. Essas informações são essenciais para que as fases da integração de esquemas (vistas na subseçao 2.1.2) sejam realizadas, e consequentemente a integração de dados de um modo geral.

Com o auxílio do documento *DEst*, que contém o caminho dos campos, é realizado a verificação de quais são os documentos de origem dos campos considerados equivalentes, gerando as informações dos esquemas. Assim o casamento dos esquemas é realizado. É importante notar que quando há coleções de documentos que possuem mais de dois documentos JSON, os mesmos são processados ao mesmo tempo (de uma única vez), para a geração do casamento dos mesmos. Relatar estas informações é essencial para que, por exemplo, a integração de esquemas prossiga.

Outro ponto a ser observado é que o nó de decisão localizado na 1ª fase pode ser direcionado para esta atividade, tendo em vista que os documentos JSON podem ser idênticos. Neste caso os mesmos podem ser representados de forma única.

Entrada: Documentos *DElem* e *DEst*, ou seja, os campos equivalentes e a estrutura dos campos.

Atividade: Descrita pelo Algoritmo 15.

Saída: O *Casamento dos esquemas*.

O funcionamento da atividade de *Casar esquemas* pode ser visualizada em detalhes no Algoritmo 15.

Algoritmo 15: Atividade *Casar esquemas*

Entrada: Documentos *DElem* e *DEst*
Saída: O *Casamento dos esquemas*

```
repita
    se esquemas pertencem ao nó de decisão então
        Grava primeiro documento JSON
    fim
    senão
        para cada par de elementos do documento DElem faça
            se par de elementos está contido em DEst então
                Grava par de campos equivalentes com seus documentos de origem;
            fim
        fim
    fim
    Grava Casamento dos esquemas;
até montar o Casamento dos esquemas;
```

Exemplo: O processo de um modo geral chega ao final com esta atividade. Como mencionado anteriormente, um trecho do casamento dos esquemas pode ser visto na Listagem 3.26, onde os elementos considerados equivalentes recebem informações quanto aos seus documentos JSON de origem, sendo essenciais para futuras verificações.

3.7 CONSIDERAÇÕES FINAIS

As subseções tratadas neste capítulo se referem ao objetivo principal desta dissertação. Por meio de explicações e exemplos é possível compreender como todo o processo para casamento de esquemas funciona. O detalhamento das fases presentes no processo é essencial para obter um melhor entendimento.

De uma forma geral, o processo para casamento de esquemas inicia na 1ª fase com um pré-casamento dos esquemas, ou seja, realiza um pré-processamento dos documentos JSON, determinando se os mesmos são idênticos, assim como são extraídas informações essenciais para as fases seguintes. Sequencialmente, na 2ª fase, uma análise dos esquemas é realizada, tendo como principal objetivo testar os campos quanto às suas equivalências decorrente das análises dos radicais, dos caracteres e do conhecimento. Primeiramente os elementos ancestrais são testados, sendo que os considerados equivalentes extraem seus elementos descendentes para serem também analisados.

Na 3ª fase uma análise das instâncias é realizada para auxiliar na determinação dos elementos equivalentes, pois o processamento desta fase é realizado com base nos elementos

considerados não equivalentes na 2ª fase. Uma extração das instâncias dos elementos é realizada, aplicando a sobreposição de dados sobre os mesmos, assim como o uso de dicionário de dados (quando disponível). E para finalizar o processo, a 4ª fase, realiza uma espécie de mesclagem dos elementos considerados equivalentes decorrente das outras fases. Um documento contendo estas informações é gerado, ou seja, o casamento dos esquemas propriamente dito.

Os exemplos apresentados neste capítulo são reduzidos por questões de espaço, sendo que no estudo de caso (tratado no Capítulo 4) mais detalhes podem ser analisados. É importante ressaltar que as fases são compostas por entradas, atividades e saídas, sendo que algumas atividades possuem subatividades.

4 AVALIAÇÃO DE RESULTADOS

A análise de quanto o processo para casamento de esquemas foi benéfico, assim como seu funcionamento quanto às implementações, pode ser visualizada neste capítulo. A implementação descrita está relacionada aos códigos de execução realizados em linguagem Java, destacando as principais classes e métodos utilizados.

Um estudo de caso é abordado, tendo em vista, a sua importância no momento de descrever o processo com exemplos de documentos JSON maiores. Por meio de descrições é possível acompanhar o passo a passo que é realizado entre as atividades contidas nas fases. Os resultados obtidos são analisados na seção de validação por meio de medidas de revocação e precisão.

4.1 IMPLEMENTAÇÃO

Algumas das atividades descritas no Capítulo 3 são realizadas de forma manual, mas existem algumas que foram implementadas em linguagem Java. Pode-se concluir que o processo para casamento de esquemas é executado de forma semiautomática, necessitando de um especialista para conduzir o processo. As atividades que foram realizadas manualmente, não são demonstradas nesta seção. Todo passo a passo da aplicação do processo em uma coleção de documentos JSON pode ser visto na Seção 4.2.

Diferentes classes foram criadas para descrever as atividades, além disso inúmeros métodos foram utilizados para atender às necessidades de cada atividade e/ou fase. A demonstração da implementação é descrita separadamente por atividades presentes no processo. Mais detalhes de como cada atividade funciona pode ser observada no Capítulo 3.

A classe principal da implementação, além de instanciar as outras classes, realiza a leitura da coleção de documentos JSON contidos em determinadas pastas. A classe principal também grava os documentos *DEst* e *DCIns*. Primeiramente na Listagem 4.1 é ilustrado o trecho onde são lidos os documentos JSON, sendo possível observar a utilização do *JsonReader* e *JsonStructure*. Este é um supertipo que abrange os dois tipos estruturados em JSON (objetos e matrizes), e aquele lê um objeto JSON ou uma estrutura de matriz de uma fonte de entrada.

```
1 public class CasamentoJSON{
2
3     public static void main(String[] args) throws FileNotFoundException,
    IOException {
4 ...
5 //-------------Percorre os documentos JSON na pasta
```

```
6         for (m=0;m<arquivo.length;m++){
7
8             reader = Json.createReader(new FileReader(arquivo[m]));
9             JsonStructure lendoArq = reader.read();
10 ...
11     }
12 }
```

Listagem 4.1 – Trecho da classe principal onde os documentos JSON são lidos

Para determinar a estrutura dos documentos JSON, ou seja, gravar o documento *DEst* foi utilizado o JSONPath[5]. Como mencionado anteriormente (Subseção 3.3.3), o uso do JSONPath resulta em expressões de caminho de cada campo. Para obter todas as expressões é utilizado a opção de listar todo o caminho (*Option.AS_PATH_LIST*) desde a raiz, representada pelo símbolo $. A Listagem 4.2 ilustra o trecho onde é aplicado o JSONPath.

```
1 public class CasamentoJSON{
2
3     public static void main(String[] args) throws FileNotFoundException,
      IOException {
4 ...
5 //-------------DEst
6             Configuration conf = Configuration.builder().options(Option.
      AS_PATH_LIST).build();
7             ArrayList<String> Caminho= new ArrayList();
8             Caminho = JsonPath.using(conf).parse(lendoArq).read("$..*");
9
10             for (int zz=0;zz<Caminho.size();zz++){
11                 caminhos.add("\n Doc: "+m);
12                 caminhos.add(Caminho.get(zz));
13             }
14 ...
15     }
16 }
```

Listagem 4.2 – Utilização do JSONPath para determinar o documento *DEst*

A construção do documento *DCIns* pode ser visualizada na Listagem 4.3. A API Java JSR 353 foi utilizada com o objetivo de percorrer todo o documento JSON extraindo informações relevantes para o processo descrito nesta dissertação. Por meio de um *parser* é percorrido os documentos de entrada, gerando o arquivo de texto dos campos e valores (*DCIns*). Os eventos presentes na API Java JSR 353 são denominados *KEY_NAME* (relacionado aos campos) e *VALUE_STRING* (relacionado às instâncias).

```
1 public class CasamentoJSON{
2
3     public static void main(String[] args) throws FileNotFoundException,
      IOException {
4 ...
5 //-------------DCIns
```

[5] https://github.com/json-path/JsonPath

```
6         StringWriter stWriter = new StringWriter();
7         try (JsonWriter jsonWriter = Json.createWriter(stWriter)) {
8       jsonWriter.write(lendoArq);
9     }
10        String DocJson = stWriter.toString();
11      JsonParser parser = Json.createParser(new StringReader(DocJson));
12        while (parser.hasNext()) {
13            JsonParser.Event evento = parser.next();
14            if (evento == KEY_NAME){
15                Campo= parser.getString();
16            }
17            if (evento == JsonParser.Event.VALUE_STRING){
18                campo.add ("\nDoc: "+m);
19                campo.add("Campo: "+Campo);
20                campo.add("Valor: "+parser.getString());
21            }
22        }
23 ...
24    }
25 }
```

Listagem 4.3 – Utilização da API Java JSR para determinar o documento *DCIns*

Diversas classes são instanciadas na classe principal, a primeira delas diz respeito à análise dos radicais dos elementos ancestrais. Para ser aplicado, o algoritmo de Porter é utilizado, sendo que o mesmo está disponível para testes por meio da linguagem Java[6]. A classe *MatrizRadical* contém os métodos para realizar a extração dos radicais dos elementos ancestrais e descendentes, dependendo da etapa em que se encontra o processo. A Listagem 4.4 ilustra o preenchimento da matriz *Rad*, onde é verificado se os radicais são iguais ou não, atribuindo 1 (um) ou 0 (zero), respectivamente. Um arquivo com extensão *.csv* é gravado com as informações da matriz *Rad*.

```
1  public class MatrizRadical extends CasamentoJSON {
2      public void RadAncestral() throws IOException{
3          ...
4          double Rad[][]= new double[radical.size()][radical.size()];
5          for(int i=0; i<radical.size()-1; i++){
6              for(int j=i+1; j<radical.size(); j++){
7                  if(radical.get(i) == null ? radical.get(j) == null :
     radical.get(i).equals(radical.get(j))){
8                      Rad[i][j] = 1;
9                  }else
10                     {
11                         Rad[i][j] = 0;
12                     }
13             }
14         }
15 ...
16     }
17 }
```

