UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

KEYLLA RAMOS SAES

Abordagem para integração automática de dados estruturados e não
estruturados em um contexto *Big Data*

São Paulo

2019

KEYLLA RAMOS SAES

**Abordagem para integração automática de dados estruturados e não
estruturados em um contexto *Big Data***

Versão corrigida

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 22 de novembro de 2018. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Luciano Vieira de Araújo

São Paulo

2019

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)
CRB-8 4936

Saes, Keylla Ramos

Abordagem para integração automática de dados estruturados e não estruturados em um contexto *Big Data* / Keylla Ramos Saes ; orientador, Luciano Vieira de Araújo. – 2019.

103 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, em 2018.

Versão corrigida

1. Banco de dados 2. Banco de dados relacionais. 3. Big Data. I. Araújo, Luciano Vieira de, orient. II. Título.

CDD 22.ed.– 005.74

Dissertação de autoria de Keylla Ramos Saes, sob o título “**Abordagem para integração automática de dados estruturados e não estruturados em um contexto *Big Data***”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 22 de novembro de 2018 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Luciano Vieira de Araújo

Universidades de São Paulo
Escola de Artes, Ciências e Humanidades
Presidente

Prof. Dr. João Marcelo Borovina Josko

Universidade Federal do ABC

Prof. Dr. Marcelo Lauretto

Universidades de São Paulo
Escola de Artes, Ciências e Humanidades

Prof. Dr. Helton Hideraldo Biscaro

Universidades de São Paulo
Escola de Artes, Ciências e Humanidades

Aos meus pais Alfredo e Lucia, que são a luz no meu caminho...

Agradecimentos

A realização deste mestrado contou com importantes apoios e incentivos, sem os quais não teriam se tornado uma realidade e aos quais estarei eternamente grata.

Aos meus pais, Alfredo e Lucia, a base familiar foi o que possibilitou a conquista dos objetivos traçados em minha vida, auxiliando-me em todos os momentos com muita compreensão, confiança e incentivo.

Ao meu orientador Luciano, pelas proveitosas conversas, sempre mostrando-me perspectivas novas e amplas, dirigindo-me na longa jornada de aprendizagem e conhecimento.

Aos meus velhos amigos pela paciência nesse longo período de ausência e aos novos amigos mestrando do Programa de Pós-graduação em Sistemas de Informação (PPgSI) da EACH, pela oportunidade de aprendermos juntos, compartilhando experiências e momentos.

Às minhas amigas Fernanda e Elaine que me ajudaram nas revisões deste trabalho com muita paciência, boa vontade e amizade.

Ao meu noivo Elthon que compartilhou comigo os momentos difíceis dessa caminhada, apoiando-me sempre com seu carinho, respeito e amor.

Aos participantes da banca de qualificação pelas observações e direcionamentos que me auxiliaram a guiar este trabalho.

E, finalmente, agradeço à Deus pela minha vida, pela paz nos momentos em que me encontrei incapaz de prosseguir e pela força que me fez superar obstáculos e dificuldades e concluir com êxito este mestrado.

*“A maior recompensa para o trabalho do homem não é o que ele ganha com isso, mas o
que ele se torna com isso.”*

(John Ruskin)

Resumo

SAES, Keylla Ramos. **Abordagem para integração automática de dados estruturados e não estruturados em um contexto *Big Data***. 2019. 103 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2018.

O aumento de dados disponíveis para uso tem despertado o interesse na geração de conhecimento pela integração de tais dados. No entanto, a tarefa de integração requer conhecimento dos dados e também dos modelos de dados utilizados para representá-los. Ou seja, a realização da tarefa de integração de dados requer a participação de especialistas em computação, o que limita a escalabilidade desse tipo de tarefa. No contexto de *Big Data*, essa limitação é reforçada pela presença de uma grande variedade de fontes e modelos heterogêneos de representação de dados, como dados relacionais com dados estruturados e modelos não relacionais com dados não estruturados, essa variedade de representações apresenta uma complexidade adicional para o processo de integração de dados. Para lidar com esse cenário é necessário o uso de ferramentas de integração que reduzam ou até mesmo eliminem a necessidade de intervenção humana. Como contribuição, este trabalho oferece a possibilidade de integração de diversos modelos de representação de dados e fontes de dados heterogêneos, por meio de uma abordagem que permite o uso de técnicas variadas, como por exemplo, algoritmos de comparação por similaridade estrutural dos dados, algoritmos de inteligência artificial, que através da geração do *metadados integrador*, possibilita a integração de dados heterogêneos. Essa flexibilidade permite lidar com a variedade crescente de dados, é proporcionada pela modularização da arquitetura proposta, que possibilita que integração de dados em um contexto *Big Data* de maneira automática, sem a necessidade de intervenção humana.

Keywords: Dados estruturados. Dados não estruturados. Banco de dados relacionais. Banco de dados não relacionais. NoSQL. Integração de dados. Integração de dados heterogêneos. *Big Data*.

Abstract

SAES, Keylla Ramos. ***Approach for automatic integration of structured and unstructured data in a Big Data context.*** 2019. 103 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2018.

The increase of data available to use has piqued interest in the generation of knowledge for the integration of such data bases. However, the task of integration requires knowledge of the data and the data models used to represent them. Namely, the accomplishment of the task of data integration requires the participation of experts in computing, which limits the scalability of this type of task. In the context of Big Data, this limitation is reinforced by the presence of a wide variety of sources and heterogeneous data representation models, such as relational data with structured and non-relational models with unstructured data, this variety of features an additional complexity representations for the data integration process. Handling this scenario is required the use of integration tools that reduce or even eliminate the need for human intervention. As a contribution, this work offers the possibility of integrating diverse data representation models and heterogeneous data sources through the use of varied techniques such as comparison algorithms for structural similarity of the artificial intelligence algorithms, data, among others. This flexibility, allows dealing with the growing variety of data, is provided by the proposed modularized architecture, which enables data integration in a context Big Data automatically, without the need for human intervention.

Keywords: Structured data. Unstructured data. Relational database. Non-relational database. NoSQL. Data integration. Heterogeneous data integration. Big Data.

Lista de figuras

Figura 1 – Passos de uma abordagem de integração de dados	24
Figura 2 – Exemplo de sistema de gerenciamento de bancos federados (FDBMS) e seus componentes	27
Figura 3 – Exemplo de neurônio computacional	33
Figura 4 – Abordagens de integração em diferentes níveis arquiteturais	36
Figura 5 – Distribuição dos artigos nas categorias de arquiteturas de integração . .	40
Figura 6 – Distribuição de artigos nas categorias de arquiteturas de acordo com sincronismo do dado	41
Figura 7 – Distribuição temporal dos artigos por arquitetura de integração	41
Figura 8 – Distribuição dos modelos de representação de dados não estruturados nos artigos avaliados	42
Figura 9 – Estrutura de organização de padrão de dados de domínio da Web . . .	43
Figura 10 – Processo de integração entre dados não sensíveis disponibilizados na nuvem e dados sensíveis disponibilizados localmente	45
Figura 11 – Arquitetura geral com base em IoT	46
Figura 12 – Abordagem mediador Wrapper	47
Figura 13 – Arquitetura de federação de dados Unity	49
Figura 14 – Exemplo conceitual de integração	54
Figura 15 – Exemplo detalhado de integração de dados heterogêneos	55
Figura 16 – Abordagem proposta para integração de dados heterogêneos	56
Figura 17 – Avaliador de representação	61
Figura 18 – Inteligência de integração	65
Figura 19 – Exemplo de uso de metadados integrador como entrada de um algoritmo de agrupamento	67
Figura 20 – Exemplo de uso de metadados integrador como entrada de um algoritmo de rede neural	67
Figura 21 – Executor de Integração	69
Figura 22 – Estatística de integração	70
Figura 23 – Utilização do arquivo integrado no Dublin Core	71
Figura 24 – Interface de envio de dados para análise e integração automática	73

Figura 25 – Exemplo de Metadados Integrador	74
Figura 26 – Interface de sugestão de integração	80
Figura 27 – Interface com exemplo das estatísticas de processamento da integração e avaliação de resultados	80
Figura 28 – Exemplo de arquivo gerado pela integração consolidada	81
Figura 29 – Interface para geração de metadados Dublin Core	81
Figura 30 – Experimento 01	84
Figura 31 – Experimento 02	84
Figura 32 – Experimento 03	85
Figura 33 – Experimento 04	86
Figura 34 – Experimento 05	87
Figura 35 – Experimento 06	87
Figura 36 – Experimento 07	88
Figura 37 – Experimento 08	89
Figura 38 – Experimento de integração de dados heterogêneos	89
Figura 39 – Diagrama do processo de seleção de busca	102

Lista de algoritmos

Algoritmo 1 – Normaliza dados pelo maior valor da lista recebida	75
Algoritmo 2 – Realiza agrupamento dos atributos dos metadados integradores	78
Algoritmo 3 – Executa uma rede neural para identificação dos atributos similares dos metadados integradores	79

Lista de tabelas

Tabela 1 – Exemplos de tipos de dados que podem gerar erros em um processo de integração	25
Tabela 2 – Tabela com elementos Dublin Core simples	30
Tabela 3 – Casos de uso para integração de dados	37
Tabela 4 – Arquiteturas de integração: vantagens e desvantagens	39
Tabela 5 – Grau de priorização das arquiteturas	40
Tabela 6 – Atributos do <i>Metadados Integrador</i>	64
Tabela 7 – Casos de uso de integração de dados estruturados e não estruturados e suas complexidades	83
Tabela 8 – Critérios de inclusão e exclusão	102
Tabela 9 – Extração de informação de artigos relacionados às arquiteturas de integração entre dados estruturados e não estruturados	103

Lista de abreviaturas e siglas

BI	<i>Business Intelligence</i>
DCMI	<i>Dublin Core Metadata Initiative</i>
ETL	<i>Extraction, Transformation and Load</i>
FDBMS	<i>Federation meta-database management system</i>
MER	Modelo entidade relacionamento
NoSQL	<i>Not only SQL</i>
RDF	<i>Resource Description Framework</i>
SGBD	Sistemas gerenciadores de banco de dados
SQL	<i>Structured query language</i>
XML	<i>eXtensible Markup Language</i>

Sumário

1	Introdução	17
1.1	<i>Objetivos</i>	20
1.2	<i>Metodologia</i>	20
1.3	<i>Contribuições esperadas</i>	21
1.4	<i>Justificativa</i>	21
2	Integração de dados	23
2.1	<i>Integração de dados estruturados</i>	23
2.2	<i>Integração de dados não estruturados</i>	25
3	Federação de dados e metadados	27
3.1	<i>Banco de dados federados</i>	27
3.2	<i>Padrão de metadados Dublin Core</i>	28
4	Aprendizagem aplicada a integração de dados	31
4.1	<i>Modelos de Aprendizagem</i>	32
4.1.1	<i>Redes Neurais Artificiais</i>	33
4.1.2	<i>Agrupamentos ou Clustering</i>	34
4.2	<i>Integração de dados com o uso de técnicas de aprendizagem</i>	34
5	Trabalhos relacionados	36
5.1	<i>Arquiteturas de integração</i>	36
5.2	<i>Trabalhos correlatos à revisão sistemática</i>	37
5.3	<i>Análise da Revisão Sistemática</i>	38
5.4	<i>Classificação dos artigos nas arquiteturas de integração</i>	42
5.5	<i>Discussões sobre a revisão sistemática</i>	50
6	Abordagem proposta para o Integrador de dados	53
6.1	<i>Integração automática de dados estruturados e não estruturados</i>	53
6.2	<i>Arquitetura do Integrador de dados</i>	55
6.2.1	<i>Fluxo dos dados no processo de integração</i>	57
6.3	<i>Módulos da arquitetura do Integrador de dados</i>	58

6.3.1	Gestão de parâmetros	58
6.3.2	Arquivos de entrada: Acesso aos dados e metadados	59
6.3.3	Semântica conhecida pelo usuário	59
6.3.4	Avaliador de representação	59
6.3.5	Inteligência de integração	65
6.3.6	Executor de integração	68
6.3.7	Estatísticas da integração	70
6.3.8	Arquivo I - Dados Integrados	71
6.3.9	Arquivo M - Metadados dos dados integrados	71
6.3.10	Retorno da integração	71
7	Ferramenta: Integrador de Dados	73
7.1	<i>Integrador de Dados</i>	73
7.2	<i>Normalização de dados</i>	74
7.2.1	Normalização de dados numéricos	75
7.2.2	Normalização de dados Categóricos	75
7.3	<i>Componente de sugestão de integração</i>	76
8	Experimentos	83
8.1	<i>Casos de uso para integração de dados heterogêneos</i>	83
8.2	<i>Experimento 01 - Integração de dados do Bolsa família com uso do algoritmo de similaridade</i>	83
8.3	<i>Experimento 02 - Integração de dados do Bolsa família com uso do algoritmo de agrupamento</i>	84
8.4	<i>Experimento 03 - Integração de dados do Bolsa família entre os anos de 2016 e 2018</i>	85
8.5	<i>Experimento 04 - Integração de dados do Bolsa família com uso do algoritmo de rede neural</i>	85
8.6	<i>Experimento 05 - Integração de dados do Bolsa Família com Bolsa alimentação PRACE</i>	86
8.7	<i>Experimento 06 - Integração de cadastro de cachorros e imagens de cachorros</i>	87
8.8	<i>Experimento 07 - Integração de imagens de sapatos</i>	88

8.9	<i>Experimento 08 - Caso de uso</i>	88
8.10	<i>Proposta para futuros Experimentos</i>	89
9	Discussões, conclusão e trabalhos futuros	90
9.1	<i>Conclusão</i>	91
9.2	<i>Trabalhos futuros</i>	92
	Referências¹	93
	 APÊNDICES	 100
.1	<i>Protocolo da revisão sistemática</i>	101
.2	<i>Artigos detalhados da revisão sistemática</i>	103

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Os sistemas de banco de dados mais populares têm sido aqueles que utilizam dados estruturados baseados no modelo de representação relacional e manipulam dados com a linguagem SQL (*Structured Query Language*) (CODD, 1970). Com o avanço da internet e a popularização do uso das redes sociais apresentado em Bello-Orgaz, Jung e Camacho (2016), surgiram novas representações de dados e uma revolução começou a acontecer com os Sistemas Gerenciadores de Banco de Dados (SGBD), como foi apresentado por Lawrence (2014a). Esses diversos modelos de representações (modelo de grafo, modelo de documento, modelo chave e valor e modelo híbrido) possuem dados armazenados em banco de dados conhecidos como NoSQL (*Not Only SQL*) Han et al. (2011), e são manipulados por linguagens específicas de cada SGBD NoSQL. (PADHY; PATRA; SATAPATHY, 2011) Neste trabalho, o termo dado não estruturado é utilizado para definição de dados não relacionais, ou seja, dados que não se adequam em uma representação organizada em um modelo relacional, baseada em tuplas e atributos como foi apresentado em Blumberg e Atre (2003) São considerados não estruturados textos longos, grafos, imagens, vídeos etc. Mesmo que tais dados possam ser armazenados nos SGBDs relacionais por meio de um atributo BLOB (Binary Long Object), serão considerados não estruturados devido à sua forma de organização, diversidade, tamanho e principalmente por cada tipo não estruturado conter diferentes dados e informações, por exemplo, uma imagem contém uma informação alvo que difere de um grafo. (STOREY; SONG, 2017)

O grande volume de dados disponível na Internet, nas empresas e instituições tem sido referido com o uso do termo em inglês - *Big Data* (Grande dado em português) (BRYANT; KATZ; LAZOWSKA, 2008). *Big Data* foi definido inicialmente na literatura pelo acrônimo 3 V's Chen, Mao e Liu (2014): volume, variedade e velocidade. O termo volume refere-se à quantidade de dados que necessitam ser processados e que aumentam a cada dia; o termo variedade refere-se à diversidade de fontes de dados e da multiplicidade de formas como os dados são representados. Ou seja, dados relacionais, documentos, texto, sons, vídeos, fotos, postagens em redes sociais e etc. O termo velocidade refere-se à velocidade com que os dados são gerados e conseqüentemente à necessidade de recursos para processá-los a contento.

O conceito de 3 V's (volume, variedade e velocidade) foi ampliado por outros trabalhos

que definem inclusive o conceito representado pelo acrônimo 5 V's (volume, variedade, velocidade, veracidade e valor), como é o caso de Lomotey e Deters (2014) que acrescenta os termos veracidade e valor. Essa extensão do conceito inicial está diretamente relacionada ao contexto de utilização do dado, sendo veracidade definida como precisão dos dados após a aplicação de métodos de limpeza e qualidade de dados, e valor está definido pelo custo em coletar, tratar e armazenar o dado. Além disso, o dado passa a ser tratado como uma mercadoria, passando a ter valor e inclusive podendo ser vendido. O valor intrínseco do dado e o seu custo de armazenamento pode ser usado para avaliar a viabilidade de armazenar um dado ou não, ou seja, se o valor intrínseco do dado é menor que o seu custo de armazenamento, por que mantê-lo?

Algumas análises necessitam combinar dados distribuídos em bancos de dados distintos. A tarefa de combinar esses dados, seja transportando-os, seja realizando apontamentos aos dados é conhecida como integração de dados.(RAHM, 2016) O interesse pela descoberta de informação e conhecimento em dados cresce junto com o volume e a variedade disponível para análise. Entretanto, a manipulação e gestão desses dados de grande variedade e volume apresentam desafios logísticos significativos. Oxford... (2018)

A integração de dados não é um processo trivial, nem mesmo nos casos básicos, nos quais os modelos de representação são homogêneos. Por exemplo, um caso no qual os dados são representados segundo o mesmo modelo de dados, apesar de estarem representados em bancos de dados distintos. Neste caso, cuja representação é única, os dados atendem às regras do modelo relacional, sendo estruturados de acordo com as normas do modelo entidade relacionamento. (CHEN, 1988) Assim, as tabelas e seus campos apresentam uma estrutura definida, os atributos seguem a ordem e tipo de dados de cada tabela. Mesmo com uma definição clara da estrutura e com o uso do mesmo modelo de representação, a integração dos dados ainda demanda conhecimento sobre dados armazenados, o que demanda especialistas de computação para sua interpretação.

O desafio de integração é ampliado ao se considerar dados que estão representados em modelos heterogêneos, como é o caso de integração com dados não estruturados ou *NoSQL*. Um exemplo dessa dificuldade é a análise integrada dos sistemas de apoio a decisão que geram perfis de clientes baseados em históricos de compras. Essas informações podem ser extraídas de sistemas de compras e armazenadas em banco de dados relacional específico para o negócio, processo este também conhecidos como *Business intelligence*

ou simplesmente *BI* (CHEN; CHIANG; STOREY, 2012). Com o surgimento das redes sociais e a interação direta com o cliente final, cresce a possibilidade de conhecer a opinião do cliente sobre as empresas e produtos disponibilizados em comentários nas redes sociais, sendo que esses dados são armazenados em modelos de representação não relacional, como, por exemplo, o modelo documento. Ou seja, a capacidade de análise poderia ser ampliada com a análise conjunta dos dados (perfil por histórico de compra e opinião do cliente), porém existem desafios a serem superados para a realização deste tipo de análise devido aos dados encontrarem-se em modelos de representação distintos e não integrados.

Os sistemas computacionais são projetados para atender a um processo de negócio específico, e geralmente não são projetados para integração com outros sistemas, de forma isolada. Assim, quando se deseja o acesso integrado a diferentes sistemas, as fontes e seus dados que normalmente não se encaixam devem ser agrupados e adaptados por funcionalidade, regras de negócio e conciliações adicionais deverão ser implementadas. (ZIEGLER; DITTRICH, 2004) Embora o objetivo seja fornecer uma visão homogênea e unificada de dados de diferentes fontes, a tarefa de integração pode depender de alguns aspectos, sendo eles:

1. visão arquitetônica de um sistema de informação;
2. conteúdo e a funcionalidade dos sistemas de componentes;
3. tipo de informação gerida por sistemas de componentes (dados alfanuméricos, dados multimídia dados estruturados, semiestruturados e não estruturados);
4. requisitos relativos à autonomia de sistemas de componentes;
5. uso pretendido de sistema integrado de informações (acesso somente leitura ou gravação),
6. requisitos de desempenho;
7. recursos disponíveis (tempo, dinheiro, recursos humanos, know-how, etc.).

Segundo Ziegler e Dittrich (2004) vários tipos de heterogeneidade geralmente precisam ser considerados em um processo de integração de dados. Estas incluem diferenças em:

1. hardware e sistemas operacionais;
2. software de gerenciamento de dados;
3. modelos de dados, esquemas e semântica de dados;
4. regras de negócios;
5. restrições de integridade.

