Universidade do Minho
Escola de Engenharia

José Fernando Pereira Magalhães

Abordagem Semântica para a Integração de Dados em *Big Data Warehouses*

Dissertação de Mestrado

Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação da

Professora Doutora Maribel Yasmina Santos

Outubro de 2019

"The important thing is not to stop questioning. Curiosity has its own reason for existence."

Albert Einstein

AGRADECIMENTOS

Uma longa caminhada aproxima-se do fim e a saudades já se intensificam, um misto de sentimentos que se divide entre o objetivo que foi alcançado e a nostalgia. Parece que foi ontem o dia da primeira matrícula e quando dou de conta estou a terminar a dissertação. Foram ultrapassados grandes obstáculos que me fizeram crescer, definir objetivos e mudar a minha visão sobre muitos contextos. Começo por dedicar este trabalho à minha avó paterna, que infelizmente não se encontra presente fisicamente para me observar a alcançar este objetivo, mas que sempre me acompanhou e me deu força para que o alcançasse.

Aos meus pais, que são um exemplo a seguir, pelo esforço demonstrado todos os dias e por me fazerem ver que é necessário resiliência e dedicação para alcançar os nossos objetivos. À minha irmã, que me enche de orgulho vê-la a fazer o seu percurso com objetivo de ser e fazer melhor do que eu consegui. Não tenho dúvidas que conseguirá. Aproveito para agradecer à restante família, aos meus avós e tios que sempre acompanharam o meu percurso. Agradeço todos os dias por pertencer a esta família!

À Inês, a pessoa mais importante que a universidade me deu a oportunidade de conhecer, por seres a pessoa que me acompanhou em todas as batalhas, por todos os ensinamentos, por me ajudares a ser melhor a cada dia que passa, por tornares todos os momentos menos bons em momentos de grande felicidade. Foram sem dúvida 5 anos de aprendizagem contigo que me proporcionaram os melhores anos que vivi. É um orgulho olhar para trás e ver o teu percurso!

À professora Maribel Yasmina Santos, um dos principais motivos por abraçar este desafio, que sempre se mostrou disponível e durante este percurso se revelou fundamental. Pelo exemplo de profissional que é, pela confiança depositada em mim, pelas oportunidades que me proporcionou e por todas as ajudas e sugestões.

Ao pessoal do lid4, em particular ao João, ao Carlos e à Carina que desde o primeiro momento me ajudaram nas minhas dificuldades e essencialmente pelo espírito de equipa e boa disposição partilhado. Ao João, por todo o tempo despendido, por toda a sabedoria e discussões partilhadas.

Por fim, agradecer a toda a gente que direta ou indiretamente contribuiu para este percurso, a todos os meus amigos e colegas que tiveram que lidar com o meu feitio, pelas noites de estudo, pelas noites reféns passadas no buraco, pelos jantares e almoços todos juntos. Obrigado.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

Big Data não é um domínio trivial, tanto ao nível de investigação, como de desenvolvimento. Atualmente, o volume de dados produzido tem aumentado exponencialmente devido à utilização de dispositivos como, por exemplo, *smartphones*, *tablets,* dispositivos inteligentes e sensores. Esta proliferação de dados que se apresentam em formatos estruturados, semiestruturados e não estruturados foi acompanhada pela popularidade do conceito de *Big Data*, que pode ser caracterizado como o volume, velocidade e variedade que os dados apresentam e que não conseguem ser processados, armazenados e analisados através de ferramentas e métodos tradicionais. As organizações, inseridas em ambientes altamente competitivos, visam a obtenção de vantagens competitivas perante os seus concorrentes, comprometendo-se a extrair o maior valor das tecnologias com o objetivo de melhorar a sua tomada de decisão. A título de exemplo, os *Data Warehouses* surgem como componentes centrais no armazenamento de dados, no entanto, estes repositórios de dados regem-se por modelos relacionais que os impossibilita de responder às exigências de *Big Data*. Consequentemente, surge a necessidade da adoção de novas tecnologias e modelos lógicos capazes de colmatar os desafios de *Big Data*, originando assim os *Big Data Warehouses,* que utilizados em tecnologias como Hadoop ou bases de dados NoSQL garantem uma maior flexibilidade e escalabilidade na manipulação de dados em contextos *Big Data*. A dimensão do *Big Data Warehouse* conduz a um acréscimo de complexidade nos domínios de Governança de Dados e *Data Quality* devido ao grande volume de dados que é continuamente armazenado. Contudo, inserido num domínio intrínseco a *Data Quality, Data Profiling* vem colmatar alguns destes desafios através da produção de metadados sobre os conjuntos de dados que chegam ao *Big Data Warehouse,* ganhando assim uma nova importância na integração entre as novas fontes de dados e os dados que já subsistem no *Big Data Warehouse.* Desta forma, o principal objetivo deste trabalho é propor, desenvolver e validar uma ferramenta de *Data Profiling* que permita inspecionar novas fontes de dados, derivando e armazenando informação relevante para a sua integração no *Big Data Warehouse.*

Palavras-Chave

Big Data Warehouse, *Data Profiling*, Governança de Dados, Integração, Metadados

Abstract

Big Data is not a trivial domain regarding the research and development topic. Currently, the amount of data produced has increased due to the use of gadgets such as smartphones, tablets, smart devices, and sensors. Bearing that in mind, the proliferation of data presented in structured, semi-structured and unstructured formats was accompanied by the popularity of the Big Data concept that can be characterized by volume, velocity, and variety of data which cannot be processed, stored and analyzed through traditional tools. The organizations inserted in highly competitive environments aim to obtain competitive advantages over their competitors, committing themselves to extract the highest value of the technologies in order to improve their decision making. For example, Data Warehouses appear as central components in data storage supported by rigid models. However, these data repositories can no longer answer the high demand of Big Data reality. Therefore, there is the need to adopt new technologies and logical models capable of solving Big Data challenges, giving the rise to Big Data Warehouses which are used in technologies such as Hadoop or NoSQL databases to ensure higher flexibility and scalability in data manipulation in Big Data contexts. The Big Data Warehouse size leads to an increase in the complexity concerning the domains of Data Governance and Data Quality, due to the high volume of data that is continuously stored. Nevertheless, embedded in the Data Quality domain, Data Profiling approach solves some of these challenges producing metadata about datasets which are being sent to the Big Data Warehouse, raising awareness to the relevance of the integration between new data sources and data which is already stored in the Big Data Warehouse. Considering all information exposed, the main purpose of this work is to propose, develop and validate a Data Profiling tool that allows inspecting new data sources, storing and deriving relevant information to its integration in Big Data Warehouse.

Keywords

Big Data Warehouse, Data Governance, Data Profiling, Integration, Metadata.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

Lista de Abreviaturas, Siglas e Acrónimos

Este documento usa uma lista de siglas ou acrónimos listada de seguida:

ACID – Atomicity, Consistency, Isolation, Durability

API – Application Programming Interface

BASE – Basically Available, Soft state, Eventual consistency

BDW – Big Data Warehouse

BI – Business Intelligence

CAP – Consistency, Availability, Partition tolerance

CRM – Customer Relationship Management

CRUD – Create, Read, Update, Delete

DBLP – Digital Bibliography & Library Project

DSRM – Design Science Research Methodology

DW – Data Warehouse

ERP – Enterprise Resource Planning

ETL – Extract Transform and Load

GFS – Google File System

HDFS – Hadoop Distributed File System

HiveQL – Hive Query Language

HOLAP – Hybrid OLAP

JSON – JavaScript Object Notation

MOLAP – Multidimensional OLAP

NoSQL – Not Only SQL

OLAP – Online Analytical Processing

OLTP – Online Transaction Processing

RAM – Random Acess Memory

RDBS – Relational Database Systems

ROLAP – Relational OLAP

SCD – Slowly Changing Dimension

SQL – Structured Query Language

XML – Extensible Markup Language

1. INTRODUÇÃO

Este capítulo tem como objetivo apresentar o enquadramento e a motivação do tema da dissertação, os objetivos e resultados esperados no seu desenvolvimento e a abordagem de investigação selecionada. Encontra-se também o planeamento das tarefas e a organização do documento.

1.1. Enquadramento e Motivação

Com os progressos na recolha, armazenamento e processamento de dados e tendo em conta o surgimento de novas fontes de dados, nomeadamente, redes sociais, *smartphones*, *tablets*, *cloud computing* e sensores, o volume de dados recolhidos pelas organizações tem aumentado progressivamente (Cassavia, Dicosta, Masciari, & Saccà, 2014; Dumbill, 2012).

Esta elevada taxa de produção de grandes volumes de dados, gerados a diferentes velocidades e variedades, está associada a um fenómeno caracterizado por *Big Data* que tipicamente é definido como "os dados cujo volume, variedade e velocidade impõem desafios na utilização das tecnologias e modelos tradicionais" (Krishnan, 2013).

Os Data Warehouses (DWs) surgem como componentes centrais no armazenamento de dados, no entanto, estes repositórios de dados regem-se por modelos relacionais bem definidos, que os impossibilita de responder às exigências de *Big Data*. À medida em que a dimensão do *Data Warehouse* evolui, a sua estrutura torna-se mais complexa e menos uniforme devido à sobrecarga imposta pelo grande número de relações entre trabelas, que se traduz no aumento do número de operações de *join* que influenciam o seu desempenho e dificultam a evolução do *Data Warehouse* em resposta às necessidades dos negócios em constante mudança (Baton & Bruggen, 2017; Robinson, Webber, & Eifrem, 2013).

Perante o que foi exposto, *os Big Data Warehouses (BDWs)* surgem como peças centrais para o armazenamento adequado de grandes volumes de dados, suportados por modelos de dados e tecnologias bem definidas, que facilitam o processamento e análise de dados sob diferentes perspetivas (Costa & Santos, 2016). Estes sistemas caracterizam-se como sendo sistemas escaláveis, de alta performance e capazes de lidar com o aumento de volume, variedade e velocidade dos dados, permitindo a extração e produção de informação, com a finalidade de ser utilizada para melhorar os processos de tomada de decisão (Tria, Lefons, & Tangorra, 2014).

Surge assim a necessidade do desenvolvimento de aplicações para o uso intensivo de dados altamente disponíveis e escaláveis. É neste sentido que surge o Hadoop[1], um *framework open-source* baseado em *MapReduce* e HDFS *(Hadoop Distributed File System),* utilizado para armazenar, processar e analisar grandes volumes de dados (Thusoo et al. 2010). Associado ao ecossistema Hadoop, surge o Hive[2], um repositório para a implementação de DW sobre o Hadoop (E. Costa, Costa, & Santos, 2018).

Em suma, considerando os conceitos e tecnologias mencionadas anteriormente, a principal motivação para a realização desta dissertação centra-se na necessidade de facilitar a evolução do *Big Data Warehouse* através da integração de novas fontes de dados. Para que esta integração seja possível, é necessário inspecionar e avaliar as novas fontes de dados que chegam ao *Big Data Warehouse* através de um conjunto de indicadores para avaliar a afinidade, similaridade e *joinability* entre os conjuntos de dados que chegam ao *Big Data Warehouse* e os que já subsistem no mesmo. Desta forma, é necessário que sejam conhecidos os requisitos e as tecnologias que permitam o armazenamento da informação (metadados) resultante de processos de *Data Profiling*[3] e compreender a relevância que estes processos proporcionam na evolução de uma temática recente na comunidade.

1.2. Objetivos e Resultados Esperados

Com esta dissertação pretende-se propor, implementar e desenvolver uma ferramenta de *Data Profiling* que permita inspecionar novas fontes de dados, derivando informação relevante para a sua integração no *Big Data Warehouse.*

Atendo à finalidade e aos objetivos definidos, espera-se em termos de resultados:

- Sistematizar o estado da arte, no que diz respeito a abordagens de implementação de ferramentas de *Data Profiling,* Governança de Dados e algoritmos de similaridade;
- Contextualização das tecnologias que permitam a implementação da ferramenta proposta anteriormente;
- Identificação e desenvolvimento de um conjunto de rotinas com o objetivo de caracterizar a distribuição e qualidade dos atributos dos novos conjuntos de dados e dos dados já existentes no *Big Data Warehouse*;

[1] https://hadoop.apache.org/
[2] https://hive.apache.org/
[3] Este termo encontra-se abordado na subsecção 2.3.1.

- Avaliação do desempenho das diferentes medidas de similaridade com a perspetiva de integração na ferramenta de *Data Profiling;*
- Identificação e desenvolvimento de um conjunto de rotinas com o objetivo de analisar a similaridade entre novas fontes de dados e os dados já existentes num *Big Data Warehouse,* utilizando as medidas de similaridade;
- Proposta e implementação de uma arquitetura de Governança de Dados baseada em grafos para armazenar e gerir toda a informação preveniente dos objetivos anteriores.

1.3. Abordagem de Investigação

Para o desenvolvimento desta dissertação, foi selecionada uma metodologia de investigação para o desenvolvimento da investigação científica que, entre outros, orienta o processo de revisão bibliográfica na exploração dos conceitos e na descoberta dos trabalhos inerentes ao tema da dissertação.

1.3.1. Metodologia de Investigação

Para a elaboração da presente dissertação, será utilizada uma metodologia de investigação enquadrada na área de sistemas de informação, *Design Science Research Methodology (DSRM) for Information Systems*. Esta metodologia encontra-se dividida em seis fases, nomeadamente, a identificação do problema e motivação, a definição de objetivos para a solução, a conceção e desenvolvimento, a demonstração, a avaliação e, por fim, a comunicação (Peffers, Tuunanem, Rothenberger, & Chatterjee et al. 2007). Na Figura 1, apresenta-se cada uma das fases mencionadas anteriormente.

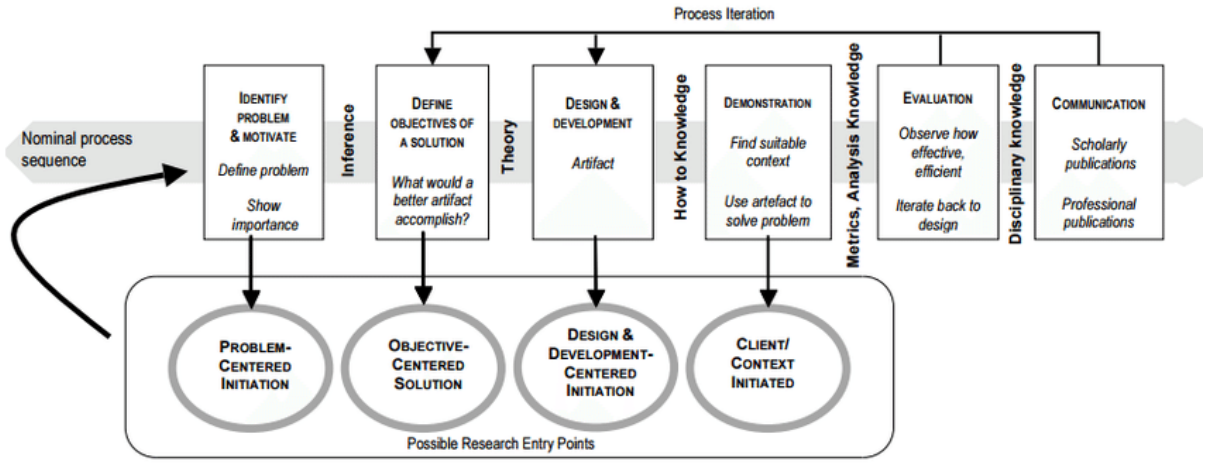

Figura 1. Design science research methodology for information systems. Adaptado de (Peffers et al. 2007).

Assumindo a abordagem metodológica de investigação selecionada e as datas de entregas intermédias exigidas pela instituição de ensino, surgem as seguintes tarefas a desenvolver:

1. **Identificação do problema e motivação:** definir especificamente o problema e justificar o valor que a solução final poderá apresentar no enquadramento da dissertação. Esta fase inclui também a criação de um documento com o resumo e enquadramento da dissertação, assim como a apresentação dos principais objetivos, identificação da abordagem de investigação e calendarização das tarefas;
2. **Definir objetivos da solução:** para a definição dos objetivos é necessário obter conhecimentos dos métodos e tecnologias do domínio em questão. Assim, é iniciado o processo de revisão de literatura, que conduz ao levantamento do "estado da arte", que se trata de identificar, analisar e compreender os conceitos e técnicas já existentes que se relacionam com o tema da dissertação. Durante a elaboração da revisão do estado da arte, é efetuado o enquadramento tecnológico para a aquisição de conhecimento sobre cada uma das tecnologias e a sua utilidade no contexto desta dissertação;
3. **Conceção e Desenvolvimento:** aquando conhecidas as tecnologias, inicia-se a criação de um artefacto que pode conter modelos, métodos e novas propriedades técnicas sobre os recursos utilizados;
4. **Demonstração:** consiste na implementação e validação da ferramenta proposta. Nesta fase, é aprovado se o que foi desenvolvido é, ou não, capaz de resolver os problemas apresentados;
5. **Avaliação:** observar e avaliar o resultado da demonstração executada. Esta tarefa envolve a comparação dos resultados propostos para a solução com os resultados atuais observados na fase da demonstração;
6. **Comunicação:** comunicar e apresentar o problema, a sua importância, a sua utilidade e o rigor da sua conceção a profissionais da área. Esta fase inclui a escrita e apresentação da dissertação, assim como a possibilidade da publicação de artigos relacionados em conferências/jornais da área.

1.3.2. Processo de Revisão de Literatura

Para a concretização do trabalho proposto nesta dissertação foi necessário proceder à criação da revisão de literatura. Nesta fase inicial são definidos os critérios de pesquisa, fontes de pesquisa e palavras-chave que são os principais elementos para a pesquisa e a recolha de documentos.

Para a pesquisa de documentos definiu-se as bases de dados de referência, nomeadamente, IEEE Xplore, Web of Science, Scopus, dblp *(Digital Bibliography & Library Project)* e Google Scholar, nas quais foram aplicadas um conjunto de palavras-chave para a realização da pesquisa.

A identificação e seleção da literatura ocorreu entre outubro de 2018 e janeiro de 2019, salientando também que as palavras-chave definidas surgem em prol do tema da dissertação e foram *Big Data*, *Data Warehouse*, *Big Data Warehouse*, *Data* Evolution, *Data Similarity*, *Joinability*, *Affinity*, *Data Profiling* e *Data Integration*, tendo sido considerado o cruzamento entre as várias palavras referidas nas respetivas bases de dados.

Deste modo, a Figura 2 retrata o processo de tomada de decisão adotado com o propósito de selecionar a literatura relevante para fundamentar esta investigação. Resumidamente, no fluxograma da Figura 2 a literatura é classificada em três categorias:

- **Leitura relevante** – realizar uma análise cuidada;
- **Leitura potencialmente relevante** – realizar uma leitura com baixo grau de profundidade, com o propósito de perceber se a documentação se revela importante para a investigação;
- **Leitura não relevante** – a documentação não se revela útil para esta investigação.

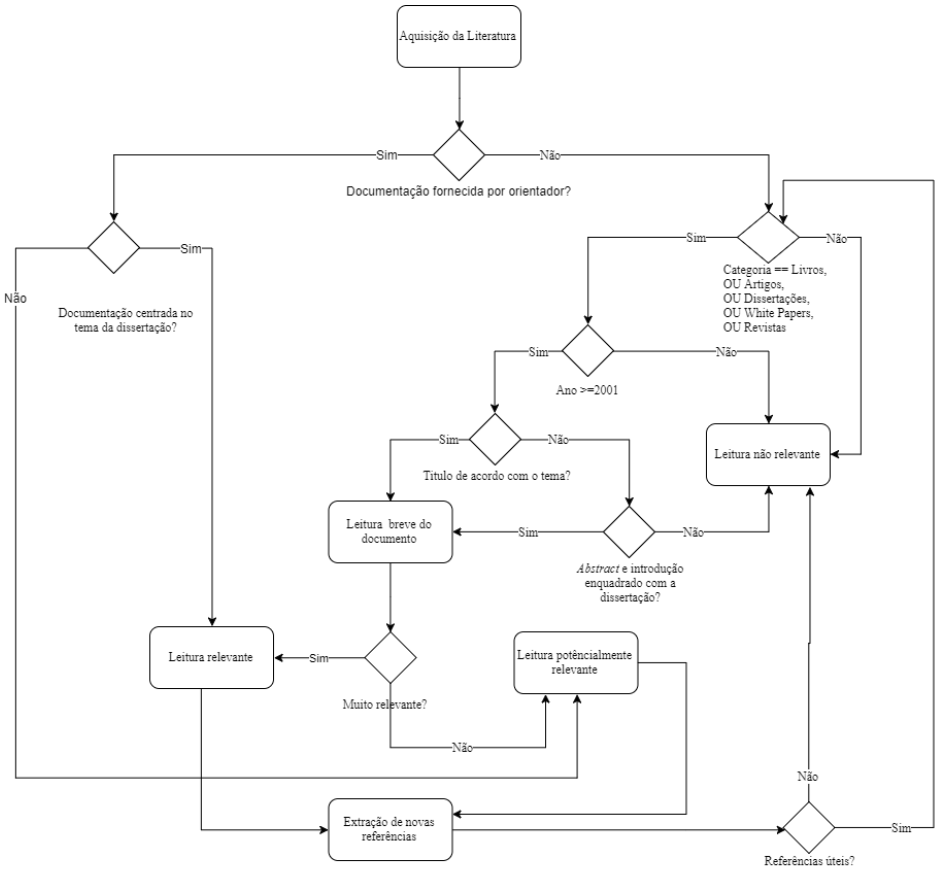

Figura 2. Processo de seleção de literatura.

1.4. Organização do Documento

Esta dissertação está organizada em seis capítulos que de seguida serão sumariamente apresentados.

O primeiro capítulo diz respeito à introdução, na qual se apresenta o enquadramento, a motivação, a finalidade, os objetivos, a abordagem de investigação e o plano de trabalho.

O segundo capítulo é referente ao enquadramento concetual, no qual se apresentam os conceitos associados ao fenómeno *Big Data* (características, oportunidades, problemas e desafios) e a comparação entre várias abordagens de armazenamento, desde as abordagens tradicionais às abordagens capazes de suportar fenómenos *Big Data*, terminando com alguns trabalhos relacionados com *Data Profiling* e Governança de Dados [4] em contextos *Big Data*.

O terceiro capítulo é referente ao enquadramento tecnológico, focando essencialmente em tecnologias inerentes ao ecossistema Hadoop, como o Hive, Spark e Ranger abordando também a ferramenta de armazenamento de metadados Atlas assim como também a base de dados inerente à mesma.

O quarto capítulo apresenta as infraestruturas utilizadas nos testes das medidas de similaridade, os conjuntos de dados utilizados, seguindo-se de uma explicação sobre as medidas de similaridade implementadas e que pretendem ser avaliadas e, por fim, o protocolo e preparação para a execução dos testes. Posteriormente à realização dos testes, as medidas de similaridade selecionadas são testadas num caso real no âmbito do genético resultando esse trabalho num grafo de similaridade entre os conjuntos de dados. A execução dos pontos anteriores resulta na proposta do algoritmo utilizado para as duas dimensões de avaliação (*headers* e conteúdo de dados).

O quinto capítulo resulta dos resultados obtidos do quarto capítulo que essencialmente apresenta a proposta e implementação da arquitetura de Governança de Dados para a integração de dados e, como tal, é apresentado o conjunto de dados utilizado para a implementação e contextualização da arquitetura.

O sexto e último capítulo apresenta as conclusões a retirar do trabalho realizado seguindo-se das dificuldades e limitações apresentadas ao longo da dissertação e propõe o trabalho futuro associado ao tema em questão.

Por fim, o documento termina com as referências bibliográficas e apêndices que suportam a dissertação.

---

[4] Este termo encontra-se abordado na secção 2.3.

2. *BIG DATA WAREHOUSES*

De forma a enquadrar os objetivos da presente dissertação, neste capítulo estão apresentados os principais conceitos associados às diversas temáticas abordadas neste documento. Posto isto, será apresentado o conceito de *Big Data* e as suas características, oportunidades e problemas inerentes. De modo a dar continuidade à caracterização deste fenómeno, é efetuada uma descrição das diversas bases de dados, tradicionais, NoSQL e New SQL. Também na linha do armazenamento, é realizada uma comparação entre os *Data Warehouses* tradicionais e os *Big Data Warehouses*, terminando com uma abordagem aos vários modelos de dados. Por fim, aborda-se a importância da presença da Governança de Dados em *Big Data Warehouses,* destacando a importância das tarefas e processos de *Data Profiling* para a aquisição e produção de metadados. Apresentam-se também os trabalhos relacionados com esta dissertação e termina-se com a apresentação do mapa concetual, que sistematiza os principais conceitos desta dissertação.

2.1. *Big Data*

Ao longo dos últimos anos, o volume de dados tem aumentado em grande escala devido a um conjunto de fatores, nomeadamente, ao aumento das fontes de dados e de mecanismos de recolha de dados que, consequentemente, produzem grandes volumes de dados (Chen, Mao, & Liu, 2014; S. Fan, Lau, & Zhao, 2015). Os dados armazenados apresentam-se cada vez mais relevantes e, no futuro, serão um componente essencial para o sucesso das organizações (Kubina, Varmus, & Kubinova, 2015). Em determinados contextos, nomeadamente na Indústria 4.0, a análise de dados cuja variedade e volume são elevados, é vista como um dos aspetos mais importantes para produzir valor para as organizações (Costa & Santos, 2017; Kagermann 2013).

Contudo, o volume, a velocidade e a heterogeneidade dos dados têm imposto grandes desafios às tecnologias tradicionais, nomeadamente na sua aquisição, gestão e processamento de dados, não conseguindo ser suficientemente eficientes e eficazes no seu tempo de resposta (Acharjya, 2016; Krishnan, 2013).

As decisões que anteriormente eram baseadas em pressupostos, são agora tomadas com base na informação proveniente dos dados, sendo um dos principais desafios das organizações a necessidade de obter a informação relevante no tempo certo (Kubina et al. 2015). Perante isso, devido ao facto de as tecnologias tradicionais apresentarem grandes desafios em relação a fatores como tempo de resposta,

escalabilidade e desempenho, surge a necessidade de apostar em tecnologias e soluções baseadas em conceitos de *Big Data* (Oussous, Benjelloun, Ait Lahcen, & Belfkih, 2018).

Recentemente, o estudo realizado pela consultora Excelacom, presente na Figura 3, ilustra os dados produzidos por diversas fontes em apenas um minuto. Como se pode observar, para além do aumento do volume de dados nas diversas plataformas, destaca-se também o potencial da informação que pode ser extraída e analisada, de forma a tornar o processo de decisão de uma organização mais eficaz e eficiente (Kubina et al. 2015).

Big Data é uma temática que acarreta inúmeros desafios, não só na ambiguidade relativamente à sua definição, modelos e arquiteturas, mas também nos desafios relacionados com o ciclo de vida de *Big Data*. Apesar do termo ser associado a grandes volumes de dados, está longe de ser essa a sua única característica ou o seu único desafio (Zikopoulos, 2011; Costa & Santos, 2017; J. Fan, Han, & Liu, 2014).

O termo *Big Data* começou a ser utilizado em finais de 1970 como um sistema base de dados paralelo, tendo como objetivo responder ao aumento do volume de dados (Chen et al. 2014). Posteriormente, em 2002, surgem tecnologias associadas aos conceitos *Big Data*, nomeadamente o GFS *(Google File System)* e o modelo de programação *MapReduce*. Em 2006, a Apache lançou um dos seus principais projetos *open-source*, adotado pela Yahoo, chamado de Hadoop (Krishnan, 2013).

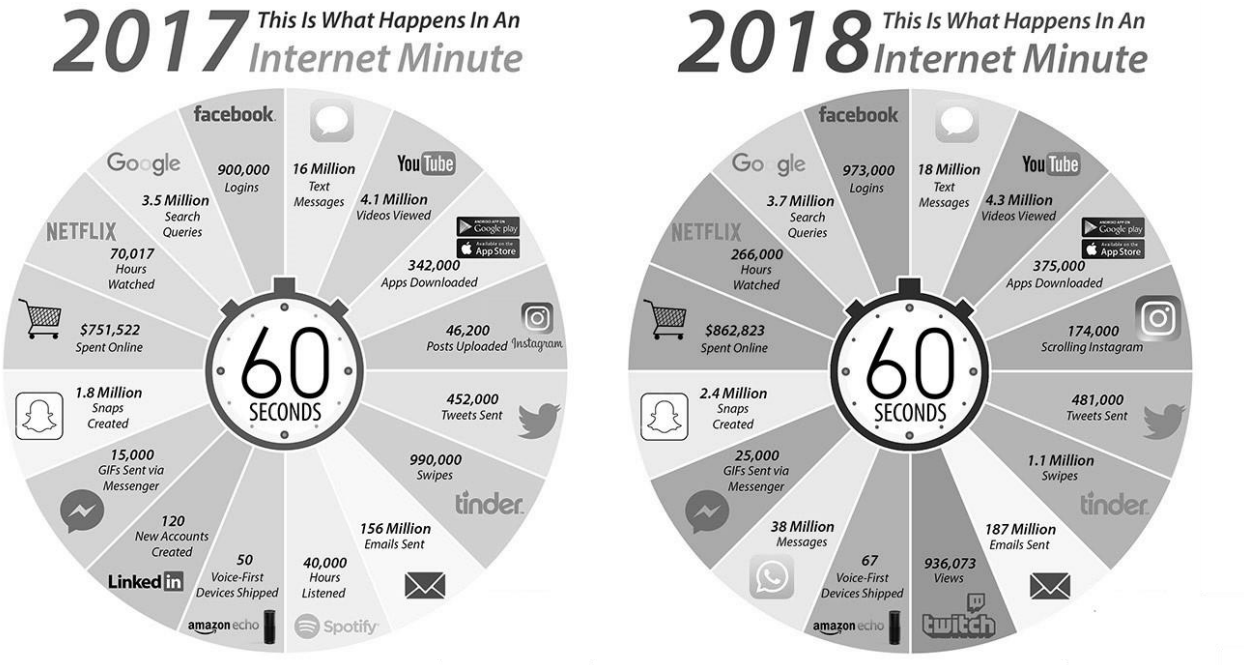

Figura 3. Taxa de crescimento do volume de dados. Fonte: Excelacom.com.

2.1.1. O Conceito

Segundo Krishnan (2013), o maior fenómeno que captou a atenção de organizações de grande dimensão desde o surgimento da *Internet* foi *Big Data*. Contudo, apesar da sua importância ser devidamente reconhecida, o mesmo não acontece quando se trata da sua definição, geralmente envolta em ambiguidade em estabelecer um limite na classificação e quantificação de "grandes volumes de dados" (Chen et al. 2014).

Assim, as definições dos vários autores divergem pela forma como são apresentadas, já que alguns autores definem *Big Data* pelas suas características, outros baseiam-se na argumentação do aumento das fontes de dados tradicionais com adição de dados não estruturados e outros através da quantificação de *Big Data* (Ward & Barker, 2013).

Zikopoulos (2011) destaca a importância do *Big Data* nas organizações que atualmente têm acesso a um vasto conjunto de dados, mas não têm capacidade de extrair valor dos mesmos, devido maioritariamente ao seu formato semiestruturado ou não estruturado. Perante isso, os autores definem *Big Data* como os dados que não conseguem ser processados ou analisados através de processos ou ferramentas tradicionais.

A ótica de Dumbill (2012) nesta temática vai ao encontro de Zikopoulos (2011), afirmando que *Big Data* se refere aos dados que excedem a capacidade de processamento das base de dados tradicionais. O autor complementa a sua afirmação acrescentando que o volume de dados é muito vasto, gerado a grandes velocidades e não está adequado à arquitetura das bases de dados tradicionais. Portanto, é necessário selecionar métodos e técnicas alternativas para processar e analisar estes dados com o objetivo de extrair valor dos mesmos.

Em 2013, Krishnan (2013) definiu *Big Data* como os vastos volumes de dados disponíveis em vários níveis de complexidade, gerados a diferentes velocidades e variedades que não conseguem ser processados utilizando tecnologias, métodos de processamento e algoritmos tradicionais. *Big Data* permite o acesso a grandes volumes de dados que podem ser úteis para obter padrões e tendências implícitas nos mesmos, sendo uma oportunidade para as organizações aumentarem a sua eficácia e eficiência ao nível dos processos de negócio (Costa & Santos, 2017; Henning, 2013).

No seu trabalho, Ward e Barker (2013) reconhecem o crescimento exponencial da área nos últimos anos, alertando para a falta de rigor na associação de *Big Data* ao armazenamento e análise de dados. Os autores reconhecem que a falta de rigor se deve maioritariamente à dificuldade em quantificar *Big Data*, destacando a Intel como uma das poucas organizações que define *Big Data* pela sua

quantificação, associando *Big Data* a uma organização que semanalmente produz em média 300 *terabytes (TB)* de dados. Contudo, Ward e Barker (2013) argumentam que definir *Big Data* pela inadequação das tecnologias tradicionais em processar grandes volumes de dados pode ser relativamente perigoso, devido aos avanços constantes das tecnologias, particularmente, os computadores quânticos. Por fim, os autores concluem que as definições de *Big Data* incluem pelo menos um dos seguintes aspetos: volume, complexidade e tecnologia. Em contraste, os autores acabam mesmo por incluir na sua própria definição de *Big Data*, o armazenamento e a análise de grandes e complexos conjuntos de dados utilizando conjuntos de técnicas e método adequados.

Não somente investigadores, mas também organizações, têm um ponto de vista nesta temática. Em 2010, a Apache Hadoop reconheceu *Big Data* como o conjunto de dados que não consegue ser extraído e processado através de recursos computacionais convencionais. Em conformidade, a Oracle centraliza a sua definição ao nível da infraestrutura, afirmando que *Big Data* deriva do valor extraído das bases de dados relacionais com informação sobre o negócio, ou seja, informação sobre as transações provenientes dos ERP's (*Enterprise Resource Planning)* com a adição de novas fontes de dados não estruturados (Ward & Barker, 2013).

Em síntese, existem várias opiniões distintas acerca da temática, contudo, é notável que algumas delas convergem em alguns aspetos. Como tal, De Mauro, Greco e Grimaldi (2015), perante a ambiguidade do tema, realizaram um estudo com o objetivo de propor uma definição formal de *Big Data*, baseada nas suas características e em coerência com as definições normalmente utilizadas. Em resultado, De Mauro (2015) define *Big Data* pelo seu volume, velocidade e variedade de dados, motivando a aplicação de métodos e tecnologias específicas, com o propósito de criar valor para a sociedade e para as organizações. Na perspetiva do autor deste documento, *Big Data* encontra-se associado às três principais características, volume, variedade e velocidade, que excedem a capacidade na extração, enriquecimento, transformação e armazenamento de dados em ferramentas, arquiteturas e modelos tradicionais. Algumas características atrás mencionadas não são consideradas diretamente nesta definição, uma vez que se encontram inerentes à utilização de *Big Data* (por exemplo, a utilização de *Big Data* deve-se à complexidade que os dados apresentam e o intuito dessa utilização será, nitidamente, a obtenção de valor da sua implementação).

2.1.2. Principais Características

No seu começo, *Big Data* surgiu maioritariamente associado a grandes volumes de dados, no entanto, o seu volume está longe de ser a sua única característica ou o seu único desafio (Zikopoulos,

Paul et al. 2011). Em 2001 o autor Doug Laney (2001), em detrimento do crescimento do volume de dados, define os desafios e oportunidades na sua gestão em três dimensões diferentes, caracterizando *Big Data* pelo aumento do seu volume, velocidade e variedade. O volume é referente à quantidade de dados gerada e extraída das diversas fontes de dados, a velocidade à rapidez com que os dados entram e saem do sistema de armazenamento e a variedade aborda os diferentes formatos e fontes de dados (Krishnan, 2013).

Atualmente, as organizações recorrem maioritariamente a esta definição como um modelo para classificar *Big Data,* salientando também que estas são as características maioritariamente utilizadas pela comunidade, representadas na Figura 4 (De Mauro, Greco, & Grimaldi, 2016; Krishnan, 2013; Zikopoulos, 2011).

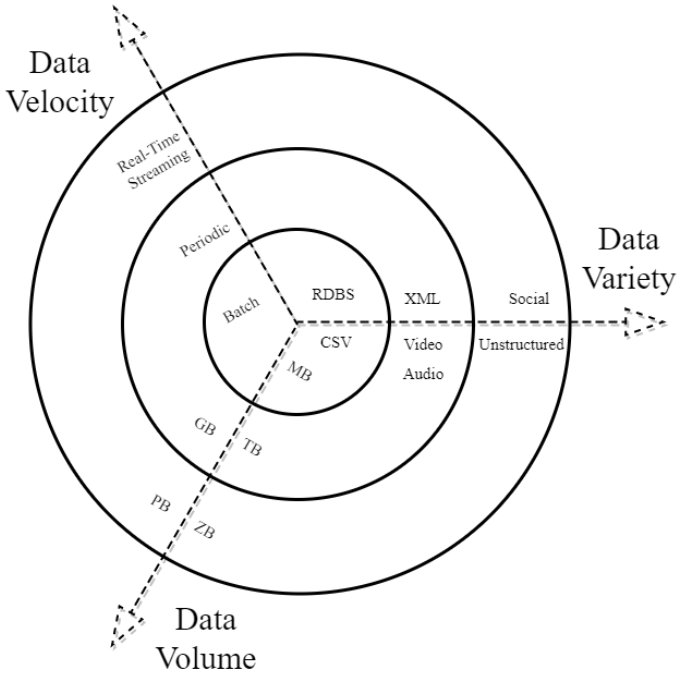

Figura 4. Modelo dos 3 V's. Baseado em (Zikopoulos, 2011).

Em consequência das afirmações anteriores, são descritas cada uma destas características que se encontram ilustradas na Figura 4.

Iniciando pela característica que a comunidade mais associa a *Big Data,* o **volume** de dados refere-se à magnitude de dados que as organizações armazenam com o propósito de melhorar o processo de tomada de decisão (Gandomi & Haider, 2015; Janet, Schroeck, Shockley, Morales, & Tufano, 2012; Kaisler, Armour, Espinosa, & Money, 2013). Os dados podem ser provenientes de diversas fontes, nomeadamente de redes sociais, sensores, tráfego rodoviário, transferências bancárias, entre outros. Contudo, tal volume de dados, por vezes desconhecido e sem estrutura, não consegue ser processado

e armazenado em sistemas tradicionais por razões discutidas anteriormente (Khan, Uddin, & Gupta, 2014).

Atualmente, vários autores não mencionam um limite concreto para se associar o volume de dados a *Big Data,* afirmando que frequentemente o volume se enquadra entre Terabytes (TB) e Petabytes (PB) embora esse limite tenha tendência a sofrer uma mudança para Zettabytes (ZB). Esta tendência é impulsionada não só pelo aumento da capacidade de armazenamento, permitindo o armazenamento de conjuntos de dados com volume superior, bem como ao aumento dos dispositivos capazes de produzir dados através das interações com os diversos serviços (Gandomi & Haider, 2015; Katal, Wazid, & Goudar, 2013; Krishnan, 2013). Apesar do volume de dados disponível nas organizações ter vindo a aumentar, a percentagem de dados a serem analisados tem vindo a diminuir (Zikopoulos, 2011).

Gandomi e Haider (2015) prosseguem mencionando que a definição de volume é variável, nomeadamente na sua periodicidade e no seu tipo de dados. Em outras palavras, dois conjuntos de dados do mesmo tamanho podem requerer diferentes tipos de tecnologias e métodos para executar o seu processamento.

Devido à proliferação de dispositivos inteligentes e sensores, os dados presentes nas organizações têm vindo a ficar complexos, pois estes incluem dados em formatos estruturados, semiestruturados e não estruturados. Assim, surge a característica **variedade.**

A variedade representa a heterogeneidade de estruturar um conjunto de dados, representando uma mudança fundamental nos requisitos de análise de dados tradicionais, de modo a incluir dados semiestruturados e não estruturados como parte do processo da tomada de decisão de uma organização (Gandomi & Haider, 2015; Zikopoulos, 2011).

Krishnan (2013) refere que os dados se apresentam em vários formatos que podem variar desde base de dados relacionais, documentos, imagens, texto, áudio, dados de redes sociais e dados de sensores. Todos estes formatos são totalmente dissemelhantes e para o autor não existe um controlo concreto perante a sua estrutura, podendo esta tomar os formatos mencionados anteriormente.

Janet (2012) vai ao encontro da observação de Krishnan (2013) e Gandomi (2015), fazendo referência não somente aos diferentes tipos de dados mas também às diversas fontes de dados, definindo esta característica pelo seu grau de complexidade em processar diferentes tipos de dados. Mediante o exposto, as organizações sentem a necessidade de integrar e analisar fontes de dados tradicionais em conjunto com fontes de dados não tradicionais, com proveniência interna ou externa das organizações (Janet et al. 2012).

Kaisler (2013), no seu ponto de vista analítico, aponta para esta característica como o maior obstáculo do processamento de grandes volumes de dados, assumindo que estruturas de dados não alinhadas e uma semântica inconsistente representam desafios significativos na sua análise. Com a finalidade de diminuir a complexidade no processamento de vários formatos de dados, Krishnan (2013) retrata a disponibilidade de metadados com o propósito de identificar qual o conteúdo destes dados, alertando que a ausência de metadados poderá conduzir a atrasos no processamento, desde a extração até ao armazenamento.

Assim como o volume e a variedade de dados sofreram uma mudança, o mesmo ocorre com a **velocidade** a que estes dados são gerados, armazenados e analisados (Janet et al. 2012; Zikopoulos, 2011).

Em referência à velocidade, para Gandomi e Haider (2015) esta diz respeito tanto à rapidez na qual os dados são produzidos, como à velocidade que estes devem ser analisados. Os aumentos de dispositivos digitais provocam um crescimento na taxa a que os dados são gerados, impulsionando uma necessidade de análise em tempo real.

Zikopoulos (2011) e Khan (2014) sugerem que esta característica se aplique à rapidez a que os dados fluem até ao repositório, afirmando que o volume e a variedade de dados mencionados anteriormente são uma consequência da velocidade a que estes são produzidos. Os dados produzidos são provenientes de fontes heterogêneas, variando entre *batch* e *streaming* (Chandarana & Vijayalakshmi, 2014; Katal et al. 2013).

Em contextos de *Big Data* é importante não só a velocidade a que os dados são armazenados no repositório mas também todo o processo até à tomada de decisão, com o propósito de maximizar a eficiência nos processos de negócio (Janet et al. 2012; Khan et al. 2014).

Atualmente, os dados produzidos não conseguem ser extraídos, armazenados e analisados através de tecnologias tradicionais, surgindo a necessidade de um sistema de processamento capaz de trabalhar com velocidades escaláveis e tamanhos extremamente variáveis, num curto período de tempo (Krishnan, 2013).

Em virtude do que foi referido anteriormente, em torno do modelo dos três Vs atribuídos a *Big Data*, Krishnan (2013) identifica três novas características que surgem do cruzamento entre volume, variedade e velocidade como ilustra a Figura 5.

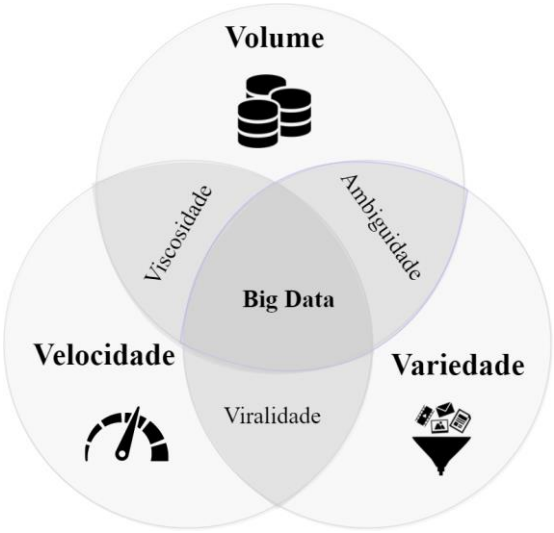

Figura 5. Modelo dos 3Vs com características adicionais. Baseado em (Krishnan, 2013).

Estas características apresentadas por Krishnan (2013) na Figura 5, são descritas da seguinte forma:

- **Ambiguidade:** esta característica manifesta-se entre a variedade e o volume. A falta de metadados é um dos principais fatores para a sua presença. A título de exemplo, um "M" ou "F" podem significar um género (Masculino ou Feminino) ou podem expressar ser Segunda-feira (em inglês: *Monday*) e Sexta-feira (em inglês: *Friday*);
- **Viscosidade:** esta característica manifesta-se entre o volume e a velocidade, medindo a resistência para o fluxo no volume de dados, podendo estar representada nos fluxos de dados, regras de negócio e limitações tecnológicas;
- **Viralidade:** esta característica manifesta-se entre a variedade e a velocidade. Avalia e descreve a velocidade com que os dados são partilhados na rede. Por exemplo, *re-tweets* que são partilhados de um *tweet* original são uma boa prática para avaliar um tópico ou uma tendência.

Com o passar do tempo, vários autores defenderam que as características apresentadas anteriormente não são suficientes para caracterizar *Big Data* (Chandarana & Vijayalakshmi, 2014; Chen et al. 2014; Janet et al. 2012; Khan et al. 2014). Consequentemente, surgem características que são constantemente mencionadas, tal como **valor** e **veracidade** (Chen et al. 2014). As restantes características identificadas na literatura, nomeadamente, validade, volatilidade, variabilidade e complexidade, não tão referidas pela comunidade, e surgem através da extensão do modelo dos 5 V's

proposto por Chandarana e Vijayalakshmi (2014) e, posteriormente, do modelo dos 7 V's proposto por Khan (2014).

No que diz respeito ao **valor,** Khan (2014) afirma que esta característica tem uma razão especial. Ao contrário das características mencionadas anteriormente, o autor assume que esta característica é o resultado pretendido do processamento de grandes volumes, velocidades e variedades de dados, uma vez que o interesse é extrair o máximo valor deste processamento. Os dados que inicialmente são extraídos, na sua forma original (sem tratamento), tipicamente apresentam um valor reduzido relativamente ao seu volume (Gandomi & Haider, 2015). A junção de vários tipos de dados tem o propósito de extrair conhecimento que anteriormente era desconhecido, com o objetivo final de obter vantagens competitivas para as organizações (Chandarana & Vijayalakshmi, 2014).

Em referência à **veracidade**, antes de ser considerada uma característica, a comunidade científica assumia que os dados que chegavam aos repositórios eram precisos e devidamente tratados (Khan et al. 2014).

De acordo com Janet (2012), a veracidade refere-se ao nível de confiança associado aos vários tipos de dados. A qualidade dos dados é um requisito importante em *Big Data*, no entanto, mesmo com os melhores métodos de tratamento de dados não se consegue remover a imprevisibilidade que se encontra presente em alguns dados, a título de exemplo, condições meteorológicas, fatores económicos e os sentimentos, que apesar de incertos por natureza, podem ser valiosos a longo prazo. A velocidade com que os dados fluem não permite despender períodos de tempo excessivos no seu tratamento, conduzindo à necessidade de estabelecer mecanismos para lidar com dados imprecisos e precisos (Chandarana & Vijayalakshmi, 2014).

A **variabilidade** refere-se à variação da velocidade no fluxo de dados, de acordo com os diferentes picos e reduções periódicas de velocidade (Gandomi & Haider, 2015).

A **validade** de dados evidencia ter o mesmo conceito de veracidade. De facto, a validade está enquadrada na precisão e exatidão dos dados, em relação à utilização pretendida. Por outras palavras, os dados podem não conter qualquer erro de veracidade, porém, podem não ser considerados válidos se não forem devidamente compreendidos (Khan et al. 2014).

A **complexidade** destaca-se pelo facto de *Big Data* impor o desafio de integrar, tratar e transformar dados provenientes de diferentes fontes de dados (Gandomi & Haider, 2015).

A **volatilidade** em *Big Data* é associada à política de retenção de dados numa organização, referindo o período de tempo em que os dados devem ser mantidos e o momento a partir do qual podem ser destruídos (Khan et al. 2014).

A Figura 6 resume as várias características identificadas ao longo desta subsecção.

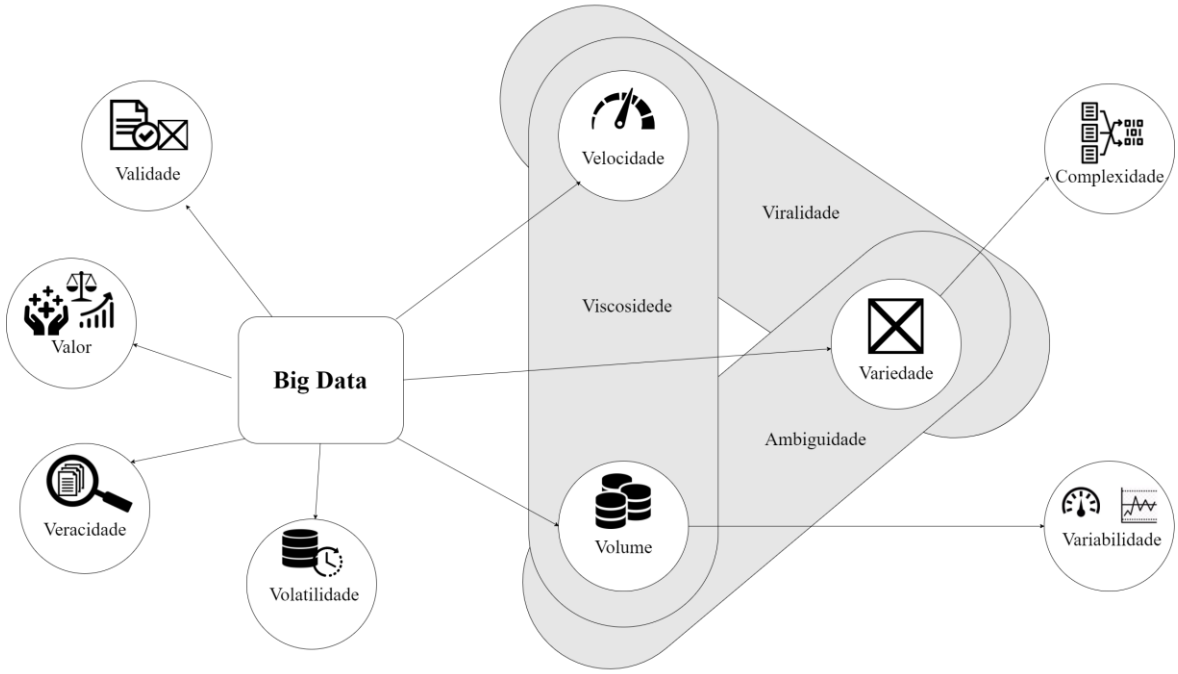

Figura 6. Características principais do Big Data identificadas na literatura. Baseada em (Costa & Santos, 2017).

2.1.3. Oportunidades, Problemas e Desafios do *Big Data*

Recentemente, as organizações têm vindo a manifestar um interesse significativo nas potencialidades de *Big Data*, em resultado do seu potencial e da criação de vantagens competitivas para as organizações que dispõem de grandes planos para acelerar a sua investigação (Chen et al. 2014). Embora a quantidade de grandes volumes de dados esteja drasticamente a aumentar, de igual forma, surge um conjunto de oportunidades, problemas e desafios que será abordado nesta subsecção (J. Fan et al. 2014).

Fan (2014) afirma que a base do surgimento das novas oportunidades se deve maioritariamente ao aumento do volume de dados. Em consequência do volume de dados extraídos em *Big Data*, surge a oportunidade de fornecer serviços personalizados, adaptando os mesmos aos consumidores. Esta capacidade de armazenar grandes volumes de dados, a diferentes velocidades e variedades, permite armazenar os dados históricos do tráfego de rede, possibilitando a identificação da fonte e do destino de uma eventual ameaça, reforçando assim a segurança da *internet.*

Na sua perspetiva, Philip Chen e Zhang (2014) afirmam que, futuramente, retirar conhecimento proveniente dos dados será uma competição entre organizações. Os autores prosseguem enumerando um conjunto de vantagens que podem ser obtidas através de *Big Data*, tal como ilustra a Figura 7, na

qual se consegue obter um enquadramento das oportunidades de *Big Data* ao nível dos processos organizacionais. Destaca-se que 51% das organizações inquiridas acreditam que *Big Data* irá auxiliar na melhoria da eficiência dos seus processos operacionais.

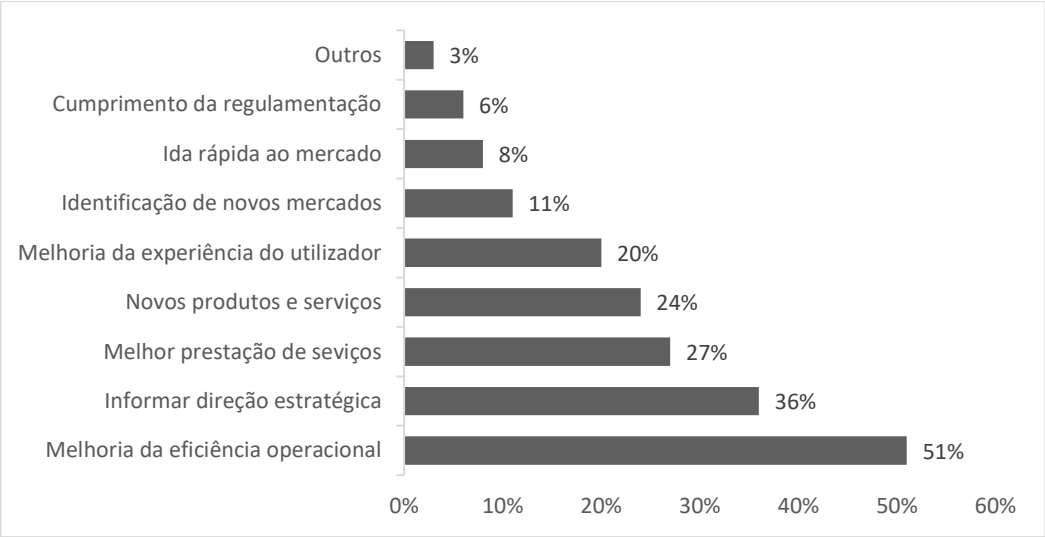

Figura 7. Oportunidades de Big Data nos processos organizacionais. Retirados de (Philip Chen & Zhang, 2014).

No entanto, vários autores presentes na literatura mencionam um conjunto de áreas em que *Big Data* pode desempenhar um papel crucial. Perante isso, através dos contributos de Oussous (2018) e Chandarana (2014), a Figura 8 apresenta não só as áreas em que *Big Data* se pode enquadrar, mas também evidencia alguns exemplos de contributos que podem ser desempenhados nas mesmas.

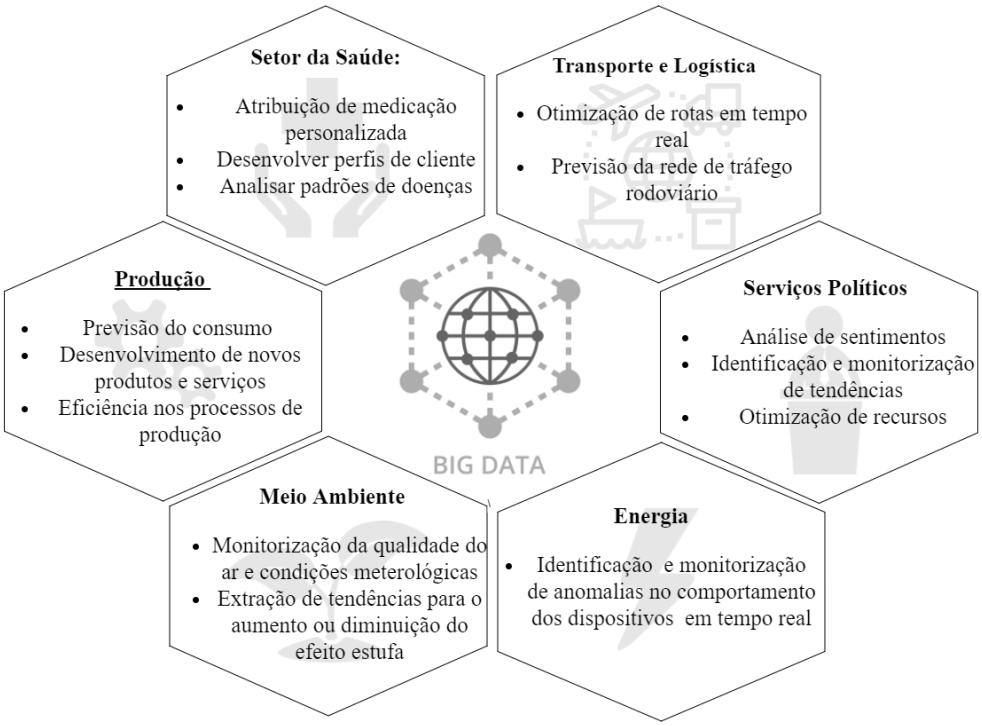

Figura 8. Áreas de intervenção do Big Data. Baseado em *(Chandarana & Vijayalakshmi, 2014; Oussous et al. 2018)*.

Todavia, apesar de se tratar de um domínio que proporciona muitas oportunidades, *Big Data* acarreta também vários desafios e problemas, devido à necessidade de selecionar métodos e tecnologias para extrair, armazenar, gerir e analisar grandes volumes de dados, a diferentes velocidades e variedades, algo que não consegue ser suportado pelos sistemas convencionais (Chen et al. 2014).

De seguida será apresentado um conjunto de desafios a respeito do *Big Data*, que se apresentam divididos em quatro categorias, baseados em Costa e Santos (2017). Em cada uma das categorias ("Dilemas Gerais do *Big Data*", "Desafios no Ciclo de Vida *Big Data"*, "Ambientes Seguros, Privados e Monitorizados em *Big Data"* e "Mudança Organizacional") será realizada uma breve descrição sobre a mesma e apresentado um conjunto de desafios e/ou problemas identificados.

Desafios Gerais do *Big Data*

Esta categoria inclui um conjunto de desafios mais gerais, incluindo a falta de rigor na definição de *Big Data* e a falta de *standards* no que diz respeito a modelos e arquiteturas. Tais desafios ocorrem devido ao *Big Data* não ser uma temática estruturalmente definida e os modelos existentes não sofrerem verificações rigorosas.

Iniciando por um desafio que é abordado anteriormente nesta dissertação, **o conceito de *Big Data***, é regularmente visto como uma especulação comercial ao invés de ser um tópico de investigação científica. Dessa forma, para Chen (2014), existe a necessidade de uma definição rigorosa e holística de *Big Data*, um modelo estrutural e, por fim, um sistema teórico de *Data Science*.

Os autores mencionam a **falta de uniformização em ambientes de *Big Data***, em questões referentes à avaliação da qualidade de dados e mecanismos de *benchmarking.* No mesmo ponto de vista, a falta de mecanismos de *benchmarking* com o propósito da comparação entre diferentes tecnologias é agravado constantemente pela evolução tecnológica em contextos *Big Data* (Baru, Bhandarkar, Nambiar, Poess, & Rabl, 2013).

Um desafio emergente para os investigadores e utilizadores de *Big Data* é a "**qualidade vs quantidade"**. Cada vez mais os utilizadores têm acesso a grandes volumes de dados acreditando que com a quantidade suficiente de dados conseguem ser capazes de explicar qualquer fenómeno *Big Data.* Em contraste, um utilizador de *Big Data* deve concentrar-se na qualidade de dados, por outras palavras, apesar de os dados não se apresentarem disponíveis na sua totalidade, existe uma quantidade de dados de alta qualidade que pode ser utilizada para tirar conclusões precisas e de valor (Kaisler et al. 2013). Frequentemente, quando se lida com grandes volumes de dados existem algumas questões que são difíceis de responder (Chandarana & Vijayalakshmi, 2014), nomeadamente, quais os dados que se consideram irrelevantes e quais as técnicas de seleção dados que se consideram relevantes? Como se

estima o valor extraído pelos dados? Como se garante a autenticidade dos dados? Regularmente, é debatido de que forma é que *Big Data* representa melhor a população em comparação com um conjunto de dados inferior. Obviamente, existem questões que variam consoante o contexto, no entanto, os autores estão convictos que maior quantidade de dados não significa melhor qualidade (Costa & Santos, 2017).

Desafios no Ciclo de Vida *Big Data*

Estes desafios fazem referência a dificuldades técnicas em ambientes *Big Data*, nomeadamente nos mecanismos de extração, integração, limpeza, transformação, armazenamento, processamento, análise e governança de dados.

As organizações têm vindo a sofrer grandes mudanças a nível tecnológico, aumentando cada vez mais o número de dispositivos capazes de produzir dados. O aumento da quantidade de dados "explode" cada vez que é criado um novo meio de armazenamento (Kaisler et al. 2013; Katal et al. 2013). Em consequência, surgem desafios ao nível do **armazenamento e processamento de dados.** Philip Chen e Zhang (2014) afirmam que o processo de armazenamento e processamento em *Big Data* sofreu uma mudança ao nível dos dispositivos de armazenamento, da arquitetura de armazenamento e dos mecanismos de acesso aos dados. Posto isto, existe a necessidade de repensar nestes mecanismos com o propósito de tornar cada vez mais eficiente a entrada e a saída de dados (I/O), uma vez que a disponibilidade dos dados é uma das principais prioridades para a extração de conhecimento.

