

**UNIVERSIDADE DE SÃO PAULO**

Instituto de Ciências Matemáticas e de Computação

**Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens em ambientes paralelos e distribuídos**

**Guilherme Muzzi da Rocha**

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C<sup>2</sup>MC)



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

**Guilherme Muzzi da Rocha**

Processamento eficiente de consultas analíticas estendidas  
com predicado de similaridade sobre um *data warehouse* de  
imagens em ambientes paralelos e distribuídos

Dissertação apresentada ao Instituto de Ciências  
Matemáticas e de Computação – ICMC-USP,  
como parte dos requisitos para obtenção do título  
de Mestre em Ciências – Ciências de Computação e  
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e  
Matemática Computacional

Orientadora: Profa. Dra. Cristina Dutra de  
Aguiar Ciferri

**USP – São Carlos**  
**Outubro de 2020**



**Guilherme Muzzi da Rocha**

Efficient processing analytical queries extended with  
similarity search predicate over an image data warehouse in  
parallel and distributed environments

Dissertation submitted to the Institute of Mathematics  
and Computer Sciences – ICMC-USP – in  
accordance with the requirements of the Computer  
and Mathematical Sciences Graduate Program, for  
the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and  
Computational Mathematics

Advisor: Profa. Dra. Cristina Dutra de Aguiar Ciferri

**USP – São Carlos**  
**October 2020**



*Este trabalho é dedicado aos professores do Brasil,  
os quais acreditam no estudo para o desenvolvimento do país e  
são a base de conhecimento da sociedade e deste trabalho.*





# AGRADECIMENTOS

---

---

Agradeço à Deus por me capacitar e abençoar no desenvolvimento desta dissertação de mestrado.

Agradeço à minha amada namorada Ana Ellen Mattos. Por me apoiar e incentivar deste a ideia de fazer um mestrado, normalmente desvalorizado pela sociedade. Por ser, diariamente, meu porto seguro em todo este processo, tanto com ligações ao final do dia quanto finais de semana juntos. Por me dar todo o suporte emocional, desde madrugadas de trabalho e estresse até conquistas alcançadas. Eu te amo muito meu amor, obrigado por caminhar ao meu lado nesta fase tão especial para nossa história.

Agradeço aos meus amados pais, Geofredo Borges da Rocha e Fabiana Augusta Garcia Muzzi, pelo apoio incondicional de sempre, por tudo que fizeram e fazem por mim. Amo muito vocês, sou muito grato e feliz por ter ao meu lado pais incríveis, que me apoiam em todas as áreas da minha vida. Agradeço também ao meu amado irmão Gabriel Muzzi da Rocha, por ser um verdadeiro irmão nesta jornada, em uma etapa tão difícil para nós. Agradeço pelo apoio sensacional, pelas longas conversas por telefone, dividindo conquistas, experiências, angústias, preocupações e nos apoiando, entendendo um ao outro. Eu te amo muito, Tibe.

Agradeço aos irmãos que a vida me deu, Bruno, Matheus, Tiba e Vinícius. Por todas orações, apoio e torcida nesta etapa que não nos vimos muito, mas que sabíamos que estávamos ali para o que precisar. Contem comigo sempre, amo vocês.

Agradeço toda minha família, pelas orações e torcida por mim. Em especial, ao meu tio Fernando Augusto Garcia Muzzi, por ter sido minha inspiração para ingressar na área da computação e área acadêmica.

Agradeço à minha orientadora Cristina Dutra de Aguiar Ciferri, por ser muito mais do que uma excelente orientadora. Agradeço por acreditar no meu potencial, pela orientação incrível, pelo olhar humano para lidar com cada um, pelas conversas, conselhos e apoio em questões a parte da pesquisa. Evolui muito neste mestrado, profissionalmente e pessoalmente, serei eternamente grato por tudo. Estendo o agradecimento ao grupo de pesquisa, em especial, ao João Pedro e Piero, pelas colaborações importantes.

Agradeço aos funcionários, professores e ao próprio instituto ICMC/USP. Meu carinho e gratidão à USP por toda evolução, desde a graduação, estarão sempre presentes. Por fim, agradeço a CAPES e a FAPESP (processo 2018/10607-3) pelo apoio financeiro, o qual foi imprescindível para o desenvolvimento deste trabalho.



*“A mente que se abre a uma nova ideia  
jamais voltará ao seu tamanho original.”  
(Albert Einstein)*



# RESUMO

ROCHA, G. M. **Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens em ambientes paralelos e distribuídos**. 2020. 138 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Consultas analíticas sobre *data warehouses* convencionais possuem alto custo computacional, desde que requerem a realização de caras operações de junção-estrela considerando grandes volumes de dados. Este custo é ainda maior quando se consideram *data warehouses* de imagens. Além deles serem mais volumosos, as consultas analíticas são estendidas com predicado de similaridade de imagens, o qual também requer a realização de onerosas operações de cálculos de distância. Nesta dissertação de mestrado, investiga-se esse desafio. Considera-se o contexto de imagens médicas, a sua importância para a tomada de decisão analítica e seu impacto para a sociedade. Neste contexto, as aplicações requerem o compartilhamento e o gerenciamento de grandes volumes de dados, sendo necessário o uso de um *framework* de processamento paralelo e distribuído. Nesta dissertação, são propostos quatro métodos para o processamento eficiente de consultas analíticas estendidas com predicado de similaridade de imagens em Spark. O método *BloOmniImg* integra a técnica *bloom filter cascade join* para o processamento da junção-estrela com a técnica *Omni* para o processamento das operações de distância. Por ser baseado em uma estrutura de dados probabilística, ele é indicado para sistemas computacionais com pouca memória primária disponível. O método *BrOmniImg* também usa a técnica *Omni*, porém a integra com a técnica *broadcast join*. Por ser baseado em uma estrutura de dados exata, ele é indicado para sistemas computacionais cuja memória primária disponível é suficiente para armazenar essa estrutura. O método *BrOmniImg<sup>CF</sup>* estende o método *BrOmniImg* de forma a utilizar também o predicado convencional das consultas para reduzir ainda mais o número de operações de cálculos de distância entre as imagens. Por fim, o método *SimSparkOLAP* processa eficientemente consultas analíticas estendidas com predicado de similaridade de imagens, geográficos e socioeconômicos. Os métodos propostos foram validados por meio de testes de desempenho experimentais. Os resultados demonstraram que os métodos proveram ganho de desempenho de até 66,93% frente aos trabalhos relacionados existentes na literatura. Adicionalmente, os métodos propostos foram usados para investigar o potencial semântico do processamento destas consultas analíticas, visando mostrar como os resultados obtidos nesta dissertação de mestrado podem ser usados para enriquecer o suporte à tomada de decisão da área médica.

**Palavras-chave:** *Data warehouse* de imagens, consultas OLAP estendidas com predicado de similaridade, processamento paralelo e distribuído, imagens médicas.



# ABSTRACT

ROCHA, G. M. **Efficient processing analytical queries extended with similarity search predicate over an image data warehouse in parallel and distributed environments**. 2020. 138 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Analytical queries over conventional data warehouses have a high computational cost, as they require processing the expensive star join operation considering huge volumes of data. This cost is even greater in image data warehousing environments. First, image data warehouses are more voluminous. Second, analytical queries are extended with similarity search predicate, also requiring the processing of costly operations to calculate the distance between images. In this master thesis, we focus on this challenge. We consider the context of medical images, due to the importance of the analytical decision-making over them and their impact on the society. In this context, medical applications usually share and manage large volumes of data. Thus, they require the use of a parallel and distributed processing framework. We propose four methods for the efficient processing of analytical queries extended with image similarity search predicate in Spark. The first method is *BloOmnImg*, which is based on the integration of the bloom filter cascade join and the Omni techniques for processing star joins and distance calculations, respectively. *BloOmnImg* uses a probabilistic structure. Then, it is indicated for computational systems with low primary memory. The second method is *BrOmnImg*, which also uses the Omni technique, but integrate it with the broadcast join technique. *BrOmnImg* is based on a exact structure, then it is indicated for computational systems with primary memory enough to store this structure. The *BrOmnImg<sup>CF</sup>* method extends *BrOmnImg* by using the conventional predicate to further reduce the distance calculations between images. Finally, *SimSparkOLAP* extends *BrOmnImg* for the efficient processing analytical queries extended with image, geographical and socioeconomic predicates. The proposed methods were validated through performance tests considering different data volumes and configurations for the predicates involved. The results showed the significant performance gain of the methods over related work. The validation of the methods also considered examples of semantic queries and identified their importance to enrich the medical decision-making process.

**Keywords:** Image data warehouse, OLAP queries extended with similarity search predicate, parallel and distributed processing, medical images.





# LISTA DE ILUSTRAÇÕES

---

---

Figura 1 – Hipercubo de Dados. Fonte: Adaptado de (CIFERRI <i>et al.</i> , 2013). . . . .	39
Figura 2 – Exemplo de esquema-estrela de um ambiente de <i>data warehousing</i> da área médica. Fonte: Adaptado de (TEIXEIRA, 2015). . . . .	42
Figura 3 – Exemplo de esquema floco de neve de um ambiente de <i>data warehousing</i> da área médica. Fonte: Elaborado pelo autor. . . . .	43
Figura 4 – Exemplos dos espaços de equidistâncias gerados pela distância $L_p$ . (a) Distância Manhattan ( $L_1$ ), (b) Distância Euclidiana ( $L_2$ ) e (c) Distância Chebychev ( $L_\infty$ ). Fonte: (TEIXEIRA, 2015). . . . .	48
Figura 5 – Exemplo de consulta por abrangência e k-vizinhos mais próximos. Fonte: Adaptado de (ANNIBAL, 2011). . . . .	50
Figura 6 – Exemplo de <i>mbOr</i> (área cinza) para um centro de consulta $s_q$ de raio $r_q$ com elementos representativos $f_1$ e $f_2$ . Fonte: Adaptado de (TRAINA <i>et al.</i> , 2007)	51
Figura 7 – Processo ETL de imagens. Fonte: Adaptado de (TEIXEIRA <i>et al.</i> , 2015). . . . .	52
Figura 8 – Esquema-estrela DW de imagens. Fonte: (TEIXEIRA, 2015). . . . .	53
Figura 9 – Quatro diferentes esquema-estrela para um DW de imagens médicas. (a) Esquema EBM1: armazenamento dos vetores de características e dos atributos para pesquisa por similaridade de cada camada perceptual conjuntamente em uma mesma tabela de dimensão. (b) Esquema EBM2: armazenamento de cada vetor de características em uma tabela de dimensão separada por camada perceptual e armazenamento dos atributos para pesquisa por similaridade em uma outra tabela de dimensão separada por camada perceptual. (c) Esquema EBM3: armazenamento de todos os vetores de características em uma única tabela de dimensão e armazenamento dos atributos para pesquisa por similaridade em tabelas de dimensão separadas por camada perceptual. (d) Esquema EBM4: armazenamento dos vetores de características como fatos na tabela de fatos e armazenamento dos atributos para pesquisa por similaridade em tabelas de dimensão separadas por camada perceptual. Fonte: Adaptado de (ROCHA; CIFERRI, 2018). . . . .	54
Figura 10 – Processamento de consulta analítica estendida com predicado de similaridade. Fonte: Adaptado de (TEIXEIRA <i>et al.</i> , 2015). . . . .	55
Figura 11 – Arquitetura HDFS. Fonte: Adaptado de (BRITO, 2017). . . . .	59
Figura 12 – Exemplo de aplicação de função <i>map</i> e <i>reduce</i> . Fonte: Elaborado pelo autor.	60

Figura 13 – Exemplo de inserção e consulta na estrutura <i>bloom filter</i> . Fonte: Adaptado de (PATGIRI <i>et al.</i> , 2019) . . . . .	63
Figura 14 – Técnica <i>bloom filter cascade join</i> para o processamento de junção. Fonte: Elaborado pelo autor. . . . .	64
Figura 15 – Técnica <i>broadcast join</i> para o processamento de junção. Fonte: Elaborado pelo autor. . . . .	64
Figura 16 – Processo de revisão sistemática. Fonte: (NAKAGAWA <i>et al.</i> , 2017). . . . .	68
Figura 17 – Fluxo da fase de Condução. Fonte: Elaborado pelo autor. . . . .	75
Figura 18 – Condução da revisão sistemática. Fonte: Elaborado pelo autor. . . . .	75
Figura 19 – Exemplo de arquitetura Hadoop para DW da área médica. Fonte: Adaptado de (RAJA; SIVASANKAR, 2014). . . . .	79
Figura 20 – Esquema-estrela do DW de imagens utilizado como base para a descrição dos métodos propostos, chamado de DWbase. Fonte: Elaborado pelo autor. . . . .	84
Figura 21 – Visão geral do método <i>BloOmnImg</i> . Fonte: Elaborado pelo autor. . . . .	86
Figura 22 – Testes de desempenho considerando a variação do erro da estrutura <i>bloom filter</i> , utilizada no método <i>BloOmnImg</i> , de acordo com a Tabela 5. Fonte: Elaborado pelo autor. . . . .	91
Figura 23 – Testes de desempenho comparando o método proposto <i>BloOmnImg</i> com o método concorrente SBFCJ, variando a seletividade do predicado de similaridade. Fonte: Elaborado pelo autor. . . . .	92
Figura 24 – Número de cálculos de distância realizadas no processamento do predicado de similaridade para os métodos <i>BloOmnImg</i> e SBFCJ, além do resultado deste predicado. Fonte: Elaborado pelo autor. . . . .	92
Figura 25 – Testes de desempenho comparando <i>BloOmnImg</i> com SBFCJ, considerando a variação da seletividade do predicado convencional e dimensionalidade das camadas perceptuais envolvidas de acordo com a Tabela 6. Fonte: Elaborado pelo autor. . . . .	93
Figura 26 – Exemplos sintéticos de consultas semânticas com agrupamentos providas por <i>BloOmnImg</i> . Fonte: Elaborado pelo autor. . . . .	95
Figura 27 – Visão geral do método <i>BrOmnImg</i> . Fonte: Elaborado pelo autor. . . . .	98
Figura 28 – Visão geral do método <i>BrOmnImg</i> <sup>CF</sup> . Fonte: Elaborado pelo autor. . . . .	100
Figura 29 – Testes de desempenho comparando o método proposto <i>BrOmnImg</i> com o método concorrente SBJ, variando a seletividade do predicado de similaridade. Fonte: Elaborado pelo autor. . . . .	105
Figura 30 – Testes de desempenho comparando o método proposto <i>BrOmnImg</i> com o método concorrente SBJ, para o Volume 2 de dados, variando a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor. . . . .	106

Figura 31 – Testes de desempenho comparando o método proposto <i>BrOmnImg</i> com o método concorrente SBJ, para o Volume 1 de dados, variando a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor. . . . .	106
Figura 32 – Testes de desempenho comparando o método proposto <i>BrOmnImg<sup>CF</sup></i> com o método proposto <i>BrOmnImg</i> , para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 50% (C4) de seletividade do predicado convencional. Fonte: Elaborado pelo autor. . . . .	107
Figura 33 – Testes de desempenho comparando o método proposto <i>BrOmnImg<sup>CF</sup></i> com o método proposto <i>BrOmnImg</i> , para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 33% (C7) de seletividade do predicado convencional. Fonte: Elaborado pelo autor. . . . .	108
Figura 34 – Testes de desempenho comparando o método proposto <i>BrOmnImg<sup>CF</sup></i> com o método proposto <i>BrOmnImg</i> , para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 0,08% (C10) de seletividade do predicado convencional. Fonte: Elaborado pelo autor. . . . .	108
Figura 35 – Porcentagem dos ganhos de performance para os testes de desempenho comparando <i>BrOmnImg<sup>CF</sup></i> com <i>BrOmnImg</i> , variando a seletividade do predicado convencional e de similaridade. Fonte: Elaborado pelo autor. . . . .	109
Figura 36 – Dados de imagens, geográficos e socioeconômicos como fatores de similaridade. Fonte: Elaborado pelo autor. . . . .	112
Figura 37 – Três diferentes esquemas-estrela para um DW com fatores de similaridade de imagens, geográficos e socioeconômicos: <i>jointed</i> , <i>split</i> e <i>normalized</i> . Fonte: Elaborado pelo autor. . . . .	114
Figura 38 – Visão geral do método <i>SimSparkOLAP</i> . Fonte: Elaborado pelo autor. . . . .	116
Figura 39 – Testes experimentais de <i>SimSparkOLAP</i> focado na performance das consultas, considerando os métodos propostos. Fonte: Elaborado pelo autor. . . . .	119
Figura 40 – Exemplos sintéticos de consultas semânticas providas por <i>SimSparkOLAP</i> . Fonte: Elaborado pelo autor. . . . .	120



# LISTA DE ALGORITMOS

---

---

Algoritmo 1 – <i>BloOmnImg</i> . . . . .	87
Algoritmo 2 – <i>BrOmnImg</i> . . . . .	99
Algoritmo 3 – <i>BrOmnImg</i> <sup>CF</sup> . . . . .	102