A variedade de modelos de representação dos dados existentes cria um desafio para a realização das tarefas de integração de dados, como: acesso, avaliação e transformação dos dados. Em um contexto Big Data, além da variedade, o grande volume dos dados amplia a complexidade dessas tarefas de integração. Esse cenário gera oportunidades para o surgimento de abordagens que apresentem alternativas para integração automática de dados representados em modelos heterogêneos. Esse trabalho avalia o estado da arte para integração de dados estruturados e não estruturados e apresenta uma proposta de abordagem para integração automática desses dados heterogêneos. No capítulo 2 apresenta-se a fundamentação sobre integração de dados estruturados e não estruturados. O capítulo 3 apresenta a conceituação sobre federação de dados e metadados, conceito que embasa parte da abordagem proposta. No capítulo 4 são apresentados os conceitos sobre aprendizagem de máquina e discute-se sua aplicação como estratégia para integração de dados. O capítulo 5 traz os trabalhos relacionados através de uma revisão sistemática. O capítulo apresenta a abordagem proposta e o capítulo 7 traz detalhes da implementação que foi realizada, assim como o capítulo 8 apresenta os resultados dos experimentos que são discutidos no capítulo 9, que encerra com a conclusão deste trabalho.

1.1 Objetivos

O objetivo geral deste trabalho é propor uma abordagem para integração automática de dados estruturados e não estruturados em um contexto de Big Data.

Objetivos específicos:

- Avaliar a literatura relacionada ao tema;
- Propor uma abordagem para integração automática de dados estruturados e não estruturados de forma a possibilitar análises integradas;
- Oferecer uma arquitetura para reduzir o esforço de integração de dados;
- Validar a abordagem proposta com dados disponíveis em bancos de dados públicos.

1.2 Metodologia

Para desenvolvimento deste trabalho, utilizou-se a seguinte metodologia:

- Levantamento bibliográfico para identificação do estado da arte da integração de dados heterogêneos;
- Avaliação de técnicas de integração de dados heterogêneos e enriquecimento de dados;
- Definição de uma arquitetura para identificar atributos integradores entre banco de dados;
- Proposição uma interface visual para configuração do processo de integração realizada com a implementação na linguagem Python 3.54, com o uso do Framework Django para aplicações Web em uma máquina Intel Core i5 2.20GHz, 8GB de memória e sistema operacional Windows de 64 bits;
- Definição de atributos de avaliação semântica dos dados pelo *metadados integrador*.

1.3 Contribuições esperadas

Oferecer uma abordagem para a integração de dados estruturados e não estruturados de forma automática, de forma a permitir maior agilidade na extração de conhecimento. Com uma abordagem de integração de dados heterogêneos automática, o esforço manual realizado em processos de integração de dados será diminuído através de contribuições realizadas pela abordagem de integração proposta:

- Integração de dados realizada automaticamente com busca em histórico de integrações previamente realizadas, ou identificadas no processamento atual com modelos inteligentes;
- Desenvolvimento de uma estrutura de *Metadados Integrador* que permita a comparação de dados homogêneos e heterogêneos;
- Integração de dados realizada de forma automática pelas sugestões obtidas no *Metadados Integrador*, que poderão ser apenas validadas, se desejado e parametrizado na arquitetura.

1.4 Justificativa

Para realizar a integração de dados é necessário o conhecimento técnico e funcional sobre o dado que está sendo tratado de forma a desenvolver uma abordagem adequada

de representado dos dados. O desafio para desenvolver essa representação é que, segundo Doshi et al. (2013), cresce diariamente a variedade e o volume de dados.

Essa variedade e volume de dados adicionam um grau de complexidade a atividade de mapeamento dos atributos a serem integrados, que são realizados de forma manual e por especialistas em computação na maioria dos casos, e automatizados em ferramentas de integração como ferramentas de ETL ou ferramentas de ingestão de dados em um contexto Big Data. Caso ocorram alterações nas estruturas dos dados ou em seu conteúdo o processo de integração precisa ser manualmente revisto, o que dificulta escalar este tipo de atividade pela limitação de especialistas em computação.

Com isso, este trabalho propõe uma abordagem de integração automática para apoiar o trabalho de integração de dados heterogêneos, simplificando as atividades dos especialistas em computação.

2 Integração de dados

Um processo de integração de dados visa unificar dados, provendo um acesso uniforme a dados oriundos de múltiplas origens, ou seja, permitir que dados localizados em diversos locais possam ser acessados como se estivessem juntos.(RAHM, 2016)

A integração de dados pode ser realizada através de processos de transporte de dados, conhecidos como ETL para arquiteturas de Datawarehouse ou como ingestão de dados no caso de arquiteturas Big Data. A integração de dados também ocorre por processos de apontamento de dados, ou seja, quando não existe movimentação de dados. Esse processo de apontamento é conhecido como dados federados. Para a realização de um processo de integração, a estrutura do dado é um ponto que deve ser considerado. Esta seção apresentará os desafios da integração de dados estruturados e não estruturados.

2.1 Integração de dados estruturados

Desde a criação dos primeiros computadores, diversos sistemas foram desenvolvidos nas organizações para atuar em áreas específicas, solucionando problemas operacionais e transacionais, por exemplo, sistemas de geração de pedidos de vendas, sistemas contábeis, sistemas de estoque, sistemas de atendimento, cadastros de produtos e clientes. Esses sistemas, em sua maioria, foram desenvolvidos com arquiteturas distintas entre si, com diferentes equipes, tecnologias na camada de aplicação e na camada de banco de dados. Esse é um cenário comum na maioria das organizações, que encontram um ecossistema legado com aplicações de tecnologias distintas, sendo sua complexidade proporcional ao tamanho da organização. Quando os gestores dessas organizações necessitam realizar análises diversas sobre o andamento da operação para tomada decisões, essa diversidade de sistemas torna-se um problema. Para tratar esse tema, em meados dos anos 80 e 90 popularizaram-se as técnicas de aplicações analíticas, com uma proposta de abstrair as diversas tecnologias transacionais, com processos de integração de dados em um repositório único conhecido por Data warehouse Kimball e Ross (2011), que servia como fonte de análise com uma visão consolidada dos dados estruturados da organização, em uma área que surgia conhecida Business intelligence (BI).

O artigo de Havel et al. (2014) exemplifica as principais tarefas para uma abordagem utilizada de integração de dados estruturados, descritas na Figura 1, que incluem o pré-processamento de dados (limpeza e enriquecimento de dados), resolução de entidade (correspondência de dados), fusão de entidade (junção de modelos de metadados como esquemas e ontologias).

Figura 1 – Passos de uma abordagem de integração de dados

Fonte: Adaptado de (HAVEL et al., 2014)

Em uma abordagem desse tipo apresentada na Figura 1, a integração de dados é realizada por etapas, com objetivo de extrair dados em diferentes fontes e estruturas e agrupando-os em um modelo de representação unificado. Tradicionalmente, isso requer o desenho de um processo comum de extração, transformação e carga (ETL) que é aplicado separadamente à cada fonte de dados diferente, unificando a estrutura para o modelo destino, normalmente um data warehouse com modelo de representação analítico. Uma das principais atividades desse processo é a realização de um mapeamento dos dados do modelo origem para o modelo destino gerando, quando necessário, regras de validação. Essa atividade é realizada manualmente por especialistas, que necessitam conhecer os modelos de dados, a tecnologia envolvida e as regras de negócio representadas nos dados. Posteriormente o processo manual é automatizado em um ETL que é repetidamente executado para incorporar inclusões de novos dados de origem no modelo unificado. (LEE et al., 2015a)

Esse tipo de abordagem de integração necessita de processos de conversão de tipo de dado que normalizem as estruturas do dado de origem e destino. Por exemplo, qualquer alteração no tipo do dado (alfanumérico, numérico, data, etc), pode gerar erros durante o processamento da integração, podendo em alguns casos para-la totalmente, por exemplo, alterar um atributo alfanumérico para numérico ou alterar um atributo data para numérico ou um atributo numérico para data, como pode ser visto na Tabela 1.

Tabela 1 – Exemplos de tipos de dados que podem gerar erros em um processo de integração

Tipo dado destino	Tipo dado recebido na integração	Resultado
Numérico	Alfanumérico	Erro tipo dado
Data	String	Erro tipo dado
Data	Numérico	Erro tipo dado

Fonte: Keylla Ramos Saes, 2018

A Tabela 1 apresenta três exemplos de dados de origem em estruturas distintas da estrutura de destino, que se existirem em um processo de integração poderão gerar erros na execução do processo de integração de dados estruturados. Esse tipo de problema deve ser identificado na etapa de mapeamento dos dados, antes do desenvolvimento e execução da integração. Existem conceitos que auxiliam o mapeamento dos dados em um processo de integração dos dados, por exemplo, as correspondências semânticas, os metadados, as ontologias e esquemas de banco de dados. O mapeamento de dados permite que os computadores expressem dados semânticos e interprete as definições existentes entre as entidades. Entretanto as mudanças nos dados geram a necessidade de revisão desses instrumentos de mapeamentos e podem ser um obstáculo para a integração automática de dados, devido a dependência de especialistas humanos para sua revisão. (REIS; PRUSKI; REYNAUD-DELAÎTRE, 2015)

2.2 Integração de dados não estruturados

A integração de dados estruturados apresenta o desafio de normalizar os conceitos de negócio e tipos de dados em modelos de representação homogêneos. A normalização é necessária para que conceitos similares representados de forma distinta possam ser unificados durante o processo de integração. Uma das dificuldades para a realização da normalização de dados é entender o significado do dado. (HARIKUMAR; REETHIMA; KAIMAL, 2014)

Já a integração de dados estruturados com dados não estruturados apresenta um desafio predecessor a normalização de conceitos, que se referente à normalização do modelo de representação de cada estrutura a ser utilizado, pela identificação de elementos que possam ser utilizados para a correspondência de dados, entre um modelo relacional com outros modelos de representação distintos, sejam o modelo de grafo, modelo documento, modelo chave-valor ou modelo híbrido.

O problema de correspondência de dados pode ser definido como a tarefa de encontrar relações semânticas entre elementos (por exemplo, atributos de relação como chaves primárias e estrangeiras, tags XML ou tags json) de dois modelos de representação distintos. Resolver esse problema é essencial em ambientes de integração de dados que usam uma variedade de fontes de dados, com bancos de dados heterogêneos, data warehouses e repositórios de dados da Web. Este problema geralmente ocorre devido as diferentes possibilidades de expressar um mesmo conceito do mundo real na representação do dado, pela diversidade de modelos de representações passíveis de serem utilizado, como por exemplo: modelo entidade relacionamento, modelo de documentos, modelo de grafos, modelo chave-valor e modelo híbrido (CARVALHO et al., 2013).

A revisão sistemática apresentada neste trabalho, classificou em 5 arquiteturas de integração de modelos de representação de dados heterogêneos:

1. Arquitetura customizada;
2. Arquitetura MapReduce;
3. Arquitetura de componentes geográficos;
4. Arquitetura middleware ou serviços de dados;
5. Arquitetura federação ou virtualização de dados.

Como foi apresentada nessa revisão, as arquiteturas possuem níveis diferentes customização, sendo muitas desenvolvidas para resolver problemas específicos, e diferentes níveis de automatização, dependendo em sua maioria de intervenção de especialistas de computação, que é um limitante devido à variedade e volume de dados.

Os bancos de dados não estruturados ou NoSQL muitas vezes não possuem metadados bem definido, o que pode limitar algumas atividades de mapeamento de dados durante o processo de integração de dados. Existem na literatura propostas de técnicas para mapeamento das informações dispersas na rede, por exemplo, Linked Data Bizer, Heath e Berners-Lee (2009), Web Semântica Berners-Lee, Hendler e Lassila (2001) e Ontologias Bechhofer (2009) que podem auxiliar nas atividades de entendimento do conteúdo e relações entre os dados. Entretanto, essas propostas ainda necessitam de especialistas da computação para defini-las e mantê-las, o que dificulta escalar o processo de integração de dados.

3 Federação de dados e metadados

3.1 Banco de dados federados

Uma das arquiteturas apresentadas na revisão sistemática realizada neste trabalho foi arquitetura federação ou virtualização de dados, também definida na literatura Sheth e Larson (1990), como sistema de federação de dados. Sistemas de banco dados federados permitem que bancos de dados com modelos de representação heterogêneos cooperem entre si, por um componente de coordenação que permite que dados dispersos em modelos de representação distintos, armazenados em distintos bancos de dados possam ser acessados de uma forma centralizada, conforme descrito na Figura 2.

Figura 2 – Exemplo de sistema de gerenciamento de bancos federados (FDBMS) e seus componentes

Fonte: Adaptado de (SHETH; LARSON, 1990)

Nos sistemas de bancos federados pode-se encontrar os seguintes componentes de arquitetura: dados, banco de dados, comandos, processadores, esquemas e metadados, e mapeamentos. O uso de metadados é uma possível abordagem para integrar dados heterogêneos.

Metadados são informações descritoras que permitem o entendimento sobre o dado, Kolaitis (2005) também conhecido como "dados sobre o dado". Os metadados oferecem um conjunto de dados chamados de elementos, cujo número é variável de acordo com o padrão, sendo esses elementos informações sobre o dado, como por exemplo: origem, tipo, descrição, qualidade, entre outros. Especialmente para banco de dados relacionais com

informações estruturadas a criação e manutenção de metadados é relevante, pois facilita os processos de tratamento, extrações, transformações, integrações e análises que podem ser realizados sobre esses dados Serhani et al. (2016).

Existem diversos tipos de metadados Duval (2001), como os que descrevem tamanho, tipo do dado e chaves, ou metadados operacionais, que descrevem quem produziu o dado, quando foi atualizado, quando foi acessado e quem consome aquele dado, por exemplo. Já os metadados de negócio Hüner, Otto e Österle (2011), descrevem o nome do termo de negócio, regras de validação, entre outros. Algumas organizações mantêm estruturas de metadados com objetivos de facilitar o acesso e a integração de diferentes dados, cita-se dentre elas:

- Dublin Core (WEIBEL et al., 1998)
- DSpace (SMITH et al., 2003)
- DataOne (ALLARD, 2012)
- Ckan (WINN et al., 2013)

A seguir é apresentado o padrão Dublin Core, como um exemplo de metadados utilizado para compartilhamento de dados em bancos federados.

3.2 Padrão de metadados Dublin Core

O Dublin Core é um padrão de metadados criado e atualizado pela Dublin Core Metadata Initiative (DCMI) a partir de um primeiro workshop que foi realizado, e é mantido por profissionais que se reúnem anualmente para discutir o tema. O Dublin Core tem como objetivo criar mecanismos que facilitem a recuperação de dados na Internet, utilizando-se de padrões de metadados conforme descrito em Weibel et al. (1998). O padrão Dublin Core possui três características:

- **Simplicidade:** elementos com entendimento semântico simples, sem a necessidade de longos treinamentos para seu entendimento;
- **Interoperabilidade semântica:** modelo comum que aumenta a possibilidade de interoperabilidade entre diversas áreas;
- **Flexibilidade:** seus elementos são opcionais, e podem ser repetidos se necessário, e modificáveis utilizando-se de qualificadores.

A interoperabilidade com metadados Dublin Core é classificada em quatro níveis, são eles:

1. **Definições de termos compartilhados:** São usadas definições linguísticas naturais, os termos podem ser referenciados de qualquer maneira, ou seja, conformidade com domínios e intervalos específicos não é um requisito, o texto é de escrita livre.
2. **Interoperabilidade semântica formal:** A interoperabilidade semântica é baseada no uso preciso e correto da semântica RDF (*Resource Description Framework*) formal por Klyne e Carroll (2006), incorporada no modelo de dados de grafo RDF e em vocabulários baseados em RDF. Este nível corresponde ao uso implícito ou explícito dos semânticos RDF subjacentes aos termos DCMI. Assim, qualquer uso dos termos precisa ser preciso na sua conformidade com o modelo RDF, os domínios e intervalos de termos.
3. **Descrição interoperabilidade sintática:** Com os grafos especificados pelo RDF, o modelo direciona noções de descrições e conjunto de descrições, fornecendo uma base para validação e troca de registros de metadados.
4. **Descrição interoperabilidade do perfil:** Fornece um modelo de informação e expressão XML de restrições estruturais em um conjunto de descrições, com um pacote de documentação necessários para apresentar um aplicativo de metadados para máxima interoperabilidade e reutilização - elementos como requisitos funcionais, um modelo de domínio e um conjunto de descrições de perfil abrangendo o conjunto completo de metadados.

O padrão Dublin Core possui dois níveis: simples e qualificado. O Dublin Core simples compreende quinze elementos não obrigatórios, que podem inclusive aparecer diversas vezes. O Dublin Core qualificado possui os mesmos elementos do nível simples, incluindo três elementos adicionais (audiência, proveniência e titular dos direitos), bem como um grupo de qualificadores que refinam a semântica dos elementos. A lista de componentes do Dublin Core simples é detalhada na Tabela 2, cuja primeira coluna é apresentado o tipo de tag, que é um agrupador dos elementos e está distribuído em 3 grupos:

1. Relacionados ao conteúdo do recurso (Conteúdo);
2. Relacionados a propriedade intelectual (Propriedade Intelectual);

3. Relacionados a utilização do recurso (Utilização).

A segunda coluna da Tabela 2 apresenta os elementos aceitos no metadado Dublin Core e a terceira coluna uma descrição sobre o uso e domínio de cada tag ou elemento.

Tabela 2 – Tabela com elementos Dublin Core simples

Tipo	Tag (elemento)	Descrição
Conteúdo	Title (Título)	Nome pelo qual o recurso é formalmente conhecido.
Propriedade Intelectual	Creator (Autor)	Pode uma pessoa ou uma organização responsável pela criação do conteúdo intelectual.
Conteúdo	Subject (Assunto / palavras-chave)	O assunto será expresso com palavras-chave, descritores ou códigos de classificação que descrevem o tema. O uso de vocabulários são controlados.
Conteúdo	Description (Descrição)	Pode incluir tabelas de conteúdo, referências para uma representação de conteúdo ou um texto ou resumo do conteúdo.
Propriedade Intelectual	Publisher (Editor)	Inclui uma pessoa, uma organização ou serviço responsável por disponibilizar o recurso em sua forma atual.
Propriedade Intelectual	Contributor (Contribuidor / Colaborador)	Inclui uma pessoa, uma organização não especificada em um elemento criados, mas que tenha feito contribuições intelectuais para o recurso.
Utilização	Date (Data)	Será associada a criação ou disponibilização do recurso. A melhor prática recomendada é definida em um perfil da ISO 8601 que inclui (entre outros) datas dos formulários AAAA e AAAA-MM-DD.
Conteúdo	Type (Tipo do recurso)	Descrição de categorias gerais, funções, espécies ou níveis de agregação para o conteúdo, recomenda-se utilizar vocabulário controlado, por exemplo, home page, relatório técnico, ensaio, dicionário, etc.
Utilização	Format (Formato)	Pode incluir o tipo da mídia ou as dimensões do recurso, pode ser usado para determinar o software, hardware ou outro equipamento necessário para mostrar ou operar o recurso.
Utilização	Identifier (Identificador do recurso)	Recomenda-se utilizar o string ou número conforme um sistema de identificação formal de forma a identificar o recurso de forma exclusiva. Exemplo: Identificadores globais únicos, como o Padrão Internacional Números de livros (ISBN).
Conteúdo	Source (Fonte)	O presente recurso pode ser derivado de uma fonte de recurso inteira ou em parte, recomenda-se utilizar o string ou número conforme um sistema de identificação formal.
Utilização	Language (Idioma)	A recomendação para o melhor uso dos valores do elemento língua é definida pela RFC 1766 que inclui um código de língua em 2 letras(do padrão ISO 639), seguido opcionalmente pelo código do país em 2 letras também (do padrão ISO 3166).
Conteúdo	Relation (Relação)	Um identificador de um segundo recurso e sua relação com o recurso atual. Este elemento é usado para expressar vínculos entre recursos relacionados. Os relacionamentos devem ser selecionados de uma lista enumerada.
Conteúdo	Coverage (Abrangência / Cobertura)	As características espaciais ou temporais do conteúdo intelectual do recurso. A cobertura espacial refere-se a uma região física usando nomes de locais ou coordenadas (por exemplo, longitude e latitude). A cobertura temporal refere-se ao que o recurso é em vez de quando foi criado ou disponibilizado.
Propriedade Intelectual	Rights (Direitos)	Conterá uma declaração de gerenciamento de direitos para o recurso. Informações de Direitos frequentemente abrangem Direito de Propriedade Intelectual (Intellectual Property Rights - IPR), Copyright, e várias propriedades de Direitos.

Fonte: Adaptado de (DUBLIN... , 2018)

Os elementos descritos na Tabela 2 compõe o metadados que identifica o dado que está sendo compartilhado no banco de dados federado, auxiliando dessa forma quem deseje utilizar o dado que foi disponibilizado. O completo preenchimento dos elementos dos metadados proporciona a ampliação de seu uso. Como a geração do metadados Dublin Core é de responsabilidade do gerador do dado, é pequeno o risco dos utilizadores obterem informações incorretas ou privadas devido ao mapeamento descuidado de dados do gerador do metadados para o esquema Dublin Core.