Garantir a **qualidade *dos* dados** e adicionar **valor** aos mesmos através da sua preparação torna-se um desafio (Philip Chen & Zhang, 2014). A qualidade dos dados influencia a utilização de *Big Data* na medida em que dados que apresentam uma qualidade reduzida estão a desperdiçar recursos no processamento e armazenamento correspondente. A qualidade de dados é maioritariamente definida pela sua precisão, redundância e consistência. Portanto, surge a necessidade de investigar novos métodos e técnicas com o propósito de tornar mais eficazes e eficientes os mecanismos de enriquecimento dos dados (Chen et al. 2014; Khan et al. 2014).

Quando um ser humano absorve informação, uma grande quantidade de heterogeneidade é confortavelmente tolerada. Porém, o mesmo não acontece com os algoritmos tradicionais que apenas conseguem lidar com dados homogéneos (Agrawal et al. 2011). A **heterogeneidade** advém de múltiplas fontes e de diversos tipos de dados, nomeadamente, dados estruturados, semiestruturados e não estruturados.

A **escalabilidade** tornou-se crucial para o armazenamento e análise dados (Costa & Santos, 2017). Manipular grandes volumes de dados requer o *redesign* das bases de dados e algoritmos de

forma a extrair valor dos dados (Cuzzocrea, Song, & Davis, 2011). Anteriormente, os sistemas de processamento de dados tradicionais necessitavam de considerar o paralelismo entre os nós do *cluster,* porém, a preocupação atual centra-se no paralelismo dentro de um único nó (Katal et al. 2013).

Em ambientes *Big Data*, a **governança dos dados** é um processo complexo no que diz respeito ao controlo e autoridade em relação a grandes volumes de dados, provenientes de fontes distintas. Manipular dados num ambiente heterogéneo para planear e garantir a rastreabilidade dos dados torna-se quase impossível sem as ferramentas de governança adequadas (Costa & Santos, 2017).

Concluindo esta categoria, Chen (2014) afirma que as organizações enfrentam um grande conjunto de desafios no ciclo de vida do *Big Data*. Ultrapassar esses desafios depende de um conjunto de fatores, nomeadamente, ao estado de maturidade da organização, à utilização de sistemas tradicionais e à utilização de formatos incompatíveis de dados que podem impor dificuldades na sua integração e extração de valor de *Big Data.*

Ambientes Seguros, Privados e Monitorizados em *Big Data*

Atualmente, a privacidade e segurança dos dados é uma das temáticas que mais preocupam as organizações (Chen et al. 2014). É relevante mencionar que as organizações devem arquitetar modelos de segurança de *Big Data* com o objetivo de atingir um elevado grau de precisão na prevenção e especificação de ameaças (Costa & Santos, 2017).

Chen (2014) afirma que as soluções *Big Data* enfrentam grandes desafios relacionados com a **privacidade dos dados**. Este desafio incluí dois aspetos principais, nomeadamente, a proteção da privacidade pessoal durante o processo de extração de dados (hábitos, interesses pessoais) e proteção da privacidade dos dados ao longo dos seus fluxos.

Nesta categoria, a **propriedade dos dados** também se apresenta como um desafio crítico, particularmente em cenários relacionados com as redes sociais. Apesar de grandes volumes de dados se encontrarem armazenados em servidores das organizações, não implica necessariamente que estas sejam as proprietárias dos dados (Chen et al. 2014). Os autênticos proprietários são as pessoas que criam as páginas ou contas, no entanto, as organizações atuam como se os dados fossem da sua propriedade e a legislação tende a beneficiá-las, permitindo que não eliminem permanentemente os dados, mesmo quando os utilizadores os requerem. Utilizadores podem não mencionar alguns factos sobre si, no entanto existe um receio quanto ao uso inadequado dos seus dados através do cruzamento de informação pessoal com grandes conjuntos de dados de elevada qualidade, permitindo a inferência de novos factos sobre a pessoa, factos esses que podem ser altamente intrusivos (Agrawal et al. 2011, 2011; Chen et al. 2014; Manyika et al. 2011).

Posto isto, regularmente se lida com informação sensível referente aos utilizadores, nomeadamente em questões relacionadas com a saúde ou finanças. Perante tal informação, é relevante a aplicação de **leis e regulamentação** (Costa & Santos, 2017; Kaisler et al. 2013; Katal et al. 2013). Com o objetivo de obter valor proveniente da utilização de soluções *Big Data,* os dados têm de estar acessíveis e utilizáveis e, sendo assim, as organizações devem cumprir estes requisitos pela adesão a esta regulamentação (Chandarana & Vijayalakshmi, 2014). Em conformidade, tais requisitos encaminham as organizações a refletir e a tomar decisões sobre a seleção dos dados, ou seja, quais os dados que podem trazer valor para a organização, assumindo que existem conjuntos de dados que não podem ser utilizados devido ao cumprimento da regulamentação (Agrawal et al. 2011). Kaisler (2013) coloca um conjunto de questões que devem ser consideradas: Quais as regras ou legislações que devem existir a respeito de introduzir dados que contêm informação pessoal, provenientes de várias fontes, num único repositório de dados? As regras devem aplicar-se ao repositório total ou apenas a partes que contêm informação relevante? Que regras devem existir para a proibição e armazenamento de dados sobre indivíduos?

Em síntese, a respeito desta categoria, pode-se considerar que os vários autores mencionados anteriormente sentem a falta de *standards* definidos a respeito da privacidade e segurança dos dados e que certamente será um tópico em que as questões legais devem ser repensadas a respeito da simplicidade de copiar, integrar e utilizar recorrentemente os dados de diferentes pessoas (Manyika et al. 2011).

Mudança Organizacional

Big Data é uma das tendências mais recentes da atualidade e seguramente que é um tema que suscita interesse a várias organizações. Porém, frequentemente os administradores das organizações não compreendem o valor proveniente deste fenómeno e principalmente como extrair esse valor (Krishnan, 2013; Manyika et al. 2011).

Um dos principais desafios nesta categoria é **compreender o valor** proveniente do *Big Data*, que se deve maioritariamente à falta de conhecimento em realizar análise de dados, apresentando como principal obstáculo a transformação para uma organização orientada aos dados (LaValle, Lesser, Shockley, & Hopkins, 2011).

Em muitas áreas de negócio, as organizações necessitam de monitorizar tendências de forma a obter vantagens competitivas em relação aos seus concorrentes, no entanto, Manyika (2011) afirma que ocasionalmente as organizações têm falta de talento, rigor e iniciativas para a implementação de soluções *Big Data.*

Conduzir a **mudança de paradigma na organização** para que seja possível a implementação de uma solução *Big Data* requer a introdução da análise de dados nas suas funções operacionais e nas principais áreas de negócio, transformando os processos de negócio, fornecendo destaques relacionados com os clientes, produtos e serviços (Costa & Santos, 2017). McAfee e Brynjolfsson (2012) argumentam que a implementação de soluções *Big Data* acarreta inúmeros desafios, nomeadamente, a necessidade de uma forte componente de liderança, de cientistas de dados capazes de compreender e manipular ferramentas *Big Data* e a necessidade de modificar a cultura organizacional.

2.1.4. Processamento de Dados

O processamento de dados tem-se mostrado como um tópico complexo de abordar desde os primeiros dias da computação. Segundo Krishnan (2013), processamento de dados pode ser definido como a extração, processamento e gestão de dados com o propósito de fornecer informação aos utilizadores finais.

O processamento de dados tradicional segue o ciclo de vida num ambiente em que os dados sofrem primeiramente uma análise e, posteriormente, com base na análise é concebido um modelo de dados e uma estrutura de base de dados com o objetivo de os processar e armazenar. Nesta abordagem os dados extraídos encontram-se por natureza estruturados e discretos em relação ao seu volume, uma vez que todo o processo é pré-definido com base nos seus requisitos. Domínios como a qualidade e limpeza dos dados não são intitulados como problemáticos, pois sofrem tratamento nos sistemas tradicionais (Krishnan, 2013).

Porém, o ciclo de vida do processamento de dados tradicional difere tipicamente do processamento de dados em ambientes de *Big Data*. No ciclo de vida do processamento de *Big Data,* representado através da Figura 9, não existe a necessidade de iniciar o processo através da transformação dos dados para que estes "encaixem" no modelo relacional. Os dados primeiramente são extraídos e carregados, tipicamente, para um sistema de ficheiros distribuído e, posteriormente, a camada de metadados será aplicada aos dados e a estrutura de dados para o seu conteúdo será criada. Uma vez concebida a estrutura de dados, os dados sofrem as devidas transformações. O resultado final deste processamento permitirá retirar análises e padrões relevantes perante o contexto em que os dados se inserem (Krishnan, 2013).

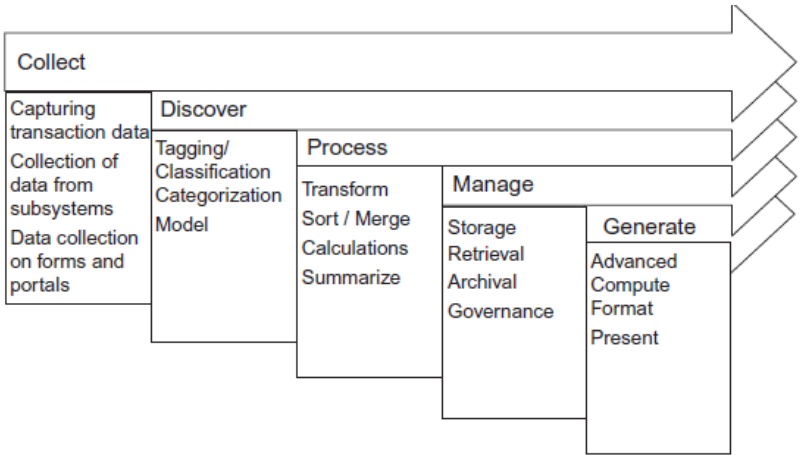

Figura 9. Ciclo de vida do processamento de dados Big Data. Retirado de (Krishnan, 2013).

Krishnan (2013) refere que para obter um desempenho positivo no processamento de grandes volumes de dados a diferentes velocidades e variedades é necessária uma arquitetura orientada a ficheiros. O autor prossegue com a sua opinião referindo que para conceber uma infraestrutura e um processamento eficiente existe a necessidade de compreender o fluxo de dados para processar *Big Data.* Perante as dificuldades inerentes ao processamento de dados em ambientes *Big Data*, o autor propõe uma visão superficial do fluxo de processamento em *Big Data*, presente na Figura 10.

De seguida é descrita cada uma das quatro fases principais apresentadas pelo autor, para o processamento de dados em ambientes *Big Data:*

1. **Recolha:** nesta fase os dados são recolhidos de diferentes fontes de dados e carregados num sistema de ficheiros, tipicamente intitulado de área de estágio;
2. **Carregamento:** nesta fase, os dados são carregados com a aplicação de metadados, aplicando a estrutura de dados pela primeira vez, apresentando-se prontos para a fase de transformação;
3. **Transformação:** nesta fase, os dados são transformados de acordo com as regras de negócio. Esta fase inclui múltiplos passos para a sua execução, por vezes complexas devido ao seu tipo de conteúdo;
4. **Extração de dados:** nesta fase, o conjunto de dados pode de ser extraído para várias finalidades, nomeadamente, tarefas de análise, relatórios operacionais, integração num *Data Warehouse*, visualização, entre outras.

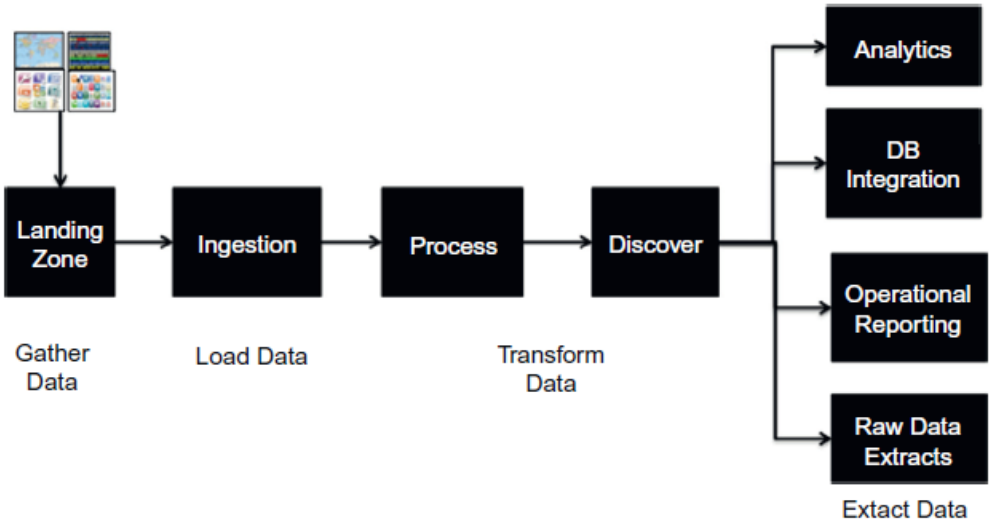

Figura 10. Fluxo de processamento em *Big Data. Retirado de (Krishnan, 2013).*

Terminadas as fases principais do fluxo de processamento de *Big Data,* que permitem processar dados de uma forma flexível, é relevante também mencionar os tipos de processamento existentes em ambiente *Big Data.* De acordo com Du (2015), o processamento de dados pode ser enquadrado com estes três tipos:

1. **Processamento em *Batch:*** este tipo de processamento é utilizado para processar dados em formato *batch,* através da sua extração, processamento e carregamento para o destino predefinido. Os dados são distribuídos pelo sistema de ficheiros, divididos em ficheiros de menor dimensão. As desvantagens neste modelo de processamento são as incapacidades de executar tarefas de recursividade e interatividade;
2. **Processamento em tempo real*:*** neste tipo de processamento os dados são processados à medida em que chegam ao repositório de dados e os resultados são quase imediatos. Caracteriza-se pela capacidade de executar *queries* em paralelo, ao invés das sequências em formatos *batch.* Anteriormente, este tipo de implementação era caracterizado pelo seu elevado custo, no entanto, atualmente os preços das memórias RAM estão cada mais acessíveis e este tipo de implementação encontra-se mais exequível em comparação com o passado;
3. **Processamento em *streaming:*** neste tipo de processamento, os dados são continuamente processados, obtendo resultados à medida em que os novos dados surgem.

2.2. Armazenamento de Dados

As organizações estão constantemente a ser desafiadas em diversas áreas e os desafios emergem desde *stakeholders*, informação tecnológica, necessidades de negócio, entre outros. No enquadramento das tecnologias da informação, a preocupação da evolução da capacidade de extrair, armazenar e processar grandes volumes de dados tem aumentado a necessidade de desenvolver novos ambientes, utilizando novas tecnologias que requerem mudanças nos sistemas de informação das organizações e nas suas capacidades analíticas (C. Costa & Santos, 2016).

Na presença de características como o volume de dados, em *Big Data,* o armazenamento de dados é uma das temáticas mais importantes. Posto isto, esta secção é iniciada com uma descrição das bases de dados SQL (*Structured Query Language*), NoSQL (*Not Only SQL)* e NewSQL. Posteriormente, segue-se o conceito de *Data Warehouse* e a distinção entre os sistemas analíticos que tipicamente fornecem suporte aos *Data Warehouses* e os sistemas transacionais (OLTP), que fornecem suporte às bases de dados operacionais. Por fim, aborda-se o conceito de *Big Data Warehouse* e termina-se com uma abordagem aos modelos de dados.

2.2.1. Bases de Dados SQL, NoSQL e NewSQL

Nesta temática das bases de dados SQL, NoSQL e NewSQL, existe um grande desafio que se enquadra na seleção do sistema de base de dados mais adequado para um contexto. A filosofia de "*no one sizes fits all*" suportada por Lim, Han e Babu (2013) retrata o que foi mencionado anteriormente. Posto isto, face à diversidade de bases de dados disponíveis, torna-se difícil fornecer uma solução única (modelo de bases de dados único) que seja adequada para todos os contextos (Bach & Werner, 2014). Lim (2013) apresenta as principais questões que devem ser consideradas na seleção do sistema de armazenamento: deve-se utilizar um sistema SQL, NoSQL ou NewSQL? Dentro dos sistemas NoSQL, deve-se utilizar que tipo de base de dados? Que operações se pretende que os sistemas executem?, entre muitas outras questões. A resposta a este conjunto de questões revelam um grau de experiência elevado na área, visto que executar tarefas de *benchmarking* destes sistemas sem a aplicação estar totalmente desenvolvida é uma tarefa tipicamente complexa (Lim et al. 2013).

Esta diversidade é refletida na Figura 11, que representa a separação entre as bases de dados relacionais e as não relacionais. Os sistemas bases de dados apresentam-se divididos entre transacionais (ao nível das operações) e analíticos (ao nível da tomada de decisão) (Bach & Werner, 2014; Lim et al. 2013).

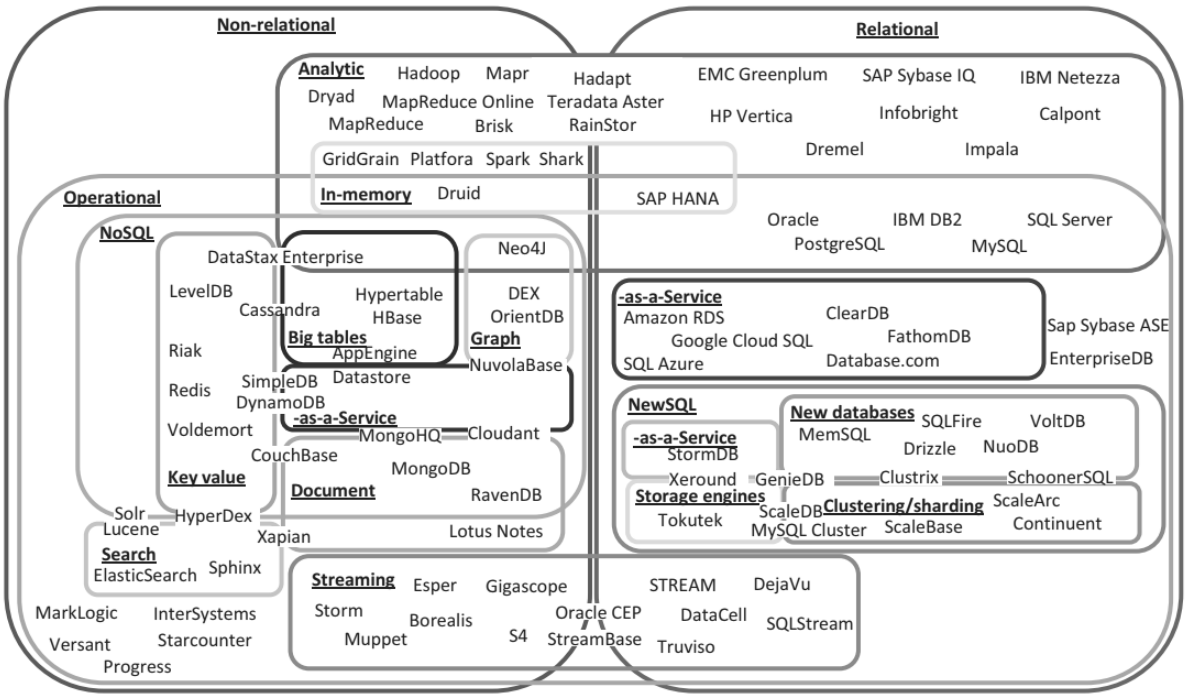

Figura 11. Categorização da variedade de bases de dados disponíveis. Retirado de (Lim et al. 2013).

Existem três principais sistemas de bases de dados, nomeadamente, RDBS (*Relational Database Systems)* ou *SQL*, *NoSQL e NewSQL.* As bases de dados SQL são suportadas por um modelo relacional e caracterizam-se pela utilização de vários tipos de chaves, nomeadamente, chave primária que é denominada como a chave mais importante na tabela devido à identificação de cada linha inserida na tabela de forma única, chaves estrangeiras, entre outras. Este modelo de base de dados utiliza as propriedades ACID *(Atomicity, Consistency, Isolation, Durability)* que constituem as características mais relevantes das bases de dados SQL (Fatima & Wasnik, 2016; Krishnan, 2013). Apesar das qualidades provenientes do processamento de transações ACID, estes sistemas não são capazes lidar com os requisitos de escalabilidade e disponibilidade exigidos em contextos *Big Data* (Fatima & Wasnik, 2016; Krishnan, 2013; Stonebraker, 2010; Vaish, 2013).

Por consequência, as bases de dados NoSQL e *NewSQL* emergiram. A popularidade das bases de dados não relacionais tem vindo a aumentar devido às suas características, nomeadamente, velocidade, acessibilidade e escalabilidade (Y. Li & Manoharan, 2013).

NoSQL embora seja tratado como "*Not Only SQL",* inicialmente foi pensado na combinação de apenas duas palavras "No" e "SQL", o que implicaria a separação em relação à linguagem SQL, embora autores como Vaish (2013), Bach e Werner (2014) afirmam que a sua definição implica a separação em relação ao modelo relacional que suporta as RDBS, ao invés da sua linguagem.

Vaish (2013) e Bonnet (2011) reconhecem que as base de dados NoSQL oferecem um conjunto de vantagens face às bases de dados tradicionais, que são apresentadas de seguida:

- **Flexibilidade na representação de dados**: implementações NoSQL apresentam uma representação dos dados flexível. A estrutura uma vez definida, apresenta disponibilidade para uma possível transformação ao longo do tempo (adição de novos atributos aos registos);
- **Tempo de desenvolvimento:** em alguns cenários, o tempo de implementação de bases de dados NoSQL é reduzido, visto que não é necessário lidar com a complexidade das *queries* SQL, devido a não suportar relações entre tabelas;
- **Velocidade:** bases de dados NoSQL apresentam operações de leitura e escrita (I/O) mais eficientes e mecanismos de agregação de dados mais rápidos;
- **Custos**: grande parte das bases de dados NoSQL são *open-source;*
- **Escalabilidade**: ao invés das bases de dados tradicionais, as bases de dados NoSQL são caracterizadas por apresentarem uma escalabilidade horizontal. Bases de dados relacionais requerem muita complexidade para garantir as propriedades ACID.

Porém, Bonnet (2011) identifica algumas debilidades:

- **Consistência dos dados:** não apresentam garantias na consistência de dados. Os utilizadores que implementam soluções NoSQL têm que manter em mente que o principal interesse destas bases de dados não é a consistência. NoSQL não apresenta transações ACID;
- **Migração de dados:** migração de dados de sistemas SQL para NoSQL não é um processo trivial, exigindo alguma experiência em modelos de dados NoSQL em contextos *Big Data.*

Tendo em conta a análise das características e dos contributos que as bases de dados NoSQL podem apresentar face às bases de dados SQL, pode-se constatar que as bases de dados NoSQL oferecem vantagens em questões de escalabilidade, disponibilidade e acessibilidade, no entanto, este tipo de base de dados não é a solução para todos os problemas, particularmente, em contextos específicos em que a consistência de dados é uma característica relevante (Vaish, 2013).

ACID vs BASE

As propriedades ACID (*Atomicity, Consistency, Isolation, Durability)* são uma das principais diferenças entre as bases de dados SQL e NoSQL. Em alguns cenários específicos, as propriedades ACID conferem a confiabilidade da base de dados. A transação deve ser **atómica**, por outras palavras, uma falha numa parte da transação origina uma falha total da transação. A informação deve ser **consistente**,

ou seja, uma transação não pode comprometer o estado da base de dados. Múltiplas transações ao mesmo tempo não causam impacto umas nas outras: este é um requisito de **isolamento** e, por fim, a informação deve permanecer no sistema de armazenamento mesmo após reiniciar a aplicação: **durabilidade** (Bonnet et al. 2011; Cassavia et al. 2014; Costa & Santos, 2017; Oussous et al. 2018; Vaish, 2013)**.**

Porém, alguns autores sugerem o BASE *(Basic Availability, Soft state, Eventual Consistency)* em detrimento do ACID, dando menos relevância à consistência e serialização, e valorizando o melhor desempenho, escalabilidade e disponibilidade. A **disponibilidade básica** assegura que cada pedido obtém uma resposta. Uma das vantagens das bases de dados NoSQL, referida anteriormente neste documento é o seu **estado flexível**, visto que, o seu estado pode alterar ao longo do tempo sem comprometer a base de dados. Por fim, o sistema pode **eventualmente** ser **consistente** (Bonnet et al. 2011).

Teorema CAP

No ano 2000, Eric Brewer apresentou um teorema que consistia em três fatores que devem ser considerados quando se desenvolve e implementa sistemas de dados em ambientes distribuídos. Estes três fatores são a consistência, disponibilidade e a tolerâncias a partições, representados na Figura 12, ficando denominado de teorema CAP.

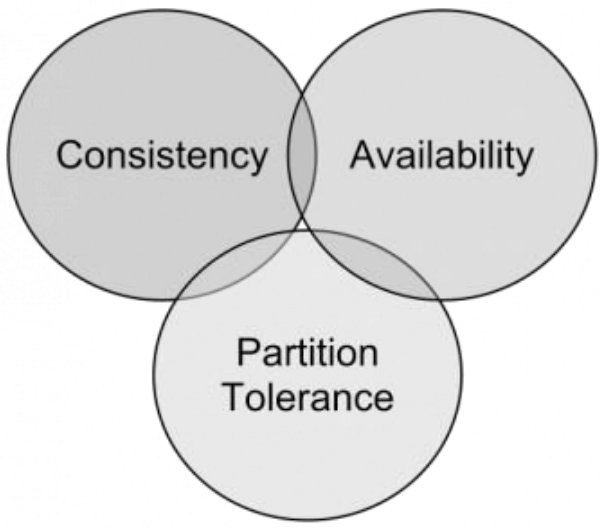

Figura 12. Teorema CAP de Eric Brewer.

Assim, segundo Bonnet et al. (2011), Brewer (2012) e Cassavia et al. (2014), as diversas bases de dados NoSQL podem ser classificadas de acordo com o teorema de CAP, no entanto, não são capazes

de responder às três propriedades simultaneamente, mas podem dar resposta a pelo menos duas das três, que são:

1. **Consistência:** caso seja executada uma operação de atualização de dados, todos os utilizadores têm de ser capazes de ter a mesma visão sobre o mesmo conjunto de dados. Esta propriedade vai ao encontro da propriedade de atomicidade nas transações ACID;
2. **Disponibilidade:** um sistema é considerado disponível se for estruturado e implementado de forma a que permita que uma operação (escrita ou leitura de dados) seja concluída mesmo que algum nó do *cluster* falhe ou algum componente de *hardware* ou *software* esteja em manutenção;
3. **Tolerância à partição:** as operações têm de ser capazes de atingir a sua execução na totalidade mesmo que algum nó se apresente indisponível.

Perante o que foi mencionado anteriormente, através da Figura 13, pode-se observar a classificação das diversas bases de dados de acordo com o teorema de CAP.

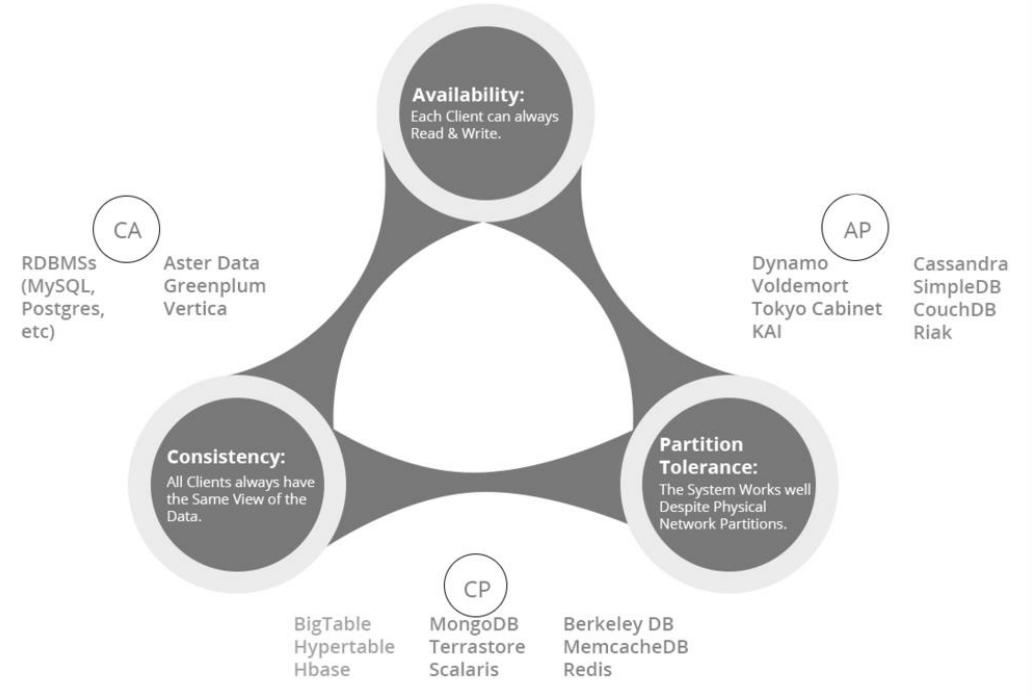

Figura 13. Classificação das bases de dados NoSQL segundo o teorema de CAP. Adaptado de (Bonnet et al. 2011).

Através da análise à Figura 13 pode-se verificar alguns exemplos de bases de dados NoSQL e SQL, e constatar que algumas delas conseguem garantir a consistência e a disponibilidade **(CA)**, significando que o sistema está limitado ao nível da escalabilidade. Por outro lado, existem soluções capazes de garantir a consistência e a tolerância à partição **(CP)**, correndo o risco de alguns dados não

se apresentarem disponíveis caso ocorra algum erro num dos nós do *cluster.* Por fim, algumas soluções conseguem garantir a disponibilidade e a tolerância à partição **(AP)**, embora os dados retornados da base de dados, por vezes, possam não apresentar o nível de precisão pretendido.

As bases de dados NoSQL são tipicamente classificadas em quatros tipos diferentes de acordo com o seu modelo de dados, tal como representado na Figura 14 (Cassavia et al. 2014; C. Costa & Santos, 2016; Vaish, 2013):

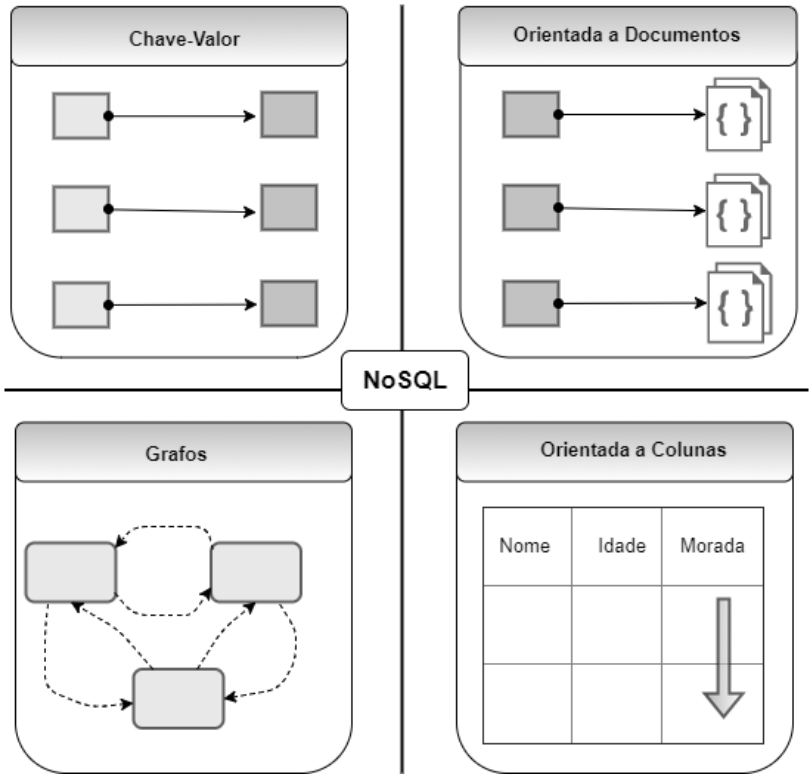

Figura 14. Tipos de base de dados NoSQL.

- **Bases de dados chave-valor:** este tipo de base de dados consiste num conjunto de pares chave-valor únicas, que permitem aceder ao respetivo valor associado à chave. Devido à sua estrutura, as chaves permitem o mapeamento de objetos mais complexos, nomeadamente, listas, outros pares chave-valor, entre outros. Tipicamente suportam operações CRUD (*Create, Read, Update, Delete),* no entanto, não suportam operações de agregação e de junção (Gessert, Wingerath, Friedrich, & Ritter, 2017; Vaish, 2013). Exemplos: Voldemort[5], Redis[6];
- **Bases de dados orientadas a colunas:** ao invés das bases de dados relacionais, as bases de dados orientadas a colunas agrupam os dados em colunas. As colunas são logicamente

[5] https://www.project-voldemort.com/voldemort/
[6] https://redis.io/

agrupadas em famílias de colunas, que contêm um número ilimitado de colunas, possibilitando a alteração do modelo de dados através da criação de novas colunas ao longo do tempo. A manipulação de colunas permite um melhor desempenho na computação de funções de agregação, particularmente em grandes volumes de dados (C. Costa & Santos, 2016; Gessert et al. 2017; Vaish, 2013). Exemplos: Apache Cassandra[7], HBase[8];

- **Bases de dados orientadas a documentos:** as bases de dados orientadas a documentos são caracterizadas como uma base de dados chave-valor que restringe os dados a formatos semiestruturados, tais como, formatos JSON[9]. Os documentos são indexados através de chaves, permitindo superar os sistemas de ficheiros tradicionais (Costa & Santos, 2016; Vaish, 2013). Exemplos: MongoDB[10] e CouchDB[11];
- **Bases de dados grafos:** representam uma categoria especial entre as bases de dados NoSQL. As bases de dados de grafos são propostas para determinados contextos em que a relação entre as entidades é relevante. Este tipo de base de dados não se destina a armazenar grandes volumes de dados, no entanto, permite o armazenamento de dados complexos. É tipicamente composta por um conjunto de nós e arestas, representando o primeiro as entidades e o segundo o tipo de relações entre entidades (Bach & Werner, 2014). Exemplo: Neo4j[12].

Como foi mencionado anteriormente, as propriedades ACID retratam a principal diferença entre as bases de dados SQL e NoSQL. Porém, recentemente uma nova abordagem tem sido proposta pela comunidade: bases de dados *NewSQL.* Estas bases de dados relevam-se promissoras, conservando e suportando as propriedades ACID provenientes das bases de dados tradicionais e, ao mesmo tempo, igualam o desempenho escalável dos sistemas NoSQL (Cattell, 2011; Fatima & Wasnik, 2016; Kepner et al. 2016). As bases de dados VoltDB[13] e NuoDB[14] são exemplos de bases de dados NewSQL. Tendo em conta que o foco desta dissertação incide sobre as bases de dados NoSQL, mais especificamente, em bases de dados orientadas a colunas e a grafos, não se considera importante alongar a discussão sobre as diferenças entre NoSQL e NewSQL.

[7] http://cassandra.apache.org/
[8] https://hbase.apache.org/
[9] https://www.json.org/
[10] https://www.mongodb.com/
[11] http://couchdb.apache.org
[12] https://neo4j.com/
[13] https://www.voltdb.com/
[14] https://www.nuodb.com/

2.2.2. Sistemas de *Data Warehousing*

Esta subsecção tem como objetivo apresentar uma síntese referente a um sistema de *Data Warehousing* e como este se enquadra num sistema de *Business Intelligence* (BI). Os sistemas de BI são abordados superficialmente nesta dissertação, visto que, o seu enquadramento não recai sobre os mesmos.

Data Warehousing é o fenómeno que surgiu como consequência do armazenamento de grandes volumes de dados e da necessidade de analisar os dados com o objetivo de melhorar o processo de tomada de decisão (Golfarelli & Rizzi, 2009; Vaisman & Zimányi, 2012).

De acordo com Santos e Ramos (2017), um DW é um repositório construído para a consolidação de informação da organização num formato válido e consistente, com a finalidade de permitir uma análise dos dados. Os sistemas de BI integram a atividade de exploração do sistema de DW, nomeadamente, elaboração de relatórios e consultas que possibilitam a monitorização da evolução dos indicadores de negócio.

Existem dois autores que se destacam nesta temática, Inmon (2002) e Kimball(2013), no entanto, ambos têm abordagens diferentes. A abordagem de Inmon (2002) centra-se numa abordagem *top-down* que tipicamente se caracteriza pelo desenvolvimento do DW com base no modelo de dados da empresa. Ao invés, a abordagem de Kimball (2013) é apresentada como uma abordagem *bottom-up,* que é representada pelo fluxo inicial entre as fontes de dados e os *Data Marts* e, posteriormente, um fluxo de dados entre *Data Marts* e o DW organizacional. Os *Data Marts* são subconjuntos de dados focalizados num departamento ou repositório específico da organização, tipicamente caracterizados como um subconjunto de um DW (Gardner, 1998; Kimball & Ross, 2013).

De acordo com Inmon (2002), um DW consiste num conjunto de dados orientado por assunto, integrado, catalogado temporalmente e não volátil, que suporta os gestores no processo de tomada de decisão. Importa clarificar os atributos da sua definição, nomeadamente:

- **Orientados por tema ou assunto:** num DW os dados são organizados em torno dos principais assuntos das organizações, tais como, produtos, vendas, encomendas ou clientes, ao contrário do que sucede nos sistemas operacionais, que estão vocacionados para o processamento de transações diárias;
- **Integrados:** um DW é tipicamente construído a partir de múltiplas fontes de dados. Através de técnicas de limpeza e integração, aplicadas aos dados, é possível obter uma visão unificada do conjunto de dados total de uma organização;

- **Variáveis no tempo:** o DW deve fornecer uma perspetiva de análise histórica dos dados. Tipicamente são caracterizados por apresentarem os dados num horizonte temporal entre 5 a 10 anos, ao invés dos dados operacionais que apresentam um horizonte temporal relativamente curto;
- **Não volátil:** um DW apenas suporta dois tipos de operações: o carregamento (carregamento inicial e refrescamento) dos dados e o acesso aos dados carregados para consulta. Assim, os dados nunca podem ser alterados ou eliminados (podem surgir algumas exceções através das SCDs (*Slowly Changing Dimensions)*.

Numa visão complementar, segundo Kimball (2013), num DW existem quatro componentes principais que se encontram representados na Figura 15.

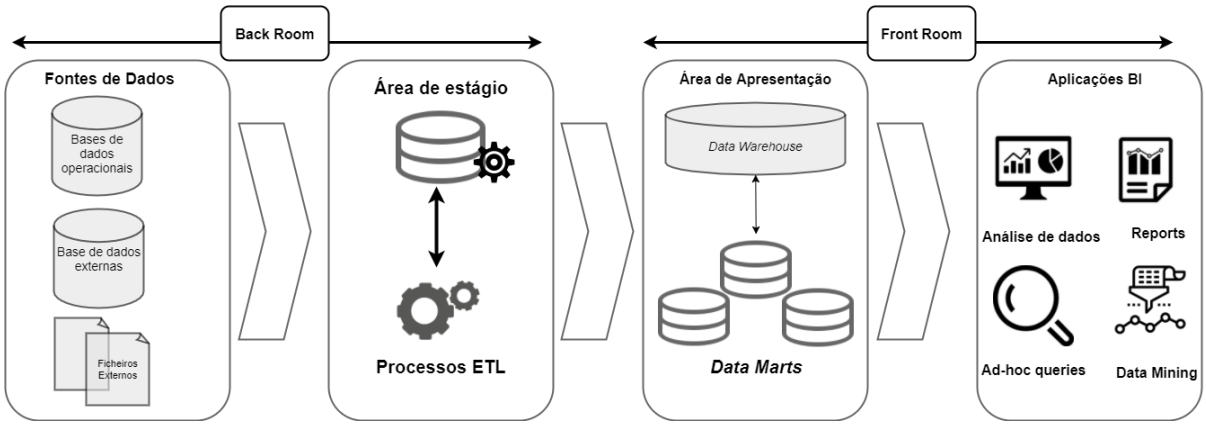

Figura 15. Arquitetura do Data Warehouse. Adaptada de (Kimball & Ross, 2013).

1. **Fontes de dados:** os dados são provenientes de múltiplas fontes de dados, nomeadamente, ERP's, CRM's *(Customer Relationship Management)* ou outras fontes de dados externas à organização;
2. **Área de Estágio e Processos de ETL** (*Extract Transform and Load)***:** os dados são carregados para uma área de estágio para que posteriormente sejam efetuados os processos de ETL. A fase de extração é realizada através da leitura e compreensão dos dados para a área de estágio. A fase de transformação requer um conjunto de transformações, nomeadamente, limpeza dos dados (correção de valores inválidos, alteração do formato dos atributos, entre outros). Por fim, a última etapa diz respeito carregamento para o DW*;*
3. **Área de apresentação:** é o local no qual os dados se encontram organizados e armazenados para possíveis consultas por parte dos utilizadores finais. Estes dados podem

posteriormente ser distribuídos pelos diversos *Data Marts,* dependendo das necessidades da organização;

4. **Aplicações *Business Intelligence:*** Conjunto de funcionalidades que podem ser fornecidas aos utilizadores para o suporte à tomada de decisão.

Existem na literatura diferentes propostas de arquiteturas para DWs. As organizações devem alinhar a seleção da arquitetura com as suas necessidades organizacionais (Vaisman & Zimányi, 2012).

Através da Figura 16 é possível observar superficialmente vários tipos de arquitetura retiradas do trabalho de Gardner (1998), no entanto, o autor pressupõe que a organização deve optar por uma arquitetura que melhor satisfaz as suas necessidades, a título de exemplo, ou pela implementação de um DW organizacional, ou pela implementação de *Data Marts* independentes, ou ainda pela implementação de *Data Marts* dependentes do DW organizacional (Figura 16).

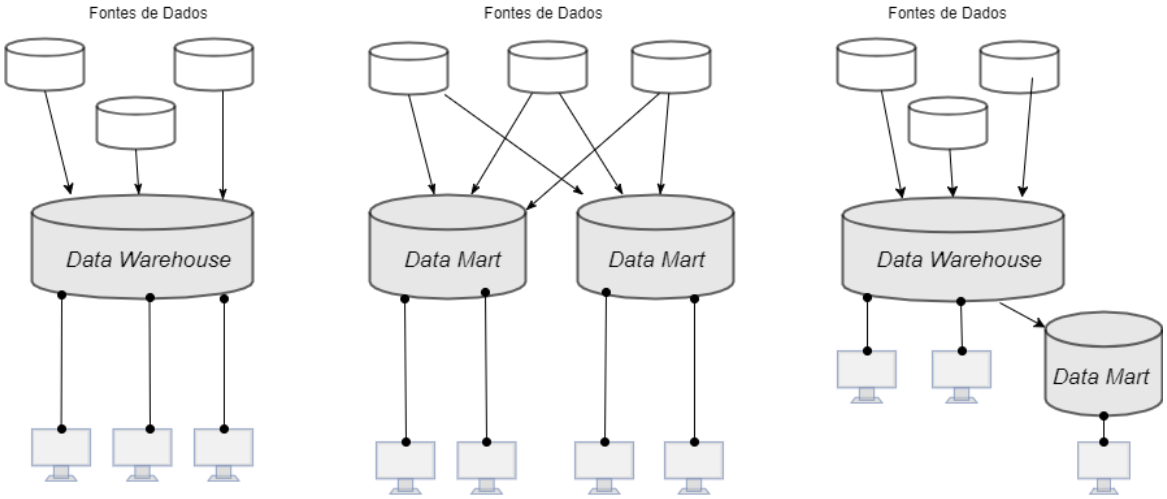

Figura 16. Arquitetura para a implementação de um Data Warehouse e *Data Marts. A*daptado de *(Gardner, 1998)*.

2.2.3. Sistemas Operacionais vs Sistemas Analíticos

Numa organização é possível identificar sistemas OLTP (*Online Transaction Processing)*, cuja finalidade se centra no registo de transações em tempo real que ocorrem no seu funcionamento diário (registo de novos produtos, vendas, encomendas, entre outros) (Chaudhuri & Dayal, 1997; Qin, Qian, & Zhao, 2015).

Um *Data Warehouse*, caracterizado anteriormente, é tipicamente apresentado como um sistema OLAP (*Online Analytical Processing*), cuja finalidade é suportar a tomada de decisão de executivos, gestores e analistas (Sá, 2010; Santos & Ramos, 2017; Vaisman & Zimányi, 2012).

Através dos contributos de Gardner (1998), Sá (2010), Santos e Ramos (2017), é possível sumarizar, através da Tabela 1, as principais diferenças entre uma base de dados operacional e um *Data Warehouse.*

*Tabela 1. Bases de dados operacionais vs Data Warehouses. A*daptada *de (Sá, 2010; Santos & Ramos, 2017).*

Bases de Dados Operacionais	***Data Warehouses***
Aplicações informáticas operacionais	Analistas de negócio e administradores executivos
Objetivos operacionais (OLTP)	Suporte à tomada de decisão (OLAP)
Acessos de leitura e escrita	Acessos maioritariamente só de leitura
Acesso a poucos registos de cada vez	Acesso a muitos registos de cada vez
Dados atualizados em tempo real	Carregamentos periódicos de mais dados
Estrutura otimizada para atualizações	Estrutura otimizada para processamento de questões

A par da síntese apresentada na Tabela 1, é importante realçar que apesar de os *Data Warehouses* apresentarem inúmeras vantagens perante os sistemas operacionais, com o surgimento do fenómeno *Big Data* a computação de cubos OLAP com diferentes volumes, velocidades e variedades de dados tem-se revelado um desafio neste domínio, originando os *Big Data Warehouses* (Chen et al. 2012).

2.2.4. *Big Data Warehouse*

Atualmente, o fenómeno caracterizado de *Big Data* tem estimulado grandes desafios para a execução de *Data Warehousing*, isto é*,* as regras inerentes aos dados relacionais não podem ser aplicadas em imagens ou vídeos, ou em alguns formatos de dados provenientes de sensores (Golfarelli & Rizzi, 2009; Vaisman & Zimányi, 2012). Em cenários cada vez mais competitivos e modernizados, a análise de dados não estruturados fornece informação de caracter valioso para as organizações, conduzindo à obtenção de vantagens competitivas (Russom, 2011; Wang, Can, Kazemzadeh, Bar, & Narayanan, 2012). Além disso, *Open Data* permite às organizações produzirem novas oportunidades com o objetivo de fortalecer a economia. A título de exemplo, uma organização enquadrada no setor da

educação é capaz de fornecer informação a respeito da situação das escolas e universidades, suportando as escolhas do futuro de um aluno (Tria et al. 2014).

Porém, a análise de dados com as características intrínsecas a *Big Data* requerem a adoção de *Big Data Warehouses* que permitam a extração e a produção de informação de valor, com o propósito de ser utlizada em processos de tomada de decisão (Golfarelli & Rizzi, 2009; Tria et al. 2014).

Data Warehouses tradicionais são tipicamente caracterizados de acordo com duas metodologias (Tria et al. 2014):

- **Orientado aos dados:** consiste na definição do seu modelo multidimensional em conformidade com os modelos das suas fontes de dados;
- **Orientado a requisitos:** consiste na definição do seu modelo multidimensional em conformidade com os seus objetivos e necessidades de negócio.

Tal como previsto, ambas as metodologias tradicionais não são capazes de satisfazer os desafios que surgem em contextos de *Big Data.* No entanto, Tria (2014) reconhece que cada uma das metodologias possuem vantagens importantes. Assim, para suportar um *Data Warehouse* em contextos de *Big Data*, o autor adota uma metodologia **híbrida** que considera as características mais importantes de ambas as metodologias tradicionais.

Resumindo as várias abordagens para projetar um *Big Data Warehouse,* o autor menciona também uma abordagem **incremental***,* que é baseada em abordagens ágeis e uma abordagem **automática** que consiste em garantir uma conceção rápida, ainda que sejam acrescentadas novas fontes de dados (Tria et al. 2014). As metodologias devem adotar novos modelos lógicos, como os modelos utilizados para bases de dados NoSQL, de modo a garantir a escalabilidade, flexibilidade e um desempenho superior (Cattell, 2011; Giorgini, Rizzi, & Garzetti, 2008).

Na mesma linha de trabalho, Costa e Santos (2017) fornecem uma definição mais detalhada de *Big Data Warehouse*, definindo-o como um sistema de armazenamento escalável, de alta performance e flexível, capaz de lidar com o aumento do volume, variedade e velocidade de dados.

Consequentemente, o conceito de *Big Data Warehouse* emergiu como um tópico de interesse dentro dos sistemas *Big Data*, proporcionando novas características relevantes, nomeadamente (C. Costa & Santos, 2017):

- Armazenamento e integração flexível, incluindo dados internos e externos às organizações;
- Elevado desempenho com respostas em *real-time;*
- Elevada escalabilidade a um custo reduzido através de *hardware* comum;

- Processamento de dados altamente distribuído;
- Cargas de trabalho complexas (exemplo: consultas de *queries ah hoc,* execução de modelos de *Data Mining* em grandes volumes de dados).

Nas metodologias mencionadas anteriormente, por vezes, é notória a falta de diretrizes a partir das quais os profissionais desta área se possam orientar. Ocasionalmente, os autores deixam o seu contributo relativamente ao tipo de arquitetura que deve ser tido em conta, ou o método de implementação. Apesar de serem contributos importantes, por vezes é necessário fornecer componentes lógicos e a representação do fluxo de dados entre os diversos componentes, com o propósito de auxiliar os profissionais na implementação de um *Big Data Warehouse*.

Assim, a Figura 17 ilustra a arquitetura de um *Big Data Warehouse*, proposta por Costa e Santos (2017), na qual se destacam os diversos componentes lógicos e os fluxos de dados inerentes. Esta arquitetura está representada em três camadas principais, nomeadamente:

- **Extração, Preparação e Enriquecimento dos dados:** os dados extraídos são provenientes da componente *data provider* (exemplo: sensores, sistemas transacionais, sistemas legados, entre outros). A sua extração pode ser concretizada através de duas formas distintas, nomeadamente, mecanismos *streaming* e em *batch*. Os dados que se apresentam em *batch* são primeiramente armazenados no sistema de ficheiros distribuído (*sandbox storage)*, não sofrendo qualquer tipo de processamento. Ao invés, dados extraídos através de mecanismos de *streaming* são imediatamente processados (preparados e enriquecidos), com o objetivo de apresentarem mais valor na eventualidade de serem analisados em tempo-real. Apesar disso, em contextos específicos, os dados extraídos através de mecanismos *streaming* podem também ser armazenados no sistema de ficheiros distribuído como um caminho opcional;
- **Plataformas de Organização e Distribuição dos dados:** o *Big Data Warehouse* apresenta três componentes de armazenamento diferentes, classificados em duas diferentes categorias: sistema de ficheiros e armazenamento indexado. O componente *sandbox storage* é uma área de armazenamento altamente flexível, sem a necessidade de definir qualquer tipo de metadados. Por sua vez, o componente *batch storage* é um sistema de armazenamento indexado para armazenar e processar *Big Data* através de ficheiros *batch*. Ao invés do componente *sandbox storage*, este componente é alicerçado pela definição de metadados para suportar uma análise estruturada. Por fim, o componente *streaming storage*, de igual forma que o componente anterior, é um sistema de armazenamento indexado, que lida com os dados que chegam em

formatos *streaming*. Estes dois últimos componentes encontram-se indexados para suportar as cargas de trabalho do *Big Data Warehouse* da forma eficaz e eficiente;

- **Análise, Visualização e Acesso aos Dados:** o componente principal desta camada corresponde ao elemento *distributed query engine*, que é responsável por consultar os dados armazenados nos sistemas indexados, combinando os dados em *streaming* e em *batch* numa mesma consulta. Este componente é utlizado para explorar os dados armazenados no *Big Data Warehouse*, porém, também fornece dados aos componentes de *data science* e de visualização de dados. Por fim, o componente *Data Provider* representa todos os stakeholders interessados em consumir dados provenientes do *Big Data Warehouse* (analistas de dados, utilizadores externos, sistemas externos, gestores operacionais ou gestores administrativos*)*.

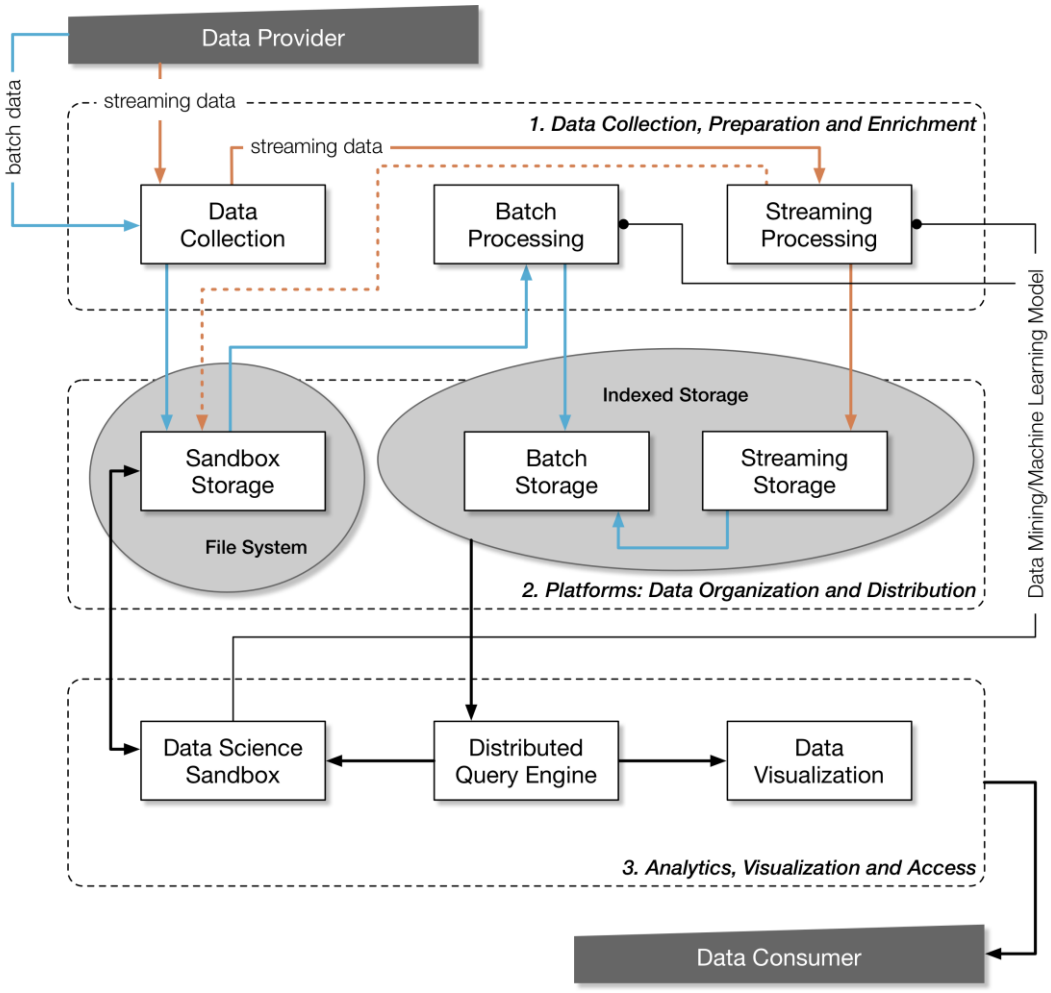

Figura 17. Arquitetura de um Big Data Warehouse. Retirado de (C. Costa & Santos, 2017).

O foco desta dissertação está enquadrado com a primeira camada da arquitetura presente na Figura 17, evidenciando o processo de enriquecimento através da obtenção do perfil dos dados que estão a chegar ao *Big Data Warehouse* e perceber de que forma é que os dados que se encontram a

chegar através de mecanismos de *streaming* ou *batch*, em vários formatos, se podem integrar com os dados que subsistem na segunda camada da arquitetura proposta por Costa e Santos (2017).

2.2.5. Modelos de Dados

Os modelos de dados são o elemento central na implementação e desenvolvimento de sistemas de informação, de modo a assegurar que todas as necessidades de dados são asseguradas (C. Costa & Santos, 2016).

Em contextos tradicionais, os modelos de dados relacionais são definidos através de regras acerca dos requisitos que estes modelos devem considerar. No entanto, num contexto de *Big Data*, devido particularmente às características das bases de dados NoSQL, as tarefas de modelação de dados sofrem algumas mudanças, visto que, estas bases de dados são caracterizadas por serem *schema-free*. Os modelos, neste contexto, são implementados considerando as consultas a que devem responder (Durham, Rosen, & Harrison, 2014).

Segundo os autores Elmasri e Navathe (2010), existem vários modelos de dados que se caracterizam de acordo com os conceito que utilizam para descrever a estrutura da base de dados. Encontram-se na Tabela 2, as categorias de modelos que são consideradas pelos autores.

Tabela 2. Modelos de dados e conceitos associados. Baseado em (Elmasri & Navathe, 2010).

Modelo de Dados	**Descrição**	**Conceitos**
Concetual	Caracterizado também como modelo de alto nível, cuja função é fornecer conceitos que são próximos da forma como os stakeholders compreendem os dados	Entidades; Atributos; Relações;
Lógico	Caracterizado também como modelo de implementação, fornece conceitos que podem ser compreendidos pelos utilizadores, apresentando também detalhes de como os dados se enquadram no modelo de dados	Estrutura dos registos que dependem do sistema de base de dados a utilizar;
Físico	Caracterizado também como modelo de baixo nível, detalhando como os dados são armazenados no sistema de armazenamento	Formato dos registos; Ordem os registos;

Na linha da implementação dos modelos em *Data Warehouses*, são destacados três níveis de abstração que se encontram representados através da Figura 18, nomeadamente o modelo concetual, o modelo lógico e o modelo físico (Dehdouh, Bentayeb, Boussaid, & Kabachi, 2015).

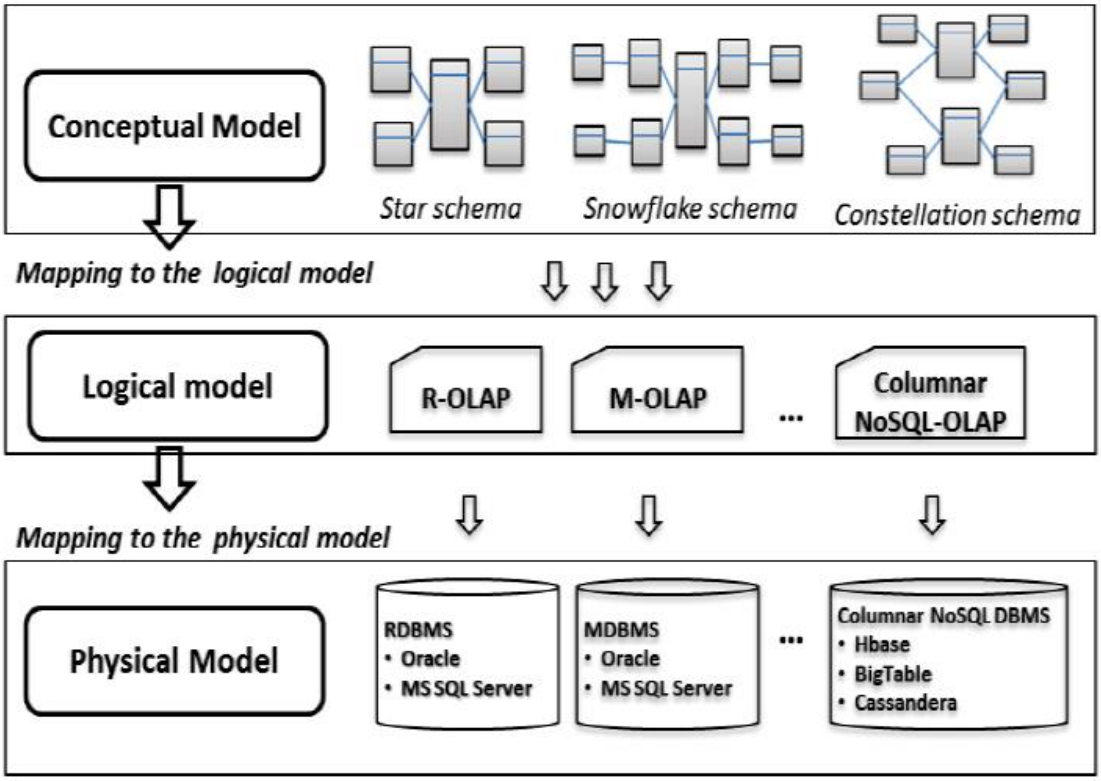

Figura 18. Processo de implementação em Data Warehouses. Retirado de *(Dehdouh et al. 2015)*.