# LISTA DE TABELAS

---

---

Tabela 1 – Áreas de definições e construções dos conceitos que envolvem consulta por similaridade. Fonte: Elaborado pelo autor. . . . .	46
Tabela 2 – Conceitos dos temas de pesquisa abordados na revisão sistemática. Fonte: Elaborado pelo autor. . . . .	81
Tabela 3 – Volume de dados do DW de imagens projetado para os experimentos de <i>BloOmnImg</i> . Fonte: Elaborado pelo autor. . . . .	88
Tabela 4 – Características de cada camada perceptual considerada nos testes de <i>BloOmnImg</i> . Fonte: Elaborado pelo autor. . . . .	88
Tabela 5 – Configurações definidas para os testes de desempenho de <i>BloOmnImg</i> , os quais variam o erro $p$ da estrutura <i>bloom filter</i> (Equação 4.1). Fonte: Elaborado pelo autor. . . . .	89
Tabela 6 – Configurações definidas para os testes de desempenho de <i>BloOmnImg</i> , as quais variam as camadas perceptuais e a seletividade do predicado convencional. Fonte: Elaborado pelo autor. . . . .	89
Tabela 7 – Configurações definidas para uma análise de semântica das consultas nos casos de teste de <i>BloOmnImg</i> , as quais variam o agrupamento final dos dados. Fonte: Elaborado pelo autor. . . . .	90
Tabela 8 – Número de imagens retornadas em cada configuração definida na Tabela 6. Fonte: Elaborado pelo autor. . . . .	93
Tabela 9 – Configurações definidas para os testes de desempenho de <i>BrOmnImg</i> , as quais variam a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor. . . . .	103
Tabela 10 – Configurações definidas para os testes de desempenho de <i>BrOmnImg<sup>CF</sup></i> , as quais variam a seletividade do predicado convencional e de similaridade. Fonte: Elaborado pelo autor. . . . .	104
Tabela 11 – Número de imagens retornadas para as configurações definidas para validação do método <i>BrOmnImg<sup>CF</sup></i> (Tabela 10). Autor: Elaborado pelo autor. . . . .	107
Tabela 12 – Volume de dados para a realização dos experimentos de consultas analíticas estendidas com predicado de fatores de similaridade. Fonte: Elaborado pelo autor. . . . .	117
Tabela 13 – Configurações para a realização de experimentos, com foco em desempenho, de consultas analíticas estendidas com predicado de fatores de similaridade de dados geográficos e socioeconômicos. Fonte: Elaborado pelo autor. . . . .	118





# LISTA DE ABREVIATURAS E SIGLAS

---

---

BI	<i>Business Intelligence</i>
CBIR	<i>Content-Based Image Retrieval</i>
CBMIR	<i>Content-Based Medical Image Retrieval</i>
dBJI	<i>distributed Bitmap Join Index</i>
DW	<i>Data Warehouse</i>
ETL	<i>Extract, Transform, Load</i>
HDFS	<i>Hadoop Distributed File System</i>
HF	<i>Hull of Foci</i>
HOLAP	<i>Hybrid OLAP</i>
K-nn	<i>K-nearest neighbor</i>
LSH	<i>Locality Sensitive Hashing</i>
MAM	<i>Método de Acesso Métrico</i>
mbOr	<i>minimum-bounding-Omni-region</i>
MOLAP	<i>Multidimensional OLAP</i>
OLAP	<i>On-line Analytical Processing</i>
OLTP	<i>On-line Transaction Processing</i>
RDD	<i>Dados Distribuídos e Resilientes</i>
ROLAP	<i>Relacional OLAP</i>
SBFCJ	<i>Spark Bloom Filter Cascade Join</i>
SBJ	<i>Spark Broadcast Join</i>



# SUMÁRIO

---

---

1	<b>INTRODUÇÃO</b>	29
1.1	Considerações iniciais	29
1.2	Contextualização	29
1.3	Motivação	31
1.4	Objetivo	32
1.5	Estrutura da dissertação	34
2	<b>DATA WAREHOUSING</b>	37
2.1	Considerações iniciais	37
2.2	Contextualização	37
2.3	Modelagem em nível conceitual	38
2.3.1	<i>Aspectos estáticos</i>	38
2.3.2	<i>Aspectos dinâmicos</i>	39
2.4	Modelagem em nível lógico	41
2.4.1	<i>Esquema-estrela</i>	41
2.4.2	<i>Esquema floco de neve</i>	41
2.5	Considerações finais	43
3	<b>DATA WAREHOUSING DE IMAGENS</b>	45
3.1	Considerações iniciais	45
3.2	Contextualização	45
3.3	Consulta por similaridade	46
3.3.1	<i>Extração de características</i>	47
3.3.2	<i>Funções de distância</i>	48
3.3.3	<i>Espaço métrico</i>	49
3.3.3.1	<i>Operadores de similaridade</i>	49
3.3.3.2	<i>Técnica OMNI</i>	50
3.4	Arquitetura do ambiente de <i>data warehousing</i> de imagens <i>ImageDWE</i>	51
3.5	Considerações finais	53
4	<b>PROCESSAMENTO PARALELO E DISTRIBUÍDO</b>	57
4.1	Considerações iniciais	57
4.2	Contextualização	57
4.3	HDFS	58

4.4	<b>Framework MapReduce</b>	60
4.5	<b>Framework Spark</b>	61
4.6	<b>Bloom filter cascade join</b>	62
4.7	<b>Broadcast join</b>	63
4.8	<b>Considerações finais</b>	64
<b>5</b>	<b>REVISÃO SISTEMÁTICA</b>	<b>67</b>
5.1	<b>Considerações iniciais</b>	67
5.2	<b>Contextualização</b>	67
5.3	<b>Planejamento</b>	68
5.3.1	<b>Objetivos</b>	68
5.3.2	<b>Questões de pesquisa</b>	69
5.3.3	<b>Identificação dos estudos</b>	69
5.3.3.1	<i>Fontes de busca</i>	70
5.3.3.2	<i>Definições prévias</i>	70
5.3.3.3	<i>Palavra-Chave</i>	70
5.3.3.4	<i>Strings de busca</i>	71
5.3.4	<b>Crítérios de seleção</b>	73
5.3.4.1	<i>Crítérios de inclusão</i>	73
5.3.4.2	<i>Crítérios de exclusão</i>	73
5.3.4.3	<i>Procedimento de seleção</i>	74
5.4	<b>Condução</b>	74
5.4.1	<b>Seleção dos estudos</b>	74
5.4.2	<b>Síntese dos dados</b>	75
5.4.2.1	<i>Junção-estrela em Hadoop</i>	76
5.4.2.2	<i>Similaridade de imagens em Hadoop</i>	77
5.4.2.3	<i>Arquitetura de um ambiente de data warehousing da área médica processado em Hadoop</i>	78
5.4.2.4	<i>Operações de junção-estrela e de similaridade em um ambiente de data warehousing da área médica em Hadoop</i>	80
5.5	<b>Considerações finais</b>	81
<b>6</b>	<b>MÉTODO <i>BloOmnImg</i></b>	<b>83</b>
6.1	<b>Considerações iniciais</b>	83
6.2	<b>Características do DW de imagens e consultas de interesse</b>	83
6.3	<b>Descrição do método <i>BloOmnImg</i></b>	84
6.3.1	<b>Método</b>	85
6.3.2	<b>Algoritmo</b>	86
6.4	<b>Validação</b>	87
6.4.1	<b>Ambiente de teste</b>	88

6.4.2	<i>Resultados</i> . . . . .	90
6.5	Considerações finais . . . . .	94
7	<b>MÉTODOS</b> <i>BrOmnImg</i> E <i>BrOmnImg<sup>CF</sup></i> . . . . .	97
7.1	Considerações iniciais . . . . .	97
7.2	Descrição do método <i>BrOmnImg</i> . . . . .	97
7.2.1	<i>Método</i> . . . . .	97
7.2.2	<i>Algoritmo</i> . . . . .	98
7.3	Descrição do método <i>BrOmnImg<sup>CF</sup></i> . . . . .	99
7.3.1	<i>Método</i> . . . . .	100
7.3.2	<i>Algoritmo</i> . . . . .	101
7.4	Validação . . . . .	102
7.4.1	<i>Ambiente de teste</i> . . . . .	102
7.4.2	<i>Resultados de BrOmnImg</i> . . . . .	104
7.4.3	<i>Resultados de BrOmnImg<sup>CF</sup></i> . . . . .	106
7.5	Considerações finais . . . . .	109
8	<b>MÉTODO</b> <i>SimSparkOLAP</i> . . . . .	111
8.1	Considerações iniciais . . . . .	111
8.2	Motivação . . . . .	111
8.3	Descrição do método <i>SimSparkOLAP</i> . . . . .	112
8.3.1	<i>Esquemas</i> . . . . .	113
8.3.2	<i>Método</i> . . . . .	114
8.4	Validação . . . . .	116
8.4.1	<i>Ambiente de teste</i> . . . . .	116
8.4.2	<i>Resultados</i> . . . . .	118
8.5	Considerações finais . . . . .	120
9	<b>CONCLUSÃO</b> . . . . .	123
9.1	Considerações iniciais . . . . .	123
9.2	Contribuições . . . . .	123
9.2.1	<i>Método BloOmnImg</i> . . . . .	124
9.2.2	<i>Método BrOmnImg</i> . . . . .	125
9.2.3	<i>Método BrOmnImg<sup>CF</sup></i> . . . . .	125
9.2.4	<i>Método SimSparkOLAP</i> . . . . .	126
9.3	Divulgação de resultados . . . . .	126
9.4	Dificuldades encontradas . . . . .	127
9.5	Trabalhos futuros . . . . .	128
	<b>REFERÊNCIAS</b> . . . . .	131



---

# INTRODUÇÃO

---

## 1.1 Considerações iniciais

Neste capítulo é descrita a introdução desta dissertação de mestrado, cujo objetivo vislumbra o processamento de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens, considerando ambientes paralelos e distribuídos. Do ponto de vista de imagens, embora o trabalho desenvolvido tenha aplicações em diversos contextos, ele tem como motivação a manipulação de imagens médicas. Na seção 1.2 é feita a contextualização da dissertação. Na seção 1.3 é discutida a motivação para o desenvolvimento da dissertação, ou seja, é salientada a lacuna existente na literatura. Na seção 1.4 são detalhados os objetivos da dissertação, a hipótese que permeia a mesma e as contribuições. Por fim, na seção 1.5 é detalhada a estrutura desta dissertação.

## 1.2 Contextualização

Com o desenvolvimento de grandes organizações, o foco de interesse das aplicações de banco de dados mudou. Ao invés do armazenamento dos dados, essas organizações estão interessadas em obter informações a partir desses. Ou seja, elas estão interessadas em fornecer, a partir da análise dos dados, suporte à tomada de decisão considerando o viés gerencial. No final dos anos 80, ambientes de *data warehousing* surgiram para atender esta necessidade (CHAUDHURI; DAYAL, 1997; MOHANIA *et al.*, 1999; VAISMAN; ZIMÁNYI, 2014).

Um ambiente de *data warehousing* consiste de uma série de tecnologias, ferramentas e técnicas voltadas à extração dos dados de provedores autônomos, heterogêneos e distribuídos, ao armazenamento desses dados no *Data Warehouse* (DW) e à realização de consultas analíticas executadas sobre o DW (CIFERRI, 2002). O DW consiste no componente mais importante do ambiente, sendo caracterizado por ser um banco de dados que armazena dados integrados,

não-voláteis, orientados a assunto e históricos (KIMBALL; ROSS, 2002). Todo este processo é executado em duas etapas:

- Etapa *Extract, Transform, Load* (ETL), na qual dados dos provedores são extraídos, traduzidos, filtrados (ou seja, passam por um processo de limpeza), integrados e finalmente armazenados no DW.
- Etapa de consulta, nas quais consultas analíticas, denominadas consultas *On-line Analytical Processing* (OLAP), são realizadas em um DW para oferecer ao usuário informações que deem suporte à tomada de decisão (ARIYACHANDRA; WATSON, 2010; CHAUDHURI; DAYAL; NARASAYYA, 2011; GOLFARELLI; RIZZI, 2009; GONZALES *et al.*, 2011).

Em implementações relacionais do DW, seus dados são armazenados segundo o esquema-estrela, o qual é composto por uma tabela de fatos que se relaciona com várias tabelas de dimensão. Uma característica das consultas OLAP é que elas são executadas utilizando como base este esquema-estrela, ou seja, elas são caracterizadas por realizarem diversas junções entre a tabela de fatos e cada uma das tabelas de dimensão envolvidas na consulta, bem como requererem a resolução de condições de seleção e de agrupamentos. Essas junções são denominadas de junção-estrela (BRITO *et al.*, 2016).

Um DW convencional armazena apenas dados convencionais, como dados do tipo numérico, alfanumérico e data. Dados convencionais possuem relação de ordem total, ou seja, eles podem ser comparados por meio de operações de igualdade, desigualdade e de ordem (maior, menor), por exemplo. Como resultado, ambientes de *data warehousing* convencionais oferecem suporte para as etapas de ETL e OLAP com base somente em dados convencionais. Por exemplo, em uma aplicação da área médica, é possível realizar a seguinte consulta: “*Quantos pacientes, com idade maior do que 40 anos, tiveram diagnóstico de câncer pulmonar no estado de São Paulo nos últimos 3 anos?*”.

Um ambiente de *data warehousing* de imagens estende o ambiente de *data warehousing* convencional para também manipular imagens. No ambiente de *data warehousing* de imagens *ImageDWE*, usado como base nesta dissertação de mestrado, as imagens são representadas por meio de suas características intrínsecas, ou seja, por meio de seus vetores de características e atributos para pesquisa por similaridade (TEIXEIRA *et al.*, 2015). Como resultado, (i) a etapa de ETL é estendida para também realizar a extração das características intrínsecas das imagens; (ii) o DW de imagens estende o DW convencional para armazenar essas características em tabelas de fatos e tabelas de dimensão; e (iii) a etapa de consulta é estendida para oferecer suporte para consultas analíticas estendidas com predicado de similaridade de imagens.

Ambientes de *data warehousing* de imagens enriquecem a capacidade das organizações com relação à tomada de decisões, desde que permitem que uma nova gama de consultas analíticas possa ser realizada. Essa nova gama de consultas integra predicado convencional com



predicado de similaridade de imagens. Por exemplo, pode-se realizar a seguinte consulta em uma aplicação da área médica: “*Quantas imagens de câncer pulmonar, similares à determinada imagem, pertencem a pacientes com idade maior do que 40 anos no estado de São Paulo nos últimos 3 anos?*”.

## 1.3 Motivação

O contexto de um ambiente de *data warehousing* convencional faz com que o DW seja muito volumoso (CIFERRI, 2002). Isso está relacionado ao fato de que diversos provedores fornecem dados a cada instante, com uma frequência característica de cada aplicação. Com isso, o DW é usualmente povoado considerando uma frequência alta, além de ser caracterizado por armazenar dados históricos, ou seja, relativos a um longo período de tempo (KIMBALL; ROSS, 2002). Consultas OLAP executadas neste DW são usualmente caras, desde que a junção-estrela tem custo computacional alto por lidar com tabelas de fatos muito volumosas.

Considerando um ambiente de *data warehousing* de imagens, pode-se verificar que o DW de imagens é ainda muito mais volumoso, desde que ele armazena os dados convencionais e as características intrínsecas das imagens (ANNIBAL, 2011). Além disso, em uma aplicação de imagens médicas, ou seja, um ambiente no qual o DW armazena imagens de exames médicos e dados convencionais relacionados à área médica, a frequência de povoamento desse DW é muito alta. Por exemplo, a aplicação pode estar relacionada a uma rede de hospitais de uma determinada cidade ou região, sendo os provedores numerosos e as atividades da área médica muito frequentes (SEBAA *et al.*, 2018). Consultas OLAP estendidas com predicado de similaridade de imagens executadas neste DW de imagens são muito mais caras, desde que envolvem onerosos cálculos de distância entre imagens (TRAINA *et al.*, 2007) em adição à já custosa operação de junção-estrela.

Ambientes de *data warehousing* de imagens podem ser beneficiados pelo uso de ambientes com alto poder de armazenamento e processamento. Além de possibilitar o processamento das consultas OLAP estendidas com predicado de similaridade, estes ambientes também oferecem subsídios para a alta disponibilidade dos dados e tolerância à falhas, com estratégias de replicação dos dados. No contexto da área médica, estes ambientes facilitam o compartilhamento de informações, de forma que, com a visão geral dos dados de uma rede médica, o suporte à tomada de decisão torna-se mais robusto. Por exemplo, o estudo descrito em (SEBAA *et al.*, 2018) utiliza ambientes paralelos e distribuídos para distribuir com equidade o número de médicos de cada unidade da rede, de acordo com a necessidade caracterizada pela análise dos dados.

No que tange a ambientes paralelos e distribuídos, pode-se utilizar tanto *clusters* de computadores ou ambientes de computação em nuvem (CHEN; MAO; LIU, 2014). Para simplificar a interação do usuário com tais ambientes, pode-se utilizar *frameworks* de processamento paralelo e distribuído, como o MapReduce (DEAN; GHEMAWAT, 2008) e Spark (ZAHARIA *et*

*al.*, 2010). Estes *frameworks* abstraem do usuário as complexidades inerentes ao paralelismo, facilitando a utilização e permitindo a otimização das consultas analíticas (SHI *et al.*, 2015).

Conforme detalhado no Capítulo 5, existem diversos estudos na literatura que investigam o processamento eficiente da junção-estrela usando *frameworks* de processamento paralelo e distribuído (AFRATI; ULLMAN, 2010; BLANAS *et al.*, 2010; BRITO *et al.*, 2016; BRITO *et al.*, 2020; HAN *et al.*, 2011; TAO *et al.*, 2013; ZHANG; WU; LI, 2013). Segundo (BRITO *et al.*, 2016), a utilização desses *frameworks* aumenta significativamente o desempenho computacional das consultas analíticas. Adicionalmente, também existem estudos que investigam operações de similaridade de imagens utilizando esses *frameworks* (GIANGRECO; KABARY; SCHULDT, 2014; LI *et al.*, 2017; NGUYEN *et al.*, 2016; NGUYEN; HUH, 2017). Esses estudos também mostram ganhos no desempenho no cálculo das operações de similaridade. Já outros estudos na literatura são voltados à proposta de arquiteturas baseadas em processamento paralelo e distribuído para armazenar DWs da área médica (DASH *et al.*, 2019; ISTEPHAN; SIADAT, 2015; ISTEPHAN; SIADAT, 2016; KUO *et al.*, 2015; RAJA; SIVASANKAR, 2014; SEBAA *et al.*, 2017; SEBAA *et al.*, 2018). Esses estudos mostram a necessidade e o interesse das organizações para se disponibilizar DWs da área médica utilizando *frameworks* de processamento paralelo e distribuído.