Listagem 4.4 – Verificação dos radicais para preencher a matriz *Rad*

[6] https://tartarus.org/martin/PorterStemmer/java.txt

Assim como a classe anterior, a classe *MatrizJaro* possui métodos que realizam o cálculo da análise de caracteres tanto para os elementos ancestrais quanto para os descendentes. O uso da dependência *org.apache.commons.text.similarity* é essencial para calcular o valor da aplicação de *Jaro Winkler* sobre os elementos. Na Listagem 4.5 uma matriz denominada *Jaro* recebe o resultado do cálculo do método *distanceJaro* com os parâmetros das palavras em análise. É possível observar também que o método *replace* é utilizado para retirar os caracteres que podem influenciar no resultado.

```
1  public class MatrizJaro extends CasamentoJSON {
2      public void JaroAncestral() throws IOException{
3          ....
4          double Jaro[][]= new double[Palavras.length][Palavras.length];
5          for(int i=0; i<Palavras.length-1; i++){
6              for(int j=i+1; j<Palavras.length; j++){
7                  Palavras[i]=(Palavras[i].replace(" ",""));
8                  Palavras[j]=(Palavras[j].replace(" ",""));
9                  Palavras[i]=(Palavras[i].replace(";",""));
10                 Palavras[j]=(Palavras[j].replace(";",""));
11
12             Jaro[i][j]=MatrizJaro.distanceJaro(Palavras[i], Palavras[j]);
13
14             }
15         }
16 ...
17     }
18 }
```

Listagem 4.5 – Preenchimento da matriz *Jaro* com os valores do cálculo de *Jaro Winkler*

A última análise realizada nos elementos é o de conhecimento, esta pode ser observada na classe *MatrizWuPalmer*. Os elementos ancestrais e descendentes são analisados dependendo em que estágio se encontra o processo. Para realizar o cálculo foi utilizado o *Wordnet Similarity for Java*[7], considerando os resultados obtidos na medida de *Wu & Palmer*, trazendo valores no intervalo de 0 (zero) a 1 (um). A Listagem 4.6 ilustra o método que realiza o cálculo e retorna o valor da similaridade, sendo utilizado para preencher a matriz *WuP*.

```
1  public class MatrizWuPalmer extends CasamentoJSON {
2  ...
3      public static double compute(String palavra1, String palavra2) {
4          WS4JConfiguration.getInstance().setMFS(true);
5          double similaridade = new WuPalmer(db).calcRelatednessOfWords(
       palavra1, palavra2);
6          return similaridade;
7      }
8  ...
9  }
```

Listagem 4.6 – Método *compute* que calcula a medida de *Wu & Palmer* para os elementos em análise

[7] https://github.com/Sciss/ws4j

As três matrizes geradas nas classes descritas anteriormente (*Rad*, *Jaro* e *WuP*) são gravadas em arquivos com extensões *.csv* para serem utilizadas na próxima etapa. O cálculo de equivalência (presente na 2ª fase do processo) recebe como entrada as matrizes geradas anteriormente. A classe *CalculoEquiv* possui métodos que geram uma matriz de equivalência tanto para os elementos ancestrais quanto para os descendentes. Alguns trechos do cálculo de equivalência podem ser visualizados na Listagem 4.7.

```
public class CalculoEquiv extends CasamentoJSON {
...
    public void GerandoMEquAnc() throws IOException{
    ...
    double Resultado[][]= new double[n_linhas][n_linhas];
    //inicio do calculo de equivalencia
    for (i=0; i<n_linhas; i++)
    {
        for (j=i; j<n_linhas; j++)
        {
           // Se pelo menos um dos 3 for igual a 1 considera equivalente
            if ((jaro.Jaro[i][j] == 1.0) || (radical.Rad[i][j] == 1.0) || (
    wupalmer.WuP[i][j] == 1.0))
            {
            Resultado[i][j] = 1;}
            else{
                // Todos igual a 0 considera nao equivalente
                    if (jaro.Jaro[i][j] == 0 && radical.Rad[i][j] == 0 &&
    wupalmer.WuP[i][j] == 0)
                    {
                  Resultado[i][j] = 0;}
                    else{
                        cont=0;
                        if (jaro.Jaro[i][j] >0.0 && jaro.Jaro[i][j] < 1.0)
                            {
                                Aux = jaro.Jaro[i][j];
                                cont++;}
                        if (radical.Rad[i][j] >0.0 && radical.Rad[i][j] <
    1.0)
                                {
                                    Aux = radical.Rad[i][j];
                                    cont++;}
                        if (wupalmer.WuP[i][j] >0.0 && wupalmer.WuP[i][j] <
     1.0)
                                {
                                    Aux = wupalmer.WuP[i][j];
                                    cont++;}
                        if (cont==1)
                            {
                                //se valor maior que ponto de corte(0.6)
                                if (Aux > ponto_corte)
                                {
                                    Resultado[i][j]=1;}
                                else{
                                    Resultado[i][j]=0;}
                            }else{
                        if (cont>1){
//mais de um com valores entre 0 < x < 1
```

```
media_ponderada = (pesoA*radical.Rad[i][j] + pesoB*jaro.Jaro[i][j] + pesoC*
    wupalmer.WuP[i][j])/6;
//Se o resultado for maior que ponto de corte = 0.5
               if (media_ponderada > ponto_corte_AHP)
                         {
                             Resultado[i][j] = 1; }
                         else{ Resultado[i][j] = 0;}
...
    }
}
```

Listagem 4.7 – Trechos do cálculo de equivalência para as matrizes geradas em cada técnica aplicada

Como pode ser visto no Capítulo 3 em mais detalhes, a próxima atividade extrai os pares de elementos considerados equivalentes e não equivalentes. Para extrair as palavras foi necessário carregar a matriz resultante, assim como os elementos ancestrais ou descendentes. Um trecho da obtenção dos elementos dos documentos *DDesC* e *DDesN* pode ser observado na Listagem 4.8. As palavras advindas dos elementos descendentes (documento *DDes*) são carregados em um vetor denominado *Plv*. O método *replace* é utilizado para retirar espaços em branco. Uma verificação é realizada para encaixar as palavras testadas com os valores das matrizes resultantes do cálculo de equivalência. Tanto os elementos descendentes quanto os ancestrais são extraídos das matrizes de equivalência da mesma forma.

```
...
for (k=0; k<Plv.length; k++){
    for (l=0;l<Plv.length;l++){
        Plv[k]=(Plv[k].replace(" ",""));
        Plv[l]=(Plv[l].replace(" ",""));
        if ((l>k) && (k != l)){
            if (matD.MEquDesc[k][l] == 1.0){
                palavras_DDesC.add("\n"+(Plv[k]+" - "+Plv[l]));
            }
            if (matD.MEquDesc[k][l] == 0.0){
                    palavras_DDesN.add((Plv[k]+"; "+Plv[l]+";"));
            }
...
```

Listagem 4.8 – Extração dos pares de elementos considerados equivalentes e não equivalentes

Outra classe implementada diz respeito ao uso da medida de similaridade de *Jaccard*. Esta utilizada para medir a sobreposição de dados presente nas instâncias dos elementos considerados não equivalentes (*DDesN*) na 2ª fase. Para realizar este cálculo foi necessário extrair as instâncias dos campos do documento *DDesN* e logo após aplicar a medida de *Jaccard*.

A dependência *org.apache.commons.text.similarity* também foi utilizada no método *AplicaJaccard*, descrita resumidamente na Listagem 4.9. As *ArrayList* denominadas *valorA* e *valorB* se referem ao par de instâncias extraídas. Todos os pares são percorridos e atribuídos os

valores da aplicação da medida de *Jaccard*. O ponto de corte foi testado para determinar os pares de campos que possuem instâncias com sobreposições de dados.

```
public void AplicaJaccard(){
...
JaccardSimilarity obj = new JaccardSimilarity();
for (int g=0; g<valorB.size();g++){
    for (int gd=0; gd<valorA.size();gd++){
    Double res = obj.apply((String)valorA.get(gd), (String)valorB.get(g));
//houve sobreposicao se maior ou igual a 0.55
        if (res >= 0.55){
            sobrepo.add(Plv[kk-1]+" - "+  Plv[kk]);
        }
...
}
```

Listagem 4.9 – Aplicando a medida de *Jaccard* nas instâncias dos elementos descendentes não equivalentes

As Listagens descritas nesta seção são resumidamente apresentadas por questões de espaço. Todas as classes são instanciadas na classe principal, e os métodos utilizados. Um maior detalhamento de todo o passo a passo do processo para casamento de esquemas pode ser visto no Capítulo 3 e no estudo de caso (Seção 4.2), tendo em vista que esta seção apresenta apenas as atividades que foram implementadas, não considerando as atividades realizadas manualmente.

4.2 ESTUDO DE CASO

A fim de melhorar o entendimento sobre como todo o processo para casamento de esquemas funciona, esta seção tem o objetivo de demonstrar um estudo de caso. O domínio dos documentos JSON utilizado foi de publicações científicas, onde os arquivos de entrada são referências exportadas de artigos e demais trabalhos científicos de bibliotecas digitais *PubMed*[8] e *Bibsonomy*[9]. Por questões de organização do texto dois documentos foram utilizados para demonstrar como o processo para casamento de esquemas funciona, mas se uma coleção possui mais documentos o modo de realizar é o mesmo.