Na ótica de Dehdouh (2015), os **modelos conceptuais** dizem respeito ao modelo multidimensional referente a um *Data Warehouse.* Portanto, de acordo com a Figura 18, a modelação multidimensional de um *Data Warehouse* é conseguida através da implementação de esquemas em estrela, em floco de neve ou em constelação (Kimball & Ross, 2013; Santos & Ramos, 2017). As características de cada um destes é apresentada na Tabela 3.

Tabela 3. Esquemas de modelação multidimensional de um Data Warehouse. Baseado em *(Santos & Ramos, 2017)*.

Modelo	Caracterização
Estrela	Um esquema em estrela integra uma única tabela de factos, o centro da estrela e múltiplas tabelas de dimensão ligadas à tabela de factos. Entre as tabelas de dimensão e as tabelas de facto existe, tipicamente, uma relação de um para muitos *(1: n).* A tabela de facto corresponde, tipicamente, ao assunto que se pretende

Modelo	Caracterização
	analisar, normalmente a uma componente de negócio (vendas, compras, encomendas, entre outras)
Floco de Neve	Um esquema em floco de neve é um esquema cujas dimensões estão parcial ou completamente normalizadas. O esquema em estrela e o esquema em floco de neve são semelhantes face ao conteúdo dos dados, embora o esquema em floco de neve apresente uma estrutura mais complexa. Este esquema por vezes é associado à dificuldade na sua interpretação e perda de desempenho no processamento de questões devido à sua normalização
Constelação	Um esquema em constelação é um esquema que integra múltiplas tabelas de factos que partilham dimensões comuns. É habitualmente visto como como um conjunto integrado de esquemas em estrela

A ligação entre a camada referente ao modelo concetual e a camada referente ao modelo lógico é efetuada de acordo com três abordagens: ROLAP *(Relational OLAP),* MOLAP *(Multidimensional OLAP),* HOLAP *(Hybrid OLAP).* Porém, como se pode observar em secções anteriores, a computação de **modelos lógicos** tradicionais em *Data Warehouses* não são adequados para grandes volumes de dados presentes em contextos de *Big Data.* Os autores Costa e Santos (2016) afirmam que em contextos de *Big Data*, nos quais as bases de dados NoSQL são tipicamente utilizadas, os modelos de dados lógicos são caracterizados por serem *schema-free*, por outras palavras, diferentes linhas numa tabela podem conter diferentes colunas de dados.

As bases de dados NoSQL caracterizadas por serem *schema-free*, não sendo necessárias grandes preocupações com a estrutura dos dados, no entanto, em contextos *Big* Data pode ser necessário adicionar estrutura aos dados quando as tarefas analíticas o exigem (**modelo físico**). Deste modo, dependendo do contexto, podem ser transformados num modelo de dados baseado em grafos, por exemplo, no Neo4J quando se trata de armazenar dados complexos, ou num modelo de dados baseado em colunas, como é o caso do HBase (C. Costa & Santos, 2016).

Com base nas afirmações anteriores, em cenários *Big Data,* existe a necessidade de adicionar uma estrutura aos dados nos momentos em que as tarefas analíticas são executadas. Perante essa necessidade, Costa e Santos (2016), no seu trabalho, propõem um processo de estruturação de um modelo de dados no Hive através da transformação de um modelo de dados multidimensional, utilizado

num *Data Warehouse* tradicional. A vantagem desta transformação é a capacidade de considerar todas as necessidades da organização, expressas no seu modelo de dados operacionais (OLTP) e utilizar essa informação para identificar esquemas de dados adequados para suportar tarefas analíticas em *Big Data*.

No âmbito desta dissertação, o foco está relacionado com o acesso e o processamento de grandes volumes de dados da forma mais rápida possível, ou seja, existe a necessidade de execução de um conjunto de algoritmos para a avaliação do perfil dos dados provenientes das fontes de dados, por vezes complexos, com o propósito de estabelecer relações semânticas entre as diversas tabelas presentes no *Big Data Warehouse.* Assim, é necessário a seleção do modelo de dados, da base de dados e das tecnologias adequadas para otimizar este tipo de processamento.

2.3. Governança de Dados

Dada a quantidade de dados que habitualmente são guardados num BDW e à necessidade de conhecer esses dados, a área de governança de dados descreve como as organizações gerem todos os dados que passam pelos seus processos de negócio. As organizações necessitam cada vez mais de governança de dados para compreender algumas questões inerentes a este domínio, tais como: O que sabemos sobre estes dados? Qual a sua proveniência? Como podem ser utilizados? Como se comportam face às políticas e regras da organização?

Na ótica de Smallwood (2014), a governança de dados não se trata de uma aplicação técnica, mas sim de um conjunto de processos, funções, padrões e métricas que garantem a utilização eficaz e eficiente dos dados, isto é, a governança de dados fornece uma abordagem holística para gerir e melhorar a informação interna e externa à organização.

Através da Figura 19, é possível observar que a governança de dados inclui o cruzamento de diferentes domínios, nomeadamente segurança, processos, privacidade, integridade, usabilidade, interoperabilidade e toda a gestão interna e externa dos fluxos de dados dentro de uma organização (DAMA, 2017). Ao longo desta secção e no âmbito desta dissertação, o foco está subjacente a um conjunto de domínios de governança de dados: *Data Quality, Data Profiling e Metadata Management*.

Big Data, como mencionado anteriormente, é um dos fenómenos que está a emergir e a transformar as organizações (Zikopoulos, 2011). Porém, outros domínios estão a modifica-las tal como *Data Science* (Waller & Fawcett, 2013). Para ambas as tendências, o maior desafio centra-se na transformação dos dados que, tipicamente, se podem apresentar incorretos ou desatualizados (Berti-Équille & Borge-Holthoefer, 2015). A confiabilidade dos resultados obtidos neste tipo de domínios depende da confiabilidade dos dados analisados numa determinada tarefa. Esta confiabilidade é avaliada

utilizando métricas de qualidade de dados, que fornecem aos seus consumidores métricas sobre o nível de precisão que os seus resultados podem produzir (Ardagna, Cappiello, Samá, & Vitali, 2018).

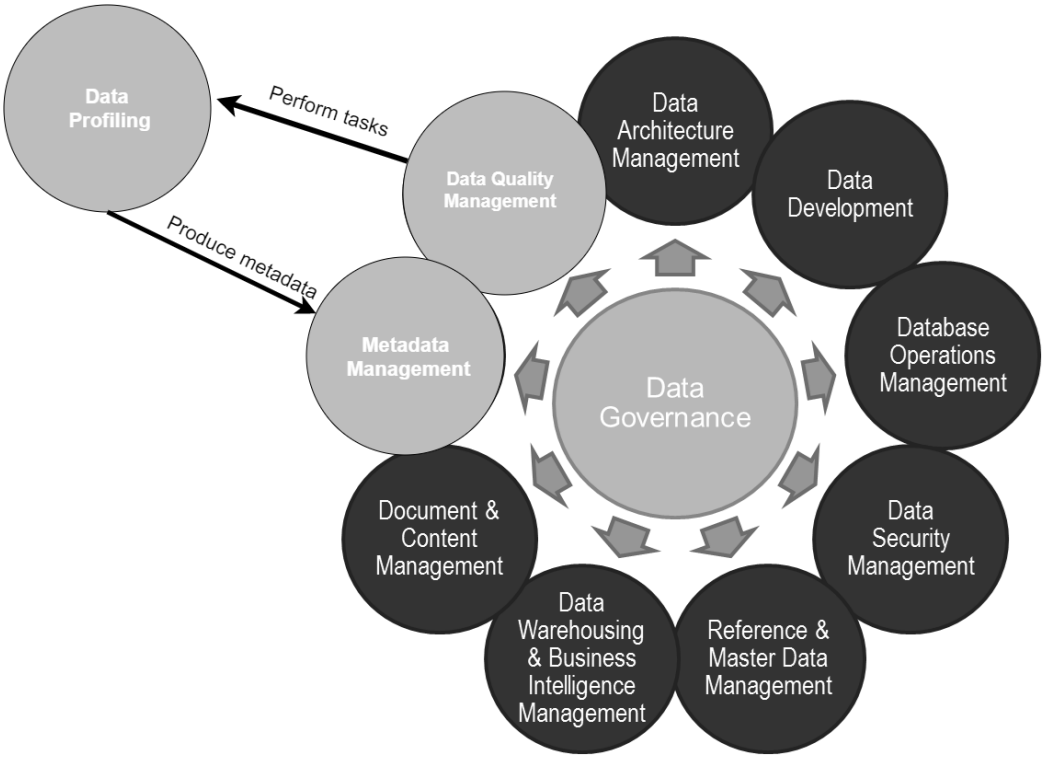

Figura 19. Domínios de Governança de Dados. Baseado em (DAMA 2017).

A comunidade apresenta um vasto conjunto de dimensões relacionadas com a **qualidade de dados** em contextos *Big Data*. No entanto, as mais utilizadas dizem respeito à precisão, consistência, completude e atualidade dos dados (Ardagna et al. 2018; Hazen, Boone, Ezell, & Jones-Farmer, 2014; Juddoo, 2015).

De seguida, serão descritas todas as dimensões mencionados anteriormente:

- **Precisão:** grau em que os dados são equivalentes aos seus valores reais, ou seja, esta dimensão é avaliada através da comparação de valores externos com valores que são conhecidos (ou considerados) como sendo corretos;
- **Consistência:** refere-se à não conformidade das regras semânticas definidas sobre um conjunto de dados e tipicamente expressadas por restrições de integridade;
- **Completude:** definida como o grau em que uma extração de dados inclui todos os valores de dados que eram expectáveis;
- **Atualidade dos dados:** expressa o grau em que os dados são atuais para uma determinada tarefa (dimensão importante para cenários de conversão monetária por exemplo).

Juddoo (2015) refere que *Data Quality* está associado a um conjunto de atividades, cuja finalidade é melhorar a qualidade dos dados com base nas dimensões identificadas. Algumas atividades mais citadas são:

- ***Data Profiling:*** análise das fontes de dados com o propósito de produzir metadados sobre conjuntos de dados;
- ***Data Cleansing:*** conjunto de técnicas cujo principal objetivo é transformar os dados com base nas regras de negócio;
- **Definição das regras de qualidade de dados:** regras a utilizar para determinar se existem problemas com determinados conjuntos de dados e seleção dos conjuntos de dados para a atividade *Data Cleansing.*

Ardagna (2018) refere que o grande desafio de *Big Data* está relacionado com a transformação dos dados em decisões que proporcionem valor na análise dos mesmos. Contudo, os dados extraídos podem apresentar informação desatualizada, incorreta e não confiável. Perante os desafios impostos em contextos *Big Data* e considerando as dimensões descritas anteriormente, Ardagna (2018) propõe uma arquitetura orientada aos serviços de qualidade dos dados, representada na Figura 20.

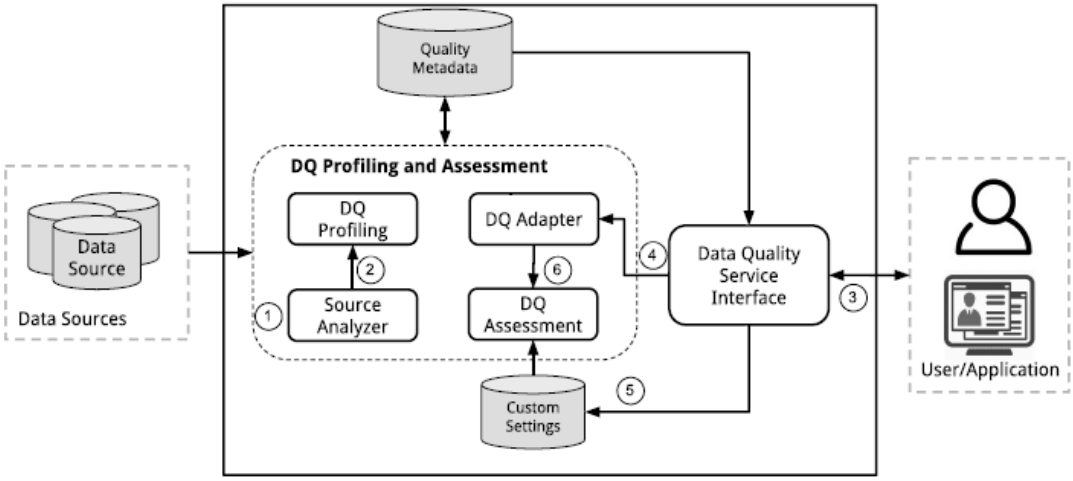

Figura 20. Arquitetura orientada aos serviços de qualidade de dados. Retirado de (Ardagna et al. 2018).

Perante a arquitetura de Ardagna (2018) e contextualizando com o tema desta dissertação, o foco está enquadrado no módulo ***DQ Profiling and Assessment*.** Este módulo é dividido em dois componentes principais*: DQ Profiling e DQ Assessment.*

O componente *DQ Assessment* é caracterizado pelo cálculo das dimensões da qualidade de dados, deste modo, as métricas são calculadas de acordo com o tamanho e o tipo de dados selecionado. Destacando o componente que vai ao encontro da visão desta dissertação, o ***DQ Profiling*** fornece um

conjunto de métricas capazes de avaliar e monitorizar a qualidade de um conjunto de dados (Naumann, 2014). De igual forma, é responsável pela produção de um conjunto de metadados que descrevem a fonte de dados e as suas principais características (por exemplo: máximo, mínimo, médias e distribuição de valores) (Ardagna et al. 2018).

Deste modo, partindo dos domínios da Figura 19, e com base no que foi mencionado por Ardagna (2018), surge a necessidade de explorar o conceito de ***Data Profiling*.**

2.3.1. *Data Profiling*

O conceito de *Data Profiling* é importante para os domínios relacionados com Governança de Dados, Qualidade de Dados e Gestão de Metadados, devido à necessidade de verificar a qualidade dos dados que, tipicamente, se apresentam num formato estruturado, semiestruturado e não estruturado (Dai et al. 2016). O *Data Profiling* é definido como um subconjunto de atividades do processo de Qualidade de Dados (Rodrigues, 2017). Sempre que um *data scientist* recebe um novo conjunto de dados, este necessita de inspecionar o seu formato, estrutura e algumas amostras de dados, com o objetivo de determinar o que o conjunto de dados pode proporcionar em termos de análise e de melhorias para os processos de negócio (Abedjan, Golab, & Naumann, 2015; Rodrigues, 2017). Desse modo, o *data scientist* desenvolve um nível de conhecimento sobre os dados que lhe permite de forma eficiente executar processos de armazenamento e processamento sobre os mesmos. A comunidade caracteriza estas ações de extração de conhecimento sobre um conjunto de dados como *Data Profiling* (Naumann, 2014).

Porém, como seria de esperar, o *Data Profiling* não se trata apenas de inspecionar a distribuição de valores, mas também de tarefas como a inferência do seu tipo de dados e a mineração de dependências, entre outras que são necessárias para obter uma compreensão total dos dados (Abedjan et al. 2015). Em contextos de *Big Data*, devido ao seu elevado volume, velocidade e variedade, os dados não conseguem ser processados através de técnicas tradicionais, mas sim através de mecanismos de *Data Mining* e *Machine Learning,* com o objetivo de explorar e minerar os dados. Assim, o *Data Profiling* ganha uma nova importância, sendo caracterizado como uma tarefa que determina quais o dados a minerar e como interpretar os resultados (Abedjan, Golab, & Naumann, 2017).

Os metadados são um componente importante, citando Agrawal (2011), "se apenas armazenamos os conjuntos de dados no repositório sem qualquer tipo de referência ou metadados é improvável que alguém o consiga encontrar, muito menos reutilizar. Com os metadados adequados haverá alguma esperança, no entanto, os desafios vão permanecer".

Abedjan (2015) e Naumann (2014) enumeraram um conjunto de casos de uso relacionados com *Data Profiling*, nomeadamente:

- **Exploração de dados:** administradores de bases de dados e investigadores são confrontados com novos conjuntos de dados sobre os quais não têm qualquer conhecimento. Em alguns casos, o seu formato e a sua estrutura de dados são totalmente desconhecidos, conduzindo a tarefas de *Data Profiling* para obtenção de conhecimento sobre os dados;
- **Integração de dados:** frequentemente, os conjuntos de dados que vão ser integrados são desconhecidos e é necessário obter conhecimento sobre a relação semântica entre as colunas e as tabelas, assim como, as suas dependências. Através do processamento de tarefas de *Data Profiling,* o foco é compreender a heterogeneidade dos dados através da sua estrutura e da semântica dos dados. Um caso de *Data Profiling* é o *Schema Matching,* isto é, encontrar correspondências semanticamente corretas entre os elementos de dois esquemas (Euzenat & Shvaiko, 2013);
- **Limpeza de dados:** o resultado da obtenção do perfil de dados pode ser utilizado para avaliar e monitorizar a qualidade inerente ao conjunto de dados, com o objetivo de fornecer conhecimento para a execução de processos de limpeza de dados;
- **Otimização de consultas:** tipicamente, os metadados gerados incluem o cálculo de várias métricas, tais como valores únicos e distribuição de valores, entre outros. Este tipo de metadados frequentemente faz parte das estatísticas básicas recolhidas pelas bases de dados. Segundo o autor, estes metadados auxiliam na otimização da execução de consultas à base de dados.

Porém, Juddoo (2015) indica um conjunto de desafios na execução dos casos de uso mencionados anteriormente em contextos de *Big Data,* nomeadamente*:*

- **Desempenho da computação:** a complexidade computacional associada ao *Data Profiling* depende do volume do conjunto de dados, devido à presença de processos que executam operações de forma a abranger todas as combinações possíveis, acarretando grandes cargas de trabalho;
- **Escalabilidade:** a escalabilidade dos métodos executados nas tarefas de *Data Profiling* é importante devido ao aumento do volume de dados se apresentar em constantemente crescimento, exigindo um processamento baseado em ambientes distribuídos;

A Relevância de Matching

O *Matching* retrata uma das tarefas associadas a um caso de uso em *Data Profiling,* nomeadamente a integração de dados. A integração de diferentes conjuntos de dados representa uma importante motivação para tarefas de *Matching* (Bellahsene, Bonifati, & Rahm, 2011)

Bernstein, Madhavan e Rahm (2011) argumentam que o *Matching* é a correspondência entre elementos de dois conjuntos de dados diferentes. O principal objetivo desta tarefa é a construção de um conjunto de dados unificado perante conjuntos desenvolvidos de forma independente.

De seguida, estão presentes um conjunto de técnicas de *Matching* propostas pela comunidade (Bernstein et al. 2011a; Madhavan, Bernstein, & Rahm, 2001):

- ***Schema matching:*** com base na comparação de esquemas de dados diferentes;
- ***Graph matching:*** com base na comparação das relações entre diferentes elementos;
- ***Usage-based matching:*** com base na análise das consultas realizadas à base de dados através dos *logs*, nos quais os utilizadores utilizam frequentemente cláusulas *join* entre diferentes conjuntos de dados;
- ***Linguistic matching:*** com base no nome ou descrição de um elemento utilizando técnicas de similaridade entre *strings* ou *substrings;*
- ***Using auxiliary Information:*** com base em dicionários, acrónimos e listas de incompatibilidade;
- ***Instance-based matching:*** com base em elementos similares. Os elementos podem assumir-se como similares se as suas estatísticas e metadados forem similares;
- ***Constraint-based matching:*** com base no tipo de dados, distribuição de valores, chaves estrangeiras e valores únicos, entre outros.

As técnicas descritas anteriormente vão ao encontro do principal objetivo (integrar diferentes conjuntos de dados) e, perante diferentes contextos, algumas tendem a fornecer melhores resultados, nomeadamente técnicas cujo processo de *Matching* se encontre focado na análise do conteúdo dos dados. A comunidade apresenta um conjunto de algoritmos de similaridade que são recorrentemente utilizados para a análise de conteúdo, nomeadamente a medida de similaridade de ***Cosine***, ***Jaccard***, ***Jaro-Winkler*** e ***Levenshtein*** (C. Li, Lu, & Lu, 2008; Shirkhorshidi, Aghabozorgi, & Wah, 2015; Xiao, Wang, Lin, & Shang, 2009). Na abordagem levada a cabo por Li et al. (2008), o autor avalia a performance das diferentes medidas de similaridade, de acordo com o limite imposto pelo utilizador na avaliação de diferentes entidades, nomeadamente *strings*, documentos e vetores. Por sua vez, Xiao et

al. (2009) desenvolveu um algoritmo utilizando as medidas de similaridade mencionadas anteriormente cujo objetivo era apresentar um *top-k* de pares similares entre dois conjuntos de dados diferentes. O autor menciona também que a seleção dos *top-k* pares similares apresenta um nível de redundância inferior, quando comparado com a abordagem de Li et al. (2008), na qual os pares similares são filtrados através do limite de similaridade definido pelo utilizador. Assim, em contextos nos quais o conhecimento dos dados é totalmente desconhecido, definir um *top-k* pares similares será a abordagem mais indicada, uma vez que o valor da similaridade associado varia de acordo com o seu contexto.

Todas estas medidas de similaridade encontram-se propostas na implementação prática da presente dissertação, na secção 4.3 e, como tal, a Tabela 4 apresenta as suas características.

Tabela 4. Medidas de Similaridade (Bao & DAi, 2016; C. Li et al. 2008; Xiao et al. 2009).

Medida	Descrição	Fórmula
Cosine	Medida de similaridade que mede o ângulo de *cosine* entre dois vetores num espaço vetorial. Se o ângulo for 0º a medida de similaridade será 1, o que representa que os vetores são paralelos e com o mesmo sentido. É tipicamente utilizado em cenários positivos variando o seu valor entre o intervalo [0,1]	$\cos(\mathrm{V1}, \mathrm{V2}) = \frac{V1.V2}{\lvert\lvert V1\rvert\rvert * \lvert\lvert V2\rvert\rvert}$
Jaccard	Medida de similaridade que expressa o grau de interseção entre dois conjuntos. O seu valor varia entre o intervalo [0,1]	$\mathrm{J}(\mathrm{V1}, \mathrm{V2}) = \frac{\lvert V1 \cap V2\rvert}{\lvert V1 \cup V2\rvert}$
Jaro-Winkler	Tipicamente utilizado na comparação entre duas *Strings.* O seu valor varia entre o intervalo [0,1]	$jW(S1, S2) = \frac{m}{3a} + \frac{m}{3b} + \frac{m-t}{3m}$ **Legenda:** **a** e **b -** comprimento associado às *Strings* **m -** Número de correlações entre caracteres **t -** Número de transposições
Levenshtein	Tipicamente utilizado na comparação entre duas *Strings.* O seu valor varia entre o intervalo [0,1]. esta medida tem como base o cálculo a distância de *Levenshtein*	$\mathrm{L}(\mathrm{S1}, \mathrm{S2}) = \frac{distanceLevenshtein(S1, S2)}{\max(S1, S2)}$ **Legenda:** **max (S1, S2):** comprimento máximo entre a String S1 e S2

Uma Visão Sobre os Metadados

Em relação aos metadados, e como mencionado anteriormente neste documento, estes são o resultado esperado do processamento de tarefas de *Data Profiling,* face a um conjunto de dados (Alserafi, Abelló, Romero, & Calders, 2016). Numa perspetiva diferente, os metadados são definidos como os dados que descrevem os dados (Naumann, 2014).

Existem várias formas de categorizar diferentes tipos de metadados: através da sua instância, da sua aplicação ou das suas propriedades semânticas (Naumann, 2014). A categorização de metadados com maior relevância na comunidade foi proposta por Naumann (2014), que se encontra ilustrada na Figura 21. O autor refere que podem ser gerados metadados em três diferentes categorias: para colunas individuais, para colunas múltiplas e para dependências.

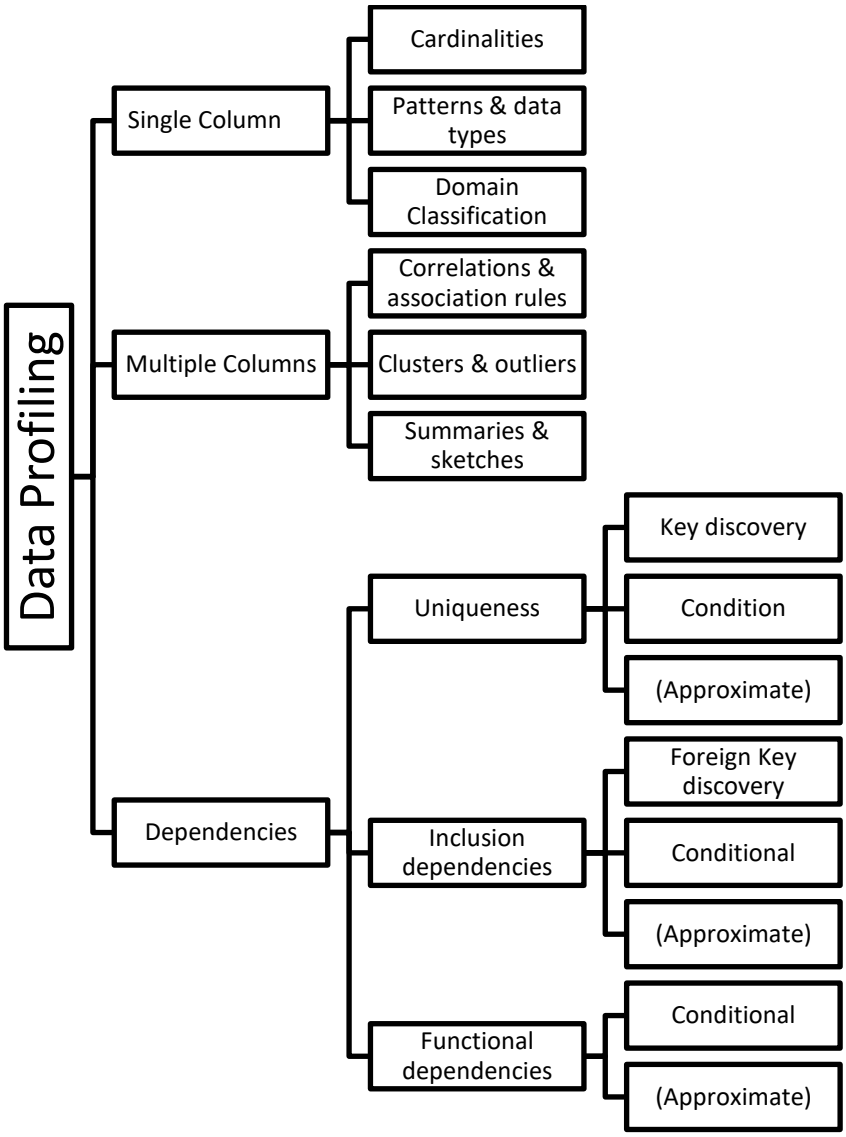

Figura 21. Classificação das tarefas de Data Profiling. Retirado de (Naumann, 2014).

A análise de colunas individuais caracteriza-se pela realização de processos básicos de *Data Profiling* e tipicamente, os metadados produzidos neste tipo de processos incluem o número de valores distintos, número de linhas do conjunto de dados, identificação de quartis, número de valores que se encontram a *null* e os valores máximos, mínimos e médios, dependendo do seu tipo de dados (*string, numeric, date)*. Eventualmente, em técnicas mais avançadas de *Data Profiling,* perante um conjunto de dados podem ser identificados os padrões e a distribuição de valores por atributos.

A análise de múltiplas colunas caracteriza-se pela identificação de regras de associação e correlações entre duas ou mais colunas. Além do mais, abordagens de *clustering* de dados conduzem à descoberta de conjuntos de dados similares e eventuais *outliers.*

Por fim, as dependências descrevem as relações entre as diversas colunas. No entanto, o autor menciona que esta categoria acarreta um conjunto de desafios, devido ao número de colunas que é analisado. Um objetivo comum no processamento das dependências é a identificação de chaves (primárias ou estrangeiras) para uma determinada tabela. Esse processo pode ser realizado através de dependências por inclusão e dependências funcionais. As dependências por inclusão revelam todos os valores ou combinações de um conjunto de dados "A" que estão presentes no conjunto de dados "B". Esta definição de dependência por inclusão é, tipicamente, associada a relações de similaridade por outros autores presentes na comunidade (Maccioni & Torlone, 2017). Por sua vez, dependências funcionais retratam que um conjunto de valores de uma coluna determina os valores de outra, isto é, os valores de dados do conjunto "A" determinam os valores dos dados do conjunto "B".

Apesar de ser um grande contributo para o domínio de *Data Profiling*, Naumann (2014) está consciente que esta categorização tradicional de metadados não se enquadra em ambientes *Big Data* devido ao elevado grau de heterogeneidade que os dados apresentam*.* No entanto, no contexto desta dissertação, com o objetivo de integração de dados estruturados num BDW, esta proposta apresenta-se para já adequada, uma vez que o foco será a realização de tarefas de *Data Profiling* no domínio da integração, com o objetivo de extrair as relações entre os diversos conjuntos de dados presentes no BDW. Neste domínio de integração, é importante compreender o grau de similaridade entre os diversos conjuntos de dados e de que forma é que se irá proceder à sua integração, tendo em conta a sua estrutura (Abedjan et al. 2017). Assim, técnicas de *Data Profiling* e *Schema Matching* são relevantes para este contexto, de forma a alinhar dois conjuntos de dados diferentes com o objetivo de encontrar atributos ou instâncias similares (Bernstein et al. 2011).

2.3.2. Trabalhos Relacionados

Abordagem Tradicional

Numa abordagem tradicional (fora de contextos *Big Data)*, os DWs são a principal fonte de dados para os processos de tomada de decisão. A integração de DWs tem vindo a ser um domínio discutido na comunidade devido ao elevado volume e heterogeneidade que os dados apresentam, reconhecendo a necessidade de abordagens para mapear diferentes esquemas através de diferentes componentes, tais como: dimensões, atributos e factos (Banek, Vrdoljak, & Tjoa, 2007).

Banek et al (2007) reconhece que os principais desafios neste domínio de integração se associam à utilização de diferentes nomes e estruturas para descrever a mesma informação. A título de exemplo, a dimensão que descreve um hospital poderá ser intitulada de "hospital" num componente DW e "clínica" em outro componente. Assim, o autor reconhece a necessidade de integrar diferentes esquemas de dados de forma automática, com o objetivo de reduzir a duração de processos complexos de integração num DW. Como tal, os autores propõem uma abordagem para calcular a similaridade semântica entre diferentes conjuntos de dados, através da taxonomia proveniente da WordNet, uma base de dados que contém um elevado número de palavras. No seu estudo, os autores concluem que esta abordagem oferece uma redução no processo de integração de diferentes esquemas num DW. No entanto, a plataforma WordNet não consegue abranger domínios específicos, sendo necessária a criação de ontologias por *experts* no domínio em questão.

A fim de melhorar o processo de tomada de decisão, as organizações tendem a utilizar, cada vez mais, dados externos à organização, em formatos estruturados, semiestruturados e não estruturados. Perante tal facto, Deb Nath, Hose, e Pedersen (2015) alertam que as ferramentas tradicionais de extração, transformação e carregamento de dados não estão adequadas para lidar com este tipo de dados devido, maioritariamente, à heterogeneidade que estes apresentam. As tecnologias DW e OLAP são executadas eficientemente quando se encontram aplicadas a dados estáticos e bem organizados em termos de estrutura. Assim, os autores propõem a SETL, uma *framework* para suportar a semântica entre os dados nos processos de ETL, de modo a aumentar a eficiência nos processos de integração e a colmatar os desafios relativamente à sua heterogeneidade, através da criação de ontologias entre os dados. Como resultado, os autores referem que a SETL fornece uma melhor produtividade, quando comparada a ferramentas de ETL tradicionais. Não obstante, uma vez que se encontra aplicada a um DW, na eventualidade de serem acrescentadas novas fontes de dados, pode levar, no limite, ao *redesign* do modelo multidimensional.

Abdellaoui e Nader (2015) associam um DW a um repositório de dados capaz de consolidar informação proveniente de diversas fontes através de processos de ETL. Os autores apontam para a taxonomia como um dos principais desafios na integração de dados heterogéneos e evidenciam a importância que a criação de ontologias apresenta no domínio da integração, com o objetivo de reduzir heterogeneidade e garantir uma integração automática dos dados, resolvendo problemas de sintaxe e semântica. A abordagem proposta pelos autores, representada na Figura 22, segue as seguintes fases: definição de requisitos, seleção das fontes de dados, conceção do modelo concetual e lógico do DW, execução dos processos ETL e análise de dados.

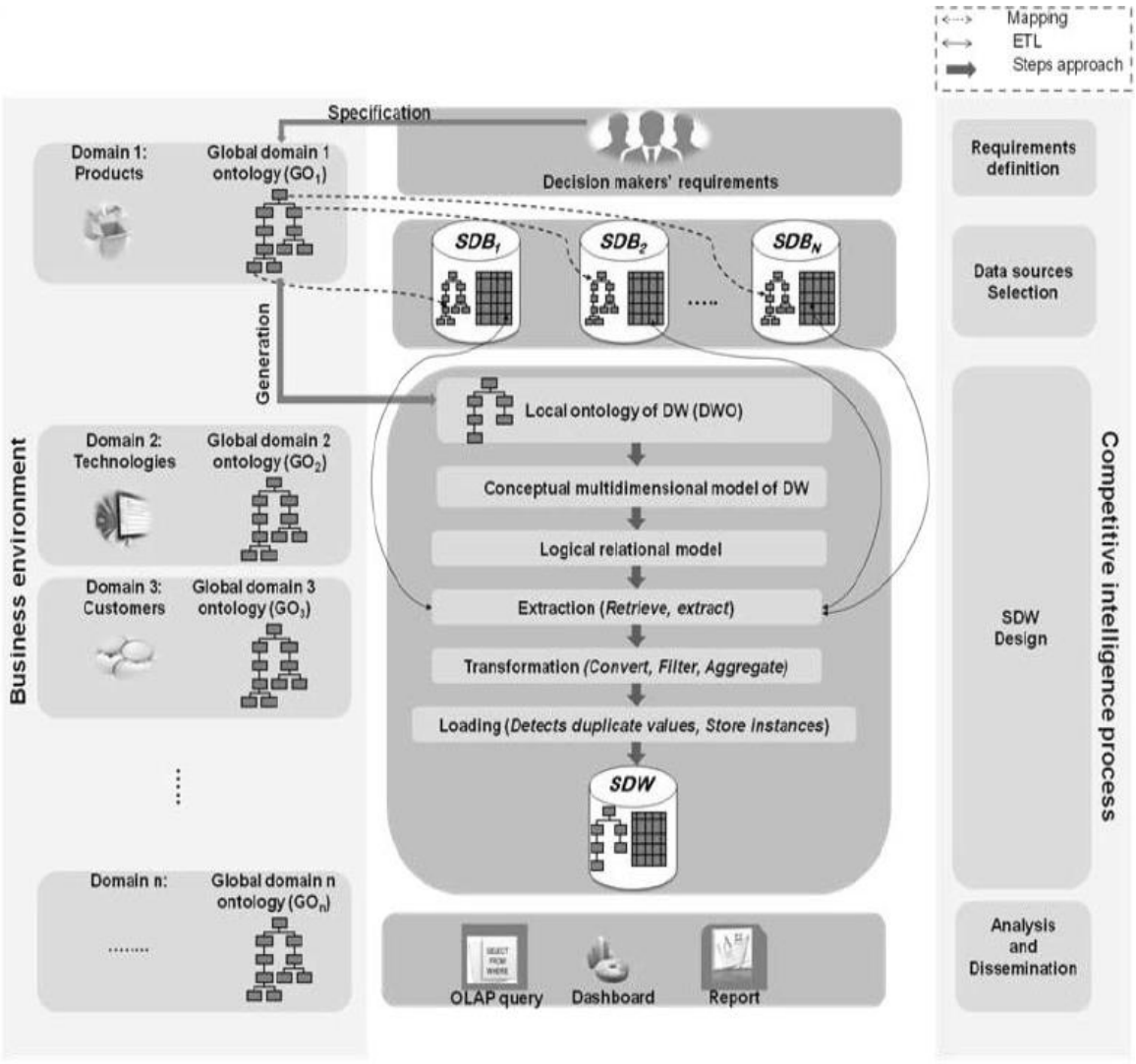

Figura 22. Abordagem proposta por Abdellaoui e Nader (2015).

Apesar do contributo prestado por Abdellaoui e Nader (2015) apresentar uma grande relevância para o domínio da integração de dados, a sua abordagem consiste na obtenção de um modelo multidimensional através da derivação de factos e dimensões provenientes das ontologias de um domínio

específico. Porém, no âmbito desta dissertação existe, a necessidade de focar em estratégias e mecanismos de Governança de Dados, nomeadamente o armazenamento de metadados com a informação associada aos dados, para a eventualidade de surgirem novas fontes de dados. A abordagem referente à Figura 22 rege-se por meio de um DW com uma estrutura rígida e a adição de novos conjuntos de dados pode levar ao *redesign* do modelo multidimensional, com a finalidade de considerar o novo conjunto de dados.

El Hajjamy, Alaoui e Bahaj (2018) alertam para a heterogeneidade presente nos dados quando estes são integrados, nomeadamente a nível sintático (terminologias dos dados), semântico (interpretação e significados diferentes) e estrutural (estrutura do conjunto de dados). Os autores contribuem com uma abordagem semiautomática para integração de diversas fontes de dados num DW. As integrações das diversas fontes de dados executaram-se através de técnicas de medição de similaridades a nível sintático, semântico e estrutural entre os diversos componentes. Os autores reconhecem que a sua abordagem acrescenta valor ao domínio em questão, mas, estão conscientes que os métodos de similaridade apresentados não são eficientes para as quantidades de dados em contextos *Big Data*, indicando esses pormenores como trabalho futuro.

Assumindo as abordagens anteriores, denota-se a necessidade da criação automática de DWs com base nas ontologias produzidas. No entanto, no âmbito desta dissertação é necessário não só a integração de novos conjuntos de dados com os dados que já subsistem no BDW, como também o armazenamento de metadados com informação sobre a similaridade e *joinability* entre os diferentes conjuntos de dados presentes no BDW que, numa perspetiva de Governança de Dados, irá auxiliar reduzir o tempo nos processos de integração e análise de dados.

Abordagem com *Big Data*

Em contextos tradicionais, as atividades de modelação, extração, limpeza e transformação de dados revelam-se fundamentais, mas conduzem a que o processo de análise seja infinito. Em resposta a esse desafio, as organizações orientadas a contextos *Big Data* estão a adotar estratégias ágeis que descartam qualquer pré-processamento antes de realizar o carregamento de um conjunto de dados para o *Data Lake,* um repositório de dados que se apresentam no seu formato natural (sem tratamento de dados) e sem objetivos de análise definidos (Maccioni & Torlone, 2017). Contudo, essas estratégias apenas reduzem o esforço inicial no domínio da gestão de dados, não descartando a necessidade, por exemplo, da execução de processos de *Data Quality* ou *Schema Matching.* Assim, é necessário um longo processo de preparação de dados antes de ser executada qualquer tarefa analítica (Castro Fernandez et al. 2017).

Maccioni e Torlone (2017) levaram a cabo um trabalho no qual é proposto uma *framework, KAYAK,* representada na Figura 23. O objetivo principal desta *framework* é auxiliar os *data scientists* na definição e otimização dos processos de preparação de dados. Os autores destacam ainda que esta *framework* é capaz de catalogar automaticamente metadados provenientes de um conjunto de dados, assim como identificar diferentes relações entre os conjuntos de dados presentes no *Data Lake*.

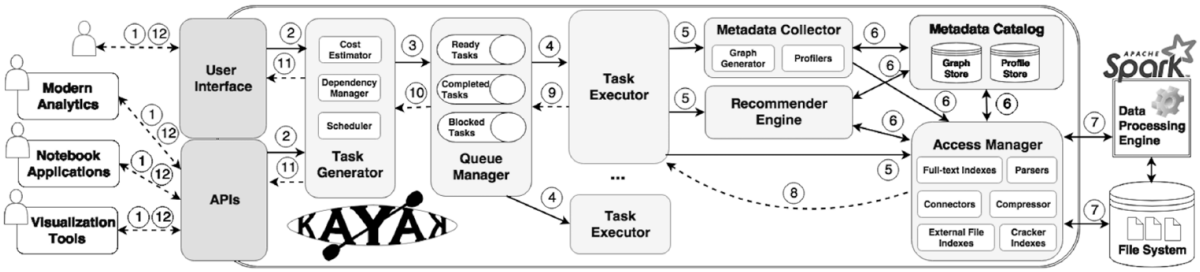

Figura 23. Framework lógica KAYAK. Retirado de (Maccioni & Torlone, 2017).

Com o propósito de compreender esta *framework*, os autores demonstram que esta apresenta um conjunto de primitivas para a preparação, exploração e análise de dados. Uma primitiva pode ser decomposta em uma ou mais tarefas (por exemplo, a primitiva P1 está dividida na tarefa Tb, Tc, Td).

No que diz respeito à gestão de metadados, a *framework* realiza estas tarefas através do processamento de um conjunto de primitivas com esse propósito. Esta *framework* extraí metadados *intra-dataset* e *inter-dataset*, armazenando os metadados num sistema de armazenamento centralizado com o objetivo de facilitar o seu acesso. A análise *intra-dataset* especifica relações entre atributos referentes ao mesmo conjunto de dados, ao invés da análise *inter-dataset* que especifica relações entre diferentes conjuntos de dados através da extração de um conjunto de propriedades, nomeadamente *joinability* e *affinity.* Intuitivamente, *joinability* mede a percentagem de valores comuns entre dois conjuntos de dados, enquanto que *affinity* mede a força semântica das relações entre atributos (Das Sarma et al. 2012).

De seguida, na *framework* presente na Figura 23, Maccioni e Torlone (2017) identificam um conjunto de componentes capazes de interagir entre si:

- ***User Interface* e *APIs:*** este componente permite aos utilizadores submeter primitivas e visualizar os seus resultados. Adicionalmente, este componente fornece um conjunto de APIs para que outras aplicações sejam capazes de interagir com a *framework;*
- ***Task Generator:*** este componente tem o propósito de receber as primitivas do utilizador e produzir as tarefas correspondentes. Este componente é de igual forma capaz de produzir um conjunto de informação sobre cada tarefa a executar, tais como os custos e dependências entre tarefas;

- ***Queue Manager:*** componente que assegura a execução assíncrona das tarefas, por outras palavras, se uma tarefa está dependente de outras tarefas para a sua execução, este componente assegura que a tarefa não será executada sem que as suas dependências sejam executadas;
- ***Task Executor:*** este componente é caracterizado pelo processamento de tarefas. Presente num ambiente escalável, tipicamente, é instanciado inúmeras vezes para uma computação distribuída e paralela;
- ***Metadata Collector:*** este componente é responsável pela extração de metadados produzidos no decorrer das tarefas associadas a *profiling;*
- ***Metadata Catalog:*** este componente é responsável pelo armazenamento de metadados. Encontra-se dividido entre *profile store*, cujo objetivo é armazenar os metadados *intra-dataset,* e *graph store,* que armazena metadados *inter-dataset;*
- ***Acess Manager:*** este componente é responsável por facultar o acesso aos dados, constituindo a *interface* entre o sistema de armazenamento e o componente *Data Processing Engine;*
- ***Recommender Engine:*** este componente ajuda o utilizador fornecendo um conjunto de consultas recomendadas e indo ao encontro dos interesses do utilizador.

Além desta *framework (KAYAK)*, têm surgido outras na literatura com o propósito de gerir os metadados sobre o conteúdo presente num *Data Lake*. No contributo de Alserafi et al (2016), os autores mencionam que o seu trabalho se alavanca devido ao reduzido leque de abordagens na extração e gestão de metadados. Como tal, Alserafi et al (2016) propõem uma *framework* para gerir os metadados ao longo da vida útil do *Data Lake*. Esta *framework* apresenta-se composta por três fases, representadas na Figura 24, através do seu processo *BPMN*.

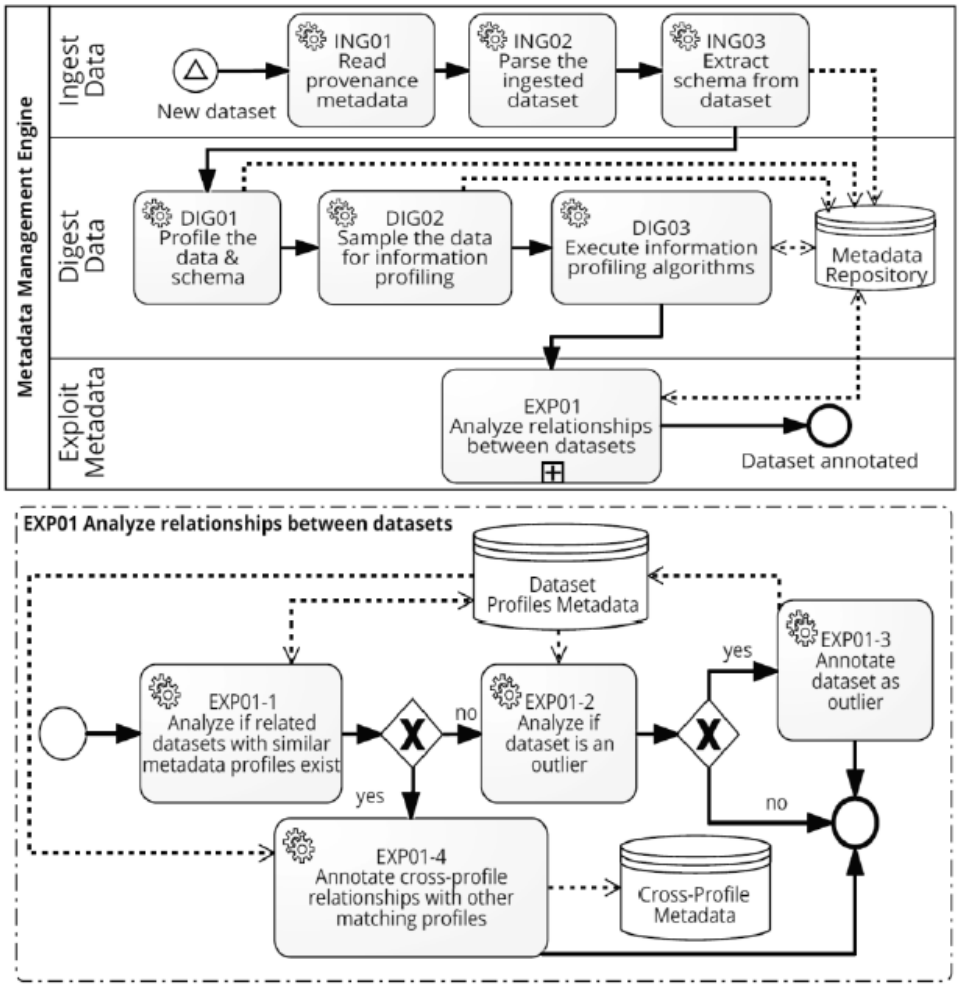

Figura 24. Processo BPMN da framework para a Gestão de Metadados. Retirado de (Alserafi et al. 2016).

1. A primeira fase diz respeito à ***Data Ingestion***, que se inicia aquando o carregamento de um novo conjunto de dados. Esta fase inclui a análise aos metadados inerentes ao conjunto de dados, a extração do esquema de dados e o armazenamento do conjunto de dados no *Data Lake* com os metadados do esquema associado (*schema metadata).*
    - Na atividade **ING01**, o conjunto de dados é localizado através dos metadados de origem. Em seguida, na atividade **ING02**, o conjunto de dados é analisado para verificar a veracidade estrutural. Na atividade **ING03**, o conjunto de dados é analisado para extrair os metadados associado ao seu esquema, sendo armazenados no respetivo repositório *(metadata repository).*
2. A segunda fase, ***Data Digestion***, é caracterizada pela análise dos dados armazenados no *Data Lake*, sendo realizados processos de *Data Profiling* e *Schema Profiling* capazes de gerar metadados, que posteriormente são armazenados no repositório.

- A atividade **DIG01** cria o *data profile* e o *schema profile* através de estatísticas simples e algoritmos de *profiling* (Naumann, 2014). De seguida, a atividade **DIG02** apresenta uma instância do conjunto de dados para melhorar a eficiência dos algoritmos de *profiling* executados na atividade **DIG03**. Na atividade DIG03, o conjunto de dados e o seu *profile* são comparados com os restantes conjuntos de dados presentes no *Data Lake.*

3. A terceira e última fase, ***Metadata Exploitation***, inclui a procura de relações entre todos os conjuntos de dados presentes no *Data Lake* através da similaridade dos metadados armazenados na fase *Data Ingestion.*
    - A atividade **EXP01** identifica as relações entre conjuntos de dados, através dos metadados armazenados no repositório. Esta subactividade inicia-se em **EXP01-1**, que verifica o grau de similaridade entre os atributos e os conjuntos de dados presentes no *Data Lake*. Se a similaridade entre esses atributos exceder o limite proposto pelo utilizador, o fluxo segue para a subactividade **EXP01-4**, que armazena o grau de similaridade entre os conjuntos de dados no componente *Cross-Profile Metadata*, um repositório para armazenamento de relações entre conjuntos de dados. Caso contrário, se não existirem conjuntos de dados similares o conjunto de dados será verificado na subactividade **EXP01-2** para averiguar se corresponde a um possível *outlier*. O conjunto de dados é classificado como *outlier* se não existirem atributos correspondentes (*matching attributes)* a outros conjuntos de dados presentes no *Data Lake*, sendo armazenada essa informação na atividade **EXP01-3**.

Hai, Geisler e Quix (2016) referem que em contextos de *Big Data*, a alta diversidade de fontes de dados, tipicamente, resulta em silos de informação, ou seja, conjuntos de sistemas de gestão de dados não integrados e com esquemas heterogêneos. Contudo, a informação com valoro, resulta, ou seja, da análise integrada da informação presente nesses silos. Assim, a gestão de metadados representa um domínio crucial, não só na integração de dados, mas também na sua compreensão, uma vez que a ausência de metadados torna difícil a compreensão da estrutura e semântica dos dados. De forma a colmatar tais desafios, Hai et al. (2016) propuseram o Constance, um sistema inteligente para a gestão de um *Data Lake.* O Constance apresenta dois principais objetivos, nomeadamente a gestão de metadados e o *matching* de metadados com base na semântica de dados.

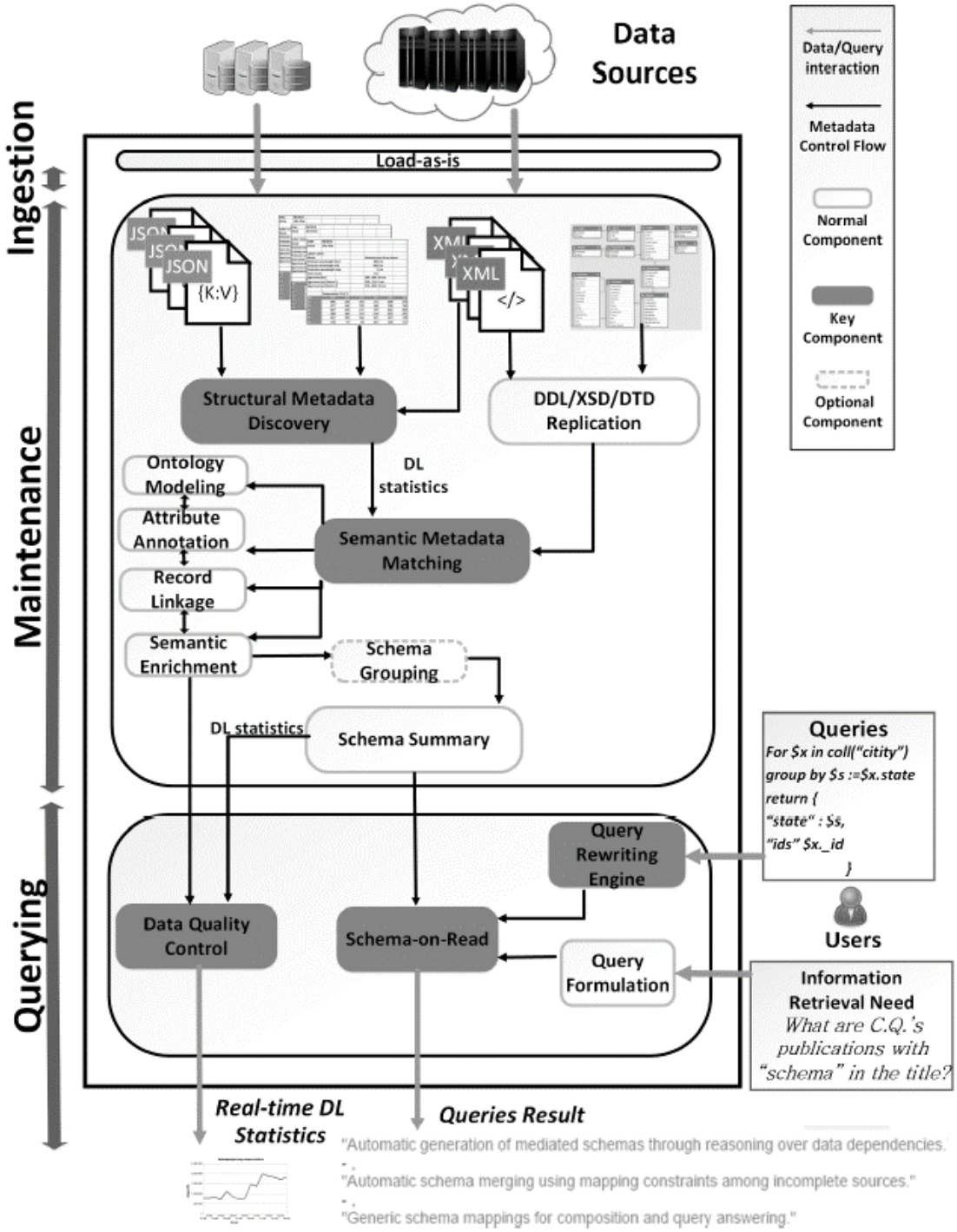

Figura 25. Constance: Intelligent Data Lake System. Retirado de (Hai et al. 2016).

A arquitetura proposta na Figura 25 utiliza o Hadoop como sistema de ficheiros e apresenta-se dividida em três principais camadas, nomeadamente Ingestão, Manutenção e Consulta:

- Ingestão – camada responsável pelo carregamento dos dados provenientes das diversas fontes heterogéneas para o *Data Lake;*

- Manutenção – camada responsável pela gestão de metadados. É efetuado o cálculo dos metadados implícitos aos dados (tipo de dados, relações e estatísticas) que, posteriormente, são agrupados através da similaridade dos metadados (componente *Schema Grouping);*
- Consulta – camada responsável por assegurar, não só a qualidade dos dados, mas também as funções de modelação, com o objetivo de adicionar estrutura aos dados e fornecer os mesmos às aplicações externas.

Zhu, Deng, Nargesian e Miller (2019) propõem o JOSIE, um novo algoritmo para a integração de dados. A quantidade de conjuntos de dados em *Big Data* é elevada e integrar estes conjuntos de dados num *Data Lake* requer grandes níveis de complexidade, exigindo assim uma elevada capacidade de computação. O seu principal objetivo é, perante uma tabela e uma coluna proposta pelo utilizador para executar operações *join,* encontrar conjuntos de tabelas presentes no *Data Lake* que sejam capazes de se integrar. Como tal, e com o objetivo de maximizar a performance do seu algoritmo, o autor executa operações *distinct* nas colunas que são comparadas, eliminando os valores repetidos resultando assim no aumento da performance do seu algoritmo. Este tipo de algoritmo é importante para a investigação, contudo, numa perspetiva automatizada é pretendido que o utilizador não seja necessário para escolher a tabela e a coluna que pretende fazer *join*, visto que existem contextos em que o conteúdo dos dados é totalmente desconhecido.

Em síntese, os trabalhos realizados respondem aos seus objetivos, mas pensando naquilo que se pretende desenvolver nesta dissertação existem questões às quais nenhum dos trabalhos anteriores consegue dar resposta. Tanto a *framework* proposta por Alserafi et al. (2016) como a *framework* proposta por Maccioni e Torlone (2017) estão associadas à gestão de metadados presentes num *Data Lake.* Ao contrário dos dados presentes num *Data Lake*, os dados presentes num BDW apresentam um propósito já definido. Assim, através dos resultados esperados da presente dissertação é possível analisar a similaridade entre as novas fontes de dados e os dados que já subsistem no BDW*,* derivando informação relevante para a sua integração e facilitando a sua evolução.

2.4. Mapa de Conceitos

Com o propósito de sintetizar o enquadramento concetual, é ilustrado, na Figura 26, um mapa com os conceitos mais relevantes abordados ao longo deste capítulo. Assim, dependendo da abordagem que foi realizada ao conceito, é apresentado um conjunto de informação sobre o mesmo, que engloba a sua definição, as suas características e eventuais desafios, problemas e oportunidades.

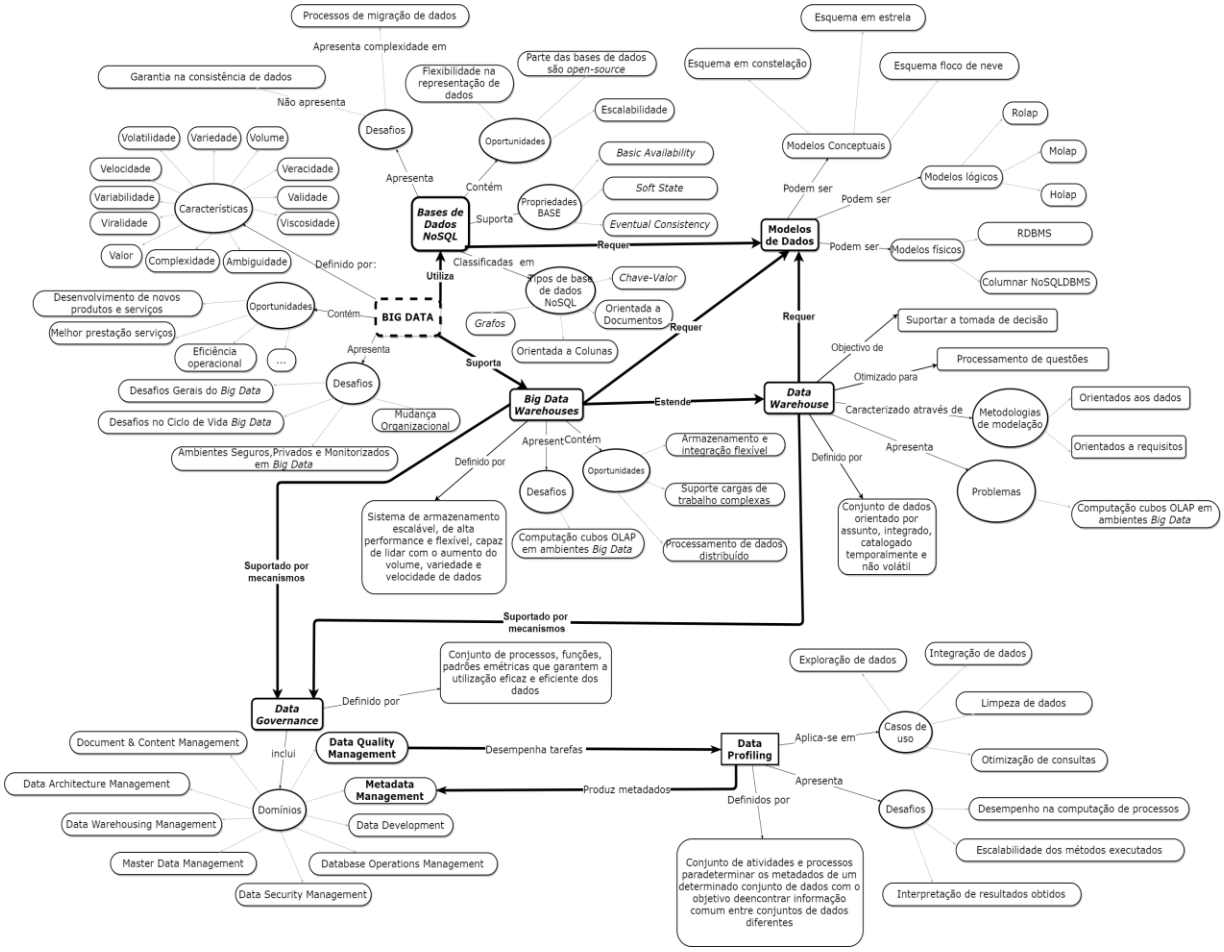

Figura 26. Mapa de conceitos do enquadramento concetual.