4 Aprendizagem aplicada a integração de dados

O desenvolvimento de um sistema comumente é realizado para a resolução de problema ou processo específico e as regras sistêmicas que são implementadas no software refletem as regras do processo, problema ou negócio em questão, assim como a estrutura de dados desenvolvida para armazenar esses dados também refletem as mesmas regras. Entretanto, com o passar do tempo os processos que foram base das regras utilizadas no desenvolvimento do sistema mudam, gerando a necessidade de diversas intervenções e manutenções para adaptação dos sistemas, demandando assim também, alterações nas estruturas de dados mantidos por esses sistemas.

Essas mudanças em sua grande maioria não poderiam ser previstas pelos profissionais que desenvolveram os sistemas, porque em geral não é possível desenhar uma solução para um problema que muda ou evolui. Este é o caso de diversas organizações e entidades, que ampliam sua atuação, mudam de área, otimização processos e realizam fusões com outras organizações, entre outras mudanças possíveis. (RUSSEL; NORVIG, 2004) Sob essa perspectiva, uma abordagem interessante seria que os sistemas fossem capazes de aprender sua estrutura de armazenamento pelos seus dados e pudesse evoluir automaticamente as regras que foram manualmente codificadas pelo desenvolvedor. Seria interessante que o sistema aprendesse as regras funcionais através dos dados que são inseridos em seu modelo de representação e as utilizasse.

As abordagens de integração de dados são baseadas em regras que são codificadas manualmente pelos desenvolvedores nos sistemas de integração de dados. Dessa forma, podemos considerar que uma integração de dados é um agente sistêmico, pois realiza ações baseada em regras com resultados esperados que foram programados por um agente humano, ou seja, necessita de uma intervenção humana para evoluir. Para possibilitar uma evolução automática de uma solução de integração, uma possibilidade seria a utilização de algoritmos de aprendizagem para evolução das regras de integração. Existem diversos algoritmos de aprendizagem com objetivos específicos conforme será apresentado na sessão 4.1.

4.1 Modelos de Aprendizagem

A aprendizagem na ótica computacional consiste em buscar dentro de um espaço de hipóteses possíveis por aquela que será adequada, ou que tenha bom desempenho para a questão avaliada. Para isso é necessário um conjunto de dados de treinamento para que o algoritmo possa selecionar a opção adequada, e com isso aprender a selecioná-la em outros conjuntos além do conjunto de treinamento. (RUSSEL; NORVIG, 2004)

Segundo (RUSSEL; NORVIG, 2004), para o desenvolvimento dessa abordagem inteligente é importante realizar alguns passos de avaliação, entendimento e definição da solução:

1. **Que componente deve ser melhorado** - É preciso identificar dentre os componentes sistêmicos qual deverá evoluir com a utilização dos dados.
2. **O conhecimento prévio que o agente já tem** - Deve-se mapear todas as informações sobre o componente em questão: os estados possíveis, as propriedades, estados desejáveis, metas, caminhos e utilizações.
3. **Que representação é usada para os dados do componente** - Identificar as entradas, os atributos, e as saídas.
4. **Que tipo de aprendizagem está disponível** - Identificar o tipo de aprendizagem mais indicado para a temática em questão:
 - **Aprendizagem supervisionada** Na aprendizagem supervisionada o agente é treinado com alguns exemplos de dados de entrada e saída e aprende uma função que faz o mapeamento da entrada para a saída baseado no treinamento realizado pelo instrutor.
 - **Aprendizagem não supervisionada** O agente aprende com os padrões dos dados na entrada, embora não seja fornecido nenhuma indicação explícita sobre o significado do dado. Essa técnica permite encontrar padrões existentes nos dados, pelas similaridades e diferenças.
 - **Aprendizagem semi-supervisionada** Na aprendizagem semi-supervisionada os dados de treinamento são mínimos, e apenas alguns exemplos são rotulados em um grande universo de dados. O algoritmo fará o melhor aprendizado possível com coleção de exemplos não rotulados.

- **Aprendizagem por reforço** Diferentemente de outros modelos de aprendizagem, na aprendizagem por reforço não existem dados de entrada/saída para ser utilizado no aprendizado. Este modelo aprende a partir de uma série de reforços, recompensas ou punições e ele precisa obter experiência dos possíveis estados, ações, transições e recompensas do sistema para que seja ótimo.

4.1.1 Redes Neurais Artificiais

Rede neural é um modelo de aprendizagem supervisionada, ou seja, necessita de exemplos de entrada com resultados esperados na saída para realização do treinamento e evolução dos resultados. O modelo de rede neural artificial é uma abstração realizada de forma comparativa ao neurônio humano, ou seja, um neurônio computacional é uma unidade que se conecta a outro por ligações direcionadas assim como o neurônio humano. Essas ligações podem ser ativadas, e recebem um peso que determina a força e o sinal da conexão. (JAIN; MAO; MOHIUDDIN, 1996)

As propriedades gerais da rede são definidas pelas propriedades dos neurônios, como é exemplificado na figura 3.

Figura 3 – Exemplo de neurônio computacional

Fonte: Adaptado de (RUSSEL; NORVIG, 2004)

O dado é recebido pelas ligações de entrada do neurônio, com um peso correspondente. A função de ativação é aplicada e permite gerar mudanças nos pesos de forma a produzir variações nos resultados de saída e assim buscar a convergência para o resultado esperado de aprendizagem. Existem dois tipos de conexões para as redes neurais: as redes neurais com **alimentação para frente**, ou seja, a rede tem conexões em uma só direção. Nessa forma de conexão à rede só possui o estado atual e os pesos. Em um outro tipo de conexão, conhecida como **rede recorrente**, a rede alimenta sua entrada com as saídas de

execuções anteriores, ou seja, a rede possui os dados da entrada que é o estado atual, os pesos e informações de iterações anteriores. Em ambos os tipos, a cada iteração é realizada um processo de retro propagação e ajustes nos erros da rede neural. A retro propagação e ajuste dos erros da camada de saída para as camadas anteriores permitem a evolução do algoritmo no decorrer do treinamento. (RUSSEL; NORVIG, 2004)

4.1.2 Agrupamentos ou *Clustering*

Aprendizagem por agrupamento ou por *clustering* é uma técnica de aprendizado não-supervisionado, ou seja, quando não há uma classe associada a cada exemplo do universo de dados. Agrupar os dados é selecionar uma coleção inteira ou um agrupamento de hipóteses a partir do espaço de hipóteses, ou seja, essa técnica agrupa os dados pelas características similares dos mesmos. Para o desenvolvimento dessa técnica é importante uma clara definição dos objetos e atributos representados nos conjuntos de dados. Um modo de pensar na ideia de conjunto é considerá-lo genérico e ampliar o espaço de hipóteses. (RUSSEL; NORVIG, 2004)

Existem diversas formas de representação visual para os agrupamentos. Uma dessas formas, é o dendograma. O dendograma é um diagrama de árvore que exhibe os grupos formados por agrupamento de observações em cada passo e em seus níveis de similaridade, especialmente mostrando relações taxonômicas. (OXFORD... , 2018) O dendograma implementado neste trabalho utiliza a distância Euclidiana para apresentar as distâncias entre os elementos do problema em questão.

4.2 Integração de dados com o uso de técnicas de aprendizagem

Partindo do princípio que um agente computacional pode ser melhorado pela aprendizagem a partir de seus dados, a complexidade da integração automática de dados estruturados e não estruturados pode ser minimizada com o uso de técnicas de aprendizagem. A literatura apresenta o uso de abordagens de aprendizado direcionadas para integrar os dados, como:

- Em Li e Clifton (1994) é proposta a utilização de rede neural para identificação dos campos de matching de bases federadas, utilizando para isso informações do

metadados e principalmente informações geradas pelos padrões dos dados inferidos pela rede neural.

- Já em Harikumar, Reethima e Kaimal (2014) é proposta a criação de um esquema global de dados pela proposição da integração dos atributos mais relevantes de um domínio de conhecimento. Para encontrar os atributos relevantes foi utilizado múltiplas regressões lineares e decomposição de valores singulares (SVD).
- No artigo de Zhang et al. (2016), é apresentada a proposta de desenvolver um "Cross Task Labelling", que realiza um enriquecimento de rótulos inexistentes ou parcialmente existentes em um banco de dados com busca em um conjunto multidimensional de rótulos sem a intervenção humana com uso algoritmos de *Support Vector Machine*(SVM).

De acordo com os artigos apresentados, algoritmos de aprendizagem podem ser utilizados para identificar atributos de integração e, por uma avaliação do contexto dos dados envolvidos na integração, verificar afinidade entre os dados sem a necessidade de intervenção humana. Dentre os diversos algoritmos inteligentes existentes, o algoritmo inteligente de rede neural tem emergido como uma técnica que pode ser utilizada em uma série de aplicações segundo Widrow, Rumelhart e Lehr (1994), inclusive para o tema discutido nesse trabalho.

5 Trabalhos relacionados

Para a avaliar do estado da arte relacionado à integração de banco de dados representadas em modelos heterogêneos e abordagens para automação foi conduzida uma revisão sistemática sobre esse tema.

5.1 Arquiteturas de integração

Algumas abordagens comparativas sobre integração de dados têm sido desenvolvidas em artigos encontrados na literatura. Antes mesmo da popularização de dados não estruturados, foi realizada uma discussão em Ziegler e Dittrich (2004) sobre a integração semântica dos dados estruturados e o seu contexto aplicado que pode gerar uma ambiguidade de interpretação, demonstrando a complexidade na integração de dados homogêneos. Essa ambiguidade existente gera a necessidade do mapeamento humano para interpretar o dado para posteriormente poder usá-lo no processo de integração. Na figura 4 é apresentada uma associação dos tipos de integração nos diferentes níveis da arquitetura. Este é um ponto discutido neste trabalho, que contribui com o histórico de processos de integração de dados estruturados e suas arquiteturas de integração nas camadas de uma aplicação. Entretanto, não tratar dados do tipo não estruturado é uma limitação encontrada no artigo. Porém os desafios relatados para a integração mesmo de dados homogêneos, também são encontrados e frequentemente ampliados no contexto de dados representados em modelos heterogêneos.

Figura 4 – Abordagens de integração em diferentes níveis arquiteturais

Fonte: Adaptado de (ZIEGLER; DITTRICH, 2004)

Já o trabalho de Rahm (2016), sugere que o processo de integração não seja realizado apenas pela integração de origens de dados pelo emparelhamento dos campos chaves dos registros, mas que também suporte à integração de metadados. Esse trabalho de Rahm (2016) esboça diferentes casos de uso e a discussão percorre abordagens comparativamente. A tabela 3, apresentada como uma comparação do nível de integração das abordagens discutidas, destacando a abordagem *Integrated Ontology* com o mais baixo grau de integração e *Entity Search Engines e Comparison Portal* com os mais altos graus de automação no processo de integração, dentre os tipos discutidos. Este trabalho de Rahm (2016) apresenta uma classificação, que aborda a perspectiva de relacionar as abordagens pelo use case, não apresentando as abordagens como uma visão de arquitetura técnica de integração.

Tabela 3 – Casos de uso para integração de dados

Use Case	Data Integration 1	Data Integration 2	#Sources	Degree of automated data integration
(1) Meta-Search	Virtual	Metadata	Low-Medium	Medium
(2) Open Data	Physical Collection	Primarily Metadata	Very High	High, but limited integration
(3) Integrated Ontology	Physical	Metadata	Low-Medium	Low-Medium
(4) Knowledge	Physical	Data + Metadata	Low-High	Medium-High
(5) Entity search engines	Physical	Data (+metadata)	Very High	High
(6) Comparison Portal	Physical/Híbrid	Data + Metadata	High	High

Fonte: Adaptado de (RAHM, 2016)

5.2 Trabalhos correlatos à revisão sistemática

A literatura apresenta algumas revisões, porém essas não trazem a perspectiva de integração de dados, e sim uma perspectiva mais ampla do contexto *Big Data*. O trabalho em Chen, Mao e Liu (2014) realizou uma revisão sistemática para identificar as tecnologias, processo de integração de dados e aplicações de negócio. Entretanto não abordou as arquiteturas de integração de dados heterogêneos. A revisão na literatura realizada em Caldarola e Rinaldi (2015) apresentou um levantamento das tecnologias mais difundidas sob o arcabouço de *Big Data*, e como contribuição trouxe um *framework* para

avaliação qualitativa das soluções de *Big Data*, sem foco específico em integração. Já em Siddiqi, Karim e Gani (2017) realizou-se uma pesquisa sobre perspectiva de arquiteturas armazenamento de dados com tecnologias *Big Data*. Como contribuição, o artigo apresenta uma taxonomia de tecnologias de armazenamento de *Big Data* para auxiliar analistas de dados e pesquisadores a entender e selecionar um mecanismo de armazenamento que atenda às suas necessidades, porém sem foco em integração de dados.

5.3 *Análise da Revisão Sistemática*

A variedade de bancos de dados, modelos de representação e linguagens de manipulação de dados são um desafio para os especialistas de computação na escolha de uma arquitetura de solução para o desenvolvimento de uma integração de dados, que somado à falta de uma definição de categorias que permitam classificar as técnicas existentes para então compará-las, geram atividades adicionais para que os especialistas definam a solução de integração mais adequada ao problema.

Como contribuição, esta revisão sistemática propôs cinco categorias para classificação de técnicas de integração de modelo de representação de dados heterogêneos para apoiar as atividades de integração. Na tabela na Tabela 4 são apresentadas as principais vantagens e desvantagens de cada categoria. Também foi avaliado o grau de reuso de cada categoria.

Tabela 4 – Arquiteturas de integração: vantagens e desvantagens

Arquitetura	Reuso	Vantagens	Desvantagens
Customizada	Baixo	- Atendimento a demandas específicas e problemas complexos de negócio.	- Arquiteturas construídas de forma personalizada para solucionar um problema. - Dependentes de um cenário específico de uma organização, entidade ou instituição. - Necessitam de equipe especializada, com conhecimento técnico e do negócio representado nos dados.
Componentes geográficos	Baixo	- Atendimento a demandas específicas e problemas complexos de negócio. - Utilizam a referência geográfica como chave de integração e enriquecimento dos dados.	- Arquiteturas construídas de forma personalizada para solucionar apenas um problema. - Dependentes de um cenário específico de uma organização, entidade ou instituição.
<i>MapReduce</i>	Baixo	- Atendimento a demandas específicas e problemas complexos e assíncronos, com alto volume de dados.	- Arquiteturas construídas de forma personalizada para solucionar um problema. - Não atendimento de problemas síncronos, por demandar processamento <i>batch</i>
<i>Middleware</i> ou serviços de dados	Médio	- Infraestrutura de <i>software</i> facilita a interação entre os módulos de software distribuído. - Abstração usada em ambiente heterogêneo que requer comunicação homogênea dos dados.	- Proposição de um ou mais componentes customizáveis responsáveis pela realização de tarefas intermediárias na integração. - Dependentes de um cenário específico de uma organização, entidade ou instituição.
Federação ou virtualização de dados	Alto	- Abstração entre o modelo de representação origem e o modelo de representação destino da integração. - Uso de metadados que descrevem os dados armazenados nos bancos de dados federados, necessitando baixa customização.	- Necessidade de conhecimento sobre os metadados dos bancos envolvidos na federação.

Fonte: Keylla Ramos Saes, 2018

Os artigos avaliados foram analisados pela similaridade entre as técnicas neles apresentadas. Dessa forma, cada artigo pode ter recebido mais de uma categoria, porém foi aplicado um critério de priorização, descrito na tabela 5, para que fosse classificado, ao final, em somente uma categoria. Foi adotado o critério de prevalectimento da categoria com menor valor na coluna prioridade.

Tabela 5 – Grau de priorização das arquiteturas

Arquitetura	Grau de priorização
Arquitetura customizada	1
Arquitetura de componentes geográficos	2
Arquitetura <i>MapReduce</i>	3
Arquitetura de <i>Middleware</i> ou serviços de dados	4
Arquitetura de federação ou virtualização de dados	5

Fonte: Keylla Ramos Saes, 2018

A tabela 5 apresenta de forma ordenada primeiro as abordagens especializadas à resolução de problemas (exemplo: arquitetura customizada) e posteriormente as abordagens mais generalistas (exemplo: arquitetura de federação ou virtualização de dados). A distribuição de artigo por categoria é apresentada na Figura 5.

Figura 5 – Distribuição dos artigos nas categorias de arquiteturas de integração

Fonte: Keylla Ramos Saes, 2018

Uma visão geral do sincronismo é apresentada na figura 6, na qual verifica-se que 80% das soluções encontradas são síncronas e apenas 20% assíncronas e apresenta a distribuição desse sincronismo entre as categorias. A categoria MapReduce apresenta alta incidência de abordagens assíncronas, enquanto as categorias middleware ou serviços de dados e federação ou virtualização de dados apresentam alta incidência de abordagens síncronas. A análise dos dados de sincronismo de cada um dos artigos apresenta uma tendência ao desenvolvimento de arquiteturas e soluções de integração de dados síncronos nos últimos anos. Essa tendência é apoiada pela necessidade dos usuários de sistemas ao acesso a informação em tempo real, para realização de análises de dados disponibilizados de forma integrada.

Conforme descrito na Figura 7 é possível observar que no ano de 2013 houve um aumento no número de publicações de artigos em todas as arquiteturas. Esse aumento pode

Figura 6 – Distribuição de artigos nas categorias de arquiteturas de acordo com sincronismo do dado

Fonte: Keylla Ramos Saes, 2018

ter sido ocasionado pela popularização das soluções Big Data e aos diversos movimentos ocorridos nesse ano no meio acadêmico e empresarial para divulgar o tema *Big Data*.

A distribuição do gráfico da Figura 7 mostra que, nos últimos anos, a arquitetura Middleware de dados ou serviços de dados tem sido a mais citada nos artigos encontrados na literatura, seguida da arquitetura de Federação ou Virtualização de Dados, sendo essas as arquiteturas que apresentam menor nível de customização comparativamente as demais arquiteturas encontradas na literatura.

Figura 7 – Distribuição temporal dos artigos por arquitetura de integração

Fonte: Keylla Ramos Saes, 2018

O gráfico da Figura 8, apresenta a distribuição dos modelos de representação nos artigos avaliados, sendo que a maior incidência foram os tipos Multi Modelos (que apresentam mais de um modelo ao mesmo tempo, sendo eles: documento ou grafo ou

chave e valor), com 40% de ocorrências, seguido do modelo de documento com 31%, o modelo de grafo com 23% e o modelo geográfico com 6%. Essa distribuição mostra que a integração com o modelo relacional permitirá explorar as funções do conteúdo do dado, a parte textual e descritiva pelos modelos de documento e chave e valor, assim como também permite explorar a integração das relações dos dados no modelo relacional com o uso dos modelos de grafo e informações de dados de localização com o modelo de representação geográfica de forma integrada, ou seja, a integração de modelos heterogêneos permitirá o desenvolvimento de diversas aplicações com o resultado dos dados integrados.

Figura 8 – Distribuição dos modelos de representação de dados não estruturados nos artigos avaliados

Fonte: Keylla Ramos Saes, 2018

5.4 Classificação dos artigos nas arquiteturas de integração

Nesta sessão são apresentadas as principais características, contribuições e limitações dos trabalhos selecionados, conforme as categorias definidas.

Arquitetura Customizada

Na classificação de arquitetura customizada se encontram os artigos cujas integrações de dados são realizadas de forma personalizada para solucionar cada tipo de integração necessária. As abordagens apresentadas nos artigos selecionados abrangem dados oriundos de sistemas como: sistemas web, sistemas transacionais e sistemas analíticos (BI). Tais soluções foram realizadas com a criação de componentes específicos para lidar com os desafios dos contextos para quais elas foram criadas, como por exemplo: estrutura dos bancos de dados, fontes e atualização de dados, e desempenho.

O primeiro artigo a ser descrito é o Rui et al. (2013) que apresenta uma classificação dos domínios dos dados da Web, de acordo com os formatos existentes: estruturados, semiestruturados e não estruturados, conforme pode ser visto na Figura 9. Esse artigo apresenta como contribuição uma proposta para padronização de dados da Web de forma que estes possam ser integrados em outras aplicações. Os dados passam por um processo inicial de classificação, sendo separados em três domínios: estruturados, semiestruturados e não estruturados com um tratamento diferenciado para cada tipo de domínio.