Originalmente os documentos apresentados estavam no formato *XML*, mas o processo explora apenas o formato *JSON*, por este ser mais recente e mais explorado em banco de dados como MongoDB e CouchDB. Para realizar a conversão dos documentos foi utilizada a ferramenta *Altova XMLSpy*. As Listagens 4.10 e 4.11 descrevem trechos dos documentos utilizados neste estudo de caso.

```
{ "Sources": {
```

[8] https://www.ncbi.nlm.nih.gov/pubmed
[9] https://www.bibsonomy.org/

```
    "@SelectedStyle": "",
    "Source": [{
        "SourceType": ["JournalArticle2"],
        "Tag": ["hu2019synoptic"],
        "Title": ["A synoptic assessment of the summer extreme rainfall
    over the middle reaches of Yangtze River in CMIP5 models"],
        "Year": ["2019"],
        "Author": [
          { "Author": [ {"NameList": [{
                    "Person": [{
                        "Last": ["Hu"],
                        "First": ["Yang"]},
                      {"Last": ["Deng"],
                        "First": ["Yi"]},
                      {"Last": ["Zhou"],
                        "First": ["Zhimin"]}
                ... ] } ] } ] } ],
        "StandardNumber1": ["ISSN: 1432-0894 DOI: 10.1007/s00382
    -019-04803-3"],
        "JournalName": ["Climate Dynamics"],
        "Month": ["May"]
        ...}, {
        "SourceType": ["JournalArticle"],
        "Tag": ["gonzalez2019contribution"],
        "Title": ["The contribution of North Atlantic atmospheric
    circulation shifts to future wind speed projections for wind power over
    Europe"],
        "Year": ["2019"],
        ...
        "JournalName": ["Climate Dynamics"],
          ... } ] } }
```


Listagem 4.10 – Documento 0 da biblioteca digital *Bibsonomy*


```
{"PubmedArticleSet": {
    "PubmedArticle": [ {
        "MedlineCitation": [ {
            "@Status": "PubMed-not-MEDLINE",
            "@Owner": "NLM",
            "PMID": [{
                "@Version": 1,
                "$a": "28578772" } ],
            "DateCompleted": [ {
                "Year": ["2017"],
                "Month": ["11"],
                "Day": ["01"] } ],
            "DateRevised": [ {
                "Year": ["2017"],
                "Month": ["11"],
                "Day": ["01"] } ],
            "Article": [ {
                "@PubModel": "Print",
                "Journal": [ {
                    "ISSN": [ {
                        "@IssnType": "Electronic",
                        "$a": "1531-6564" } ],
                    "JournalIssue": [ {
                        "@CitedMedium": "Internet",
                        "Volume": ["42"],
```

```
                        "Issue": ["6"],
                        "PubDate": [ {
                            "Year": ["2017"],
                            "Month": ["06"] } ] } ],
                    "Title": ["The Journal of hand surgery"],
                    "ISOAbbreviation": ["J Hand Surg Am"]
                  } ],
                "ArticleTitle": [ { } ],
                "Pagination": [ {
                    "MedlinePgn": ["e219-e220"] } ],
                "ELocationID": [ {
                    "@EIdType": "pii",
                    "@ValidYN": "Y",
                    "$a": "S0363-5023(17)30364-7" },
                  { "@EIdType": "doi",
                    "@ValidYN": "Y",
                    "$a": "10.1016/j.jhsa.2017.03.038" } ],
                "AuthorList": [ {
                    "@CompleteYN": "Y",
                    "Author": [ {
                        "@ValidYN": "Y",
                        "LastName": ["Luther"],
                        "ForeName": ["Gaurav Aman"],
                        ... }, {
                        "@ValidYN": "Y",
                        "LastName": ["Murthy"],
                        "ForeName": ["Praveen"],
                        "Initials": ["P"],
                        .... }]}],
              "Language": ["eng"]
```


Listagem 4.11 – Documento 1 da biblioteca digital *PubMed*

Para facilitar o entendimento, as subseções seguintes se referem às quatro fases presentes no processo. Essa distribuição tem o objetivo de demonstrar, de maneira mais claro, como o estudo de caso se comporta em cada fase do processo para casamento de esquemas.

4.2.1 Pré-casamento dos esquemas - 1ª Fase

Inicialmente, na 1ª fase, os documentos passam por um pré-processamento. A atividade *Aplicar o algoritmo Diff* processa os documentos e verifica se os mesmos são idênticos ou não. Isto é importante, pois se os mesmos forem idênticos não há necessidade de realizar todo o processamento da 2ª e 3ª fase. Por meio da aplicação do algoritmo na ferramenta *online* disponível em JSONDiff[10] foi possível observar que os documentos não são idênticos.

A atividade seguinte (*Extrair as informações*) é executada gerando os documentos *DCIns* e *DEst*. A mesma foi implementada, como pode ser visto na Seção 4.1. A Listagem 4.12 ilustra os campos e valores (instâncias) extraídas dos documentos 0 e 1. Já a Listagem 4.13 descreve a

[10] http://www.jsondiff.com/

estrutura dos documentos 0 e 1, ou seja, os caminhos que os campos pertencem.

```
[
Doc: 0, Campo: @SelectedStyle, Valor: ,
Doc: 0, Campo: SourceType, Valor: JournalArticle2,
Doc: 0, Campo: Tag, Valor: hu2019synoptic,
Doc: 0, Campo: Title, Valor: A synoptic assessment of the summer extreme
    rainfall over the middle reaches of Yangtze River in CMIP5 models,
Doc: 0, Campo: Year, Valor: 2019,
Doc: 0, Campo: Last, Valor: Hu,
Doc: 0, Campo: First, Valor: Yang,
Doc: 0, Campo: Last, Valor: Deng,
Doc: 0, Campo: First, Valor: Yi,
Doc: 0, Campo: Last, Valor: Zhou,
Doc: 0, Campo: First, Valor: Zhimin,
Doc: 0, Campo: StandardNumber1, Valor:  ISSN: 1432-0894 DOI: 10.1007/s00382
    -019-04803-3,
Doc: 0, Campo: JournalName, Valor: Climate Dynamics,
...
Doc: 1, Campo: @Status, Valor: PubMed-not-MEDLINE,
Doc: 1, Campo: @Owner, Valor: NLM,
Doc: 1, Campo: $a, Valor: 28578772,
Doc: 1, Campo: Year, Valor: 2017,
Doc: 1, Campo: Month, Valor: 11,
Doc: 1, Campo: Day, Valor: 01,
Doc: 1, Campo: Year, Valor: 2017,
Doc: 1, Campo: Month, Valor: 11,
Doc: 1, Campo: Day, Valor: 01,
Doc: 1, Campo: @PubModel, Valor: Print,
Doc: 1, Campo: @IssnType, Valor: Electronic,
Doc: 1, Campo: $a, Valor: 1531-6564,
Doc: 1, Campo: @CitedMedium, Valor: Internet,
Doc: 1, Campo: Volume, Valor: 42,
Doc: 1, Campo: Issue, Valor: 6,
Doc: 1, Campo: Year, Valor: 2017,
Doc: 1, Campo: Month, Valor: 06,
Doc: 1, Campo: Title, Valor: The Journal of hand surgery,
Doc: 1, Campo: ISOAbbreviation, Valor: J Hand Surg Am,
Doc: 1, Campo: MedlinePgn, Valor: e219-e220,
Doc: 1, Campo: @EIdType, Valor: pii,
Doc: 1, Campo: @ValidYN, Valor: Y,
Doc: 1, Campo: $a, Valor: S0363-5023(17)30364-7,
Doc: 1, Campo: @EIdType, Valor: doi,
Doc: 1, Campo: @ValidYN, Valor: Y,
Doc: 1, Campo: $, Valor: 10.1016/j.jhsa.2017.03.038,
Doc: 1, Campo: @CompleteYN, Valor: Y,
Doc: 1, Campo: @ValidYN, Valor: Y,
Doc: 1, Campo: LastName, Valor: Luther,
Doc: 1, Campo: ForeName, Valor: Gaurav Aman,
Doc: 1, Campo: @ValidYN, Valor: Y,
Doc: 1, Campo: LastName, Valor: Murthy,
Doc: 1, Campo: ForeName, Valor: Praveen,
Doc: 1, Campo: Initials, Valor: P
]
```

Listagem 4.12 – Documento *DCIns*

```
[
Doc: 0, $['Sources'],
```

```
Doc: 0, $['Sources']['@SelectedStyle'],
Doc: 0, $['Sources']['Source'],
Doc: 0, $['Sources']['Source'][0],
Doc: 0, $['Sources']['Source'][0]['SourceType'],
Doc: 0, $['Sources']['Source'][0]['Tag'],
Doc: 0, $['Sources']['Source'][0]['Title'],
Doc: 0, $['Sources']['Source'][0]['Year'],
Doc: 0, $['Sources']['Source'][0]['Author'],
Doc: 0, $['Sources']['Source'][0]['StandardNumber1'],
Doc: 0, $['Sources']['Source'][0]['JournalName'],
Doc: 0, $['Sources']['Source'][0]['Month'],
Doc: 0, $['Sources']['Source'][0]['SourceType'][0],
Doc: 0, $['Sources']['Source'][0]['Tag'][0],
Doc: 0, $['Sources']['Source'][0]['Title'][0],
Doc: 0, $['Sources']['Source'][0]['Year'][0],
Doc: 0, $['Sources']['Source'][0]['Author'][0],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]['NameList'],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
     ['NameList'][0],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
     ['NameList'][0]['Person'],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
     ['NameList'][0]['Person'][0],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
     ['NameList'][0]['Person'][0]['Last'],
Doc: 0, $['Sources']['Source'][0]['Author'][0]['Author'][0]
     ['NameList'][0]['Person'][0]['First'],
...
Doc: 1, $['PubmedArticleSet'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['@Status'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['@Owner'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['PMID'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['DateCompleted'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['DateRevised'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'],
...
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['@PubModel'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['Journal'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['ArticleTitle'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['Pagination'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
```

```
        ['Article'][0]['ELocationID'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'],
...
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['@CompleteYN'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['Author'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['Author'][0],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['Author'][0]['@ValidYN'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['Author'][0]['LastName'],
Doc: 1, $['PubmedArticleSet']['PubmedArticle'][0]['MedlineCitation'][0]
        ['Article'][0]['AuthorList'][0]['Author'][0]['ForeName'],
...]
```

Listagem 4.13 – Documento *DEst*

4.2.2 Análise do esquema - 2ª Fase

A próxima fase a ser realizada é a análise do esquema (2ª fase), para isto é necessário o documento *DEst* para extrair os elementos ancestrais. Como mencionado anteriormente, esta atividade (*Extrair os elementos ancestrais*) foi realizada manualmente, gerando o documento denominado *DAnc* que pode ser visualizado na Listagem 4.14.

```
Sources; Source; Author; Author; NameList; Person; PubmedArticleSet;
PubmedArticle; MedlineCitation; PMID; DateCompleted; DateRevised; Article;
Journal; ISSN; JournalIssue; PubDate; Pagination; ELocationID; AuthorList;
Author; AffiliationInfo; PublicationTypeList; PublicationType;
MedlineJournalInfo; CommentsCorrectionsList; CommentsCorrections;
PubmedData; History; PubMedPubDate; ArticleIdList; ArticleId;
```