Este capítulo de levantamento do "estado da arte" tem como principal objetivo demonstrar a importância da temática onde se enquadra a presente dissertação, com o foco principal em analisar o estado da arte de alguns conceitos em contextos *Big Data* e compreender de que forma é que esta dissertação vem a contribuir para a evolução deste fenómeno. Assim, a Figura 26 relaciona os conceitos chave que representam uma base de conhecimento para a compreensão do contexto desta dissertação. Ao longo deste capítulo foi possível explorar diferentes conceitos e abordagens, mencionando que um número relevante de autores se focam na implementação de BDW e de bases de dados NoSQL, uma vez que o seu leque de opções é elevado, outrora, a governança desses BDW não se encontre

aprofundado na literatura. É possível observar alguns trabalhos em contextos de *Big Data*, no entanto, os objetivos desses trabalhos centram-se na gestão de metadados de um *Data Lake* que, apesar de ser utilizado para armazenar *Big Data*, se trata de um repositório de dados que se apresentam no seu formato natural (sem qualquer tipo de processamento) e sem qualquer objetivo analítico definido (sem propósito definido), não conseguindo assim responder aos objetivos propostos nesta dissertação.

3. Tecnologias para a Integração e Governança de Dados em *Big Data Warehouses*

Neste capítulo, será apresentado um conjunto de soluções para o desenvolvimento da ferramenta proposta. De forma a contextualizar, é apresentado o ecossistema Hadoop, assim como alguns dos seus principais componentes. Posteriormente, expõem-se as principais características do Hive e a sua arquitetura, terminando com uma abordagem detalhada ao Atlas como um repositório de metadados.

3.1. Ecossistema Hadoop

Como tem vindo a ser referido, *Big Data* apresenta dois principais desafios: como armazenar e processar dados que apresentam características *Big Data* e, mais importante ainda, como retirar informação desses dados com vista a obter vantagens competitivas (Krishnan, 2013).

Posto isto, o Hadoop surge como uma solução *open-source* baseada no GFS (Google File System) e *MapReduce* (Bakshi, 2012). O Hadoop caracteriza-se pela capacidade de armazenar e processar grandes volumes de dados, através do armazenamento e processamento distribuído por meio de um *cluster* concebido por intermédio de *commodity hardware* (Holmes, 2012; Krishnan, 2013). Esta solução, atualmente é adotada em grandes plataformas como Facebook, Yahoo e Twitter, é capaz de fornecer uma computação escalável, distribuída e confiável (Holmes, 2012).

Com o propósito de compreender esta solução, a Figura 27 representa superficialmente a sua arquitetura, na qual são destacados dois principais componentes: *HDFS* para armazenamento e *MapReduce* para questões computacionais.

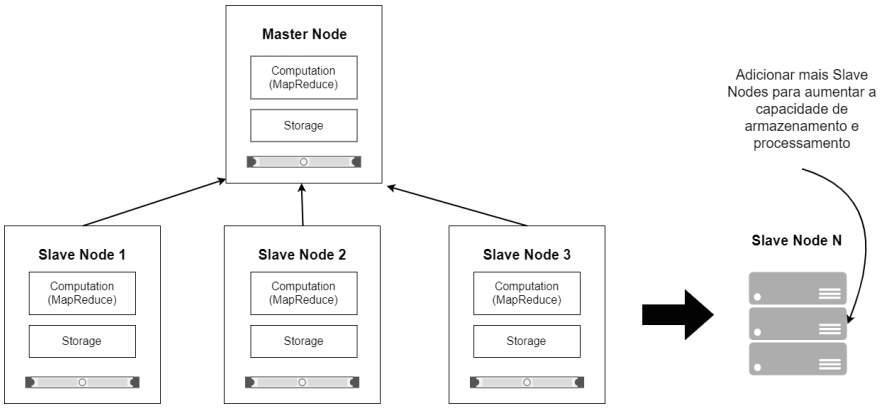

Figura 27. Arquitetura Hadoop. Adaptada de (Holmes, 2012).

O ecossistema Hadoop fornece uma panóplia de tecnologias definidas, principalmente, para funções de armazenamento, consulta e processamento. A Tabela 5 apresenta algumas destas tecnologias presentes neste ecossistema (Apache, 2018; Costa & Santos, 2017; Shvachko, Kuang, Radia, & Chansler, 2010).

Tabela 5. Tecnologias do ecossistema Hadoop (Apache, 2018; Shvachko et al. 2010).

Tecnologia	Caracterização
Ambari	Gere e monitoriza o *cluster* Hadoop, suportando outras tecnologias, nomeadamente, HDFS, *MapReduce,* Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig e Sqoop
Cassandra	Base de dados *multi-master*, escalável, sem pontos únicos de falha.
HBase	Base de dados orientada a colunas, distribuída e escalável, que suporta grandes volumes de dados estruturados
Hive	*Data Warehouse* que suporta o armazenamento de grandes volumes de dados, permitindo a execução de consultas *ad hoc* sobre os mesmos
ZooKeeper	Serviço centralizado, para estabelecer a coordenação entre aplicações distribuídas
Spark	Modelo de programação distribuída que suporta uma ampla gama de aplicações, incluindo mecanismos de ETL, *machine learning*, processamento em *stream* e computação de grafos
Ranger	O projeto Ranger pretende disponibilizar uma ferramenta centralizada de gestão de segurança do *cluster* Hadoop
Pig	Baseada na linguagem SQL, utilizada para a criação de programas para o Hadoop. Utiliza uma linguagem própria: Pig Latin
Tez	Modelo de programação de fluxo de dados, coordenado através do YARN, que fornece uma flexibilidade e um grande desempenho no processamento de dados em *batch* ou *streaming*
Flume	Utilizado para mover grandes volumes de dados provenientes de conjuntos de dados *streaming* para bases de dados centralizadas
Oozie	Serviço para agendar fluxos de trabalho presentes no Hadoop
Presto	Permite a execução de consultas dos dados armazenados no HDFS

3.1.1. Hadoop Distributed File System (HDFS)

O HDFS é um sistema de ficheiros distribuído, caracterizado pela sua elevada disponibilidade, escalabilidade e tolerância a faltas (Krishnan, 2013). Apresenta-se otimizado para mecanismos de leitura e escrita de ficheiros de grande dimensão (entre *gigabytes* até *petabytes*) (Holmes, 2012).

A arquitetura lógica do HDFS inclui dois principais componentes: *NameNode* e *DataNodes*. O ***NameNode*** caracteriza-se por representar um servidor *master,* cujo objetivo é gerir o sistema de ficheiros e monitorar o acesso aos ficheiros. Adicionalmente, o *NameNode*, através de metadados, gere o mapeamento dos blocos de dados pelos *DataNodes* associados. Por sua vez, os ***DataNodes*** representam os *slaves nodes* presentes na Figura 27. Tipicamente, um *cluster* pode abranger um grande número de *DataNodes*, capazes de executar múltiplas tarefas simultaneamente. Os *DataNodes* são responsáveis não só pela resposta aos pedidos de leitura e escrita provenientes de aplicações externas, mas também pelas tarefas de replicação indicadas pelo *NameNode* (Apache, 2018; S. Fan et al. 2015; Holmes, 2012; Krishnan, 2013).

Para garantir as características mencionadas anteriormente, cada bloco de dados é escrito por defeito três vezes em cada nó do *cluster*, de forma a que uma falha de um nó não implique a capacidade de processar ou aceder aos dados (Holmes, 2012).

3.1.2. *MapReduce*

O *MapReduce* é um modelo de programação de fluxo de dados, que permite o processamento de grandes volumes de dados com o propósito de solucionar problemas computacionais ao nível da escalabilidade (Krishnan, 2013). Assim, é caracterizado pelo processamento paralelo e distribuído sobre grandes volumes de dados (Holmes, 2012).

Este modelo simplifica o processamento paralelo ao abstrair a complexidade envolvida nos sistemas distribuídos, nomeadamente computação paralela, distribuição de trabalho e manipulação de *hardware* e *software* não confiável. Com esta abstração, o *MapReduce* permite que os programadores se concentrem nas necessidades do negócio ao invés de se focarem em problemas do sistema distribuído (Shvachko et al. 2010).

O seu processamento é dividido em duas fases principais: a fase de mapear (*map)*, que consiste no *input* de um par chave-valor no qual se processa um conjunto de pares chave-valor intermédios, e a fase de reduzir (*reduce)*, em que é executada a agregação de todos os valores intermédios com a mesma chave intermédia (Holmes, 2012).

3.2. *Data Warehouse Hive*

O Hive é um *Data Warehouse open-source*, concebido sobre o Hadoop, cujo propósito é simplificar as consultas sobre grandes volumes de dados armazenados num ambiente distribuído. Concebido em 2007 por intermédio do Facebook, este DW tem como propósito de simplificar o acesso ao *MapReduce* (e mais recentemente ao Tez e Spark) por intermédio da linguagem SQL para a manipulação de grandes volumes de dados (Holmes, 2012; Lam, 2010).

Este *Data Warehouse* inclui a sua própria linguagem, HiveQL, cujo objetivo é organizar e executar consultas aos dados armazenados no Hive, imitando a sintaxe da linguagem SQL para a criação, carregamento e consultas de tabelas (Apache, 2018; Holmes, 2012; Krishnan, 2013). Desta forma, a arquitetura do Hive, ilustrada através da Figura 28, apresenta um conjunto de componentes, nomeadamente (Holmes, 2012; Lam, 2010):

1. ***Metastore:*** armazena os metadados sobre os esquemas das tabelas, colunas e partições, além de outros metadados do sistema;
2. ***Driver:*** este componente recebe *queries* provenientes dos componentes *CLI (Command-Line Interface), web UI e Thrift Server.* Inclui a compilação dos *inputs,* otimiza a sua computação e executa os pedidos através dos seguintes componentes:
    i. ***Compiler*:** compila as *queries* em HiveQL provenientes do *driver*, procedendo à análise semântica e verificação do tipo de dados com a ajuda do esquema presente no *metastore*;
    ii. ***Optimizer*:** otimiza o plano concebido pelo *compiler;*
    iii. ***Execution Engine*:** uma vez que as tarefas de compilação e execução estão completas, este componente é responsável pela execução dessas tarefas, tendo em conta as suas dependências;
3. **Thrift Server, JDBC/ODBC:** fornece uma *interface* por meio do servidor *JDBC/ODBC* e API's para a integração entre o Hive e as aplicações externas;
4. **CLI e web UI:** duas interfaces do cliente que permitem a comunicação com o ecossistema Hadoop por meio do *driver.*

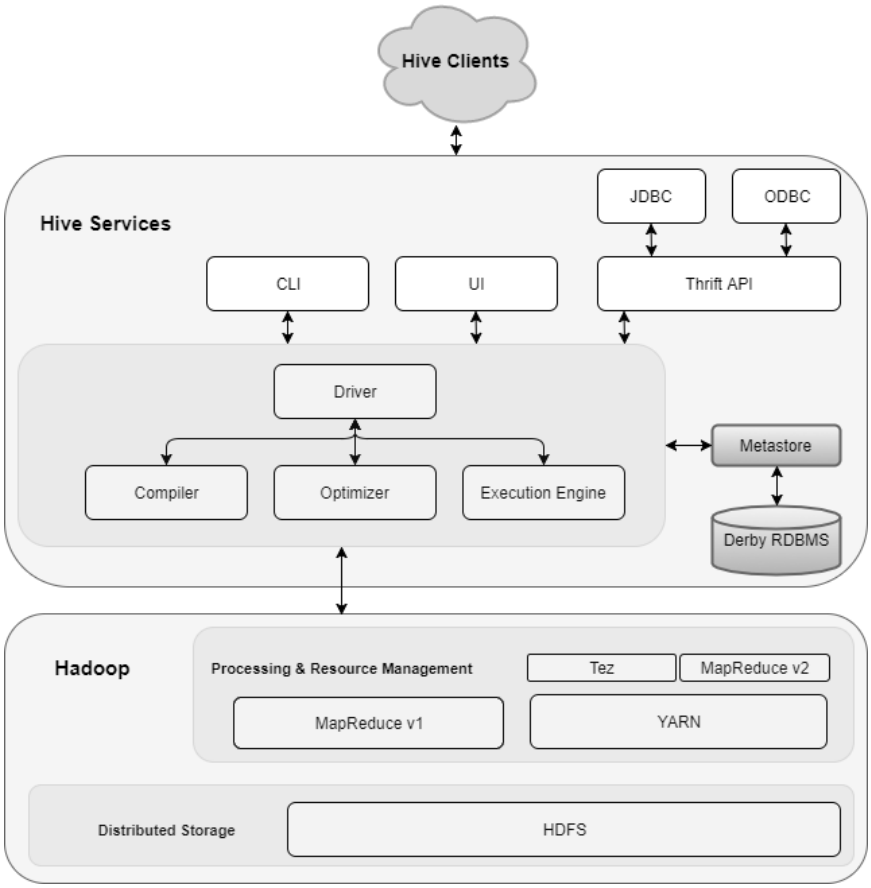

Figura 28. Arquitetura do Hive. Adaptado de (Holmes, 2012; Krishnan, 2013).

Tendo em consideração a Figura 28, esta representa o típico fluxo em que as aplicações externas (*Hive Clients)*, por intermédio das *interfaces* CLI, *web UI* ou utilizando o *Thrift,* submetem o comando HiveQL como *input* para o *driver.* Por sua vez, o *Driver* envia o comando para o *compiler*, que através do metastore concebe um plano de execução. Este plano é otimizado através do *Optimizer* e executado por meio do componente *Execution Engine* (Holmes, 2012; Krishnan, 2013).

Hive Metastore

As bases de dados Hive dizem respeito a um conjunto de tabelas que são constituídas pelos dados que estão armazenados no DW e pela associação dos metadados armazenados no *metastore* (Krishnan, 2013). Tipicamente, a informação inerente aos metadados contém informação sobre o número de colunas e o seu tipo de dados em relação uma determinada tabela (Holmes, 2012; Lam, 2010). De acordo com Krishnan (2013), os dados armazenados no Hive tipicamente residem no HDF. Dessa forma, alguns metadados podem incluir informação relativa à localização física dos dados de uma determinada tabela, assim como informação sobre as suas partições ou *buckets*.

Tabelas, Partições e *Buckets*

O Hive é a ferramenta que permite a implementação de DW em contextos *Big Data*, organizando os dados em tabelas, partições e *buckets* (ficheiros dentro de uma diretoria ou partição de uma tabela) (Capriolo, Wampler, & Rutherglen, 2012; E. Costa et al. 2018; Thusoo et al. 2010).

Por intermédio das diretorias do HDFS, o Hive é capaz de realizar uma partição dos dados com o objetivo de melhorar a performance das *queries* (Holmes, 2012). No contexto das partições, os dados são distribuídos horizontalmente e organizados de uma forma hierárquica. Um particionamento adequado pode trazer benefícios para um DW ao nível do acesso, carregamento e processamento de dados (E. Costa et al. 2018).

Teoricamente, a organização de dados num conjunto mais pequeno conduz a um aumento no desempenho na execução de *queries* (Santos & Costa, 2016). No entanto, de acordo com E. Costa et al (2018), em determinados cenários o particionamento pode não ser uma abordagem adequada para determinadas *queries*, afirmando que a escolha do atributo ou dos atributos é um componente importante para este processo devendo estes ter uma **baixa** cardinalidade, evitando a criação de um número elevado de diretorias no HDFS.

Por sua vez, os *buckets* podem ser associados a tabelas ou partições, que tipicamente se apresentam armazenados dentro das partições ou em diretorias da tabela, dependendo se a tabela se encontra ou não particionada (Apache, 2018; Holmes, 2012). Na visão de Capriolo (2012), os *buckets* permitem a segmentação de grandes conjuntos de dados em conjuntos mais simples de manipular, afirmando que *bucketing* é a técnica ideal para executar *joins* entre tabelas de forma mais eficiente. Existem algumas considerações a ter em conta para a adoção de ambas as técnicas, no entanto, Santos e Costa (2016), reconhecem que ao invés do que sucede com as partições, *buckets* são adequados para atributos com uma **elevada** cardinalidade.

3.3. Atlas como Repositório de Metadados

Com o surgimento da era de *Big Data*, juntamente com aplicações capazes de produzir grandes volumes de dados, torna-se complexo compreender o grau de correlação entre os diversos tipos de dados. Assim emergem desafios de escalabilidade e flexibilidade associados às bases de dados relacionais, que tipicamente se encontram alicerçadas num modelo relacional rígido, apresentando uma difícil adaptação a diferentes cenários em contextos reais (Huang & Dong, 2013; Huo, Wang, Hu, & Yang, 2011).

Tradicionalmente, as bases de dados relacionais foram projetadas para armazenar dados num formato tabular, contudo, à medida que a dimensão da base de dados evolui, a sua estrutura torna-se mais complexa e menos uniforme e o seu modelo relacional tende a ficar sobrecarregado com um grande número de relações entre tabelas. Por sua vez, o aumento do número de relações entre tabelas traduz-se, de igual forma, no aumento do número de operações *join*, que impedem o desempenho e dificultam a evolução da base de dados em resposta às necessidades de negócios em constante mudança (Baton & Bruggen, 2017; Robinson et al. 2013; Vicknair et al. 2010).

Porém, alguns tipos de bases de dados NoSQL apresentam desafios semelhantes, nomeadamente as bases de dados chave-valor, orientadas a documentos e orientadas a colunas, que armazenam conjuntos de dados não relacionados em documentos, colunas e valores (Cattell, 2011). Assim, de forma a relacionar os dados, é armazenado um identificador para que seja possível facultar a agregação entre os mesmos. No entanto, requer que se execute um conjunto de operações *join* ao nível da camada aplicacional que rapidamente influencia o seu desempenho (Robinson et al. 2013).

Logo, face aos desafios apresentados anteriormente, as bases de dados de grafos encontram-se otimizadas para o armazenamento de relações entre os dados, facultando uma representação e manipulação mais simplificada entre conjuntos de entidades. As bases de dados de grafos são caracterizadas pela aplicação de um modelo de grafos com o propósito de armazenar, gerir e atualizar os dados (operações CRUD) e relações (Huang & Dong, 2013).

Assim, com base no que foi referido anteriormente e com a adição de desafios no domínio de Governança de Dados (por exemplo: Qual a proveniência dos dados?, Como podem ser utilizados?, Qual o seu comportamento face às políticas e regras da organização?), surge o Atlas como uma ferramenta com um conjunto escalável e extensível de serviços de gestão de metadados e de Governança de Dados, possibilitando de forma eficiente a descoberta de informação sobre os seus dados, o seu significado, a sua localização, as suas características e a sua utilização (Hortonworks, 2017).

Como tal, o Atlas apresenta características ao nível dos metadados, classificação de dados, *lineage* e segurança de dados, que são descritas de seguida (Atlas, 2018; Hortonworks, 2017):

- Tipos predefinidos para os metadados, permitindo ainda a possibilidade de definir e gerir novos tipos de dados;
- Uma interface para visualizar o *lineage* dos dados à medida que estes se movem através dos vários processos, mantendo um histórico das fontes de dados e da forma como os dados foram gerados;

- Capacidade de criar dinamicamente as classificações *(TAGS)* sobre os dados; por exemplo, classificação dos processos de negócio, classificação da qualidade dos dados, classificação da utilização dos dados e definição de políticas de privacidade de dados através da integração com o Ranger;
- Propagação de classificações via *lineage*, garantindo que, de forma automática, estas seguem os dados à medida que estes passam por vários processos;
- Uma interface intuitiva para pesquisar entidades por tipo, classificação e valor do atributo, utilizando SQL como linguagem de consulta;
- Segurança para o acesso a metadados, permitindo controlar o acesso a instâncias e a operações como adicionar, atualizar ou remover;
- Permuta de metadados com outras ferramentas.

A arquitetura de alto nível referente ao Atlas encontra-se ilustrada na Figura 29, seguindo-se a descrição dos seus principais componentes (Atlas, 2018).

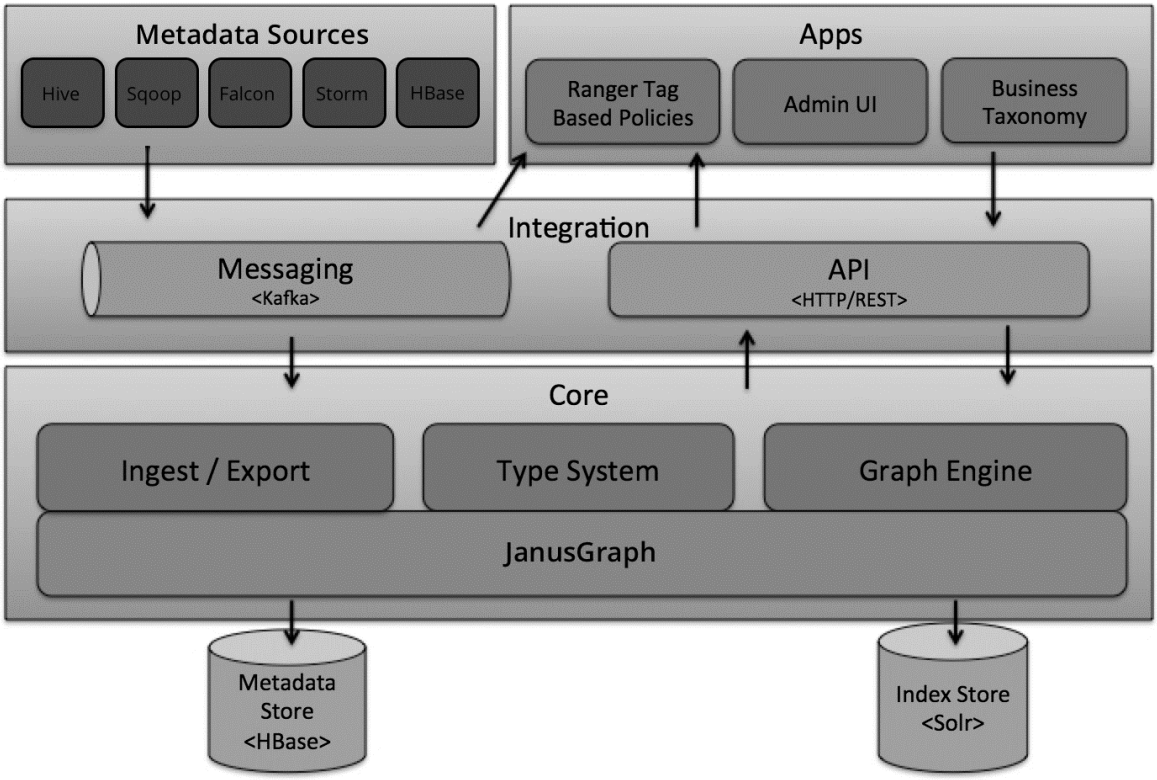

Figura 29. Arquitetura do Apache Atlas. Retirado de Atlas (2018).

- ***Core***
    - ***Type System***: O Atlas permite que seja definido um modelo para os objetos dos metadados que se deseja gerir. O modelo é composto por "tipos" e pelas instâncias dos "tipos", que são intituladas de "entidades", representando os objetos dos metadados que são geridos. Portanto, o *Type System* permite definir e gerir os tipos e as entidades, assim, todos os objetos dos metadados geridos pelo Atlas, tais como as tabelas Hive, são modelados desta forma. A modelação no Atlas permite que sejam definidos metadados técnicos e metadados de negócio, para além de permitir definir relacionamentos entre os dois;
    - ***Graph Engine***: Internamente, o Atlas utiliza um modelo de grafos para gerir os objetos dos metadados. Esta abordagem fornece grande flexibilidade e permite a manipulação eficiente dos relacionamentos entre os objetos dos metadados. O componente *Graph Engine* é responsável pela tradução entre os tipos e as entidades do sistema e o modelo de grafos subjacente. Este mecanismo permite também criar índices para os objetos dos metadados, para que estes possam ser procurados com relativa eficiência. Assim, o Atlas incorpora no seu sistema a base de dados de grafos Janus[15] para armazenar os metadados, utilizado dois tipos de armazenamento: o *Metadata Store*, configurado como padrão para o HBase, e o *Index Store*, configurado para Solr[16];
    - ***Ingest/Export***: O componente *Ingest* permite que metadados sejam adicionados ao Atlas. Por outro lado, o componente *Export* expõe as alterações dos metadados detetadas pelo Atlas, sendo criadas como eventos. Desta forma, as alterações dos metadados podem ser verificadas em tempo real através desses eventos;
- ***Integration***: Os metadados podem ser geridos recorrendo-se a dois métodos:
    - ***API*** (*Application Programming Interface*): Todas as funcionalidades do Atlas são apresentadas aos utilizadores finais através de uma REST (*REpresentational State Transfer*) API que permite que os tipos e as entidades sejam criados, atualizados ou excluídos;

[15] https://janusgraph.org/
[16] http://lucene.apache.org/solr/

- ***Messaging***: Os utilizadores podem optar por integrar-se com o Atlas utilizando uma interface de mensagens baseada no Kafka[17]. Esta abordagem pode ser útil para a correspondência de objetos dos metadados para o Atlas e para consumir os eventos relacionados às alterações dos metadados;

- ***Metadata Sources***: O Atlas suporta a integração com múltiplas fontes, sendo que, atualmente, suporta a Gestão de Metadados provenientes do HBase[18], Hive, Sqoop[19], Storm[20] e Kafka. Existem modelos de metadados que o Atlas define nativamente para representar objetos e, por outro lado, existem componentes no Atlas que possibilitam a ingestão de objetos dos metadados dessas fontes, sejam eles provenientes em tempo real ou em *batch;*
- ***Applications***: Os metadados que são geridos no Atlas são utilizados por várias aplicações de governança:
    - ***Atlas Admin UI***: É uma aplicação baseada na *web* que permite aos administradores e cientistas de dados descobrir e anotar os metadados. É importante a existência de uma interface de pesquisa e uma linguagem semelhante à SQL para consultar os tipos e os objetos dos metadados geridos pelo Atlas. Esta componente baseia-se na utilização da REST API do Atlas;
    - ***Tag Based Policies***: O Apache Ranger[21] é uma solução para a gestão da segurança no ecossistema Hadoop, podendo ser integrada com diversas ferramentas. Os administradores da segurança podem definir políticas orientadas por metadados ao integrarem o Atlas com o Ranger, proporcionando uma governança mais eficaz;
    - ***Business Taxonomy***: Permite que os utilizadores definam um conjunto hierárquico de termos de negócio que representam o domínio do negócio e os associe às entidades dos metadados geridos pelo Atlas.

Uma visão sobre a JanusGraph

A base de dados **JanusGraph**, utilizada pelo Atlas, é uma base dados de grafos, otimizada para o armazenamento de relações entre os dados, apresentando-se assim ao nível dos requisitos propostos anteriormente neste documento. Além disso, os benefícios principais associados a esta base de dados

[17] https://kafka.apache.org/
[18] https://hbase.apache.org/
[19] http://sqoop.apache.org/
[20] http://storm.apache.org/
[21] http://ranger.apache.org/

estão relacionados com as relações entre os dados e com a escalabilidade no processamento de travessias de grafos *(graph traversal)* em tempo real e no processamento de *queries* analíticas (He, 2018).

A arquitetura da JanusGraph, que se encontra ilustrada na Figura 30, demonstra como as aplicações podem interagir com esta base de dados, sendo possível realizar esta interação de duas formas distintas (JanusGraph,2019):

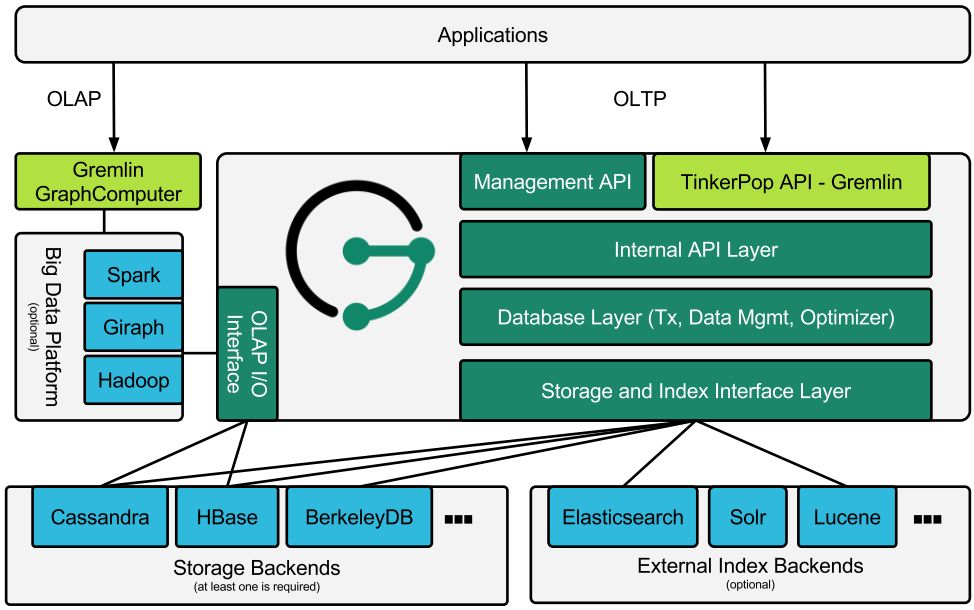

Figura 30. Arquitetura JanusGraph. Retirado de JanusGraph (2019).

Este capítulo de exploração de tecnologias permite o conhecimento daquelas que poderão vir a integrar na solução desenvolvida. Cada uma das tecnologias apresentadas irá representar um papel importante na concretização da ferramenta de *Data Profiling*, sendo que estas tecnologias integram o ecossistema do Hadoop.

4. AMBIENTE DE TESTES E DADOS UTILIZADOS

Em conformidade com os objetivos identificados na secção 1.2, para a concretização de uma ferramenta de *Data Profiling* em ambientes *Big Data,* é necessária a exploração dos principais conceitos envolvidos e das tecnologias de possível aplicação e a proposta da arquitetura que permita a concretização deste tipo de ferramentas. Para que a solução seja concebida, as tecnologias e os algoritmos considerados devem ser aqueles que apresentam um melhor desempenho.

Como tal, um dos principais desafios na conceção de uma ferramenta de *Data Profiling* reside na integração de dados, ou seja, descobrir como as novas fontes de dados têm a capacidade de ser integradas com os dados que já subsistem no BDW.

Estes desafios podem ser colmatados através de técnicas e algoritmos de integração de dados. No entanto, é necessário perceber o impacto que as diferentes abordagens de integração terão ao nível do **tempo de processamento** e do **valor da similaridade** associado para, então, inferir algum tipo de padrão. Assim, a fase experimental da presente dissertação vai integrar dois principais momentos:

- Compreender as variações das métricas apresentadas anteriormente associadas às diferentes medidas de similaridade através de vários cenários de teste;
- Propor, implementar, desenvolver e validar a arquitetura de Governança de Dados para integração de dados.

Assim, numa primeira fase, é apresentado o ambiente de testes em termos de infraestrutura física, o conjunto de dados, as medidas de similaridade utilizadas e, por fim, o protocolo de testes que será seguido. Os resultados obtidos serão utilizados como base para identificar em que cenários de *Data Profiling* a sua utilização apresentará mais vantagens, com a perspetiva de otimizar a utilização e o desempenho da ferramenta.

4.1. Infraestrutura de Testes

A infraestrutura de testes integra num *cluster* Hadoop de 5 nós (1HDFSNameNode e 4 HDFS Data Nodes), cada um com as seguintes características de *hardware:*

- Intel i5, quad core, com velocidade entre 3.1-3.3 GHz;
- 32 GB de Random Access Memory (RAM), com 24GB disponível para processamento de consultas;
- 1 *gigabit Ethernet card;*

- Samsumg 850 EVO 500GB *Solid State Drive* (SSD), com capacidade de ler 540MB/s e de escrever 520MB/s.

O sistema operativo utilizado em todos os nós presentes no *cluster* é o CentOS7 com um sistema de ficheiros XFS. A distribuição do *cluster* Hadoop foi atualizada relativamente à versão utilizada por E. Costa (2018) para a versão Hortonworks Data Platform (HDP) 3.1.0.

Por fim, foi adicionado um novo nó com as mesmas características dos nós apresentados anteriormente, exceto a componente da memória RAM e o número de *cores* presentes no processador, apresentando assim 16GB de memória RAM e um processador Intel i5, dual core. Neste novo nó, foi instalado a versão 1.1.0 do Atlas, via Ambari, ficando alocado os componentes Atlas Metadata Client e Atlas Metadata Server no mesmo.

4.2. Conjuntos de Dados

Para a escolha dos conjuntos de dados a utilizar, foram consideradas duas questões importantes. É importante selecionar um conjunto de dados que permita testar diferentes volumes de dados, assim como um conjunto de dados cujo modelo de dados permita diferentes perspetivas de modelação. Anteriormente à escolha do conjunto de dados, foi realizada uma revisão de literatura nas vertentes de comparação de diferentes medidas de similaridade, de modo a que seja possível utilizar o mesmo padrão de testes ou o mesmo conjunto de dados. No entanto, a literatura apenas tem comparado e analisado as diferentes medidas de similaridade entre diferentes palavras (*Strings)* (O'Shea, Bandar, Crockett, & McLean, 2010), não sendo aplicável para a presente dissertação, que pretende a análise entre diferentes colunas/vetores de dados.

Como tal, para a realização dos testes efetuados na presente dissertação é utilizado o TPC *Benchmark* DS (TPC-DS)[22], desenvolvido por Nambiar e Poess (2006), que é um *benchmark* que modela várias abordagens tipicamente aplicáveis a um sistema de suporte à decisão, retratando o contexto de um fornecedor retalhista com diversos canais de venda, tais como, lojas físicas, catálogos e vendas online. Este *benchmark* consiste num modelo de dados numa perspetiva *snowflake*, que representa múltiplos esquemas em estrela com as suas dimensões associadas.

O modelo base do *benchmark* apresenta **7** tabelas de facto *("Store_Sales", "Store_Returns", "Web_Sales", "Web_Returns", "Catalog_Sales", "Catalog_Returns", "Inventory"*) e **17** tabelas de

[22] http://www.tpc.org/tpcds/

dimensão *("Date_Dim", "Store", "Item", "Customer_Demographics", "Customer", "Income_Band", "Household_Demographics", "Customer_Address", "Time_Dim", "Promotion", "Reason", "Catalog_Page", "Call_Center", "Ship_Mode", "Warehouse", "Web_Page")*. No entanto, o modelo de dados que serve como base para a fase de testes apresenta apenas a tabela de facto ***"Store_Sales"*** e as dimensões associadas, tal como se encontra representado na Figura 31**.**

Recorrendo a este conjunto de dados pré-definido, é possível configurar o tamanho da base de dados desejada, através da sua extração de acordo com o Fator de Escala (FE) pretendido. Assim, é possível extrair, por exemplo, uma base de dados com fator de escala 10 que gera uma tabela de factos com cerca de 28804040 linhas (aproximadamente 10GB) e as dimensões com os tamanhos correspondentes (calculadas de acordo com as fórmulas da Figura 31).

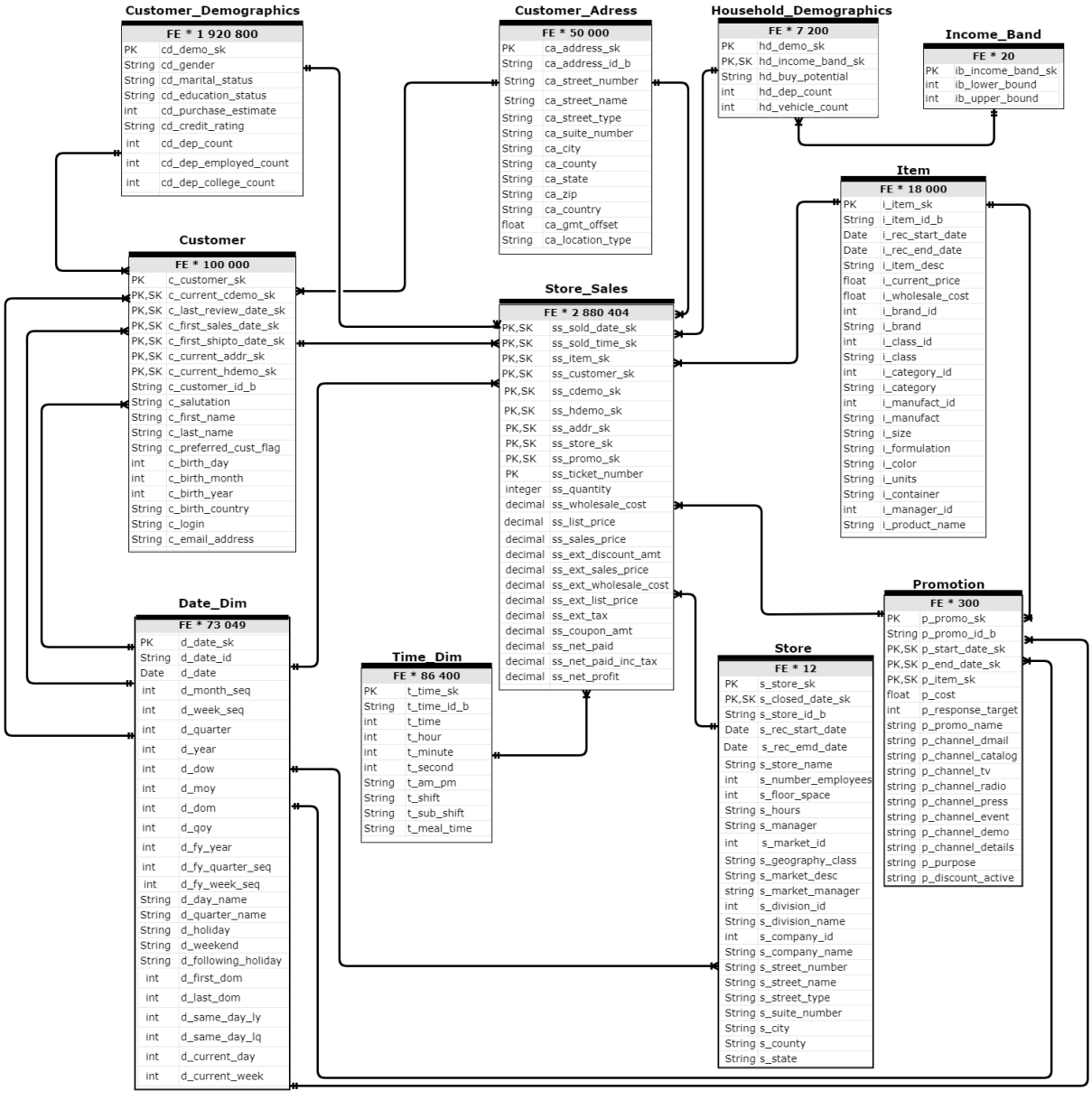

Figura 31. Modelo de Dados TPC-DS Benchmark. (Nambiar e Poess, 2006).

No domínio da integração de dados, o conjunto de dados proveniente do TPC-DS apresenta diferentes cenários e uma vantagem de testar os algoritmos de similaridade com diferentes FE. Porém, o seu conjunto de dados é caracteristicamente sintético sendo gerados segundo um padrão e sem qualquer anomalia. Como tal, de forma a contornar essa questão, foi fornecido um conjunto de dados pelo orientador sobre **Genoma Humano**, com extrações no âmbito da saúde.

Estes conjuntos de dados encontram-se devidamente documentados/analisados por Román (2018) na sua tese de doutoramento. O principal objetivo deste conjunto de dados é avaliar o valor da similaridade dos algoritmos perante um conjunto de dados inserido num contexto real.

Este conjunto de dados é composto por 4 subconjuntos de dados que se encontram em https://github.com/josemagalhaes1996/smart-distributed-crawler-big-data-integration, nomeadamente "*GWAS"* (1060 registos e 39 colunas), "*Ensembl*" (32856 registos e 36 colunas), "*DisGeNET"* (2947 registos e 16 colunas) e, por fim, "*AlzForum"* (398 registos e 14 colunas). Os subconjuntos de dados não apresentam um elevado número de registos, mas, se for realizado o produto cartesiano entre o conjunto de dados "*Ensembl*" e "*GWAS"* observa-se a execução dos algoritmos para 1404 pares de atributos.

O conjunto de dados "*DisGeNET"* diz respeito a uma plataforma integrada que aborda uma variedade de questões relacionadas com base genética de doenças humanas. Este conjunto integra uma base de dados preenchida por especialistas do contexto, abrange informação sobre doenças complexas e inclui informação sobre modelos de doenças animais. Além disso, possui uma pontuação baseada em evidências para priorizar associações entre genes e doenças. Por sua vez, o conjunto de dados *"Ensembl"* é um sistema de interpretação genética, ou seja, é um navegador de genes de vertebrados que apoia pesquisas entre diferentes genes, assim como a sua evolução, com sequência de variações e regulação transicional. Segue-se o conjunto de dados "*AlzForum"* que apresenta frequências alélicas de um foco populacional. Por fim, o conjunto de dados *"GWAS"* diz respeito a uma base de dados *on-line*, que integra os resultados de estudos genéticos, criado pelo Instituto Nacional de Pesquisa dos Genes Humanos em 2008. Dado o resumo de cada conjunto de dados, é possível observar a presença de um elemento comum entre os diferentes conjuntos de dados: **genes/cromossomas.**

Resumindo, o conjunto de dados TPC-DS tem como objetivo ser utilizado para avaliar o tempo de processamento das medidas de similaridade perante diferentes FE e o conjunto de dados do Genoma tem como propósito avaliar o valor da similaridade.

4.3. Medidas de Similaridade

Em ambientes *Big Data*, o tempo de processamento de operações do tipo *join* entre diferentes entidades é tipicamente elevado. Não obstante, o resultado desse tipo de operações pode, ou não, ser vantajoso para o processo de tomada de decisão, devido à possibilidade de os dados presentes nesse tipo de operações não apresentarem relações semânticas entre si. Uma entidade está tipicamente associada a um objeto, coluna e linha e o principal desafio diz respeito às diferentes formas de representação da mesma (Tirumali, 2016). Desse modo, os mecanismos de similaridade são fundamentais para avaliar as relações semânticas entre entidades, com a perspetiva de demonstrar se estamos perante a mesma entidade.

Definir relacionamentos entre diferentes conjuntos de dados através das medidas de similaridade é uma componente essencial nas operações de limpeza, análise e integração de dados. Na secção 2.3.2 encontram-se presentes as medidas que foram propostas ao longo desta dissertação, destacando um elevado número de trabalhos, nomeadamente, os de Arasu (2006), Bayardo (2007), Gravano et al.(2001), nos quais as medidas são executadas *in-memory*. Contudo, em contextos de *Big Data* é importante que estas medidas sejam executadas em ambientes distribuídos, permitindo assim a distribuição da carga pelos vários nós do *cluster*.

Assim, para a realização de testes, são utilizadas quatro medidas de similidade apresentadas anteriormente na secção 2.3.1, nomeadamente *Cosine*, *Jaccard*, *Levenshtein* e *Jaro-Winkler*. Todas as medidas mencionadas anteriormente foram importadas do *package java-string-similarity*[23] presente no GitHub.

Com o propósito de auxiliar os resultados obtidos na fase de testes, é importante compreender como se processam as medidas importadas do *package*. Assim, de seguida, será explicado pormenorizadamente cada algoritmo.

1. ***Jaccard:*** esta medida permite a análise da similaridade entre duas *Strings*, sendo que a sua execução requer como parâmetros dois objectos do tipo *String*. De forma a avaliar a similaridade entre as duas *Strings,* cada objecto *String* despoleta um processo de *profiling,* presente do trabalho de Ukkonen (1992)*,* no qual são eliminados todos os espaços vazios e todos os caracteres especiais. Neste processo é também realizada a distribuição dos caracteres associados a cada objeto *String* sendo, posteriormente, armazenados num *HashMap*, tendo

[23] https://github.com/tdebatty/java-string-similarity

como *key* o caracter e como *value* o número de vezes que este se encontra no objecto *String*. Depois do processo de *profiling* ser executado para cada objeto *String*, é realizada a intersecção entre os dois *HashMaps* sobre a união dos mesmos (fórmula presente na secção 2.3.1). Esta medida, apesar de realizar o processo de *profiling,* não considera a distribuição dos caracteres. Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 32 retrata um exemplo do cálculo da similaridade de *Jaccard* entre duas *Strings.*

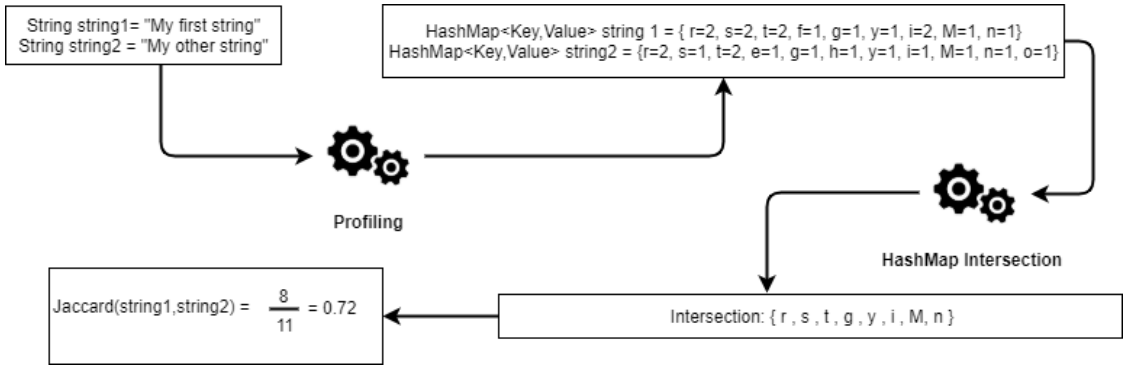

Figura 32. Exemplificação do cálculo da similaridade de Jaccard.

2. ***Cosine:*** esta medida permite a análise da similaridade entre duas *Strings*, sendo que a sua execução requer como parâmetros dois objectos do tipo *String*. De forma a avaliar a similaridade entre as duas *Strings,* cada objecto *String* despoleta um processo de *profiling,* presente no trabalho de Ukkonen (1992)*,* no qual são eliminados todos os espaços vazios e todos os caracteres especiais. No decorrer do processo, para cálculo futuro é também realizada a distribuição dos caracteres associados a cada objeto *String*. Tipicamente, a distribuição destes caracteres é armazenada num *HashMap*, tendo como *key* o caracter e como *value* o número de vezes que este se encontra no objecto *String*. Posteriormente ao processo de *profiling* ser executado para cada objeto *String*, é calculado o produto vetorial da intersecção entre os caracteres dos objectos *String*, tendo por base a distribuição dos valores obtidos no processo de *profiling*. Uma vez que o produto vetorial é finalizado, o cálculo desta medida é efetuado através da fórmula presente na secção 2.3.1). Em comparação com a medida de *Jaccard*, esta medida considera a distribuição dos valores, um facto importante na avaliação dos resultados. Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 33 retrata um exemplo do cálculo da similaridade de *Cosine* entre duas *Strings.*

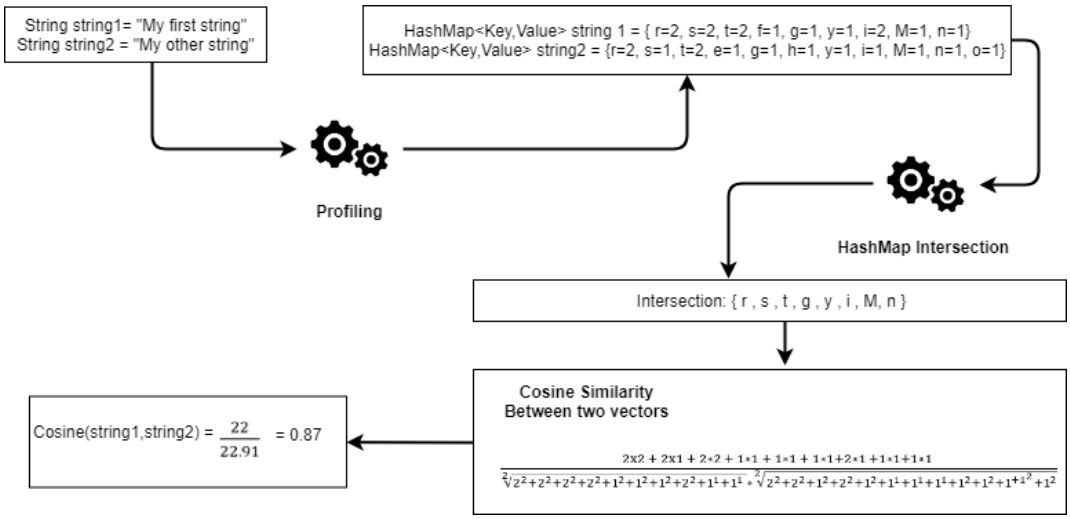

Figura 33. Exemplificação do cálculo da similaridade de Cosine.

3. ***Levenshtein:*** esta medida permite a análise da similaridade entre duas *Strings*, encontrando-se otimizada para comparação entre palavras (não se encontra otimizada para frases). A similaridade é calculada com base na **distância de *Levenshtein***, cujo objetivo é a obtenção do número de caracteres diferentes entre dois objectos *String*. Posteriormente, a distância de *Levenshtein* é dividida pelo tamanho do maior objeto *String*, sendo o cálculo desta medida efetuado através da fórmula presente na secção 2.3.1). Assim, pode afirmar-se que o tamanho dos objectos *Strings* é um fator a considerar e que, de certa forma, influencia o resultado da medida de similaridade, uma vez que se o tamanho é elevado e a distância de *Levenshtein* é reduzida, o resultado da similaridade será reduzido quando comparado com outras medidas de similaridade. Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 34 retrata um exemplo do cálculo da similaridade de *Levenshtein* entre duas *Strings.*

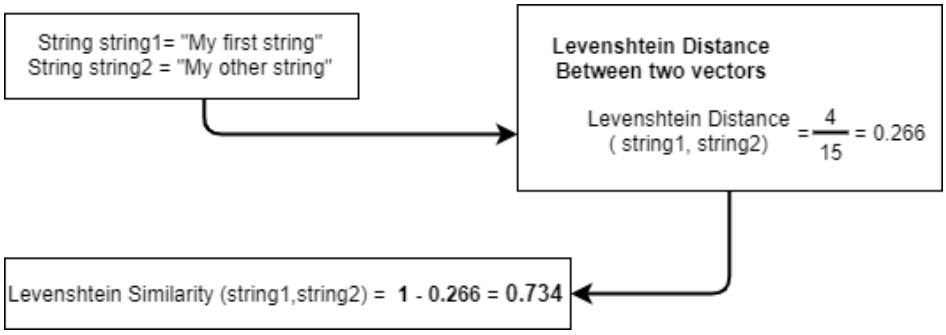

Figura 34. Exemplificação do cálculo da similaridade de Levenshtein.

4. ***Jaro-Winkler:*** esta medida permite a análise da similaridade entre duas *Strings*. À semelhança da medida anterior, esta medida de similaridade encontra-se otimizada para objectos *String* de curta dimensão. Esta medida considera não só os caracteres em comum entre os dois objectos

String, mas também as transposições, ou seja, o número de vezes que os caracteres comuns se encontram em posições diferentes em relação à *String*. Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 35 retrata um exemplo do cálculo da similaridade de *Jaro-Winkler* entre duas *Strings.*

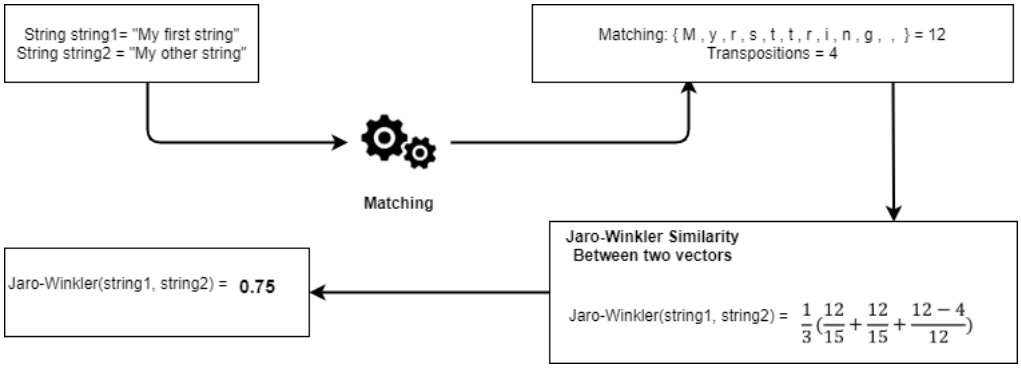

Figura 35. Exemplificação do cálculo da similaridade de Jaro-Winkler.

É importante compreender como todas as medidas são executadas e ter conhecimento de todas as etapas até atingir o resultado da similaridade. É possível observar que algumas medidas requerem um número maior de procedimentos e que acarretam uma carga de trabalho superior ao sistema. Assim, na Tabela 6 encontra-se o custo de cada medida na sua execução (**m** e **n** representam o comprimento das *Strings* a ser comparadas).

Tabela 6. Custo de computação das medidas de similaridade.

Medida de Similaridade	**Custo**
Levenshtein	O(m*n)
Jaro-Winkler	O(m*n)
Jaccard	O(m+n)
Cosine	O(m+n)

Segundo a Tabela 6, é possível observar que as medidas *Levenshtein* e *Jaro-Winkler* têm um custo de computação mais elevado quando comparadas com as medidas de *Jaccard* e *Cosine*, uma vez que se trata do produto entre os tamanhos das *Strings*. Este tipo de informação é relevante para a avaliação do tempo de processamento na fase de testes.

Como é referido ao longo desta secção, as medidas de similaridade sintetizadas anteriormente são importadas de um *package*. No entanto, e como forma de tirar proveito das funções distribuídas Spark, são apresentadas duas medidas. Estas medidas são baseadas em Zhu et al. (2019) (apenas na

ótica de executar os cálculos da medida tendo em conta apenas os valores distintos), e têm como objetivo reduzir um dos principais problemas desta temática, mais precisamente o tempo de processamento de grande quantidade de dados.

A primeira medida, ao invés das medidas anteriores, recebe, como parâmetros, dois objetos do tipo *Dataset<Row>*. Esta medida é baseada na distribuição de valores e, como tal, é calculada a distribuição de valores **apenas** para a coluna a ser comparada através da função Spark *countByValue*. Posteriormente, é executada a função *intersection,* cujo objetivo é avaliar os valores comuns entre as duas colunas dos diferentes conjuntos de dados. Esta função inclui a função *distinct*, evitando assim valores duplicados que resulta no aumento de performance da medida. De seguida, é analisado o número de vezes que os valores intersetados se repetem ao longo da coluna, através da distribuição de valores calculada anteriormente e, por fim, é calculado o grau de similaridade, através da soma dos valores (da função *reduce)* dividida pelo número de linhas total da coluna a ser comparada.

Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 36 retrata um exemplo entre duas colunas.

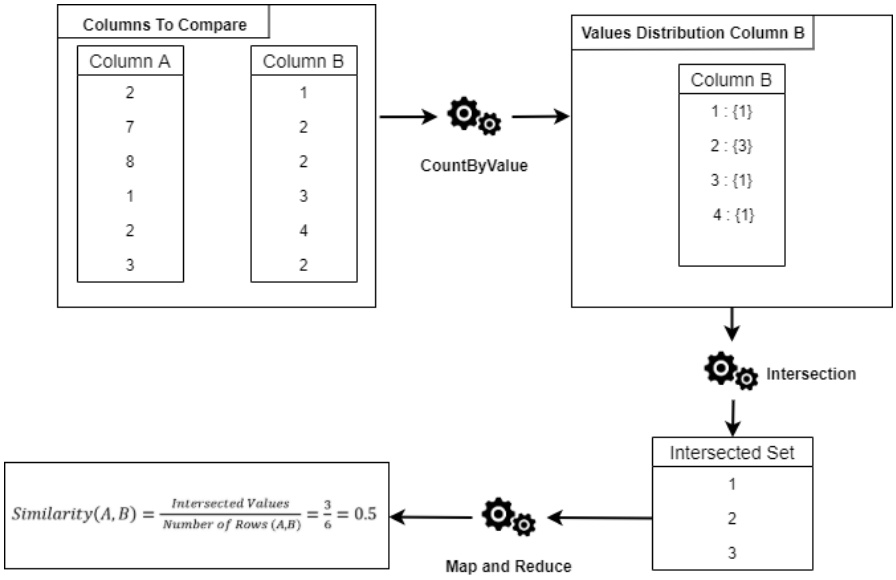

Figura 36. Exemplificação do cálculo da medida de similaridade baseada na distribuição de valores.

Para terminar, é utilizada uma medida que calcula a similaridade com base na intersecção e união dos valores distintos, também conhecida como medida de Jaccard. Esta medida recebe como parâmetros dois objetos do tipo *Dataset<Row>*. No prosseguimento do cálculo desta medida é executada a função *intersection,* cujo objetivo é avaliar os valores comuns entre as colunas dos diferentes conjuntos de dados. Esta função aplica o método *distinct* com o propósito de evitar os valores duplicados, resultando no aumento do desempenho da medida. Por fim, é realizado o cálculo da similaridade através

do número de valores distintos intersectados (intersecção) sobre o número de valores distintos entre ambos os conjuntos de dados (união). É importante mencionar que esta medida retrata a medida de Jaccard, no entanto, neste ambiente, a medida de Jaccard está otimizada para comparação de Strings, ou invés da presente medida que está otimizada para grandes volumes de dados, ou seja, é aplicada ao conjunto todo ao invés de realizar o *"profiling"* de uma String. Com o objetivo de se compreender melhor o funcionamento desta medida, a Figura 37 retrata um exemplo entre duas colunas.

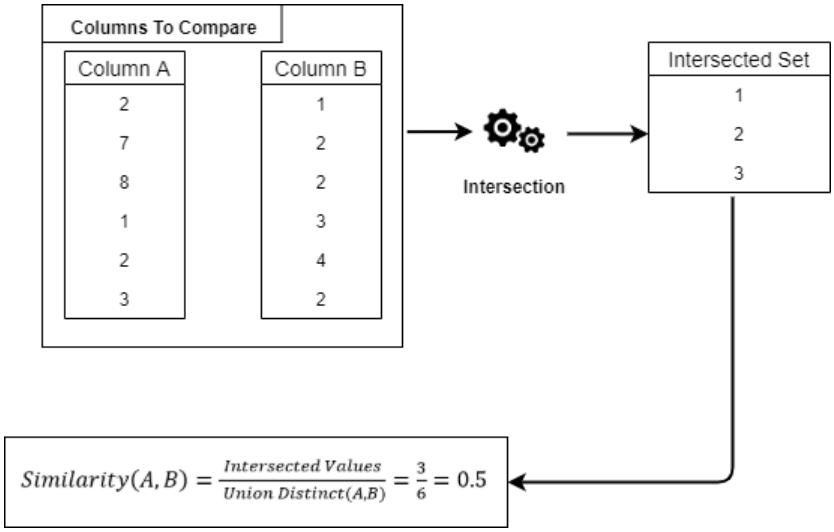

Figura 37. Exemplificação do cálculo da medida Jaccard para grandes quantidades de dados.

4.4. Protocolo de Testes

A integração dos novos conjuntos de dados no BDW está divida em duas dimensões distintas, uma vez que é necessário avaliar os ***headers*** e o **conteúdo** dos dados. Esta decisão advém do facto de, tipicamente, o conteúdo dos dados ser semelhante, tais como chaves primárias (particularmente em formatos *auto increment)*. Porém, a entidade que se pretende integrar poderá ser dissemelhante. Assim, recorre-se também à análise da similaridade dos *headers* como um contributo na integração de dados. Como tal, no presente capítulo pretende investigar-se as medidas de similaridade nas dimensões referidas anteriormente, ao nível do tempo de processamento e do valor da similaridade associado.

Para efetuar esta análise, aplicar-se-ão os seguintes cenários de teste, explicados de seguida:

1. **Cenário A**: Avaliação das medidas de similaridade nos *headers,* através da análise dos indicadores mencionados anteriormente. Este tópico está divido em dois sub cenários:
    - Cenário **AR:** Integração de tabelas cujos atributos se relacionem entre si;
    - Cenário **ASR:** Integração de tabelas cujos atributos não se relacionem entre si.

2. **Cenário B:** Avaliação das medidas de similaridade ao nível do conteúdo dos atributos, através da análise dos indicadores mencionados anteriormente. Pretende analisar-se como estas medidas se comportam perante diferentes volumes de dados.

A definição das siglas atribuídas à identificação dos cenários segue a seguinte nomenclatura: AR (**A**tributos com **R**elacionamento), ASR (**A**tributos **S**em **R**elacionamento).

Com estes testes pretende perceber-se qual a estratégia de integração que apresenta mais vantagens num BDW, comparando os resultados obtidos nos vários cenários e interpretando o impacto das diferentes medidas de similaridade com o aumento no volume de dados, sendo apenas aplicados os quatro FE (1,3,5 e 10) para o cenário B. Em relação ao cenário A, não se considera relevante testar os quatro FE, uma vez que o principal objetivo de análise são os *headers,* que não são modificados com o aumento do volume de dados.