Figura 9 – Estrutura de organização de padrão de dados de domínio da Web

Fonte: Adaptado de (RUI et al., 2013)

No artigo Benker (2013) é descrito uma arquitetura de integração entre dados OLTP (*Online Transaction Processing*) e OLAP (*Online Analytical Processing*) para automatizar o processo de aquisição de passagens. Uma das contribuições deste trabalho é a proposta de arquitetura complementar para lidar de forma automatizada com alguns tipos de erros gerados durante a carga de dados.

O trabalho apresentado em Ndume et al. (2013) traz uma arquitetura de integração com uma visão transacional pode ser verificada pela abordagem de coleta e integração de dados heterogêneos de saúde. O artigo também explora a metodologia de coleta de dados de Autopsia Verbal (VA) usando dispositivos móveis contribuindo com uma abordagem para integração desses dados heterogêneos.

A proposta em Doshi et al. (2013) visa atender a necessidade de integração de variedade de dados com grande volume e com a manutenção das propriedades ACID dos bancos relacionais, como aplicações de e-commerce e BI. Como contribuição esta

abordagem apresentada diminui o risco de interromper as atividades transacionais devido à manutenção das propriedades ACID entre os dois bancos de dados.

Já em Grund et al. (2013) a arquitetura que propõe flexibilidade e desempenho ao combinar os diferentes tipos de dados em um único mecanismo de armazenamento de dados e processamento em memória que permite realizar consultas e trabalhar com vários tipos: em linhas, colunas, partições híbridas e também gráficos nativamente. Uma proposta similar é apresentada em Takahashi et al. (2013), entretanto nesse caso a discussão baseia-se no tipo do dado (valores numéricos, textos, JSON, XML, etc.), ao invés do modelo de representação do dado.

Em (ZHAO et al., 2014) foi apresentado uma proposta de migração contínua de esquema de um banco MySQL (SQL) para um banco MongoDB (NoSQL). Para solucionar o problema da impossibilidade da execução de junções de tabelas no MongoDB um algoritmo de busca em grafos foi desenvolvido para representar as seleções de dados. Um processo de validação dos dados nos dois bancos foi desenvolvido para garantir a efetividade do processo.

Arquitetura MapReduce

Os artigos que apresentaram uma arquitetura de integração com o uso de processamento dos dados em duas etapas, sendo a primeira composta por uma busca de palavras chaves, seguida de um segundo processo de redução dos dados foram classificados nesta categoria MapReduce. Os artigos selecionados apresentaram discussões amplas sobre arquiteturas de integração de dados heterogêneos, como temas sobre chaves de integração e unificação de dados, porém aplicados a solução de um problema específico.

O artigo Malhotra et al. (2015) apresenta uma abordagem para que a escalabilidade de uma arquitetura de dados seja realizada com a utilização de nuvem e com a criação de uma arquitetura de integração com utilização da técnica *MapReduce*, explorando a possibilidade de manter dados sensíveis em uma banco de dados local e dados não sensíveis na nuvem de maneira a escalar o sistema, como demonstrado na Figura 10. A utilização da técnica de MapReduce também foi encontrada em Qin et al. (2012) com a estratégia de criação de uma plataforma de *Big Data Analytics*, abrangendo discussões sobre integração

de dados estruturados e não estruturados para posterior aplicação de técnicas avançadas analíticas sobre os dados heterogêneos.

Figura 10 – Processo de integração entre dados não sensíveis disponibilizados na nuvem e dados sensíveis disponibilizados localmente

Fonte: Adaptado de (MALHOTRA et al., 2015)

O artigo Lee et al. (2015b) cita um problema conhecido em integração de dados é a complexidade para definição de chaves para junção dos dados. É proposto uma abordagem para encontrar as relações entre bancos de dados com a técnica de *MapReduce*. Já o artigo Ma, Dong e Yang (2014) explora um outro nível mais granular de relação de dados. Este artigo apresenta uma discussão sobre como deduplicar um dado, ou seja, como encontrar ou compor um registro único em um banco de dados com registros duplicados. Como contribuição essa arquitetura traz a capacidade de processamento massivo, a técnica de *Map reduce* é utilizada para otimizar a eficiência para unificar dados, podendo executar 3 processos distintos de unificação em um mesmo conjunto de dados, conforme discutido no artigo.

Arquitetura de Componentes Geográficos

Os artigos que apresentaram uma arquitetura de integração para resolução de problemas geográficos ou de rotas foram classificados nesta categoria. Esses artigos tratavam a integração de dados heterogêneos com característica geográficas, ou seja, continham

tratamento de latitude e longitude, ou utilizavam softwares geográficos como ArcGIS e PostGIS para solucionar os problemas apresentados. As arquiteturas encontradas estavam direcionadas para resolução de problemas específicos.

O artigo em Houssaini e Badri (2012) utiliza uma arquitetura de componentes geográficos para o monitoramento de uma rede rodoviária. Já em Fang et al. (2014) a arquitetura foi desenvolvida para análise de informações climáticas, conforme descrita na Figura 11, e em Ngan (2014) foi detalhada uma arquitetura para análise de rotas utilizada para tomada decisões militares.

Figura 11 – Arquitetura geral com base em IoT

Fonte: Adaptado de (FANG et al., 2014)

Arquitetura *Middleware* de dados ou serviços de dados

A arquitetura de *Middleware* de dados ou serviços de dados possui uma camada de serviços de dados que permite o acesso a diversas fontes de dados por essa uma camada de abstração, independentemente da tecnologia ou localização do banco de dados. Geihs (2001) Os artigos que desenvolveram serviços de dados ou discutiram essa camada de middleware de dados foram classificados nesta categoria. O emprego de virtualização para diferentes tipos de dados, tecnologias e localizações físicas pode solucionar diversos problemas existentes, com a simplificação da integração.

O artigo Wu (2012) apresenta um modelo de integração baseado em *Web Services* distribuídos, contendo uma camada de usuário, uma camada intermediária e uma camada de fontes de dados. Já em Garg e Kaur (2015) é proposta uma arquitetura com mediador, que atua como um barramento de acesso a diversos tipos de serviços de dados, como pode ser visto na Figura 12. Na proposta em Hong e Rong (2014), a tecnologia em nuvem

é citada na abordagem, que utiliza dados na nuvem e dados locais. Um barramento de serviços foi desenvolvido para fornecer integração, compartilhamento e gerenciamento de dados entre o servidor e o cliente nesse sistema de distribuição.

Figura 12 – Abordagem mediador Wrapper

Fonte: Adaptado de (GARG; KAUR, 2015)

O artigo Chen et al. (2015) discute sobre junções de dados estruturados e não estruturados dos diversos serviços em múltiplos passos com uso de um modelo de metadados. Em Dey et al. (2015) a arquitetura com serviços genéricos foi proposta, com acesso aos metadados que descrevem propriedades dos diversos conjuntos de dados espalhados entre bancos heterogêneos, como bancos de dados relacionais, *Datawarehouse*, *NoSQL* ou *NewSQL*. Esses serviços permitem a pesquisa sobre os conjuntos de dados úteis, sejam eles brutos ou derivados. Já em Vathy-Fogarassy e Húgyák (2017) é apresentada uma solução generalizada para consulta em sistemas de banco de dados relacionais e não relacionais ao mesmo tempo com utilização de metamodelo, abrangendo as heterogeneidades estruturais, semânticas e sintáticas dos sistemas de banco de dados com uso de objetos JSON.

O artigo Schreiner, Duarte e Mello (2015) apresenta uma arquitetura que utiliza estratégias de traduções entre modelos, cuja a camada de middleware permite traduzir de esquemas relacionais e comandos SQL para esquemas equivalentes em banco NoSQL orientado a chave (documento, chave-valor e colunar). Já no artigo Oliveira et al. (2016), a abordagem extrapola somente a tradução, apresentando um Middleware de integração de dados hospitalares que funciona como integrador, auditor de normas, tradutor, além de ser uma fonte secundária de dados quando a fonte primária não está disponível.

O artigo Mehmood, Culmone e Mostarda (2017) demonstra que um dos requisitos existentes para integração de dados heterogêneos é a possibilidade de integrá-los em tempo

real, e propõe uma abordagem utilizando leitura de dados de sensores. Porém, o *Data Middleware* ou *Data Services Architecture* é uma camada adicional que pode onerar o tempo de processamento da integração em tempo real, tema também discutido no artigo. Em Giannakouris et al. (2016) é apresentada proposta para otimizar o custo em utilizar os middlewares de multimotores ou multi banco de dados. Já o artigo Chang, Tsai e Wang (2016) traz uma abordagem para otimizar a performance de um *Data Middleware*, propondo um cache de alto desempenho e uma opção para selecionar a melhor ferramenta para processamento dos dados de forma a receber o resultado mais rapidamente.

Em Kolev et al. (2016) é proposta a criação de uma linguagem funcional semelhante ao SQL, porém que permite consultar dados heterogêneos ao mesmo tempo, a CloudMdsQL (*Cloud Multidastore Query Language*). A utilização dessa linguagem é ser feita na interface nativa de cada banco de dados, pelas chamadas de funções. Em (XU et al., 2016) foi desenvolvido um *middleware* de dados com MySQL e Hive com a proposta de selecionar dados de diversas aplicações abstraindo os detalhes técnicos de cada banco de dados, com uma linguagem para integração que abstrai a linguagem do banco de dados origem.

Arquitetura de federação ou virtualização de dados

Federação de dados ou arquitetura de virtualização é uma solução na qual vários bancos de dados funcionam como um único banco. Cada banco de dados que pertence a federação possui uma interligação com o federado ou banco *master*. As consultas são realizadas no banco federado que devolverá as referências dos resultados para os dados para o *master* que as apresenta como se pertencessem a um único sistema virtual. Os artigos encontrados que possuem este tipo de comportamento foram agrupados nessa classificação.

Arquiteturas para unificar operações de banco de dados e seus mecanismos de armazenamento para que desenvolvedores de aplicativos possam interagir com bancos relacionais e não relacionais são apresentadas em Gadepally et al. (2015) e em Sellami, Bhiri e Defude (2016), sendo que este último que propõe dados virtuais que atuam como mediadores e interagem com os bancos de dados integrados pela arquitetura, permitindo a execução de consultas simples e complexas sobre qualquer banco de dados heterogêneo mapeado. Da mesma forma, a arquitetura descrita pela Figura 13, permite que os usuários consultem e juntem dados não relacionais (NoSQL) e com dados relacionais (SQL) em

uma única consulta SQL. Para isso um modelo genérico de arquitetura de virtualização foi desenvolvido em Lawrence (2014b) permitindo aos sistemas NoSQL serem consultados usando o SQL e interagir sem qualquer suporte JDBC.

Figura 13 – Arquitetura de federação de dados Unity

Fonte: Adaptado de (LAWRENCE, 2014b)

Em Abolhassani et al. (2017) é apresentada uma abordagem para enriquecer a entrada de dados com consulta a um grafo *Resource Description Framework* (RDF), melhorando a integração e a consistência dos dados para as ferramentas de desenvolvimento, assim como em Groger, Schwarz e Mitschang (2014) que apresenta o enriquecimento de dados em um data warehouse com conteúdo não estruturado. Esta proposta baseia-se em links entre os elementos do data warehouse e os itens de conteúdo, que são representados em uma estrutura orientada por grafo.

No artigo Pino, Ravidá e Scibilia (2013) uma comparação entre algumas tecnologias que implementam banco de dados federado e uma arquitetura de software customizada foi realizada, por um caso de uso para integração de dados entre as unidades administrativas municipais, apresentando contribuições e limitações de cada tecnologia.

Em Karpathiotakis et al. (2017) é apresentada uma solução que realiza uma virtualização que avalia a complexidade da fonte de dados, realizando o processo em duas fases, primeiro produzindo um plano de acesso e depois otimizando-o. Já no artigo Guerra et al. (2017) é proposto uma solução com banco federado para reunir, consolidar

e disponibilizar informações científicas e de inovação que atuam em algumas regiões da Europa.

A Tabela 9, apresentada do apêndice deste trabalho, destaca outras informações que foram extraídas dos artigos, como o ano de publicação, o sincronismo dos dados e o modelo de representação dos dados.

5.5 *Discussões sobre a revisão sistemática*

A integração de dados ainda é uma questão em aberto, inclusive pesquisas continuam investigando soluções para integração entre dados estruturados ou homogêneos, ou seja, integração cujos dados estão no mesmo modelo representação. Esse cenário se torna ainda mais complexo quando os dados são representados por modelos de dados heterogêneos e se enquadram em um escopo de Big Data, que além da variedade de fontes de dados, também a velocidade e o volume da geração de dados adicionam complexidade o processo de integração. Um desafio é possibilidade de escalar o processo de integração de dados, sejam eles representados em modelos homogêneos ou heterogêneos.

Essa revisão permitiu uma avaliação das diferentes arquiteturas, desde as classificadas como arquiteturas customizadas até os diferentes níveis de automatização, sendo possível verificar que a integração manual ainda ocorre com frequência.

As arquiteturas contidas no grupo arquiteturas customizadas tratam o tema de maneira específica quanto à resolução de um problema, com dependência de especialistas que conheçam o negócio, não abordando componentes de forma generalista. Esses artigos foram considerados neste trabalho, pois embora as soluções sejam específicas, as técnicas aplicadas podem ser ampliadas, automatizadas e generalizadas em soluções futuras.

Os artigos classificados como arquiteturas *MapReduce* sugerem um pré-processamento dos dados. Ou seja, são arquiteturas assíncronas, porém possuem alta capacidade para tratamento e transformações eventualmente necessárias para o uso do dado, e pode ser uma estratégia utilizada para integrar dados se o requisito de integração não necessite de respostas em tempo real.

A arquitetura de componentes geográficos tem evoluído na resolução de problemas espaciais e informacionais e embora algumas soluções utilizem componentes genéricos como ARCGis e o PostGis, que utilizam banco de dados com representação geográfica do dado,

estas arquiteturas ainda apresentaram um alto nível de customização resolvendo problemas espaciais específicos, sendo pouco generalista. A classificação foi mantida pois acredita-se que as mesmas possam ser automatizadas e generalizadas, assim como as arquiteturas customizadas. Em contrapartida, a arquitetura de middleware ou serviços de dados cria uma camada que abstrai a origem do dado, oferecendo algum tipo de padronização dos dados pelos serviços. Entretanto para definir essa padronização é necessário um trabalho de customização que, embora em menor escala que outras arquiteturas, ainda persiste.

Já a arquitetura de federação ou virtualização de dados exige o menor nível de customização dentre as demais arquiteturas discutidas neste trabalho realizando um apontamento para a origem do dado, porém ainda necessita de uma camada de mapeamento ou modelo canônico. Portanto, pode-se dizer que essa arquitetura que implementa um sistema de gerenciamento de banco de dados federado representa uma interface simplificada entre a camada de aplicação e diferentes bancos de dados que podem manter suas estruturas originais para uma visão integrada do dado (PINO; RAVIDÁ; SCIBILIA, 2013).

Como apresentado neste artigo, 80% das soluções encontradas são síncronas e apenas 20% assíncronas, demonstrando assim uma tendência de soluções que atendam requerimentos de resposta em tempo real. O sincronismo é um aspecto a ser discutido durante a seleção da abordagem de integração para solução de um problema, principalmente quando este problema envolver o tratamento de grandes volumes de dados como no caso de soluções *Big Data*.

Observou-se que os artigos que contém a palavra-chave *Big Data* apresentaram arquiteturas mais genéricas e escaláveis, o que é interessante pela perspectiva que podem ser aplicadas a diversos problemas, porém perdem especificidades e otimizações que foram encontradas em arquiteturas que foram desenvolvidas para resolver um problema específico.

O processo de mapeamento de dados de forma automatizada é complexo, pois em alguns casos as exceções geradas por uma regra de mapeamento só são solucionadas por interpretação humana, o que dificulta a escala dos processos de integração e é citado como tendência para novos estudos. Além disso, em situações cuja a integração é realizada de forma automática, o controle dos resultados é muito importante, como a performance e tempo de execução da integração, dado que os volumes são altos em soluções *Big Data*. Diversos artigos tratam do tema como uma tendência de estudo para evolução da área.

Os artigos que apresentaram soluções para um escopo específico de integração sugerem aumentar o escopo de abrangência de dados, com a inclusão de mais entidades,

banco de dados ou assuntos de negócio, estratégia que visa aumentar a probabilidade da descoberta de conhecimento. Essa possibilidade tem motivado a crescente busca pela integração de dados nas organizações e instituições. Uma vez alcançada, a integração automática de dados permitirá que os especialistas se dediquem a tarefa de análise de dados já integrados, corroborando com a expansão do conhecimento, ao invés da árdua tarefa de integrá-los.

6 Abordagem proposta para o Integrador de dados

Segundo Zhao (2007), uma solução para o problema de análise de dados distribuídos em diversos bancos de dados é a combinação de diferentes técnicas de integração que ofereçam aplicações para solução em larga escala. Essa abordagem tem sido utilizada em aplicações apresentadas neste trabalho, como é o caso da ferramenta SEMINT de Li e Clifton (2000), que propôs uma abordagem de integração voltada para dados estruturados e com o uso de algoritmos inteligentes. Nossa abordagem baseia-se na composição de técnicas de integração para diferentes modelos e tipos de dados para apoiar a integração de dados em cenários heterogêneos. Para tanto, é apresentada uma arquitetura para integração flexível e expansível para atender e apoiar o processamento das demandas de integração de diferentes modelos de representação de dados. A arquitetura busca encontrar a similaridade entre os atributos de diferentes conjuntos de dados, indicando quais poderiam ser integrados.

6.1 *Integração automática de dados estruturados e não estruturados*

A abordagem proposta visa viabilizar a integração automática de dados. Para tanto, os dados são avaliados com o uso de seus metadados para verificar qual a similaridade entre eles e qual o nível de integração possível por meio de uma avaliação prévia da heterogeneidade dos dados e dos metadados disponíveis. O Integrador de Dados possui componentes para avaliação do contexto de integração dos dados e utiliza os recursos oferecidos pela arquitetura para que o processo ocorra de forma automática.

Para exemplificar a abordagem proposta, um diagrama de execução da integração de dados é apresentado na Figura 14. Nesse diagrama são apresentados três elementos que representam dados a serem avaliados, são eles: uma cozinheira, uma laranja e uma estudante de artes. Qual a possível relação entre eles? O processo proposto não conhece a priori essa resposta. Entretanto, durante o processo de avaliação dos dados em suas representações, foi identificada a palavra "fruta" em cada uma delas. Essa palavra não foi descrita previamente, ela surge da análise de similaridade entre seus atributos, realizada tanto na estrutura da representação ou em um metadados. Como resultado dessa análise, surgem os possíveis atributos de integração entre os dados, com o seu potencial de integração associado: alto, médio ou baixo. Na Figura 14, as linhas entre cada um dos dados mostra o potencial de

integração entre eles e as chaves de integração relacionadas. No exemplo, a comparação entre o metadados da cozinheira *faço bolo de fruta* teve índice alto de integração com o metadados *fruta* do dado *laranja*, gerando dessa forma uma sugestão de integração *pode resultar em bolo*. Já a comparação entre o metadados da cozinheira, o *faço suco de fruta* com o metadados *faço sabonete de fruta* do dado *estudante de artes* gerou um índice baixo de integração, não produzindo dessa forma nenhuma sugestão de integração.

Figura 14 – Exemplo conceitual de integração

Fonte: Keylla Ramos Saes, 2018

O exemplo da Figura 14 é representado na Figura 15 com o uso de uma visão baseada em atributos. Na Figura 15 são apresentados arquivos intermediários como uma abordagem de análise de similaridade entre atributos, no exemplo é utilizado o termo *fruta* como um atributo.

Figura 15 – Exemplo detalhado de integração de dados heterogêneos

Fonte: Keylla Ramos Saes, 2018

6.2 Arquitetura do Integrador de dados

Devido à diversidade de fontes dos dados, com modelos de representação heterogêneos, a solução proposta proporciona recursos para a integração e uso de módulos para atender demandas de modelos e tipos de dados distintos, o que oferece uma abordagem flexível para lidar com a diversidade de dados e seus modelos. Dessa forma, esse trabalho propõe uma arquitetura de integração modular, com componentes alternativos para atender à variação e evolução dos requisitos de integração e dos dados. Como forma de funcionamento, cada componente trata seus modelos de entrada de dados e gera saídas de dados de acordo com o modelo de representação previamente definido, neste trabalho referenciado como *Metadados Integrador*. Como cada componente executa uma ação específica, eles possuem ações independentes e não acopladas. Dessa forma, podem ser combinados de acordo com a necessidade de integração. Essa estratégia modular permite que os componentes sejam desenvolvidos individualmente e que possam ser trocados, alterados ou melhorados, sempre que necessário. Para tanto, basta atender as definições de entrada e saída de dados, fornecida pelo *Metadados Integrador*. O sequenciamento dos

7. **Histórico de integração:** Armazena o histórico das integrações realizadas para futuras consultas;
8. **Estatísticas da integração:** gera indicadores sobre o processamento da integração, como quantidade de linhas processadas, tempo de processamento entre outras;
9. **Arquivo I:** arquivo formado pelo resultado do processo de integração;
10. **Arquivo M:** arquivo com o metadados do arquivo gerado pelo processo de integração;
11. **Retorno da integração:** recebe a avaliação do usuário sobre a qualidade do processo automático de integração realizado e armazena essa informação no histórico inteligência da integração.
12. **Armazenamento do retorno:** envia as estatísticas do processamento com a respectiva avaliação do usuário sobre a qualidade da integração realizada para o histórico de integração, podendo ser utilizado para qualificar futuras integrações;

A abordagem proposta apresenta o diferencial de ser projetada para facilitar a sua ampliação, uma vez que permite a inclusão de módulos de análise de outros modelos de representação de dados, desde que sejam alinhados com o modelo de entrada e saída dos metadados. Essa característica oferece flexibilidade no atendimento aos novos modelos de representação que podem surgir a qualquer momento em um contexto Big Data, permitindo assim a expansão da arquitetura.