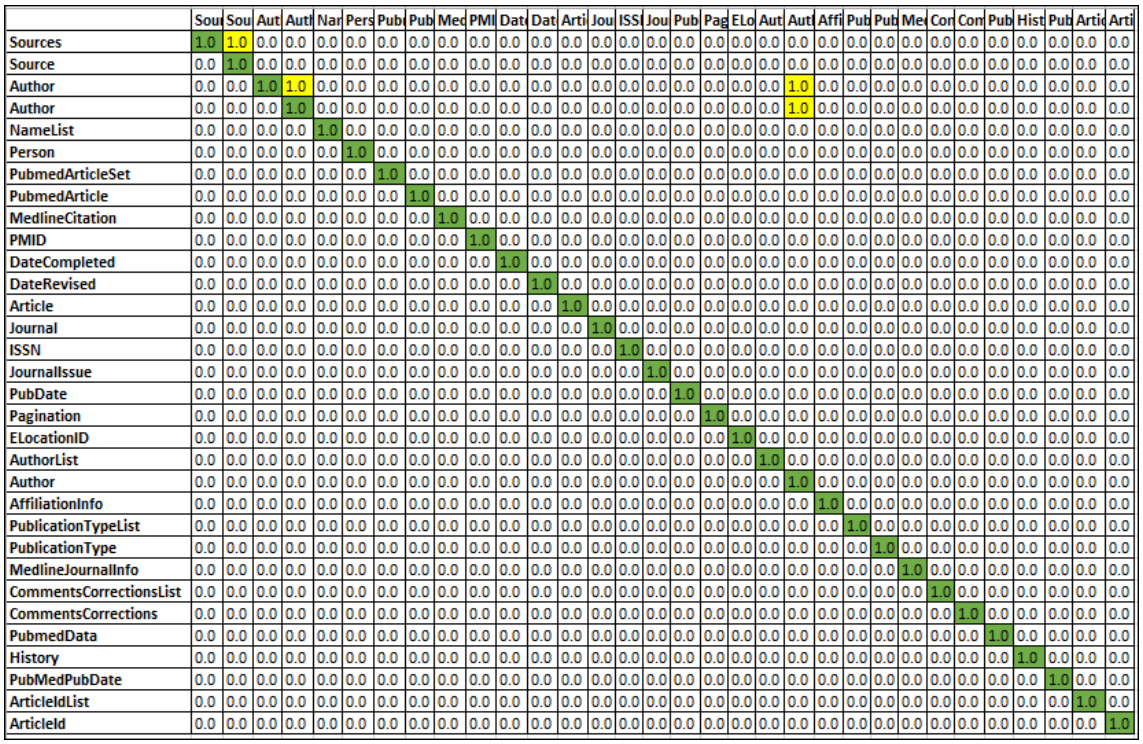
Listagem 4.14 – Documento *DAnc*

Para testar o quanto os elementos ancestrais são equivalentes, a próxima atividade *Avaliar a similaridade* possui três subatividades correspondentes às diferentes técnicas aplicadas. A primeira delas é a de *Analisar os radicais*, onde os radicais foram extraídos e analisados, gerando uma matriz de similaridade *MSim* que pode ser visualizada na Figura 20.

É importante notar que por ser uma matriz simétrica, os elementos abaixo da diagonal principal não são considerados, assim como a diagonal principal, pois é a similaridade do elemento com ele mesmo. Os elementos em destaque são os que possuem equivalências, ou seja, adquiriram valor 1 (um).

As Figuras 21 e 22 ilustram as outras técnicas, respectivamente análise de caracteres e

Figura 20 – Matriz de similaridade *MSim* da análise dos radicais nos elementos ancestrais



	Sou	Sou	Aut	Aut	Nar	Pers	Pub	Pub	Med	PMI	Dat	Dat	Arti	Jou	ISS	Jou	Pub	Pag	ELo	Aut	Aut	Affi	Pub	Pub	Me	Cor	Cor	Pub	Hist	Pub	Artic	Arti
Sources	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Source	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NameList	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Person	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubmedArticleSet	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubmedArticle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MedlineCitation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PMID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateCompleted	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateRevised	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Article	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISSN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalIssue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AffiliationInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PublicationType	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MedlineJournalInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrectionsList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrections	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
PubmedData	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
History	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
PubMedPubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
ArticleIdList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
ArticleId	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

de conhecimento, aplicadas nos elementos ancestrais. Em todas as técnicas é possível observar que os elementos são testados todos com todos, atribuindo valores aos pares em análise.

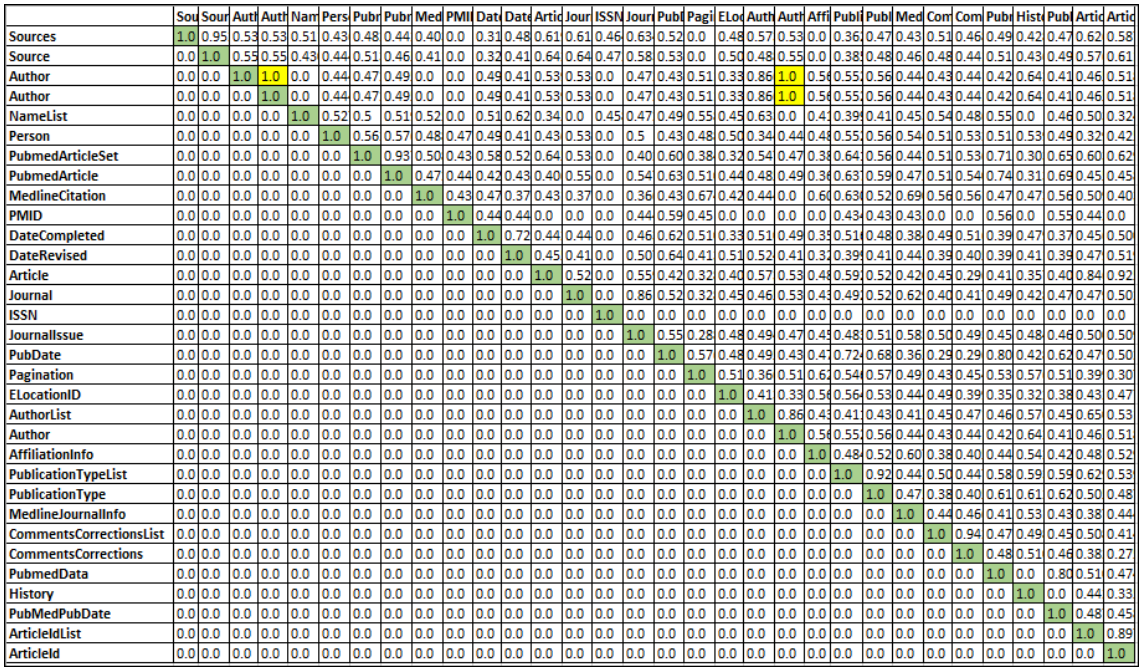
As matrizes apresentadas são levadas em consideração para a próxima atividade que realiza o cálculo de equivalência. Uma única matriz de equivalência *MEqu* é gerada a partir dos cálculos aplicados (descritas na Seção 3.4). A Figura 23 ilustra a matriz *MEqu* gerada para os elementos ancestrais.

Os elementos considerados equivalentes estão em destaque (valor 1.0). Para demonstrar como a atividade *Aplicar o cálculo de equivalência* funciona, os casos previstos na Tabela 4 (Seção 3.4) são explicados em detalhes.

O primeiro caso gera como resultado o valor 1 se qualquer uma das três técnicas obtiver valor 1. O par de elementos ancestrais que pode ser observado é *Author* e *Author* (pertencentes ao Documento 0 e 1, respectivamente) que obteve total equivalência na análise dos radicais, sendo assim, na matriz de equivalência, o valor gerado foi 1 para este par de elementos.

No segundo caso se todas as técnicas obtiver valor 0 (zero) o resultado gerado na matriz de equivalência é 0 (zero), ou seja, não equivalente. Observando o par de elementos *NameList* e *Journal* (pertencentes ao Documento 0 e 1, respectivamente) é possível visualizar que nas três

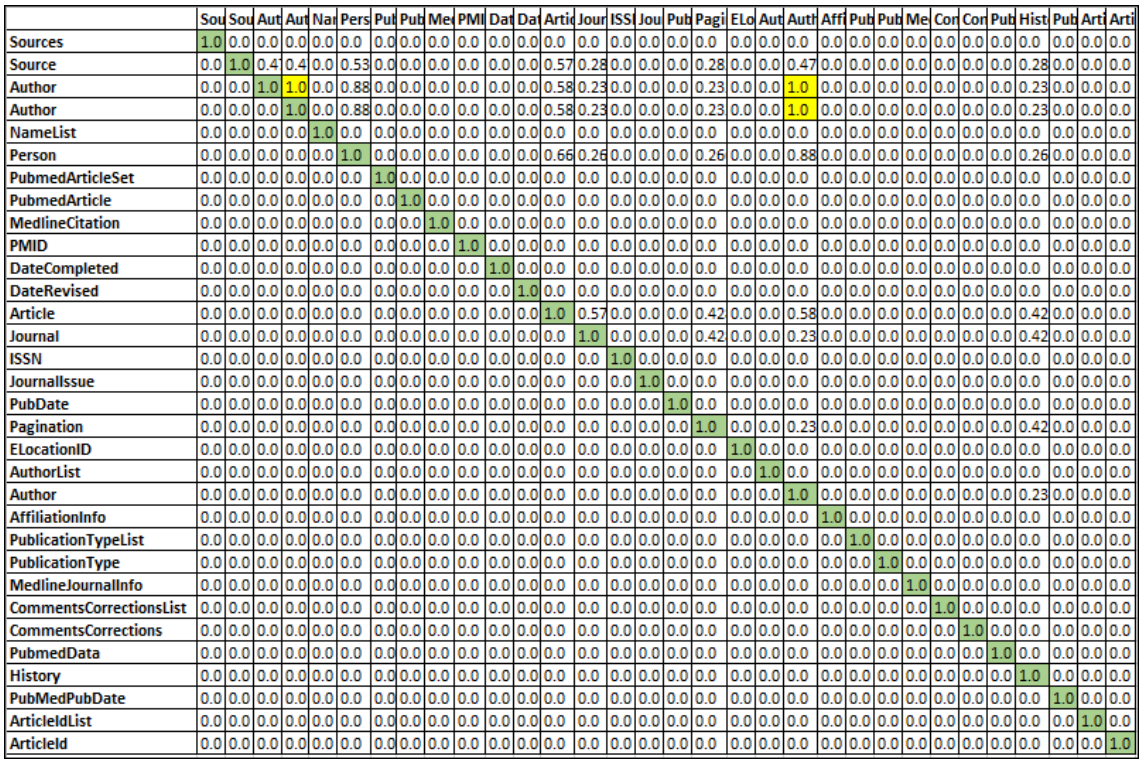
Figura 21 – Matriz de similaridade *MSim* da análise dos caracteres nos elementos ancestrais