Entretanto, no melhor do conhecimento do autor desta dissertação, e de acordo com as pesquisas realizadas com base na revisão sistemática (Capítulo 5), nenhum estudo existente na literatura considera todos os aspectos conjuntamente, ou seja, o processamento eficiente da junção-estrela sobre um DW de imagens médicas que oferece suporte para o processamento eficiente de operações de similaridade, considerando ambientes paralelos e distribuídos. Esta dissertação de mestrado visa preencher essa lacuna identificada na literatura.

## 1.4 Objetivo

Esta dissertação de mestrado possui como objetivo principal a investigação do processamento de consultas analíticas estendidas com predicado de similaridade sobre *data warehouses* de imagens, considerando ambientes paralelos e distribuídos. O trabalho é baseado no ambiente de *data warehousing* de imagens *imageDWE* (TEIXEIRA *et al.*, 2015), o qual tem sido desenvolvido pelo grupo de pesquisa da orientadora desta dissertação. Adicionalmente, dentre os *frameworks* de processamento paralelo e distribuído, Spark armazena saídas intermediárias em memória principal ao invés de armazená-las em disco, como o MapReduce. Isso faz com que Spark usualmente garanta melhor desempenho no processamento da junção-estrela (BRITO *et al.*, 2016). Portanto, o trabalho é baseado no *framework* Spark.

Com base nesse objetivo, define-se a seguinte hipótese:

**Hipótese.** *É possível processar eficientemente consultas analíticas estendidas com predicado de*

*similaridade sobre um data warehouse de imagens em ambientes paralelos e distribuídos.*

Como consequência da investigação da hipótese supradescrita, a presente dissertação introduz as seguintes contribuições:

- Proposta do método *BloOmnImg* (acrônimo para *Bloom Filter Omni for analytical Image queries*), o qual integra as técnicas *bloom filter cascade join* e *Omni* para o processamento eficiente das consultas de interesse em Spark. O *bloom filter cascade join* (TARKOMA; ROTHENBERG; LAGERSPETZ, 2012) é uma estrutura de dados probabilística que é usada para filtrar dados de interesse durante o processamento da junção-estrela. A técnica *Omni* (TRAINA *et al.*, 2007) é usada para diminuir a quantidade de cálculos de distância nas operações de similaridade de imagens. Por ser baseado em uma estrutura de dados probabilística, *BloOmnImg* é indicado para sistemas computacionais com pouca memória primária disponível.
- Proposta do método *BrOmnImg* (acrônimo para *Broadcast Omni for processing analytical Image queries*), o qual integra as técnicas *broadcast join* e *Omni* para o processamento das consultas. O *broadcast join* (BLANAS *et al.*, 2010) é uma técnica que utiliza uma estrutura de dados exata que é usada para realizar a junção-estrela diretamente, de modo paralelo, em cada nó do *cluster*. A técnica *Omni* (TRAINA *et al.*, 2007) é usada para diminuir a quantidade de cálculos de distância nas operações de similaridade de imagens. Por ser baseado em uma estrutura de dados exata, *BrOmnImg* é indicado para sistemas computacionais que possuem quantidade suficiente de memória primária disponível.
- Proposta do método *BrOmnImg<sup>CF</sup>* (acrônimo para *Broadcast Omni for processing analytical Image queries with Conventional Filter*), o qual estende o método *BrOmnImg* utilizando o predicado convencional e a técnica *Omni* para diminuir a quantidade de cálculos de distância no processamento do predicado de similaridade. *BrOmnImg<sup>CF</sup>* é mais restrito do que *BrOmnImg*, desde que ele requer que a tabela de fatos tenha um relacionamento unário com a tabela que armazena os vetores de características das imagens.
- Proposta do método *SimSparkOLAP* (acrônimo para *OLAP queries extended with Similarity Factors in Spark*), o qual estende *BrOmnImg* para processar, de modo eficiente e concomitantemente, diversos fatores de similaridade, como geográfico, socioeconômico e de imagens. A proposta desse método também inclui a especificação de vários esquemas-estrela que consideram diferentes configurações de armazenamento dos fatores de similaridade. *SimSparkOLAP* garante o processamento eficiente de uma nova gama de consultas analíticas estendidas com esses fatores de similaridade, enriquecendo ainda mais a tomada de decisão da área médica.

O trabalho desenvolvido pode ser aplicado a qualquer ambiente de *data warehousing* de imagens. Entretanto, do ponto de vista de imagens, o trabalho foi realizado considerando

como motivação a manipulação de imagens médicas, principalmente devido à importância da necessidade da análise dessas imagens e de seu impacto para a sociedade.

Ademais, a produção dos seguintes artigos científicos também pode ser considerada como contribuições oriundas da presente dissertação de mestrado:

- Artigo intitulado “*The SimilarQL framework: similarity queries in plain SQL*” (TRAINA *et al.*, 2019), publicado em *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC)*, 2019. Neste artigo investiga-se o processamento de consultas que consideram diferentes operações de similaridade, como *range query*, *K-nn query*, *join range query*, *join K-nn query* e diferentes fatores de similaridade, como geográficos e socioeconômicos.
- Artigo intitulado “**Processamento Eficiente de Consultas Analíticas Estendidas com Predicado de Similaridade em Spark**” (ROCHA; CIFERRI, 2019a), publicado na sessão de *short papers* do XXXIV Simpósio Brasileiro de Banco de Dados (SBBD), 2019. Neste artigo, é descrita uma versão preliminar do método *BrOmnImg*. O artigo foi classificado entre os 5 melhores artigos do evento, tendo sido convidado para ser estendido e submetido no *Journal of Information and Data Management*. Esse artigo estendido está em fase final de preparação, e descreve a versão final do método *BrOmnImg* e também o método *BrOmnImg<sup>CF</sup>*.
- Artigo intitulado “**Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um data warehouse de imagens em ambientes paralelos e distribuídos**” (ROCHA; CIFERRI, 2019b), publicado no Workshop de Teses e Dissertações em Banco de Dados (WTDBD) do XXXIV Simpósio Brasileiro de Banco de Dados (SBBD), 2019. O artigo descreve o objetivo da dissertação de mestrado, bem como os resultados preliminares obtidos.
- Artigo intitulado “*Healthcare decision-making over a geographic, socioeconomic, and image data warehouse*”, aceito no evento científico *International Workshop on Intelligent Data – From Data to Knowledge*, 2020. Neste artigo é descrita uma versão preliminar do método *SimSparkOLAP*.

## 1.5 Estrutura da dissertação

Além do presente capítulo introdutório, esta dissertação de mestrado possui mais oito capítulos, estruturados da seguinte forma:

- No Capítulo 2 são descritos os principais conceitos relacionados a um ambiente de *data warehousing* convencional.

- No Capítulo 3 são abordados conceitos relacionados a um ambiente de *data warehousing* de imagens, além de ser introduzido o *imageDWE*.
- No Capítulo 4 são descritos os principais conceitos relacionados aos ambientes paralelos e distribuídos.
- No Capítulo 5 é realizada uma revisão sistemática, a qual aborda as fases de planejamento e condução da mesma.
- No Capítulo 6 é proposto o método *BloOmnImg* para o processamento eficiente da junção-estrela estendida com predicado de similaridade, utilizando a estrutura probabilística *bloom filter* e a técnica Omni.
- No Capítulo 7 são propostos os métodos *BrOmnImg* e *BrOmnImg<sup>CF</sup>* para o processamento eficiente da junção-estrela estendida com predicado de similaridade, utilizando a técnica *broadcast join* e a técnica Omni.
- No Capítulo 8 é proposto o método *SimSparkOLAP* para o processamento eficiente da junção-estrela estendida com predicado de similaridade de imagens, geográficos e socioeconômicos.
- No Capítulo 9 são descritas as conclusões do trabalho, incluindo um resumo das contribuições, as dificuldades encontradas e os possíveis trabalhos futuros.



---

## DATA WAREHOUSING

---

### 2.1 Considerações iniciais

Este capítulo descreve os conceitos de ambientes de *data warehousing* utilizados no desenvolvimento do trabalho. Na seção 2.2 é realizada uma contextualização destes ambientes. A arquitetura destes ambientes pode ser definida em três camadas: nível conceitual, nível lógico e nível físico. Na seção 2.3 são descritas as principais características do nível conceitual. Na seção 2.4 são apresentados os principais conceitos presentes no nível lógico. O nível físico está relacionado com o uso de índices bitmap de junção (GOLFARELLI; RIZZI, 2018) e de visões materializadas (VAISMAN; ZIMÁNYI, 2014) como forma de otimização no processamento de consultas. Como esses conceitos não são usados no desenvolvimento do trabalho, eles não serão descritos neste capítulo. O capítulo é finalizado na seção 2.5 com as considerações finais.

### 2.2 Contextualização

Ao contrário de muitos conceitos da computação, o conceito de ambientes de *data warehousing* foi desenvolvido na indústria e não no meio acadêmico, pois o mesmo está diretamente relacionado ao levantamento de dados estratégicos que auxiliam a tomada de decisão em uma organização (GOLFARELLI; RIZZI, 2018). Estes dados são obtidos por meio de consultas analíticas sobre o DW.

Um DW consiste em um banco de dados que possui dados de fontes autônomas, heterogêneas e distribuídas. Os dados dessas fontes são extraídos e integrados, de forma que possíveis inconsistências sejam eliminadas (INMON, 2005). Nessa integração, também é realizada a limpeza dos dados. Como resultado, gera-se um banco de dados (DW) orientado a assunto, ou seja, define-se um ou mais assuntos de interesse frente aos dados que são extraídos e do que pretende-se investigar a partir desses dados (VAISMAN; ZIMÁNYI, 2014). O DW também é não volátil e histórico (INMON, 2005). A primeira característica está relacionada ao fato de

que os dados não sofrem alteração por um longo período de tempo, enquanto que a segunda característica está relacionada ao fato de que os dados permanecem armazenados no DW por um longo período de tempo (GOLFARELLI; RIZZI, 2018).

Com os dados carregados no DW, a etapa de consulta OLAP pode ser realizada. Conforme descrito no Capítulo 1, esta etapa compreende o processamento de consultas analíticas voltadas à tomada de decisão de uma organização. Além de oferecer suporte para a realização de consultas OLAP, a partir do DW é possível gerar relatórios estatísticos, realizar mineração de dados e utilizar ferramentas de *Business Intelligence* (BI).

## 2.3 Modelagem em nível conceitual

O modelo de dados multidimensional permite a visualização dos dados do DW de acordo com um número variável de dimensões, de acordo com as características do DW que está sendo projetado. Na seção 2.3.1 são descritos aspectos estáticos desse modelo, enquanto que na seção 2.3.2 são descritos os aspectos dinâmicos desse modelo.

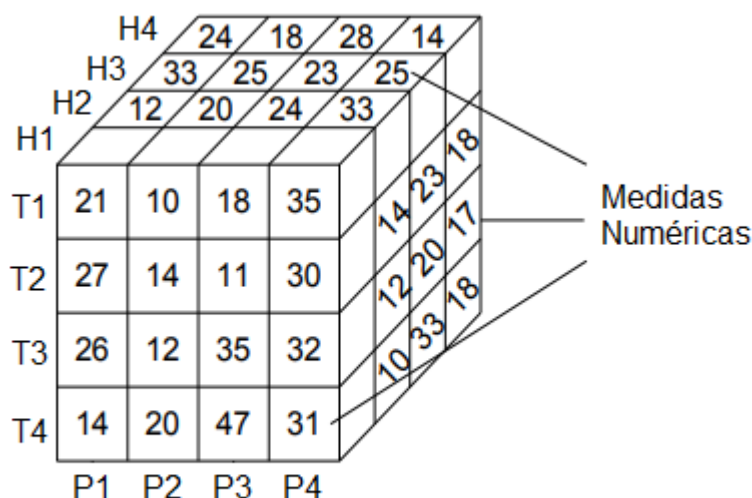
### 2.3.1 Aspectos estáticos

O modelo multidimensional em nível estático é definido por um cubo multidimensional que possui medidas numéricas, as quais representam os objetos de análise. Além disso, também estão presentes as dimensões, que representam as dimensões do cubo. Com isso, a medida numérica é definida pelo conjunto das dimensões (CHAUDHURI; DAYAL, 1997). Por exemplo, considerando o DW de uma organização médica, pretende-se armazenar a quantidade de exames de um paciente, em um determinado hospital, em uma determinada data. A quantidade de exames é a medida numérica, pois caracteriza-se pelo que pretende-se avaliar. O paciente, hospital e data, são as dimensões, as quais definem a quantidade de exames.

Graficamente, o nível estático pode ser representado por um hipercubo de dados multidimensional, no qual cada eixo do cubo representa uma dimensão e cada célula do cubo representa a medida numérica de acordo com os valores dos eixos (dimensões) (GOLFARELLI; RIZZI, 2018; CIFERRI *et al.*, 2013; HARINARAYAN; RAJARAMAN; ULLMAN, 1996). A Figura 1 ilustra para o exemplo dado o hipercubo de dados, ou seja, cada eixo do cubo representa uma dimensão, sendo o eixo  $x$  representando a dimensão *paciente*, eixo  $y$  representando a dimensão *data* (trimestre) e eixo  $z$  representando a dimensão *hospital*. Cada célula do produto representa a medida numérica *quantidade de exames*. Considerando a coordenada [P1, T1, H1], foram realizados 21 exames no paciente P1 no primeiro trimestre T1 no hospital H1.

As dimensões são compostas de atributos inerentes à mesma, sendo que, dentre esses atributos há a possibilidade de haver uma ou mais hierarquias de atributos. Por exemplo, a dimensão *data* do hipercubo da Figura 1 pode ser composta pelos atributos *mês*, *trimestre*, *semestre* e *ano*. É possível definir uma hierarquia entre esses atributos, de forma a relacionar



Figura 1 – Hipercubo de Dados. Fonte: Adaptado de (CIFERRI *et al.*, 2013).

atributos de maior granularidade, ou seja, menor nível de detalhe (ex. *ano*) com atributos de menor granularidade, ou seja, com maior nível de detalhe (ex. *mês*) (VAISMAN; ZIMÁNYI, 2014).

O operador de ordenação parcial ( $\preceq$ ) (HARINARAYAN; RAJARAMAN; ULLMAN, 1996) é amplamente utilizado para representar formalmente a hierarquia de atributos. Considerando a data, a hierarquia pode ser expressa como *ano*  $\preceq$  *semestre*  $\preceq$  *trimestre*  $\preceq$  *mês*. Semanticamente, o operador de ordenação parcial indica que meses podem ser agregados em trimestres, trimestres podem ser agregados em semestres e semestres podem ser agregados em anos. A partir da hierarquia de atributos, define-se o conceito de reticulado de cuboides (HARINARAYAN; RAJARAMAN; ULLMAN, 1996). Um reticulado de cuboides é composto por um conjunto de visões (ou agregações) e a relação de dependência entre essas visões. Ele deve conter, pelo menos, a visão com os dados mais detalhados. Adicionalmente, ele pode conter uma visão completamente agregada, a qual pode ser calculada a partir de qualquer outra visão.

### 2.3.2 Aspectos dinâmicos

Os aspectos dinâmicos relacionados ao modelo multidimensional estão ligados às operações OLAP que podem ser realizadas no modelo. Estas operações permitem manipular o cubo de dados de diferentes maneiras, com o objetivo de oferecer informações provenientes destas operações, para que possam ser úteis em uma possível tomada de decisão de uma organização (CIFERRI *et al.*, 2013).

Operações OLAP incluem:

- **Slice and dice:** Restringe o cubo de dados em um determinado valor ou faixa de valores. Para a operação *slice* é selecionado somente um valor fixo de alguma dimensão. Por

exemplo: “Qual a quantidade de exames por paciente por hospital no trimestre T1?”. Como resultado da operação de *slice*, ocorre a diminuição da dimensionalidade do cubo de dados. Em contrapartida, a operação *dice* seleciona uma faixa de valores para um atributo. Por exemplo: “Qual a quantidade de exames por paciente por hospital entre os trimestres T2 e T3?”. Diferentemente da operação de *slice*, a operação de *dice* não diminui a dimensionalidade do cubo de dados (CHAUDHURI; DAYAL, 1997);

- **Drill-down:** Permite a visualização dos dados de um nível de maior granularidade para um nível de menor granularidade, ou seja, de dados menos detalhados para dados mais detalhados (VAISMAN; ZIMÁNYI, 2014). Essa operação é importante para que o usuário possa especificar a sua busca de forma mais detalhada conforme haja a necessidade. Por exemplo, dada uma visualização da quantidade de exames por paciente por hospital por semestre, ao perceber algo que pode ser explorado dentro de um determinado semestre, realiza-se a operação *drill-down* para que seja exibida a quantidade de exames por paciente por hospital por trimestre;
- **Roll-up:** De forma oposta à operação *drill-down*, a operação *roll-up* permite a visualização dos dados de um nível de menor granularidade para um nível de maior granularidade, ou seja, de dados mais detalhados para dados menos detalhados (VAISMAN; ZIMÁNYI, 2014). Por exemplo, dada a visualização quantidade de exames por pacientes por hospital por semestre, obtém-se uma visão mais ampla ao realizar a operação *roll-up* para que seja exibida a quantidade de exames por paciente por hospital por ano;
- **Drill-across:** Relaciona medidas numéricas de tabelas de fatos diferentes que compartilham uma ou mais dimensões em comum. Como resultado dessa operação, é possível visualizar estas medidas numéricas conjuntamente (CHAUDHURI; DAYAL, 1997). Por exemplo, dada duas medidas numéricas (quantidade de exames e lucro) que compartilham as dimensões paciente, hospital e data, é possível consultar a quantidade de exames e o lucro da organização por paciente por hospital por trimestre. Além da necessidade de se compartilhar dimensões em comum, para que a realização da operação *drill-across* seja possível, também é necessário que as dimensões compartilhadas estejam no mesmo nível de granularidade. Em situações em que isso não ocorre, antes de se aplicar a operação de *drill-across*, é necessário realizar a operação de *roll-up* ou *drill-down* nas dimensões compartilhadas, deixando-as em conformidade;
- **Pivot:** Rotaciona o cubo de dados multidimensional de forma a (re)organizar a ordem na qual as dimensões do cubo são exibidas (CHAUDHURI; DAYAL, 1997). Por exemplo, ao invés de se visualizar a quantidade de exames por paciente por hospital por trimestre pode-se visualizar o cubo de dados como a quantidade de exames por hospital por paciente por trimestre.