Para além disso, na dimensão de avaliação dos ***headers*** serão aplicadas as medidas de *Jaccard*, *Cosine*, *Levenshtein* e *Jaro-Winkler*, e na dimensão de avaliação do **conteúdo** dos dados será aplicada a medida com base na distribuição de valores e a medida de similaridade com base na intersecção e união dos valores. Na dimensão dos *headers,* não serão aplicadas as medidas que são executadas para avaliação da similaridade do conteúdo de dados, uma vez que estas só iriam apresentar resultados de similaridade positivos **apenas** se os *headers* se apresentarem exatamente iguais, ao contrário das outras medidas que, através do algoritmo proposto por Ukkonen (1992), permitem uma manipulação mais eficiente de objetos *String*.

Assim, a Figura 38 apresenta os vários cenários de teste a executar e as suas variantes, nomeadamente os cenários A e B.

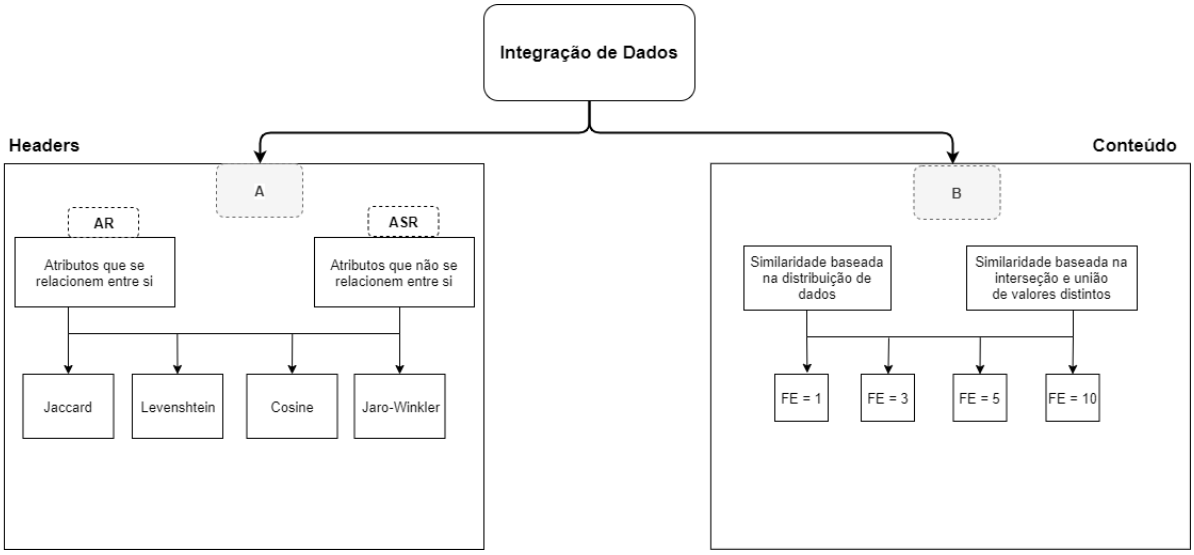

Figura 38. Cenários de Teste.

Resumindo, e tendo em conta a Figura 38, serão testadas as **duas** dimensões diferentes (*headers* e conteúdo). No que diz respeito à avaliação da dimensão de avaliação dos *headers,* serão realizados **dois** cenários teste, o primeiro com tabelas cujos atributos apresentam um relacionamento entre si e, o segundo, com tabelas cujos atributos **não** apresentam relacionamento si. Ambos os cenários irão ser testados para medidas indicadas na Figura 38. Para a avaliação da dimensão do conteúdo dos dados, será utilizada a medida com base na **distribuição de valores** e a medida com base na **intersecção e união de valores distintos** para os quatro FE diferentes (1,3,5 e 10). Relativamente aos conjuntos de dados utilizados em ambos os cenários, no cenário A será utilizado o conjunto de dados do TPC-DS e, no cenário B, será utilizado o conjunto de dados do TPC-DS para a avaliação escalabilidade dos algoritmos, e o conjunto de dados do Genoma Humano para avaliação do valor da similaridade associado.

4.5. Preparação para os Testes

Previamente à elaboração dos testes, é necessário responder a um conjunto de desafios, nomeadamente no tipo (operações de junção ou união) de integração entre as novas fontes de dados e os dados que já subsistem no BDW. No domínio da integração de dados, previamente a se integrar duas tabelas, é necessário compreender o domínio de ambas e se o tipo de integração que se executa se trata de uma **união** *(union)* ou **junção** (*join)* entre tabelas.

Sintetizando estes dois tipos de integração, ambos são utilizados para combinar dados de uma ou mais tabelas, sendo que, a diferença está na forma como são combinados. Tipicamente, as funções de **junção** *(join)* combinam colunas de duas tabelas diferentes; a título de exemplo, se duas tabelas se submeterem a uma operação *join* (tipicamente através do uso de chaves estrangeiras), as colunas da primeira tabela serão apresentadas em conjunto com as novas colunas que são integradas. Por sua vez, a **união** de tabelas tem como objetivo a inserção de novas linhas ao invés de novas colunas. Como tal, pretende-se frisar que a presente dissertação apenas responde aos desafios dos conjuntos de dados que perspetivam uma **junção** com conjuntos de dados que já subsistem no BDW. Esta decisão deve-se a uma decisão do autor do documento que, devido ao tempo proposto para realizar a dissertação, irá abordar primeiramente os cenários que perspetivem uma junção e, em trabalho futuro, serão abordados os cenários que se trate de uma união.

Além da apresentação dos resultados obtidos, é importante detalhar a preparação dos cenários de teste, isto é, todas as tarefas necessárias para preparar e executar os vários testes. É importante mencionar

que a tarefa 1 e 2 são executadas com auxílio do orientador encontrando-se documentadas para uma possível replicação dos testes.

1. Gerar os dados TPC-DS: É necessário gerar a fonte de dados do TPC-DS com os diferentes FE associados (1,3,5 e 10) e armazená-los no HDFS. Uma vez que o TPC-DS contém um número elevado de estrelas, é necessário selecionar a(s) estrela(s) que se pretendem gerar;
2. Criação do *Cluster*. O *cluster* Hortonworks selecionado apresenta a infraestrutura presente na secção 4.1. A escolha tem como base o facto de ser um *cluster open-source* e ser capaz de suportar *commodity-hardware;*
3. Configurações das Aplicações: Posteriormente à instalação do *cluster*, é necessário configurar um conjunto de aplicações através do Ambari, nomeadamente a configuração do Hive, HDFS e do Spark2;
4. Implementação dos algoritmos: É necessário executar o *build* da aplicação desenvolvida em Spark. A aplicação está presente em https://github.com/josemagalhaes1996/smart-distributed-crawler-big-data-integration;
5. Execução das experiências: É necessário executar um conjunto de classes na aplicação concebida, estando estas classes presentes no package "Benchmark" do *link* mencionado anteriormente no ponto 4. As classes que fazem referência ao cenário A são as classes *"SimilarityMesuresHeaders"*. As classes que fazem referência ao cenário B são as classes "DistributionValuesSimilarity" e "IntersectionUnionSimilarity". Os resultados das experiências são discutidos na secção 4.6.

4.6. Resultados Obtidos

Finalizada a análise do conjunto de dados e expostos os vários cenários de teste, nesta secção será apresentado e analisado os resultados obtidos que constituirão uma base para sugerir algumas recomendações sobre a implementação da ferramenta proposta.

Integrar os novos conjuntos de dados e verificar as combinações entre dois ou mais conjuntos de dados, acarreta desafios principalmente nas vertentes de escalabilidade e *performance*. A título de exemplo, nos conjuntos de dados utilizados para o cenário **AR** (avaliação das medidas de similaridade entre a tabela "*Store_Sales*" e "*Promotion")*, a tabela "*Store_Sales"* contém 23 atributos e a tabela "*Promotion"* contém 19 atributos, o que gera 437 pares de atributos (*n*m)*. Numa fase de testes, é necessário ter em conta todos os pares de atributos, no entanto, é importante a conceção de estratégias

que serão discutidas no final do presente capítulo que minimizem o número de comparações necessárias entre tabelas, uma vez que apenas é necessário um par de atributos para se realizar a integração entre dois conjuntos dados diferentes.

4.6.1. Cenário A – Avaliação da similaridade dos *headers*

Este cenário, tem como base o modelo utilizado na secção 4.2. Para o cenário **AR**, no qual é necessário a presença de atributos comuns a ambas as tabelas, serão utlizadas as tabelas "*Promotion"* e "*Store_Sales"* do conjunto de dados TPC-DS, que se interligam através do atributo *"p_promo_sk"* e *"ss_promo_sk"*. Para além desses atributos, ambas as tabelas têm a tabela "*Item"* em comum, sendo esta representada como *"p_item_sk"* na tabela "*Promotion"* e como *"ss_item_sk"* na tabela "*Store_Sales"*. Por sua vez, para o cenário **ASR**, no qual **não** é necessária a presença de atributos comuns a ambas as tabelas, serão utilizadas as tabelas "*Income_band"* e "*Store_Sales"* que, segundo o modelo apresentado na secção 4.2, não apresentam qualquer relação.

Para ambos cenários, pretende averiguar-se como se comportam as medidas de similaridade perante situações em que estejam presentes, ou não, atributos comuns a ambas as tabelas.

Cenário AR- Atributos com Relacionamento

Esta secção apresenta os resultados da similaridade obtidos associados às medidas de similaridade *Cosine*, *Jaccard*, *Jaro-Winkler* e *Levenshtein*. Neste cenário são utilizadas as tabelas "*Store_Sales"* e *"Promotion"*, as quais se podem integrar através do atributo *"p_promo_sk"* (pertencente à tabela *"Promotion")* e *"ss_promo_sk"* (pertencente à tabela *"Store_Sales").*

É importante mencionar que este cenário apenas retrata a avaliação da similaridade entre os *headers*, sendo importante também avaliar a similaridade do conteúdo dos dados. Sem efetuar qualquer tipo de análise, é possível identificar algumas semelhanças nos nomes dos atributos mencionados anteriormente, nomeadamente *"Item"* e *"promo",* que retratam uma *substring* semelhante a ambos os atributos das diferentes tabelas.

A Figura 39 apresenta a média do valor da similaridade (intervalo entre 0 e 1) associado às várias medidas de similaridade perante as experiências realizadas (com este cálculo da média de similaridade apenas se pretende apresentar uma visão geral do cenário, sendo discutidos os resultados pormenorizadamente ao longo do mesmo). Considerando a Figura 39, é possível verificar que a medida *Jaro-Winkler* apresenta uma média de similaridade superior (0.5290) às restantes medidas, tendo a medida de *Jaccard* o valor médio da similaridade menos elevado seguida pela medida de *Cosine* (0.0929) e, posteriormente, *Levenshtein* (0.1869). Através deste tipo de análise, não é possível tirar uma

conclusão credível da medida que melhor se adapta à avaliação dos *headers*, uma vez que obter a média de similaridade mais elevada não significa que se está perante a medida mais adequada. Esta consideração advém do facto de que apenas é importante obter uma similaridade notável nos atributos em que realmente **são** similares, de modo a que o utilizador esteja apto a tecer algumas críticas sobre os resultados obtidos. A obtenção de uma média de similaridade elevada confirma que o algoritmo avaliou pares de atributos com uma similaridade considerável que, de certa forma, poderá confundir o utilizador final com um conjunto de questões pertinentes tais como "Qual é o par de atributos mais semelhante num conjunto de 10 pares de atributos com uma similaridade elevada?".

Média de Cosine Similarity	0,0929
Média de Jaccard Similarity	0,0177
Média de Jaro-Winkler Similarity	0,5290
Média de Levenshtein Similarity	0,1869

Figura 39. Resultados Médios da Similaridade do Cenário AR.

Com as afirmações expostas anteriormente, é pretendido que se chegue à conclusão que é importante obter um valor notável de similaridade, **mas** também é importante que os algoritmos de similaridade detetem quando os pares de atributos não são similares e lhes atribuam um reduzido valor de similaridade de modo a filtrar os pares atributos que não são similares. Essa é uma das justificações para as médias das medidas de *Jaccard*, *Cosine* e *Levenshtein* se apresentarem em valores reduzidos quando comparadas às medidas de *Jaro-Winkler*. Em contextos *Big Data*, que pelo próprio contexto apresenta diversidades no volume, variedade e velocidade de dados, quanto menor o número de decisões que o utilizador tiver de tomar, maior será a fluidez nos processos de integração. Com isto pretende argumentar-se que, perante um processo de integração, é mais plausível tomar decisões sempre que se está perante dois ou três possíveis pares de atributos de similaridades, do que estar perante dez ou vinte pares de atributos (que no limite podem nem ser similares).

Analisadas as características das tabelas utilizadas no presente cenário importa agora observar os resultados obtidos. Devido ao número de pares de atributos gerados entre as tabelas *"Promotion"* e *"Store_Sales"* (437 pares de atributos), apenas são selecionados 25 pares de atributos. Esses 25 pares de atributos são os pares que apresentaram uma similaridade mais elevada, representados através da Tabela 7, porém é possível consultar os resultados completos no Apêndice 1 – Resultados Cenário AR. Os resultados encontram-se ordenados através da média das quatro medidas de forma decrescente sendo que

os resultados situados no intervalo entre 0,80 e 1 encontram-se em cor verde e os resultados situados entre 0,70 e 0,80 encontram-se em cor laranja.

Tabela 7. Resultados da similaridade do Cenário AR.

Pares de Atributos *(Promotion- Store_Sales)*	***Jaccard***	***Cosine***	***Jaro-Winkler***	***Levenshtein***
p_promo_sk-ss_promo_sk	**0.70**	**0.84**	**0.87**	**0.82**
p_item_sk-ss_item_sk	0.67	**0.82**	**0.90**	**0.80**
p_end_date_sk-ss_sold_date_sk	0.41	0.65	**0.81**	0.67
p_start_date_sk-ss_sold_date_sk	0.30	0.63	0.68	0.60
p_promo_id_b-ss_promo_sk	0.36	0.57	0.69	0.50
p_promo_name-ss_promo_sk	0.36	0.57	0.69	0.50
p_discount_active-ss_ext_discount_amt	0.39	0.61	0.61	0.47
p_promo_sk-ss_cdemo_sk	0.21	0.42	0.66	0.55
p_promo_sk-ss_hdemo_sk	0.21	0.42	0.66	0.55
p_start_date_sk-ss_store_sk	0.16	0.51	0.67	0.47
p_item_sk-ss_cdemo_sk	0.07	0.34	**0.74**	0.55
p_item_sk-ss_hdemo_sk	0.07	0.34	**0.74**	0.55
p_end_date_sk-ss_sold_time_sk	0.09	0.36	**0.72**	0.47
p_start_date_sk-ss_sold_time_sk	0.08	0.38	0.66	0.40
p_promo_sk-ss_store_sk	0.06	0.29	0.66	0.45
p_item_sk-ss_store_sk	0.07	0.31	0.63	0.45
p_item_sk-ss_promo_sk	0.07	0.22	0.68	0.45
p_start_date_sk-ss_item_sk	0.05	0.33	0.63	0.40
p_start_date_sk-ss_addr_sk	0.05	0.25	0.69	0.40
p_end_date_sk-ss_store_sk	0.11	0.33	0.61	0.31
p_end_date_sk-ss_item_sk	0.06	0.29	0.63	0.38
p_promo_sk-ss_item_sk	0.07	0.22	0.67	0.40
p_cost-ss_wholesale_cost	0.19	0.42	0.45	0.29
p_item_sk-ss_customer_sk	0.06	0.20	0.66	0.43
p_end_date_sk-ss_addr_sk	0.06	0.19	**0.70**	0.38

Considerando a Tabela 7, é possível verificar os resultados da similaridade por pares de atributo, tendo em conta as diferentes medidas de similaridade. Assim, confirma-se que, de facto, os pares de atributos *"p_promo_sk - ss_promo_sk"* e *"p_item_sk - ss_item_sk"* obtiveram uma similaridade superior em todas as medidas, como foi mencionado anteriormente, uma vez que em ambos os atributos existem *substrings* semelhantes, tais como *"item"*, *"promo"* e *"sk"*. Destaca-se também o par de atributos "*p_end_date_sk - ss_sold_date_sk"* que obtiveram um resultado de similaridade 0.81 na medida de *Jaro-Winkler*, 0.65 na medida de *Cosine* e 0.67 de na medida de *Levenshtein*.

De acordo com o mencionado na análise da Figura 39, é possível confirmar que a medida *Jaro-Winkler* apresenta resultados de similaridade elevados, tais como os primeiros pares de atributos *("p_promo_sk-ss_promo_sk"* e *"p_item_sk-ss_item_sk")* que obtiveram um resultado de similaridade

de 0.87 e 0.90, respetivamente. De igual forma, num aspeto menos positivo, é importante salientar que esta medida apresenta um resultado de similaridade de 0.7 no último par de atributos na Tabela 7 *("p_end_date_sk-ss_addr_sk")*, par esse que não apresenta nenhum nome/entidade em comum. Os resultados desta medida, de certa forma, encontram-se refletidos na média da mesma. No entanto, verifica-se que apesar de as restantes medidas apresentarem um valor médio inferior, estas exibem um valor elevado nos elementos que efetivamente apresentam semelhanças entre si.

Segunda a Tabela 7, a medida de *Jaccard* apresenta como valor mais elevado de similaridade 0.70 no par de atributos "*p_promo_sk - ss_promo_sk"*, que se trata de um valor reduzido quando comparado com as restantes medidas, apresentando também valores de similaridade reduzidos nos restantes pares de atributos. Isso deve-se, maioritariamente, ao método de cálculo da medida em questão. Como é possível observar na secção 4.3, esta medida tem como referência de similaridade o número de caracteres em comum mas não considera a sua distribuição, ou seja, o número de vezes que estes se repetem ao longo da *String*, afetando assim os seus resultados.

Na Figura 40, encontra-se representada uma comparação dos resultados da similaridade obtidos entre as medidas de *Jaccard* e *Cosine*, sendo possível verificar que estas medidas se relacionam entre si, na medida em que reconhecem com elevada similaridade; os pares de atributos *"p_promo_sk - ss_promo_sk"* e *"p_item_sk - ss_item_sk"* e, posteriormente, é registado um decréscimo no valor da similaridade no mesmo intervalo entre pares de atributos, registados na Figura 40 através dos quadrados de cor preta.

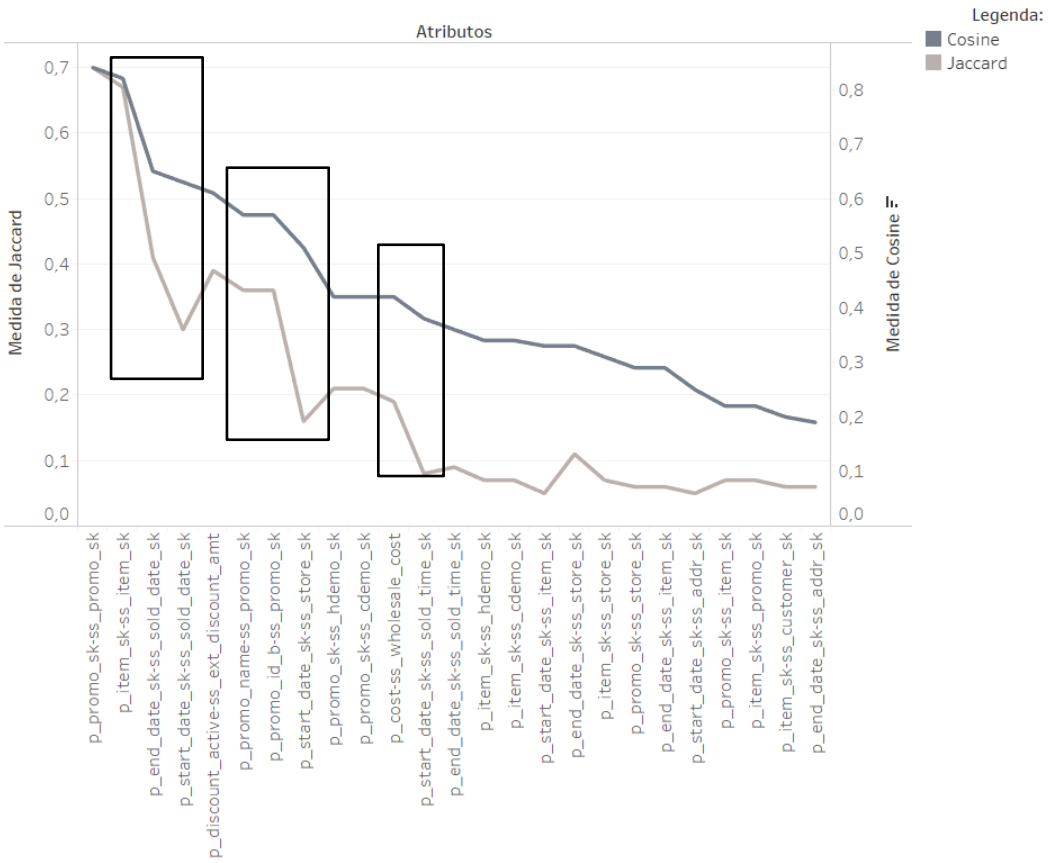

Figura 40. Comparação da similaridade entre Jaccard e Cosine. Cenário AR.

A análise realizada anteriormente à Figura 40 vem confirmar as afirmações presentes na secção 4.3, nas quais é mencionado que os métodos de *Jaccard* e *Cosine* seguem uma fórmula de cálculo semelhante. No entanto, a medida de *Cosine* considera a distribuição de caracteres repetidos em ambas as *Strings*, sendo essa a razão de apresentar valores de similaridade mais elevados do que a medida de *Jaccard*.

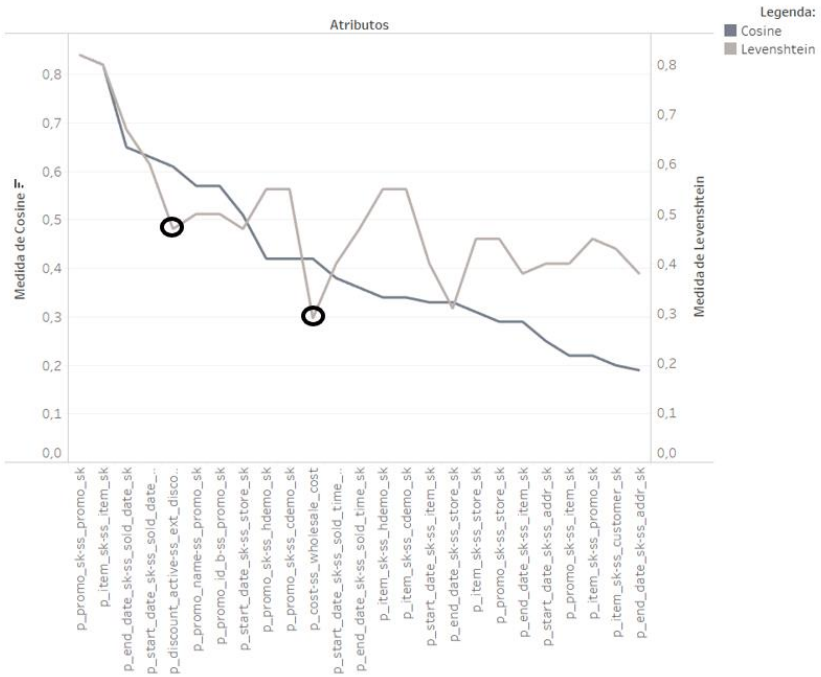

Figura 41. Comparação da similaridade entre Levenshtein e Cosine. Cenário AR.

Recorrendo agora à Figura 41, é possível fazer uma análise idêntica à anterior, mas agora comparando as medidas *Cosine* e *Levenshtein*. É possível observar que os primeiros pares de atributos têm, em ambas as medidas, resultados semelhantes, porém, a medida de *Levenshtein* apresenta um valor da similaridade superior à medida de *Cosine* em 64% das observações.

Não obstante, é possível verificar na Figura 41 um conjunto de pares de atributos nos quais a medida de *Levenshtein* regista um decréscimo acentuado nos valores da similaridade circundados de cor preta. Este **fenómeno** corrobora o que foi descrito na secção 4.3, na qual é mencionado que o tamanho dos objectos *Strings* é um fator a considerar e que, de certa forma, influencia o resultado da medida de similaridade, uma vez que se o tamanho é elevado e a distância de *Levenshtein* reduzida, o resultado da similaridade será reduzido quando comparado com outras medidas de similaridade. Tal afirmação é apresentada pelo autor do presente documento e verifica-se nos exemplos representados na Figura 41 no par de atributos entre "*p_cost - ss_wholesale_cost",* no qual o *objeto "p_cost"* e *"ss_ wholesale _cost"* são comparados. É possível averiguar que ambos os objetos *String* possuem o objeto "*cost"* em comum contribuindo, assim, para que a distância de *Levenshtein* não seja elevada. Porém, uma vez que a fórmula de calculo utiliza o comprimento da maior *String* como divisor que, neste contexto, seria *"ss_ wholesale _cost"* (19 caracteres) o resultado será reduzido quando comparado a outras medidas de similaridade. É relevante mencionar as particularidades das medidas de similaridade para que, posteriormente, sejam sintetizados os resultados para a escolha da medida utilizada na plataforma, que

será aquela que apresentar valores mais elevados em ambos os cenários, tendo sempre em conta o objetivo dos mesmos.

No que diz respeito ao tempo de processamento, uma vez que nenhum dos pares de atributos demorou **mais** de 1 segundo a ser calculado para todas as medidas de similaridade, a análise do tempo de processamento foi efetuada em **milissegundos**. Ainda assim, apenas 29 dos 437 pares de atributos (cerca de 6.63% da totalidade) obtiveram um tempo de processamento de 1 milissegundo, não se observando nenhum par de atributos com tempo de processamento superior a 1 milissegundo.

Todavia, é possível verificar, através da Figura 42, que a medida de similaridade *Jaro-Winkler* registou 14 pares de atributos com um tempo de processamento de 1 milissegundo (48% dos elementos), seguindo-se a medida de *Cosine* com 9, *Levenshtein* com 4 e, por fim, a medida de *Jaccard* com 2 pares de atributos.

De acordo com a Tabela 6, presente na secção 4.3, as medidas em que era esperado um tempo de processamento mais elevado dizem respeito às medidas de *Jaro-Winkler* e *Levenshtein*, devido à complexidade que é representada através de **n * m**, em que **n** e **m** representam o número de caracteres das respetivas *Strings* computadas. Através da Figura 42, verifica-se que a medida de *Levenshtein* obteve um resultado inferior à medida de *Cosine*, medida essa cuja complexidade é representada por **n + m**. Deste modo, tendo em conta o cálculo efetuado para o resultado da similaridade de *Cosine*, é possível que tal resultado seja influenciado pelo cálculo da distribuição de caracteres em cada *String*, uma vez que acarreta uma maior carga ao sistema ao invés da medida de *Jaccard* que não efetua a distribuição de caracteres, apresentando apenas 2 pares de atributos com tempo de processamento de 1 milissegundo.

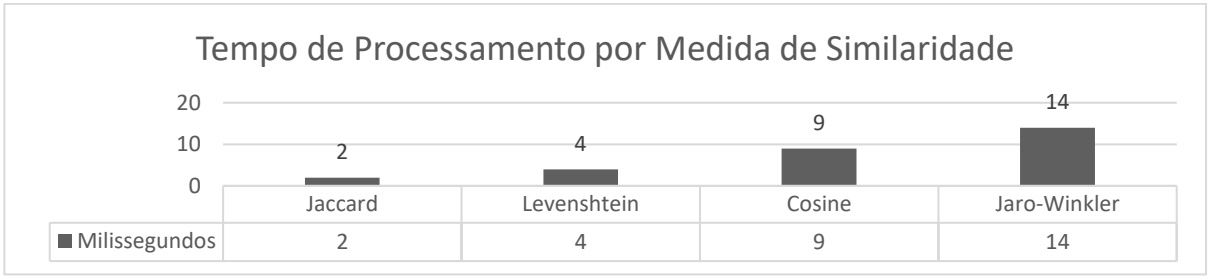

Figura 42. Tempo de Processamento dos Headers por Medida de Similaridade entre Promotion e Store_Sales. Cenário AR.

Cenário ASR – Atributos sem relacionamento

Esta secção apresenta os resultados da similaridade obtidos associados às medidas de similaridade *Cosine*, *Jaccard*, *Jaro-Winkler* e *Levenshtein*. Neste cenário são utilizadas as tabelas

"Store_Sales" e *"Income_Band"* que, ao invés das tabelas utilizadas no cenário **AR**, e segundo o modelo de dados apresentado na secção 4.2, **não** apresentam atributos em comum.

É relevante mencionar que este cenário apenas retrata a avaliação da similaridade entre os *headers* das tabelas mencionadas anteriormente e, uma vez que as tabelas não apresentam atributos em comum, é esperado que os resultados provenientes das medidas de similaridade não apresentem uma similaridade semelhante ao cenário anterior, sendo esse o principal objetivo deste cenário, ou seja, avaliar as medidas capazes de detetar contextos em que os *headers* não são similares.

A Figura 43 apresenta a média do valor da similaridade (intervalo entre 0 e 1) associado às várias medidas de similaridade perante as experiências realizadas. Considerando assim a Figura 43, é possível verificar que a medida de *Levenshtein* apresenta uma média de similaridade mais elevada (0.51) quando comparada às restantes medidas, tendo a medida de *Jaccard* o valor médio da similaridade menos elevado (0.01) seguida pela medida de *Jaro-Winkler* (0.06) e posteriormente pela medida de *Cosine* (0. 19).

Através deste tipo de análise, não é possível tirar uma conclusão credível da medida que melhor se adapta à avaliação dos *headers*, mas é possível observar que a medida de *Levenshtein*, perante a sua média, apresenta algumas debilidades em reconhecer que as tabelas analisadas não têm atributos em comum. A tabela *"Income_Band"* é composta apenas por 3 atributos, nomeadamente *"ib_income_band_sk"*, "*ib_lower_bound"* e *"ib_upper_bound",* e o sub conjunto de *Strings* que é comum aos atributos de ambas as tabelas trata-se apenas da *substring "sk"*, pertencente ao atributo *"ib_income_band_sk"* e*,* como tal, é previsível que os pares de atributos com o valor de similaridade mais elevado apresentem a *substring "sk".*

A interpretação dos resultados poderá variar perante o leitor do documento, uma vez que na presente temática subsistem um conjunto de questões, nomeadamente "A partir de que valor de similaridade se considera um par de atributos similar?", "A partir de 0.7 é considerado similar?", "Inferior a 0.5 não é similar?".

Média de Cosine Similarity	0,1961
Média de Jaccard Similarity	0,0100
Média de Jaro-Winkler Similarity	0,0622
Média de Levenshtein Similarity	0,5151

Figura 43. Resultados Médios da Similaridade do Cenário ASR.

A obtenção de uma média de similaridade reduzida para as medidas de similaridade *Cosine*, *Jaccard* e *Jaro-Winkler*, confirma que o algoritmo não reconheceu a presença de atributos similares, retratando assim o principal objetivo deste cenário.

Com as afirmações expostas anteriormente, é pretendido enfatizar a importância de os algoritmos de similaridade detetarem contextos em que os pares de atributos não são similares e lhes atribuam um reduzido valor de similaridade de modo a filtrar os pares atributos que não são similares, sendo essa uma das justificações para as médias das medidas de *Jaro-Winkler*, *Cosine* e *Levenshtein*.

Analisando as médias das medidas de similaridade obtidas no cenário **AR**, presentes na Figura 39, verifica-se a permanência nos valores médios em relação às medidas de *Jaccard* e *Cosine*, ou seja, ambas as medidas mantêm os valores médios reduzidos, ao invés da medida de *Jaro-Winkler* que no cenário **AR** apresentava uma média elevada (0.5290) e no presente cenário apresenta uma média reduzida (0.062). O mesmo se verifica na medida de *Levenshtein* que, no cenário **AR** apresenta uma média reduzida e, no presente cenário, apresenta uma média elevada (0.5151), não demonstrando ir ao encontro do objetivo do mesmo (uma vez que os atributos não se relacionam, seria de esperar uma média de similaridade reduzida).

Analisadas as características das tabelas utilizadas no presente cenário, importa agora observar os resultados obtidos. Devido ao número de pares de atributos gerados entre as tabelas *"Income_Band"* e *"Store_Sales"* (70 pares de atributos), apenas são selecionados os 25 pares de atributos, representados na Tabela 8 (é possível consultar os resultados na sua totalidade no Apêndice 2 – Resultados Cenário ASR). Os resultados encontram-se ordenados através da média das quatro medidas de forma decrescente, sendo que os resultados situados no intervalo entre 0.60 e 1 encontram-se em cor verde e os resultados situados no intervalo entre 0.30 e 0.60 encontram-se em cor laranja.

Tabela 8. Resultados da similaridade do Cenário ASR.

Pares de Atributos *(Promotion- Store_Sales)*	***Jaccard***	***Jaro-Winkler***	***Levenshtein***	***Cosine***
ib_income_band_sk-ss_sold_time_sk	0,08	**0,38**	**0,65**	**0,35**
ib_income_band_sk-ss_sold_date_sk	0,04	**0,31**	**0,67**	**0,41**
ib_income_band_sk-ss_customer_sk	0,08	0,28	**0,68**	**0,35**
ib_income_band_sk-ss_store_sk	0,04	0,29	**0,63**	**0,35**
ib_income_band_sk-ss_item_sk	0,05	0,25	**0,61**	**0,35**
ib_income_band_sk-ss_promo_sk	0,04	0,24	**0,63**	**0,35**
ib_income_band_sk-ss_cdemo_sk	0,04	0,16	**0,69**	**0,35**
ib_income_band_sk-ss_hdemo_sk	0,04	0,16	**0,64**	0,29
ib_income_band_sk-ss_addr_sk	0,05	0,17	0,55	0,29
ib_income_band_sk-ss_net_paid_inc_tax	0,07	0,24	0,52	0,16
ib_lower_bound-ss_customer_sk	0,04	0,15	0,57	0,14
ib_income_band_sk-ss_wholesale_cost	0	0,12	0,51	0,24

Pares de Atributos *(Promotion- Store_Sales)*	***Jaccard***	***Jaro-Winkler***	***Levenshtein***	***Cosine***
ib_lower_bound-ss_ext_discount_amt	0,04	0,12	0,5	0,21
ib_upper_bound-ss_ext_discount_amt	0,04	0,12	0,5	0,21
ib_lower_bound-ss_net_apid	0	0	0,55	0,29
ib_upper_bound-ss_customer_sk	0,04	0,15	0,51	0,14
ib_upper_bound-ss_promo_sk	0	0	0,54	0,29
ib_upper_bound-ss_net_apid	0	0	0,54	0,29
ib_income_band_sk-ss_net_apid	0	0	0,59	0,24
ib_lower_bound-ss_wholesale_cost	0	0	0,59	0,24
ib_income_band_sk-ss_coupon_amt	0	0,07	0,51	0,24
ib_upper_bound-ss_coupon_amt	0	0,16	0,52	0,14
ib_lower_bound-ss_ticket_number	0	0,07	0,55	0,19
ib_upper_bound-ss_ticket_number	0	0,07	0,55	0,19

Considerando a Tabela 8, é possível verificar os resultados da similaridade por pares de atributo, tendo em conta as diferentes medidas de similaridade. Portanto, confirma-se que, de facto, os pares de atributos cuja *substring "sk"* é comum a ambos os atributos obtiveram uma similaridade superior nas medidas de *Cosine*, *Jaro-Winkler* e *Levenshtein*, notando-se uma debilidade nesta particularidade em relação à medida *Jaccard*. Esta particularidade é notória nas primeiras 8 linhas da Tabela 8. É importante que as medidas de similaridade detetem este tipo de particularidades, ou seja, perante um cenário real, apesar de não reconhecer que se trata da mesma entidade, com a presença da *substring "sk"* em ambos os atributos é possível reconhecer que existe algo em comum entre os pares de atributos; não obstante, é essencial reconhecer que esta informação não é suficiente para retirar uma conclusão do que se trata.

Perante o objetivo pretendido neste cenário, é possível confirmar que as medidas de *Jaccard*, *Cosine* e *Jaro-Winkler* apresentam resultados de similaridade menos elevados, destacando a medida de *Jaccard* que classificou 20% dos pares de atributos com valores de similaridade superiores a 0 (zero), atribuindo aos restantes 80% a similaridade 0 (zero). Por sua vez, a medida de *Jaro-Winkler* classificou 41% dos pares de atributos com valores de similaridade superiores a 0 (zero), atribuindo aos restantes 59% a similaridade 0 (zero). Por fim, a medida de *Cosine* classificou todos os pares de atributos com valores de similaridade superiores a 0 (zero). Porém, verifica-se que cerca de 76% dos pares de atributos se encontram com valores de similaridade inferiores a 0.20 encontrando-se os restantes pares de atributos num intervalo entre 0.2 e 0.41 (valor máximo obtido pela medida de *Cosine)*.

Na Figura 44, está representada a comparação dos resultados da similaridade obtidos entre as medidas de *Cosine*, *Jaccard* e *Jaro-Winkler*, sendo possível observar a presença de 2 principais *clusters* de dados, nos quais se evidencia que o *Cluster* 2 incide sobre todos os pares de atributos com a presença da *substrings "sk"* em ambos os atributos, como por exemplo o par de atributos *"ib_income_band_**sk***

- *ss_sold_time_**sk**".* No que diz respeito ao *Cluster* 1, verifica-se que incide maioritariamente sobre os pares de atributos com a similaridade reduzida. Este tipo de análise foi realizado com auxílio da ferramenta Tableau, através do algoritmo de *clustering* k-means[24] que, perante um determinado número de *cluster*s k, particiona os dados em *cluster*s k. Cada *cluster* apresenta um centro (centróide) que diz respeito ao valor médio de todos os pontos no *cluster*. Por sua vez, o algoritmo k-means localiza os centros por meio de um procedimento iterativo que minimiza as distâncias entre os pontos individuais de um *cluster* e o centro do *cluster*.

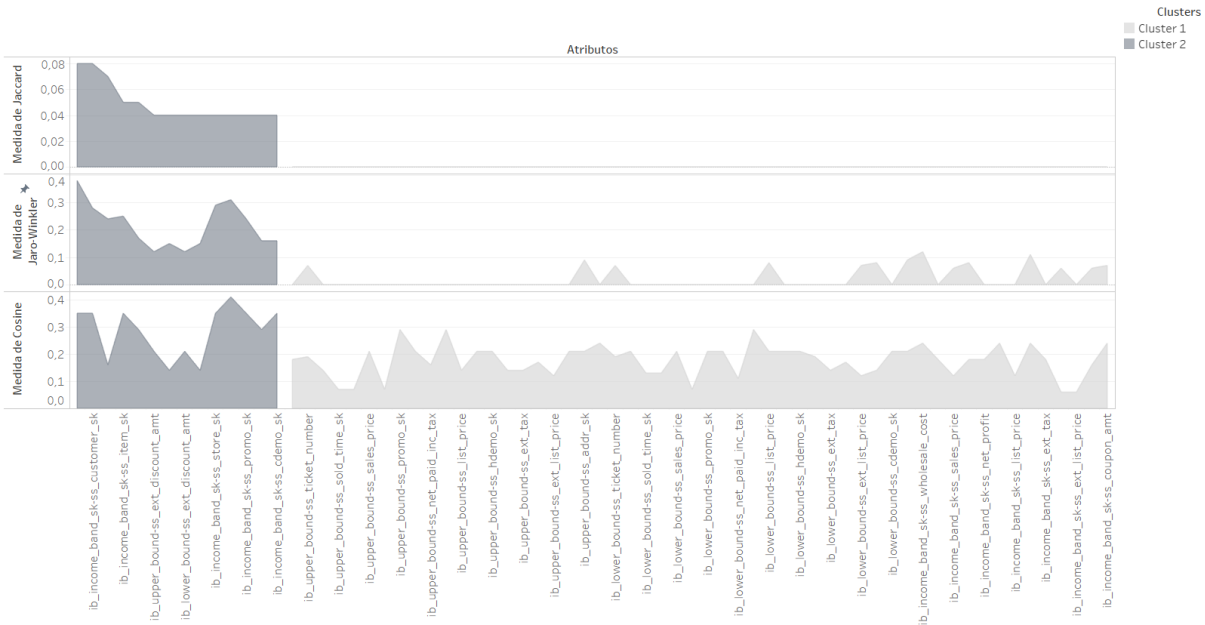

Figura 44.Comparação da similaridade entre Jaccard e Jaro-Winkler. Cenário ASR.

Finalmente, é importante tecer algumas observações sobre os cenários realizados anteriormente. Os dados provenientes do conjunto de dados TPC-DS, apesar de sintéticos, são dados cujos atributos são capazes de fornecer um leque elevado de diferentes nomes e diferentes entidades que se encontram, ou não, relacionados. No cenário **AR,** é importante evidenciar os valores elevados das medidas de similaridade de *Jaro-Winkler* e de *Cosine*, assim como as debilidades em algumas particulares apresentadas pelas medidas de *Jaccard* e *Levenshtein*. No que diz respeito à medida de *Jaccard*, os seus resultados no cenário **AR** apresentam-se reduzidos quando, comparados com as restantes medidas, devido maioritariamente à fórmula de cálculo da similaridade (esclarecido anteriormente). Verifica-se, também que as medidas de *Cosine*, *Jaccard* e *Jaro-Winkler* são capazes de reconhecer contextos em que os pares de atributos/entidades não são similares, ao invés da medida de

[24] https://help.tableau.com/current/pro/desktop/en-us/clustering.htm

Levenshtein (cenário **ASR)** que apresenta valores de similaridade considerados elevados para o cenário/objetivo em questão. O método de cálculo das medidas de similaridade é uma particularidade a ter em conta, como a sua complexidade que, neste cenário, não é um fator crucial devido aos tempos de processamento apresentados, no entanto, no contexto do Cenário B será certamente um fator a considerar.

Em suma, apesar da medida de *Jaro-Winkler* apresentar valores de similaridade elevados no cenário **AR**, considera-se que a medida de *Cosine* apresenta uma maior credibilidade, tendo em conta não só os resultados obtidos nos diferentes cenários apresentados mas também o seu cálculo da similaridade, uma vez que conta com duas particularidades, nomeadamente o *matching* de caracteres entre dois objectos *String* e a análise da distribuição de caracteres dos objetos *String*, ou seja, quantas vezes se repete cada caracter em cada objeto *String*. A medida de *Jaro-Winkler* apresenta um elevado número de pares de atributos com valores de similaridade elevada e, em contextos *Big Data*, é mais plausível tomar decisões sempre que se está perante dois ou três possíveis pares de atributos similaridades do que estar perante dez ou vinte pares de atributos (que, no limite, podem nem ser similares).

4.6.2. Cenário B – Avaliação da similaridade do conteúdo dos dados

Prosseguindo com o protocolo de testes, é agora realizada a avaliação da similaridade do conteúdo dos dados que tem como base o modelo utilizado na secção 4.2. Através dos testes a realizar no presente cenário, pretende perceber-se qual a estratégia de integração que apresenta mais vantagens num BDW.

Ao nível de análise do tempo de processamento, pretende avaliar-se o impacto no processamento de dados nas diferentes medidas de similaridade com o aumento no volume de dados, sendo aplicados os quatro FE (1,3,5 e 10). Esta análise será realizada com o conjunto de dados do TPC-DS utilizando as tabelas "*Promotion"* e "*Store_Sales"*.

Ao nível de análise do tempo do valor da similaridade, pretende avaliar-se valor da similaridade gerado por cada medida. Esta análise será realizada com o conjunto de dados do Genoma utilizando os sub conjuntos de dados "*Ensembl"* e "*DisGeNET"*.

Pretende também, analisar-se a relação entre o tempo de processamento e o valor da similaridade associado a cada medida, a titulo de exemplo, quer isto dizer que é possível a existência de eventuais medidas cujos resultados de similaridade apresentam 86% de similaridade e o seu tempo de processamento estar compreendido nos 20 minutos e uma medida diferente apresentar 85% de

similaridade e o seu tempo de processamento estar compreendido nos 5 minutos. A diferença de 1% no valor da similaridade, num contexto de *Big Data,* revela-se importante para a organização? É importante debater estas questões e concluir que medida apresenta mais vantagens relativamente aos dois indicadores.

A análise do conteúdo de dados é um *workload* que acarreta uma elevada carga de trabalho para o *cluster*, exigindo que seja realizado o cruzamento entre todos os de atributos das tabelas a integrar, a título de exemplo, o cruzamento entre a tabela "*Promotion"* e "*Store_Sales"* geram 437 pares de atributos, ou seja, é necessário executar as funções dos métodos de similaridade 437 vezes. Assim, é pertinente também avaliar a escalabilidade das medidas de similaridade, ou seja, qual o tempo de execução com um fator de escala de 1? E com um fator de escala de 10? Atingirá uma escalabilidade linear? É importante tecer algumas considerações e estar ciente das realidades *Big Data*, uma vez que se está a testar um conjunto de dados com um tamanho inferior aos conjuntos de dados num cenário real, ou seja, sempre que surgir um novo conjunto de dados será que é necessário comparar com todas as tabelas existentes na organização? É possível imaginar a carga de trabalho sempre que seja necessário integrar um novo conjunto de dados e comparar o mesmo com as tabelas existentes nos BDWs de organizações como Amazon, Facebook ou Google? É certo que as organizações mencionadas anteriormente dispõem de infraestrutura altamente sofisticada, mas também se considera apenas um novo conjunto de dados. Porém, em organizações dessa dimensão o número de novos conjuntos de dados é significativo.

Relativamente ao cenário com um **FE 1**, este cenário apresenta como principal objetivo escolher a melhor medida a nível de tempo e resultado de similaridade. É **pertinente** questionar o porquê de não utilizar as medidas de Cosine, *Jaccard*, *Levenshtein* e *Jaro-Winkler* para a análise de similaridade do conteúdo de dados e isso deve-se ao facto de essas medidas não se encontrarem otimizadas para cenários *Big Data* sendo que a demonstração de tal afirmação vem a confirmar-se nos tempos de execução presentes na Tabela 9. Os tempos de execução apresentados na Tabela 9 apenas faz referência ao processamento de 5 dos 437 pares de atributos (produto cartesiano entre as colunas de *"Store_Sales"* e *"Promotion")* e, como se pode verificar, as medidas de similaridade de *Cosine*, *Jaccard*, *Levenshtein* e *Jaro-Winkler* apresentam tempos de processamento elevados quando comparadas, por exemplo, à medida de similaridade com base na distribuição de valores que, para um FE 1, obteve um resultado de **1.9** horas para processar os pares de atributos na sua **totalidade** (437 pares de atributos). Estas medidas (*Jaccard*, *Cosine*, *Levenshtein* e *Jaro-Winkler)* apresentam resultados satisfatórios e estão otimizadas para a comparação entre duas *Strings* (como por exemplo a análise de similaridade da

dimensão *headers)* sendo que neste contexto de análise de conteúdo de dados apresentam desafios de escalabilidade.

Tabela 9. Resultados do tempo de processamento das medidas de similaridade para um FE 1.

Medida de Similaridade	Tempo de processamento
Jaccard	6.9 horas
Jaro-Winkler	11 horas
Levenshtein	10.4 horas
Cosine	7.6 horas

Perante a Tabela 9, é possível observar que os algoritmos de *Cosine*, *Jaccard*, *Levenshtein* e *Jaro-Winkler* não se encontram otimizados para a análise de conteúdo de dados em contextos de *Big Data*, uma vez que apenas que se trata da análise do conteúdo de dados para um **FE 1**. O cálculo destas medidas de similaridade é uma particularidade a ter em conta, assim como a sua complexidade que, na análise da dimensão dos *headers* não é um fator crucial devido aos tempos de processamento se apresentarem reduzidos, mas, no contexto da dimensão de análise dos conteúdo de dados foi um fator a considerar. Como tal, não serão consideradas para a análise os FE 3, 5 e 10, uma vez que não acrescentam valor à análise.

A Tabela 10 apresenta os tempos de processamento para as medidas de similaridade com base na distribuição de valores e com base na intersecção e união dos valores para os FE 3, 5 e 10.

Tabela 10. Resultados do tempo de processamento das medidas de similaridade para um FE 3, 5 e 10.

Fatores de Escala	Medida de similaridade	Tempo de processamento	
FE 3	Medida com base na distribuição de valores	3.6 horas	
	Medida com base na interseção e união dos valores	2.4 horas	**-33.4%**
FE 5	Medida com base na distribuição de valores	5.7 horas	

Fatores de Escala	Medida de similaridade	Tempo de processamento	
	Medida com base na interseção e união dos valores	3.0 horas	-47%
FE 10	Medida com base na distribuição de valores	9.2 horas	
	Medida com base na interseção e união dos valores	3.3 horas	-64%

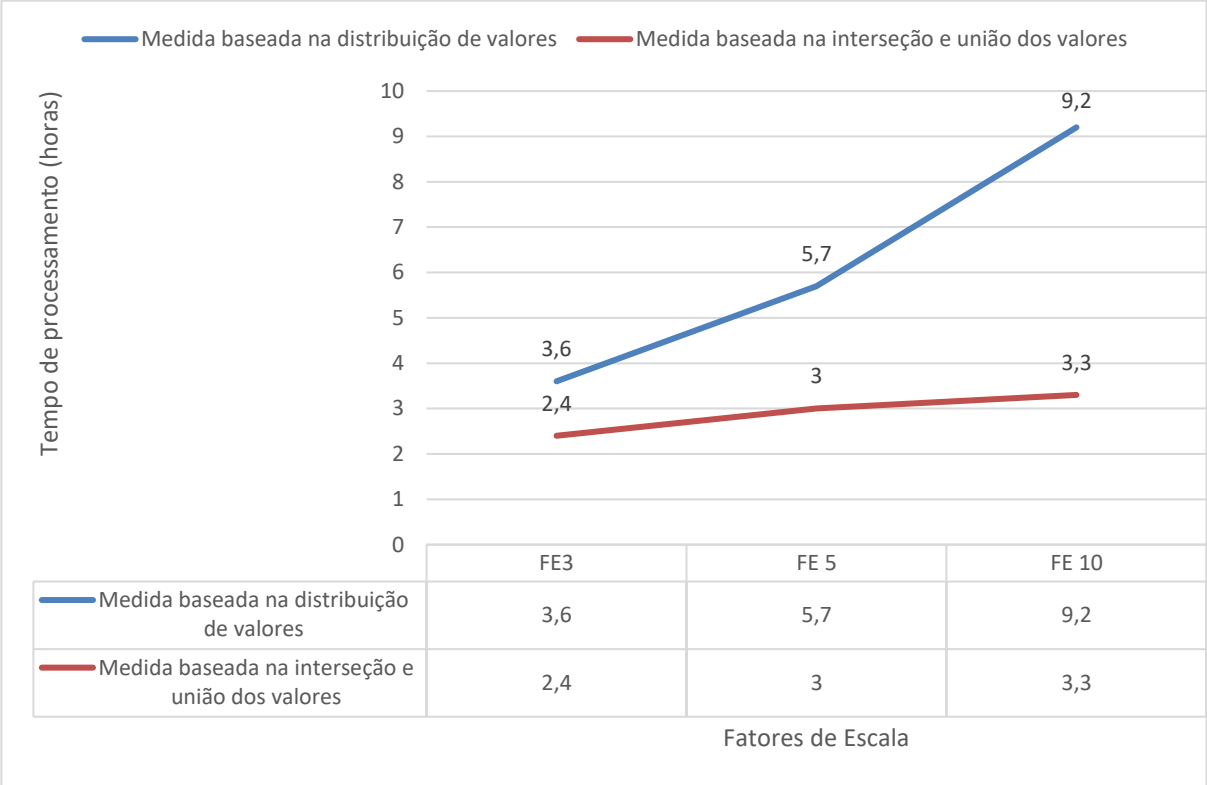

Figura 45. Evolução do tempo de processamento das medidas de similaridade por Fator de Escala.

Para além da Tabela 11, que nos indica os tempos de processamento de cada medida de similaridade, é possível observar na Figura 45 um gráfico da comparação dos tempos entre ambas as medidas.

Numa análise global, verifica-se que a medida baseada na intersecção e união dos valores apresenta uma performance superior quando comparada à medida baseada na distribuição de valores. Verifica-se que, para um FE 3, as medidas apresentam intervalos de tempo similares, no entanto, para um FE 5 e 10, verifica-se que a discrepância no intervalo de tempo de execução tende a aumentar.

Para um FE 5, o tempo de processamento da medida de similaridade com base na interseção e união de valores é inferior (**47%)** quando comparada à medida baseada na distribuição de valores. Para um FE 10, o tempo de processamento da medida de similaridade com base na interseção e união de valores é inferior (**64%)** quando comparada à medida baseada na distribuição de valores.

Observa-se que a medida baseada na distribuição de valores apresenta desafios sempre que o fator de escala aumenta, isto é, com o aumento do fator de escala o número de valores distintos para processar a distribuição de valores aumenta, resultando, por sua vez, no aumento do tempo de processamento final. O mesmo fenómeno não se verifica, com base na intersecção e união de valores, uma vez que não apresentou diferenças significativas com o aumento do fator de escala.

Para cenários em que o tempo de processamento é um fator **crucial**, é necessário utilizar a medida de similaridade que executa o processo de similaridade no menor tempo, por exemplo, no cenário FE 10 existe uma diferença de 5.9 horas entre ambas as medidas, que é um indicador relevante para ter em conta aquela que será a melhor medida.

Em *Big Data,* a otimização de processos é um fator importante e quanto menor o tempo de processamento dos processos maior será o intervalo de tempo para tomar decisões, isto é, um processo de similaridade que demore 10 horas a ser executado é um processo que torna uma organização muito dependente do mesmo uma vez que, no limite, os conjuntos de dados podem nem se apresentar como similares e são descartadas 10 horas em que podiam ser tomadas outras decisões no âmbito da integração, destacando também o impacto de um processo deste calibre ao ser executado num *cluster* (Chang, Tsai, & Wang, 2016).

Porém, apesar de a medida baseada na interseção e união de valores ser mais eficiente no tempo de processamento quando comparada à medida baseada na distribuição de valores, é necessário avaliar como esta medida se comporta relativamente ao valor da similaridade.

Uma vez que o tempo de processamento já se encontra analisado, segue-se a avaliação do valor de similaridade de ambas as medidas. Recapitulando, para a análise do mesmo serão utilizados conjuntos de dados associados a cenários reais, nomeadamente o conjunto de dados "*GWAS*" e *"Ensembl*" (Genoma Humano).

A Tabela 11 apresenta os valores dos resultados da similaridade de dados para a medida de similaridade baseada na distribuição de valores (Medida Distribuição de Valores) e para a medida de similaridade baseada na intersecção e união dos valores (Medida Intersecção e União). Por questões de espaço no documento, a Tabela 11 apenas apresenta os resultados de similaridade **positivos,** assumindo que os restantes atributos não acrescentam informação para a análise em questão uma vez

que apresentam valores de similaridade reduzidos. Os resultados de similaridade assinalados com cor verde representam os resultados de similaridade que se destacam na sua respetiva medida.

Na Tabela 11, contabilizam-se 31 pares de atributos com uma similaridade superior a 0, sendo possível analisar que ambas as medidas obtiveram valores de similaridade próximos, observando que o par de atributos "*chromosome/scaffold name - chr_id*" apresenta o valor da similaridade mais elevada para ambas as medidas, com **95.52%** de similaridade para a medida baseada na distribuição de valores e com **90.48%** para a medida baseada na intersecção e união dos valores. Na questão da similaridade, é importante referir que ambas as medidas, independentemente da fórmula de cálculo, foram capazes de identificar o par de atributos com maior similaridade no conjunto de dados.

Perante os resultados apresentados pelas duas medidas, o utilizador é capaz de identificar, tanto numa medida como na outra, qual o par de atributos que é similar para uma eventual integração entre os diferentes conjuntos de dados.

Tabela 11. Resultados da similaridade do conteúdo de dados para ambas as medidas.

Pares de Atributos	**Medida Distribuição de Valores(%)**	**Medida Intersecção e União(%)**
chromosome/scaffold name - chr_id	**95,52**	**90,48**
variant consequence - context	**64,35**	**42,11**
chromosome/scaffold name - intergenic	35,13	9,09
Strand - intergenic	35,13	**66,67**
transcript strand - intergenic	35,13	50
chromosome/scaffold name - beta	8,06	6,25
Strand - chr_id	4,58	8,7
transcript strand - chr_id	4,58	8,33
global minor allele count (all individuals) - chr_id	2,96	1,67
gene stable id - snp_gene_ids	1,9	2,51
associated gene with phenotype - reported gene(s)	1,89	1,46
associated gene with phenotype - mapped_gene	1,05	1,08
global minor allele count (all individuals) - beta	1,01	1,61
global minor allele count (all individuals)- upstream_gene_distance	0,82	1,46

Pares de Atributos	Medida Distribuição de Valores(%)	Medida Intersecção e União(%)
distance to transcript - downstream_gene_distance	0,82	0,63
global minor allele count (all individuals) - pvalue_mlog	0,76	2,31
gene stable id - upstream_gene_id	0,55	0,66
gene stable id - downstream_gene_id	0,55	0,66
distance to transcript - beta	0,38	0,72
distance to transcript - pvalue_mlog	0,28	0,84
chromosome/scaffold name - upstream_gene_distance	0,27	0,6
global minor allele count (all individuals)- downstream_gene_distance	0,27	1,46
variant start in cdna (bp) - downstream_gene_distance	0,27	0,6
variant name - snps	0,19	0,2
chromosome/scaffold position start (bp) -chr_pos	0,19	0,2
variant start in translation (aa) - pvalue_mlog	0,19	1,2
chromosome/scaffold name - merged	0,09	9,09
associated gene with phenotype - snps	0,09	0,2
transcript strand - merged	0,09	50
variant start in cdna (bp) - pvalue_mlog	0,09	1,1
pubmed id - pubmedid	4,82	5,23

Existem alguns resultados dos pares de atributos que necessitam de uma análise **detalhada** aos seus resultados, nomeadamente os pares de atributos "*transcript strand - merged"*, "*chromosome/scaffold name - intergenic",* "*Strand - intergenic"* e *"transcript strand - intergenic"*. É possível verificar que em todos os atributos referidos anteriormente se deteta uma discrepância significativa no valor da similaridade em ambas as medidas. Por exemplo, no par de atributos *"transcript strand - merged"* para a medida com base na distribuição de valores verifica-se uma similaridade de **0.09%** e para a medida com base na união de distribuição de valores verifica-se uma similaridade de **50%**. Tal fenómeno está inerente à fórmula de cálculo de ambas as medidas, isto é, a medida com base na intersecção e união de valores analisa quantos valores intersetados e distintos existem entre ambos os conjuntos mas **não** verifica quantas vezes esses valores se repetem em ambos os conjuntos e, perante isso, poderá existir um valor único que seja frequente num

conjunto de dados e não ser comum a ambos os conjuntos. A distribuição de valores, neste contexto, é uma função importante, no entanto, verificou-se que essa função apresenta problemas de escalabilidade sempre que aumentam o número de valores distintos.

Por fim, é importante tecer algumas críticas sobre ambas as medidas. A medida baseada na intersecção e união de valores apresenta resultados mais eficientes no tempo de processamento quando comparada à medida baseada na distribuição de valores, embora, esta última consiga obter resultados mais elevados de similaridade. É importante frisar que apesar da medida baseada na interseção e união dos valores apresentar valores inferiores de similaridade, esta medida, é capaz de detetar os pares de atributos similares apresentando uma diferença de 5% de similaridade quando comparada à medida baseada na distribuição de valores.

Como tal, em contextos de *Big Data,* o tempo de processamento é um fator a ter em conta devido aos volumes de dados elevados e a medida baseada na distribuição de valores apresenta desafios nessa vertente, devido aos motivos apresentados anteriormente (elevado número de valores distintos para calcular a distribuição de valores). Uma das vantagens da medida baseada na intersecção e união de valores é que se trata de uma medida **bidirecional**, ou seja, o valor da similaridade entre a interseção do conjunto A com o conjunto B é o mesmo valor da intersecção entre a interseção do conjunto B com o conjunto A, ao invés do que acontece com a medida baseada na distribuição de valores em que o valor da similaridade entre o conjunto A e B é **diferente** do valor da similaridade entre B e A devido à sua fórmula de cálculo.

4.7. Caso Real Aplicado ao Domínio Genético

No domínio da integração de dados, mais concretamente nos testes ao algoritmo de similaridade proposto, é importante testar o mesmo em contextos reais, ou seja, apesar do conjunto de dados proveniente do TPC-DS apresentar diferentes cenários, os seus conjuntos de dados são caracteristicamente sintéticos, sendo gerados segundo um padrão e sem qualquer anomalia. Assim, essa componente será testada através de um conjunto de dados fornecido pelo orientador sobre Genomas no âmbito da saúde.

Inicia-se esta abordagem com a seleção das medidas que apresentaram melhores resultados/vantagens nos cenários realizados anteriormente. Assim, selecionou-se a medida de *Cosine* para avaliar os *headers* e a medida baseada na intersecção e união dos valores para avaliação do conteúdo. O resultado final consiste num grafo de similaridade entre todos os conjuntos de dados.