	Sou	Sour	Auth	Auth	Nam	Pers	Pubr	Pubr	Med	PMI	Dat	Date	Artic	Jour	ISSN	Jour	Publ	Pagi	ELo	Auth	Auth	Affi	Publ	Publ	Med	Com	Com	Pubr	Hist	Pub	Artic	Artic
Sources	1.0	0.95	0.53	0.53	0.51	0.43	0.48	0.44	0.40	0.0	0.31	0.48	0.61	0.61	0.46	0.63	0.52	0.0	0.48	0.57	0.53	0.0	0.36	0.47	0.43	0.51	0.46	0.49	0.42	0.47	0.62	0.58
Source	0.0	1.0	0.55	0.55	0.43	0.44	0.51	0.46	0.41	0.0	0.32	0.41	0.64	0.64	0.47	0.58	0.53	0.0	0.50	0.48	0.55	0.0	0.38	0.48	0.46	0.48	0.44	0.51	0.43	0.49	0.57	0.61
Author	0.0	0.0	1.0	1.0	0.0	0.44	0.47	0.49	0.0	0.0	0.49	0.41	0.53	0.53	0.0	0.47	0.43	0.51	0.33	0.86	1.0	0.56	0.55	0.56	0.44	0.43	0.44	0.42	0.64	0.41	0.46	0.51
Author	0.0	0.0	0.0	1.0	0.0	0.44	0.47	0.49	0.0	0.0	0.49	0.41	0.53	0.53	0.0	0.47	0.43	0.51	0.33	0.86	1.0	0.56	0.55	0.56	0.44	0.43	0.44	0.42	0.64	0.41	0.46	0.51
NameList	0.0	0.0	0.0	0.0	1.0	0.52	0.5	0.51	0.52	0.0	0.51	0.62	0.34	0.0	0.45	0.47	0.49	0.55	0.45	0.63	0.0	0.41	0.39	0.41	0.45	0.54	0.48	0.55	0.0	0.46	0.50	0.32
Person	0.0	0.0	0.0	0.0	0.0	1.0	0.56	0.57	0.48	0.47	0.49	0.41	0.43	0.53	0.0	0.5	0.43	0.48	0.50	0.34	0.44	0.48	0.55	0.56	0.54	0.51	0.53	0.51	0.53	0.49	0.32	0.42
PubmedArticleSet	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.93	0.50	0.43	0.58	0.52	0.64	0.53	0.0	0.40	0.60	0.38	0.32	0.54	0.47	0.38	0.64	0.56	0.44	0.51	0.53	0.71	0.30	0.65	0.60	0.62
PubmedArticle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.47	0.44	0.42	0.43	0.40	0.55	0.0	0.54	0.63	0.51	0.44	0.48	0.49	0.36	0.63	0.59	0.47	0.51	0.54	0.74	0.31	0.69	0.45	0.45
MedlineCitation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.43	0.47	0.37	0.43	0.37	0.0	0.36	0.43	0.67	0.42	0.44	0.0	0.60	0.63	0.52	0.69	0.56	0.56	0.47	0.47	0.56	0.50	0.40
PMID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.44	0.44	0.0	0.0	0.0	0.44	0.59	0.45	0.0	0.0	0.0	0.0	0.43	0.43	0.43	0.0	0.0	0.56	0.0	0.55	0.44	0.0
DateCompleted	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.72	0.44	0.44	0.0	0.46	0.62	0.51	0.33	0.51	0.49	0.35	0.51	0.48	0.38	0.49	0.51	0.39	0.47	0.37	0.45	0.50
DateRevised	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.45	0.41	0.0	0.50	0.64	0.41	0.51	0.52	0.41	0.32	0.39	0.41	0.44	0.39	0.40	0.39	0.41	0.39	0.47	0.51
Article	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.52	0.0	0.55	0.42	0.32	0.40	0.57	0.53	0.48	0.59	0.52	0.42	0.45	0.29	0.41	0.35	0.40	0.84	0.92
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.86	0.52	0.32	0.45	0.46	0.53	0.43	0.49	0.52	0.62	0.40	0.41	0.49	0.42	0.47	0.47	0.50
ISSN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalIssue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.55	0.28	0.48	0.49	0.47	0.45	0.48	0.51	0.58	0.50	0.49	0.45	0.48	0.46	0.50	0.50
PubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.57	0.48	0.49	0.43	0.47	0.72	0.68	0.36	0.29	0.29	0.80	0.42	0.62	0.47	0.50
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.51	0.36	0.51	0.62	0.54	0.57	0.49	0.43	0.45	0.53	0.57	0.51	0.39	0.30
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.41	0.33	0.56	0.56	0.53	0.44	0.49	0.39	0.35	0.32	0.38	0.43	0.47
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.86	0.43	0.41	0.43	0.41	0.45	0.47	0.46	0.57	0.45	0.65	0.53
Author	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.56	0.55	0.56	0.44	0.43	0.44	0.42	0.64	0.41	0.46	0.51
AffiliationInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.48	0.52	0.60	0.38	0.40	0.44	0.54	0.42	0.48	0.52
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.92	0.44	0.50	0.44	0.58	0.59	0.59	0.62	0.53
PublicationType	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.47	0.38	0.40	0.61	0.61	0.62	0.50	0.48
MedlineJournalInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.44	0.46	0.41	0.53	0.43	0.38	0.44
CommentsCorrectionsList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.94	0.47	0.49	0.45	0.50	0.41
CommentsCorrections	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.48	0.51	0.46	0.38	0.27
PubmedData	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.80	0.51	0.47
History	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.44	0.33
PubMedPubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.48	0.45
ArticleIdList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.89
ArticleId	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

Figura 22 – Matriz de similaridade *MSim* da análise do conhecimento nos elementos ancestrais



	Sou	Sou	Aut	Aut	Nar	Pers	Pul	Pub	Me	PMI	Dat	Da	Artic	Jour	ISS	Jou	Pub	Pagi	ELo	Aut	Auth	Affi	Pub	Pub	Me	Cor	Cor	Pub	Hist	Pub	Arti	Arti
Sources	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Source	0.0	1.0	0.4	0.4	0.0	0.53	0.0	0.0	0.0	0.0	0.0	0.0	0.57	0.28	0.0	0.0	0.0	0.28	0.0	0.0	0.47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.28	0.0	0.0	0.0
Author	0.0	0.0	1.0	1.0	0.0	0.88	0.0	0.0	0.0	0.0	0.0	0.0	0.58	0.23	0.0	0.0	0.0	0.23	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.0	0.0	0.0
Author	0.0	0.0	0.0	1.0	0.0	0.88	0.0	0.0	0.0	0.0	0.0	0.0	0.58	0.23	0.0	0.0	0.0	0.23	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.0	0.0	0.0
NameList	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Person	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.66	0.26	0.0	0.0	0.0	0.26	0.0	0.0	0.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.0	0.0	0.0
PubmedArticleSet	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubmedArticle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MedlineCitation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PMID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateCompleted	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateRevised	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Article	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.57	0.0	0.0	0.0	0.42	0.0	0.0	0.58	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.0	0.0	0.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.42	0.0	0.0	0.23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.0	0.0	0.0
ISSN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalIssue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.0	0.0	0.0
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.0	0.0	0.0
AffiliationInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PublicationType	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MedlineJournalInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrectionsList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrections	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
PubmedData	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
History	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
PubMedPubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
ArticleIdList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
ArticleId	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

Figura 23 – Matriz de equivalência *MEqu* das análises dos elementos ancestrais

	Sou	Sou	Aut	Aut	Nar	Per	Pub	Pub	Me	PM	Dat	Dat	Arti	Jou	ISS	Jou	Pub	Pag	ELo	Aut	Aut	Affi	Pub	Pub	Me	Cor	Cor	Pub	His	Pub	Arti	Arti
Sources	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Source	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Author	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Author	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NameList	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Person	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubmedArticleSet	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0
PubmedArticle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
MedlineCitation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PMID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateCompleted	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DateRevised	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Article	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
Journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ISSN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JournalIssue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
Pagination	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ELocationID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AuthorList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Author	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AffiliationInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PublicationTypeList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
PublicationType	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
MedlineJournalInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrectionsList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
CommentsCorrections	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
PubmedData	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
History	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
PubMedPubDate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
ArticleIdList	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
ArticleId	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

técnicas aplicadas, todos obtiveram valor 0 (zero).

O próximo caso previsto diz respeito aos valores gerados no intervalo de 0 a 1. Se uma (apenas uma) das técnicas obtiver valor no intervalo de 0 a 1, é testado se o mesmo é maior ou menor igual a 0,6. Dependendo do resultado, a matriz de equivalência considera equivalente ou não equivalente. Os pares de elementos que se pode observar é *Person* e *ArticleId*, onde a única técnica que obteve valor no intervalo (0,42) foi a análise de caracteres. O valor gerado foi 0 (zero), pois 0,42 é menor que 0,6. Já o par *Source* e *ArticleId* obteve 0,61 em apenas uma técnica, gerando equivalência na matriz *MEqu*, ou seja, valor 1 (um).

O último caso previsto não se encaixa em nenhuma das condições descritas anteriormente, ou seja, quando mais de uma técnica obtiver valores no intervalo de 0 a 1 é aplicado a média ponderada com os pesos de cada técnica. Dependendo do valor gerado, é verificado se maior ou menor igual a 0,5. Um exemplo é o par de elementos *Person* e *Author* (pertencente ao Documento 0 e 1, respectivamente) que obtiveram os valores: *Radical*= 0; *Jaro*= 0,44; e *WuP*= 0,88. Realizando a média ponderada o valor resultante é 0,51, gerando o valor 1 (equivalente) na matriz *MEqu*, pois 0,51 é maior que o ponto de corte de 0,5.