## 2.4 Modelagem em nível lógico

A representação em nível lógico do modelo de dados multidimensional pode ser feita usando-se sistema Multidimensional OLAP (MOLAP) ou sistema Relacional OLAP (ROLAP). Pode-se também usar as duas formas de representação, ou seja, *Hybrid OLAP (HOLAP)* (VAIS-MAN; ZIMÁNYI, 2014). Neste trabalho, considera-se que o DW é armazenado em sistemas ROLAP e, portanto, apenas as características desse armazenamento são descritas a seguir.

Sistemas ROLAP armazenam os dados do cubo de dados em estruturas relacionais, e utilizam a tecnologia dos sistemas gerenciadores de banco de dados relacionais para o oferecimento das funcionalidades relacionadas. Existem dois esquemas amplamente utilizados para a representação relacional dos dados do DW, o esquema-estrela e o esquema floco de neve (KIMBALL; ROSS, 2002), que são descritos nas seções 2.4.1 e 2.4.2.

### 2.4.1 Esquema-estrela

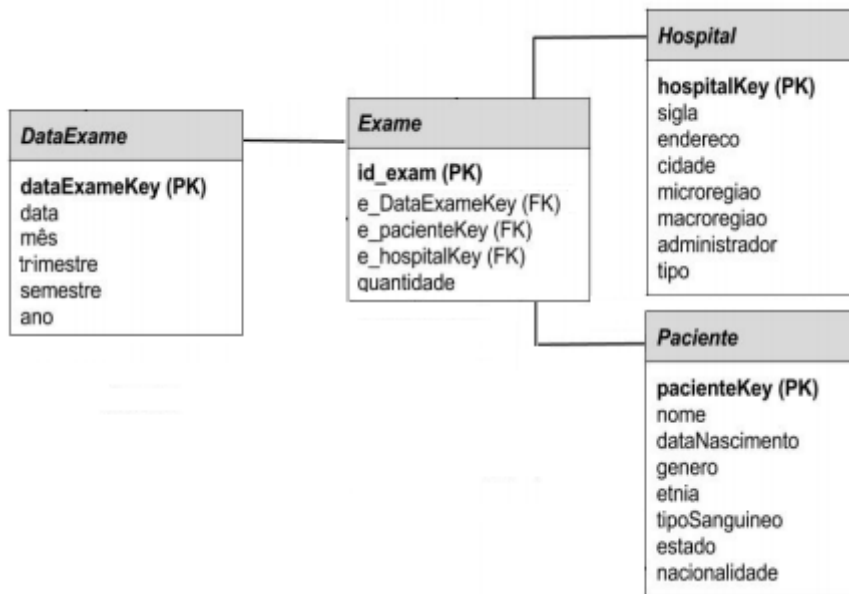
Em um esquema-estrela existem tabelas de fatos e tabelas de dimensão. A tabela de fatos é visualmente localizada no centro da estrela e armazena as medidas numéricas. A tabela de fatos também armazena as referências para as tabelas de dimensão, ou seja, há uma chave estrangeira na tabela de fatos para cada tabela de dimensão considerada. A chave primária da tabela de fatos é uma chave composta por todas as chaves estrangeiras nela presente. As tabelas de fatos são finas e longas, ou seja, usualmente são definidas em termos de poucas dimensões, porém possuem muitos dados armazenados (GOLFARELLI; RIZZI, 2018).

As tabelas de dimensão são localizadas visualmente nas extremidades da estrela e armazenam uma chave primária e os atributos inerentes da dimensão, sendo eles descritivos e hierárquicos. Os dados são desnormalizados, ou seja, existe redundância no armazenamento dos atributos de uma hierarquia. Uma característica das tabelas de dimensões é que elas são largas e curtas, ou seja, possuem muitos atributos, porém não contêm muitos dados quando comparadas à tabela de fatos (GOLFARELLI; RIZZI, 2018). A Figura 2 ilustra um esquema-estrela de um ambiente de *data warehousing* no contexto da área médica. A tabela de fatos *Exame* possui como medida numérica a quantidade de exames por paciente por hospital por data. As tabelas *Paciente*, *Hospital* e *DataExame* são as tabelas de dimensão.

### 2.4.2 Esquema floco de neve

O esquema floco de neve possui as mesmas características do esquema-estrela no que se refere à tabela de fatos. Ele difere-se do esquema-estrela com relação às tabelas de dimensão, nas quais os atributos que pertencem às hierarquias de atributos são normalizados, de forma que não haja redundância de dados para esses atributos. Com isso, cada atributo da hierarquia de atributos é armazenado em uma tabela de dimensão específica (KIMBALL; ROSS, 2002). Essas tabelas de dimensão são relacionadas entre si usando-se como base a hierarquia de atributos,

Figura 2 – Exemplo de esquema-estrela de um ambiente de *data warehousing* da área médica. Fonte: Adaptado de (TEIXEIRA, 2015).

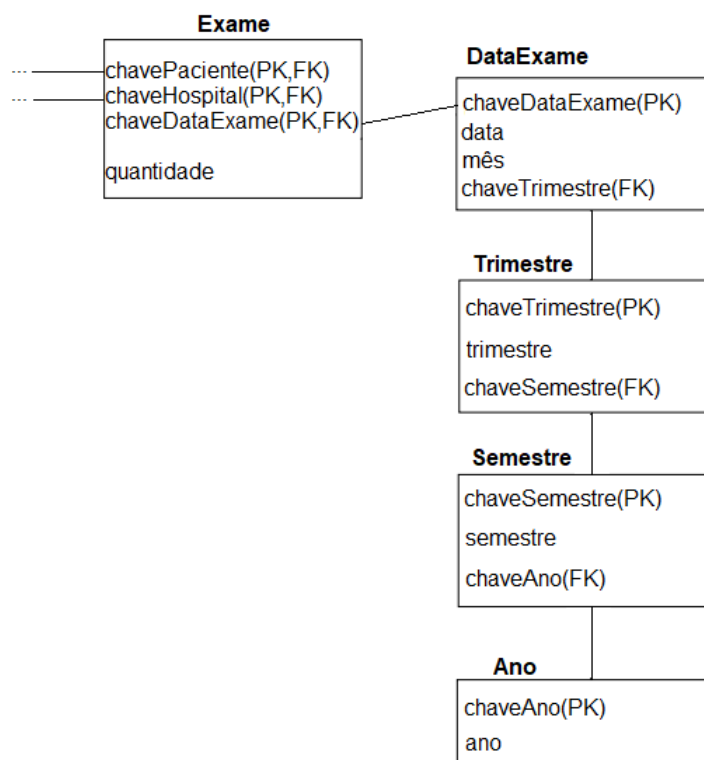


sendo que apenas a tabela de dimensão referente ao atributo de menor nível de granularidade é ligada com a tabela de fatos. Cada tabela de dimensão possui a chave estrangeira que referencia a dimensão que armazena o próximo atributo na hierarquia de atributos, e assim sucessivamente, para todos os atributos da hierarquia.

A principal diferença entre o esquema floco de neve e o esquema-estrela é a normalização dos atributos de dimensão que pertencem à hierarquia. O esquema-estrela não normaliza estes atributos, requerendo maior espaço de armazenamento, gerando redundância e dificultando a garantia da consistência. O esquema floco de neve normaliza estes atributos, eliminando a redundância e facilitando a garantia da consistência, além de requerer menor espaço de armazenamento (PONNIAH, 2004). Porém, a redundância usualmente garante melhor desempenho no processamento de consultas OLAP. Isso está relacionado ao fato de que a redundância dos atributos reduz o número de junções necessárias para a realização das consultas OLAP, sendo que a junção é uma operação de alto custo computacional. Como resultado, o esquema-estrela usualmente garante melhor desempenho no processamento de consultas analíticas quando comparado com o esquema floco de neve.

A Figura 3 exemplifica um esquema floco de neve em um ambiente de *data warehousing* da área médica. Este esquema possui uma hierarquia de atributos na dimensão *DataExame*. A ordenação parcial destes atributos é definida por  $ano \preceq semestre \preceq trimestre \preceq mês$ . Cada atributo da hierarquia é armazenado em uma tabela de dimensão. A nível de exemplificação, as dimensões *Paciente* e *Hospital* foram abstraídas no esquema floco de neve.

Figura 3 – Exemplo de esquema floco de neve de um ambiente de *data warehousing* da área médica.  
 Fonte: Elaborado pelo autor.



## 2.5 Considerações finais

Neste capítulo foram descritos conceitos relacionados aos ambientes de *data warehousing* que são usados como base para o desenvolvimento do trabalho. Foram detalhados aspectos da camada de nível conceitual, englobando aspectos estáticos e dinâmicos. Foram apresentados os tipos de modelagem em nível lógico, como esquema-estrela e esquema floco de neve.

O ambiente de *data warehousing* descrito nesse capítulo oferece uma gama de consultas baseadas em atributos convencionais. Porém, dados de imagens podem ser incorporados a este ambiente, oferecendo uma nova gama de consultas que podem ser realizadas a partir do conteúdo destas imagens. No próximo capítulo, Capítulo 3, são descritos conceitos relacionados ao ambiente de *data warehousing* de imagens.



---

## ***DATA WAREHOUSING DE IMAGENS***

---

---

### **3.1 Considerações iniciais**

Este capítulo descreve conceitos relacionados a ambientes de *data warehousing* de imagens. Na seção 3.2 é feita uma contextualização desses ambientes. Na seção 3.3 são descritas as principais características das consultas por similaridade. Na seção 3.4 é detalhada a arquitetura do ambiente de *data warehousing* de imagens *ImageDWE*, o qual é utilizado como base para o desenvolvimento desta dissertação de mestrado. O capítulo é finalizado na seção 3.5 com as considerações finais.

### **3.2 Contextualização**

DWs convencionais armazenam apenas dados convencionais, os quais possuem uma relação de ordem e permitem a utilização de operações relacionais de comparação. Por exemplo, o atributo idade de um paciente pode ser comparado a uma determinada idade, utilizando as operações maior e menor. Em adição aos dados convencionais, um DW de imagens também armazena dados não convencionais das imagens. Porém, comparar dados de imagens utilizando operações relacionais não possui sentido semântico. Faz mais sentido uma consulta que compara o quanto uma determinada imagem é similar a outra imagem. Dessa forma, a incorporação de dados de imagens em DWs permite que uma nova gama de consultas interessantes neste contexto seja realizada. Essa nova gama de consultas engloba predicado convencional e predicado de similaridade de imagens.

Esta dissertação de mestrado usa como base o conceito de data warehouse de imagens introduzido em (Annibal *et al.*, 2010): um DW de imagens é projetado segundo um esquema-estrela diferenciado que possui, além de tabelas de dimensão com dados convencionais, uma ou mais tabelas que armazenam dados relacionados à manipulação de imagens, sendo que essas tabelas podem ser tabelas de dimensão ou tabelas de fatos.

Para que sejam realizadas operações com sentido semântico sobre o DW de imagens, uma das abordagens é extrair características intrínsecas destas imagens, como cor, textura e forma, e armazená-las como vetores de características. Assim, estas características permitem que operações que comparam imagens sejam realizadas, as quais são operações de similaridade.

### 3.3 Consulta por similaridade

As consultas por similaridade envolvendo imagens, no contexto de recuperação de imagens por conteúdo - *Content-Based Image Retrieval* (CBIR) - estão relacionadas à comparação de imagens de alguma forma. A forma de comparar as imagens é, normalmente, realizada por uma função de distância, a qual calcula quão similares duas imagens são. Esta função utiliza os vetores de características das imagens (obtidos pelos extratores de características) para a execução deste cálculo. Ao definir a operação de similaridade, tem-se a consulta por similaridade (TRAINA *et al.*, 2007).

O processo de planejamento de uma aplicação que ofereça suporte para consultas por similaridade envolve alguns conceitos, os quais são descritos na Tabela 1. Nessa tabela também são mostrados quem é o especialista responsável por definir ou construir os pontos principais que envolvem esses conceitos e um exemplo de profissional especialista responsável por estas definições no contexto de recuperação de imagens médicas por conteúdo - *Content-Based Medical Image Retrieval* (CBMIR).

Tabela 1 – Áreas de definições e construções dos conceitos que envolvem consulta por similaridade. Fonte: Elaborado pelo autor.

Conceitos	Definido por	Ex: CBIR Médico
Extração de características	Especialistas da aplicação	Médico
Função de distância	Especialistas do domínio de dados	Profissional de processamento de imagens
Ambiente para executar buscas por similaridade	Especialistas em recuperação de dados	Engenheiro de dados

Uma das técnicas utilizadas para modelar o vetor de características é o espaço métrico, e, a partir dessa técnica, surge o conceito de Método de Acesso Métrico (MAM). Um MAM consiste de uma estrutura de indexação para que a consulta por similaridade seja executada de uma maneira mais eficiente, reduzindo o espaço de busca. Existem diversos métodos de acesso métrico presentes na literatura, como VP-tree (YANILOS, 1993), M-tree e variações (CIACCIA; PATELLA; ZEZULA, 1997; SKOPAL; HOKSZA, 2007; SKOPAL; LOKOČ, 2008; ZHOU *et al.*, 2003), Slim-tree e variações (POLA; TRAINA; TRAINA JR., 2009; TRAINA JR.



*et al.*, 2002; VIEIRA *et al.*, 2004), Onion-tree (CARÉLO *et al.*, 2009; CARÉLO *et al.*, 2011) e técnica Omni (TRAINA *et al.*, 2007).

Na seção 3.3.1 são descritos conceitos de extração de características para que seja definido o vetor de características. Na seção 3.3.2 são expostos os principais conceitos de função de distância, assim como, exemplos de funções de distância mais utilizados para a realização de consultas por similaridade. Na seção 3.3.3 são introduzidos os conceitos de espaço métrico, assim como os principais operadores de similaridade. Por fim, na seção 3.3.3.2 é detalhada a técnica Omni, a qual é usada no ambiente *imageDWE* e também nos métodos propostos nesta dissertação.

### 3.3.1 Extração de características

A extração de características é um processo importante no contexto de busca por similaridade, pois as características extraídas guiam a semântica da consulta por similaridade. No caso de CBIR, as imagens podem ser comparadas por diferentes aspectos, como cor, textura ou forma. Portanto, os extratores de característica extraem, por meio de uma representação numérica na forma de vetores de características, as características de uma imagem com base em uma determinada percepção. Tais percepções devem ser, idealmente, definidas pelo especialista da aplicação (Tabela 1).

Existem diferentes estudos consolidados na área de processamento de imagens e visão computacional. A categorização de descritores de interesse para a presente pesquisa é descrita a seguir, com base nos aspectos considerados (DATTA *et al.*, 2008; GONZALEZ; WOODS, 2008).

- **Cor:** Esta característica da imagem possui uma estratégia muito utilizada para sua representação, o histograma. Um histograma de cor representa a quantidade de *pixels* existentes na imagem para cada cor. Dessa forma, se a imagem possui  $N$  cores, um vetor *hist* de tamanho  $N$  é utilizado para representar o histograma desta imagem, sendo que  $hist[i]$  define a quantidade de *pixels* da cor  $i$  que estão presentes na imagem.
- **Textura:** A extração de características de texturas em imagens não é um processo intuitivo como a característica de cor. Métodos conhecidos por obterem bons resultados para esta característica são os descritores de Haralick (Haralick; Shanmugam; Dinstein, 1973; Haralick, 1979). Eles se baseiam em uma abordagem estatística, mais precisamente, utilizam matrizes de coocorrência para representar as proximidades dos pixels. Dada uma matriz  $P(i, j)$ , as células representam o número de vezes em que o *pixel* de intensidade  $i$  é vizinho do *pixel* de intensidade  $j$  dada uma direção e uma distância.

### 3.3.2 Funções de distância

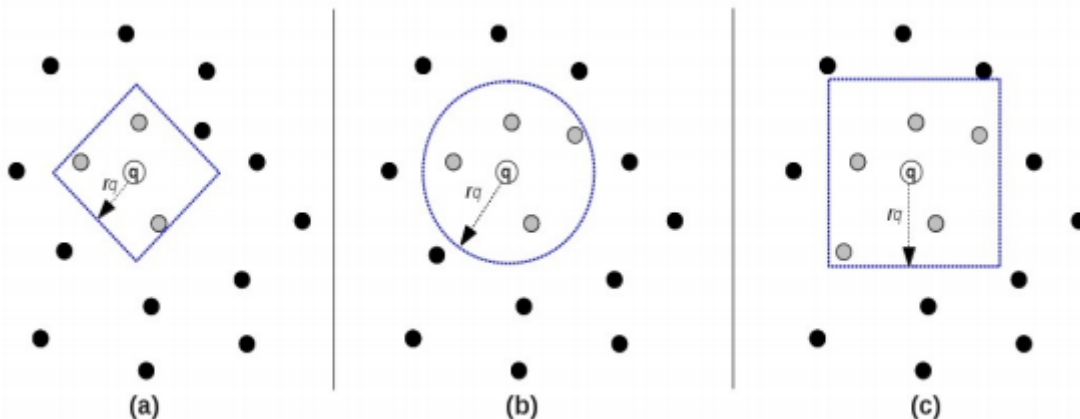
A partir das características extraídas das imagens, uma função de distância utiliza estas representações numéricas para calcular a similaridade entre duas imagens. Entretanto, para cada contexto de consulta por similaridade, usualmente existe uma determinada função de distância que responde à consulta de maneira mais satisfatória. Com isso, existem muitas funções de distância na literatura. As funções de distância mais conhecidas e utilizadas são as funções da família Minkowski (WILSON; MARTINEZ, 1997), sendo estas, baseadas nas normas  $L_p$ , definidas a seguir.