6.2.1 Fluxo dos dados no processo de integração

O fluxo de funcionamento dos módulos é apresentado na Figura 16. Inicialmente, no canto superior esquerdo da figura, a seta marcada com o número (1) indica o registro de informações de parâmetros no módulo de *Gestão de parâmetros* que é responsável por provê-los aos diversos componentes da arquitetura. Logo abaixo a seta (2) são indicados os *arquivos* a serem integrados, e opcionalmente, o passo (3) indica a interface para que o usuário *informe a Semântica de integração* por ele conhecida, se desejar. As informações entram no módulo (4), *Avaliador de representação* para serem avaliadas quanto ao modelo de representação dos dados e extração de metadados e, se necessário, são convertidas e normalizadas para facilitar o processamento interno dos dados. Os arquivos normalizados são enviados para o componente de *Inteligência de integração*(5), que avalia se esses dados já foram integrados anteriormente, caso positivo busca essa integração no histórico,

caso negativo calcula o potencial de integração. As sugestões de integração são enviadas ao componente *Executor da integração*. O Executor da integração executa a junção dos dados de acordo com as sugestões recebidas (6), e envia o histórico da integração (7) para o componente de inteligência de integração para armazená-lo. Os dados sobre a integração realizada são enviados para a *Estatística de integração*(8). O arquivo de dados integrados (9) é gerado, assim como o arquivo de metadados integrados (10) pelo Executor da integração. O parecer sobre a qualidade da integração é alimentado opcionalmente pelo usuário no módulo *Retorno da integração*(11), por meio de informações geradas nos passos (7),(8),(9) e (10), e o retorno é enviado ao módulo Inteligência da Integração(12), como classificador qualitativo da integração realizada, sob a perspectiva do usuário.

6.3 Módulos da arquitetura do Integrador de dados

Os módulos da abordagem proposta que estão apresentados na Figura 16 são detalhados nessa sessão.

6.3.1 Gestão de parâmetros

Para que a arquitetura possa ser dinâmica e ajustável ao ambiente e aos objetivos do usuário, os filtros e limites são definidos pelo usuário no módulo de Gestão de parâmetros. Esse módulo permite a criação e manutenção dos parâmetros da arquitetura de integração, como por exemplo:

1. Diretório de entrada de dados a serem integrados;
2. Limite mínimo para o índice de similaridade;
3. Diretório de saída de dados integrados;
4. Formato de saída;

O componente de gestão de parâmetros possui também a funcionalidade de associar cada extrator de dados ao dado do *Metadado Integrador* (utilizado durante o processo de integração), permitindo assim um rastreamento dos componentes que geram cada informação do metadados. Além disso, o módulo de gestão de parâmetros mantém uma lista de extensões de arquivos que são aceitos para integração, juntamente com os programas que executam aquela extensão. Essa informação pode ser usada posteriormente como parte

dos atributos de integração do dado. Também possui uma lista dos tipos de dados de cada banco de dados e sua normalização para os tipos usados no Integrador de dados, que são: numérico, alfanumérico e data. Outra funcionalidade deste módulo é a manutenção da associação entre o padrão Dublin Core e o *Metadados Integrador*. Os parâmetros são armazenados em banco de dados, em uma instância de metadados do Integrador de Dados e estão disponíveis para o sistema durante o processo de integração.

6.3.2 Arquivos de entrada: Acesso aos dados e metadados

O acesso aos dados é fundamental para o processo de integração. A arquitetura proposta permite o acesso a dados estruturado ou não estruturados, acompanhados ou não por seus metadados. Para tanto, os dados podem ser obtidos de arquivos ou endereço para acesso ou mesmo conexões para bases de dados. Após sua obtenção, os arquivos de dados e metadados são processados de acordo com o formato de representação. O respectivo formato é identificado e processado de acordo com suas regras de formação, como as utilizadas em arquivos CSV, TXT, XML ou JSON. Formatos proprietário podem ser incluídos com o uso de módulos específicos.

6.3.3 Semântica conhecida pelo usuário

Esse módulo permite indicar campos que possam ser utilizados na integração como campos similares, ou chaves de integração entre dois dados ou arquivos. Para isso serão apresentados ao usuário o metadados integrador de cada atributo para que seja selecionado os atributos relacionados. Essa informação de similaridade é submetida ao módulo *Avaliador de representação* para ser usada de forma prioritária em relação aos demais metadados gerados durante o processo de integração de dados, pois foram informados pelo usuário.

6.3.4 Avaliador de representação

Nesse módulo ocorre a identificação do modelo de representação do dado que está sendo recebido como fonte para integração. Para isso, o primeiro passo com os arquivos recebidos (a) é extrair os metadados físicos (b). Na sequência, com a extensão extraída,

o identificador de representação (c) direcionará o dado recebido para ser tratado com o extrator mais adequado ao modelo de representação, podendo ser um extrator para dados estruturados (d), como por exemplo:

1. Extrator de metadados de SGBD's relacionais;
2. Extrator de metadados por conteúdo estatístico;
3. Entre outros.

Ou ainda, extratores para dados não estruturados (e), como por exemplo:

1. Extrator de metadados de bases federadas;
2. Extrator de metadados de SGBD's não relacionais (NoSQL);
3. Extrator inteligente de características de texto;
4. Extrator inteligente de características de imagem;
5. Entre outros.

Independentemente do dado ser estruturado ou não estruturado, de acordo com a arquitetura, a saída dos extratores deve ser no formato do *Metadados Integrador*. Esse *Metadados Integrador* é enriquecido (f) de acordo com informações disponíveis em cada modelo de representação e de acordo com suas especificidades. O diagrama de funcionamento do *Avaliador de representação* é apresentado na figura 17.

4.(a): Avaliador de representação - Arquivos de entrada

Como fonte de dados são aceitos arquivos de dados ou conexões com banco de dados estruturados ou não estruturados e, a partir dele, são gerados metadados para realização da integração. No caso dos dados estruturados, podem-se obter os metadados das fontes de algumas formas, como por exemplo:

1. Informados pelos usuários no módulo "Semântica informada";
2. Recebidos como arquivos de entrada;
3. Extraídos diretamente do catálogo do SGBD;
4. Gerados através dos dados pelo integrador.

Já os modelos de representação não estruturados apresentam maior complexidade. Para alguns tipos é possível extrair metadados do SGBD NoSQL, entretanto na maioria dos

Figura 17 – Avaliador de representação

Fonte: Keylla Ramos Saes, 2018

casos essa tarefa demanda algoritmos inteligentes para geração de seus metadados, pois diferentemente de dados estruturados, eles não possuem metadados com estruturas claras e bem definidas.

4.(b): Avaliador de representação - Extrator de metadados físicos

Os arquivos de entrada são lidos pelo do extrator de metadados físicos com o objetivo de extrair as principais informações dos arquivos, como por exemplo: nome, extensão, tamanho, autor, tamanho do arquivo em *bytes*. Essas informações são armazenadas e posteriormente utilizadas no enriquecimento do *Metadados Integrador* de cada um dos arquivos.

4.(c): Avaliador de representação - Identificador de representação

Devido à variedade de modelos de representação aceitos nesta abordagem, a arquitetura precisa identificar e direcionar o dado ao extrator mais adequado. Para cada arquivo

recebido, pela análise da extensão do mesmo, que foi identificada previamente pelo extrator de metadados físicos, e com uso do cadastro de extensões mantido pelo módulo gestão de parâmetros, cada modelo de representação é direcionado para ser tratado individualmente por um extrator específico. No caso de o arquivo de entrada ser uma conexão com banco de dados, a conexão do banco é usada para identificar o extrator adequado àquele banco de dados. Este componente permite os seguintes modelos de representação: colunas, bases federadas, imagem, texto, além disso, permite conexão com bancos de dados estruturados e conexão para banco de dados não estruturados.

4.(d): Avaliador de representação - Extratores para dados estruturados

No caso de dados estruturados, a extração de características específicas do dado demanda o conhecimento sobre sua estrutura. Para obter tais informações estruturais, a arquitetura propõe extratores específicos de dados estruturados. Esses extratores oferecem duas formas para obtenção de dados para integração:

1. Conexão ativa: O extrator conecta-se no SGBD e extrai seu metadados;
2. Recebimento de arquivos: Os dados são enviados ao diretório de entrada do extrator, nos formatos texto colunar, JSON ou XML. Dependendo do formato do arquivo, pode ser necessário utilizar um processo de normalização de arquivos, sendo este módulo responsável pela identificação de tal necessidade.

No caso do dado a ser integrado estar armazenado em sistemas gerenciadores de banco de dados (SGBD) relacionais, com modelo de representação para dados estruturados, normalmente os metadados são armazenados em estruturas internas do SGBD, conhecidas como tabelas de catálogo. Alguns SGBD's permitem a extração desses metadados com seleção de dados ou processos de extração. Esse módulo realiza uma conexão ativa com o SGBD, por um arquivo de conexão e automaticamente realiza a extração dos metadados, como por exemplo: nome dos atributos, tamanho, tipos de dados, restrições, chave primária, chave única, precisão do dado, escala do dado, restrição de valor default, chave estrangeira e restrição de valor de domínio. O extrator é específico para determinado SGBD. Novos extratores podem ser incluídos para proporcionar a análise de SGBD sem extrator próprio.

Outro caso de extração de metadados é o recebimento do dado por meio de arquivo. Nesse caso, o metadados pode ser inferido da análise dos dados ou submetido para o

integrador. O processo de inferência do metadados é feito pela identificação de nomes para atributos e tipo de dados. A distribuição estatística dos dados é realizada para fornecer a faixa de valores contidas nos dados. Essa informação é utilizada para permitir a avaliação das faixas de valores cobertos por atributos similares.

Por tratar-se de grande volume de dados, as métricas podem ser calculadas baseadas em amostra de dados. O percentual de representatividade da amostra deverá ser parametrizado no módulo de gestão de parâmetros. O cálculo da amostra baseada nesse percentual é realizado com o uso da informação de tamanho dos arquivos que estão sendo processados, obtida pelo **Extrator de metadados físico**. No caso de conexão direta ao SGBD, esta informação é extraída do catálogo do SGBD, referente aos dados que estarão sendo processados.

4.(e): Avaliador de representação - Extratores para dados não estruturados

Diferentemente dos dados estruturados, que possibilitam definir previamente a lista de características a ser extraída, os dados não estruturados não são baseados em uma estrutura fixa a ser utilizada na formação do o *Metadados Integrador*. Portanto esse módulo, permitirá a inclusão de lista de características dinâmicas, para suportar os diversos modelos de representação.

A arquitetura proposta é expansível, portanto, permite incorporar extratores de outros modelos de representação. Como exemplo, um **Extrator inteligente de características de imagens**, pode extrair características de uma imagem com a utilização de técnicas de inteligência artificial. Esse processo deve ser capaz de receber um conjunto de imagens e dele extrair suas características do *Metadados Integrador*. Já o extrator de informações físicas dos arquivos de deve produzir outras informações para formar a lista de características, como por exemplo:

1. Image width (em pixels)
2. Image height (em pixels)
3. Bits/pixel
4. Pixel format (ex. RGB)
5. Compression rate
6. Compression

As informações combinadas desses extratores de imagem permitem o enriquecimento do *Metadados Integrador*, aumentando as possibilidades de integração.

4.(f): Avaliador de representação - Metadados integrador enriquecido

O *Metadados integrador* é um arquivo utilizado pela arquitetura proposta para descrever os dados. Esse arquivo é utilizado para apoiar o processo de integração, uma vez que sua análise permite identificar as possíveis compatibilidades entre os dados, e assim apoiar a seleção dos atributos com maior potencial para a integração. Portanto, para que a integração ocorra na arquitetura proposta, cada fonte de dados deve possuir um *Metadados Integrador*, que é gerado pelo módulo de avaliação de representação. Esse *Metadados Integrador* é enriquecido durante o processo de integração e armazenado ao final. Ou seja, a cada utilização de um dado no processo de integração, informações desses atributos são adicionadas ao *Metadados Integrador* e passam a apoiar futuras integrações. A Tabela 6 apresenta os atributos, que formam o *Metadados Integrador*, na ordem em que aparecem no arquivo, sendo que os atributos serão usados quando aplicáveis ao modelo de representação:

Tabela 6 – Atributos do *Metadados Integrador*

n.	Dimensão do Metadado	Atributo	Descrição	Componente relacionado na arquitetura
1	Origem	Modelo-dado	Estruturado / Não Estruturado	Extrator de metadados físicos
2	Origem	Programa-executor	Programa que executa ou gera o dado	Gestão de parâmetros
3	Origem	Autor-dado	Responsável ou gerador do dado	Extrator de metadados físicos
4	Especificação	Nome	Nome do atributo	Extrator de dados estruturados
5	Especificação	Tamanho	Tamanho (em dígitos) do atributo	Extrator de dados estruturados
6	Especificação	Tipo	Tipo do dado (Ex.: Text, Number, etc)	Extrator de dados estruturados
7	Especificação	Restricao-nulo	Restrição de nulidade do dado	Extrator de dados estruturados
8	Especificação	Chave-primaria	Identificador único	Extrator de dados estruturados
9	Especificação	Chave-unica	Identificador único	Extrator de dados estruturados
10	Especificação	Precisao-dado	Precisão em dígitos	Extrator de dados estruturados
11	Especificação	Escala-dado	Escala do dado	Extrator de dados estruturados
12	Especificação	Restricao-valor	Restrição de valor padrão	Extrator de dados estruturados
13	Especificação	Chave-estrangeira	Chave única de tabela relacionada	Extrator de dados estruturados
14	Especificação	Restricao-dominio	Restrição de valor de domínio aceito	Extrator de dados estruturados
15	Estatística	Valor-minimo	Valor mínimo aceito para o dado	Extrator de estatísticas
16	Estatística	Valor-maximo	Valor máximo aceito para o dado	Extrator de estatísticas
17	Estatística	Valor-medio	Valor médio aceito para o dado	Extrator de estatísticas
18	Estatística	Coefficiente-variancia	Coefficiente de variância	Extrator de estatísticas
19	Estatística	Desvio-padrao	Desvio padrão	Extrator de estatísticas
20	Semântica	Lingua-dado	Língua (português, inglês, etc)	Extrator de linguagem
21	Semântica	Area-representacao	Assunto ou área de representação	Extrator de linguagem
22	Interpretação	Sentimento-dado	Sentimento positivo ou negativo	Extrator de linguagem
23	Interpretação	Lista-caracteristicas	Características (cor, tamanho, etc)	Extrator de Características

Fonte: Keylla Ramos Saes, 2018

6.3.5 Inteligência de integração

O módulo *Inteligência de integração* é responsável por encontrar a melhor opção de integração, seja utilizando uma integração que foi realizada anteriormente, ou seja executando uma nova análise para integração. Como resultado, ele oferece uma sugestão de integração para o módulo seguinte da arquitetura.

Inicialmente, o módulo *Inteligência de integração* recebe os arquivos de *Metadados Integradores* em 5.(a). Na sequência, utiliza o histórico das integrações para verificar integrações já realizadas 5.(b) como ponto de partida para próximas integrações, e a partir disso, o processo de normalização 5.(c) de dados é realizado no dado que é utilizado pelos algoritmos de sugestão de chave de integração. O caminho a ser percorrido nos processos desse módulo é variável e é escolhido e parametrizado pelo usuário, tanto pela disponibilidade de informação para processar, quanto pela viabilidade do volume de dados, ou seja, o modelo de representação do dado de entrada e o *Metadados Integrador* gerado são utilizados para definir qual ou quais processos deverão ser executados [etapas 5.(d), 5.(e) ou 5.(f)], gerando ao final uma sugestão de chaves de integração 5.(g), como pode ser visto na Figura 18.

Figura 18 – Inteligência de integração

O componente **5.(b).Verifica histórico de integração** funciona como uma memória das sugestões de integração que agiliza o tempo de processamento de novas sugestões. Além disso, esse histórico é utilizado para treinar o modelo de sugestões de forma a otimizar a busca de novos potenciais de integração.

Dependendo do algoritmo dos dados recebidos no *Metadados Integrador*, pode ser necessária realizar a normalização de dados pelo componente **5.(c).Executa Normalização de dados**. Por exemplo, se for utilizado um algoritmo de rede neural, este não aceita dados descritivos, esperando valores da entrada em domínio entre (0-1). Como os dados do *Metadados Integrador* descrevem o dado, possivelmente não estarão nesse intervalo de valor, portanto é necessário um processo de adequação a essa faixa, ou seja, realização da normalização de dados. Neste exemplo, após a normalização, os dados são analisados com o uso de redes neurais como está apresentada no exemplo genérico na Figura 20.

Como a abordagem trata de dados não estruturadas em um contexto *Big Data*, a variedade de dados pode apresentar alguns desafios técnicos para o processamento dos dados distintos. Para lidar com esse cenário, a arquitetura oferece as três seguintes abordagens para identificação de atributos:

1. **5.(d) Agrupamento de informações: processo que utiliza técnicas de segmentação ou *clusters* de dados.** Esse processo pode trabalhar com alta variedade e alto volume de dados, e buscará de maneira automática, com uso de algoritmos de classificação, que geram os segmentos ou *clusters* de dados pela similaridade entre os dados. Os dados contidos no *Metadados integrador* são utilizados como entrada para treinamento do algoritmo, como pode ser visto na figura 19.

Figura 19 – Exemplo de uso de metadados integrador como entrada de um algoritmo de agrupamento

Fonte: Keylla Ramos Saes, 2018

2. 5.(e) **Algoritmo de aprendizagem:** processo que utiliza algoritmos de aprendizagem para seleção automática dos atributos com potencial de integração. Essa opção necessita que o algoritmo seja treinado com diversos exemplos para que a máquina aprenda a integrar de forma mais eficiente. Os dados contidos no *Metadados integrador* são utilizados como entrada para treinamento do algoritmo, como pode ser visto na figura 20.

Figura 20 – Exemplo de uso de metadados integrador como entrada de um algoritmo de rede neural

Fonte: Keylla Ramos Saes, 2018

O treinamento desse modelo de aprendizagem pode ser realizado de duas formas:

- a) **Treinamento manual do algoritmo de aprendizagem:** Deve ser realizado por um especialista de computação, com o fornecimento de exemplos de entradas

e saídas resultantes. Esse treinamento é limitado a capacidade humana de treinar a máquina e pode ser um limitador em um contexto Big Data.

- b) **Treinamento automático do algoritmo de aprendizagem:** Executado por meio do resultado dos grupos gerados pelo processo de agrupamento que são utilizados como exemplos para a seção de treinamento do algoritmo. Os exemplos dentro de um grupo podem ser usados como exemplos de similaridade e seus grupos usados como saídas previstas.

A possibilidade de combinar as opções de integração, como a combinar a opção 5.(d) Algoritmos de agrupamento para treinar o modelo de maneira automática na opção 5.(e), gerando grupos prévios como exemplos de entrada, permite a flexibilidade do processo e com isso a escala do processo de integração. Os dados contidos no *Metadados integrador* são utilizados como entrada para treinamento do algoritmo, como pode ser visto na figura 20.

3. **5.(f) Sugestão por similaridade estrutural:** processo que utiliza algoritmo baseado em regras com três informações estruturais para definição da similaridade: Nome do atributo, tamanho do atributo e tipo do atributo. Essa é a opção mais simples, porém limitada a dados que possuem as estruturas definidas, ou seja, limitando-se aos dados estruturados.

Como a arquitetura é modular, outros algoritmos podem ser utilizados, desde que recebam como entrada o *Metadados Integrador*. A seleção do algoritmo mais adequado ao problema pode ser realizada por configuração na arquitetura no módulo de gestão de parâmetros.

6.3.6 Executor de integração

Neste módulo, os dados que foram identificados com alto índice de similaridade são utilizados para realizar a integração, conforme apresentado na Figura 21.