Depois de gerado a matriz *MEqu*, a extração dos nomes dos elementos considerados equivalentes é realizada. Isto é necessário tendo em vista que a matrizes carregam apenas os valores e não os nomes. Nas figuras anteriores os nomes dos elementos foram colocados para melhorar a visualização dos pares de ancestrais em análise.

A atividade *Extrair os elementos* é realizada por meio de implementação, e uma verificação se são elementos ancestrais ou não é realizada manualmente. Assim, neste estágio do processo é gerado o documento *DCan* com os elementos ancestrais considerados equivalentes, ou seja, os pares que obtiveram valor 1 na matriz de equivalência *MEqu* da Figura 23. A Listagem 4.15 ilustra o documento *DCan*.

```
[
Sources - Source, Sources - Article, Sources - Journal, Sources -
    JournalIssue, Sources - ArticleIdList,
Source - Article, Source - ArticleId,
Author - Author, Author - Person, Author - AuthorList, Author - Author,
     Author - Person, Author - AuthorList, Author - Author,
NameList - DateRevised, NameList - AuthorList,
Person - Author,
PubmedArticleSet - PubmedArticle, PubmedArticleSet - Article,
    PubmedArticleSet - PubDate, PubmedArticleSet - PublicationTypeList,
    PubmedArticleSet - PubmedData, PubmedArticleSet - PubMedPubDate,
    PubmedArticleSet - ArticleIdList, PubmedArticleSet - ArticleId,
PubmedArticle - PubDate, PubmedArticle - PublicationTypeList,
    PubmedArticle - PubmedData, PubmedArticle - PubMedPubDate,
MedlineCitation - Pagination, MedlineCitation - AffiliationInfo,
    MedlineCitation - PublicationTypeList, MedlineCitation -
    MedlineJournalInfo,
DateCompleted - DateRevised, DateCompleted - PubDate,
DateRevised - PubDate,
Article - ArticleIdList, Article - ArticleId,
Journal - JournalIssue, Journal - MedlineJournalInfo,
PubDate - PublicationTypeList, PubDate - PublicationType, PubDate -
    PubmedData, PubDate - PubMedPubDate,
Pagination - AffiliationInfo,
AuthorList - Author, AuthorList - ArticleIdList,
AffiliationInfo - MedlineJournalInfo,
PublicationTypeList - PublicationType, PublicationTypeList -
    ArticleIdList,
PublicationType - PubmedData, PublicationType - History, PublicationType
    - PubMedPubDate,
CommentsCorrectionsList - CommentsCorrections,
PubmedData - PubMedPubDate,
ArticleIdList - ArticleId]
```

Listagem 4.15 – Documento *DCan*

A partir destes elementos ancestrais, a atividade *Extrair os elementos descendentes* é processada para cada par de ancestrais. Para isto o documento com a estrutura (*DEst*) é utilizado manualmente. A Listagem 4.16 ilustra um dos documentos *DDes* com os elementos descendentes do par de elementos ancestrais equivalentes *Person - Author*.

```
Last; First; Middle; @ValidYN; LastName; ForeName; Initials;
   AffiliationInfo;
```

Listagem 4.16 – Documento *DDes* contendo os elementos descendentes do par de ancestrais *Person - Author*

Conforme o andamento do processo, a próxima atividade a ser realizada é *Avaliar a similaridade*. Assim como foi realizado para os elementos ancestrais, esta atividade se aplica, neste estágio, aos elementos descendentes. As três técnicas são aplicadas a todos os descendentes e geradas as matrizes de equivalência. Como já foi exemplificado como as técnicas são geradas, assim como a explicação da atividade *Aplicar o cálculo de equivalência*, apenas a matriz de equivalência é ilustrada. Para os elementos da Listagem 4.16 é ilustrado a matriz *MEqu* na Figura 24.

Figura 24 – Matriz de equivalência *MEqu* das análises dos elementos descendentes

	Last	First	Middle	@ValidYN	LastName	ForeName	Initials	AffiliationInfo
Last	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
First	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Middle	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
@ValidYN	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
LastName	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
ForeName	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Initials	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
AffiliationInfo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fonte: Autor

Diferente do que foi realizado para os elementos ancestrais, a atividade *Extrair os elementos* obtém não apenas os pares de elementos equivalentes, mas também os não equivalentes. É importante notar que os documentos *DDes* (com os elementos descendentes de cada par de ancestrais) são examinados e unificados em um único documento. Os que são equivalentes ficam em (*DDesC*) e os não equivalentes em (*DDesN*). Para exemplificar, as Listagens 4.17 e 4.18 ilustram trechos dos documentos com os elementos candidatos ao casamento de esquemas (*DDesC*) e os não candidatos (*DDesN*), sendo direcionados aos exemplos dos pares obtidos na Figura 24.

```
[
...
Last - LastName,
LastName - ForeName,
...]
```

Listagem 4.17 – Documento *DDesC* dos elementos contidos na matriz *MEqu* da Figura 24

```
[...
Last - First,
Last - Middle,
Last - @ValidYN,
Last - ForeName,
Last - Initials,
Last - AffiliationInfo,
First - Middle,
First - @ValidYN,
...]
```

Listagem 4.18 – Documento *DDesN* dos elementos contidos na matriz *MEqu* da Figura 24

4.2.3 Análise das instâncias - 3ª Fase

Os dois documentos (*DDesC* e *DDesN*) são utilizados nas fases seguintes. Quanto aos não candidatos ao casamento dos esquemas (*DDesN*), há uma última verificação se os mesmos podem ser equivalentes quanto às suas instâncias. A atividade *Extrair valores dos elementos não-candidatos* é realizada por meio de implementação e um trecho pode ser visualizado no documento *DVal* na Listagem 4.19.

```
[...
Campo: First
Campo: @ValidYN

valor B[0] = Y;
valor A[0] = Yang;
valor B[0] = Y;
valor A[1] = Yi;
valor B[0] = Y;
valor A[2] = Zhimin;
...]
```

Listagem 4.19 – Documento *DVal*

A atividade *Analisar a sobreposição de dados* aplica a medida de *Jaccard* sobre as instâncias dos elementos considerados não equivalentes para verificar se existem sobreposições de dados. A Listagem 4.20 ilustra o valor obtido na medida de *Jaccard* quanto às instâncias em análise do par de elementos *First - @ValidYN*. Os pares de campos que obtiveram valores de sobreposição de dados nas suas instâncias (maiores que o ponto de corte de 0,55) são descritos no documento *DSob* (Listagem 4.21).

```
[...
Par dos campos----> First - @ValidYN
Par dos valores----> Yi; - Y;
JaccardSimilarity 0.66
...]
```

Listagem 4.20 – Valores obtidos para as instâncias

```
[Year - Day,
$a - Year,
ISOAbbreviation - MedlineTA,
$a - @PubStatus,
Day - Year,
SourceType - Tag,
Year - $a,
First - @ValidYN,
Month - Hour]
```

Listagem 4.21 – Documento *DSob*

Tanto a Listagem 4.20 quanto a 4.21 foram implementadas, como pode ser vista na Seção 4.1. A obtenção dos elementos que não obtiveram sobreposição de dados *DSobN* é realizado manualmente. Como a coleção de documentos JSON utilizada nesse estudo de caso não possui um dicionário de dados, a atividade *Utilizar dicionário de dados* não é executada.

4.2.4 Mesclagem dos esquemas - 4ª Fase

A última fase realiza a combinação das análises e casa os esquemas. Para que isto ocorra, os documentos *DSob*, *DDic* e *DDesC* são utilizados. O resultado é o documento *DElem* que reune todas as análises em um único documento. A Listagem 4.22 ilustra trechos do documento *DElem*, onde os elementos considerados equivalentes (tanto pela análise dos campos quanto pelas instâncias) são descritos, destacando as linhas 8, 9 e 18 que se referem aos elementos analisados nas Listagens 4.17 e 4.18. Como não há elementos no documento *DDic* o mesmo não é considerado.

```
1  [
2  ...
3  Title - Journal,
4  Title - ArticleTitle,
5  Year - Month,
6  Author - AuthorList,
7  ...
8  Last - LastName,
9  LastName - ForeName,
10 ...
11 Year - Day,
12 $a - Year,
13 ISOAbbreviation - MedlineTA,
14 $a - @PubStatus,
15 Day - Year,
16 SourceType - Tag,
17 Year - $a,
18 First - @ValidYN,
19 Month - Hour]
```

Listagem 4.22 – Documento *DElem*

Depois de combinar as análises, a próxima atividade é *Casar os esquemas*. A saída gerada são as informações sobre os esquemas, ou seja, quais elementos podem ser casados e a quais documentos de origem os mesmos pertencem. Para gerá-la, o documento com a estrutura (*DEst*) é necessário. Para obter o *Casamento dos esquemas* (Listagem 4.23) a consulta ao *DEst* determina os documentos de origem de cada elemento.

```
[
...
Documento 0 -> Title - Documento 1 -> Journal,
Documento 0 -> Title - Documento 1 -> ArticleTitle,
Documento 0 -> Year - Documento 0 -> Month,
Documento 0 -> JournalName - Documento 1 -> Journal,
Documento 0 -> Author - Documento 1 -> AuthorList,
Documento 0 -> Person - Documento 1 -> Author,
Documento 0 -> Last - Documento 1 -> LastName,
Documento 1 -> LastName - Documento 1 -> ForeName,
...
Documento 1 -> Year - Documento 1 -> Day,
Documento 1 -> $a - Documento 1 -> Year,
Documento 1 -> ISOAbbreviation - Documento 1 -> MedlineTA,
Documento 1 -> $a - Documento 1 -> @PubStatus,
Documento 1 -> Day - Documento 1 -> Year,
Documento 0 -> SourceType - Documento 0 -> Tag,
Documento 1 -> Year - Documento 1 -> $a,
Documento 0 -> First - Documento 1 -> @ValidYN,
Documento 1 -> Month - Documento 1 -> Hour
]
```

Listagem 4.23 – Casamento dos esquemas dos documentos 0 e 1

É importante notar que quando houver mais de dois documentos na coleção (para ser aplicado o processo para casamento de esquemas), os mesmos são testados de forma única, para que um especialista obtenha uma visão de todos os elementos equivalentes possíveis. Outra importante observação é que pode ocorrer de pares pertencentes ao mesmo documento JSON de origem serem considerados equivalentes.