No contexto CBIR, cada imagem possui um vetor de características de tamanho  $n$ , sendo  $n$  o número de características (FELIPE, 2005). Com isso, pode-se definir duas imagens,  $Q$  e  $C$ , com vetores de características  $Q = (q_1, q_2, \dots, q_n)$  e  $C = (c_1, c_2, \dots, c_n)$ , respectivamente. O cálculo de distância entre estas duas imagens, utilizando a função  $L_p$ , é representada na Equação 3.1. O parâmetro  $p$  caracteriza as variações das funções de distância da família Minkowski. Ao definir  $p = 1$ , tem-se a função de distância  $L_1$  (Manhattan). Se  $p = 2$ , define-se a função  $L_2$  (Euclidiana). Por fim, outra variação bastante usada é calcular o limite da função  $L_p$  com  $p$  tendendo ao infinito, com isso, tem-se  $L_\infty$  (Chebychev).

$$L_p(Q, C) = \sqrt[p]{\sum_{i=1}^n |q_i - c_i|^p} \quad (3.1)$$

A Figura 4 ilustra, em um espaço bidimensional, o espaço de equidistância das principais variações da função  $L_p$  para o cálculo de distância a partir de um centro de consulta  $q$  e com raio de abrangência  $r_q$ . Pode-se notar que a quantidade de objetos presentes no raio de abrangência é diferente para cada variação  $L_p$ . Com isso, a definição da função de distância é um processo importante (Tabela 1).

Figura 4 – Exemplos dos espaços de equidistâncias gerados pela distância  $L_p$ . (a) Distância Manhattan ( $L_1$ ), (b) Distância Euclidiana ( $L_2$ ) e (c) Distância Chebychev ( $L_\infty$ ). Fonte: (TEIXEIRA, 2015).



### 3.3.3 Espaço métrico

A partir das características extraídas dos objetos que estão sendo analisados (por exemplo, imagens), deve-se definir um ambiente para que o mesmo permita a realização de operações de similaridade. Duas principais abordagens são as mais utilizadas na literatura, o espaço vetorial e o espaço métrico (TRAINA *et al.*, 2007). O espaço vetorial é um ambiente em que os objetos possuem suas características intrínsecas representadas em um vetor de tamanho fixo, por exemplo um histograma de cor em escala de cinza. Ou seja, se um objeto é representado por um vetor de tamanho  $n$ . Tal objeto pode ser representado por um ponto em um espaço vetorial de  $n$  dimensões (TRAINA *et al.*, 2007).

Já no espaço métrico, as características intrínsecas de um objeto podem não possuir um tamanho fixo de dimensões, pois existem aplicações em que as características variam de objeto para objeto. Por exemplo, em aplicações que envolvem impressões digitais existem pontos de interesse que são extraídos, sendo que estes pontos variam entre si (TRAINA *et al.*, 2007). Com isso, em um espaço métrico  $M$  é possível calcular a similaridade entre dois objetos, de forma que  $M = (U, d)$ , sendo  $U$  o conjunto de objetos e  $d$  a função de distância que deve medir a similaridade entre os objetos (CIACCIA; PATELLA, 2002). O modelo de espaço métrico é menos restritivo, e inclui o espaço vetorial e funções de distância apropriadas para esse contexto.

#### 3.3.3.1 Operadores de similaridade

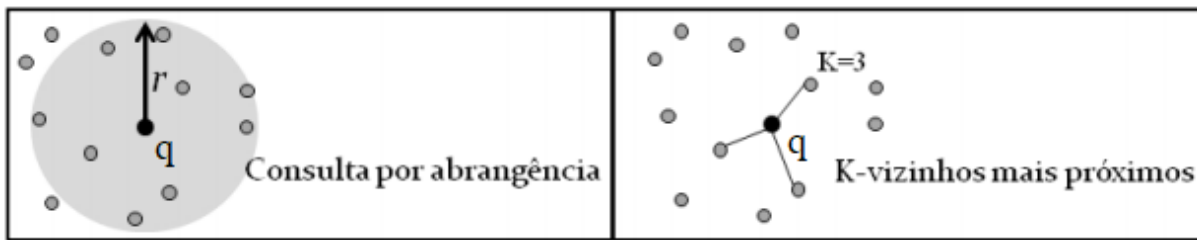
Existem dois operadores por similaridade: comparação por abrangência (*range query*) e comparação por vizinhança - *K-nearest neighbor* (*K-nn query*) (CHÁVEZ *et al.*, 2001). Dado um objeto como centro de consulta  $q$  e o espaço métrico  $M$ , esses tipos de busca são definidos da seguinte forma:

- **Range query** -  $Rng(q, r)$ : A consulta por abrangência retorna todos os elementos do conjunto  $U$  que estão a uma distância de  $q$  igual ou menor a  $r$ .
- **K-nn query** -  $K-nn(q, k)$ : A consulta por vizinhança retorna os  $k$  elementos do conjunto  $U$  mais próximos à  $q$ .

Para os dois tipos de operadores, o cálculo de similaridade, entre o centro de consulta  $q$  e os elementos do  $U$ , é realizado utilizando a função de distância  $d$ . A partir das distâncias estas operações são realizadas. A Figura 5 ilustra estas operações.

Um dos maiores desafios de aplicações que envolvem consultas por similaridade é reduzir o *gap* semântico, ou seja, a diferença do resultado esperado pelo especialista da aplicação e o resultado que realmente é retornado. Este *gap* ocorre com diferentes usuários tendo objetivos diferentes com a realização da consulta, definindo então, diferentes parâmetros para a execução da mesma. Assim, a qualidade de uma consulta por similaridade é medida de forma subjetiva de acordo com o interesse do usuário. Com isso, um CBIR pode permitir a realização de

Figura 5 – Exemplo de consulta por abrangência e k-vizinhos mais próximos. Fonte: Adaptado de (ANNIBAL, 2011).



consultas por similaridade envolvendo diferentes percepções, ou seja, diferentes características que representam uma imagem, para que o usuário possa definir quais interessam melhor para sua consulta. Por exemplo, um médico pode estar interessado em consultas por similaridade envolvendo as cores da imagem, outro pode estar interessado em medir a similaridade com base na textura. Assim, diferentes extratores de características devem ser utilizados para extrair as diferentes percepções da imagem, por meio de vetores de características. Desse modo, cada percepção extraída caracteriza-se como uma camada perceptual, neste caso, camadas perceptuais de cor e textura definem dois conjuntos  $U_1$  e  $U_2$ . Ou seja, para cada camada perceptual define-se seu vetor de características, espaço métrico e estrutura de indexação (ANNIBAL, 2011).

### 3.3.3.2 Técnica OMNI

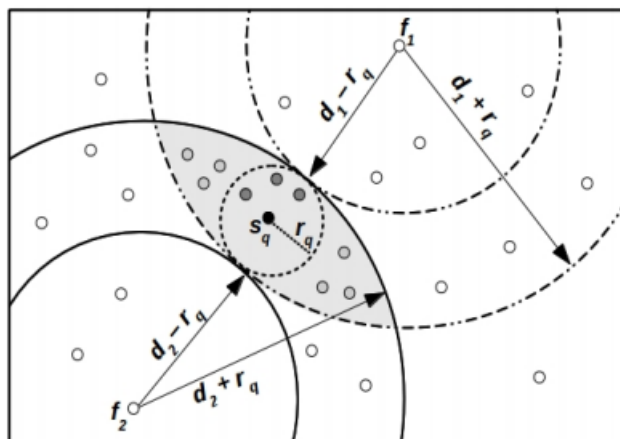
A técnica Omni é um método de acesso métrico baseado em disco e dinâmico (TRAINA *et al.*, 2007). Por meio da definição de elementos representativos, define-se uma faixa no espaço métrico de forma que os objetos localizados fora desta faixa não precisam ser verificados, ou seja, não há falsos negativos. Em contrapartida, os objetos que estão dentro da faixa precisam ter sua distância calculada para o centro de consulta, pois pode haver falsos positivos. Isso significa que os objetos pertencentes a esta faixa são candidatos à resposta, sendo necessário calcular a distância desses objetos.

Ao definir  $n$  elementos representativos, ou seja, determinados objetos pertencentes ao conjunto  $U$ , armazena-se a distância destes elementos para todos os outros objetos deste conjunto. Dada uma consulta por abrangência com centro de consulta  $s_q$  e raio  $r_q$ , a consulta é realizada usando a técnica Omni em duas etapas: filtragem e refinamento.

A etapa de filtragem consiste em formar a região no espaço métrico que caracteriza os elementos candidatos à resposta; esta região é a *minimum-bounding-Omni-region* (mbOr). Uma faixa é definida no espaço métrico para cada elemento representativo. Esta faixa é definida pelo raio (distância  $d_n$  do elemento representativo até o centro de consulta  $s_q$ ) somado e subtraído do raio de abrangência  $r_q$  ( $d_n + r_q$  e  $d_n - r_q$ ). A mbOr consiste da região de intersecção das faixas de cada elemento representativo. A Figura 6 ilustra a mbOr para um espaço métrico com dois elementos representativos  $f_1$  e  $f_2$ . Percebe-se que a área cinza é a mbOr, sendo que, os elementos que estão fora desta área não precisam ser considerados para uma consulta envolvendo o centro

de consulta  $s_q$  e de raio  $r_q$ . Porém, os objetos dentro da área cinza são candidatos à resposta e precisam ter sua distância para  $s_q$  calculada na fase de refinamento. Se a distância for igual ou menor à  $r_q$ , o objeto pertence à resposta (TRAINA *et al.*, 2007).

Figura 6 – Exemplo de *mbOr* (área cinza) para um centro de consulta  $s_q$  de raio  $r_q$  com elementos representativos  $f_1$  e  $f_2$ . Fonte: Adaptado de (TRAINA *et al.*, 2007)



O número de elementos representativos e quais são esses elementos representativos são questões importantes a serem definidas, pois a localização dos elementos representativos pode aumentar ou diminuir a *mbOr*. Segundo (TRAINA *et al.*, 2007), os elementos representativos mais apropriados são os elementos da borda do conjunto, desde que eles tendem a gerar uma *mbOr* pequena. Esses elementos podem ser obtidos pelo algoritmo *Hull of Foci* (HF) (TRAINA *et al.*, 2007).

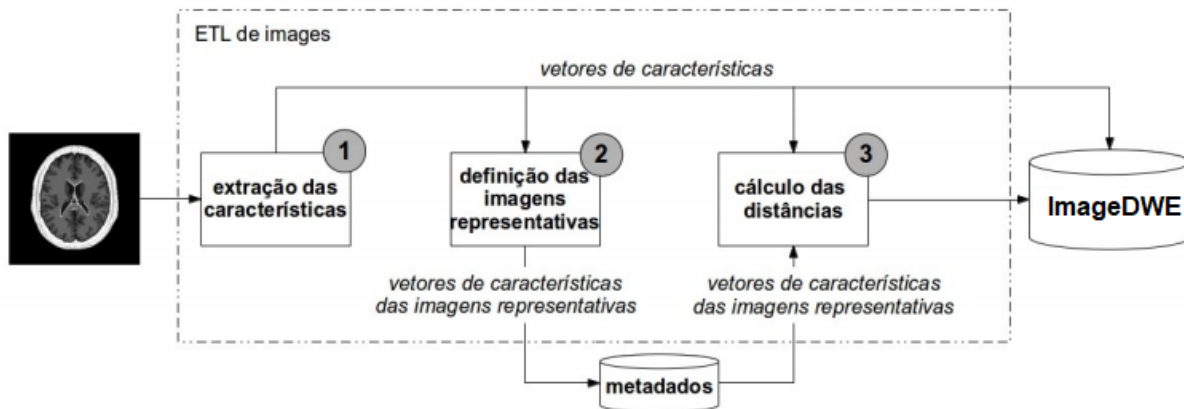
### 3.4 Arquitetura do ambiente de *data warehousing* de imagens *ImageDWE*

Nesta seção é detalhada a arquitetura do ambiente de *data warehousing* de imagens *ImageDWE*, o qual é usado para o desenvolvimento desta dissertação de mestrado. O ambiente provê aspectos relacionados ao processo ETL estendido para a extração de características de imagens e atributos de similaridade, ao projeto do DW de imagens e ao processamento de consultas OLAP estendidas com predicado de similaridade de imagens. Enquanto a manipulação de dados convencionais é feita de acordo com as funcionalidades providas por ambientes de *data warehousing* convencionais, a manipulação de dados de imagens representa a inovação introduzida.

A primeira etapa do processo ETL de imagens é a extração das características das imagens, sendo as mesmas armazenadas em um vetor de características para cada camada perceptual definida. Com a extração das características são definidos os elementos representativos para a utilização da técnica Omni (seção 3.3.3.2). Assim, para cada imagem do conjunto de dados, calcula-se a distância entre as mesmas e os elementos representativos. A Figura 7 ilustra

o processo ETL de imagens, sendo que a definição dos elementos representativos e o cálculo das distâncias (etapas 2 e 3) devem ser realizados para cada camada perceptual. Como resultado do processo de ETL, o vetor de características e as distâncias para cada elemento representativo são armazenados no DW de imagens.

Figura 7 – Processo ETL de imagens. Fonte: Adaptado de (TEIXEIRA *et al.*, 2015).

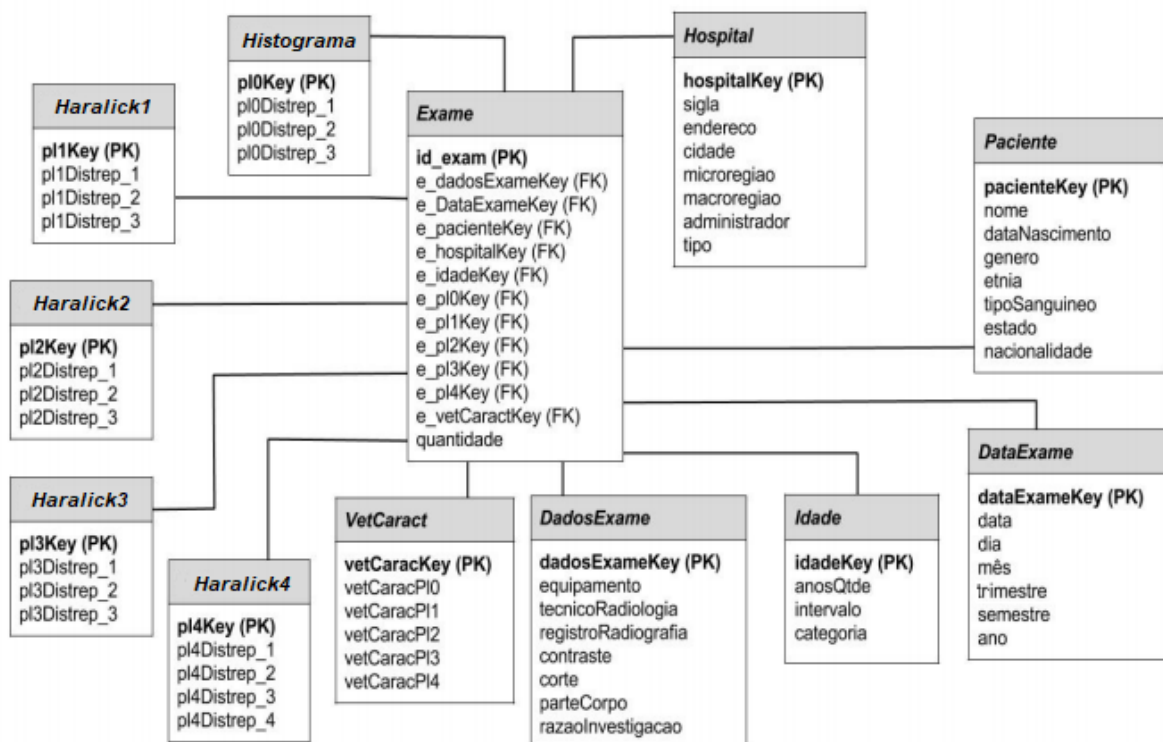


Conforme definido em (Annibal *et al.*, 2010), um DW de imagens possui um esquema-estrela diferenciado que possui, além de tabelas de dimensão com dados convencionais, uma ou mais tabelas que armazenam dados relacionados à manipulação de dados sobre imagens, sendo que estas tabelas podem ser tabelas de dimensão ou tabela de fatos. A Figura 8 ilustra um DW de imagens da área médica, o qual é projetado de acordo com a tabela de fatos *Exame* e as tabelas de dados médicos convencionais *Idade*, *Paciente*, *Hospital*, *DataExame* e *DadoExame*. As tabelas de dados de imagem armazenam a distância das imagens para os elementos representativos de cada camada perceptual. As tabelas *Histograma*, *Haralick1*, *Haralick2*, *Haralick3* e *Haralick4* representam, respectivamente, as camadas perceptuais de histograma de cor, Haralick Variância, Haralick Entropia, Haralick Uniformidade e Haralick Homogeneidade. Além disso, o vetor de características de cada camada perceptual é armazenado na tabela *VetCaract*.