Figura 21 – Executor de Integração

Fonte: Keylla Ramos Saes, 2018

No componente **Executa integração do dado** acontece um processo que é similar a um programa de ETL, que realiza extração, transformação e carga de dados, no qual as chaves primárias ou secundárias que foram identificadas são usadas para a realização das junções entre os dados. Nesse caso, ao invés do uso de chaves primárias das tabelas, a integração é realizada a partir dos dados com similaridade que foram identificados pelo módulo de inteligência de integração. Para isso, é realizada uma leitura nos dados buscando as informações para integração. Para cada linha é realizada uma validação do conteúdo e caso o dado origem seja compatível com o dado destino, é gerada uma nova linha integrada em um arquivo de saída, e assim sucessivamente. Um *Metadados Integrador* é gerado para o arquivo que acaba de ser integrado.

O componente **Executa integração do dado** recebe as sugestões de chaves de integração e os *Metadados Integradores* 6.(a) enriquecido pelo processo de inteligência de integração. Na sequência o processo Executa integração de dados 6.(b) realiza as validações e junções de dados baseados nas sugestões que foram recebidas como chaves integradoras e gera 6.(c) um novo *Metadado Integrador* 6.(d) para o arquivo que foi integrado. Já o componente 7.Gerador de históricos de integração armazena os *Metadados Integradores*, os dados identificados como similares e que foram utilizados no processo de

integração para a formação do históricos das integrações realizadas. Esse histórico permite o reaproveitamento processamentos de integrações que foram realizadas previamente para apoiar novos processos de integração.

6.3.7 Estatísticas da integração

O módulo de estatísticas da integração, representado na Figura 22, possui uma interface que apresenta ao usuário os resultados ou estatísticas 8.(a) do processo de integração realizado. Dentre as informações apresentadas, podemos ressaltar:

1. Arquivos origem envolvidos
2. Tamanho dos arquivos processados
3. Total de linhas integradas
4. Integrações realizadas

Figura 22 – Estatística de integração

Fonte: Keylla Ramos Saes, 2018

6.3.8 Arquivo I - Dados Integrados

Após aplicar as chaves de sugestão entre os dados de todos os arquivos envolvidos na integração, é gerado um 9.Arquivo I com o resultado da integração, apresentando em uma linha todos os dados que foram relacionados.

6.3.9 Arquivo M - Metadados dos dados integrados

Os arquivos de metadados resultantes são gerados (10.Arquivo M) de acordo com os formatos disponíveis, Dublin Core por exemplo, e podem ser utilizados para o compartilhamento dos dados, como por exemplo, em bases federadas. A Figura 23 exemplifica a geração do metadados integrador no padrão Dublin Core pelo Arquivo M. A geração do metadados já padronizados, permitirá uma ampliação no uso dos resultados da integração realizada, inclusive por outras pessoas não envolvidas na integração, por facilitar a carga dos dados em bases federadas. Por exemplo, uma integração de dados realizada com dados na área de geologia sobre os solos de uma região pode ser insumo para outra pesquisa de biologia sobre os animais que vivem no solo desta mesma região, se os dados forem disponibilizados em bases federadas e públicas.

Figura 23 – Utilização do arquivo integrado no Dublin Core

Fonte: Keylla Ramos Saes, 2018

6.3.10 Retorno da integração

Nesse componente, o usuário avalia o metadados resultante selecionando as informações relevantes. Essa classificação realizada é armazenada no histórico de integrações, de forma que o componente de inteligência da integração possa utilizá-la como indicador de qualidade da integração sugerida e permitindo melhoria dos próximos processos de sugestão de integração. Caso o usuário informe que a qualidade foi alta, o componente

de inteligência de integração utilizará essa informação sem a necessidade de reprocessar os dados, assim como se a qualidade for baixa, essa possibilidade de integração será desconsiderada pelo componente de inteligência de integração.

7 Ferramenta: Integrador de Dados

Como parte do trabalho de pesquisa realizado, a abordagem proposta foi implementada como um arcabouço para integração de dados, chamada de Integrador de Dados. Para a criação desse protótipo funcional foram desenvolvidos os módulos propostos na arquitetura apresentada no capítulo anterior. Foram escolhidas técnicas e abordagem clássicas para compor o conjunto inicial de módulos oferecidos. Conjunto esse que poderá ser ampliado com a introdução de novas técnicas e módulos adicionais.

7.1 Integrador de Dados

A ferramenta Integrador de Dados oferece as funcionalidades de controle de acesso, escolha e definição de parâmetros, módulo de identificação de extensões de arquivos de dados aceitas, módulo de normalização de dados, geração de metadados integrador, três abordagens para identificação de atributos integradores, geração de metadados dos dados integrados resultantes da integração, em padrão Dublin Core, estatísticas do processo de integração e avaliação da qualidade do resultado da integração automática.

A Figura 24 apresenta a interface utilizada para selecionar o algoritmo de integração desejado e permite selecionar múltiplos arquivos e submetê-los para integração automática dos dados. Esta funcionalidade de escolha do algoritmo foi desenvolvida apenas para a realização de testes e experimentos. Em trabalhos futuros, pode-se eleger o melhor algoritmo e parametrizá-lo no módulo gestão de parâmetros, e já aplicá-lo diretamente na integração de forma totalmente automática.

Figura 24 – Interface de envio de dados para análise e integração automática

Fonte: Keylla Ramos Saes, 2018

7.2 Normalização de dados

Os dados e metadados recebidos das diversas origens podem ter conteúdo muito distintos, mesmo quando possuem estruturas homogêneas, ou podem ser heterogêneos, e possuírem uma distribuição que inviabilize a comparação por algoritmos. Dessa forma, é necessária utilizar uma estratégia como normalização de dados, para que os metadados possam ser comparados de maneira automática. O módulo de normalização cria arquivos com *Metadados Integradores* normalizados para cada fonte de dados, dessa forma, todas as fontes contarão com a mesma estrutura e componentes no arquivo de metadados com conteúdos em distribuições que podem ser analisadas automaticamente. Um exemplo de *Metadados Integrador* é apresentado na figura 25, que mostra a estrutura do metadados para um arquivo de entrada que contém dois dados, sendo eles UF e CODIGO SIAFI MUNICIPIO. Como pode ser visto na figura, nem todos os atributos são preenchidos, mas todos existem na estrutura do *Metadados Integrador*, permitindo dessa forma uma comparação entre os metadados.

Figura 25 – Exemplo de Metadados Integrador

```

1  [{"A_Nome": "UF",
2    "Area_representacao": "",
3    "Autor_dado": "",
4    "Coeficiente_variancia": "NI",
5    "Desvio_padrao": "NI",
6    "Chave_estrangeira": "",
7    "Chave_primaria": "",
8    "Chave_unica": "",
9    "Escala_dado": "",
10   "Lingua_dado": "",
11   "Lista_caracteristicas": "",
12   "Precisao_dado": "",
13   "Programa_executor": "",
14   "Restricao_dominio": "",
15   "Restricao_nulo": "NO",
16   "Restricao_valor": "",
17   "Sentimento_dado": "",
18   "Tamanho": "2",
19   "Tipo": "Nao Numerico",
20   "Valor_maximo": "NI",
21   "Valor_medio": "NI",
22   "Valor_minimo": "NI"
23 }, {
24   "A_Nome": "CODIGO SIAFI MUNICIPIO",
25   "Area_representacao": "",
26   "Autor_dado": "",
27   "Coeficiente_variancia": "4529067.53684",
28   "Desvio_padrao": "2128.1605994",
29   "Chave_estrangeira": "",
30   "Chave_primaria": "",
31   "Chave_unica": "",
32   "Escala_dado": "",
33   "Lingua_dado": "",
34   "Lista_caracteristicas": "",
35   "Precisao_dado": "",
36   "Programa_executor": "",
37   "Restricao_dominio": "",
38   "Restricao_nulo": "NO",
39   "Restricao_valor": "",
40   "Sentimento_dado": "",
41   "Tamanho": "4",
42   "Tipo": "Numerico",
43 }
```

Fonte: Keylla Ramos Saes, 2018

Existem diversas estratégias de normalização de dados, para o desenvolvimento desse trabalho, foram escolhidas duas abordagens que atendem ao caso base, entretanto devido à flexibilidade da arquitetura, outras estratégias de normalização podem ser aplicadas no futuro.

7.2.1 Normalização de dados numéricos

Em problemas de integração de dados, existem casos de uso cujos valores recebidos como fontes de dados são numéricos e esparsos, com limites e distâncias grandes. Nesses casos, para que os algoritmos tenham maior precisão, é necessária a realização de uma normalização dos dados fontes. Uma possibilidade de normalização é realizar ajustes em todos os valores recebidos de acordo com o maior valor da lista. O componente desenvolvido encontra o maior valor da lista de entrada e divide cada dado recebido pelo maior valor, fazendo com que todos os números sejam adequados em uma mesma escala, conforme pode ser visto no exemplo do Algoritmo 1.

Algoritmo 1 Normaliza dados pelo maior valor da lista recebida

Require: normalizaValorMax (listaEntrada)

```

1: maior = max(listaEntrada)
2: for item in listaEntrada do
3:   valorNormal = (item / maior)
4:   listaSaida.append(valorNormal)
5: listaSaida

```

Fonte: Keylla Ramos Saes, 2018

7.2.2 Normalização de dados Categóricos

Outro cenário comum em problema de integração de dados são fontes de dados categóricas, podendo ser de tipos nominais ou ordinais. Um exemplo de dado nominal seria o atributo “sexo da pessoa” ou “cor”. Esse tipo de dado possui um conteúdo descritivo conhecido, porém não ordenado. Já dados ordinais, mostraram uma ordenação, como por exemplo “mês”. Alguns algoritmos inteligentes não trabalham com dados categóricos, ou seja, não aceitam esses dados descritivos como entrada e precisam de valores numéricos. Devido a essa limitação, desenvolveu-se um componente que converte valores categóricos em valores numéricos.

7.3 Componente de sugestão de integração

Para exemplificar o conceito da utilização do *Metadados integrador*, como instrumento para integração de dados estruturados e não estruturados, foi realizada a implementação de três tipos de componentes de integração. A escolha dos componentes baseou-se nos casos de uso proposto para este trabalho no capítulo 8. Outros algoritmos também podem ser implementados devido à modularização da arquitetura. Os casos atualmente implementados são:

1. **Integração por algoritmo de índice de similaridade:** Algoritmo baseado em regras de comparação de similaridade entre as estruturas de dados, em que as chaves de integração são definidas pela análise de regras da estrutura como por exemplo, nome do dado, tamanho e tipo de dado.
2. **Integração por algoritmo de agrupamento:** Utilização algoritmos para agrupamento de dados como por exemplo, o KMeans ou o dendograma, por meio dos atributos do *Metadados integrador*, permite que sejam apresentados grupos ou *clusters* de acordo com a similaridade dos dados.
3. **Integração por algoritmo de aprendizagem:** O histórico de integrações é fator essencial nessa abordagem, que demanda o treinamento do algoritmo com exemplos de dados fontes e saída esperadas, para integrar os dados com o uso dos *Metadados integradores*.

Esses algoritmos foram selecionados por tratarem dos casos de uso utilizados nesse trabalho, porém outros algoritmos podem ser implementados na arquitetura. Isso porque, a arquitetura permite parametrizar antes da escolha dos dados e qual componente de integração é utilizado para que a integração ocorra de maneira automática.

Integração por algoritmo de índice de similaridade

Um exemplo de implementação foi realizada por meio da utilização de alguns atributos estruturais, e foi denominada de *Integração por algoritmo de índice de similaridade*. Essa implementação é focada na aplicação de regras sobre as estruturas de dados, e utiliza apenas algumas características válidas somente para dados estruturais, contidas no *Metadados Integrador* como variáveis em sua composição:

1. Nome do dado;
2. Tamanho do dado;
3. Tipo do dado.

Para essa implementação foi desenvolvido um exemplo de componente de avaliação de representação que compara os *Metadados integradores*, e possui a função principal de encontrar um fator de integração entre cada comparação realizada.

São realizados tratamentos textuais de acordo com os módulos disponíveis na estrutura, antes da comparação efetiva dos metadados:

1. Os atributos são transformados em caixa alta;
2. As *Stop Words* (palavras que não mudam o sentido, como preposições, artigos, etc) deverão ser retiradas;
3. Manter apenas o radical das palavras para comparação.

Após a aplicação dos tratamentos textuais, as características específicas dos *Metadados Integradores* são comparadas entre si. A comparação é feita baseada em regras, como por exemplo, um cálculo para similaridade e uma classificação para os resultados obtidos. Um exemplo genérico de cálculo de similaridade é apresentado cálculo do índice de similaridade (IS), descrito pela fórmula 1, que pode ser tratado por qualquer exemplo na arquitetura.

$$IS = ((SimilaridadeMetadado * 2) + (SimilaridadeTamanho) + (SimilaridadeTipo)) / 4 \quad (1)$$

Novas variáveis podem ser incluídas no modelo de cálculo, desde que existam no *Metadados Integrador*. A cada variável incluída a regra genérica de integração, representada pela Fórmula 1, a regra deve ser recriada considerando a nova variável e as novas condições de peso na fórmula por ela gerada.

Por ser uma implementação baseada em regras, definiu-se um exemplo dos limites para identificar o índice de similaridade utilizado durante os testes. A abordagem proposta permite que esses parâmetros sejam alterados no módulo Gestão de parâmetros, caso seja necessário.

- $IS \leq 89\%$ - Baixo
- $IS \geq 90\%$ - Alto

Neste exemplo, baseando-se nos limitadores desenvolvidos, os índices de similaridade com valores "Baixo" são descartados no processo de integração, por apresentarem baixa similaridade entre os dados em processamento. Já os índices de similaridade de valor "Alto" são considerados satisfatórios e indicam uma integração automática.

Esse componente apresenta a vantagem de não precisar de dados históricos ou de treinamento para executar a integração, pois é um algoritmo baseado em regras e parâmetros e não é um algoritmo com técnicas de aprendizagem. Porém, está limitado a tratar dados que possuam uma estrutura mínima para comparação, como nome do atributo, tamanho e tipo do dado, o que pode ser um limitador para alguns tipos de dados heterogêneos que não possuam essa estrutura.

Integração por algoritmo de agrupamento

Devido à variedade de atributos e diversidade no modelo de representação, uma abordagem possível é a realização de agrupamento dos dados para identificação de similaridade, e para isso sugere-se a utilização de um algoritmo de **agrupamento das informações**. A proposta para esse componente é utilizar um algoritmo classificador para gerar grupos (*clusters*), que possam representar um mesmo conceito no mundo real. Isso porque agrupar conceitos pode facilitar a análise de similaridade entre os atributos pertencentes à cada grupo com a referência conceitual no mundo real. Essa característica de agrupamento é interessante quando o contexto em é Big Data, com alto volume e variedade de dados, pois agrupa automaticamente os dados de acordo com suas características de maneira automática. O algoritmo 2 apresenta um exemplo de processo de recebimento do *Metadados Integrador* com uso de um parâmetro de lista de entrada e retorna o clusters, sendo que cada cluster retornará os atributos similares.

Algoritmo 2 Realiza agrupamento dos atributos dos metadados integradores

Require: agrupamento (listaEntrada)

matrixNormalizada = normalizaMetadadosIntegrador(listaEntrada)

matrixTransposta = transpor(matrixNormalizada)

kmeans = KMeans(parametros)

ykmeans = kmeans.fitpredict(matrixTransposta)

clusters = kmeans.clustercenters

retorna clusters

Fonte: Keylla Ramos Saes, 2018

Integração por algoritmo de aprendizagem

Outra abordagem que permite a automatização da integração é o uso de **algoritmo de aprendizagem** por classificação como por exemplo, o algoritmo 3 que é uma rede neural. Esse algoritmo utilizará o *Metadados integrador* como parâmetro de entrada dessa rede, e após um treinamento do algoritmo, ele fará associações internas baseadas no treinamento, e com regras desconhecidas dos usuários, para gerar as chaves de similaridade. O algoritmo 3 apresenta um exemplo de processo de recebimento do *Metadados Integrador* por um parâmetro de lista de entrada e retorna uma classificação para cada atributo. O histórico de integrações realizadas por especialistas é utilizado para treinar esse modelo de redes neurais, e esse treinamento não está representado no algoritmo.

Algoritmo 3 Executa uma rede neural para identificação dos atributos similares dos metadados integradores

Require: agrupamento (listaEntrada)

- 1: matrixNormalizada = normalizaMetadadosIntegrador(listaEntrada)
- 2: matrixTransposta = transpor(matrixNormalizada)
- 3: model.compile(parametros)
- 4: result = model.predict(matrixTransposta)
- 5: return result

Fonte: Keylla Ramos Saes, 2018

O algoritmo de rede neural utilizado neste trabalho foi implementado através do Keras e Tensorflow, sendo parametrizado com 18 neurônios de entrada, 9 na camada intermediária e executado com 500 épocas.

A Figura 26 apresenta o resultado do processamento realizado com os atributos integráveis e o grau de integração entre eles. Esse passo é opcional e foi adicionado neste exemplo para validação dos experimentos, mas a integração pode ser realizada sem a necessidade dessa avaliação visual.

Figura 26 – Interface de sugestão de integração

Chaves de integração sugeridas

```
{
  "Campo Origem": "UF"
  "Campo Provavel Matching": "uf"
  "Índice de Integração": "Alto"
}
```

```
{
  "Campo Origem": "Nome"
  "Campo Provavel Matching": "nome"
  "Índice de Integração": "Alto"
}
```

Integrar

Fonte: Keylla Ramos Saes, 2018

Após a finalização do processo de integração, são apresentadas as estatísticas do processamento, representados no exemplo da Figura 27. Nesse momento, o usuário pode classificar a qualidade do resultado da integração como "Correto", "Aproximado" ou "Errado". Essa classificação é utilizada para qualificar a integração realizada. O histórico de avaliação é utilizado na avaliação de estratégias de integração de dados similares.

Figura 27 – Interface com exemplo das estatísticas de processamento da integração e avaliação de resultados

Nome do primeiro arquivo	C:\Users\HP\Dropbox\0.USP_EACH_PPg5\Tese_Qual\Testes\temp_Delimitado_json.tmp
Quantidade de Linhas do primeiro	12
Nome do segundo arquivo	C:\Users\HP\Dropbox\0.USP_EACH_PPg5\Tese_Qual\Testes\cliente_web2.json
Quantidade de Linhas do segundo	7
Nome do Resultado Integração	C:\Users\HP\Dropbox\0.USP_EACH_PPg5\Tese_Qual\Testes\MetadadosBolsaFamilia.txt
Combinações de Integração	0
Índice de Integração	0

Classifique o resultado da integração realizada: Selecione

Avaliar **Voltar**

Selecione
Correta
Aproximada
Errada

Fonte: Keylla Ramos Saes, 2018

As informações apresentadas na Figura 27 são os nomes dos arquivos ou conexões fontes, o tamanho das fontes de dados, a quantidade de combinações e o índice de integração, ou similaridade encontrado nesse processamento.

Uma vez realizada a integração, os dados integrados são gerados em um arquivo. A Figura 28 apresenta um exemplo do arquivo com a integração consolidada, no formato json.

Figura 28 – Exemplo de arquivo gerado pela integração consolidada

```

1  {"email": "████████@gmail.com",
2  "nome": "JACINEIDE ██████████",
3  "genero": "Female",
4  "id": 2,
5  "ip_address": "27.56.193.2",
6  "ultimo nome": "VILAS",
7  "uf": "PE",
8  "UF": "PE",
9  "Código SIAFI Município": "2497",
10 "Nome Município": "OURICURI",
11 "Código Função": "8",
12 "Código Subfunção": "244",
13 "Código Programa": "1335",
14 "Código Ação": "8442",
15 "NIS Favorecido": "1623████████",
16 "Nome": "JACINEIDE ██████████",
17 "Fonte-Finalidade": "CAIXA - Programa Bolsa Família",
18 "Valor Parcela": "██████",
19 "Mês Competência": "mar/17"}
20
21
22

```

Fonte: Keylla Ramos Saes, 2018

Uma vez integrados, os dados integrados passam a contar com um metadado que apoia o processo de compartilhamento em bancos de dados federados, como por exemplo DataOne, Kaggle, CKAN entre outros. A interface do *Integrador de Dados* oferece o recurso para criação do metadados padrão XML para o compartilhamento do arquivo resultante, conforme exemplo da Figura 29. O padrão utilizado no exemplo é o *Dublin Core*.