Destacam-se os elementos considerados equivalentes no processo para casamento de esquemas proposto e que correspondem às mesmas entidades do mundo real: *SourceType - PublicationTypeList*; *Title - ArticleTitle*; *Author - AuthorList*; *JournalName - Journal*; *Person - Author* e *Last - LastName*.

Os documentos analisados neste estudo de caso passaram por todos os testes nas diversas atividades presentes em cada fase, gerando o casamento dos esquemas. Por questões de espaço, nem todos os pares de elementos foram ilustrados, mas pode-se afirmar que cerca de 130 pares equivalentes foram gerados, independentemente do tipo de análise. A Seção 4.3 realiza um estudo da validação do processo por meio de medidas de revocação e precisão.

4.3 VALIDAÇÃO

O objetivo desta seção é apresentar a validação do processo para casamento de esquemas descrito nesta dissertação, mais especificamente, no estudo de caso da Seção 4.2. Para realizar a validação foram utilizadas as medidas de revocação (*recall*) e precisão (*precision*).

A revocação se refere ao total de pares de elementos considerados equivalentes relevantes recuperados pelo processo. Já a precisão se refere aos pares de elementos recuperados que são relevantes. As Equações 4.1 e 4.2 descrevem como a revocação (R) e precisão (P) funcionam (STASIU, 2007).

$$Revocacao = \frac{Ra}{R} \tag{4.1}$$

$$Precisao = \frac{Ra}{A} \tag{4.2}$$

Onde: *Ra* se refere ao número de elementos retornados corretamente; *R* é o número de retornos que devem ser adquiridos; *A* é o número de elementos recuperados.

Os valores gerados na revocação e precisão dependem muito dos limiares estabelecidos (como pode ser observado no Capítulo 3). Com o intuito de escolher os melhores pontos de corte, este estudo realizou uma observação quanto às escolhas dos limiares, conforme a precisão e revocação, nos elementos ancestrais em um primeiro momento. A Tabela 5 ilustra os limiares estabelecidos para a análise dos elementos ancestrais com suas medidas de revocação e precisão.

Tabela 5 – Variações dos valores da revocação e precisão conforme os limiares estabelecidos

Ponto de corte 1	**Ponto de corte 2**	***Ra***	***R***	***A***	**Revocação**	**Precisão**
0,6	0,5	29	44	55	65,90%	52,72%
0,65	0,4	17	44	40	38,63%	42,5%
0,7	0,5	20	44	43	45,45%	46,51%

Fonte: Autor

É possível observar que os pontos de corte 0,6 e 0,5 são os melhores para serem aplicados e consequentemente serem utilizados nos elementos descendentes. O *Ponto de corte 1* se refere ao limiar estabelecido quando apenas uma das medidas de similaridades obtém valores no intervalo de 0 a 1. Já o *Ponto de corte 2* descreve o limiar da média ponderada nos casos onde mais de uma medida de similaridade está no intervalo de 0 a 1.

Assim, o estudo de caso relatado na Seção 4.2 obteve as seguintes porcentagens de revocação e precisão nos elementos testados, como pode ser visualizado na Tabela 6.

Tabela 6 – Valores de revocação e precisão para os elementos do estudo de caso

Ra	***R***	***A***	**Revocação**	**Precisão**
114	170	138	67,05%	82,60%

Fonte: Autor

Observa-se que o processo obteve 138 pares de equivalências nos elementos descendentes, sendo que 114 são considerados corretos. O número estimado de pares equivalentes que deveria ser adquirido é de 170. Portanto, as medidas aplicadas trazem bons resultados quanto à avaliação de revocação e precisão.

No estudo de caso apresentado anteriormente (Seção 4.2), foram destacados alguns pares de elementos equivalentes, resultado da aplicação do processo proposto nesta dissertação: *SourceType - PublicationTypeList*; *Title - ArticleTitle*; *Author - AuthorList*; *JournalName - Journal*; *Person - Author* e *Last - LastName*. Estes correspondem às mesmas informações do mundo real.

É importante destacar que o par de elementos *Title - ArticleTitle* (que se referem a títulos de publicações) foram corretamente considerados equivalentes, pois o processo considera a estrutura hierárquica dos documentos JSON. É possível notar que se não fosse levado em conta a estrutura hierárquica, os elementos que iriam ser considerados equivalentes eram *Title - Title* (pertencentes aos documentos 0 e 1 - Listagens 4.10 e 4.11). O primeiro se refere aos títulos de publicações e o segundo aos nomes de periódicos de publicações. Levar em conta as similaridades dos elementos ancestrais (neste caso, os elementos *Source* e *Journal*) foram essenciais para a determinação das equivalências entre os elementos.

Outro destaque são os elementos *Author - AuthorList* e *Person - Author*. As palavras *Author* se repetem, mas as designações são diferentes dentro da estrutura hierárquica dos documentos JSON. O primeiro par de elementos se refere a uma lista de autores, já o segundo indica os nomes de autores.

Assim como foi obtido inúmeras correspondências positivas, houve resultados onde os elementos não foram considerados equivalentes e deveriam, ou ainda, aqueles que obtiveram equivalências e não deveriam.

Um exemplo é o par de elementos *JournalName - Language*, que foram considerados equivalentes, mas não dizem respeito a mesma informação. Como os elementos ancestrais de *JournalName - Language* foram considerados equivalentes (*Source - Article*), os mesmos foram testados conforme o andamento do processo proposto.

Dentre as medidas de similaridades aplicadas em *JournalName - Language*, a de *Jaro Winkler* obteve valor de 0,626, enquanto as outras obtiveram valor 0 (zero). A justificativa é que os radicais não são iguais, e quanto à análise de conhecimento, a medida de *Wu & Palmer* não realiza o cálculo com nomes não válidos na língua inglesa (neste caso o elemento *JournalName*).

Uma observação sobre a análise dos radicais é que (no momento da extração dos radicais dos elementos) os caracteres especiais (como $, @, entre outros) não são considerados. Tendo isto em vista, alguns elementos tiveram que ser modificados para o cálculo ser realizado. Como pode ser visto no elemento *$a*, que originalmente era apenas o caractere *$*. É importante notar que este tipo de caso não impactou nos resultados obtidos.

Também são considerados equivalentes os elementos *Tag - Language*, que não se relacionam com as mesmas coisas. Neste caso, quando aplicado as medidas de similaridades, foram obtidos os valores de 0,57 e 0,64 (*Wu & Palmer* e *Jaro Winkler*, respectivamente), aplicando, assim, a média ponderada.

Foi possível notar que nestes casos (*JournalName - Language*; *Tag - Language*) os elementos foram considerados equivalentes devido a diferentes valores obtidos e que excederam os pontos de corte. Em alguns casos isto ocorre, mas se fosse realizado uma modificação nos valores dos pontos de corte impactaria nas equivalências obtidas corretamente.

Quando se trata de elementos que não foram determinados como equivalentes no casamento dos esquemas (mas que deveriam ter sido considerados), os elementos *StandardNumber1 - ISSN* podem ser citados. Os dois se referem aos números de ISSN das publicações. Como os elementos ancestrais não foram considerados equivalentes (*Source - Journal*), os mesmos não foram testados quanto a sua equivalência. Como mencionado anteriormente, a modificação nos valores dos pontos de corte retornaria elementos não corretos, impactando na validação do processo.

O casamento de esquemas proposto abre possibilidades para que um especialista consiga determinar quais são as equivalências válidas conforme seus objetivos. Por exemplo, nas Listagens 4.10 e 4.11 os pares *Author - Author* (documento 0, linha 9 - e documento 1, linha 45) são equivalentes, assim como os pares *Person - Author* (documento 0, linha 10 - e documento 1, linha 45). Ambos estão corretos, cabendo ao especialista determinar quais podem ser utilizados, assim como uma possível unificação dos mesmos.

4.4 CONSIDERAÇÕES FINAIS

Com o intuito de demonstrar como o processo para casamento de esquemas foi realizado, este capítulo traz informações importantes para a compreensão efetiva das diferentes atividades presentes nas fases.

Na Seção 4.1 trechos das implementações que foram realizadas são ilustrados por meio de listagens. A linguagem Java foi utilizada, sendo que diferentes classes foram criadas para as mais diversas atividades. Como o processo não é totalmente implementado, ou seja, algumas atividades foram realizadas manualmente, a seção trouxe apenas os passos realizados sem as atividades manuais.

Quanto ao estudo de caso, relatado na Seção 4.2, foi de extrema importância para acompanhar o passo a passo de todas as fases presentes no processo. Apesar de nem todos os resultados poderem ser visualizados, por questões de organização do texto, grande parte foi relatada e exemplificada, facilitando o entendimento de forma prática.

Os resultados alcançados puderam ser validados e descritos na Seção 4.3. As medidas de revocação e precisão foram explicadas e demonstradas as suas aplicações nos resultados obtidos no estudo de caso. Medir o quanto o processo trouxe bons resultados é de suma importância para demonstrar as contribuições deste estudo.

5 CONCLUSÃO E TRABALHOS FUTUROS

O estudo sobre integração de esquemas, de um modo geral, é explorado há muitos anos. As inúmeras dificuldades decorrentes das diversas heterogeneidades presentes nos dados fazem com que pesquisas voltadas para este tema ainda sejam exploradas. Uma das partes mais importantes que constitui a integração de esquemas é a realização do casamento dos esquemas. Determinar quais elementos podem ser equivalentes (entre os esquemas em análise, assim como os do mesmo esquema) é uma tarefa dificultosa e que geralmente necessita da intervenção do usuário.

A presente dissertação descreveu um estudo voltado para casamento de esquemas para documentos JSON. Este formato é um dos mais utilizados em banco de dados NoSQL orientados a documentos. Estudos voltados para o paradigma NoSQL é de sua importância, devido à sua ascensão nos últimos anos.

O processo apresentado é composto por 4 fases. A 1ª fase realiza um pré-processamento nos documentos JSON, extraindo informações que são utilizadas nas fases seguintes. Nesta fase o algoritmo *diff* é utilizado para determinar se são idênticos ou não. A extração das informações (estrutura, campos e instâncias) é realizada por meio de algoritmos descritos no Capítulo 3.

Uma análise do esquema é executada pela 2ª fase, onde a estrutura hierárquica dos documentos JSON é considerada. Isto porque, os elementos ancestrais são verificados quanto às suas similaridades e se os mesmos foram equivalentes, os seus elementos descendentes são também testados. O resultado são elementos descendentes considerados candidatos ao casamento e os elementos que não obtiveram equivalências. Levar em consideração a estrutura hierárquica é necessária, pois os documentos JSON são organizados de forma hierárquica.