O esquema da Figura 8 é um dos 4 esquemas para um DW de imagens propostos por (ANNIBAL, 2011). Estes esquemas se diferenciam no que diz respeito ao armazenamento dos vetores de características de cada camada perceptual. A Figura 9 ilustra estas diferenças por meio dos retângulos vermelhos em destaque. No esquema EBM1, armazena-se os vetores de características nas tabelas de dimensão da camada perceptual correspondente que armazena as distâncias para os elementos representativos (Figura 9a). Já o esquema EBM2, armazena os vetores de características em uma tabela para cada camada perceptual (Figura 9b). Outro modo para o armazenamento dos vetores de características de cada camada perceptual é em uma só tabela *VetCaract*, definido pelo esquema EBM3 (Figura 9c). Por fim, no esquema EBM4, é possível armazenar os vetores de características como fatos na tabela de fatos (Figura 9d).

O fluxo do processamento de consultas OLAP estendidas com predicado de similaridade de imagens no *ImageDWE* pode ser visualizado na Figura 10. O predicado convencional é

Figura 8 – Esquema-estrela DW de imagens. Fonte: (TEIXEIRA, 2015).



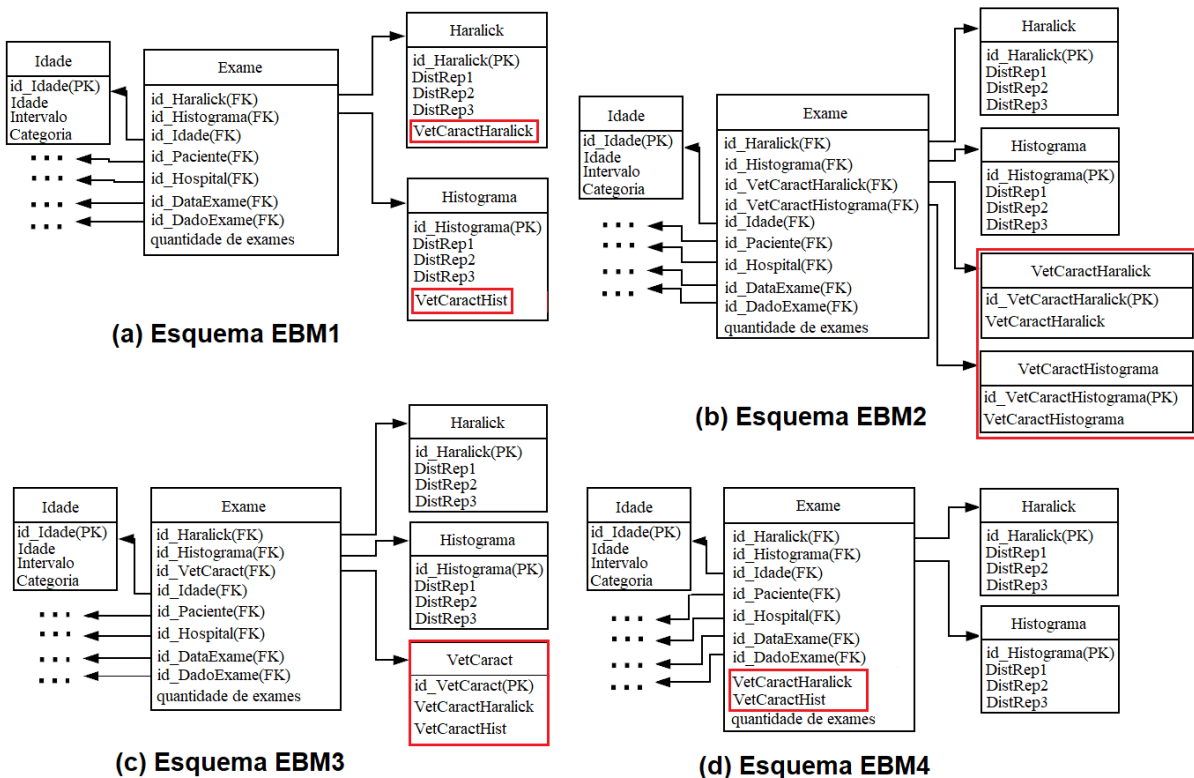
solucionado em (a). As características da imagem de centro de consulta são extraídas na etapa (b). O cálculo da distância dos elementos representativos para o centro de consulta é realizado na etapa (c). Na fase (d) ocorre a etapa de filtragem da técnica de Omni, de acordo com as distâncias para os elementos representativos da imagem de centro de consulta e de cada imagem do *ImageDWE* que atende ao predicado convencional já filtrado. Com a etapa de filtragem calcula-se a distância de similaridade entre as imagens candidatas à resposta e a imagem de centro de consulta (e), por meio dos vetores de características, caracterizando a fase de refinamento da técnica de Omni. Por fim, a resposta da consulta é finalizada e apresentada ao usuário em (f).

### 3.5 Considerações finais

Neste capítulo foram descritos conceitos relacionados aos ambientes de *data warehousing* imagens. Foram detalhados aspectos de consultas por similaridade, englobando a extração de características, a definição de funções de distância, as características do espaço métrico e o detalhamento da técnica Omni. Também foi descrito o ambiente de *data warehousing* de imagens *ImageDWE* utilizado como base para o desenvolvimento desta dissertação de mestrado, com destaque para o processo ETL de imagens, o armazenamento de imagens no DW e o processamento de consultas analíticas estendidas com predicado de similaridade.

O ambiente de *data warehousing* de imagens descrito nesse capítulo prevê um processamento centralizado das consultas analíticas estendidas com predicado de similaridade. Entretanto,

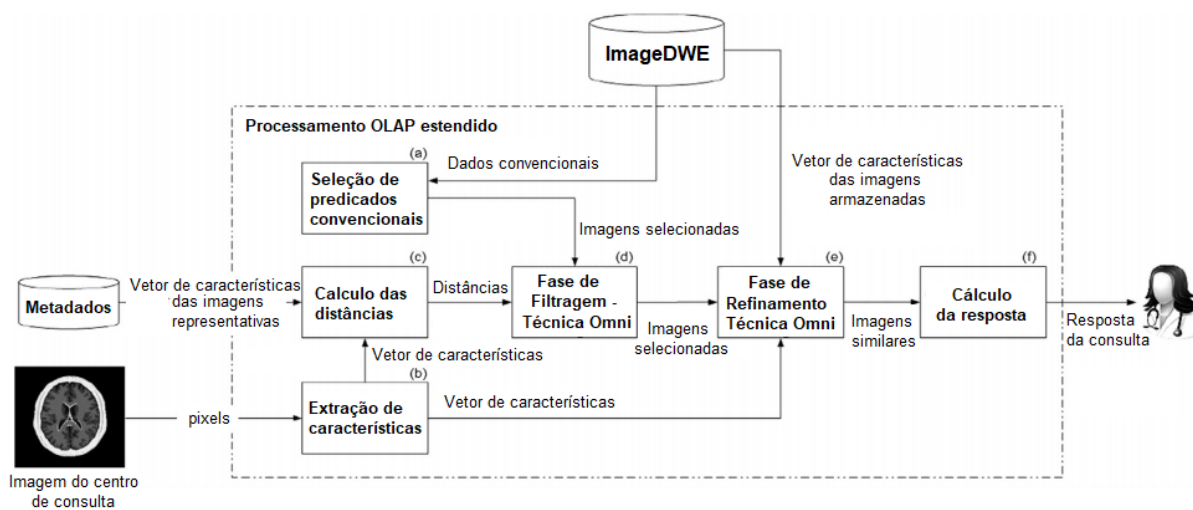
Figura 9 – Quatro diferentes esquema-estrela para um DW de imagens médicas. (a) Esquema EBM1: armazenamento dos vetores de características e dos atributos para pesquisa por similaridade de cada camada perceptual conjuntamente em uma mesma tabela de dimensão. (b) Esquema EBM2: armazenamento de cada vetor de características em uma tabela de dimensão separada por camada perceptual e armazenamento dos atributos para pesquisa por similaridade em uma outra tabela de dimensão separada por camada perceptual. (c) Esquema EBM3: armazenamento de todos os vetores de características em uma única tabela de dimensão e armazenamento dos atributos para pesquisa por similaridade em tabelas de dimensão separadas por camada perceptual. (d) Esquema EBM4: armazenamento dos vetores de características como fatos em uma tabela de fatos e armazenamento dos atributos para pesquisa por similaridade em tabelas de dimensão separadas por camada perceptual. Fonte: Adaptado de (ROCHA; CIFERRI, 2018).



devido ao grande volume de dados armazenados neste ambiente, esse processamento pode ser beneficiado pelo uso de ambientes paralelos e distribuídos. No próximo capítulo, Capítulo 4, são descritos conceitos relacionados ao processamento paralelo e distribuído.



Figura 10 – Processamento de consulta analítica estendida com predicado de similaridade. Fonte: Adaptado de (TEIXEIRA *et al.*, 2015).





---

# PROCESSAMENTO PARALELO E DISTRIBUÍDO

---

## 4.1 Considerações iniciais

Este capítulo descreve os conceitos relacionados aos ambientes paralelos e distribuídos que estão diretamente relacionados com as contribuições descritas nesta dissertação de mestrado. Na seção 4.2 é realizada uma contextualização destes ambientes. Na seção 4.3 é detalhado o sistema de arquivos distribuído *Hadoop Distributed File System* (HDFS). Nas seções 4.4 e 4.5 são detalhadas as características dos *frameworks* MapReduce e Spark, respectivamente. Além disso, as técnicas *bloom filter cascade join* e *broadcast join* são apresentadas nas seções 4.6 e 4.7, respectivamente. Estas técnicas processam a junção-estrela em ambientes paralelos e distribuídos. O capítulo é finalizado na seção 4.8 com as considerações finais.

## 4.2 Contextualização

A ideia de ambientes que coletam, armazenam e analisam dados volumosos não é muito recente, já que ambientes de *data warehousing* têm sido bastante utilizados (Capítulo 2). Porém, o termo *big data* é recente na última década. Apesar de não haver um consenso na definição desse termo, o modelo dos 3 V's (MCAFEE; BRYNJOLFSSON, 2012) é bastante aceito na literatura, e define as seguintes características:

- **Volume:** Está associada ao grande volume de dados coletados pelas organizações. O avanço de tecnologias de armazenamento acessíveis e da internet impulsionaram esta coleta massiva de dados.
- **Variedade:** Está ligada à heterogeneidade dos formatos dos dados, desde que considera-se várias fontes de dados e não existe um formato padronizado para a geração dos

dados.

- Velocidade: Está relacionado com a velocidade na qual os dados são gerados, armazenados e analisados.

Prover armazenamento e gerenciamento de dados para ambientes com estas características tem sido um grande desafio, que tem sido resolvido usando-se ambientes de computação em nuvem e ambientes de *clusters* de computadores (CHEN; MAO; LIU, 2014).

O surgimento dos *clusters* de computadores foi motivado pela possibilidade de prover um ambiente computacional com bom desempenho adicionando-se *hardware* de baixo custo quando comparado com o uso de supercomputadores. O ambiente é flexível, adicionando ou removendo componentes, e possui custo financeiro relativamente baixo. *Clusters* de computadores possuem computadores que operam de maneira paralela e distribuída, os quais são interligados por meio de uma rede de alta velocidade. As principais vantagens introduzidas pelos *clusters* de computadores são: (i) garantir alta disponibilidade do sistema, ou seja, em caso de falhas, é possível recuperar o sistema por meio de redundância; e (ii) prover balanceamento de carga, ou seja, ao receber uma tarefa, o sistema deve processá-la de modo que cada computador (nó) deste ambiente possa executar parte desta tarefa de forma paralela, e com isso, utilizar de forma eficiente seus recursos atrelado a um bom desempenho computacional (Sadashiv; Kumar, 2011).

Uma alternativa aos *clusters* de computadores são os ambientes de computação em nuvem, os quais possibilitam acesso a um conjunto de recursos computacionais compartilhados e interligados via rede, abstraindo do usuário complexidades de infraestrutura (ARMBRUST *et al.*, 2009). Ambientes de computação em nuvem provêm serviços sob demanda e que estão sempre disponíveis, em qualquer lugar e a qualquer hora. Adicionalmente, o usuário só paga pelos serviços que usar (BHOWMIK, 2017).

Além do uso de ambientes computacionais com grande capacidade de armazenamento e processamento, o gerenciamento de grandes volumes de dados também requer o uso de sistemas de arquivos distribuídos, como o HDFS e o emprego de paradigmas de programação paralela e distribuída, como os *frameworks* MapReduce e Spark.

### 4.3 HDFS

Para que os ambientes paralelos e distribuídos possam ser utilizados, há a necessidade de existir um sistema de arquivos com funcionalidades que favoreçam as características destes ambientes. Surge então o HDFS, que é um sistema de arquivos distribuído, com alta escalabilidade e tolerante a falhas, e que permite armazenar uma grande quantidade de dados com um baixo custo de *hardware*.

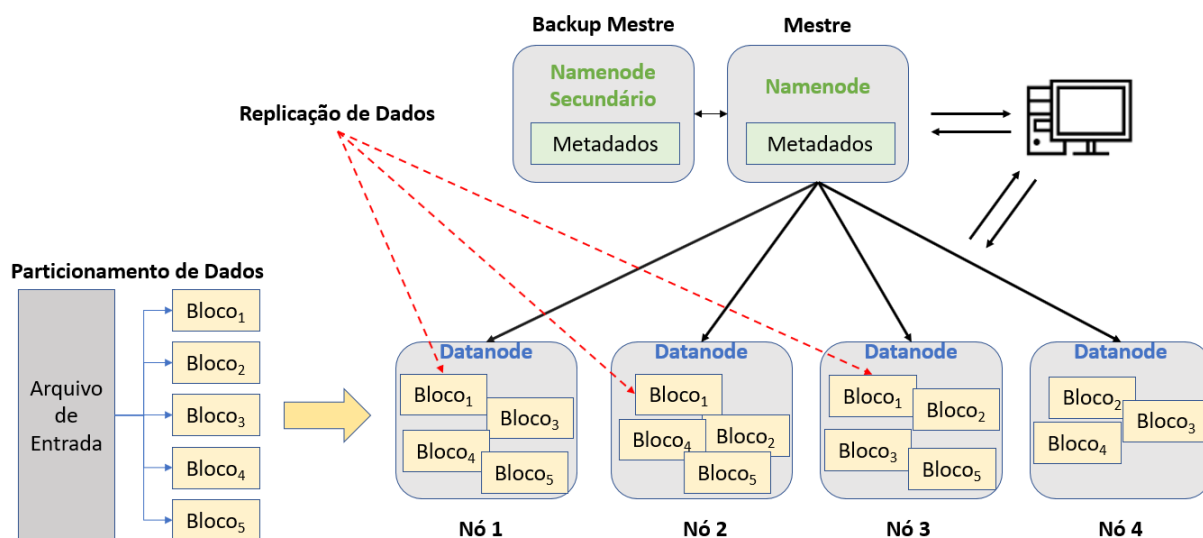
Ao armazenar um arquivo de dados, o HDFS divide este arquivo em blocos e armazena estes blocos, de modo replicado nos nós do ambiente. Por padrão, cada bloco do arquivo é

replicado em três nós. Uma das vantagens introduzidas por essa replicação refere-se ao aumento da possibilidade de que uma tarefa que precise de um determinado bloco de dados seja executada em um nó próximo a esse bloco (SHVACHKO *et al.*, 2010). Outra vantagem refere-se à tolerância a falhas. Se algum nó falhar, é possível obter dados replicados em outros nós.

O tamanho do bloco pode ser configurável, sendo o tamanho padrão de 64MB. Quanto maior o tamanho do bloco, menor a quantidade de blocos necessários para se armazenar um arquivo de dados. Como resultado, o número de acessos a disco para acessar o arquivo de dados completo é menor. Além disso, com menos blocos para se gerenciar aumenta-se a escalabilidade da arquitetura.

A arquitetura do HDFS possui dois tipos de nós: (i) *NameNode* e, (ii) *DataNode*. O *NameNode*, ou nó mestre, possui os metadados sobre a localização dos blocos dos arquivos, enquanto que os *DataNodes*, ou nós escravos, são os nós que armazenam os blocos em si. Quando uma aplicação precisa acessar um dado, ela deve primeiro acessar o *NameNode* para saber em qual nó aquele bloco está armazenado e, depois, acessar o *DataNode* correspondente. Também pode haver um *NameNode* secundário, utilizado como *backup*. Se o *NameNode* primário falhar, o *NameNode* secundário assume automaticamente a função (SHVACHKO *et al.*, 2010). A Figura 11 ilustra o exemplo de um arquivo de dados dividido em 5 blocos e armazenado de acordo com a arquitetura do HDFS.

Figura 11 – Arquitetura HDFS. Fonte: Adaptado de (BRITO, 2017).