Como tal, inicia-se a análise de similaridade da dimensão dos *headers* entre os diferentes conjuntos de dados, nos quais apenas são selecionados um *Top* 10 de pares de atributos, assumindo que os restantes pares de atributos não oferecem grande relevância para a análise em questão. É importante mencionar que esta análise é, posteriormente, suportada pela análise da similaridade do conteúdo dos dados. Inicia-se a análise à similaridade dos *headers* entre os conjuntos de dados "*GWAS*" e "*Ensembl*", que se encontra representada na Figura 46. É possível verificar que, neste *Top* 10, os valores da similaridade não se apresentam elevados, nos quais se destaca o par de atributos *"CHR_ID – PubMed ID"* com uma similaridade de 0.16 (valor entre 0 e 1).

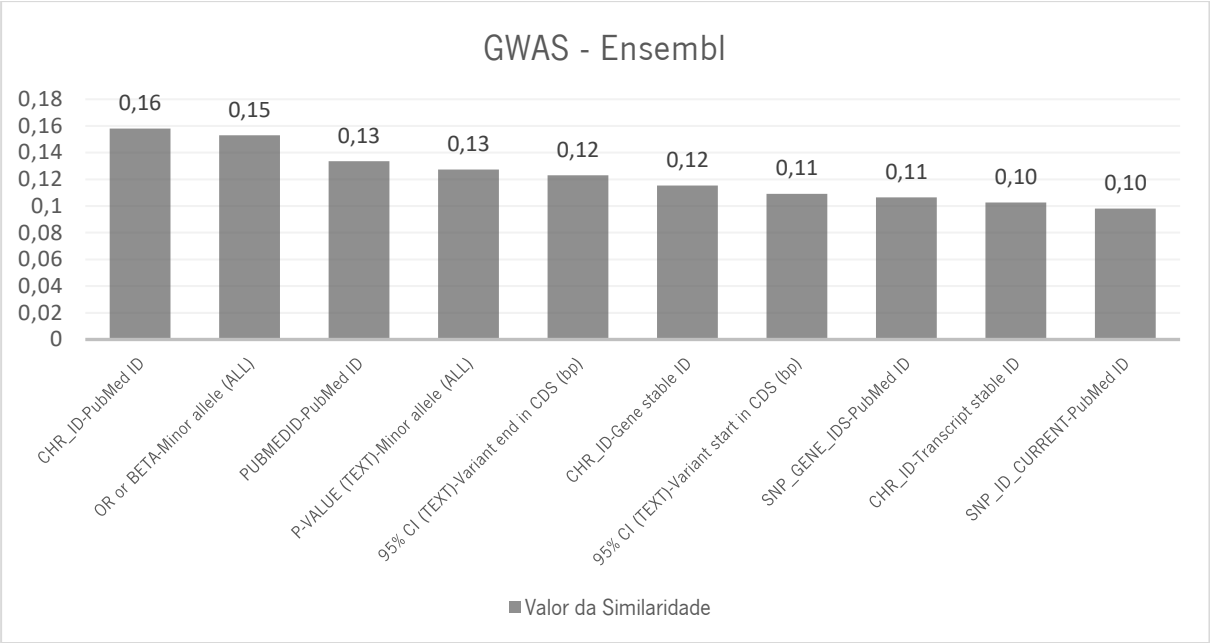

Figura 46. Análise da similaridade dos headers entre GWAS e Ensembl.

De seguida, na Figura 47, encontram-se representados apenas 7 pares de atributos, uma vez que os restantes apresentam um valor de similaridade igual a 0. É possível observar que, à semelhança da análise anterior, os pares de atributos identificados não apresentam uma similaridade elevada, destacando o par de atributos *"SNP_GENE_IDS - DSI"* com uma similaridade de 0.21. Para ambas as análises, existe a probabilidade de estes conjuntos de dados não serem similaridades, ou seja, a sua integração não irá trazer vantagens para a tomada de decisão.

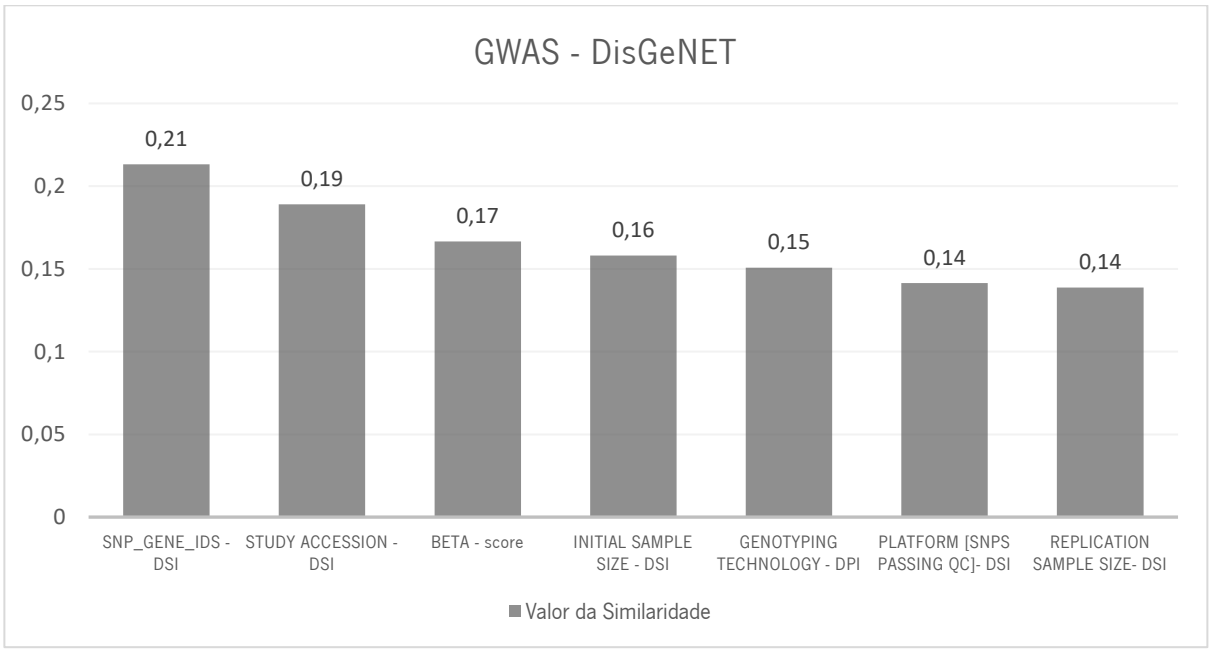

Figura 47. Análise da similaridade dos headers entre GWAS e DisGeNET.

Os resultados apresentados na Figura 48 são distintos das análises anteriores, na qual se destacam valores de similaridade consideráveis, tais como o par de atributos *"Chromosome/scaffold name - chromosome"* que apresenta um valor de similaridade de 0.64. Nos *Top* 4 presentes na Figura 48, verifica-se a existência de um elemento em comum, nomeadamente a *String* "*Chromosome"* que suporta o utilizador na tomada de decisão para uma eventual integração entre os pares de atributos em questão. A análise da similaridade do conteúdo de dados é um processo que requer uma elevada carga de trabalho para o *cluster* e, perante os resultados apresentados na similaridade dos *headers,* é possível otimizar esse processo de forma a calcular apenas a similaridade do conteúdo de dados para os pares de atributos cujo valor da similaridade dos *headers* seja superior a um determinado *threshold* definido pelo utilizador.

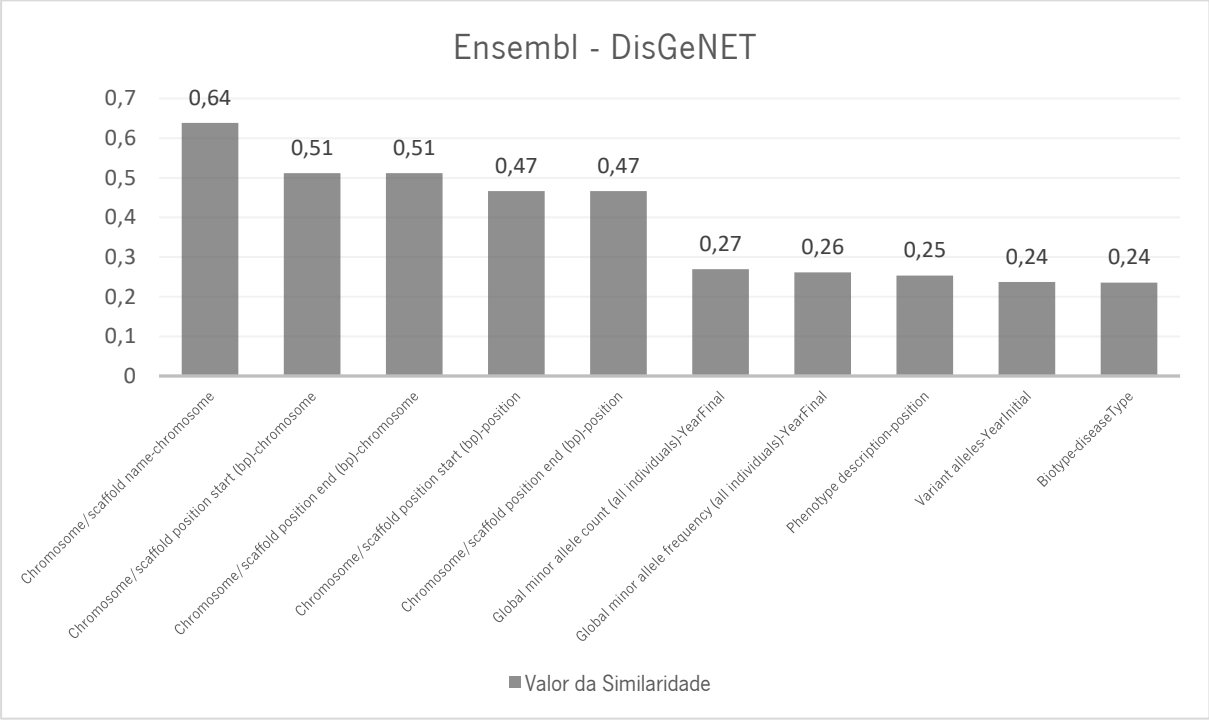

Figura 48. Análise da similaridade dos headers entre Ensembl e DisGeNET.

Nos resultados apresentados na Figura 49, destaca-se o valor de similaridade do par de atributos *"Clinical significance - Clinical Significance"* que apresenta um valor de similaridade de 0.92. Ao longo da análise realizada nesta secção, verifica-se que o conjunto de dados *"Ensembl"* apresenta valores de similaridade elevada quando comparado com outros conjuntos de dados, nomeadamente, *"AlzForum"* e *"DisGeNET".* A probabilidade de estes conjuntos se integrarem é elevada, sendo necessário recorrer à análise da similaridade do conteúdo de dados para suportar essa decisão, e o processo contrário ocorre de igual modo, ou seja, o conteúdo de dados pode ser similar, mas retratar duas entidades diferentes e,

nesse contexto, a análise de similaridade dos *headers* suporta essa decisão.

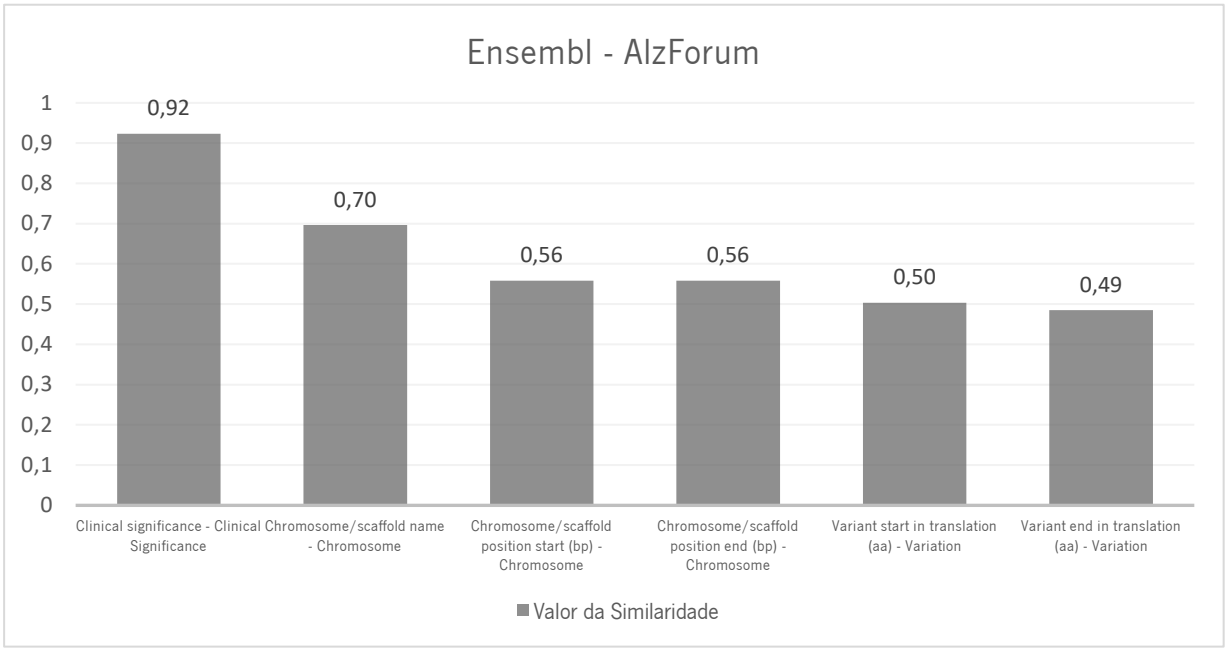

Figura 49. Análise da similaridade dos headers entre Ensembl e AlzForum.

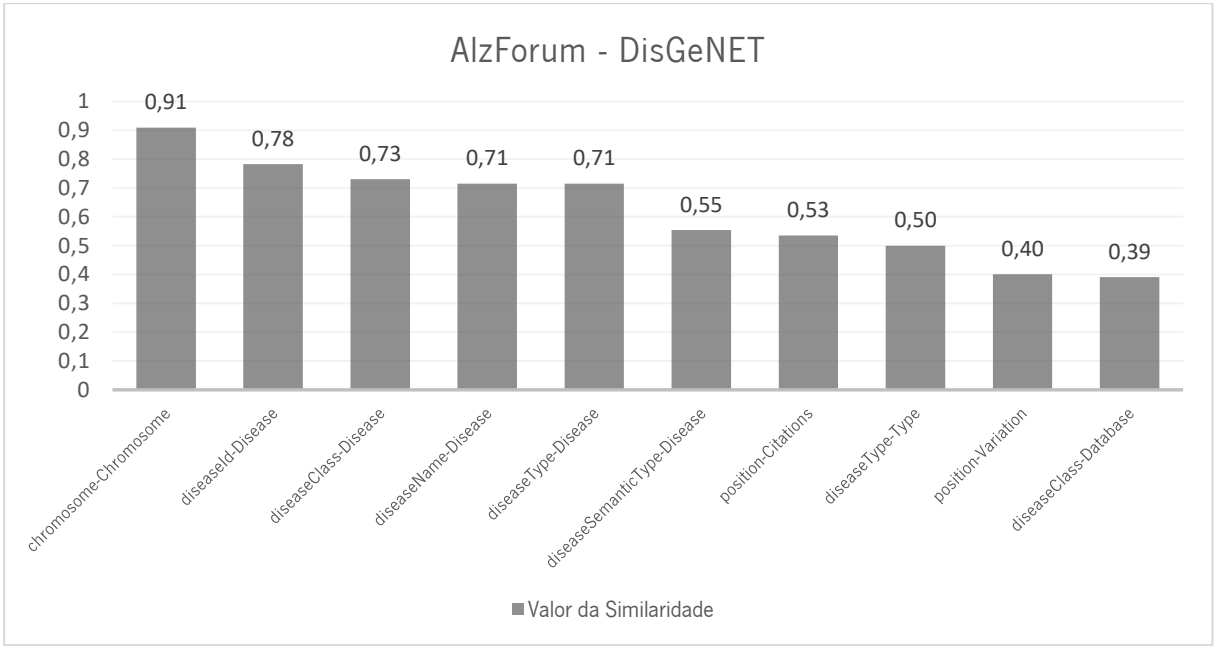

Figura 50. Análise da similaridade dos headers entre AlzForum e DisGeNET.

Nos resultados apresentados na Figura 50, destacam-se os valores de similaridade dos pares de atributos *"chromosome - Chromosome"* que apresenta um valor de similaridade de 0.91 e do par de atributos *"diseaseID - Disease"*. Com a análise destes resultados, verifica-se que na integração destes conjuntos de dados, existe um componente em comum entre ambos, nomeadamente "*disease"* que,

apesar de não ser os pares de atributos com maior valor de similaridade, é o componente que se verifica um maior número de vezes no *Top* 10.

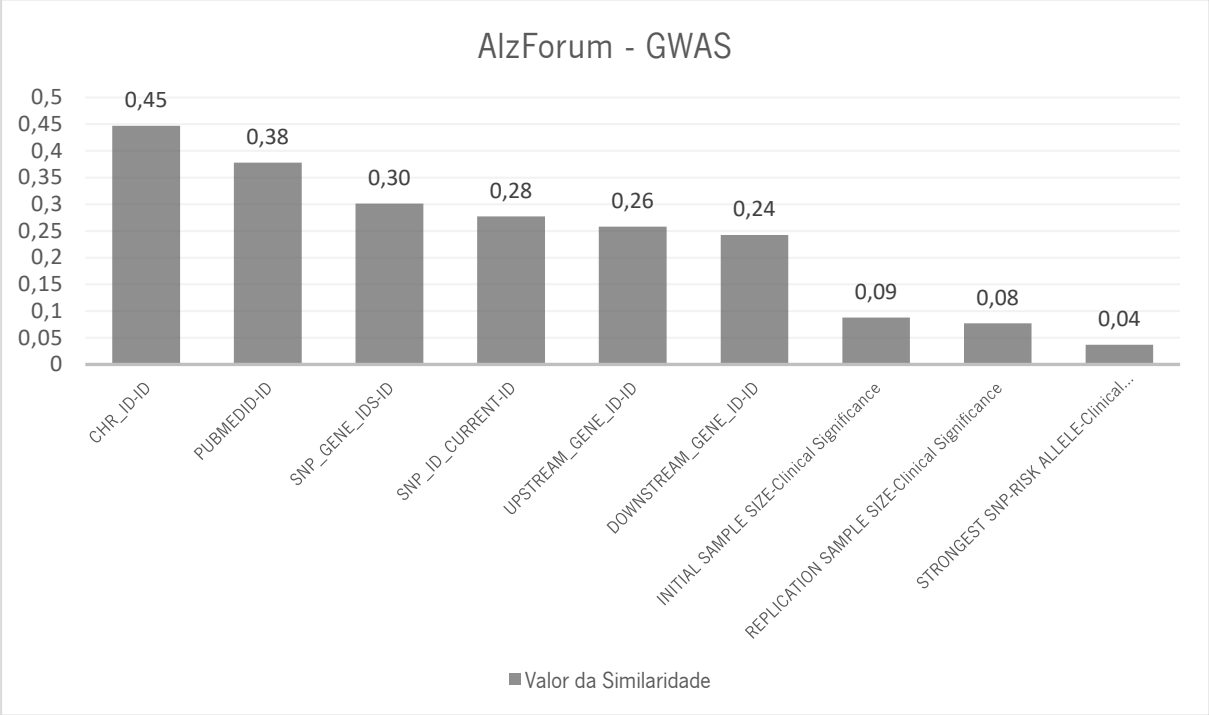

Figura 51.Análise da similaridade dos headers entre AlzForum e GWAS.

Terminando a análise dos *headers,* a Figura 51 faz referência à integração entre os conjuntos de dados *"AlzForum - GWAS"*, no qual apenas apresenta um *Top* 9 devido aos restantes pares de atributos apresentarem uma similaridade igual a 0. Verifica-se que o par de atributos "*CHR_ID – ID"* apresenta uma similaridade de 0.45, indicando essa informação que ambos os atributos se referem a um ID. Porém apenas no atributo referente conjunto de dados *"AlzForum"* se consegue obter informação suficiente para inferir que se trata de um ID de um cromossoma (CHR).

Terminada a fase de análise da similaridade dos *headers,* segue-se a análise da similaridade ao nível do conteúdo dos dados. Foi utilizada a medida de similaridade baseada na interseção e união dos valores, pelos motivo, mencionados na subsecção anterior, na qual é importante referir que se trata de uma medida bidirecional, ou seja, o valor de similaridade de integrar A com B é **igual** ao valor de similaridade de integrar B com A. É também a medida rápida (tempo de processamento).

Assim, inicia-se esta análise entre o conjunto de dados "*Ensembl – DisGeNET",* que se encontra refletido na Figura 52. Destaca-se os pares de atributos "*Chromosome/scaffold name - chromosome"* que apresentam uma similaridade de 90.48. Em contextos de integração de dados, o utilizador poderá apresentar-se indeciso na tomada de decisão sobre a seleção de um par de atributos para integrar os conjuntos de dados, mas recorrendo à análise dos *headers* é possível observar que o par de atributos

"*Chromosome/scaffold name - chromosome"* obteve uma similaridade de 0.64, ou seja, tem uma similaridade superior aos restantes pares de atributos (por exemplo, o par de atributos *"Chromosome/scaffold name - EI"* que obteve uma similaridade igual a 0) e, como tal, será o par de atributos mais indicado para realizar a integração entre ambos os conjuntos de dados.

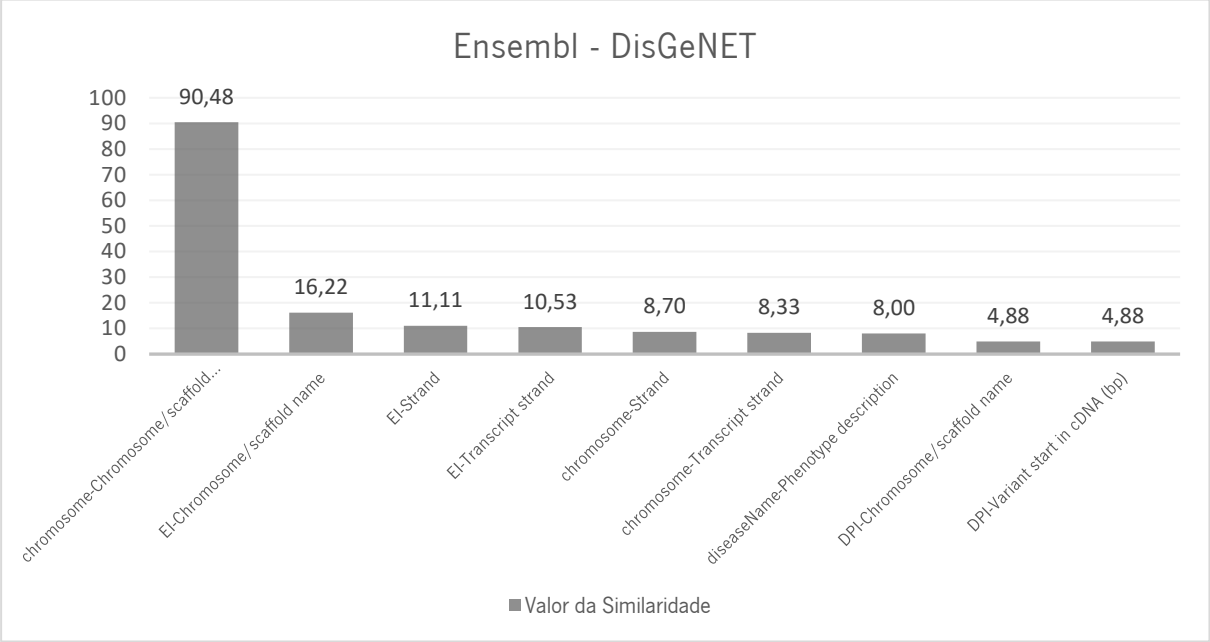

Figura 52. Análise da similaridade do conteúdo dos dados entre Ensembl e DisGeNET.

O par de atributos com o valor da similaridade mais elevado na análise dos *headers* entre *"Ensembl – AlzForum"* foi o par *"Clinical significance – Clinical Significance"* com 0.92. No entanto, tal como se pode verificar na Figura 53, esse par de atributos não apresenta uma similaridade superior a 0. Pode-se classificar que esse par de atributos não é similar uma vez que não apresenta qualquer valor em comum.

Porém verifica-se a existência de 2 possíveis pares de atributos com uma similaridade entre o intervalo 80% - 100%. Nestes casos, é necessário o utilizador obter mais informação sobre os dados, uma vez que a informação providenciada na análise dos *headers* não se verifica neste contexto. Tipicamente, existe ainda a possibilidade de recorrer a uma abordagem semântica através da análise de bibliotecas de ontologias para realizar uma análise mais profunda. Uma abordagem diferente será analisar a distribuição de dados dos pares de atributos com uma similaridade mais elevada com o propósito de verificar quais os pares de atributos têm um maior número de valores em comum.

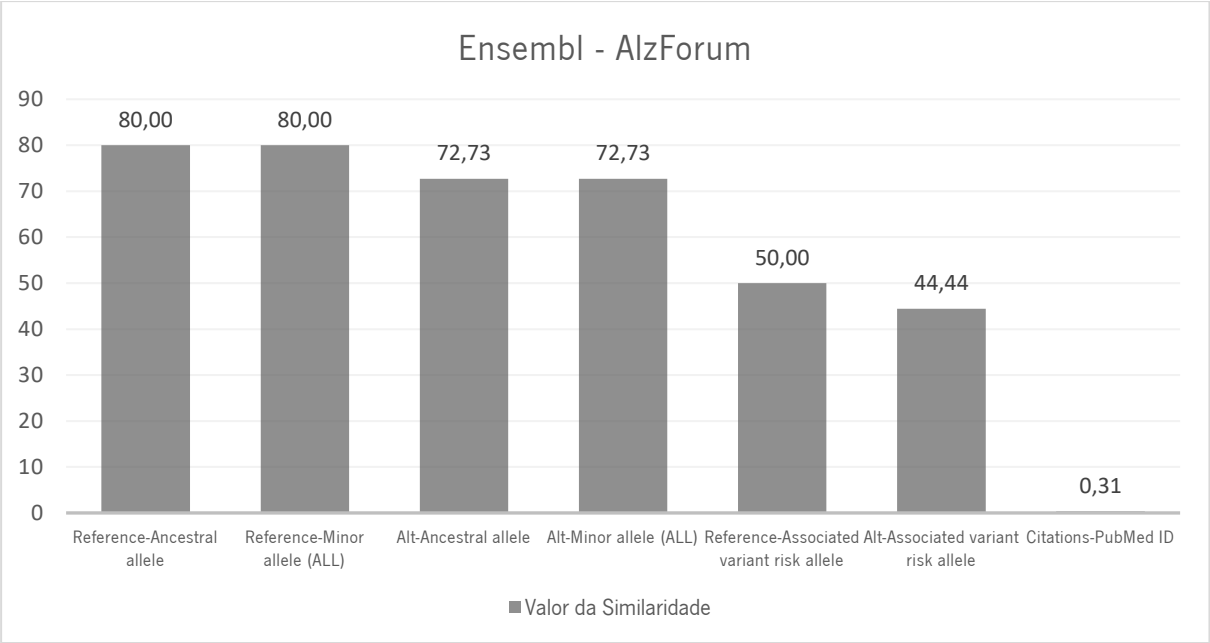

Figura 53. Análise da similaridade do conteúdo dos dados entre Ensembl e AlzForum.

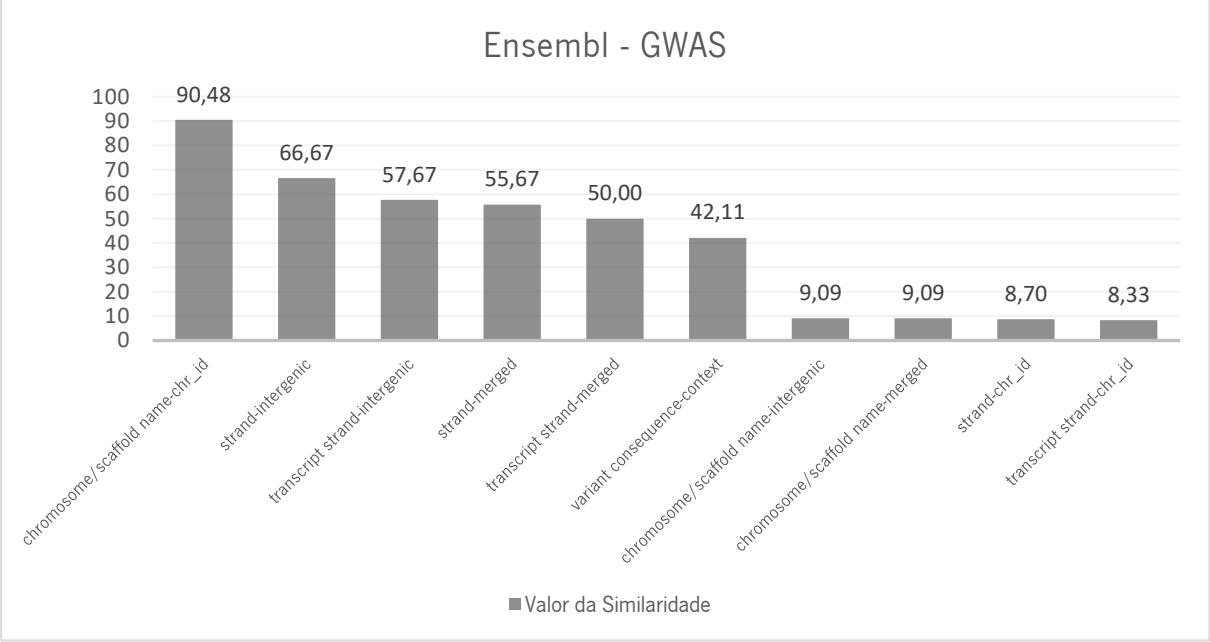

Figura 54. Análise da similaridade do conteúdo dos dados entre Ensembl e GWAS.

Para a integração entre os conjuntos de dados *"Ensembl – GWAS",* verifica-se, na Figura 54, um par de atributos com 90.48% de similaridade, no que diz respeito ao conteúdo dos dados, e, perante uma breve análise, observa-se que o par de atributos *"Chromosome/scaffold name - CHR_ID"* fazem referência ao mesmo objeto, no entanto, o atributo "*CHR_ID"* diz respeito à abreviatura de *"Chromosome".*

Para a integração entre os conjuntos de dados *"DisGeNET – AlzForum",* verifica-se, na Figura 55, que apenas 3 pares de atributos obtiveram valores de similaridade superiores a 0. Na análise da similaridade dos *headers* entre estes conjuntos de dados, o par de atributo *"diseaseName - Disease"* obteve uma similaridade de 0.71, suportando assim o utilizador na tomada de decisão de que este será um potencial par de atributos para integração de ambos os conjuntos de dados. Porém, com a análise ao conteúdo de dados verifica-se que a similaridade ao nível do conteúdo dos dados é de 7.41%, enfatizando assim a importância da execução e análise das duas dimensões (*headers* e conteúdo).

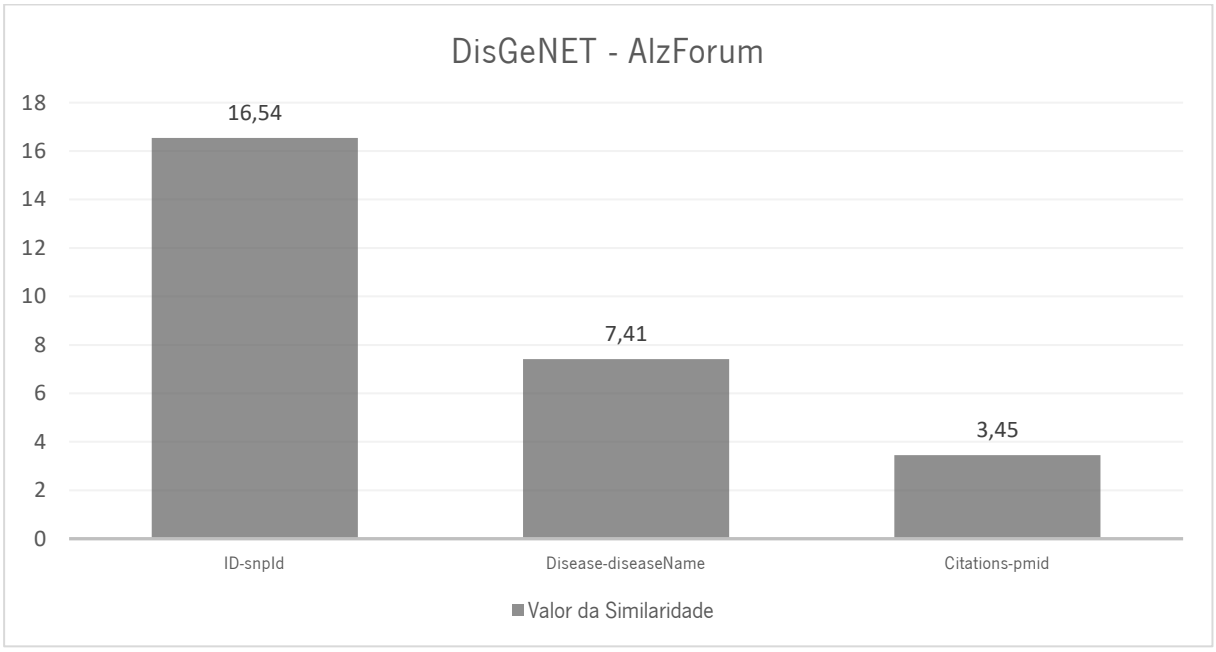

Figura 55. Análise da similaridade do conteúdo dos dados entre AlzForum e DisGeNET.

Para a integração entre os conjuntos de dados *"GWAS – AlzForum",* verifica-se, na Figura 56, que apenas 3 em 533 pares de atributos obtiveram uma similaridade maior do que 0. Porém, é importante salientar que apesar de a similaridade ser superior a 0, os resultados apresentados não acrescentam valor para uma organização numa perspetiva de integração de dados uma vez que não fazem referência ao mesmo objeto.

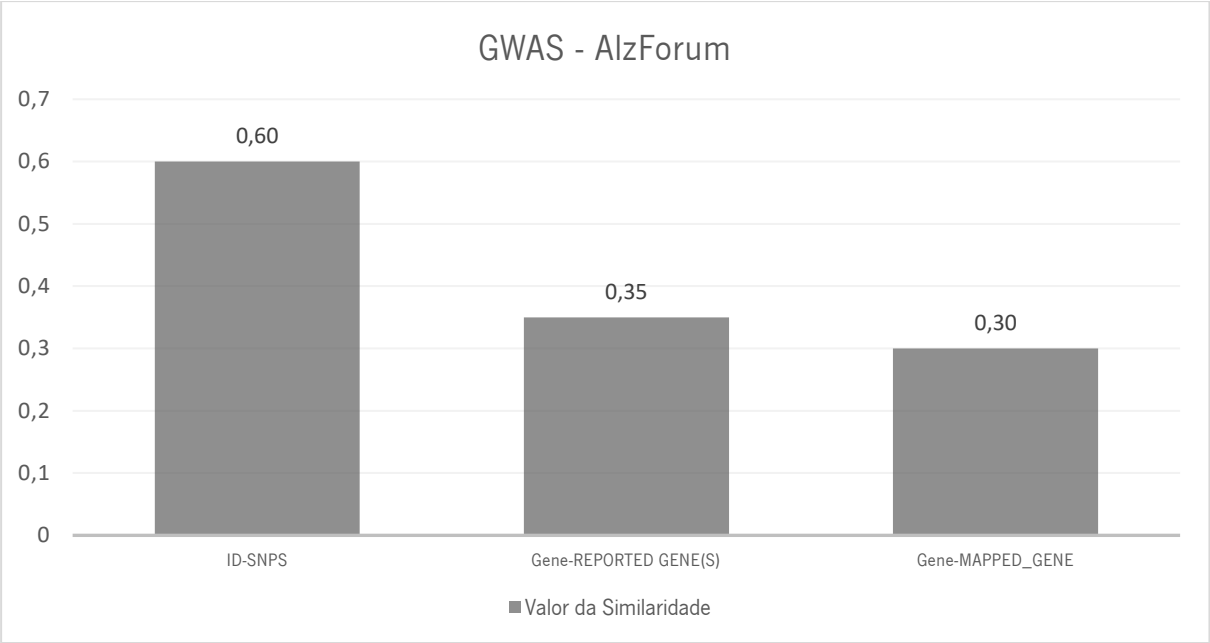

Figura 56. Análise da similaridade do conteúdo dos dados entre AlzForum e GWAS.

Para a integração entre os conjuntos de dados *"GWAS – DisGeNET",* verifica-se, na Figura 57, que o par de atributos *"chromosome – CHR_ID"* obteve uma similaridade de 100% no que diz respeito ao conteúdo dos dados e perante uma breve análise observa-se que a *String "CHR_ID"* é uma abreviatura de Chromosome. Este exemplo é uma das razões pela qual é importante analisar e testar as medidas num contexto real, no qual os utilizadores dão diferentes nomenclaturas para os mesmos atributos.

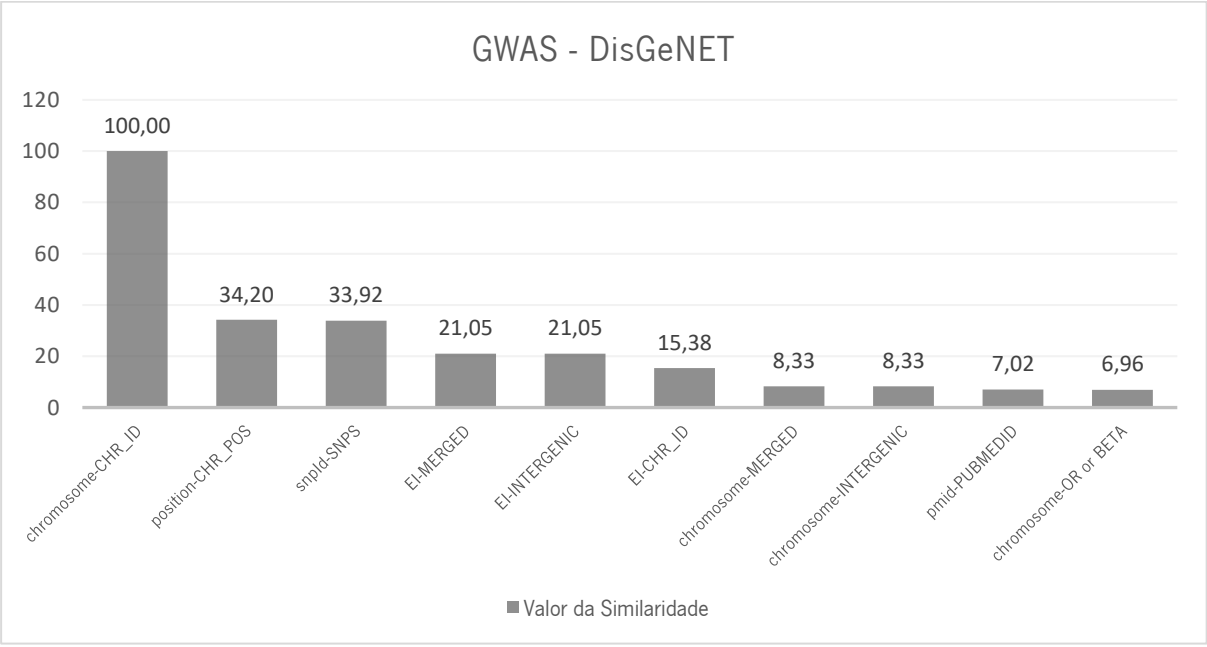

Figura 57. Análise da similaridade do conteúdo dos dados entre GWAS e DisGeNET.

É relevante frisar que estes conjuntos de dados são provenientes de um contexto real, cuja atribuição dos nomes de atributos não segue nenhum conjunto de regras para a atribuição dos mesmos. Como tal, é possível observar que nem sempre a similaridade dos *headers* está de acordo com a similaridade do conteúdo de dados, porém, esse processo é realizado com o propósito de auxiliar o utilizador na integração de dados. A dimensão de análise que deve ser tida mais em conta será a similaridade do conteúdo de dados que, perante os resultados, é possível inferir se realmente se está, ou não, perante o mesmo objeto. Assim, através da análise pormenorizada, é possível inferir o modelo baseado em grafos, representado na Figura 58, que se pretende armazenar na ferramenta Atlas, cuja informação é proveniente da ferramenta de *Data Profiling*, no qual se destaca as diversas relações de similaridade entre os vários nós assim como também os valores de similaridade associados. Na Figura 58, encontra-se representado cada conjunto de dados (cor azul) e cada atributo associado ao mesmo (cor amarela), no entanto, apenas se encontram representadas as principais relações entre atributos de conjuntos de dados diferentes devido às restrições na dimensão da imagem. É possível verificar que existem relações entre atributos de conjuntos de dados diferentes como, por exemplo, o par de atributos "*Chromosome/scaffold name - chromosome*" (pertence à tabela *"Ensembl* "e "*DisGeNET")* apresenta uma similaridade no conteúdo de dados de 85.58%. Por fim, é **importante** mencionar que a Figura 58 é o exemplo de uma primeira iteração/abordagem, pelo que numa nova iteração em trabalho futuro serão realizados ajustes com objetivo de melhorar o valor da similaridade, sendo uma das abordagens colocar todos os *headers* em maiúsculas por uma questão de uniformidade.

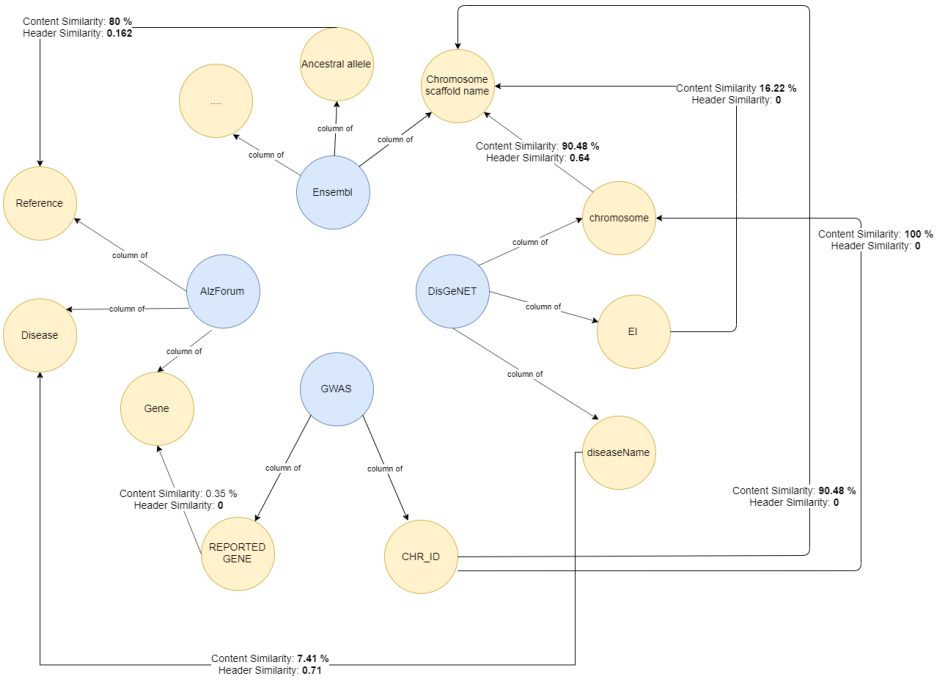

Figura 58. Grafo de similaridade aplicado ao Domínio Genético.

4.8. Síntese dos Resultados Obtidos

Esta secção marca o final do atual capítulo e, como tal, são aqui tecidas considerações sobre os diferentes aspetos abrangidos pelos testes e respetivos resultados alcançados.

Ao longo do atual capítulo, foram analisadas várias medidas de similaridade para duas dimensões diferentes, nomeadamente a dimensão dos *headers* e do conteúdo dos dados. Elaborando uma análise global em relação às medidas de similaridade para a avaliação dos *headers*, os resultam apresentam que, apesar da medida de *Jaro-Winkler* apresentar valores de similaridade elevados no cenário AR, verifica-se que a medida de *Cosine* apresenta uma maior credibilidade, tendo em conta não só os resultados obtidos nos diferentes cenários exibidos, mas também o seu cálculo da similaridade, uma vez que conta com duas particularidades, nomeadamente o *matching* de caracteres entre dois objectos *String* e a distribuição de caracteres dos objetos *String*.

Por sua vez, a medida de *Jaro-Winkler* apresenta um elevado número de pares de atributos com valores de similaridade elevada e, em contextos *Big Data*, é mais plausível tomar decisões sempre que se está perante dois ou três possíveis pares de atributos similares do que estar perante dez ou vinte pares de atributos.

Em relação às medidas de análise do conteúdo de dados, ambas as medidas apresentam valores de similaridade relevantes, sendo que ambas são capazes de identificar os pares de atributos similares. No entanto, a medida de similaridade com base na distribuição de valores apresenta desafios ao nível do tempo de processamento, devido ao cálculo da distribuição de valores por cada valor único, ou seja, à medida que o número de valores únicos aumentar irá ter um impacto direto no aumento do tempo de processamento. A medida com melhor desempenho (tempo de processamento menos elevado) diz respeito à medida de similaridade com base na intersecção e união de valores, uma vez que o seu tempo de processamento é inferior ao tempo de processamento da medida com base na distribuição de valores, acrescentando ainda que se trata de uma medida que adiciona valor à organização por se apresentar como uma medida bidirecional.

Perante as conclusões retiradas anteriormente, pode acrescentar-se que, tanto no cenário B como no caso real, foi constatada uma situação semelhante, nomeadamente o tempo elevado de processamento. Esse tempo elevado de processamento está maioritariamente refletido nos pares de atributos analisados que não trazem nenhum retorno a nível de informação para a tomada de decisão, ou seja, como é realizado um produto cartesiano entre ambos os conjuntos de dados (todos os atributos dos diferentes conjuntos de dados cruzam-se entre si), um elevado número de pares de atributos não

apresenta similaridade entre si, o que faz com que não se obtivesse o retorno esperado perante a carga imposta durante a execução do processo.

Assim, de forma a colmatar o desperdício de recursos entre pares de atributos não similares, a Figura 59 apresenta um algoritmo de similaridade integrado, ou seja, a análise dos *headers* está integrada com a análise do conteúdo de dados. Este algoritmo não é um algoritmo *standard,* requer a inserção por parte do utilizador de dois parâmetros. Com base na literatura, nomeadamente no trabalho de Zhu et al. (2019), o primeiro parâmetro refere-se ao limite *(threshold)* de similaridade pelo qual o utilizador pretende filtrar os pares de atributos em relação à similaridade dos *headers*. Esta inserção do limite de similaridade deve-se ao facto da existência de diferentes perspetivas e opiniões sobre o que realmente é similar e, perante isso, requer-se que seja o próprio utilizador a estabelecer o seu limite de similaridade. O segundo parâmetro refere-se à utilização da semântica (através de ontologias de dados) para a análise dos *headers* recorrendo ao dicionário de palavras WordNET.

A utilização da semântica deve-se ao facto da presença de atributos similares e as medidas de similaridade não serem capazes de detetar os mesmos. A título de exemplo, o atributo *"car"* e "*automobile*" com a medida de similaridade de *Cosine* obtém uma similaridade de 0, porém recorrendo à semântica (através de ontologias) obtém uma similaridade de 0.8462. Este algoritmo foi importado de um projeto *open-source* que se encontra no repositório https://github.com/jjlastra/HESML restringindo-se apenas à língua inglesa. É importante frisar que a sua utilização é facultativa, pelo que, caso seja utilizado, exige uma carga superior para o sistema (devido à procura das palavras no dicionário).

De modo a compreender a lógica apresentada na Figura 59, perante um novo par de atributos, é utilizado o algoritmo de *Cosine* para avaliar a similaridade dos *headers* e, caso a similaridade seja **superior** ao *threshold* proposto pelo utilizador, esse par de atributos é armazenado num *array* e, posteriormente, é calculada a similaridade do conteúdo de dados (através da medida de similaridade com base na intersecção e união dos valores) referente ao par de atributos. Os resultados produzidos, tanto na avaliação dos *headers* como na avaliação do conteúdo de dados, são armazenados para que, posteriormente, sejam transferidos para um repositório de metadados (capítulo 5). Na eventualidade do valor da similaridade entre os *headers* se apresentar **inferior** do que o limite proposto e o utilizador pretender recorrer à semântica, é realizado o processo semântico e, se o valor semântico produzido pelo mesmo for **superior** ao limite proposto, é realizado o mesmo processo mencionado anteriormente, caso contrário o par de atributos é descartado. É importante referir que é **classificado** os atributos como **similares** se o valor da similaridade entre os *headers* foi **superior** ao limite proposto pelo utilizador.

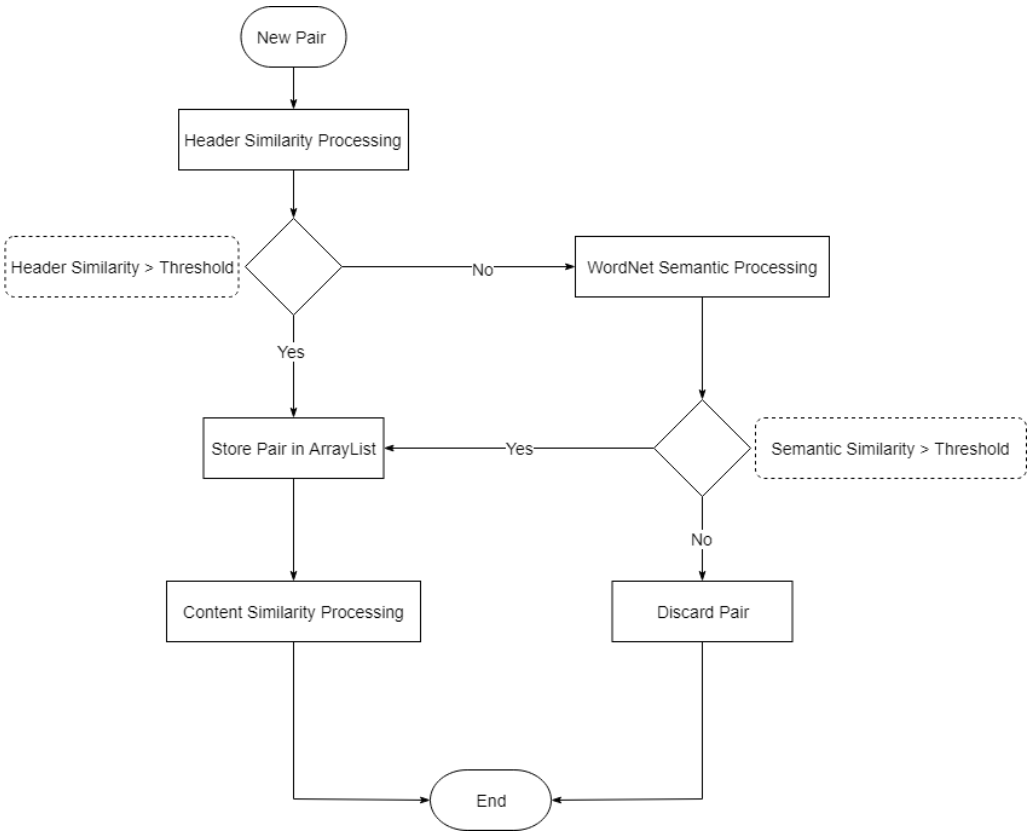

Figura 59. Algoritmo de análise da similaridade de dados.

Com este algoritmo, pretende reduzir-se o número de pares de atributos que são submetidos para a avaliação da similaridade ao nível do conteúdo de dados, que é o processo mais impactante ao nível de carga de trabalho. Como tal, surgem algumas questões pertinentes, tais como "Não será relevante analisar de igual forma todos os pares de atributos ao nível do conteúdo de dados? Com este processo, é possível que um par de atributos importante seja ignorado?". Respondendo às mesmas, é importante compreender o objetivo da execução destes processos. Se se executar todos os pares de atributos, não se irá obter melhorias no processamento e na integração de dados. No entanto, o utilizador terá a possibilidade de executar **todos** os pares de atributos, se propuser um limite de similaridade igual a 0 (zero), e, nesse sentido, o limite de similaridade não faz sentido ser integrado no algoritmo.

Porém, na suposição de ser encontrado um par de atributos através deste mecanismo, e que realmente seja similar, a nível organizacional é uma mais valia, uma vez que aumenta a eficiência na tomada de decisão e na poupança de recursos. Na eventualidade de **não** ser encontrado um par de atributos similar, o tempo de processamento dos pares de atributos selecionados é tipicamente reduzido e permite à organização repetir a execução do algoritmo e reduzir o nível de similaridade proposto de forma a executar novamente todos os pares de atributos.

5. GOVERNANÇA DE DADOS EM *BIG DATA WAREHOUSES*

Analisando o desempenho das medidas de similaridade e os algoritmos propostos anteriormente para integrar os novos conjuntos de dados num BDW, é também objetivo desta dissertação compreender como esses algoritmos podem ser integrados e complementados com outras ferramentas. Tal como se verificou nos trabalhos relacionados, têm sido propostas algumas abordagens (ferramentas e algoritmos) que procuram suportar a Governança de Dados em ambientes tradicionais e em ambientes *Big Data*. No entanto, tal como se verificou, em todas elas se identificam algumas limitações, quer seja ao nível do armazenamento de metadados quer seja devido ao facto de apresentarem direcionadas para *Data Lakes,* ao invés de BDWs.

Na presente dissertação, é notável o foco na integração e similaridade de dados devido, maioritariamente, aos desafios impostos na execução dos algoritmos. Porém, com o desenvolvimento da dissertação e com a realização da revisão de literatura, observaram-se a existência de outros aspetos a considerar sempre que se integrem novas fontes de dados. Tais aspetos fazem referência à **Qualidade** e **Segurança** de dados, por outras palavras, sempre que se integram novos conjuntos de dados é importante ter em conta a similaridade em relação aos dados que já subsistem no BDW. No entanto, com o objetivo de melhorar a tomada de decisão é relevante que estes dados se apresentem com a devida qualidade e, como tal, o cálculo de um conjunto de medidas e indicadores de qualidade revela-se importante. De forma a salientar a afirmação anterior, a Tabela 12 apresenta um conjunto de informação sobre duas potenciais colunas para integração com outra tabela num BDW. Se o utilizador considerar apenas a informação da similaridade das colunas, certamente a escolha irá recair para a coluna "*Client*" (devido à presença de uma similaridade de 100%), no entanto, com a disponibilização de uma medida de qualidade de dados, a decisão poderá ser diferente, uma vez que a coluna "*Costumer*" apesar de apresentar uma similaridade de 85%, apenas apresenta 3% de valores nulos, contribuindo para o enriquecimento do BDW.

Tabela 12. Análise de potenciais colunas para integração de dados.

Coluna	Percentagem de Nulos	Similaridade
Client	90%	100%
Costumer	3%	85%

No que à segurança de dados diz respeito, é necessário reunir informação sobre regras, permissões e políticas de segurança associadas não só às novas fontes de dados, mas também às tabelas presentes no BDW.

Assim, ao longo deste capítulo, será apresentada uma arquitetura de Governança de Dados e um caso de aplicação da mesma. É relevante referir que a arquitetura **apenas** abrange e colmata os desafios de Governança de Dados que se encontram entre a primeira camada da arquitetura da Figura 17, evidenciando o processo de enriquecimento através da obtenção do perfil dos dados que estão a chegar ao BDW e perceber de que forma é que se podem integrar com os dados que subsistem na segunda camada da arquitetura proposta por Costa e Santos (2017).

A ferramenta Atlas é utilizada como um repositório de armazenamento de metadados, permitindo o armazenamento de toda a informação presente num BDW, nomeadamente informação sobre as suas tabelas e colunas associadas. Para além disso, é armazenada também informação no que diz respeito à qualidade de dados, segurança de dados e, por fim, informação proveniente dos *Jobs* executados em Spark para a obtenção de informação das duas anteriores. O catálogo de dados organizacional presente no Atlas pode ser consultado na dissertação de Maria Inês Costa – Etiquetagem e rastreio de fontes de dados num *Big Data Warehouse* (de momento não se encontra publicada), no entanto, apenas se utilizam os nós (ferramenta basada em grafos) correspondentes aos desafios que devem ser colmatados nesta fase, mencionados ao longo desta secção.

O objetivo para esta fase é propor uma arquitetura funcional com todos os componentes integrados, sendo que em trabalho futuro se otimizará e avaliará o desempenho da mesma.

5.1. Conjunto de Dados

Nesta fase será utilizado o modelo de dados utilizado anteriormente na secção 4.2, através da tabela "*Promotion"*, proveniente do HDFS em formato *csv*, simulando uma nova fonte de dados que se pretende integrar no BDW (restantes tabelas).

5.2. Arquitetura para a Governança de Dados

A Figura 60 apresenta a arquitetura da solução de Governança de Dados, com todas as tecnologias utilizadas especificadas anteriormente. De referir que, em alguns componentes desta arquitetura, se podem considerar outras tecnologias, sendo que a escolha das tecnologias presentes na arquitetura se deve ao facto do autor deste documento se encontrar ambientado com as mesmas ao longo do seu percurso

académico(como é o caso do Spark, Atlas, Hive e Ranger) e por estar limitado temporalmente para a aprendizagem de novas tecnologias.

É relevante mencionar que todos os componentes da arquitetura se encontram devidamente articulados, à exceção do componente Controller UI, sendo este concebido em trabalho futuro.

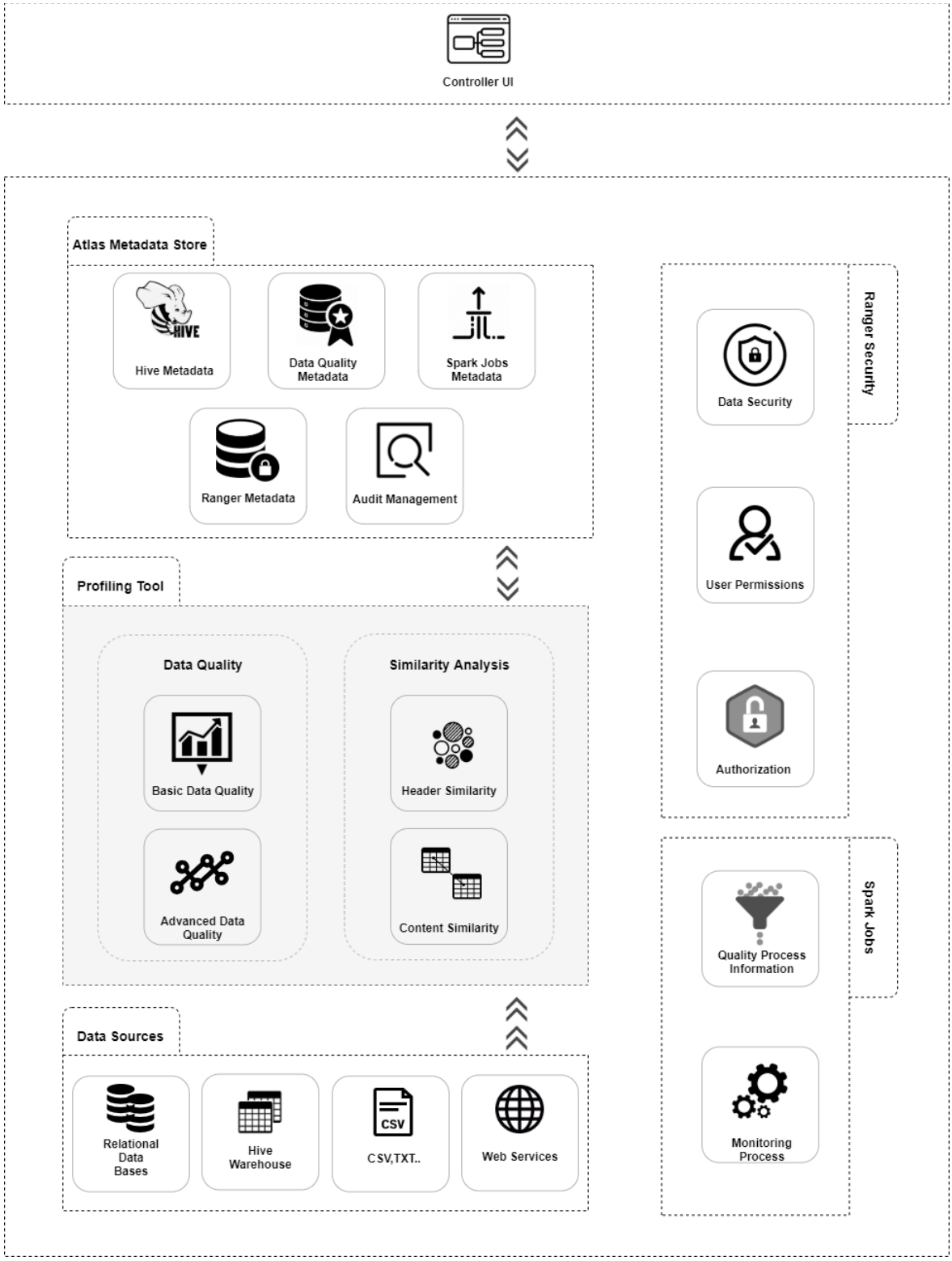

Figura 60. Arquitetura da Solução de Governança de Dados.