Figura 29 – Interface para geração de metadados Dublin Core

Metadados Dublin Core

Geração do Metadados Dublin Core

Title	C:\Testes\1201703_BolsaFamíliaFolhaPagamento_Testes.txt
Subject	
Description	
Type	Integração
Source	C:\Testes\1201703_BolsaFamíliaFolhaPagamento_Testes.txt
Relation	
Coverage	
Creator	Keylla Ramos Saes
Publisher	Keylla Ramos Saes
Contributor	Keylla Ramos Saes
Rights	
Date	18/02/2018 18:12:04
Format	
Identifier	
Language	Português
Audience	
Provenance	
RightsHolder	
InstructionalMethod	
*AccrualMethod	
*AccrualPeriodicity	
*AccrualPolicy	

Fonte: Keylla Ramos Saes, 2018

A figura 29 apresenta a interface para complementação dos dados necessários para criar os metadados no formato escolhido. No exemplo, a interface coleta os dados referentes ao padrão *Dublin Core*. Esse recurso simplifica o compartilhamento, ao evitar que o arquivo de metadados gerado necessite ser editado, manualmente ou por outros recursos, para que o compartilhamento ocorra.

8 Experimentos

8.1 Casos de uso para integração de dados heterogêneos

Durante este trabalho foram mapeados alguns casos de uso que representem diferentes cenários de integração de dados homogêneos e heterogêneos e que serão utilizados para discussão ou realização de experimentos. A Tabela 7 apresenta alguns dos casos possíveis de integração entre os modelos de representação para dados estruturados e dados não estruturados e a complexidade que apresentam quanto à manipulação de dados Big Data durante o processo de integração.

Tabela 7 – Casos de uso de integração de dados estruturados e não estruturados e suas complexidades

Núm.	Modelo de dados A	Modelo de dados B	Estrutura do dado	Complexidade Humana	Complexidade Computacional
1	Dado estruturado	Dado estruturado	Homogênea	Baixa	Baixa
2	Dado estruturado	Dado estruturado	Heterogênea	Média / Alta	Média / Alta
3	Dado estruturado	Dado não-estruturado	Heterogênea	Alta	Alta
4	Dado não-estruturado	Dado estruturado	Heterogênea	Alta	Alta
5	Dado não-estruturado	Dado não-estruturado	Heterogênea	Alta	Alta

Fonte: Keylla Ramos Saes, 2018

A complexidade da integração pode ser observada por duas perspectivas, a perspectiva da complexidade humana para interpretar a alta variedade e volume de dados e a perspectiva computacional para processar essa mesma variedade e volume de dados. Somente em casos onde a estrutura é completamente idêntica pode-se dizer que a complexidade humana para integração do dado é baixa, nos demais casos, a complexidade de realização de processos manuais é alta.

O objetivo dos experimentos é validar alguns casos definidos neste trabalho. Devido a abordagem utilizar uma estrutura pré-definida e única para qualquer modelo de representação, a validação destes casos permite extensão da validação para outros casos com dados mais heterogêneos, pois a solução utiliza a estrutura do *Metadados Integrador*.

8.2 Experimento 01 - Integração de dados do Bolsa família com uso do algoritmo de similaridade

O Experimento 01 conduziu a integração de duas amostras de 1 (um) milhão de registros dos meses de agosto e setembro de 2018, utilizando o algoritmo de índice de similaridade apresentado neste trabalho no capítulo 7. Esse algoritmo utiliza somente a

estrutura mínima do dado para identificar a similaridade para integração. A Figura 30 apresenta o resultado do experimento, onde pode ser observado que alguns atributos são identificados com alta similaridade e outros com média.

Figura 30 – Experimento 01

Amostra de 1 milhão de registros do Bolsa Família dos meses 08/18 e 09/18

```
{["Indice Integracao": "Medio", "Campo Provavel Matching": "MES REFERENCIA", "Campo Origem": "MES REFERENCIA"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "VALOR PARCELA", "Campo Origem": "VALOR PARCELA"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "MES COMPETENCIA", "Campo Origem": "MES COMPETENCIA"},
{"Indice Integracao": "Alto", "Campo Provavel Matching": "NOME FAVORECIDO", "Campo Origem": "NOME FAVORECIDO"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "NIS FAVORECIDO", "Campo Origem": "NOME FAVORECIDO"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "CODIGO MUNICIPIO SIAFI", "Campo Origem": "CODIGO MUNICIPIO SIAFI"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "NOME FAVORECIDO", "Campo Origem": "NIS FAVORECIDO"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "NIS FAVORECIDO", "Campo Origem": "NIS FAVORECIDO"},
{"Indice Integracao": "Alto", "Campo Provavel Matching": "UF", "Campo Origem": "UF"},
{"Indice Integracao": "Medio", "Campo Provavel Matching": "NOME MUNICIPIO", "Campo Origem": "NOME MUNICIPIO"}]
```

Fonte: Keylla Ramos Saes, 2018

8.3 Experimento 02 - Integração de dados do Bolsa família com uso do algoritmo de agrupamento

O experimento 02 utilizou as mesmas amostras de 1 (um) milhão de registros do bolsa família dos meses de agosto e setembro de 2018, entretanto utilizou o algoritmo de agrupamento, implementado em um dendograma. A Figura 31 apresenta uma similaridade de 100% em todos os atributos do arquivo, ou seja, todas as colunas da primeira amostra de dados encontraram uma coluna correspondente na segunda amostra de dados.

Figura 31 – Experimento 02

Fonte: Keylla Ramos Saes, 2018

8.4 Experimento 03 - Integração de dados do Bolsa família entre os anos de 2016 e 2018

O experimento 03 também foi realizado com o uso das mesmas amostras de 1 (um) milhão de registros do bolsa família dos meses de agosto e setembro de 2018, entretanto tentou integrá-las com amostras dos meses de janeiro e fevereiro de 2016, utilizando o algoritmo de agrupamento, implementado em um dendograma. Como pode ser avaliado na Figura 32, embora os dados são do mesmo programa, a estrutura dos arquivos mudou entre os anos. O atributo "Valor do benefício" encontrado nos dados de 2016, teve seu nome alterado para "Valor da parcela" nos dados de 2018, assim como o atributo "Nome do beneficiário" nos dados de 2016 foi alterado para "Nome do Favorecido". Este exemplo mostra que até mesmo dados do mesmo processo de negócio podem sofrer alterações no decorrer do tempo, e em alguns casos a mudança pode ser além da nomenclatura, o tipo do dado, ou até mesmo a inclusão ou exclusão desses dados.

Figura 32 – Experimento 03

Fonte: Keylla Ramos Saes, 2018

8.5 Experimento 04 - Integração de dados do Bolsa família com uso do algoritmo de rede neural

O experimento 04 foi conduzido com a amostras de 1 (um) milhão de registros do bolsa família dos meses de agosto e setembro de 2018, entretanto utilizou o algoritmo de rede neural. Para a realização desse experimento foi realizado um treinamento da rede neural com uma amostra de dados do bolsa família. Como pode ser visto na Figura 33, que representa uma matriz de confusão dos dados utilizados durante a etapa de testes do algoritmo, o mesmo encontrou 4 atributos similares, quando realmente eram similares

e não encontrou 56 similaridades, que realmente não existiam, gerando uma acurácia próxima a 94%. Essa abordagem exige um treinamento do modelo, sendo necessário o suporte de um especialista humano na etapa de treinamento.

Figura 33 – Experimento 04

	Similaridade não encontrada	Similaridade encontrada
Similaridade não existente	56	0
Similaridade existente	4	4

Acurácia: 0.9375%

Fonte: Keylla Ramos Saes, 2018

8.6 Experimento 05 - Integração de dados do Bolsa Família com Bolsa alimentação PRACE

O experimento 05 foi o último realizado com o uso das amostras de 1 (um) milhão de registros do bolsa família dos meses de agosto e setembro de 2018 e buscou integrá-lo com dados da bolsa alimentação PRACE. Esses dois arquivos possuem estruturas distintas e dados referentes a processos de negócio muito distintos, entretanto ambos se referem a pessoas físicas.

O experimento utilizou o algoritmo de agrupamento para realizar a identificação de similaridades, e como pode ser visto encontrou alguns atributos similares que são relevantes, como podemos destacar o atributo "Nome" do Bolsa alimentação PRACE, com o "Nome do favorecido" no Bolsa família. Nesse exemplo, podemos visualmente identificar que tratam-se de nomes de pessoas, e verificar que a sugestão de integração foi boa, como pode ser visto na Figura 34

Figura 34 – Experimento 05

8.7 Experimento 06 - Integração de cadastro de cachorros e imagens de cachorros

No experimento 06 foi utilizado um cadastro de cachorros para avaliar a similaridade com fotos de cachorros. Na implementação realizada nesse trabalho utilizou-se uma API que descreve a figura em inglês, portanto o cadastro dos cachorros também foi utilizado em inglês. O objetivo do experimento foi identificar características na imagem que pudessem indicar a integração com o cadastro dos cachorros. Foi utilizado o algoritmo de agrupamento, e com a comparação do *metadados integrador* gerado com a lista de características da imagem, e o *metadados integrador* do cadastro de cachorros, pode ser visto na Figura 35 que foi identificado alguma similaridade entre os atributos.

Figura 35 – Experimento 06

8.8 Experimento 07 - Integração de imagens de sapatos

O objetivo desse experimento foi demonstrar que os dados não estruturados, como imagens, podem ter suas características extraídas pelo *metadados integrador* e comparados entre si para encontrar as similaridades. Foi utilizado o algoritmo de agrupamento, para a comparação dos *metadados integradores* de cada imagem de tipos distintos de sapatos, que foi gerado pela a lista de características das imagens. Como mostra a Figura 36, foi identificado similaridade entre as imagens pela sua lista de características, que é um item do *metadados integrador*.

Figura 36 – Experimento 07

Fonte: Keylla Ramos Saes, 2018

8.9 Experimento 08 - Caso de uso

Esse experimento mostrou que imagens de classes distintas, poderiam ter suas características extraídas e armazenadas no *metadados integrador* e comparados entre si para encontrar as similaridades. Também foi utilizado o algoritmo de agrupamento para demonstrar a comparação dos *metadados integradores* de cada imagem. Como pode ser visto na Figura 37, o item lista de características da imagem permitiu agrupar elementos de sapatos e cachorros de forma distintas.

Figura 37 – Experimento 08

Fonte: Keylla Ramos Saes, 2018

8.10 Proposta para futuros Experimentos

Além dos experimentos acima citados, futuros experimentos podem ser realizados para demonstrar a possibilidade de automatizar a integração. Uma sugestão seria realizar a comparação entre um conjunto aleatório de fotos de pessoas com um cadastro de pessoas que possui uma estrutura json para identificar as possíveis pessoas correspondentes por características dos dados cadastrais, o que auxiliaria a consolidar um perfil completo de pessoas. Um exemplo esquemático sobre a proposta desse experimento de integração de dados heterogêneos pode ser visto na Figura 38.

Figura 38 – Experimento de integração de dados heterogêneos

Fonte: Keylla Ramos Saes, 2018

9 Discussões, conclusão e trabalhos futuros

A popularização de soluções Big Data com variedade crescente de modelos de representação de dados proporciona um novo desafio de pesquisa para a comunidade de banco de dados, que é a integração de modelos de dados estruturados e não estruturados, ou seja, integração de modelos de representação de dados heterogêneos, como foi discutido neste trabalho.

Embora avanços tenham ocorrido nos últimos anos com a evolução das arquiteturas de integração, ainda persiste a dificuldade para integrar modelos homogêneos. Essa dificuldade é ampliada ao tratarmos da integração de modelos de dados heterogêneos, considerando como solução a integração de dados sem intervenção manual, ou seja, de forma automática e síncrona. A integração totalmente automatizada ainda é limitada a alguns casos específicos devido à complexidade dos dados, como foi demonstrado na revisão sistemática realizada. Para promover o avanço dos processos de integração, novas soluções precisam ser desenvolvidas.

A normalização de dados é uma etapa de grande impacto no processo de integração de dados, sejam os dados homogêneos ou heterogêneos. Esse é o momento no qual é necessária encontrar uma escala comum para que a máquina possa realizar análises e encontrar similaridades pelos metadados e dados indicando as chaves de integração. Nos casos de compreensão mais fáceis, o volume de bancos de dados a serem integrados torna essa tarefa complexa para a realização humana. Já sob a perspectiva computacional para processamento de dados *Big Data*, existe uma complexidade menor para os casos onde a estrutura é previamente conhecida, como os modelos de representação para dados estruturados, já nos demais casos a complexidade é maior, porém a capacidade de processamento atual permite o processamento de grandes volumes de dados, o que torna viável as atividades de integração, embora sigam sendo complexas.

Este trabalho desenvolveu uma abordagem com uma arquitetura técnica modular e flexível, baseada na geração do *Metadados integrador*, que busca extrair do dado de entrada, considerando seus diferentes modelos de representação, características que permitam a realização de análises comparativas dos dados e que contribuam para integração automática dos bancos de dados. Essa contribuição reduz o oneroso trabalho de mapeamento de dados

realizado, em geral de forma manual, ou com suporte parcial de ferramentas, pelos especialistas de computação durante o desenvolvimento de um processo de integração.

Os experimentos demonstraram que até mesmo em casos básicos, como integrar dados de Bolsa família de distintos anos, pode haver heterogenia da estrutura dos dados, o que demanda a necessidade de intervenção humana para ajuste da integração. Para casos mais complexos de integração, que lidam com dados não estruturados, em um contexto de alta variedade de dados, são requeridas técnicas auxiliares para criação de metadados. Nesses casos é possível utilizar algoritmos de inteligência artificial para contribuir com a construção do *Metadados Integrador*, ao gerar informações descritivas para formatos pouco convencionais, como por exemplo, textos, fotos e vídeos.

A evolução dos processos de integração pode ser realizada de maneira automática por meio de técnicas de inteligência artificial, representadas por exemplo, por algoritmos de agrupamento e redes neurais, permitindo que as integrações evoluam e sejam otimizadas, conforme são realizadas.

Uma outra contribuição da abordagem proposta é a utilização de um componente para após integrar os dados, gerar um metadados para compartilhamento em bases federadas, em padrão de mercado, como por exemplo Dublin Core. Isso significa que o dado integrado resultante poderá ser compartilhado, ampliando assim a possibilidade de disponibilização de dados e compartilhamento de informações entre pesquisadores e analistas de dados, o que pode gerar expansão de conhecimento em diversas áreas.

9.1 Conclusão

Este trabalho apresentou uma alternativa com uma nova abordagem de integração que utilize os dados e seus diversos metadados, como elementos adicionais para uma abordagem de integração realizados de forma automática e com recursos de inteligência artificial para análise de similaridades e seleção dos atributos integradores. Como contribuições pode-se considerar:

1. A realização de uma revisão sistemática da literatura;
2. Geração de uma classificação das arquiteturas distintas para integração de dados heterogêneos;
3. Proposta de uma forma de representação por meio do *Metadados integrador*;

4. Proposta de uso de técnicas de inteligência artificial para extrair características de dados não estruturados como por exemplo, textos e imagens;
5. Uso de técnicas de inteligência artificial para encontrar de maneira automática as similaridades entre os atributos para integração.

9.2 *Trabalhos futuros*

Este trabalho apresentou uma abordagem com o uso de uma arquitetura modular e expansível para a integração de dados estruturados e não estruturados em um contexto Big Data. Essa proposta modular permite que o trabalho abra um leque de oportunidades futuras, como por exemplo:

- Abordar novos modelos de representação de dados nos experimentos;
- Estender o *Metadados Integrador* com outros tipos de atributos;
- Comparar as técnicas de inteligência artificial para selecionar a técnica mais adequadas para esse problema de integração de dados heterogêneos;
- Avaliar o desafio de integrar as diversas bases federadas;
- Avaliar outras características da integração em um contexto *Big Data*, além da variedade de fontes.

Referências¹

- ABOLHASSANI, N. et al. Knowledge graph-based query rewriting in a relational data harmonization framework. In: . [S.l.: s.n.], 2017. p. 433–438. ISBN 9781509046072. Cited By 0. Citado 2 vezes nas páginas 49 e 103.
- ALLARD, S. Dataone: Facilitating escience through collaboration. *Journal of eScience Librarianship*, v. 1, n. 1, p. 3, 2012. Citado na página 28.
- BECHHOFFER, S. Owl: Web ontology language. In: *Encyclopedia of database systems*. [S.l.]: Springer, 2009. p. 2008–2009. Citado na página 26.
- BELLO-ORGAZ, G.; JUNG, J. J.; CAMACHO, D. Social big data: Recent achievements and new challenges. *Information Fusion*, Elsevier, v. 28, p. 45–59, 2016. Citado na página 17.
- BENKER, T. A hybrid olap and oltp architecture using non-relational data components. In: . [s.n.], 2013. P222, p. 41–54. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908273494&partnerID=40&md5=1c54db7c8c1670488ed3183d7b7a2245>. Citado 2 vezes nas páginas 43 e 103.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific american*, JSTOR, v. 284, n. 5, p. 34–43, 2001. Citado na página 26.
- BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked data-the story so far. *International journal on semantic web and information systems*, v. 5, n. 3, p. 1–22, 2009. Citado na página 26.
- BLUMBERG, R.; ATRE, S. The problem with unstructured data. *Dm Review*, POWELL PUBLISHING INC, v. 13, n. 42-49, p. 62, 2003. Citado na página 17.
- BRYANT, R.; KATZ, R. H.; LAZOWSKA, E. D. *Big-data computing: creating revolutionary breakthroughs in commerce, science and society*. [S.l.]: December, 2008. Citado na página 17.
- CALDAROLA, E. G.; RINALDI, A. M. Big data: A survey. In: SCITEPRESS-SCIENCE AND TECHNOLOGY PUBLICATIONS, LDA. *Proceedings of 4th International Conference on Data Management Technologies and Applications*. [S.l.], 2015. p. 362–370. Citado na página 37.
- CARVALHO, M. G. D. et al. An evolutionary approach to complex schema matching. *Information Systems*, Elsevier, v. 38, n. 3, p. 302–316, 2013. Citado na página 26.
- CHANG, B. R.; TSAI, H. F.; WANG, Y. A. Optimized multiple platforms for big data analysis. In: *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. [S.l.: s.n.], 2016. p. 155–158. Citado 2 vezes nas páginas 48 e 103.
- CHEN, H.; CHIANG, R. H.; STOREY, V. C. Business intelligence and analytics: from big data to big impact. *MIS quarterly*, JSTOR, p. 1165–1188, 2012. Citado na página 19.
- CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. *Mobile networks and applications*, Springer, v. 19, n. 2, p. 171–209, 2014. Citado 2 vezes nas páginas 17 e 37.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. In: *Readings in artificial intelligence and databases*. [S.l.]: Elsevier, 1988. p. 98–111. Citado na página 18.

CHEN, X. et al. Creating multi-domain query plans on data services. In: *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*. [S.l.: s.n.], 2015. p. 661–664. Citado 2 vezes nas páginas 47 e 103.

CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, ACM, v. 13, n. 6, p. 377–387, 1970. Citado na página 17.

DEY, A. et al. Metadata-as-a-service. In: . IEEE Computer Society, 2015. v. 2015-June, p. 6–9. ISBN 9781479984411. ISSN 10844627. Cited By 2. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84944321906&doi=10.1109%2fICDEW.2015.7129536&partnerID=40&md5=0ae0aa9a05d04e3b0e431dce2f4e16bb>. Citado 2 vezes nas páginas 47 e 103.

DOSHI, K. A. et al. Blending sql and newsql approaches: Reference architectures for enterprise big data challenges. In: *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 163–170. Citado 3 vezes nas páginas 22, 43 e 103.

DUBLIN Core Metadata Initiative. 2018. Online; accessed 20 January 2018. Disponível em: <https://dublincore.org/>. Citado na página 30.

DUVAL, E. Metadata standards: What, who & why. *Journal of Universal Computer Science*, v. 7, n. 7, p. 591–601, 2001. Citado na página 28.

FANG, S. et al. An integrated system for regional environmental monitoring and management based on internet of things. *IEEE Transactions on Industrial Informatics*, v. 10, n. 2, p. 1596–1605, May 2014. ISSN 1551-3203. Citado 2 vezes nas páginas 46 e 103.

GADEPALLY, V. et al. D4m: Bringing associative arrays to database engines. In: . Institute of Electrical and Electronics Engineers Inc., 2015. ISBN 9781467392860. Cited By 7. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84964822753&doi=10.1109%2fHPEC.2015.7322472&partnerID=40&md5=b170bb38bf6a1ea9c48bfc920c715f68>. Citado 2 vezes nas páginas 48 e 103.

GARG, B.; KAUR, K. Integration of heterogeneous databases. In: *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in*. [S.l.: s.n.], 2015. p. 1033–1038. Citado 3 vezes nas páginas 46, 47 e 103.

GEIHS, K. Middleware challenges ahead. *Computer*, IEEE, v. 34, n. 6, p. 24–31, 2001. Citado na página 46.