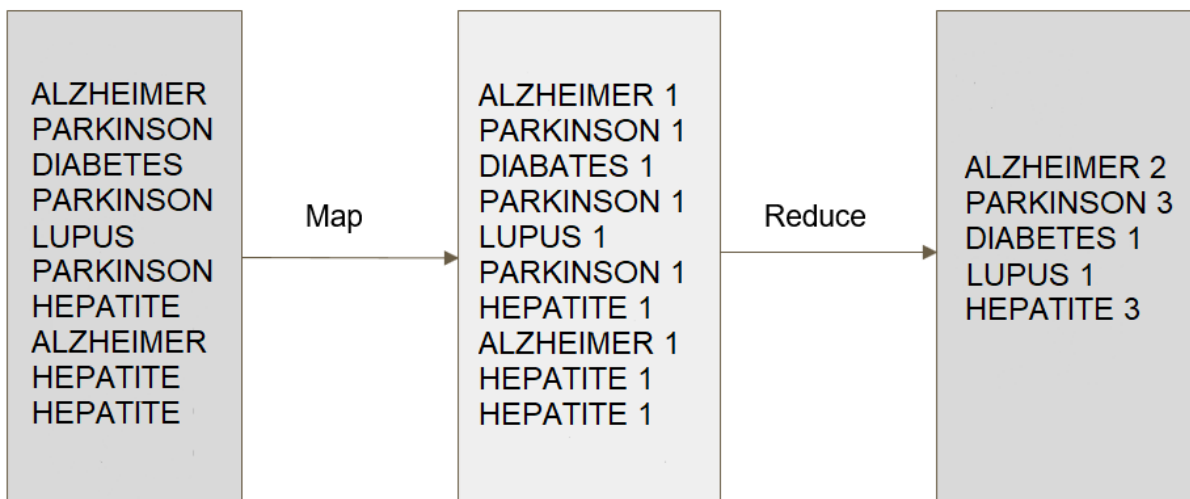
Existem *frameworks* que auxiliam no processamento paralelo e distribuído de dados e que oferecem funcionalidades que abstraem e facilitam o uso do HDFS. Basicamente, estes *frameworks* podem ser divididos em: (i) *disk-based*, em que o processamento é baseado em acessos à disco, como no caso do *framework* MapReduce; e (ii) *in-memory*, quando o processamento é realizado principalmente em memória principal, como é o caso do *framework* Spark (ZHANG *et al.*, 2015). Estes *frameworks* são detalhados nas seções 4.4 e 4.5, respectivamente.

## 4.4 Framework MapReduce

O surgimento do *framework* MapReduce foi motivado pela necessidade de se realizar uma quantidade grande de tarefas relativamente simples em um tempo aceitável. Nesse modelo, devem ser definidas duas funções, uma função *map* e uma função *reduce* (DEAN; GHEMAWAT, 2008).

A função *map* recebe os dados de entrada e processa-os de forma a gerar pares intermediários de chave e valor. A função *reduce* recebe estes pares intermediários como entrada e processa todos os valores para uma mesma chave, gerando uma saída única (DERBEKO *et al.*, 2016). A Figura 12 mostra um exemplo de aplicação do número de vezes que uma determinada palavra ocorre em um texto da área médica. A entrada é definida de acordo com cada palavra presente no texto. A função *map* processa esta entrada gerando, para cada palavra da entrada, uma chave, definida pela própria palavra, e um valor, definido pelo valor 1. Ou seja, cada palavra é tratada de forma independente, sendo o valor 1 usado para indicar que esta palavra é processada uma única vez. Dados os pares chave e valor intermediários, a função *reduce* processa os mesmos como entrada da seguinte forma. Para cada palavra usada como chave, é processada a lista de valores, referente ao número de vezes na qual a chave referente à palavra foi mapeada. Dessa forma, os valores desta lista são somados em uma variável, e então, para cada palavra (chave) é gerado o número de vezes que a palavra apareceu.

Figura 12 – Exemplo de aplicação de função *map* e *reduce*. Fonte: Elaborado pelo autor.



O processamento do *framework* MapReduce é baseado em um processo mestre e vários processos executores. O processo mestre é responsável por criar e delegar aos processos executores que estão ociosos a execução de uma determinada tarefa, referente às funções *map* e *reduce*. Já os processos executores são responsáveis por executar estas tarefas de modo paralelo (DERBEKO *et al.*, 2016).

O MapReduce é um *framework disk-based*, ou seja, ele realiza acessos a disco para ter acesso aos dados intermediários. A operação *map* armazena os pares chave e valor em disco

para que a operação *reduce* acesse o disco e processe como entrada estes dados. Os pares chave e valor são enviados pela rede para o *reducer* correto que processa-os, causando uma comunicação intensa na rede, denominada fase de *shuffle*. Como resultado, aplicações que possuem a característica iterativa, em que uma determinada etapa depende dos dados da etapa anterior, requerem muitos acessos a disco, gerando perda de desempenho computacional (SHI *et al.*, 2015).

O *framework* MapReduce pode ser utilizado pela implementação de código aberto da Apache Foundation: Apache Hadoop<sup>1</sup>.

## 4.5 Framework Spark

O *framework* Spark é do tipo *in-memory*, ou seja, armazena as saídas intermediárias em memória principal. Semelhante ao MapReduce, o Spark também possui um nó mestre que controla o fluxo das operações e distribui várias operações para serem executadas em paralelo, as quais são executadas pelos nós executores (ZAHARIA *et al.*, 2010).

Basicamente, o Spark se difere do MapReduce em dois aspectos: utilização dos conjuntos de Dados Distribuídos e Resilientes (RDD) e suporte ao agendamento de tarefas na forma de grafos direcionados e acíclicos (LI *et al.*, 2017).

Os RDDs são coleções de blocos de dados distribuídos capazes de serem reconstruídos caso a partição onde estão armazenados venha a ser perdida. RDDs armazenam saídas intermediárias em memória principal quando se deseja reutilizar essas saídas em operações futuras, evitando acessos a disco (LI *et al.*, 2017). Operações executadas sobre os RDDs incluem: (i) *parallelize*, que divide uma coleção de dados em diversas fatias que são enviadas para vários nós distintos para serem processadas; (ii) *flatMap*, que é semelhante à função *map* do MapReduce, ou seja, o usuário define uma função que, por meio de uma entrada, gera um saída intermediária; (iii) *reduce*, que é semelhante à função *reduce* do MapReduce; tal função é definida pelo usuário para que os elementos intermediários sejam associados e processados por esta função, gerando o conjunto final; (iv) *collect*, que envia todos os dados do conjunto de dados para o processo mestre; (v) *foreach*, que executa uma função específica para cada elemento do conjunto de dados; (vi) *cache*, que garante que o conjunto de dados seja mantido em memória principal após seu uso; e (vii) *save*, que persiste o conjunto de dados no banco de dados (ZAHARIA *et al.*, 2010).

No Spark, o processamento de tarefas pode conter vários estágios. Para tanto, este *framework* oferece suporte para o agendamento dos estágios na forma de grafos direcionados acíclicos. Com isso, as tarefas que não dependem de outras para serem executadas são agendadas para serem processadas paralelamente, aumentando o desempenho computacional (LI *et al.*, 2017).

---

<sup>1</sup> <<https://hadoop.apache.org>>

O *framework* Spark pode ser utilizado pela implementação de código aberto da *Apache Foundation*: Apache Spark<sup>2</sup>.

A operação de junção possui alta demanda em todos ambientes de manipulação a análise dados. Estas operações costumam ser custosas, com isso, a necessidade de otimização se faz presente, principalmente em contextos analíticos que consideram DWs, ou seja, com alto custo para processar a junção-estrela. Considerando *frameworks* de processamento paralelo e distribuído, duas técnicas que otimizam a junção-estrela se destacam: *bloom filter cascade join* (TARKOMA; ROTHENBERG; LAGERSPETZ, 2012) e *broadcast join* (BLANAS *et al.*, 2010). Estas técnicas consideram um DW convencional armazenado sobre o HDFS, ou seja, tanto a tabela de dimensão quanto a tabela de fatos estão particionadas em  $i$  e  $j$  blocos, respectivamente.

## 4.6 Bloom filter cascade join

A técnica *bloom filter cascade join* utiliza uma estrutura de dados probabilística para realizar seu processamento. Esta estrutura permite inserir elementos e consultar esses elementos, da seguinte forma. Cada elemento inserido é mapeado por um número definido de  $k$  funções *hash* em  $k$  bits da estrutura, na qual 0 representa a ausência do elemento e 1 representa a presença do elemento. O número de bits total da estrutura também é definido previamente ( $m$ ). Com isso, a consulta de um determinado elemento em *bloom filter* requer o mapeamento das funções *hash* para que os bits sejam verificados. Há três possibilidades ao consultar se um determinado elemento está nessa estrutura: (i) o elemento, de fato, está na estrutura; (ii) a estrutura acusa que o elemento está presente, porém, é um falso positivo; e (iii) o elemento não está na estrutura. Apesar da estrutura poder ter falsos positivos, por ser probabilística, não é possível haver falsos negativos. Isso significa que, se a estrutura acusa que um determinado elemento não está presente, esta informação é verdadeira.

A Figura 13 ilustra uma estrutura *bloom filter* na qual são inseridos os elementos X e Y pelo mapeamento de 3 funções *hash*,  $h_1()$ ,  $h_2()$  e  $h_3()$ . Cada bit mapeado é definido como tendo o valor 1. Durante a consulta, os bits do elemento sendo consultado primeiramente são mapeados por meio das funções *hash*. Se todos os bits tiverem valor igual a 1, o elemento pode ser um verdadeiro positivo (como o elemento X) ou um falso positivo (como o elemento Q). Porém, se ao menos um dos bits possuir o valor 0, com certeza o elemento não pertence à estrutura (como o elemento Z).

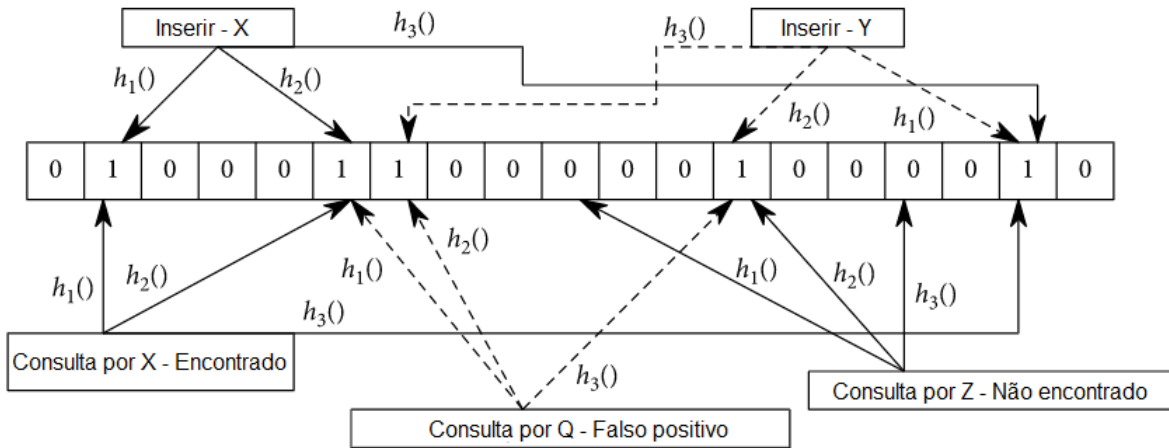
Como pode ser observado, existe a probabilidade de falsos positivos, de acordo com os parâmetros de inicialização do *bloom filter*, ou seja, número de funções *hash* ( $k$ ) e número total de bits ( $m$ ). Considerando  $n$  o número de elementos inseridos no *bloom filter*, a Equação 4.1

---

<sup>2</sup> <<https://spark.apache.org>>



Figura 13 – Exemplo de inserção e consulta na estrutura *bloom filter*. Fonte: Adaptado de (PATGIRI *et al.*, 2019)



permite calcular a probabilidade ( $p$ ) de falsos positivos.

$$p = (1 - e^{-\frac{k \cdot n}{m}})^k \quad (4.1)$$

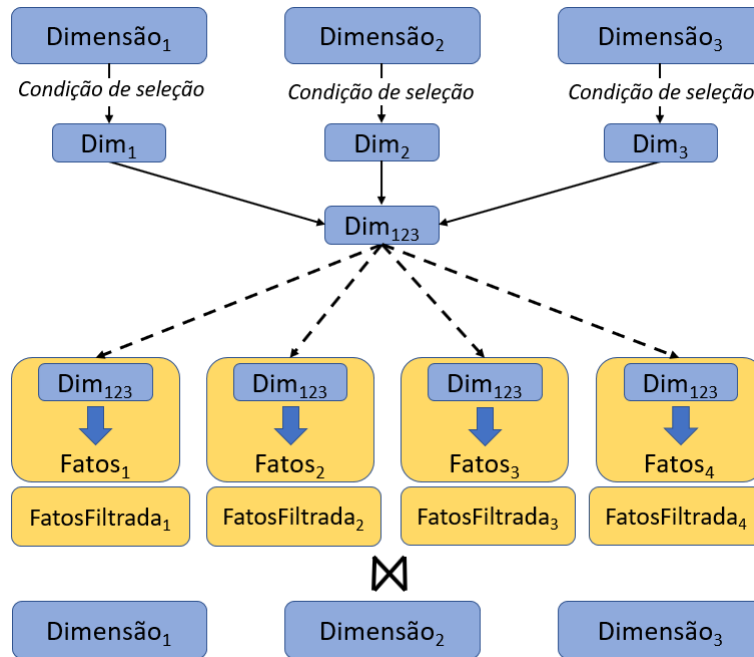
A técnica de *bloom filter cascade join* processa a junção-estrela sobre um DW convencional em um ambiente paralelo e distribuído (Figura 14). Cada condição de seleção é aplicada a cada bloco  $i$  da tabela de dimensão  $Dimensão_i$  ( $Dim_i$ ). O resultado do predicado convencional é armazenado na estrutura probabilística *bloom filter* ( $Dim_{123}$ ) e transmitido para todos os nós do *cluster* para filtrar cada bloco  $j$  da tabela de fatos  $Fatos_j$  ( $FatosFiltrada_j$ ). Isso é realizado por meio da chave estrangeira da tabela de fatos e da chave primária do predicado convencional armazenada na estrutura *bloom filter*. Como *bloom filter* é probabilístico, é necessário realizar a junção entre a tabela de fatos filtrada e a tabela de dimensão, eliminando falsos positivos. Porém, com a filtragem da tabela de fatos realizada anteriormente, esta operação é otimizada.

Por ser baseada em uma estrutura probabilística, a técnica *bloom filter cascade join* requer pouco espaço para armazenamento do predicado convencional. Com isso, ela é indicada para ambientes que não possuem muita memória primária.

## 4.7 Broadcast join

A técnica *broadcast join* processa a junção-estrela sobre um DW convencional em um ambiente paralelo e distribuído da seguinte forma (Figura 15). Cada condição de seleção é aplicada a cada bloco  $i$  da tabela de dimensão  $Dimensão_i$  ( $Dim_i$ ). Assume-se que o resultado do predicado convencional da tabela de dimensão seja pequeno o suficiente para ser transmitido (*broadcast*) para todos os nós do *cluster*, usando-se para isso uma estrutura exata (*hash map* -  $Dim_{123}$ ). Com isso, todos os nós do *cluster* que contêm os blocos do particionamento da tabela de fatos possuem  $Dim_{123}$  em memória primária. Assim, cada  $j$  bloco ( $Fatos_j$ ) checa, consultando

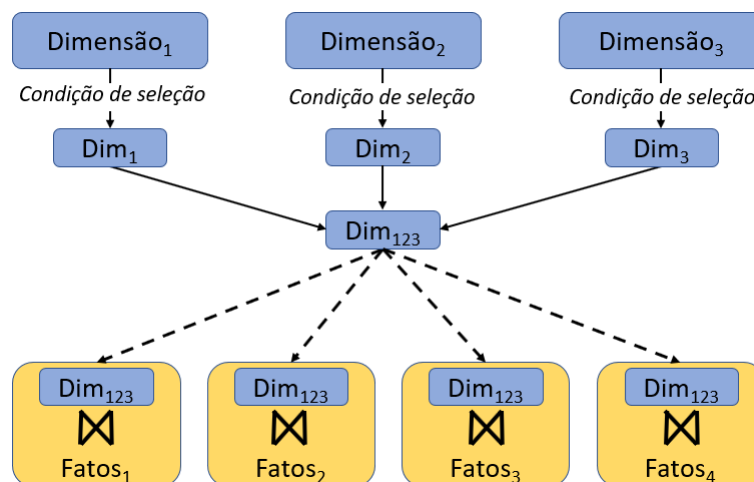
Figura 14 – Técnica *bloom filter cascade join* para o processamento de junção. Fonte: Elaborado pelo autor.



$Dim_{123}$ , se cada elemento satisfaz o predicado convencional. Dessa forma, a junção-estrela é realizada paralelamente, localmente em cada nó.

Apesar da estrutura *hash map* demandar mais espaço de armazenamento do que *bloom filter*, a mesma é exata, ou seja, esta técnica é adequada para ambientes que possuem maior capacidade de memória primária disponível.

Figura 15 – Técnica *broadcast join* para o processamento de junção. Fonte: Elaborado pelo autor.



## 4.8 Considerações finais

Neste capítulo foram descritos conceitos relacionados aos ambientes de processamento paralelo e distribuído. Foram detalhados o sistema de arquivos distribuído HDFS e os *frameworks*

de processamento paralelo e distribuído MapReduce e Spark. Também foram descritas as técnicas bloom filter cascade join e broadcast join, as quais, otimizam a junção-estrela nestes ambientes.

No próximo capítulo, Capítulo 5, é descrito o processo de revisão sistemática realizado nesta dissertação de mestrado.



---

# REVISÃO SISTEMÁTICA

---

## 5.1 Considerações iniciais

Este capítulo tem como objetivo descrever a revisão sistemática realizada para o desenvolvimento das atividades desta dissertação de mestrado. Na seção 5.2 é feita a contextualização da revisão sistemática frente aos objetivos da dissertação. Na seção 5.3 é descrito o planejamento da revisão sistemática. Na seção 5.4 é relatada a forma como foi realizada a condução da revisão sistemática. O capítulo é finalizado na seção 5.5 com as considerações finais.