Na arquitetura presente na Figura 60, pode considerar-se que as fontes de dados são principalmente, dados em formato *batch*, através de ficheiros CSV, TXT, entre outros. É possível considerar, também, bases de dados relacionais, assim como tabelas Hive. Na Figura 60, encontra-se um componente preenchido de cor cinzenta *(Profiling Tool)* que diz respeito ao componente desenvolvido pelo autor deste documento e no qual se encontra o maior contributo.

Estes dados, assim que recolhidos, são processados recorrendo à componente ***Profiling Tool***, uma ferramenta de *Data Profiling* concebida em Spark, sendo definido o seu modelo de dados e o tipo de atributos. Esta ferramenta encontra-se dividida em dois sub módulos: o módulo da qualidade de dados e o módulo da similaridade de dados. Em relação ao módulo da qualidade de dados, são calculados um conjunto básico e avançado de estatísticas e indicadores relevantes, no que diz respeito às fontes de dados e aos respetivos atributos. Estas estatísticas e indicadores relevantes para o contexto têm como base o conhecimento adquirido na revisão de literatura, situada na secção 2.3 do presente documento. Assim, a Tabela 13 apresenta as medidas e indicadores de qualidade "calculados", através da ferramenta concebida.

Tabela 13. Análise da qualidade de dados de uma coluna. Baseado em Abedjan et al. (2017) e Naumann, (2014).

Nome	Tipo de Dados	Descrição
Tipo de dados	String	Tipo de dados associado à coluna
Valor máximo	float	Valor máximo da coluna
Valor mínimo	float	Valor mínimo da coluna
Comprimento máximo	int	Número máximo de carateres da coluna
Comprimento mínimo	int	Número mínimo de carateres da coluna
Número de *nulls*	int	Número de valores nulos
Número de valores "false"	int	Se o tipo de dados da coluna é *boolean* retorna o número de valores "false"
Número de valores "true"	int	Se o tipo de dados da coluna é *boolean* retorna o número de valores "true"
Percentagem de valores preenchidos	float	Percentagem de valores preenchidos
Percentagem de valores únicos	float	Percentagem de valores únicos
Número de valores únicos	int	Número de valores únicos

Nome	Tipo de Dados	Descrição
Número de linhas	int	Número de linhas
Frequência de valores	Map<String,String>	Frequência de determinados valores da coluna

Do mesmo modo a Tabela 14 apresenta as medidas e indicadores de qualidade "calculados" pelo componente *Profiling Tool* associado a uma determinada fonte de dados.

Tabela 14. Análise da qualidade de dados de uma tabela. Baseado em Abedjan et al. (2017) e Naumann, (2014).

Nome	Tipo de Dados	Descrição
Número de colunas categóricas	int	Número de colunas do tipo nominal
Número de colunas com datas	int	Número de colunas do tipo data
Número de observações	int	Número de linhas da tabela
Número de variáveis	int	Número de colunas da tabela
Número de colunas numéricas	int	Número de colunas do tipo numérico
Tamanho da fonte de dados	float	Tamanho da fonte de dados em megabytes
Base de Dados	String	Base de dados à qual as tabelas estão associadas

Posteriormente a analisar a qualidade dos dados, é necessário averiguar de que forma as novas fontes de dados se conseguem integrar com os dados que já subsistem no BDW e, como tal, são analisadas duas dimensões diferentes, nomeadamente a análise de similaridade aos *headers* entre as tabelas do BDW e as novas fontes de dados e, posteriormente, a análise da similaridade do conteúdo dos dados. O **algoritmo** utilizado neste processo diz respeito ao algoritmo proposto anteriormente ao longo deste documento, resultando assim no fornecimento dos seguintes indicadores, demonstrados na Tabela 15.

Tabela 15. Análise da similaridade das novas fontes de dados. Baseado em Abedjan et al. (2015) e Maccioni e Torlone (2018).

Nome	Tipo de Dados	Descrição
Coluna da fonte de dados	String	Coluna a ser comparada
Coluna do BDW	String	Coluna com a qual se irá comparar
Similaridade do conteúdo de dados	int	Valor compreendido entre 0 e 100, no qual 100 representa uma elevada similaridade e 0 uma similaridade reduzida

Nome	Tipo de Dados	Descrição
Similaridade dos *headers*	String	Valor compreendido entre 0 e 1, no qual 1 representa uma elevada similaridade e 0 uma similaridade reduzida
Threshold	Float	Limite estabelecido pelo utilizador na filtragem da similaridade dos *headers*

A tecnologia Atlas, através de uma interface REST, estabelece comunicações constantes com a componente *Profiling Tool* e, devido à quantidade de informação que é calculada/inferida pela mesma, surge a necessidade de que essa informação seja armazenada, resultando assim no **aumento** da eficácia e eficiência na apresentação de resultados e na **diminuição** da carga de trabalho. Visto que não existe a necessidade de executar novamente os processos de qualidade ou similaridade, a tecnologia Atlas é utilizada tipicamente para o armazenamento de metadados. Para além de serem armazenados os metadados, esta tecnologia fornece um conjunto de informação à componente *Profiling Tool*, nomeadamente informação sobre as estatísticas e indicadores de qualidade das colunas, de forma a associar posteriormente essa informação às estatísticas de qualidade uma tabela.

Esta tecnologia também é capaz de fornecer informação relevante sobre as tabelas Hive, nomeadamente sobre o *Lineage* das tabelas (averiguar se a tabela resulta da junção de duas ou mais tabelas ou se deriva de uma fonte de dados presente do HDFS, etc), bem como eventuais alterações realizadas na tabela *(Audits)*, como informação sobre quem criou a tabela, quem realizou a última alteração ou até mesmo quem eliminou a mesma.

No trabalho realizado levado a cabo por Maria Inês Costa – Etiquetagem e rastreio de fontes de dados num *Big Data Warehouse* (de momento não se encontra publicada), é fornecido um catálogo de dados organizacional que se encontra instanciado no Atlas. No entanto, na presente dissertação apenas é fornecida informação para determinados nós, nomeadamente os nós relacionados com a qualidade e similaridade de dados, com os *Jobs* realizados em Spark, com a informação das tabelas Hive, com as permissões presentes no Ranger e, por fim, com os *Audits* das tabelas Hive.

No que diz respeito aos componentes presentes do lado direito da arquitetura da Figura 60, estes surgem da necessidade de obter informação sobre os processos executados pela componente *Profiling Tool*, resultando no armazenamento de metadados na tecnologia Atlas sobre os *Jobs* executados em Spark e informação sobre as políticas de segurança das tabelas Hive fornecidas pela tecnologia Ranger através de uma interface REST.

Um dos desafios definidos para *Big Data* está relacionado com a privacidade e as políticas definidas para os dados, pelo que é necessário ter em conta as autorizações dadas aos mesmos (C. Costa & Santos, 2017b). Assim, a Tabela 16, de acordo com o que é necessário obter sobre uma política de dados (permissão) e com base na ferramenta Ranger, apresenta a informação fornecida e armazenada na tecnologia Atlas. É **importante** mencionar que a definição dos atributos foi realizada pelo trabalho de Maria Inês Costa – Etiquetagem e rastreio de fontes de dados num *Big Data Warehouse* (de momento não se encontra publicada) tendo a presente dissertação o objetivo de realizar a integração entre o Ranger e o Atlas de forma a preencher os atributos presentes na Tabela 16.

Tabela 16. Análise das políticas de segurança do Ranger.

Nome	Tipo de Dados	Descrição
Tipo de política	String	Tipo de política, por exemplo, "Access"
ID da política	String	Identificador da política
Estado da política	Boolean	Se a política está ativa *(enabled)* retorna "true" e "false", caso contrário. Neste último caso, a política está desativada *(disabled),* fazendo o processo inverso, ou seja, ao invés de se habilitar a política, desabilita-se a mesma
Exclusão da base de dados	Boolean	Se as bases de dados estão incluídas *(include)* retorna "false". Caso contrário, retorna "true" e, neste caso, está a excluir-se as bases de dados *(exclude)* da política
Exclusão da tabela	Boolean	Se as bases de dados estão incluídas *(include)* retorna "false". Caso contrário, retorna "true" e, neste caso, está a excluir-se as bases de dados *(exclude)* da política
Exclusão da coluna	Boolean	Se as bases de dados estão incluídas *(include)* retorna "false". Caso contrário, retorna "true" e, neste caso, está a excluir-se as bases de dados *(exclude)* da política
Descrição da política	String	Descrição e comentários adicionais à política.

Nome	Tipo de Dados	Descrição
Permissões	array<String>	Reúne o(s) grupo(s) de utilizador(es), o(s) utilizador(es) aos quais se dá permissão, assim como as permissões definidas, por exemplo: “select”, “update”, “Create”, “Drop”, “Alter”, “Index”, “Lock”, entre outras e, ainda, indica se a permissão tem associada a função de administrador *(delegateAdmin)*
Utilizador da criação	String	Utilizador que criou a política no Ranger
Utilizador da última alteração	String	Último utilizador que editou a política no Ranger
Data de criação	Date	Data de criação da política no Ranger
Data da última alteração	Date	Data de edição da política no Ranger
Bases de dados afetadas	array<String>	Bases de dados associadas à política
Tabelas afetadas	array< String >	Tabelas associadas à política
Colunas afetadas	array< String >	Colunas associadas à política

Uma vez que esta temática se depara com grandes volumes de dados, é importante registar toda a informação sobre os *Jobs* executados pela componente *Profiling Tool*, ou seja, é relevante ter conhecimento sobre quem e quando o *Job* foi executado e, por fim, ter conhecimento se o *Job* teve sucesso ou não. Como tal, a Tabela 17 apresenta a informação retirada do Spark *Jobs* através de uma interface REST, sendo armazenando posteriormente na tecnologia Atlas.

Tabela 17. Análise dos Jobs executados em Spark.

Nome	Tipo de Dados	Descrição
Data de início	Date	Data de criação (inicialização) do *job*
Data de fim	Date	Data de finalização do *job*
Duração	float	Duração do *job* em milissegundos
Utilizador	String	Utilizador do Spark

Nome	Tipo de Dados	Descrição
Estado do *job*	Boolean	Corresponde ao estado do *job*, se falhou indica "false" ou se foi bem-sucedido retorna "true"

Por fim, a componente de visualização não foi concebida, devido a limitações de tempo; no entanto, é possível a integração com qualquer ferramenta de visualização devido à disponibilização de uma interface REST. O principal objetivo desta componente passa por permitir ao utilizador uma forma "amigável" de executar e visualizar os processos de qualidade de dados com as novas fontes de dados e, numa perspetiva de integração de dados, permitir a alteração do threshold presente no algoritmo de análise da similaridade dos *headers*. De frisar que a tecnologia Atlas, para além do armazenamento, possibilita a visualização dos metadados em formato web.

5.3. Implementação da Arquitetura

Posteriormente à apresentação da arquitetura de Governança de Dados, é importante que se proceda à sua implementação e, como tal, é proposto um cenário no qual surge uma nova fonte de dados em formato *batch*, nomeadamente o conjunto de dados *"Promotion"*. Inicialmente, uma vez que se trata de uma nova fonte de dados, é importante averiguar a sua qualidade. Assim, os dados passam pelo processo de análise da qualidade de dados através da ferramenta concebida em Spark, no qual é produzido um conjunto de métricas e indicadores presentes na Tabela 13 e Tabela 14, relativamente a cada coluna presente no conjunto de dados. Essa informação é armazenada no Atlas, sendo possível ser consultada através da sua plataforma de visualização. A Figura 61 apresenta a informação consultada no Atlas. Apenas é apresentado o exemplo de uma coluna do conjunto de dados (coluna *"ss_promo_sk"* do conjunto de dados *"Store_Sales")*, devido à limitação de páginas, mas a informação é gerada para todas as colunas.

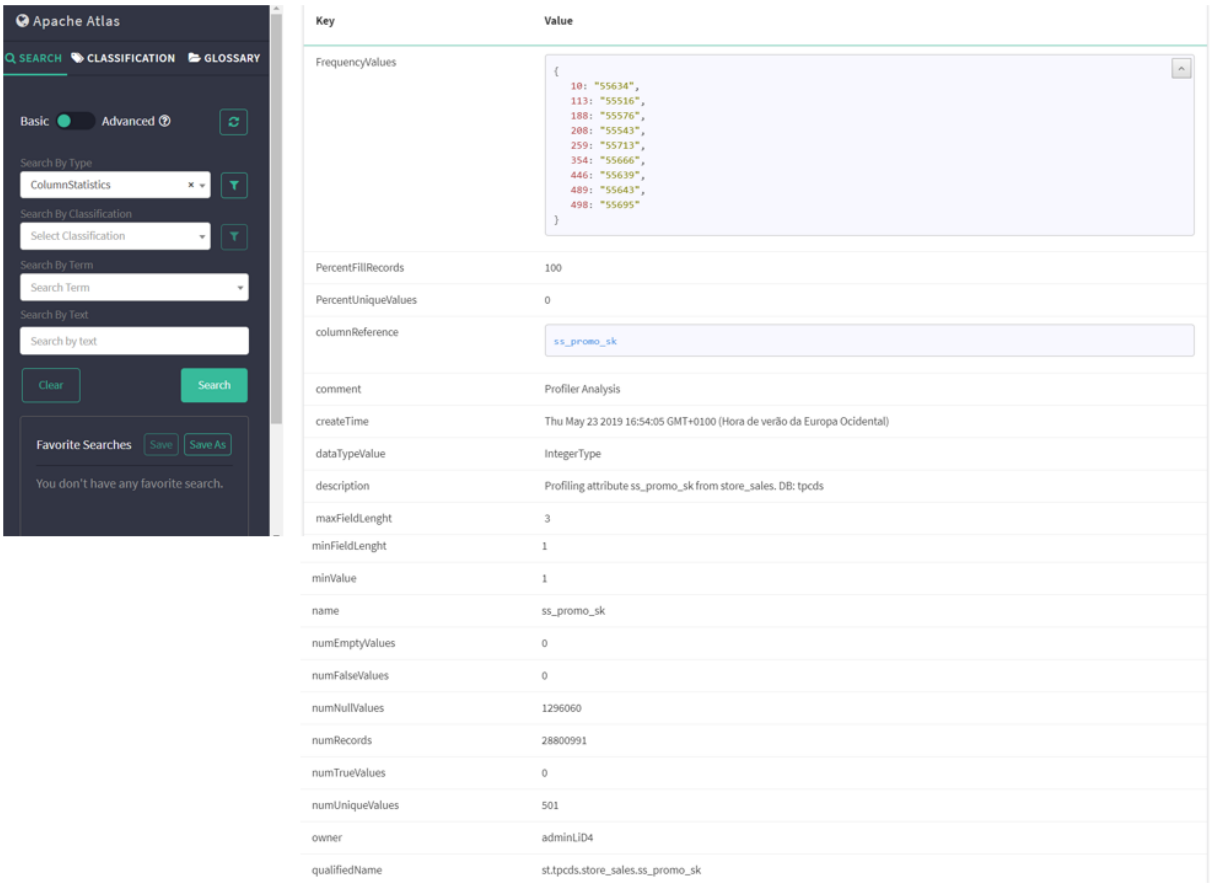

Figura 61. Análise da qualidade de dados das colunas do conjunto de dados "Promotion".

Nesta fase de análise de dados, é também criado um conjunto de indicadores mais globais da tabela, presentes na Figura 62, frisando um conjunto de pormenores, tais como a presença de todas as colunas associadas à tabela *"Promotion"*, permitindo assim uma maior facilidade de navegação entre os resultados de qualidade associados a cada coluna. Anteriormente, foi apresentado apenas um exemplo detalhado da informação sobre a qualidade de dados de uma determinada coluna (coluna *"ss_promo_sk"* do conjunto de dados *"Store_Sales")*, mas é possível observar na Figura 62 que esse processo é realizado para todas as colunas que, no contexto da Figura 62 faz referência à tabela *"Promotion"*.

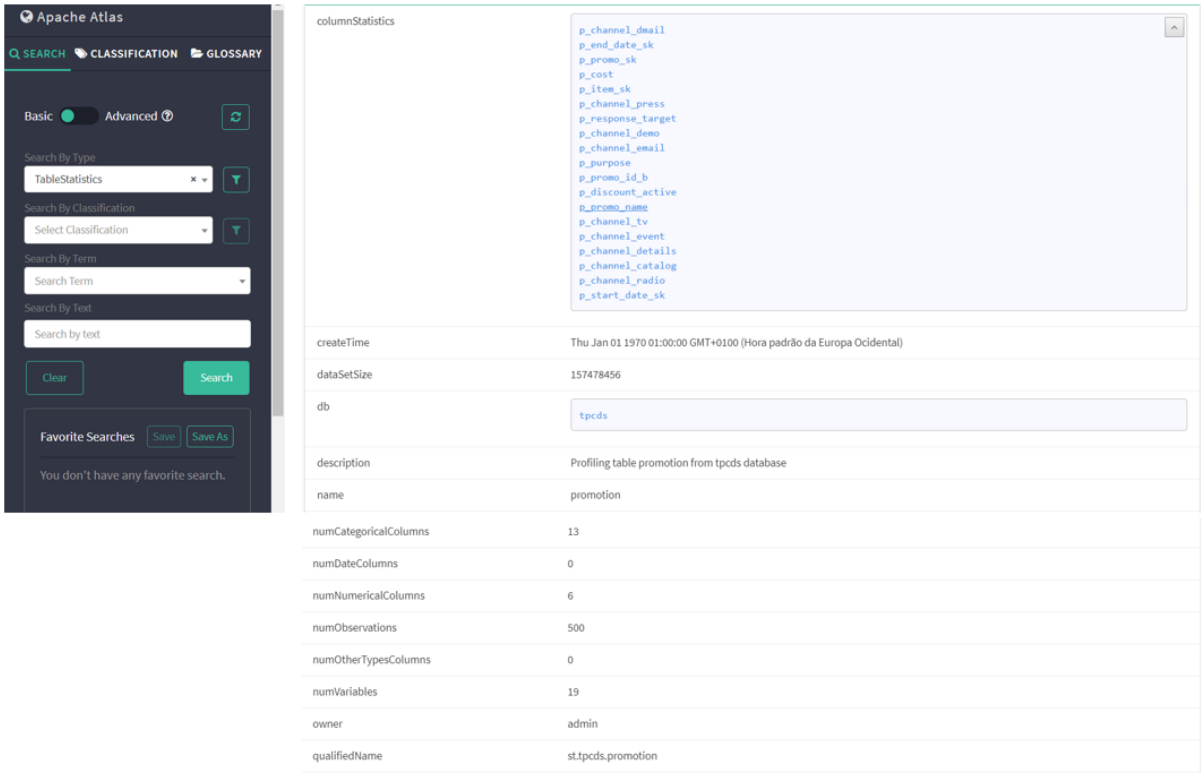

Figura 62. Análise da qualidade de dados do conjunto de dados "Promotion".

Paralelamente ao processo de qualidade de dados, a ferramenta de *Data Profiling* produz um conjunto de informação adicional de forma a auxiliar o utilizador no *Lineage* da informação. Na Figura 63, é possível ver que a tabela *"Promotion"* estava situada no HDFS sofrendo posteriormente dois processos: um processo de carregamento e um processo de análise de qualidade de dados, resultando no componente final assinalado com um círculo a vermelho na Figura 63.

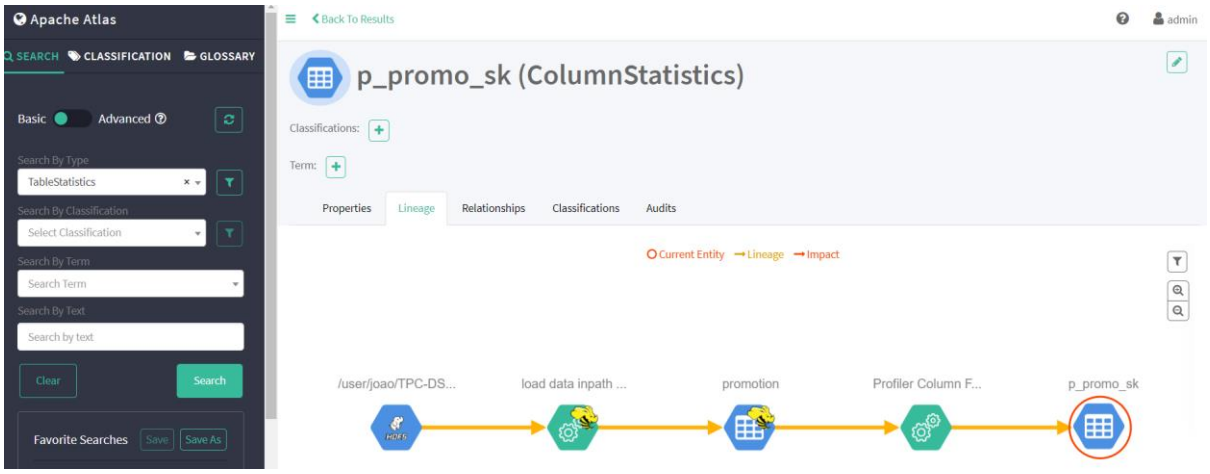

Figura 63. Lineage da informação da coluna "p_promo_sk".

Para além da informação fornecida pelo *Lineage,* o Atlas é capaz de auxiliar o utilizador com informação relativa às alterações efetuadas ao longo do ciclo de vida de um conjunto de dados *(Audit Management)*, a título de exemplo, é capaz de fornecer informação sobre quando foi criada/atualizada a tabela e qual o utilizador que realizou essa ação. A Figura 64 apresenta um exemplo associado à coluna *"p_promo_sk"*, na qual é possível observar que as estatísticas foram atualizadas 3 vezes pelo utilizador "Admin". Num contexto organizacional, esta informação é importante, na medida em que todos os elementos que manipulam dados têm uma visão holística de cada conjunto de dados, assim como as respetivas alterações.

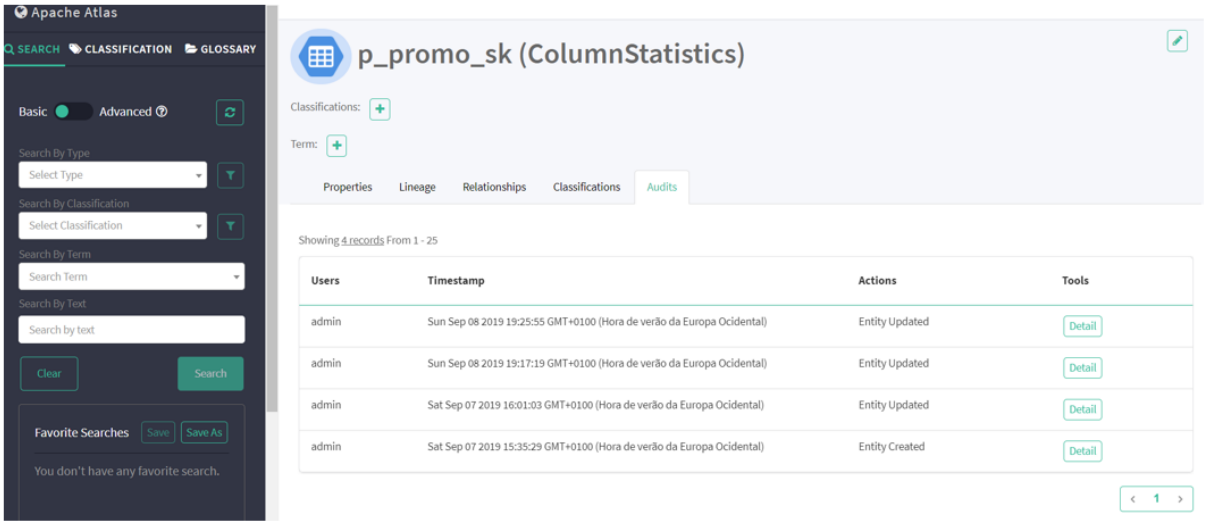

Figura 64. Audits associados à coluna "p_promo_sk".

No que diz respeito à integração de dados, é demonstrado de seguida um cenário de integração de dados entre o conjunto de dados *"Promotion",* que representa um novo conjunto de dados, e a tabela *"Store_Sales",* que se encontra presente no BDW. Perante o processo de integração, é necessário o

utilizador definir um conjunto de parâmetros associados à execução do algoritmo de similaridade dos *headers* e do conteúdo dos dados. Assim, foram executados os processos de similaridade com um **threshold de 0.5** e **sem** recurso ao dicionário WordNet (parâmetros utilizados apenas para efeitos de demonstração). Perante tais parâmetros, a Figura 65 representa os 9 pares de atributos com uma possível integração.

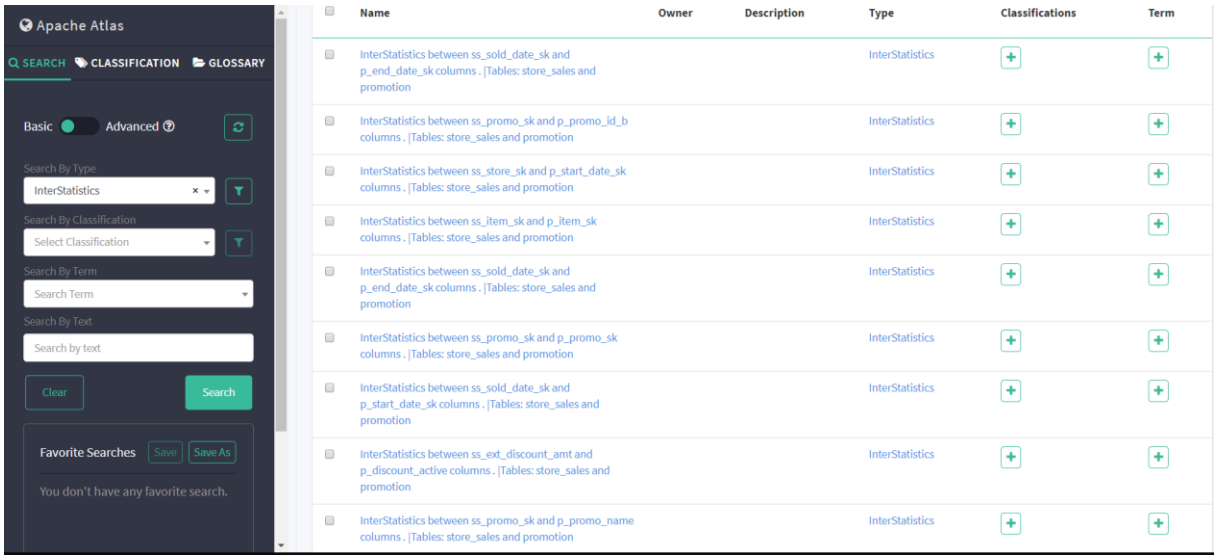

Figura 65. Número de pares de atributos com possível integração entre "Promotion" e "Store_Sales".

Perante os 9 pares de atributos, é selecionado um exemplo que se encontra presente na Figura 66 que, apesar da similaridade dos *headers* apresentar um valor de 0.6495 entre *"ss_sold_date_sk"* e *"p_end_date_sk"*, a similaridade do conteúdo de dados apenas apresenta um valor de **15.82%**, significando que se se integrasse a tabela *"Promotion"* e "*Store_Sales*" através deste par de atributos apenas **15.82%** dos dados da tabela *"Promotion"* iriam ser correspondidos.

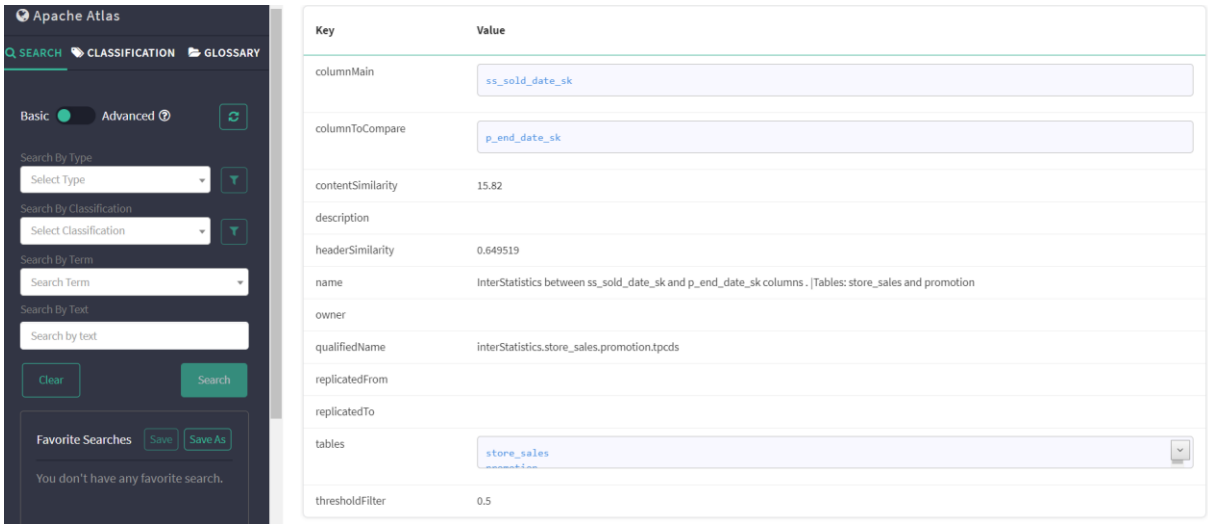

Figura 66. Resultados do processo de integração entre "Promotion" e "Store_Sales".

Estes mecanismos de armazenamento dos metadados oferecem vantagens, dado que, como esta informação se encontra armazenada, não existe a necessidade de se executar novamente os processos de similaridade, sendo necessário apenas consultar os mesmos na ferramenta Atlas. No entanto, o utilizador cada vez mais deseja que os seus processos sejam otimizados e, nessa vertente, ter de analisar todos os 9 pares de atributos (que noutras circunstâncias existe a possibilidade deste número ser mais elevado) torna o seu processo ineficiente.

De forma a corresponder a esse desafio, o Atlas fornece um mecanismo de pesquisa *(quering)* que visa a otimização desses processos, ou seja, permite ao utilizador filtrar os dados consoante as suas necessidades. A título de exemplo, a Figura 67 apresenta o resultado de uma *query* realizada pelo utilizador cuja necessidade é selecionar os pares de atributos com similaridade do conteúdo de dados seja superior a 85% (valor compreendido entre 0 e 100). Na Figura 63, é possível observar apenas um par de atributos *("p_promo_sk"* e *"ss_promo_sk").* No entanto o resultado apresenta dois pares de atributos, nomeadamente o par de atributos ***"ss_promo_sk"*** e ***"p_promo_sk"*** (presente na Figura 63) e o par de atributos ***"ss_item_sk"*** e ***"p_item_sk",*** em que ambos apresentam uma similaridade no conteúdo dos dados de 100%. O formato do resultado apresentado na Figura 63 (JSON) possibilita a comunicação com outras plataformas de visualização, permitindo ao utilizador visualizar informação relevante e pertinente para a tomada de decisão.

```
{
    requestId: "pool-2-thread-6 - b018b5e6-e24a-41e9-8813-116698b3b8f9",
    queryType: "dsl",
    query: "InterStatistics where contentSimilarity > 85",
    dataType: null,
    count: 2,
  - results: [
      - {
            owner: null,
          - $systemAttributes$: {
                modifiedTime: "Tue Sep 10 23:26:03 WEST 2019",
                createdBy: "admin",
                createdTime: "Tue Sep 10 23:26:03 WEST 2019",
                modifiedBy: "admin"
            },
            replicatedTo: null,
            replicatedFrom: null,
            qualifiedName: "interStatistics.ss_promo_sk.p_promo_sk.store_sales.promotion.tpcds",
            description: null,
            $typeName$: "InterStatistics",
            thresholdFilter: 0.5,
            headerSimilarity: 0.84327406,
          - columnMain: {
                id: "19afc63e-546c-46ab-a98e-24503a9ecf02",
                $typeName$: "hive_column",
                state: "ACTIVE",
                version: 0
            },
          - tables: [
              - {
                    id: "8b08127f-7ced-43af-b107-92f5cfcd3aa3",
                    $typeName$: "hive_table",
                    state: "ACTIVE",
                    version: 0
                },
              - {
                    id: "3814a1c4-bc3e-489b-9f11-ae042204603a",
                    $typeName$: "hive_table",
                    state: "ACTIVE",
                    version: 0
                }
            ],
          - $id$: {
                id: "eaf3a682-e99c-42ae-b3b5-7a240fb4c2ee",
                $typeName$: "InterStatistics",
                state: "ACTIVE",
                version: 0
            },
          - columnToCompare: {
                id: "d941b4da-4e52-429a-8c75-1371ab08e75c",
                $typeName$: "hive_column",
                state: "ACTIVE",
                version: 0
            },
            name: "InterStatistics between ss_promo_sk and p_promo_sk columns . |Tables: store_sales and promotion",
            contentSimilarity: 100
        },
```


Figura 67. Apresentação dos resultados em formato JSON.

A componente da segurança de dados é cada vez mais relevante para que as organizações levam em conta, sendo importante ter informação detalhada dos grupos de utilizadores com acesso aos dados e que tipo de alterações são capazes de executar *(select, update, delete, insert,* entre outros). A Figura 68 apresenta a lista de políticas presentes no *cluster* sendo possível observar o seu nome, o seu "owner" e a sua descrição.

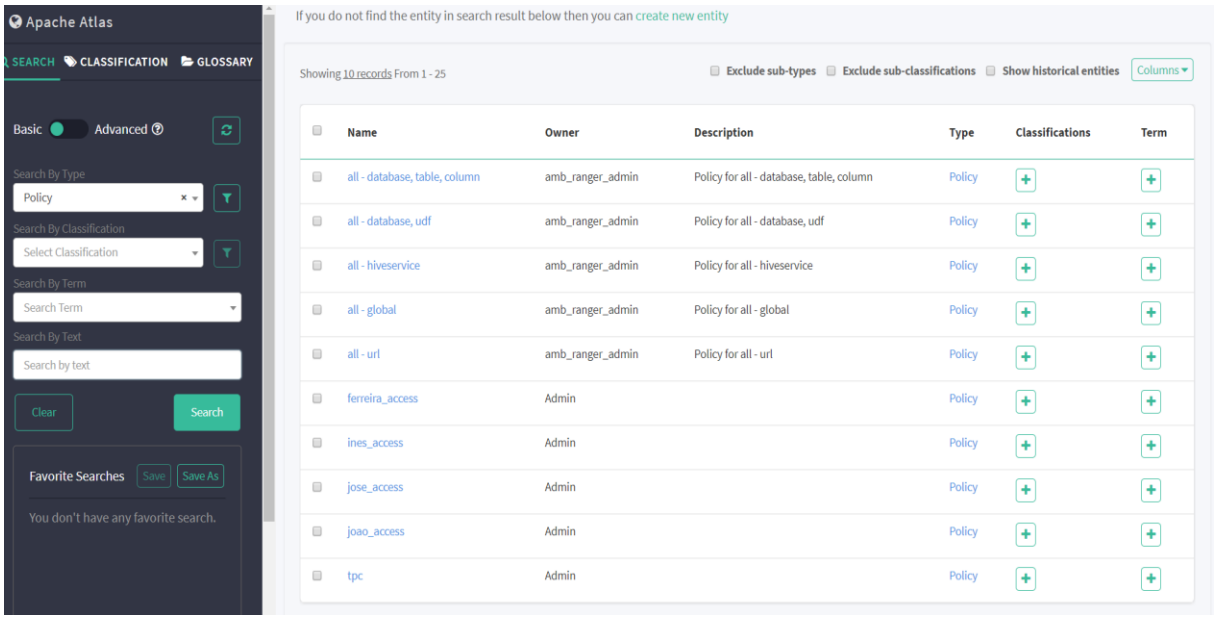

Figura 68. Lista das políticas de segurança do Ranger presentes no cluster.

Através da Figura 69, é possível observar a informação armazenada em relação à política de dados do TPC-DS. É possível observar que esta política se aplica às bases de dados onde se encontram os dados do TPC-DS nos seus diferentes FE. Uma vez que estas bases de dados contêm um alargado conjunto de tabelas e colunas, sempre que a política se aplicasse a todas as colunas/tabelas seria armazenado apenas o símbolo *****, como é possível observar na Figura 69.

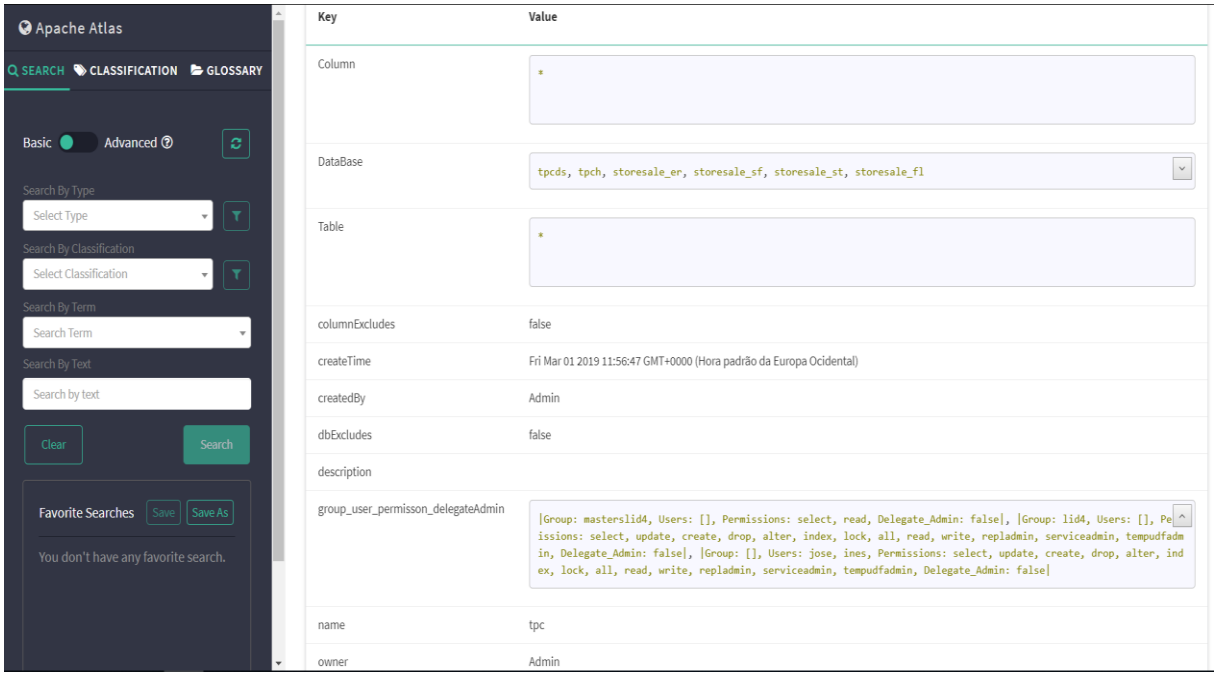

Figura 69. Exemplo da informação armazenada de uma política de segurança.

Ao longo da implementação da arquitetura, são executados um alargado número de ***processos (Processes)***, nomeadamente para executar os processos de qualidade e similaridade de dados, acrescentando ainda os processos de armazenamento das políticas de segurança provenientes do Ranger. Assim, é importante armazenar informação sobre quem executou o *Job*, quando executou e, o mais importante, se esse *Job,* foi executado com sucesso. A informação dos processos de qualidade está associada a um *Job,* tal como é possível visualizar na Figura 70, que, neste caso, se trata do processo de qualidade *(Data Profiling)* associado à tabela *"Promotion"*.

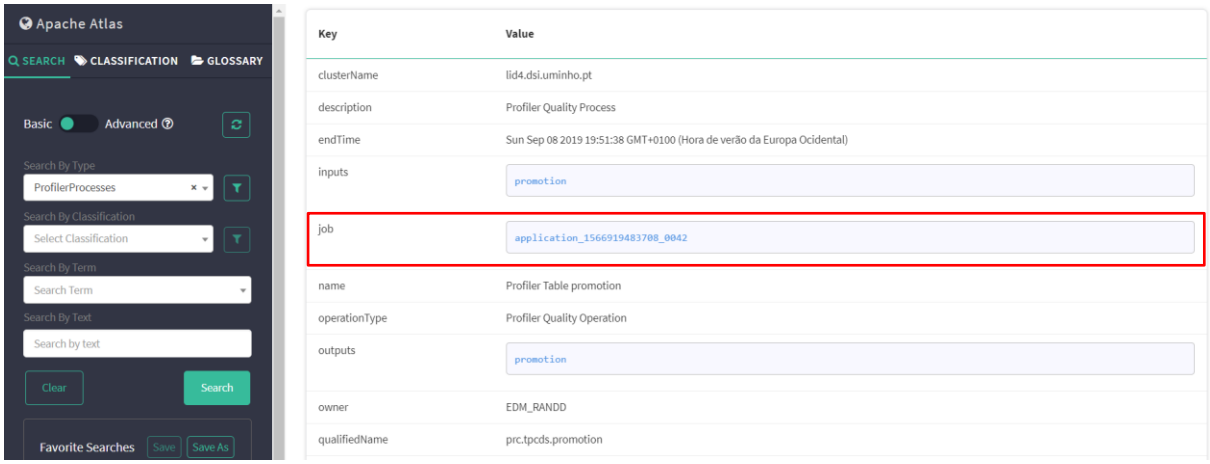

Figura 70. Informação sobre o processo de Data Profiling.

O componente *Job* presente na Figura 71 está associado ao processo da Figura 70 e permite visualizar a informação sobre a duração do *Job,* sobre o seu estado e sobre o utilizador Spark que executou o mesmo.

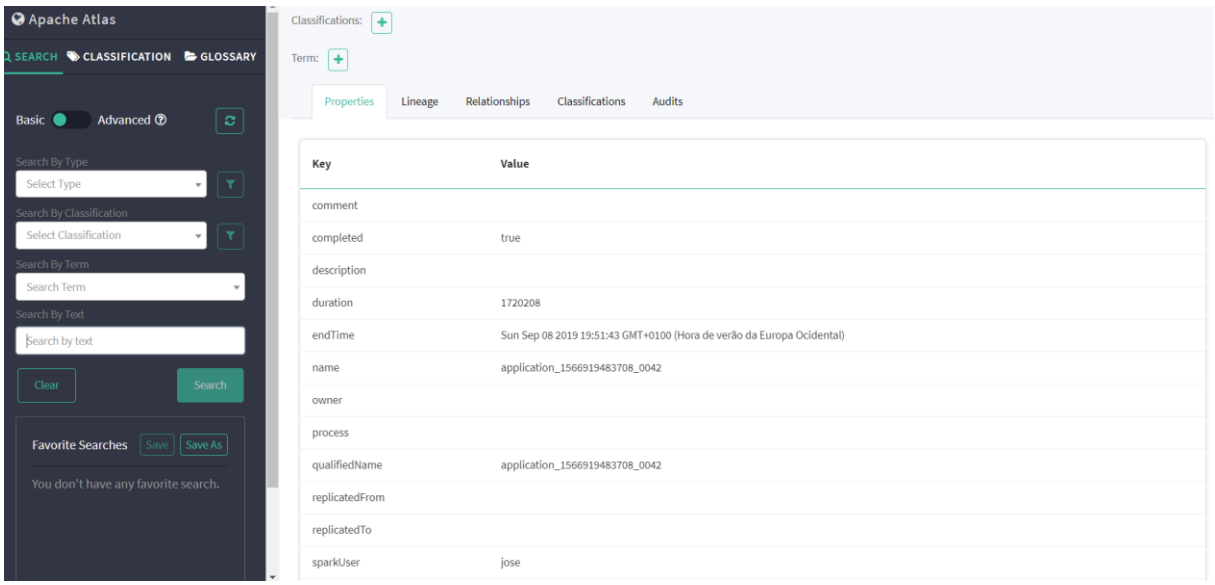

Figura 71. Informação armazenada no Atlas sobre um Spark Job.

Com este exemplo, é possível dar a conhecer as potencialidades da ferramenta de *Data Profiling* integrada com as ferramentas Atlas e Ranger em contextos de *Big Data*. Apesar de se tratar de um trabalho que ainda está em curso, nomeadamente na conceção da componente de visualização (componente cujo objetivo permitirá a execução de todos os processos mencionados anteriormente), o mesmo revela-se importante para se compreender a globalidade e uniformização de um BDW numa organização. É **importante** mencionar que foram realizadas alterações na ferramenta Atlas de modo a que fosse possível observar a informação apresentada ao longo desta secção. Essa informação necessitou de ser customizada um vez que não existia na ferramenta.

Por fim, destacam-se as principais vantagens que a implementação desta arquitetura nos oferece:

- **Visão holística de toda a organização:** é possível verificar a informação de toda a organização, juntamente com o trabalho de por Maria Inês Costa – Etiquetagem e rastreio de fontes de dados num *Big Data Warehouse* (de momento não se encontra publicada), nomeadamente informação sobre tabelas e colunas de dados presentes no BDW (informação sobre quem e quando foram criadas, atualizadas e eliminadas as tabelas), informação sobre os novos conjuntos de dados, informação global sobre os *Jobs* executados, informação sobre as políticas de segurança da organização (no acesso aos dados e aos sistemas inerentes à mesma), informação sobre os indicadores e KPI's de uma organização e, por fim, informação sobre os seus processos de negócio e que atividades estes desempenham;
- **Otimização das cargas de trabalho:** no contexto da análise de qualidade e similaridade de dados, apenas é necessário executar **uma** vez o *Job* da qualidade e da similaridade de dados sempre que chega um novo conjunto de dados já que essa informação gerada será armazenada e, mais tarde, consultada no Atlas. Em processos tipicamente "normais", é necessário verificar se um novo conjunto de dados, tem a possibilidade de se integrar com alguma tabela no BDW e, na eventualidade de essa integração não ser possível, essa informação não fica armazenada em nenhum lugar dentro da organização existindo a possibilidade, em contextos de grandes organizações, de se repetir o mesmo processo;
- **Otimização de processos:** com a presente implementação, apenas é necessária uma ferramenta para a visualização (e mais tarde execução dos *Jobs)* da informação

de toda a organização (sem necessidade de recorrer à ferramenta Ranger para visualizar as permissões).

6. CONCLUSÕES

Este documento, tal como era objetivo começa por expor o enquadramento concetual sobre os principais conceitos associados à temática desta dissertação, nomeadamente: o Big Data, os Data Warehouses, Governança de Dados e *Data Profiling* em *Big Data Warehouses*. Para além dos temas mencionados, o enquadramento concetual aborda os trabalhos que remetem para desafios semelhantes terminando com um mapa concetual cujo objetivo é estruturar os principais conceitos abordados.

Pelo intermédio do levantamento do "estado da arte" associado a *Big Data, Big Data Warehouses* e Governança de Dados, foi possível perceber que se trata de um tema atual com novos desafios na extração, processamento e armazenamento de dados. No entanto, a informação encontrada sobre a temática de *Big Data* e *Big Data Warehouses* é superior face à informação disponibilizada sobre Governança de Dados*,* que é expectável perante a atualidade do tema, mostrando a relevância nesta área.

Posteriormente, uma revisão de literatura extensa e importante conduziu ao enquadramento tecnológico, no qual foi possível conhecer um conjunto de ferramentas que suportam fenómenos *Big Data*, nomeadamente o Spark, o Hive, Ranger e o Atlas, este último suportado pela base de dados de grafos Janusgraph. Estas ferramentas representam a base desta dissertação, constituindo um conjunto de soluções importantes para o desenvolvimento de uma arquitetura de Governança de Dados.

Terminado o enquadramento concetual e tecnológico, foi apresentada a fase experimental desta dissertação, que se dividiu em dois tópicos principais: Ambientes de Testes e Governança de Dados em *Big Data Warehouses*.

No primeiro tópico, começou por se especificar a infraestrutura de testes utilizada para a realização dos testes, assim como os conjuntos de dados utilizados. Posteriormente, foram apresentadas todas as medidas de similaridade propostas para a avaliação da similaridade de duas diferentes dimensões de análise: *headers* e conteúdo de dados. De seguida, foram apresentados os cenários testes para a avaliação do desempenho das dimensões, juntamente com os principais requisitos para os executar.

Os resultados obtidos no primeiro tópico da fase experimental permitiram estudar o impacto no desempenho e no valor de similaridade de cada medida na análise dos *headers* e do conteúdo de dados. Na dimensão dos *headers,* foi possível observar que a medida de *Jaccard*, para cenários de atributos com relacionamento, apresentava valores reduzidos de similaridade quando comparadas a outras medidas, devido à sua fórmula de cálculo. Verificou-se também que as medidas de *Cosine*, *Jaccard* e

Jaro-Winkler são capazes de reconhecer contextos em que os pares de atributos/entidades não são similares, ao invés da medida de *Levenshtein* que apresentou valores de similaridade considerados elevados para o cenário/objetivo em questão. Conclui-se, nesta secção, que o método de cálculo das medidas de similaridade é uma particularidade a ter em conta, assim como a sua complexidade que, na análise da dimensão dos *headers,* não é um fator crucial devido aos tempos de processamento se apresentarem sempre reduzidos. Porém, no contexto da dimensão de análise do conteúdo de dados foi um fator a considerar. Na avaliação da dimensão dos *headers,* considera-se que a medida de *Cosine* apresenta uma maior credibilidade, tendo em conta não só os resultados obtidos nos diferentes cenários apresentados, mas também o seu cálculo da similaridade. A medida de *Jaro-Winkler* apresenta um elevado número de pares de atributos com valores de similaridade elevada e, em contextos *Big Data*, é mais plausível tomar decisões sempre que se está perante dois ou três possíveis pares de atributos similaridades do que estar perante dez ou vinte pares de atributos. No que à dimensão do conteúdo de dados diz respeito, verificou-se que ambas as medidas apresentam valores de similaridade relevantes, sendo que ambas são capazes de identificar os pares de atributos similares. No entanto, a medida de similaridade com base na distribuição de valores apresenta desafios ao nível do tempo de processamento, devido ao cálculo da distribuição de valores por cada valor único. A medida com melhor desempenho diz respeito à medida de similaridade com base na intersecção e união e valores, uma vez que o seu tempo de processamento é inferior à medida com base na distribuição de valores, acrescentando ainda que é uma medida que acrescenta valor à organização por se apresentar como uma medida bidirecional.

Posteriormente à realização dos testes, as medidas de similaridade selecionadas foram testadas num caso real no âmbito da genética resultando esse trabalho num grafo de similaridade entre os conjuntos de dados. A síntese dos resultados obtidos culminou num algoritmo de similaridade que analisa ambas as dimensões no qual foram alocadas aquelas que foram consideradas as melhores medidas de similaridade para os *headers* e para a análise do conteúdo dos dados. Este algoritmo considera, ainda, a análise semântica dos dados para os casos em que o valor da similaridade seja inferior ao *threshold* definido pelo utilizador.

Por fim, no segundo tópico da fase experimental da presente dissertação, foi apresentada uma arquitetura de Governança de Dados. Neste capítulo, começou por mencionar-se os dados utilizados para a implementação da arquitetura, seguindo-se do seu desenho na qual integraram a ferramenta de *Data Profiling* desenvolvida pelo autor da presente dissertação, o Ranger, o Spark *Jobs* e o Atlas. Após a explicação de todos os componentes da arquitetura, foi demonstrada a implementação dos mesmos

para os principais desafios apresentados na dissertação. Ao longo deste capítulo, são resumidos os principais desenvolvimentos desta dissertação, começando com a apresentação dos resultados obtidos, com a execução da mesma, prosseguindo-se com as dificuldades e limitações sentidas e, por fim, a investigação que poderá ser feita futuramente neste âmbito, de forma a ver respondidas algumas questões que foram surgindo neste trabalho, completando o trabalho desenvolvido nesta temática.

6.1. Resultados Obtidos

Esta dissertação tem como finalidade propor uma ferramenta de *Data Profiling* para a Governança de Dados de um *Big Data Warehouse*, de modo a permitir o aumento da eficiência no tratamento das novas fontes de dados. Desta forma, pode dizer-se que os objetivos propostos para este trabalho foram atingidos pois, analisando esses mesmos objetivos e comparando com o resultado final, é possível concluir-se que:

- Foi sistematizado o estado da arte no que diz respeito às abordagens de implementação de ferramentas de *Data Profiling,* Governança de dados e, por fim, algoritmos de similaridade. Para além dessas abordagens, foi realizado o levantamento do estado da arte referente às tendências *Big Data* e Armazenamento de Dados;
- Foi realizada a contextualização das tecnologias que permitam a implementação da ferramenta proposta. Neste contexto, foi sempre objetivo utilizar uma base de dados baseada em grafos que inicialmente era a Neo4J, mas, posteriormente, perante as vantagens oferecidas pelo Atlas, foi utlizada a Janusgraph (base de dados de grafos que suporta o Atlas);
- Foram identificadas e desenvolvidas um conjunto de rotinas para a análise da distribuição e qualidade dos atributos num conjunto de dados, na medida em que essas rotinas foram identificadas no levantamento do estado da arte e, posteriormente, desenvolvidas/aplicadas na ferramenta da *Data Profiling*;
- Foi avaliado o desempenho das diferentes medidas de similaridade, com vista à integração na ferramenta de *Data Profiling* destacando a medida de similaridade de *Cosine* para a avaliação dos *headers* e a medida concebidas pelo para a avaliação do conteúdo dos dados. Ainda em relação aos *headers*, foi adicionada a possibilidade de uma análise dos mesmos, recorrendo à semântica, mas em contrapartida esse processo é mais demorado*;*

- Foi identificado e desenvolvido um conjunto de rotinas com o objetivo de analisar a similaridade entre novas fontes de dados e os dados já existentes num *Big Data Warehouse,* resultando assim um grafo de similaridade que facilita a sua integração;
- Foi proposta e implementada uma arquitetura de Governança de Dados que permita armazenar e gerir toda a informação preveniente dos objetivos anteriores.

Em conclusão, pode afirmar-se que a dissertação traz um contributo importante para a área de Governança de Dados, conceito tão atual, mas muito imaturo na grande parte das organizações, nomeadamente quando se emprega no contexto de *Big Data Warehousing*.

6.2. Dificuldades e Limitações

Ao longo do desenvolvimento desta dissertação, a maior dificuldade sentida esteve sempre associada à escolha da medida de similaridade para ser utilizada ao nível dos *headers* e ao nível do conteúdo de dados. Como tal, foi necessário um grande espírito crítico até se chegar ao resultado final, ou seja, culminou no algoritmo apresentado que avaliasse as duas dimensões.

Inicialmente, a presente dissertação estava focada no impacto nos modelos de dados cada vez que se integrava uma nova fonte de dados, sendo que a primeira versão do protocolo de testes incidia na avaliação do impacto ao nível da integração de dados entre um modelo em estrela, floco de neve e desnormalizado. Este foco inicial foi abstraído e não trouxe resultados concretos no final da dissertação, no entanto, permitiu que fossem aperfeiçoadas as vertentes de modelação de dados para trabalho futuro.

Por outro lado, a ferramenta de *Data Profiling* concebida foi programada em Spark. No entanto, o *background* do autor nesta temática não era elevado, não se encontrando ambientado com esta linguagem de programação e, como tal, foi necessário investir nesta competência técnica através de tutoriais e de pequenos casos de demonstração.

Como foi mencionado anteriormente, o Atlas é uma ferramenta fundamental neste trabalho e foi, de igual modo, uma ferramenta que gerou algumas dificuldades, nomeadamente no momento da sua exploração, pois foi trabalhoso compreender o seu modo de funcionamento, ou seja, a forma como armazena e modela a informação, uma vez mais devido à falta de informação/documentação. O processo de armazenamento de metadados não foi uma dificuldade que comprometesse o trabalho, mas o processo de realização do *lineage* através da ferramenta de *Data Profiling* é um processo que causou um elevado investimento, tanto a nível lógico como a nível técnico. O mesmo se aplica na organização da informação no Atlas proveniente da ferramenta de *Data Profiling*, do Ranger e do Spark Jobs.

Ainda relativamente ao Atlas, foi constatado que, quando se inserem novas linhas a tabelas, este não considera essa ação, ou seja, não está a considerar que houve um "*update*" à entidade, tal como guarda os detalhes quando se cria uma nova tabela, pelo que uma provável melhoria será estender os tipos de operações dos processos e, assim, será possível guardar outros tipos de "*updates*".

Tendo em conta que esta dissertação produziu um grande número de resultados, fruto da execução dos diferentes cenários de teste com um número de medidas de similaridade considerável cada um, a organização destes revelou-se desafiante.

Por último, a criação de várias tabelas com diferentes volumes de dados para que se pudessem efetuar os vários cenários, implicou grandes tempos que necessitaram de ser geridos corretamente para uma apresentação de resultados rigorosos.

6.3. Trabalho Futuro

Apesar de se terem atingido todos os objetivos propostos, existem vários aspetos que podem ser estudados e melhorados no futuro. No que diz respeito à ferramenta de *Data Profiling,* um dos possíveis trabalhos futuros será avaliar a ferramenta ao nível do processamento, ou seja, apesar de integrar os algoritmos que se consideram adequados é necessário averiguar outros contextos, nomeadamente o tempo de processamento no armazenamento de informação no Atlas quando comparado a outras bases de dados. Ainda nesta ferramenta, é importante que, no futuro, o mecanismo de análise da qualidade de dados seja implementado de maneira diferente, ou seja, neste momento temos a ferramenta de *Data Profiling* a analisar a qualidade dos dados e a armazenar informação sobre a mesma, mas pode colocar-se algumas questões tais como "Apesar da informação da qualidade dos dados se encontrar armazenada os dados apresentam ou não a qualidade adequada/esperada?". Com isto pretende afirmar-se que o utilizador deve introduzir os padrões de qualidade esperados e, se o conjunto de dados não cumprir os mesmos, o utilizador seria notificado. A título de exemplo, o utilizador pretende que as colunas de análise apresentem pelo menos 45% dos valores preenchidos e, perante isso, sempre que surgir uma nova coluna que não cumpra esse requisito será notificado, tendo que, posteriormente, tomar decisões sobre a mesma. A ferramenta de visualização de dados também será um trabalho a ter em conta, no futuro, com o objetivo de implementar todos os componentes propostos na arquitetura de Governança de Dados e que permitirá uma análise mais cuidada da informação presente no Atlas permitindo também a execução dos vários processos *(Jobs)* que cubram toda a organização

Por fim, será testada futuramente que incide na integração de dados através da classificação de dados, ou seja, com o suporte do Atlas em classificar as novas fontes de dados será possível não ter de comparar os novos conjuntos de dados com todas as tabelas do BDW mas apenas com aquelas que apresentem a mesma classificação *(por exemplo: Vendas, Compras, Clientes, entre outros)*, tornando assim o seu processo eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

Abdellaoui, S., & Nader, F. (2015). Semantic data warehouse at the heart of competitive intelligence systems: Design approach. *2015 6th International Conference on Information Systems and Economic Intelligence (SIIE)*, 141–145. https://doi.org/10.1109/ISEI.2015.7358736

Abedjan, Z., Golab, L., & Naumann, F. (2015). Profiling relational data: A survey. *Springer Berlin Heidelberg*. Retrieved from http://dspace.mit.edu/handle/1721.1/106176

Abedjan, Z., Golab, L., & Naumann, F. (2017). Data Profiling: A Tutorial. *Proceedings of the 2017 ACM International Conference on Management of Data*, 1747–1751. https://doi.org/10.1145/3035918.3054772

Acharjya, D. P., & P, K. A. (2016). A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *7*(2). https://doi.org/10.14569/IJACSA.2016.070267

Agrawal, D., Bernstein, P., Bertino, E., Davidson, S., Dayal, U., Franklin, M., ... Widom, J. (2011). Challenges and Opportunities with Big Data 2011-1. *Cyber Center Technical Reports*. Retrieved from https://docs.lib.purdue.edu/cctech/1

Alserafi, A., Abelló, A., Romero, O., & Calders, T. (2016). Towards Information Profiling: Data Lake Content Metadata Management. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 178–185. https://doi.org/10.1109/ICDMW.2016.0033

Apache. (2018). Apache Hadoop. Retrieved 23 December 2018, from http://hadoop.apache.org/

Arasu, A., Ganti, V., & Kaushik, R. (2006). Efficient Exact Set-similarity Joins. *Proceedings of the 32Nd International Conference on Very Large Data Bases*, 918–929. Retrieved from http://dl.acm.org/citation.cfm?id=1182635.1164206

Ardagna, D., Cappiello, C., Samá, W., & Vitali, M. (2018). Context-aware data quality assessment for big data. *Future Generation Computer Systems*, *89*, 548–562. https://doi.org/10.1016/j.future.2018.07.014

Atlas. (2018). Apache Atlas. Retrieved 18 December 2018, from https://atlas.apache.org/

Bach, M., & Werner, A. (2014). Standardization of NoSQL Database Languages. *Communications in Computer and Information Science*, *424*, 50–60. https://doi.org/10.1007/978-3-319-06932-6_6

Bakshi, K. (2012). Considerations for big data: Architecture and approach. *2012 IEEE Aerospace Conference*, 1–7. https://doi.org/10.1109/AERO.2012.6187357

Banek, M., Vrdoljak, B., & Tjoa, A. M. (2007). Using Ontologies for Measuring Semantic Similarity in Data Warehouse Schema Matching Process. *2007 9th International Conference on Telecommunications*, 227–234. https://doi.org/10.1109/CONTEL.2007.381876

Bao, X., & DAi, S. (2016, April). Large-Scale Text Similarity Computing with Spark. Retrieved 9 July 2019, from https://www.researchgate.net/publication/302922804_Large-Scale_Text_Similarity_Computing_with_Spark

Baru, C., Bhandarkar, M., Nambiar, R., Poess, M., & Rabl, T. (2013). Benchmarking Big Data Systems and the BigData Top100 List. *Big Data*, *1*(1), 60–64. https://doi.org/10.1089/big.2013.1509

Baton, J., & Bruggen, R. V. (2017). *Learning Neo4j: Effective data modeling, performance tuning and data visualization techniques in Neo4j* (2nd Revised edition). Birmingham Mumbai: Packt Publishing.