As técnicas de similaridades apresentadas neste estudo e utilizadas na 2ª fase podem ser exploradas no Capítulo 2. A extração de radicais (algoritmo de (PORTER, 2006)), análise de caracteres (medida de *Jaro Winkler*) e análise do conhecimento (medida de *Wu & Palmer*) foram aplicadas sobre os campos (tanto ancestrais quanto descendentes) para determinar suas similaridades.

Os elementos não candidatos ao casamento de esquemas são utilizados para desenvolver a 3ª fase. Uma análise das instâncias é realizada para avaliar se os elementos possuem sobreposição de dados em suas instâncias. A medida de *Jaccard* foi utilizada para determinar as sobreposições de dados encontrados. Além disso, a 3ª fase leva em consideração o uso de um

dicionário de dados, mas é importante destacar que o mesmo só é possível de ser utilizado se estiver disponível no *dataset*.

Na 4ª fase é realizado o casamento dos esquemas propriamente dito. Por meio da combinação dos diversos resultados obtidos pelas análises anteriores, é realizado o casamento dos esquemas, ou seja, determinação de quais elementos podem ser considerados equivalentes.

Com o intuito de apresentar de forma mais satisfatória como o processo funciona, o Capítulo 4 descreveu as implementações realizadas na linguagem Java, um estudo de caso detalhado e a validação do mesmo. O estudo de caso foi importante, pois foi possível observar todas as etapas necessárias para executar o processo, tanto as atividades manuais quanto as implementadas.

Por meio da avaliação dos resultados obtidos no estudo de caso, a Seção 4.3, descreve as porcentagens obtidas nas medidas de revocação e precisão. Foi possível observar que tanto a revocação e a precisão obtiveram bons resultados (67,05% e 82,60%, respectivamente) em relação à quantidade de elementos analisados.

É importante notar que esta dissertação traz como diferencial a consideração da estrutura hierárquica dos documentos JSON para realizar a análise dos elementos, assim como a análise das instâncias contidas nos campos. Diversas pesquisas visam realizar o casamento de esquemas para os documentos XML, não levando em consideração todos os tipos de dados semiestruturados. Como o JSON é um formato leve e que está sendo cada vez mais utilizado, esta pesquisa oferece um processo para casamento de esquemas que determina quais elementos podem ser considerados equivalentes.

Alguns pontos interessantes como trabalhos futuros são:

- considerar mais de um nível de hierarquia para avaliar os ancestrais. O processo considera apenas um nível, ou seja, analisa um elemento acima do descendente (elemento pai). Por meio de mais níveis, talvez seja possível melhorar os resultados quanto ao casamento dos esquemas;

- aprimorar a determinação dos limiares por meio de técnicas e/ou estudos voltados exclusivamente para determinar bons limiares, tendo como objetivo melhorar a qualidade da separação dos elementos relevantes dos irrelevantes;

- adaptar e/ou incorporar o processo proposto nesta dissertação para ser utilizado em um processo completo de integração de esquemas;

- utilizar o algoritmo *diff* considerando apenas a estrutura e não todo o documento (estrutura e as instâncias), tendo em vista que pode ocorrer de documentos JSON possuirem instâncias diferentes, mas a estrutura ser idêntica. Isto impactaria na execução do processo, reduzindo as etapas de análises.

REFERÊNCIAS

ANHAI DOAN, A. H.; IVES, Z. Principles of Data Integration. In: . [S.l.]: Elsevier, 2012.

C. BATINI, M. L.; NAVATHE, S. B. A Comparative Analysis of Methodologies for Database Schema Integration. In: . [S.l.]: ACM Computing Surveys, Vol. 18, No. 4, 1986.

DIAS, S. A. Integração Semântica De Dados Através De Federação De Ontologias. In: . [S.l.]: Dissertação De Mestrado. Programa De Pós-Graduação Em Informática Do Departamento De Informática Do Centro Técnico E Científico Da Puc-Rio, 2006.

DIDIK DWI PRASETYA, A. P. W.; HIRASHIMA, T. The Performance Of Text Similarity Algorithms. In: . [S.l.]: International Journal Of Advances In Intelligent Informatics. Vol. 4, No. 1, Pp. 63-69., 2018.

ELMASRI, R.; NAVATHE, S. B. Fundamentals of database systems. In: . 6th.ed. [S.l.]: Pearson Education, Inc., publishing as Addison-Wesley, 2011.

FARKHUND IQBAL BENJAMIN C. M. FUNG, M. D. R. B.; MARRINGTON, A. Wordnet-Based Criminal Networks Mining for Cybercrime Investigation. In: . [S.l.]: IEEE Access. Digital Object Identifier 10.1109/ACCESS.2019.2891694, Volume 7, 2019.

GöSSNER, S. JSONPath - XPath for JSON. In: . [S.l.: s.n.], 2007.

INTERNATIONAL, E. The JSON Data Interchange Syntax. In: . [S.l.]: Disponível em *http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf*, 2017.

IPPOLITO, A. Um Estudo De Caso De Alinhamento (Matching) De Esquemas De Bancos De Dados Heterogêneos. In: . [S.l.]: Dissertação De Mestrado. Instituto De Pesquisas Tecnológicas Do Estado De São Paulo. São Paulo, 2012.

J. MADHAVAN, P. B.; RAHM, E. Generic Schema Matching With Cupid. In: . [S.l.]: In: Proc. 27 International Conference On Very Large Data Bases (Vldb), P. 48-58, 2001.

JARO, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. In: . [S.l.]: Journal of the American Statistical Association 84:414–420, 1989.

JARO, M. A. Probabilistic linkage of large public health data files (disc: p687-689). In: . [S.l.]: Statistics in Medicine 14:491–498. Joachims, T. 2002. Learning to Classify Text Using Support Vector Machines. Kluwer., 1995.

LAURI MUKKALA JUKKA ARVO, T. L.; KNUUTILA, T. TRC-Matcher And Enhanced TRC-Matcher - New Tools For Automatic XML Schema Matching. In: . [S.l.]: University Of Turku Technical Reports, No.13., 2017.

MACHADO, F. T. D. S. Um Processo Para Extração De Esquemas Conceituais Em Fontes De Dados Json Baseado Em Técnicas De Similaridade De Texto. In: . [S.l.]: Programa De Pós-Graduação Em Ciência Da Computação (Ppgcc). Universidade De Federal De Santa Maria (Ufsm), 2017.

MASSMANN, S. Evolution Of The Coma Match System. In: . [S.l.]: In: The Sixth International Workshop On Ontology Matching, 2001.

MEIKE KLETTKE, U. S.; SCHERZINGER, S. Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. In: . [S.l.]: In: BTW. Anais v.2105, p.425–444., 2015.

MILLER, G. A. Wordnet: a lexical database for english. In: . [S.l.]: Communications Of The Acm, [S.L.], V.38, N.11, P.39–41, 1995.

PETER MORK LEN SELIGMAN, A. R. J. K.; WOLF, C. The Harmony Integration Workbench. In: . [S.l.]: Part Of The Lecture Notes In Computer Science Book Series (Lncs, Volume 5383), 2006.

PIERRE BOURHIS JUAN L. REUTTER, F. S.; VRGOC, D. JSON: data model, query languages and schema specification. In: . [S.l.]: PODS'17, Chicago, IL, USA. ACM. ISBN 978-1-4503-4198-1/17/05., 2017.

PORTER, M. The Porter Stemming Algorithm. In: . [S.l.]: http://tartarus.org/martin/PorterStemmer/, 2006.

PORTER, M. F. An algorithm for suffix stripping. In: . [S.l.]: Program, 14(3):130–137., 1980.

RAHM, E.; BERNSTEIN, P. A. A Survey Of Approaches To Automatic Schema Matching. In: . [S.l.]: The Vldb Journal 10: 334–350 / Digital Object Identifier (Doi) 10.1007/S007780100057, 2001.

RANKING, D.-E. Trend Popularity. In: . [S.l.]: https://db-engines.com/en/ranking_trend, 2019.

SAATY, T. L. Decision Making With The Analytic Hierarchy Process. In: . [S.l.]: International Jornal Of Services Sciences, [S.L.], V.1, N.1, P.83–98, 2008.

SADALAGE, P. J.; FOWLER, M. NoSQL distilled: a brief guide to the emerging world of polyglot persistence. In: . [S.l.]: [S.l.]: Pearson Education, 2012.

SANTOS CARLOS A. HEUSER, V. P. Juliana B. dos; WIVES, L. K. Automatic Threshold Estimation For Data Matching Applications. In: . [S.l.]: Information Sciences, [S.L.], V.181, N.13, P.2685–2699, 2011.

SCHOFIELD, A.; MIMNO, D. Comparing Apples to Apple: the effects of stemmers on topic models. In: . [S.l.]: Transactions of the Association for Computational Linguistics, vol. 4, pp. 287–300, 2016. Action Editor: Hal Daume III., 2016.

STASIU, R. K. Avaliação da qualidade de funções de similaridade no contexto de consultas por abrangência. In: . [S.l.]: Tese (Doutorado em Ciência Computação)—Universidade Federal do Rio Grande do Sul., 2007.

TINGTING WEI YONGHE LU, H. C. Q. Z.; BAO, X. A semantic approach for text clustering using WordNet and lexical chains. In: . [S.l.]: Published by Elsevier Ltd. Expert Systems with Applications, 2014.

W. E. DJEDDI, M. T. K.; YAHIA, S. B. XMap: results for oaei 2015. In: . [S.l.]: OAEI, 2015.

WINKLER, W. E. String Comparator Metrics And Enhanced Decision Rules In The Fellegi-Sunter Model Of Record Linkage. In: . [S.l.]: U.S. Bureau of the CensusStat. Research Div., Rm. 3000-4, Washington, DC20223, 1990.

WINKLER, W. E. The state of record linkage and current research problems. In: . [S.l.]: Statistics of Income Division, Internal Revenue Service Publication R99/04. Available from http://www.census.gov/srd/www/byname.html, 1999.

WU, Z.; PALMER, M. Verb Semantics And Lexical Selection. In: . [S.l.]: In Proceedings Of The 32nd Annual Meeting Of The Association For Computational Linguistics, Las Cruces, New Mexico, 1994.