GIANNAKOURIS, V. et al. Musqle: Distributed sql query execution over multiple engine environments. In: . Institute of Electrical and Electronics Engineers Inc., 2016. p. 452–461. ISBN 9781467390040. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015196605&doi=10.1109%2fBigData.2016.7840636&partnerID=40&md5=f87d4474dec002bc5274f07915ab0c6>. Citado 2 vezes nas páginas 48 e 103.

GROGER, C.; SCHWARZ, H.; MITSCHANG, B. The deep data warehouse: Link-based integration and enrichment of warehouse data and unstructured content. In: S. REICHERT M., G. G. R.-M. (Ed.). Institute of Electrical and Electronics Engineers Inc., 2014. v. 2014-December, n. December, p. 210–217. ISSN 15417719. Cited By 1. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84937423523&doi=10.1109%2fEDOC.2014.36&partnerID=40&md5=5a090cc0c331e067366bc90e773c9a95>. Citado 2 vezes nas páginas 49 e 103.

GRUND, M. et al. Hybrid graph and relational query processing in main memory. In: *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*. [S.l.: s.n.], 2013. p. 23–24. Citado 2 vezes nas páginas 44 e 103.

GUERRA, F. et al. The re-search alps (research laboratories in the alpine area) project. In: IEEE. *High Performance Computing & Simulation (HPCS), 2017 International Conference on*. [S.l.], 2017. p. 64–69. Citado 2 vezes nas páginas 49 e 103.

HAN, J. et al. Survey on nosql database. In: IEEE. *Pervasive computing and applications (ICPCA), 2011 6th international conference on*. [S.l.], 2011. p. 363–366. Citado na página 17.

HARIKUMAR, S.; REETHIMA, R.; KAIMAL, M. Semantic integration of heterogeneous relational schemas using multiple l1 linear regression and svd. In: IEEE. *Data Science & Engineering (ICDSE), 2014 International Conference on*. [S.l.], 2014. p. 105–111. Citado 2 vezes nas páginas 25 e 35.

HAVEL, J.-M. et al. Supporting terminological standardization in conceptual models-a plugin for a meta-modelling tool. 2014. Citado na página 24.

HONG, X.; RONG, C. Cloud data integration sharing and service. In: . Institute of Electrical and Electronics Engineers Inc., 2014. ISBN 9781479933518. Cited By 1. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908680796&doi=10.1109%2fWCCAIS.2014.6916636&partnerID=40&md5=ae1397eab8e63d6d32d8970f83142e7c>. Citado 2 vezes nas páginas 46 e 103.

HOUSSAINI, S. E.; BADRI, A. A web-based spatial decision support system for effective monitoring and routing problem. In: *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*. [S.l.: s.n.], 2012. p. 669–674. Citado 2 vezes nas páginas 46 e 103.

HÜNER, K. M.; OTTO, B.; ÖSTERLE, H. Collaborative management of business metadata. *International Journal of Information Management*, Elsevier, v. 31, n. 4, p. 366–373, 2011. Citado na página 28.

IEEE Xplore Digital Library. 2018. Online; accessed 20 January 2018. Disponível em: <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>. Citado na página 101.

JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: A tutorial. *Computer*, IEEE, v. 29, n. 3, p. 31–44, 1996. Citado na página 33.

KARPATHIOTAKIS, M. et al. No data left behind: real-time insights from a complex data ecosystem. In: ACM. *Proceedings of the 2017 Symposium on Cloud Computing*. [S.l.], 2017. p. 108–120. Citado 2 vezes nas páginas 49 e 103.

KEELE, S. et al. Guidelines for performing systematic literature reviews in software engineering. In: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. [S.l.]: sn, 2007. Citado na página 101.

KIMBALL, R.; ROSS, M. *The data warehouse toolkit: the complete guide to dimensional modeling*. [S.l.]: John Wiley & Sons, 2011. Citado na página 23.

KLYNE, G.; CARROLL, J. J. Resource description framework (rdf): Concepts and abstract syntax. 2006. Citado na página 29.

KOLAITIS, P. G. Schema mappings, data exchange, and metadata management. In: ACM. *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. [S.l.], 2005. p. 61–75. Citado na página 27.

KOLEV, B. et al. The cloudmdsql multistore system. In: . Association for Computing Machinery, 2016. v. 26-June-2016, p. 2113–2116. ISBN 9781450335317. ISSN 07308078. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84979697441&doi=10.1145%2f2882903.2899400&partnerID=40&md5=dc9799cd17f2de5ed6629166f21f1fe6>. Citado 2 vezes nas páginas 48 e 103.

LAWRENCE, R. Integration and virtualization of relational sql and nosql systems including mysql and mongodb. In: IEEE. *Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*. [S.l.], 2014. v. 1, p. 285–290. Citado na página 17.

LAWRENCE, R. Integration and virtualization of relational sql and nosql systems including mysql and mongodb. In: . [s.n.], 2014. v. 1, p. 285–290. Cited By 7. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84902689362&partnerID=40&md5=5ac8839ee96d04bf6c82d83fdd36647c>. Citado 2 vezes nas páginas 49 e 103.

LEE, S. et al. Table2graph: A scalable graph construction from relational tables using map-reduce. In: IEEE. *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*. [S.l.], 2015. p. 294–301. Citado na página 24.

LEE, S. et al. Table2graph: A scalable graph construction from relational tables using map-reduce. In: *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*. [S.l.: s.n.], 2015. p. 294–301. Citado 2 vezes nas páginas 45 e 103.

LI, W.-S.; CLIFTON, C. Semantic integration in heterogeneous databases using neural networks. In: *vldb*. [S.l.: s.n.], 1994. v. 94, p. 12–15. Citado na página 34.

LI, W.-S.; CLIFTON, C. Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering*, Elsevier, v. 33, n. 1, p. 49–84, 2000. Citado na página 53.

LOMOTHEY, R. K.; DETERS, R. Towards knowledge discovery in big data. In: IEEE. *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*. [S.l.], 2014. p. 181–191. Citado na página 18.

MA, K.; DONG, F.; YANG, B. Large-scale schema-free data deduplication approach with adaptive sliding window using mapreduce. *Computer Journal*, Oxford University Press, v. 58, n. 11, p. 3187–3201, 2014. ISSN 00104620. Cited By 1. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84947760436&doi=10.1093%2fcomjnl%2fbxv052&partnerID=40&md5=7a5dc805e684eb28e12dc5b90e3116ec>. Citado 2 vezes nas páginas 45 e 103.

MALHOTRA, S. et al. Data integration of cloud-based and relational databases. In: *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*. [S.l.: s.n.], 2015. p. 83–86. Citado 3 vezes nas páginas 44, 45 e 103.

MEHMOOD, N.; CULMONE, R.; MOSTARDA, L. Modeling temporal aspects of sensor data for mongodb nosql database. *Journal of Big Data*, SpringerOpen, v. 4, n. 1, 2017. ISSN 21961115. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85016764906&doi=10.1186%2fs40537-017-0068-5&partnerID=40&md5=78c55576beb5ff2b9fc52b63c767205c>. Citado 2 vezes nas páginas 47 e 103.

NDUME, V. et al. A methodology for data collection and integration of e-health records: A case study of ifakara health institute in tanzania. In: *Information Science, Computing and Telecommunications (PACT), 2013 Pan African International Conference on*. [S.l.: s.n.], 2013. p. 131–136. Citado 2 vezes nas páginas 43 e 103.

NGAN, C. K. Geo-data fusion integrator for object-oriented spatiotemporal olap cubes. In: *Computing for Geospatial Research and Application (COM.Geo), 2014 Fifth International Conference on*. [S.l.: s.n.], 2014. p. 51–53. Citado 2 vezes nas páginas 46 e 103.

OLIVEIRA, R. et al. Open-source based integration solution for hospitals. In: *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*. [S.l.: s.n.], 2016. p. 294–299. Citado 2 vezes nas páginas 47 e 103.

OXFORD Dictionaries. 2018. Online; accessed 20 August 2018. Disponível em: <https://www.oxfordlearnersdictionaries.com/definition/english/big-data>. Citado 2 vezes nas páginas 18 e 34.

PADHY, R. P.; PATRA, M. R.; SATAPATHY, S. C. Rdbms to nosql: reviewing some next-generation non-relational database's. *International Journal of Advanced Engineering Science and Technologies*, v. 11, n. 1, p. 15–30, 2011. Citado na página 17.

PINO, C.; RAVIDÁ, S.; SCIBILIA, S. Evaluation of federated database for distributed applications in e-government. In: *Application of Information and Communication Technologies (AICT), 2013 7th International Conference on*. [S.l.: s.n.], 2013. p. 1–5. Citado 3 vezes nas páginas 49, 51 e 103.

QIN, X. et al. Beyond simple integration of rdbms and mapreduce – paving the way toward a unified system for big data analytics: Vision and progress. In: *Cloud and Green Computing (CGC), 2012 Second International Conference on*. [S.l.: s.n.], 2012. p. 716–725. Citado 2 vezes nas páginas 44 e 103.

RAHM, E. The case for holistic data integration. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, v. 9809, p. 11–27, 2016. ISSN 03029743. Cited By 0; Conference of 20th East European Conference on Advances

in Databases and Information Systems, ADBIS 2016 ; Conference Date: 28 August 2016 Through 31 August 2016; Conference Code:179909. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84984904172&doi=10.1007%2f978-3-319-44039-2_2&partnerID=40&md5=0773686940cd8d37ec10e58175afd649. Citado 3 vezes nas páginas 18, 23 e 37.

REIS, J. C. D.; PRUSKI, C.; REYNAUD-DELAÎTRE, C. State-of-the-art on mapping maintenance and challenges towards a fully automatic approach. *Expert Systems with Applications*, Elsevier, v. 42, n. 3, p. 1465–1478, 2015. Citado na página 25.

RUI, C. et al. A domain web data standardization organization method. *Information Technology Journal*, v. 12, n. 22, p. 6710–6716, 2013. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84901713581&partnerID=40&md5=16ee5db3a3473b49ba39f3577b86c15f>. Citado 2 vezes nas páginas 43 e 103.

RUSSEL, S.; NORVIG, P. Inteligência artificial. *Editora Campus*, p. 26, 2004. Citado 4 vezes nas páginas 31, 32, 33 e 34.

SCHREINER, G.; DUARTE, D.; MELLO, R. D. S. Sqltokeynosql: A layer for relational to key-based nosql database mapping. In: . [s.n.], 2015. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84966738928&partnerID=40&md5=e9772bec2e3bfd30562d64c7c3c58954>. Citado 2 vezes nas páginas 47 e 103.

SCOPUS. 2018. Online; accessed 20 January 2018. Disponível em: <https://www.scopus.com/>. Citado na página 101.

SELLAMI, R.; BHIRI, S.; DEFUDE, B. Supporting multi data stores applications in cloud environments. *IEEE Transactions on Services Computing*, v. 9, n. 1, p. 59–71, 2016. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84962018357&partnerID=40&md5=da3b367e7a27cd64460c0c192410ceb5>. Citado 2 vezes nas páginas 48 e 103.

SERHANI, M. A. et al. An hybrid approach to quality evaluation across big data value chain. In: IEEE. *Big Data (BigData Congress), 2016 IEEE International Congress on*. [S.l.], 2016. p. 418–425. Citado na página 28.

SHETH, A. P.; LARSON, J. A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, ACM, v. 22, n. 3, p. 183–236, 1990. Citado na página 27.

SIDDIQA, A.; KARIM, A.; GANI, A. Big data storage technologies: a survey. *Frontiers of Information Technology & Electronic Engineering*, Springer, v. 18, n. 8, p. 1040–1070, 2017. Citado na página 38.

SMITH, M. et al. Dspace: An open source dynamic digital repository. Corporation for National Research Initiatives, 2003. Citado na página 28.

STOREY, V. C.; SONG, I.-Y. Big data technologies and management: What conceptual modeling can do. *Data & Knowledge Engineering*, Elsevier, v. 108, p. 50–67, 2017. Citado na página 17.

- TAKAHASHI, K. et al. Design and evaluation of lifelog mashup platform with nosql database. In: . [s.n.], 2013. p. 133–139. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84896846288&partnerID=40&md5=d1f996f1ba53d706ff282e33ff9e9d04>. Citado 2 vezes nas páginas 44 e 103.
- THE ACM Digital Library. 2018. Online; accessed 20 January 2018. Disponível em: <http://portal.acm.org/dl.cfm>. Citado na página 101.
- VATHY-FOGARASSY, Á.; HUGYÁK, T. Uniform data access platform for sql and nosql database systems. *Information Systems*, Elsevier, v. 69, p. 93–105, 2017. Citado 2 vezes nas páginas 47 e 103.
- WEIBEL, S. et al. *Dublin core metadata for resource discovery*. [S.l.], 1998. Citado na página 28.
- WIDROW, B.; RUMELHART, D. E.; LEHR, M. A. Neural networks: applications in industry, business and science. *Communications of the ACM*, ACM, v. 37, n. 3, p. 93–105, 1994. Citado na página 35.
- WINN, J. et al. Open data and the academy: an evaluation of ckan for research data management. 2013. Citado na página 28.
- WU, J. Heterogeneous data integration model based on virtual view. In: *Computer Science Education (ICCSE), 2012 7th International Conference on*. [S.l.: s.n.], 2012. p. 815–817. Citado 2 vezes nas páginas 46 e 103.
- XU, J. et al. Zql: A unified middleware bridging both relational and nosql databases. In: . Institute of Electrical and Electronics Engineers Inc., 2016. p. 730–737. ISBN 9781509040650. Cited By 0. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84995376700&doi=10.1109%2fDASC-PICom-DataCom-CyberSciTec.2016.129&partnerID=40&md5=462d1a1739a4a3d1ca912aca79563204>. Citado 2 vezes nas páginas 48 e 103.
- ZHANG, Y. et al. Semi-autonomous data enrichment based on cross-task labelling of missing targets for holistic speech analysis. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. [S.l.], 2016. p. 6090–6094. Citado na página 35.
- ZHAO, G. et al. Schema conversion model of sql database to nosql. In: *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014 Ninth International Conference on*. [S.l.: s.n.], 2014. p. 355–362. Citado 2 vezes nas páginas 44 e 103.
- ZHAO, H. Semantic matching across heterogeneous data sources. *Communications of the ACM*, ACM, v. 50, n. 1, p. 45–50, 2007. Citado na página 53.
- ZIEGLER, P.; DITTRICH, K. R. Three decades of data integration-all problems solved? In: SPRINGER. *IFIP congress topical sessions*. [S.l.], 2004. p. 3–12. Citado 2 vezes nas páginas 19 e 36.

Apêndices

.1 *Protocolo da revisão sistemática*

A revisão sistemática foi realizada seguindo a metodologia descrita em Keele et al. (2007). Para a condução da revisão sistemática foi utilizada a seguinte questão de pesquisa: Quais são as abordagens descritas para integração de dados representados com o uso de modelos heterogêneos?

Como fonte de informações para este trabalho foram selecionadas duas bibliotecas on-line, a The... (2018) e IEEE... (2018) e a base indexada Scopus (2018), que constituem referências para artigos na área da computação.

Neste trabalho foram utilizadas as seguintes palavras-chaves: Data Integration, Relational, SQL, Non Relational, NoSQL, Columnar Database, Document Database, Graph Database, Key Value Database, Object Database, Multimodel Database, Multidimensional Database, Cloud Database, XML Database, Big Data Integration, architecture, model, representation, proposal, evaluation, comparison, analysis, limitation.

A string de busca usada como referência foi elaborada utilizando as palavras chaves: (((Data Integration) AND ((Relational OR SQL) AND (Non Relational OR NonSQL OR Columnar Database OR Document Database OR Graph Database OR Key Value Database OR Object Database OR Multimodel Database OR Multidimensional Database OR cloud Database OR XML Database)))) OR ((Big AND data AND integration) AND ((SQL AND nosql) OR (relational AND no AND relational)) AND (architecture OR model OR representation OR proposal OR evaluation OR comparison OR analysis OR limitation)))

A pesquisa nas bases com período anterior a 2012 não gerou resultados significativos, ou seja, o retorno de artigos para a string de busca foi baixo, totalizando menos de 20 artigos, que posteriormente são excluídos pelos critérios de exclusão utilizados neste trabalho. Devido a esse resultado de busca, a pesquisa focou nos trabalhos publicados a partir de 2012.

Conforme apresentado na figura 39 foram realizadas buscas nas bibliotecas e bases indexadas citadas e como resultado foram obtidos 881 artigos. Foram excluídos 77 artigos duplicados e após a aplicação dos critérios de inclusão e exclusão descritos na tabela 8, foram selecionados 35 artigos para leitura detalhada.

Figura 39 – Diagrama do processo de seleção de busca

Fonte: Keylla Ramos Saes, 2018

Tabela 8 – Critérios de inclusão e exclusão

Critério	Descrição
(I1)	Trabalhos relacionados a integração de dados estruturados e não estruturados.
(I2)	Trabalhos que apresentem métodos de integração com modelos de representação de dados distintos.
(I3)	Trabalhos relacionados a integração no escopo de Big Data.
(E1)	Trabalhos não abordam ou descrevem integração de dados.
(E2)	Trabalhos que realizem integração apenas de dados não estruturados.
(E3)	Trabalhos que realizem integração apenas de dados estruturados.
(E4)	Trabalhos que abordem integração somente com uso de ontologia.

Fonte: Keylla Ramos Saes, 2018

.2 Artigos detalhados da revisão sistemática

Tabela 9 – Extração de informação de artigos relacionados às arquiteturas de integração entre dados estruturados e não estruturados

Arquitetura	Modelo Representação	Sincronismo	Ano Pub.	Artigo
Customizada	Documento	Síncrono	2013	(RUI et al., 2013)
Customizada	Documento	Assíncrono	2013	(NDUME et al., 2013)
Customizada	Documento	Assíncrono	2013	(TAKAHASHI et al., 2013)
Customizada	Grafos	Síncrono	2013	(GRUND et al., 2013)
Customizada	Multi Modelo	Síncrono	2013	(BENKER, 2013)
Customizada	Multi Modelo	Síncrono	2013	(DOSHI et al., 2013)
Customizada	Grafos	Síncrono	2014	(ZHAO et al., 2014)
Componentes Geográficos	Geográfico	Síncrono	2012	(HOUSSAINI; BADRI, 2012)
Componentes Geográficos	Geográfico	Síncrono	2014	(FANG et al., 2014)
Componentes Geográficos	Multi Modelo	Assíncrono	2014	(NGAN, 2014)
Map Reduce	Multi Modelo	Assíncrono	2012	(QIN et al., 2012)
Map Reduce	Multi Modelo	Assíncrono	2014	(MA; DONG; YANG, 2014)
Map Reduce	Documento	Assíncrono	2015	(MALHOTRA et al., 2015)
Map Reduce	Grafos	Assíncrono	2015	(LEE et al., 2015b)
Middleware ou serviços de dados	Documento	Síncrono	2012	(WU, 2012)
Middleware ou serviços de dados	Documento	Síncrono	2014	(HONG; RONG, 2014)
Middleware ou serviços de dados	Documento	Síncrono	2015	(CHEN et al., 2015)
Middleware ou serviços de dados	Grafos	Síncrono	2015	(DEY et al., 2015)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2015	(GARG; KAUR, 2015)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2015	(SCHREINER; DUARTE; MELLO, 2015)
Middleware ou serviços de dados	Documento	Síncrono	2016	(OLIVEIRA et al., 2016)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2016	(CHANG; TSAI; WANG, 2016)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2016	(KOLEV et al., 2016)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2016	(XU et al., 2016)
Middleware ou serviços de dados	Documento	Síncrono	2017	(MEHMOOD; CULMONE; MOSTARDA, 2017)
Middleware ou serviços de dados	Grafos	Síncrono	2017	(VATHY-FOGARASSY; HUGYÁK, 2017)
Middleware ou serviços de dados	Multi Modelo	Síncrono	2017	(GIANNAKOURIS et al., 2016)
Federação ou virtualização de dados	Grafos	Síncrono	2013	(PINO; RAVIDÁ; SCIBILIA, 2013)
Federação ou virtualização de dados	Documento	Síncrono	2014	(LAWRENCE, 2014b)
Federação ou virtualização de dados	Multi Modelo	Síncrono	2014	(GROGER; SCHWARZ; MITSCHANG, 2014)
Federação ou virtualização de dados	Grafos	Síncrono	2015	(GADEPALLY et al., 2015)
Federação ou virtualização de dados	Documento	Síncrono	2016	(SELLAMI; BHIRI; DEFUDE, 2016)
Federação ou virtualização de dados	Grafos	Síncrono	2017	(ABOLHASSANI et al., 2017)
Federação ou virtualização de dados	Multi Modelo	Síncrono	2017	(KARPATHIOTAKIS et al., 2017)
Federação ou virtualização de dados	Multi Modelo	Síncrono	2017	(GUERRA et al., 2017)

Fonte: Keylla Ramos Saes, 2018