Bayardo, R. J., Ma, Y., & Srikant, R. (2007). Scaling Up All Pairs Similarity Search. *Proceedings of the 16th International Conference on World Wide Web*, 131–140. https://doi.org/10.1145/1242572.1242591

Bellahsene, Z., Bonifati, A., & Rahm, E. (Eds.). (2011). *Schema Matching and Mapping*. Retrieved from https://www.springer.com/us/book/9783642165177

Bernstein, P. A., Madhavan, J., & Rahm, E. (2011). Generic Schema Matching, Ten Years Later. *PVLDB*, *4*(11), 695–701.

Berti-Équille, L., & Borge-Holthoefer, J. (2015). *Veracity of Data: From Truth Discovery Computation Algorithms to Models of Misinformation Dynamics*. San Rafael, California: Morgan & Claypool Publishers.

Bonnet, L., Laurent, A., Sala, M., Laurent, B., & Sicard, N. (2011). Reduce, You Say: What NoSQL Can Do for Data Aggregation and BI in Large Repositories. *2011 22nd International Workshop on Database and Expert Systems Applications*, 483–488. https://doi.org/10.1109/DEXA.2011.71

Brewer, E. (2012). CAP twelve years later: How the 'rules' have changed. *Computer*, *45*(2), 23–29. https://doi.org/10.1109/MC.2012.37

Capriolo, E., Wampler, D., & Rutherglen, J. (2012). *Programming Hive* (1st ed.). O'Reilly Media.

Cassavia, N., Dicosta, P., Masciari, E., & Saccà, D. (2014). Data Preparation for Tourist Data Big Data Warehousing. *Proceedings of 3rd International Conference on Data Management Technologies and Applications*, 419–426. https://doi.org/10.5220/0005144004190426

Castro Fernandez, R., Deng, D., Mansour, E., Qahtan, A. A., Tao, W., Abedjan, Z., ... Tang, N. (2017). A Demo of the Data Civilizer System. *Proceedings of the 2017 ACM International Conference on Management of Data*, 1639–1642. https://doi.org/10.1145/3035918.3058740

Cattell, R. (2011). Scalable SQL and NoSQL Data Stores. *SIGMOD Rec.*, *39*(4), 12–27. https://doi.org/10.1145/1978915.1978919

Chandarana, P., & Vijayalakshmi, M. (2014). Big Data analytics frameworks. *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, 430–434. https://doi.org/10.1109/CSCITA.2014.6839299

Chang, B. R., Tsai, H.-F., & Wang, Y.-A. (2016). Optimized Multiple Platforms for Big Data Analysis. *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, 155–158. https://doi.org/10.1109/BigMM.2016.61

Chaudhuri, S., & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Rec.*, *26*(1), 65–74. https://doi.org/10.1145/248603.248616

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. *Mob. Netw. Appl.*, *19*(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

Costa, C., & Santos, M. Y. (2016). Data models in NoSQL databases for big data contexts. *Lecture Notes in Computer Science*, *9714*, 475–485. https://doi.org/10.1007/978-3-319-40973-3_48

Costa, C., & Santos, M. Y. (2017b). Big Data: State-of-the-art concepts, techniques, technologies, modeling approaches and research challenges. *IAENG International Journal of Computer Science*, *43*, 285–301. Retrieved from http://hdl.handle.net/1822/46855

Costa, C., & Santos, M. Y. (2017c). The SusCity Big Data Warehousing Approach for Smart Cities. *Proceedings of the 21st International Database Engineering & Applications Symposium*, 264–273. https://doi.org/10.1145/3105831.3105841

Costa, E., Costa, C., & Santos, M. Y. (2018). Partitioning and bucketing in hive-based big data warehouses. *Advances in Intelligent Systems and Computing*, *746*, 764–774. https://doi.org/10.1007/978-3-319-77712-2_72

Cuzzocrea, A., Song, I.-Y., & Davis, K. C. (2011). Analytics over Large-scale Multidimensional Data: The Big Data Revolution! *Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP*, 101–104. https://doi.org/10.1145/2064676.2064695

Dai, W., Wardlaw, I., Cui, Y., Mehdi, K., Li, Y., & Long, J. (2016). Data Profiling Technology of Data Governance Regarding Big Data: Review and Rethinking. In S. Latifi (Ed.), *Information Technology: New Generations* (pp. 439–450). Springer International Publishing.

DAMA. (2017). *DAMA-DMBOK: Data Management Body of Knowledge* (Second edition). Basking Ridge, New Jersey: Technics Publications.

Das Sarma, A., Fang, L., Gupta, N., Halevy, A., Lee, H., Wu, F., … Yu, C. (2012). Finding Related Tables. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 817–828. https://doi.org/10.1145/2213836.2213962

De Mauro, A., Greco, M., & Grimaldi, M. (2015, February 1). *What is big data? A consensual definition and a review of key research topics*. *1644*, 97–104. https://doi.org/10.1063/1.4907823

De Mauro, A., Greco, M., & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library Review*, *65*(3), 122–135. https://doi.org/10.1108/LR-06-2015-0061

Deb Nath, R. P., Hose, K., & Pedersen, T. B. (2015). Towards a Programmable Semantic Extract-Transform-Load Framework for Semantic Data Warehouses. *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP*, 15–24. https://doi.org/10.1145/2811222.2811229

Dehdouh, K., Bentayeb, F., Boussaid, O., & Kabachi, N. (2015). *Using the column oriented NoSQL model for implementing big data warehouses*. https://doi.org/10.5220/0005379801720183

Du, D. (2015). *Apache Hive Essentials*. Birmingham; Mumbaj: Packt Publishing.

Dumbill, E. (2012). Making Sense of Big Data. *Big Data*, *1*(1), 1–2. https://doi.org/10.1089/big.2012.1503

Durham, E.-E. A., Rosen, A., & Harrison, R. W. (2014). A model architecture for Big Data applications using relational databases. *2014 IEEE International Conference on Big Data (Big Data)*, 9–16. https://doi.org/10.1109/BigData.2014.7004462

El Hajjamy, O., Alaoui, L., & Bahaj, M. (2018). Semantic Integration of Heterogeneous Classical Data Sources in Ontological Data Warehouse. *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, 36:1–36:8. https://doi.org/10.1145/3230905.3230929

Elmasri, R., & Navathe, S. B. (2010). *Sistemas de base de dados* (Edição: 6ª). São Paulo: Pearson.

Euzenat, J., & Shvaiko, P. (2013). *Ontology Matching* (2nd ed.). Retrieved from https://www.springer.com/us/book/9783642387203

Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data Analysis. *National Science Review*, *1*(2), 293–314. https://doi.org/10.1093/nsr/nwt032

Fan, S., Lau, R. Y. K., & Zhao, J. L. (2015). Demystifying Big Data Analytics for Business Intelligence Through the Lens of Marketing Mix. *Big Data Research*, *2*(1), 28–32. https://doi.org/10.1016/j.bdr.2015.02.006

Fatima, H., & Wasnik, K. (2016). Comparison of SQL, NoSQL and NewSQL databases for internet of things. *2016 IEEE Bombay Section Symposium (IBSS)*, 1–6. https://doi.org/10.1109/IBSS.2016.7940198

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Gardner, S. R. (1998). Building the Data Warehouse. *Commun. ACM*, *41*(9), 52–60. https://doi.org/10.1145/285070.285080

Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL Database Systems: A Survey and Decision Guidance. *Comput. Sci.*, *32*(3–4), 353–365. https://doi.org/10.1007/s00450-016-0334-3

Giorgini, P., Rizzi, S., & Garzetti, M. (2008). GRAnD: A goal-oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, *45*, 4–21. https://doi.org/10.1016/j.dss.2006.12.001

Golfarelli, M., & Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies* (1st ed.). New York, NY, USA: McGraw-Hill, Inc.

Gravano, L., Ipeirotis, P. G., Jagadish, H. V., Koudas, N., Muthukrishnan, S., & Srivastava, D. (2001). Approximate String Joins in a Database (Almost) for Free. *Proceedings of the 27th International Conference on Very Large Data Bases*, 491–500. Retrieved from http://dl.acm.org/citation.cfm?id=645927.672200

Hai, R., Geisler, S., & Quix, C. (2016). Constance: An Intelligent Data Lake System. *Proceedings of the 2016 International Conference on Management of Data*, 2097–2100. https://doi.org/10.1145/2882903.2899389

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*, *154*, 72–80. https://doi.org/10.1016/j.ijpe.2014.04.018

He, J. C. (2018). Apache Atlas and JanusGraph—Graph-based Meta Data Management. *IBM Developer*. Retrieved 01 December 2018 from https://developer.ibm.com/articles/apache-atlas-and-janusgraph-graph-based-meta-data-management/

Henning, K. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. Retrieved from http://thuvienso.dastic.vn:801/dspace/handle/TTKHCNDaNang_123456789/357

Holmes, A. (2012). *Hadoop in Practice* (1 edition). Greenwich, CT, USA: Manning Publications Co.

Hortonworks. (2017). *Hortonworks Data Platform—Data Governance*. 55.

Huang, H., & Dong, Z. (2013). Research on architecture and query performance based on distributed graph database Neo4j. *2013 3rd International Conference on Consumer Electronics, Communications and Networks*, 533–536. https://doi.org/10.1109/CECNet.2013.6703387

Huo, Y., Wang, H., Hu, L., & Yang, H. (2011). A Cloud Storage Architecture Model for Data-Intensive Applications. *2011 International Conference on Computer and Management (CAMAN)*, 1–4. https://doi.org/10.1109/CAMAN.2011.5778817

Inmon, W. H. (2002). *Building the Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc.

Janet, S., Schroeck, R., Shockley, S., Morales, R., & Tufano, P. (2012). *Analytics: The real-world use of big data—How innovative enterprises extract value from uncertain data*.

JanusGraph: Distributed graph database. (2019). Retrieved 23 March 2019, from https://janusgraph.org/

Juddoo, S. (2015). Overview of data quality challenges in the context of Big Data. *2015 International Conference on Computing, Communication and Security (ICCCS)*, 1–9. https://doi.org/10.1109/CCCS.2015.7374131

Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big Data: Issues and Challenges Moving Forward. *Proceedings of the 2013 46th Hawaii International Conference on System Sciences*, 995–1004. https://doi.org/10.1109/HICSS.2013.645

Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: Issues, challenges, tools and Good practices. *2013 Sixth International Conference on Contemporary Computing (IC3)*, 404–409. https://doi.org/10.1109/IC3.2013.6612229

KAYAK: A Framework for Just-in-Time Data Preparation in a Data Lake. (2017). *Advanced Information Systems Engineering*, 474–489. Springer International Publishing.

Kepner, J., Gadepally, V., Hutchison, D., Jananthan, H., Mattson, T., Samsi, S., & Reuther, A. (2016). Associative Array Model of SQL, NoSQL, and NewSQL Databases. *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–9. https://doi.org/10.1109/HPEC.2016.7761647

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, 1–5. https://doi.org/10.1109/ASEEZone1.2014.6820689

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley Publishing.

Krishnan, K. (2013). *Data Warehousing in the Age of Big Data*. https://doi.org/10.1016/B978-0-12-405891-0.00001-5

Kubina, M., Varmus, M., & Kubinova, I. (2015). Use of Big Data for Competitive Advantage of Company. *Procedia Economics and Finance*, *26*, 561–565. https://doi.org/10.1016/S2212-5671(15)00955-7

Lam, C. (2010). *Hadoop in Action* (1st ed.). Greenwich, CT, USA: Manning Publications Co.

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*.

LaValle, S., Lesser, E., Shockley, R., & Hopkins, M. S. (2011). Big Data, Analytics and the Path From Insights to Value. Retrieved 18 November 2018, from https://hbr.org/product/big-data-analytics-and-the-path-from-insights-to-value/SMR372-PDF-ENG

Li, C., Lu, J., & Lu, Y. (2008). Efficient Merging and Filtering Algorithms for Approximate String Searches. *2008 IEEE 24th International Conference on Data Engineering*, 257–266. https://doi.org/10.1109/ICDE.2008.4497434

Li, Y., & Manoharan, S. (2013). *A performance comparison of SQL and NoSQL databases*. 15–19. https://doi.org/10.1109/PACRIM.2013.6625441

Lim, H., Han, Y., & Babu, S. (2013). How to Fit when No One Size Fits. *CIDR*.

Maccioni, A., & Torlone, R. (2017). Crossing the Finish Line Faster when Paddling the Data Lake with KAYAK. *Proc. VLDB Endow.*, *10*(12), 1853–1856. https://doi.org/10.14778/3137765.3137792

Maccioni, A., & Torlone, R. (2018). KAYAK: A Framework for Just-in-Time Data Preparation in a Data Lake. In J. Krogstie & H. A. Reijers (Eds.), *Advanced Information Systems Engineering* (pp. 474–489). Springer International Publishing.

Madhavan, J., Bernstein, P. A., & Rahm, E. (2001). Generic Schema Matching with Cupid. *Proceedings of the 27th International Conference on Very Large Data Bases*, 49–58. Retrieved from http://dl.acm.org/citation.cfm?id=645927.672191

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity. Retrieved 18 November 2018, from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation

McAfee, A., & Brynjolfsson, E. (2012, October 1). Big Data: The Management Revolution. *Harvard Business Review*, (October 2012). Retrieved from https://hbr.org/2012/10/big-data-the-management-revolution

Nambiar, R., & Poess, M. (2006, January 1). *The Making of TPC-DS*. 1049–1058.

Naumann, F. (2014). Data Profiling Revisited. *SIGMOD Rec.*, *42*(4), 40–49. https://doi.org/10.1145/2590989.2590995

O'Shea, J., Bandar, Z., Crockett, K., & McLean, D. (2010). Benchmarking Short Text Semantic Similarity. *Int. J. Intell. Inf. Database Syst.*, *4*(2), 103–120. https://doi.org/10.1504/IJIIDS.2010.032437

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431–448. https://doi.org/10.1016/j.jksuci.2017.06.001

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst.*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Philip Chen, C. L., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Qin, H., Qian, Z., & Zhao, Y. (2015). On the Research of Data Warehouse in Big Data. *2015 International Conference on Network and Information Systems for Computers*, 354–357. https://doi.org/10.1109/ICNISC.2015.126

Robinson, I., Webber, J., & Eifrem, E. (2013). *Graph Databases* (1 edition). Beijing ; Sebastopol, CA: O'Reilly Media.

Rodrigues, A. (2017). *Data Profling: Identification of Data Quality Problems through Data Analysis*. Retrieved from https://fenix.tecnico.ulisboa.pt/downloadFile/395137830839/Artigo_Ana_20Rodrigues_Data20Profiling.pdf

Román, R., & Fabián, J. (2018). *Diseño e desarrollo de un sistema de información genómica basado en un modelo conceptual holístico del genoma humano*. https://doi.org/10.4995/Thesis/10251/99565

Russom, P. (2011). Big Data Analytics. *TDWI Research*.

Sá, J. V. de O. e. (2010). *Metodologia de sistemas de data warehouse*. Retrieved from http://repositorium.sdum.uminho.pt/handle/1822/10663

Santos, M. Y., & Costa, C. (2016). Data Warehousing in Big Data: From Multidimensional to Tabular Data Models. *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering*, 51–60. https://doi.org/10.1145/2948992.2949024

Santos, M. Y., & Ramos, I. (2017). *Business Intelligence Da Informação ao Conhecimento*. FCA.

Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLOS ONE*, *10*(12), e0144059. https://doi.org/10.1371/journal.pone.0144059

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. https://doi.org/10.1109/MSST.2010.5496972

Smallwood, R. F. (2014). *Information Governance: Concepts, Strategies, and Best Practices* (1 edition). Hoboken, New Jersey: Wiley.

Stonebraker, M. (2010). SQL Databases V. NoSQL Databases. *Commun. ACM*, *53*(4), 10–11. https://doi.org/10.1145/1721654.1721659

Thusoo, A., Shao, Z., Anthony, S., Borthakur, D., Jain, N., Sarma, J. S., … Liu, H. (2010). Data warehousing and analytics infrastructure at facebook. *SIGMOD Conference*. https://doi.org/10.1145/1807167.1807278

Tirumali, P. (2016). Efficient pair-wise similarity computation using apache spark. *Master's Projects*.

Tria, F. D., Lefons, E., & Tangorra, F. (2014). Design process for Big Data Warehouses. *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 512–518. https://doi.org/10.1109/DSAA.2014.7058120

Ukkonen, E. (1992). Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, *92*(1), 191–211. https://doi.org/10.1016/0304-3975(92)90143-4

Vaish, G. (2013). *Getting Started with NoSQL*. Packt Publishing.

Vaisman, A., & Zimányi, E. (2012). Data Warehouses: Next Challenges. In M.-A. Aufaure & E. Zimányi (Eds.), *Business Intelligence: First European Summer School, eBISS 2011, Paris, France, July 3-8, 2011, Tutorial Lectures* (pp. 1–26). https://doi.org/10.1007/978-3-642-27358-2_1

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A Comparison of a Graph Database and a Relational Database: A Data Provenance Perspective. *Proceedings of the 48th Annual Southeast Regional Conference*, 42:1–42:6. https://doi.org/10.1145/1900008.1900067

Waller, M. A., & Fawcett, S. E. (2013). Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *Journal of Business Logistics*, *34*(2), 77–84. https://doi.org/10.1111/jbl.12010

Wang, H., Can, D., Kazemzadeh, A., Bar, F., & Narayanan, S. (2012). A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle. *Proceedings of the ACL 2012 System Demonstrations*, 115–120. Retrieved from http://dl.acm.org/citation.cfm?id=2390470.2390490

Ward, J. S., & Barker, A. (2013). *Undefined By Data: A Survey of Big Data Definitions*.

Xiao, C., Wang, W., Lin, X., & Shang, H. (2009). Top-k Set Similarity Joins. *2009 IEEE 25th International Conference on Data Engineering*, 916–927. https://doi.org/10.1109/ICDE.2009.111

Zhu, E., Deng, D., Nargesian, F., & Miller, R. J. (2019). JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. *Proceedings of the 2019 International Conference on Management of Data*, 847–864. https://doi.org/10.1145/3299869.3300065

Zikopoulos, P. (2011). *Understanding Big Data: Analytics For Enterprise Class Hadoop And Streaming Data, 1Ed* (1st edition). McGraw Hill Education India Pvt Ltd.

APÊNDICE

Apêndice 1 – Resultados Cenário AR

A Tabela 18 apresenta os valores da similaridade associados às medidas de *Jaccard*, *Cosine*, *Levenshtein* e *Jaro-Winkler* para o Cenário **AR**.

Tabela 18. Valores da Similaridade do Cenário AR.

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_promo_sk-ss_promo_sk	**0,70**	**0,87**	**0,82**	**0,84**
p_item_sk-ss_item_sk	0,67	**0,90**	**0,80**	**0,82**
p_end_date_sk-ss_sold_date_sk	0,41	**0,81**	0,67	0,65
p_start_date_sk-ss_sold_date_sk	0,30	0,68	0,60	0,63
p_promo_id_b-ss_promo_sk	0,36	0,69	0,50	0,57
p_promo_name-ss_promo_sk	0,36	0,69	0,50	0,57
p_discount_active-ss_ext_discount_amt	0,39	0,61	0,47	0,61
p_promo_sk-ss_cdemo_sk	0,21	0,66	0,55	0,42
p_promo_sk-ss_hdemo_sk	0,21	0,66	0,55	0,42
p_start_date_sk-ss_store_sk	0,16	0,67	0,47	0,51
p_item_sk-ss_cdemo_sk	0,07	**0,74**	0,55	0,34
p_item_sk-ss_hdemo_sk	0,07	**0,74**	0,55	0,34
p_end_date_sk-ss_sold_time_sk	0,09	**0,72**	0,47	0,36
p_start_date_sk-ss_sold_time_sk	0,08	0,66	0,40	0,38
p_promo_sk-ss_store_sk	0,06	0,66	0,45	0,29
p_item_sk-ss_store_sk	0,07	0,63	0,45	0,31
p_item_sk-ss_promo_sk	0,07	0,68	0,45	0,22
p_start_date_sk-ss_item_sk	0,05	0,63	0,40	0,33
p_start_date_sk-ss_addr_sk	0,05	0,69	0,40	0,25
p_end_date_sk-ss_store_sk	0,11	0,61	0,31	0,33
p_end_date_sk-ss_item_sk	0,06	0,63	0,38	0,29
p_promo_sk-ss_item_sk	0,07	0,67	0,40	0,22
p_cost-ss_wholesale_cost	0,19	0,45	0,29	0,42
p_item_sk-ss_customer_sk	0,06	0,66	0,43	0,20
p_end_date_sk-ss_addr_sk	0,06	**0,70**	0,38	0,19
p_promo_sk-ss_customer_sk	0,05	0,57	0,43	0,28
p_item_sk-ss_addr_sk	0,07	0,61	0,40	0,24
p_item_sk-ss_sold_time_sk	0,05	0,58	0,40	0,27
p_channel_press-ss_sales_price	0,04	0,64	0,33	0,28
p_start_date_sk-ss_promo_sk	0,05	0,65	0,33	0,24
p_promo_sk-ss_addr_sk	0,07	0,67	0,30	0,22
p_promo_sk-ss_sold_time_sk	0,05	0,61	0,33	0,25
p_channel_press-ss_net_profit	0,04	0,62	0,27	0,31
p_item_sk-ss_sold_date_sk	0,05	0,49	0,33	0,35
p_start_date_sk-ss_cdemo_sk	0,05	0,61	0,33	0,24
p_start_date_sk-ss_hdemo_sk	0,05	0,61	0,33	0,24

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_cost-ss_ext_wholesale_cost	0,15	0,45	0,24	0,38
p_channel_demo-ss_cdemo_sk	0,11	0,62	0,14	0,35
p_start_date_sk-ss_customer_sk	0,04	0,62	0,27	0,28
p_discount_active-ss_coupon_amt	0,04	0,65	0,29	0,22
p_response_target-ss_coupon_amt	0,04	0,63	0,35	0,14
p_channel_press-ss_ext_sales_price	0,04	0,61	0,28	0,25
p_end_date_sk-ss_cdemo_sk	0,05	0,61	0,31	0,18
p_end_date_sk-ss_hdemo_sk	0,05	0,61	0,31	0,18
p_cost-ss_customer_sk	0,00	0,60	0,29	0,25
p_cost-ss_coupon_amt	0,07	0,49	0,31	0,26
p_channel_demo-ss_hdemo_sk	0,11	0,62	0,14	0,26
p_promo_sk-ss_sold_date_sk	0,05	0,54	0,27	0,25
p_discount_active-ss_quantity	0,00	0,65	0,29	0,16
p_start_date_sk-ss_ext_discount_amt	0,03	0,68	0,21	0,17
p_promo_id_b-ss_net_paid_inc_tax	0,04	0,55	0,21	0,28
p_purpose-ss_promo_sk	0,00	0,60	0,36	0,11
p_promo_name-ss_net_profit	0,11	0,54	0,08	0,35
p_promo_name-ss_coupon_amt	0,00	0,68	0,31	0,09
p_discount_active-ss_list_price	0,00	0,57	0,35	0,14
p_end_date_sk-ss_customer_sk	0,05	0,57	0,29	0,16
p_channel_press-ss_wholesale_cost	0,00	0,64	0,24	0,19
p_promo_sk-ss_net_profit	0,12	0,49	0,15	0,29
p_promo_id_b-ss_cdemo_sk	0,06	0,56	0,25	0,19
p_promo_id_b-ss_hdemo_sk	0,06	0,56	0,25	0,19
p_discount_active-ss_net_apid	0,04	0,54	0,29	0,16
p_end_date_sk-ss_ext_sales_price	0,00	0,56	0,33	0,13
p_cost-ss_cdemo_sk	0,00	0,51	0,36	0,14
p_end_date_sk-ss_promo_sk	0,05	0,55	0,23	0,18
p_response_target-ss_ext_tax	0,05	0,56	0,24	0,17
p_channel_press-ss_ext_list_price	0,04	0,54	0,24	0,19
p_channel_details-ss_net_apid	0,00	0,61	0,24	0,16
p_end_date_sk-ss_ext_wholesale_cost	0,00	0,54	0,33	0,12
p_start_date_sk-ss_ext_tax	0,00	0,56	0,27	0,17
p_response_target-ss_store_sk	0,00	0,61	0,24	0,14
p_response_target-ss_net_profit	0,00	0,68	0,24	0,07
p_promo_id_b-ss_net_profit	0,11	0,54	0,08	0,26
p_channel_press-ss_list_price	0,04	0,51	0,20	0,23
p_start_date_sk-ss_ext_sales_price	0,00	0,53	0,28	0,17
p_promo_name-ss_cdemo_sk	0,06	0,48	0,25	0,19
p_promo_name-ss_hdemo_sk	0,06	0,48	0,25	0,19
p_start_date_sk-ss_net_paid_inc_tax	0,00	0,65	0,21	0,12
p_promo_id_b-ss_sold_time_sk	0,00	0,64	0,27	0,08
p_channel_radio-ss_net_apid	0,00	0,56	0,33	0,08
p_response_target-ss_net_paid_inc_tax	0,03	0,60	0,16	0,18
p_discount_active-ss_ticket_number	0,00	0,60	0,24	0,13
p_channel_press-ss_cdemo_sk	0,00	0,60	0,20	0,17
p_channel_radio-ss_net_profit	0,00	0,62	0,27	0,08
p_start_date_sk-ss_ext_wholesale_cost	0,00	0,51	0,29	0,16

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_discount_active-ss_ext_list_price	0,00	0,54	0,24	0,18
p_response_target-ss_net_apid	0,00	0,63	0,24	0,08
p_channel_details-ss_net_profit	0,00	0,55	0,24	0,14
p_end_date_sk-ss_ext_discount_amt	0,00	0,54	0,26	0,13
p_channel_details-ss_net_paid_inc_tax	0,00	0,59	0,11	0,24
p_channel_press-ss_ext_wholesale_cost	0,00	0,57	0,19	0,17
p_channel_tv-ss_wholesale_cost	0,00	0,62	0,24	0,07
p_item_sk-ss_ticket_number	0,00	0,67	0,25	0,00
p_channel_tv-ss_coupon_amt	0,00	0,60	0,23	0,09
p_purpose-ss_list_price	0,00	0,58	0,23	0,10
p_cost-ss_store_sk	0,00	0,51	0,27	0,13
p_channel_details-ss_cdemo_sk	0,00	0,58	0,18	0,16
p_end_date_sk-ss_wholesale_cost	0,00	0,55	0,29	0,07
p_channel_tv-ss_quantity	0,00	0,57	0,25	0,10
p_channel_catalog-ss_wholesale_cost	0,00	0,63	0,12	0,17
p_channel_dmail-ss_net_apid	0,00	0,56	0,27	0,08
p_channel_press-ss_net_apid	0,00	0,54	0,20	0,17
p_channel_press-ss_customer_sk	0,00	0,55	0,20	0,15
p_promo_name-ss_ticket_number	0,00	0,51	0,31	0,08
p_response_target-ss_ext_wholesale_cost	0,00	0,60	0,19	0,11
p_channel_demo-ss_coupon_amt	0,00	0,60	0,21	0,08
p_channel_demo-ss_wholesale_cost	0,00	0,60	0,24	0,07
p_channel_details-ss_quantity	0,00	0,58	0,24	0,08
p_start_date_sk-ss_ext_list_price	0,00	0,54	0,18	0,18
p_channel_catalog-ss_ext_wholesale_cost	0,00	0,55	0,19	0,15
p_channel_press-ss_promo_sk	0,05	0,46	0,13	0,25
p_channel_tv-ss_customer_sk	0,00	0,52	0,29	0,08
p_end_date_sk-ss_net_paid_inc_tax	0,00	0,61	0,21	0,07
p_channel_email-ss_net_apid	0,00	0,54	0,27	0,08
p_end_date_sk-ss_ext_tax	0,00	0,56	0,23	0,10
p_promo_name-ss_ext_discount_amt	0,00	0,56	0,26	0,07
p_channel_press-ss_store_sk	0,00	0,53	0,20	0,15
p_start_date_sk-ss_net_apid	0,00	0,54	0,27	0,08
p_channel_event-ss_wholesale_cost	0,00	0,58	0,24	0,06
p_channel_tv-ss_net_paid_inc_tax	0,00	0,53	0,21	0,14
p_channel_press-ss_hdemo_sk	0,00	0,60	0,20	0,08
p_purpose-ss_store_sk	0,00	0,60	0,27	0,00
p_cost-ss_ext_discount_amt	0,00	0,57	0,21	0,10
p_promo_name-ss_sales_price	0,05	0,46	0,21	0,16
p_purpose-ss_customer_sk	0,00	0,59	0,29	0,00
p_promo_name-ss_list_price	0,05	0,49	0,15	0,17
p_channel_event-ss_ext_wholesale_cost	0,00	0,57	0,19	0,11
p_cost-ss_promo_sk	0,00	0,59	0,27	0,00
p_item_sk-ss_net_paid_inc_tax	0,00	0,62	0,16	0,08
p_channel_details-ss_wholesale_cost	0,00	0,63	0,18	0,06
p_promo_name-ss_sold_time_sk	0,00	0,52	0,27	0,08
p_discount_active-ss_ext_tax	0,00	0,54	0,24	0,08
p_end_date_sk-ss_ext_list_price	0,00	0,55	0,24	0,07

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_discount_active-ss_net_paid_inc_tax	0,00	0,64	0,16	0,06
p_response_target-ss_ext_sales_price	0,00	0,63	0,17	0,06
p_promo_name-ss_sold_date_sk	0,00	0,58	0,27	0,00
p_channel_dmail-ss_net_profit	0,00	0,57	0,20	0,08
p_channel_email-ss_net_profit	0,00	0,57	0,20	0,08
p_channel_event-ss_net_profit	0,00	0,57	0,20	0,08
p_channel_demo-ss_net_apid	0,00	0,55	0,21	0,09
p_promo_sk-ss_sales_price	0,05	0,39	0,14	0,26
p_purpose-ss_net_profit	0,00	0,51	0,23	0,10
p_channel_event-ss_quantity	0,00	0,54	0,13	0,17
p_channel_press-ss_item_sk	0,00	0,56	0,20	0,09
p_discount_active-ss_net_profit	0,00	0,59	0,18	0,07
p_promo_id_b-ss_customer_sk	0,00	0,54	0,21	0,08
p_start_date_sk-ss_wholesale_cost	0,00	0,48	0,24	0,12
p_promo_name-ss_ext_list_price	0,04	0,48	0,18	0,14
p_channel_tv-ss_ticket_number	0,00	0,57	0,19	0,08
p_promo_name-ss_customer_sk	0,00	0,52	0,14	0,17
p_promo_id_b-ss_sales_price	0,05	0,49	0,14	0,16
p_purpose-ss_coupon_amt	0,00	0,50	0,23	0,10
p_channel_press-ss_addr_sk	0,00	0,54	0,20	0,09
p_channel_details-ss_hdemo_sk	0,00	0,58	0,18	0,08
p_channel_email-ss_cdemo_sk	0,00	0,53	0,13	0,17
p_channel_catalog-ss_net_apid	0,00	0,52	0,24	0,07
p_channel_dmail-ss_net_paid_inc_tax	0,00	0,60	0,11	0,13
p_promo_id_b-ss_net_apid	0,00	0,57	0,17	0,10
p_channel_tv-ss_cdemo_sk	0,00	0,57	0,17	0,10
p_cost-ss_list_price	0,00	0,47	0,23	0,13
p_promo_id_b-ss_ext_list_price	0,04	0,52	0,12	0,14
p_channel_email-ss_wholesale_cost	0,00	0,58	0,18	0,06
p_channel_catalog-ss_coupon_amt	0,00	0,51	0,18	0,14
p_channel_radio-ss_wholesale_cost	0,00	0,58	0,18	0,06
p_response_target-ss_sales_price	0,00	0,51	0,24	0,06
p_channel_email-ss_ext_wholesale_cost	0,00	0,55	0,14	0,11
p_item_sk-ss_list_price	0,00	0,58	0,23	0,00
p_purpose-ss_sales_price	0,00	0,50	0,21	0,09
p_channel_radio-ss_sales_price	0,00	0,61	0,20	0,00
p_channel_catalog-ss_net_paid_inc_tax	0,00	0,54	0,16	0,11
p_channel_dmail-ss_coupon_amt	0,00	0,53	0,20	0,08
p_channel_email-ss_coupon_amt	0,00	0,53	0,20	0,08
p_channel_event-ss_coupon_amt	0,00	0,53	0,20	0,08
p_channel_catalog-ss_cdemo_sk	0,00	0,53	0,12	0,15
p_channel_catalog-ss_net_profit	0,00	0,55	0,18	0,07
p_response_target-ss_ticket_number	0,00	0,55	0,12	0,13
p_channel_press-ss_quantity	0,00	0,49	0,13	0,17
p_promo_sk-ss_coupon_amt	0,00	0,56	0,23	0,00
p_purpose-ss_cdemo_sk	0,00	0,52	0,27	0,00
p_purpose-ss_hdemo_sk	0,00	0,52	0,27	0,00
p_promo_id_b-ss_sold_date_sk	0,00	0,52	0,20	0,08

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_channel_details-ss_ext_tax	0,00	0,53	0,18	0,08
p_start_date_sk-ss_list_price	0,00	0,51	0,13	0,14
p_channel_catalog-ss_ext_tax	0,00	0,53	0,18	0,08
p_channel_radio-ss_net_paid_inc_tax	0,00	0,56	0,16	0,06
p_response_target-ss_sold_time_sk	0,00	0,48	0,18	0,13
p_channel_catalog-ss_quantity	0,00	0,53	0,18	0,07
p_end_date_sk-ss_net_apid	0,00	0,63	0,15	0,00
p_response_target-ss_sold_date_sk	0,00	0,54	0,18	0,06
p_channel_details-ss_sold_date_sk	0,00	0,54	0,18	0,06
p_discount_active-ss_sold_date_sk	0,00	0,54	0,18	0,06
p_discount_active-ss_sold_time_sk	0,00	0,54	0,18	0,06
p_start_date_sk-ss_net_profit	0,00	0,51	0,20	0,07
p_item_sk-ss_ext_sales_price	0,00	0,48	0,22	0,08
p_promo_name-ss_ext_sales_price	0,04	0,44	0,17	0,14
p_discount_active-ss_ext_sales_price	0,00	0,56	0,17	0,06
p_channel_details-ss_sales_price	0,00	0,54	0,24	0,00
p_discount_active-ss_sales_price	0,00	0,54	0,24	0,00
p_channel_catalog-ss_sales_price	0,00	0,54	0,18	0,06
p_channel_demo-ss_ext_wholesale_cost	0,00	0,53	0,19	0,06
p_start_date_sk-ss_sales_price	0,00	0,51	0,13	0,13
p_channel_email-ss_net_paid_inc_tax	0,00	0,54	0,11	0,13
p_response_target-ss_wholesale_cost	0,00	0,54	0,12	0,12
p_channel_dmail-ss_wholesale_cost	0,00	0,53	0,18	0,06
p_response_target-ss_ext_list_price	0,00	0,59	0,18	0,00
p_response_target-ss_list_price	0,00	0,53	0,24	0,00
p_response_target-ss_quantity	0,00	0,59	0,18	0,00
p_channel_press-ss_net_paid_inc_tax	0,00	0,47	0,11	0,19
p_promo_sk-ss_ext_sales_price	0,04	0,38	0,11	0,23
p_promo_id_b-ss_item_sk	0,00	0,49	0,17	0,10
p_channel_dmail-ss_quantity	0,00	0,54	0,13	0,08
p_channel_email-ss_quantity	0,00	0,54	0,13	0,08
p_channel_radio-ss_cdemo_sk	0,00	0,54	0,13	0,08
p_channel_radio-ss_quantity	0,00	0,54	0,13	0,08
p_channel_event-ss_cdemo_sk	0,00	0,54	0,13	0,08
p_cost-ss_sold_date_sk	0,00	0,56	0,20	0,00
p_cost-ss_sold_time_sk	0,00	0,56	0,20	0,00
p_channel_email-ss_sales_price	0,00	0,56	0,20	0,00
p_item_sk-ss_net_profit	0,00	0,50	0,15	0,10
p_channel_tv-ss_ext_tax	0,00	0,57	0,08	0,10
p_channel_details-ss_coupon_amt	0,00	0,51	0,18	0,07
p_channel_dmail-ss_cdemo_sk	0,00	0,54	0,13	0,08
p_channel_dmail-ss_sales_price	0,00	0,55	0,20	0,00
p_channel_event-ss_sales_price	0,00	0,55	0,20	0,00
p_cost-ss_hdemo_sk	0,00	0,48	0,27	0,00
p_channel_email-ss_ext_sales_price	0,00	0,52	0,17	0,06
p_channel_event-ss_ext_sales_price	0,00	0,52	0,17	0,06
p_channel_details-ss_ext_wholesale_cost	0,00	0,55	0,14	0,05
p_channel_tv-ss_ext_wholesale_cost	0,00	0,50	0,19	0,06

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_channel_tv-ss_net_apid	0,00	0,57	0,08	0,10
p_response_target-ss_customer_sk	0,00	0,51	0,24	0,00
p_channel_event-ss_net_paid_inc_tax	0,00	0,53	0,16	0,06
p_channel_event-ss_net_apid	0,00	0,46	0,20	0,08
p_channel_email-ss_hdemo_sk	0,00	0,53	0,13	0,08
p_channel_dmail-ss_ext_discount_amt	0,00	0,58	0,11	0,06
p_purpose-ss_addr_sk	0,00	0,54	0,20	0,00
p_channel_press-ss_coupon_amt	0,00	0,45	0,13	0,15
p_promo_name-ss_net_paid_inc_tax	0,00	0,44	0,16	0,14
p_item_sk-ss_ext_list_price	0,00	0,56	0,18	0,00
p_channel_demo-ss_ticket_number	0,00	0,55	0,19	0,00
p_purpose-ss_wholesale_cost	0,00	0,48	0,18	0,08
p_channel_press-ss_ticket_number	0,00	0,54	0,13	0,07
p_channel_details-ss_ext_discount_amt	0,00	0,57	0,11	0,06
p_channel_tv-ss_hdemo_sk	0,00	0,57	0,17	0,00
p_channel_catalog-ss_customer_sk	0,00	0,48	0,12	0,13
p_discount_active-ss_customer_sk	0,00	0,56	0,18	0,00
p_channel_demo-ss_net_profit	0,00	0,58	0,07	0,08
p_channel_radio-ss_coupon_amt	0,00	0,45	0,20	0,08
p_cost-ss_ext_list_price	0,00	0,45	0,18	0,11
p_channel_demo-ss_customer_sk	0,00	0,51	0,14	0,08
p_channel_email-ss_ticket_number	0,00	0,54	0,19	0,00
p_channel_demo-ss_quantity	0,00	0,50	0,14	0,09
p_channel_details-ss_customer_sk	0,00	0,54	0,12	0,07
p_channel_event-ss_customer_sk	0,00	0,52	0,13	0,07
p_promo_id_b-ss_ext_sales_price	0,04	0,44	0,11	0,14
p_promo_sk-ss_list_price	0,06	0,40	0,08	0,19
p_channel_dmail-ss_ticket_number	0,00	0,54	0,19	0,00
p_item_sk-ss_sales_price	0,00	0,41	0,21	0,09
p_channel_demo-ss_item_sk	0,00	0,48	0,14	0,09
p_channel_catalog-ss_sold_date_sk	0,00	0,47	0,18	0,06
p_promo_id_b-ss_ext_discount_amt	0,00	0,55	0,16	0,00
p_item_sk-ss_ext_discount_amt	0,00	0,55	0,16	0,00
p_channel_demo-ss_ext_discount_amt	0,00	0,54	0,11	0,06
p_discount_active-ss_ext_wholesale_cost	0,00	0,50	0,10	0,11
p_channel_demo-ss_sales_price	0,00	0,56	0,14	0,00
p_channel_dmail-ss_ext_wholesale_cost	0,00	0,50	0,14	0,06
p_channel_radio-ss_ext_wholesale_cost	0,00	0,46	0,19	0,06
p_channel_dmail-ss_customer_sk	0,00	0,50	0,13	0,07
p_channel_email-ss_customer_sk	0,00	0,50	0,13	0,07
p_channel_radio-ss_customer_sk	0,00	0,50	0,13	0,07
p_channel_details-ss_ticket_number	0,00	0,52	0,12	0,06
p_item_sk-ss_ext_wholesale_cost	0,00	0,46	0,24	0,00
p_channel_details-ss_sold_time_sk	0,00	0,58	0,12	0,00
p_promo_sk-ss_ext_list_price	0,05	0,38	0,12	0,16
p_item_sk-ss_net_apid	0,00	0,52	0,18	0,00
p_cost-ss_net_paid_inc_tax	0,00	0,54	0,16	0,00
p_channel_tv-ss_net_profit	0,00	0,53	0,08	0,09

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_promo_id_b-ss_ticket_number	0,00	0,51	0,19	0,00
p_cost-ss_net_profit	0,00	0,47	0,23	0,00
p_response_target-ss_ext_discount_amt	0,00	0,54	0,16	0,00
p_channel_details-ss_ext_sales_price	0,00	0,53	0,17	0,00
p_channel_email-ss_item_sk	0,00	0,47	0,13	0,09
p_channel_email-ss_ext_tax	0,00	0,47	0,13	0,09
p_channel_radio-ss_addr_sk	0,00	0,47	0,13	0,09
p_channel_email-ss_ext_discount_amt	0,00	0,53	0,11	0,06
p_promo_sk-ss_net_paid_inc_tax	0,00	0,45	0,16	0,08
p_channel_radio-ss_ext_sales_price	0,00	0,52	0,17	0,00
p_cost-ss_item_sk	0,00	0,49	0,20	0,00
p_cost-ss_ext_tax	0,00	0,49	0,20	0,00
p_promo_id_b-ss_coupon_amt	0,00	0,53	0,15	0,00
p_channel_event-ss_ext_discount_amt	0,00	0,46	0,11	0,12
p_promo_id_b-ss_list_price	0,05	0,38	0,08	0,17
p_promo_name-ss_item_sk	0,00	0,52	0,17	0,00
p_promo_name-ss_ext_tax	0,00	0,52	0,17	0,00
p_channel_tv-ss_item_sk	0,00	0,52	0,17	0,00
p_channel_tv-ss_addr_sk	0,00	0,52	0,17	0,00
p_promo_name-ss_wholesale_cost	0,00	0,50	0,18	0,00
p_channel_radio-ss_hdemo_sk	0,00	0,54	0,13	0,00
p_channel_event-ss_hdemo_sk	0,00	0,54	0,13	0,00
p_promo_name-ss_ext_wholesale_cost	0,00	0,48	0,19	0,00
p_channel_tv-ss_store_sk	0,00	0,51	0,17	0,00
p_item_sk-ss_quantity	0,00	0,47	0,09	0,11
p_channel_dmail-ss_hdemo_sk	0,00	0,54	0,13	0,00
p_start_date_sk-ss_ticket_number	0,00	0,48	0,13	0,06
p_channel_demo-ss_net_paid_inc_tax	0,00	0,50	0,11	0,07
p_channel_tv-ss_sales_price	0,00	0,52	0,14	0,00
p_channel_tv-ss_ext_sales_price	0,00	0,50	0,17	0,00
p_end_date_sk-ss_sales_price	0,00	0,52	0,07	0,07
p_channel_event-ss_ticket_number	0,00	0,54	0,13	0,00
p_discount_active-ss_addr_sk	0,00	0,46	0,12	0,08
p_start_date_sk-ss_quantity	0,00	0,46	0,20	0,00
p_cost-ss_quantity	0,00	0,48	0,18	0,00
p_channel_radio-ss_ticket_number	0,00	0,54	0,13	0,00
p_promo_name-ss_net_apid	0,00	0,48	0,08	0,10
p_response_target-ss_item_sk	0,00	0,48	0,18	0,00
p_cost-ss_ticket_number	0,00	0,47	0,19	0,00
p_channel_tv-ss_sold_time_sk	0,00	0,45	0,13	0,08
p_response_target-ss_promo_sk	0,00	0,54	0,12	0,00
p_promo_name-ss_store_sk	0,00	0,49	0,17	0,00
p_purpose-ss_net_paid_inc_tax	0,00	0,47	0,11	0,08
p_channel_tv-ss_sold_date_sk	0,00	0,52	0,13	0,00
p_channel_radio-ss_ext_discount_amt	0,00	0,54	0,05	0,06
p_channel_email-ss_ext_list_price	0,00	0,47	0,12	0,06
p_channel_event-ss_ext_list_price	0,00	0,47	0,12	0,06
p_purpose-ss_ext_list_price	0,00	0,39	0,18	0,08

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_channel_tv-ss_ext_discount_amt	0,00	0,49	0,16	0,00
p_promo_id_b-ss_wholesale_cost	0,00	0,48	0,18	0,00
p_channel_catalog-ss_hdemo_sk	0,00	0,53	0,12	0,00
p_channel_details-ss_store_sk	0,00	0,53	0,12	0,00
p_purpose-ss_quantity	0,00	0,47	0,18	0,00
p_discount_active-ss_item_sk	0,00	0,53	0,12	0,00
p_promo_id_b-ss_store_sk	0,00	0,48	0,17	0,00
p_purpose-ss_ext_wholesale_cost	0,00	0,38	0,19	0,08
p_channel_demo-ss_ext_sales_price	0,00	0,48	0,17	0,00
p_response_target-ss_addr_sk	0,00	0,46	0,18	0,00
p_discount_active-ss_wholesale_cost	0,00	0,52	0,06	0,06
p_response_target-ss_cdemo_sk	0,00	0,52	0,12	0,00
p_response_target-ss_hdemo_sk	0,00	0,52	0,12	0,00
p_discount_active-ss_cdemo_sk	0,00	0,52	0,12	0,00
p_channel_demo-ss_store_sk	0,00	0,50	0,14	0,00
p_channel_catalog-ss_ticket_number	0,00	0,52	0,12	0,00
p_promo_sk-ss_ext_wholesale_cost	0,00	0,45	0,19	0,00
p_purpose-ss_ext_sales_price	0,00	0,39	0,17	0,08
p_channel_demo-ss_sold_date_sk	0,00	0,50	0,07	0,07
p_discount_active-ss_hdemo_sk	0,00	0,52	0,12	0,00
p_discount_active-ss_promo_sk	0,00	0,52	0,12	0,00
p_channel_dmail-ss_ext_sales_price	0,00	0,47	0,17	0,00
p_channel_email-ss_addr_sk	0,00	0,50	0,13	0,00
p_channel_radio-ss_item_sk	0,00	0,50	0,13	0,00
p_channel_event-ss_item_sk	0,00	0,50	0,13	0,00
p_channel_event-ss_addr_sk	0,00	0,50	0,13	0,00
p_purpose-ss_item_sk	0,00	0,43	0,20	0,00
p_item_sk-ss_ext_tax	0,00	0,53	0,10	0,00
p_channel_demo-ss_promo_sk	0,00	0,47	0,07	0,09
p_item_sk-ss_wholesale_cost	0,00	0,39	0,24	0,00
p_channel_press-ss_ext_tax	0,00	0,47	0,07	0,09
p_channel_event-ss_ext_tax	0,00	0,47	0,07	0,09
p_discount_active-ss_store_sk	0,00	0,45	0,18	0,00
p_channel_dmail-ss_store_sk	0,00	0,49	0,13	0,00
p_channel_email-ss_store_sk	0,00	0,49	0,13	0,00
p_channel_event-ss_store_sk	0,00	0,49	0,13	0,00
p_channel_details-ss_ext_list_price	0,00	0,56	0,06	0,00
p_channel_catalog-ss_ext_sales_price	0,00	0,46	0,11	0,05
p_channel_press-ss_sold_date_sk	0,00	0,49	0,07	0,07
p_channel_press-ss_sold_time_sk	0,00	0,49	0,07	0,07
p_channel_demo-ss_addr_sk	0,00	0,48	0,14	0,00
p_channel_tv-ss_ext_list_price	0,00	0,50	0,12	0,00
p_purpose-ss_sold_date_sk	0,00	0,49	0,13	0,00
p_purpose-ss_sold_time_sk	0,00	0,49	0,13	0,00
p_promo_sk-ss_wholesale_cost	0,00	0,38	0,24	0,00
p_item_sk-ss_coupon_amt	0,00	0,46	0,15	0,00
p_promo_sk-ss_ext_discount_amt	0,00	0,45	0,16	0,00
p_channel_details-ss_list_price	0,00	0,49	0,12	0,00

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_channel_catalog-ss_ext_discount_amt	0,00	0,51	0,11	0,00
p_channel_tv-ss_list_price	0,00	0,53	0,08	0,00
p_channel_catalog-ss_item_sk	0,00	0,49	0,12	0,00
p_channel_catalog-ss_addr_sk	0,00	0,49	0,12	0,00
p_channel_details-ss_item_sk	0,00	0,49	0,12	0,00
p_channel_catalog-ss_list_price	0,00	0,55	0,06	0,00
p_channel_dmail-ss_item_sk	0,00	0,47	0,13	0,00
p_channel_dmail-ss_addr_sk	0,00	0,47	0,13	0,00
p_channel_dmail-ss_ext_tax	0,00	0,47	0,13	0,00
p_channel_radio-ss_ext_tax	0,00	0,47	0,13	0,00
p_channel_demo-ss_ext_list_price	0,00	0,48	0,12	0,00
p_channel_catalog-ss_store_sk	0,00	0,48	0,12	0,00
p_promo_name-ss_addr_sk	0,00	0,52	0,08	0,00
p_promo_sk-ss_ticket_number	0,00	0,47	0,13	0,00
p_channel_dmail-ss_ext_list_price	0,00	0,48	0,12	0,00
p_channel_radio-ss_store_sk	0,00	0,46	0,13	0,00
p_channel_radio-ss_ext_list_price	0,00	0,47	0,12	0,00
p_end_date_sk-ss_coupon_amt	0,00	0,59	0,00	0,00
p_channel_details-ss_addr_sk	0,00	0,46	0,12	0,00
p_end_date_sk-ss_ticket_number	0,00	0,52	0,06	0,00
p_promo_id_b-ss_addr_sk	0,00	0,49	0,08	0,00
p_start_date_sk-ss_coupon_amt	0,00	0,51	0,07	0,00
p_channel_radio-ss_list_price	0,00	0,51	0,07	0,00
p_channel_press-ss_ext_discount_amt	0,00	0,46	0,05	0,06
p_cost-ss_net_apid	0,00	0,48	0,09	0,00
p_promo_id_b-ss_ext_wholesale_cost	0,00	0,42	0,14	0,00
p_channel_demo-ss_sold_time_sk	0,00	0,50	0,07	0,00
p_channel_dmail-ss_sold_date_sk	0,00	0,49	0,00	0,07
p_cost-ss_ext_sales_price	0,00	0,44	0,11	0,00
p_channel_email-ss_sold_time_sk	0,00	0,49	0,07	0,00
p_channel_event-ss_sold_date_sk	0,00	0,49	0,07	0,00
p_promo_sk-ss_net_apid	0,00	0,46	0,09	0,00
p_end_date_sk-ss_quantity	0,00	0,47	0,08	0,00
p_channel_demo-ss_ext_tax	0,00	0,48	0,07	0,00
p_channel_dmail-ss_sold_time_sk	0,00	0,55	0,00	0,00
p_cost-ss_addr_sk	0,00	0,34	0,20	0,00
p_promo_id_b-ss_ext_tax	0,00	0,46	0,08	0,00
p_end_date_sk-ss_net_profit	0,00	0,46	0,08	0,00
p_promo_id_b-ss_quantity	0,00	0,45	0,08	0,00
p_purpose-ss_ticket_number	0,00	0,40	0,13	0,00
p_channel_demo-ss_list_price	0,00	0,45	0,07	0,00
p_purpose-ss_net_apid	0,00	0,42	0,09	0,00
p_channel_catalog-ss_ext_list_price	0,00	0,51	0,00	0,00
p_channel_details-ss_promo_sk	0,00	0,45	0,06	0,00
p_channel_event-ss_list_price	0,00	0,44	0,07	0,00
p_channel_dmail-ss_list_price	0,00	0,44	0,07	0,00
p_channel_email-ss_list_price	0,00	0,44	0,07	0,00
p_purpose-ss_ext_tax	0,00	0,40	0,10	0,00

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
p_purpose-ss_ext_discount_amt	0,00	0,38	0,11	0,00
p_channel_email-ss_sold_date_sk	0,00	0,42	0,07	0,00
p_channel_event-ss_sold_time_sk	0,00	0,42	0,07	0,00
p_channel_radio-ss_sold_date_sk	0,00	0,49	0,00	0,00
p_channel_radio-ss_sold_time_sk	0,00	0,49	0,00	0,00
p_promo_sk-ss_quantity	0,00	0,40	0,09	0,00
p_channel_tv-ss_promo_sk	0,00	0,40	0,08	0,00
p_cost-ss_sales_price	0,00	0,33	0,14	0,00
p_channel_catalog-ss_sold_time_sk	0,00	0,41	0,06	0,00
p_promo_sk-ss_ext_tax	0,00	0,47	0,00	0,00
p_end_date_sk-ss_list_price	0,00	0,46	0,00	0,00
p_channel_event-ss_promo_sk	0,00	0,38	0,07	0,00
p_channel_dmail-ss_promo_sk	0,00	0,38	0,07	0,00
p_channel_email-ss_promo_sk	0,00	0,38	0,07	0,00
p_channel_radio-ss_promo_sk	0,00	0,38	0,07	0,00
p_channel_catalog-ss_promo_sk	0,00	0,37	0,06	0,00
p_promo_name-ss_quantity	0,00	0,40	0,00	0,00

Apêndice 2 – Resultados Cenário ASR

A Tabela 19 apresenta os valores da similaridade associados às medidas de *Jaccard*, *Cosine*, *Levenshtein* e *Jaro-Winkler* para o Cenário **ASR**.

Tabela 19. Valores da Similaridade do Cenário ASR.

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
ib_income_band_sk-ss_sold_time_sk	0,08	0,38	0,65	0,35
ib_income_band_sk-ss_sold_date_sk	0,04	0,31	0,67	0,41
ib_income_band_sk-ss_customer_sk	0,08	0,28	0,68	0,35
ib_income_band_sk-ss_store_sk	0,04	0,29	0,63	0,35
ib_income_band_sk-ss_item_sk	0,05	0,25	0,61	0,35
ib_income_band_sk-ss_promo_sk	0,04	0,24	0,63	0,35
ib_income_band_sk-ss_cdemo_sk	0,04	0,16	0,69	0,35
ib_income_band_sk-ss_hdemo_sk	0,04	0,16	0,64	0,29
ib_income_band_sk-ss_addr_sk	0,05	0,17	0,55	0,29
ib_income_band_sk-ss_net_paid_inc_tax	0,07	0,24	0,52	0,16
ib_lower_bound-ss_customer_sk	0,04	0,15	0,57	0,14
ib_income_band_sk-ss_wholesale_cost	0	0,12	0,51	0,24
ib_lower_bound-ss_ext_discount_amt	0,04	0,12	0,5	0,21
ib_upper_bound-ss_ext_discount_amt	0,04	0,12	0,5	0,21
ib_lower_bound-ss_net_apid	0	0	0,55	0,29
ib_upper_bound-ss_customer_sk	0,04	0,15	0,51	0,14
ib_upper_bound-ss_promo_sk	0	0	0,54	0,29
ib_upper_bound-ss_net_apid	0	0	0,54	0,29
ib_income_band_sk-ss_net_apid	0	0	0,59	0,24
ib_lower_bound-ss_wholesale_cost	0	0	0,59	0,24
ib_income_band_sk-ss_coupon_amt	0	0,07	0,51	0,24

Pares de Atributos	*Jaccard*	*Jaro-Winkler*	*Levenshtein*	*Cosine*
ib_upper_bound-ss_coupon_amt	0	0,16	0,52	0,14
ib_lower_bound-ss_ticket_number	0	0,07	0,55	0,19
ib_upper_bound-ss_ticket_number	0	0,07	0,55	0,19
ib_income_band_sk-ss_ext_discount_amt	0	0,06	0,59	0,16
ib_lower_bound-ss_list_price	0	0,08	0,52	0,21
ib_lower_bound-ss_addr_sk	0	0,09	0,5	0,21
ib_upper_bound-ss_addr_sk	0	0,09	0,5	0,21
ib_income_band_sk-ss_ext_wholesale_cost	0	0,11	0,44	0,24
ib_lower_bound-ss_coupon_amt	0	0,08	0,57	0,14
ib_lower_bound-ss_store_sk	0	0	0,54	0,21
ib_income_band_sk-ss_ext_tax	0	0	0,55	0,18
ib_upper_bound-ss_net_profit	0	0	0,52	0,21
ib_lower_bound-ss_net_profit	0	0	0,51	0,21
ib_income_band_sk-ss_ticket_number	0	0	0,53	0,18
ib_income_band_sk-ss_quantity	0	0,08	0,45	0,18
ib_lower_bound-ss_sales_price	0	0	0,5	0,21
ib_lower_bound-ss_ext_wholesale_cost	0	0	0,52	0,19
ib_lower_bound-ss_item_sk	0	0	0,48	0,21
ib_upper_bound-ss_item_sk	0	0	0,48	0,21
ib_upper_bound-ss_store_sk	0	0	0,54	0,14
ib_lower_bound-ss_cdemo_sk	0	0	0,47	0,21
ib_lower_bound-ss_hdemo_sk	0	0	0,47	0,21
ib_lower_bound-ss_promo_sk	0	0	0,47	0,21
ib_upper_bound-ss_cdemo_sk	0	0	0,47	0,21
ib_upper_bound-ss_hdemo_sk	0	0	0,47	0,21
ib_income_band_sk-ss_net_profit	0	0	0,49	0,18
ib_income_band_sk-ss_sales_price	0	0,06	0,48	0,12
ib_upper_bound-ss_list_price	0	0	0,52	0,14
ib_upper_bound-ss_sales_price	0	0	0,44	0,21
ib_lower_bound-ss_ext_tax	0	0	0,5	0,14
ib_upper_bound-ss_ext_tax	0	0	0,5	0,14
ib_income_band_sk-ss_ext_sales_price	0	0,06	0,51	0,06
ib_lower_bound-ss_sold_date_sk	0	0	0,5	0,13
ib_upper_bound-ss_ext_wholesale_cost	0	0	0,49	0,14
ib_upper_bound-ss_net_paid_inc_tax	0	0	0,47	0,16
ib_lower_bound-ss_ext_list_price	0	0,07	0,42	0,12
ib_upper_bound-ss_wholesale_cost	0	0	0,42	0,18
ib_lower_bound-ss_ext_sales_price	0	0	0,41	0,17
ib_lower_bound-ss_net_paid_inc_tax	0	0	0,47	0,11
ib_upper_bound-ss_ext_sales_price	0	0	0,41	0,17
ib_upper_bound-ss_sold_date_sk	0	0	0,5	0,07
ib_upper_bound-ss_quantity	0	0	0,5	0,07
ib_income_band_sk-ss_list_price	0	0	0,44	0,12
ib_lower_bound-ss_sold_time_sk	0	0	0,43	0,13
ib_upper_bound-ss_ext_list_price	0	0	0,42	0,12
ib_income_band_sk-ss_ext_list_price	0	0	0,46	0,06
ib_lower_bound-ss_quantity	0	0	0,44	0,07

Universidade do Minho
Escola de Engenharia

José Fernando Pereira Magalhães

Abordagem Semântica para a Integração de Dados em Big Data Warehouses

outubro de 2019

Abordagem Semântica para a Integração de dados em Big Data Warehouses

José Fernando Pereira Magalhães

UMinho | 2019