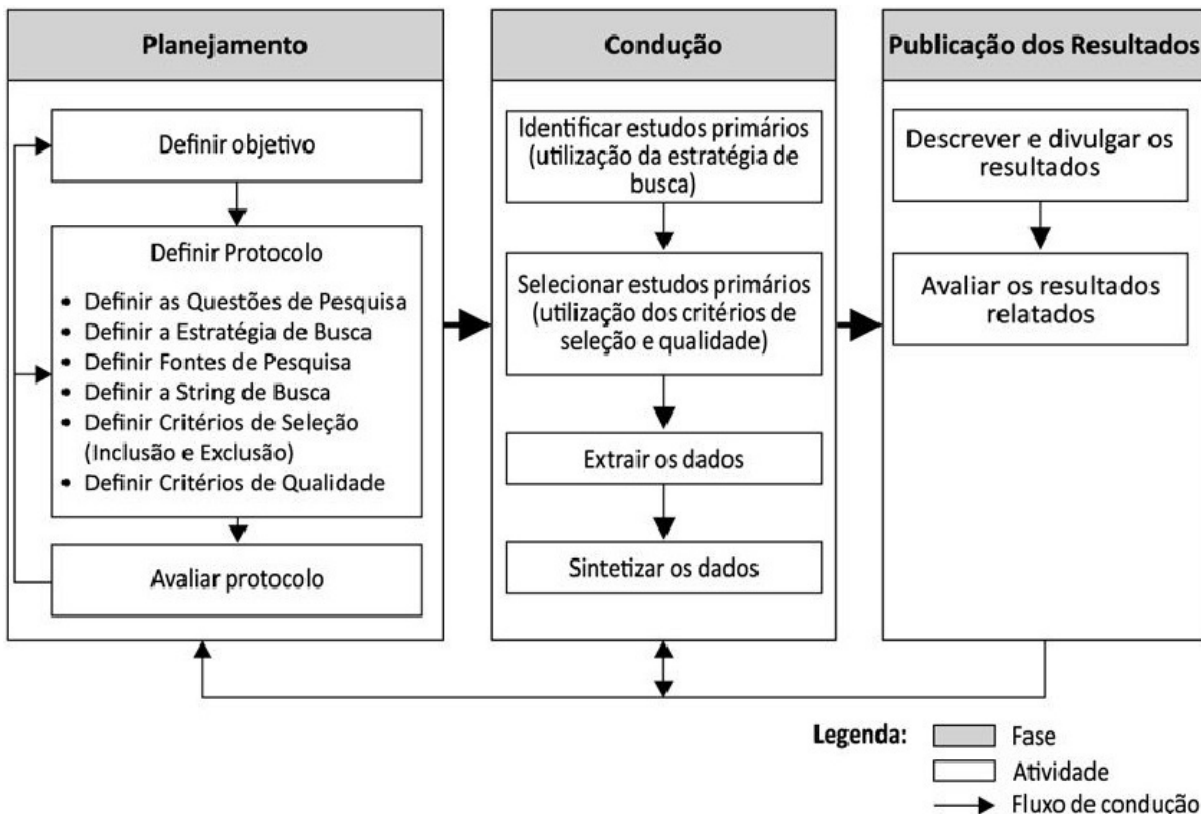
## 5.2 Contextualização

Esta dissertação de mestrado tem como objetivo investigar o processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens em ambientes paralelos e distribuídos. Adicionalmente, a dissertação tem como motivação a manipulação de imagens médicas. Sendo assim, esta dissertação engloba diferentes conceitos, como *data warehouse*, similaridade de imagens, processamento paralelo e distribuído e contexto médico. Em um projeto de pesquisa, é fundamental a realização de um levantamento do estado da arte referente aos conceitos em que o mesmo está inserido. Isto é realizado por meio de uma revisão sistemática.

A revisão sistemática é um método de sumarizar evidências de pesquisa de um modo sistemático, ou seja, seguindo um planejamento em que o mesmo possa ser reproduzido, diminuindo a possibilidade de algo tendencioso. Uma revisão sistemática possui o objetivo de identificar, selecionar, avaliar, interpretar e sumarizar estudos relevantes para o contexto em pesquisa. Com isso, é possível analisar o estado da arte relacionado à dissertação, identificar lacunas para sugerir áreas que podem ser exploradas e detectar evidências de tecnologias de acordo com o contexto da pesquisa (KITCHENHAM; CHARTERS, 2007).

A revisão sistemática descrita neste capítulo foi realizada seguindo um processo com fases e atividades bem definidas, como indicado na literatura (KITCHENHAM; CHARTERS, 2007; NAKAGAWA *et al.*, 2017). A Figura 16 ilustra o fluxo de condução do processo, o qual é flexível em reavaliar as fases, ou seja, ao identificar-se que uma fase necessita de ajustes, a mesma pode ser reavaliada e redefinida.

Figura 16 – Processo de revisão sistemática. Fonte: (NAKAGAWA *et al.*, 2017).



## 5.3 Planejamento

O planejamento é a primeira fase do processo de revisão sistemática e consiste na definição de um protocolo, ou seja, objetivos (seção 5.3.1), questões de pesquisa (seção 5.3.2), identificação dos estudos (seção 5.3.3) e critérios de seleção (seção 5.3.4). O protocolo tem como objetivo reduzir os impactos de vieses que podem existir durante a etapa de execução. Além disso, o planejamento permite direcionar as próximas etapas e compartilhar detalhes da revisão com outros pesquisadores.

### 5.3.1 Objetivos

Os objetivos desta revisão sistemática são:

- Identificar abordagens para junção-estrela em Hadoop;

- Identificar estudos que tratam a similaridade de imagens em Hadoop;
- Identificar estudos de *data warehouses* na área médica em Hadoop;
- Identificar *data warehouses* na área médica que tratam operações convencionais e de similaridade de imagens em Hadoop.

### 5.3.2 Questões de pesquisa

Esta atividade é muito importante para o processo da revisão sistemática, pois permite a definição de questões fortemente relacionadas ao objetivo da revisão. Tais questões fundamentam a análise dos trabalhos selecionados e, portanto, o foco da pesquisa. As questões de pesquisa definidas para a revisão sistemática são:

- **Questão 1:** Como realizar junções-estrela em Hadoop?
- **Questão 2:** Como realizar operações de similaridade de imagens em Hadoop?
- **Questão 3:** Como se caracteriza um *data warehouse* da área médica em Hadoop?
- **Questão 4:** Como realizar operações de junção-estrela e de similaridade de imagens sobre um *data warehouse* em Hadoop?

As questões de pesquisa podem ser estruturadas no seguinte conjunto PICO - População, Intervenção, Comparação e Resultados (*Outcomes*). A população identifica o grupo de interesse pela revisão, a intervenção pontua o que deve ser investigado, a comparação apresenta o que deve ser utilizado como base para comparação do que será investigado e, por fim, os resultados expressam a caracterização da pesquisa (BIOLCHINI *et al.*, 2007; NAKAGAWA *et al.*, 2017):

- **População:** Pesquisadores e médicos com interesse em utilizar *data warehouse* de imagens médicas;
- **Intervenção:** Operações de junção-estrela ou de similaridade em um *data warehouse* de imagens médicas de forma paralela e distribuída;
- **Comparação:** Comparar o desempenho computacional de técnicas que otimizam operações de junção-estrela ou de similaridade em ambientes paralelos e distribuídos;
- **Resultados:** Ganho em desempenho e de funcionalidades analíticas oferecidas.

### 5.3.3 Identificação dos estudos

Nesta seção são documentadas as estratégias de seleção para a busca dos estudos relevantes para a revisão, tais como fontes de busca, palavras-chave, *strings* de busca e estratégias de busca.

### 5.3.3.1 Fontes de busca

A definição das fontes de busca utilizadas para a revisão é guiada por um critério de seleção das fontes que envolve condições e características que as mesmas devem atender. Os critérios observados para a definição das fontes foram: alcance dos estudos (as fontes devem retornar uma quantidade razoável de trabalhos), atualização (os trabalhos retornados devem ser recentes) e disponibilidade (os trabalhos devem ser integralmente acessíveis). De acordo com estes critérios, foram definidas as seguintes fontes de busca:

- IEEEExplore Digital Library<sup>3</sup>;
- Springer<sup>4</sup>;
- ACM Digital Lybrary<sup>5</sup>;
- Elsevier<sup>6</sup>.

A fonte DBLP não foi considerada porque a maioria de suas publicações já estão disponíveis nas fontes mencionadas e suas capacidades de busca são limitadas quando comparadas com as bibliotecas selecionadas (BATISTA *et al.*, 2018).

### 5.3.3.2 Definições prévias

Os idiomas selecionados para serem aceitos pela revisão são o Português, por ser a língua do país de origem desta dissertação de mestrado, e Inglês, por ser a língua utilizada internacionalmente para a publicação de trabalhos científicos.

Como mencionado na seção 5.3.3.1, um dos critérios para seleção de fontes é a busca por estudos recentes. Com isso, foi definido que a busca deve abranger os estudos publicados no período dos últimos 6 anos, ou seja, de 2014 até o momento atual. Este período foi definido devido ao fato do desenvolvimento do processamento paralelo e distribuído ser mais recente atualmente, com a proposta do Spark em 2010, e também pelo fato do estudo descrito em (TEIXEIRA *et al.*, 2015) - uma das bases desta dissertação - ter feito um levantamento de estudos até o ano de 2015, não encontrando estudos correlatos anteriores a essa data e que estejam dentro do contexto da presente pesquisa.

### 5.3.3.3 Palavra-Chave

Os termos comumente utilizados no contexto em que a revisão sistemática pretende investigar definem as palavras-chave em Inglês e seus respectivos sinônimos em parênteses, como:

<sup>3</sup> IEEEExplore Digital Library: <<https://ieeexplore.ieee.org>>

<sup>4</sup> Springer Computer Science: <<https://www.springer.com/ComputerScience>>

<sup>5</sup> ACM Digital Library: <<https://dl.acm.org>>

<sup>6</sup> Computer Science – Elsevier: <<https://www.elsevier.com/physical-sciences/computer-science>>



- *Star Join; Hadoop (Spark, MapReduce);*
- *Similarity operations of images; Hadoop (Spark, MapReduce);*
- *Data Warehouse (DW, OLAP); Medical (medical images, medical area); Hadoop (Spark, Mapreduce);*
- *Data Warehouse (DW, OLAP); Medical (medical images, medical area); Similarity of images; Hadoop (Spark. MapReduce).*

O termo *star join* foi usado ao invés do termo OLAP por ele ser amplamente utilizado na literatura para o processamento de consultas analíticas em ambientes paralelos e distribuídos. Adicionalmente, o termo *Hadoop* foi definido porque observou-se que, na busca, mais artigos fortemente relacionados com a proposta desta dissertação são retornados usando-se essa palavra-chave ao invés de “processamento paralelo e distribuído”.

#### 5.3.3.4 Strings de busca

As strings de busca são definidas para serem executadas nas fontes selecionadas (seção 5.3.3.1) usando-se como base a definição das palavras-chave (seção 5.3.3.3). Com isso, por meio de operações lógicas (*AND* e *OR*) as *strings* são construídas obedecendo as línguas consideradas para a seleção do trabalho (seção 5.3.3.2). As *strings* de busca em Inglês são:

- **IEEEExplore Digital Library**
  - *((star join) AND (hadoop OR spark OR mapreduce));*
  - *((similarity operations images) AND (Hadoop OR Spark OR MapReduce));*
  - *((data warehouse OR DW OR OLAP) AND (medical OR medical images OR medical area) AND (hadoop OR spark OR mapreduce));*
  - *((data warehouse OR DW OR OLAP) AND (medical OR medical images OR medical area) AND (similarity images) AND (hadoop OR spark OR mapreduce)).*
- **Springer**
  - *(star join hadoop);*
  - *(similarity operations images hadoop spark mapreduce);*
  - *(data warehouse medical hadoop);*
  - *(data warehouse medical similarity images hadoop spark mapreduce).*
- **ACM Digital Lybrary**
  - *(star AND join) AND (hadoop OR spark OR mapreduce);*

- *(similarity AND images) AND (hadoop OR spark OR mapreduce);*
- *(data AND warehouse) OR DW OR OLAP) AND (medical OR medical AND images OR medical AND area) AND (hadoop OR spark OR mapreduce);*
- *(data AND warehouse OR DW OR OLAP) AND (medical OR medical AND images OR medical AND area) AND (similarity AND images) AND (hadoop OR spark OR mapreduce).*

- **Elsevier**

- *(star join) AND (hadoop OR spark OR mapreduce);*
- *(similarity images) AND (hadoop OR spark OR mapreduce);*
- *(data warehouse OR DW OR OLAP) AND (medical OR medical images OR medical area) AND (hadoop OR spark OR mapreduce);*
- *(data warehouse OR DW OR OLAP) AND (medical OR medical images OR medical area) AND (similarity images) AND (hadoop OR spark OR mapreduce).*

As *strings* de busca em Português são:

- **IEEEExplore Digital Library**

- *((junção estrela) AND (hadoop OR spark OR mapreduce));*
- *((operações similaridade imagens) AND (hadoop OR Spark OR MapReduce));*
- *((data warehouse OR DW OR OLAP) AND (médico OR imagens médicas OR área médica) AND (hadoop OR spark OR mapreduce));*
- *((data warehouse OR DW OR OLAP) AND (médico OR imagens médicas OR área médica) AND (similaridade imagens) AND (hadoop OR spark OR mapreduce)).*

- **Springer**

- *(junção estrela hadoop);*
- *(operações similaridade imagens hadoop mapreduce);*
- *(data warehouse médico hadoop);*
- *(data warehouse médico similaridade imagens hadoop).*

- **ACM Digital Lybrary**

- *(junção AND estrela) AND (hadoop OR spark OR mapreduce);*
- *(Similaridade AND imagens) AND (hadoop OR spark OR mapreduce);*
- *(data AND warehouse) OR DW OR OLAP) AND (médico OR imagens AND médicas OR área AND médica) AND (hadoop OR spark OR mapreduce);*

- (data AND warehouse OR DW OR OLAP) AND (médico OR imagens AND médicas OR área AND médica) AND (similaridade AND imagens) AND (hadoop OR spark OR mapreduce).

- **Elsevier**

- (junção estrela) AND (hadoop OR spark OR mapreduce);
- (similaridade imagens) AND (hadoop OR spark OR mapreduce);
- (data warehouse OR DW OR OLAP) AND (médico OR imagens médicas OR área médica) AND (hadoop OR spark OR mapreduce);
- (data warehouse OR DW OR OLAP) AND (médico OR imagens médicas OR área médica) AND (similaridade imagens) AND (hadoop OR spark OR mapreduce).

Nas *strings* de busca em Português, manteve-se algumas palavras em Inglês devido ao fato de que essas palavras são comumente usadas ao invés de sua tradução para o Português, por exemplo, *data warehouse*. Portanto, foram traduzidos somente os termos necessários. É importante salientar que buscas que utilizaram as *strings* em Português não retornaram resultado.

### 5.3.4 Critérios de seleção

Nesta atividade da fase de planejamento é definido como se caracteriza a seleção dos trabalhos, uma vez que os mesmos foram retornados das fontes de busca por meio das *strings* de busca. Com isso, são definidos critérios de inclusão, critérios de exclusão e o procedimento para a seleção dos estudos.

#### 5.3.4.1 Critérios de inclusão

O critério de inclusão dos estudos nesta revisão sistemática, visando atender aos objetivos (seção 5.3.1), é:

- Responder pelo menos uma das questões de pesquisa (seção 5.3.2).

#### 5.3.4.2 Critérios de exclusão

Tendo em mente os objetivos desta revisão sistemática, os estudos são excluídos segundo os seguintes critérios:

- Estudos que não respondem a nenhuma das questões de pesquisa;
- Estudos em que o idioma seja diferente do Inglês ou Português;
- Estudos que não sejam completos ou que não estejam disponíveis na íntegra.

### 5.3.4.3 Procedimento de seleção

O procedimento de seleção dos estudos é definido em 3 etapas:

- **Seleção Inicial:** Com base nas palavras-chave definidas na seção 5.3.3.3, são identificadas as *strings* de busca descritas na seção 5.3.3.4. Na sequência, as *strings* de busca são executadas pelas fontes de busca selecionadas na seção 5.3.3.1. Os estudos retornados são avaliados de acordo com a leitura do título e *abstract*. Se o estudo atende a algum critério de inclusão (seção 5.3.4.1), o mesmo é selecionado para a seleção final. Para os casos nos quais os estudos se enquadram em algum critério de exclusão, eles são excluídos (seção 5.3.4.2).
- **Seleção Final:** Nesta fase é realizada a leitura completa dos estudos que atenderam ao critério de inclusão na seleção inicial. Essa fase tem como objetivo identificar quais artigos ainda mantêm o critério de inclusão. Esses artigos são utilizados para extração de informações.
- **Extração de informações:** Ao realizar a leitura completa do estudo na seleção final e o mesmo manter o critério de inclusão, é realizada a extração de informações com o objetivo de sintetizar (seção 5.4.2) os principais conteúdos de forma a atender aos objetivos da revisão.

## 5.4 Condução

Esta fase consiste na execução das *strings* de busca nas fontes selecionadas, seleções dos estudos e a realização da síntese dos dados, de acordo com o protocolo definido na fase de planejamento (seção 5.3). O fluxo detalhado do processo desta fase é ilustrado na Figura 17, sendo que, a medida em que o fluxo segue, a quantidade de artigos diminui. Ao final, os estudos filtrados estão mais próximos de atender aos objetivos da revisão sistemática. Este fluxo é descrito em mais detalhes nas seções 5.4.1 e 5.4.2.

### 5.4.1 Seleção dos estudos

Processando as *strings* de busca nas fontes selecionadas, restringindo o ano de publicação (seção 5.3.3.2) e tratando estudos duplicados, foram retornados 147 estudos (Figura 18).

Os artigos retornados foram analisados de acordo com a seleção inicial, de forma que, para cada estudo foram lidos o título e o *abstract*, visando identificar estudos aceitos ou rejeitados. Dos 147 estudos analisados, 19 passaram pela seleção inicial, sendo que 128 estudos foram excluídos.

Para a seleção final, os 19 estudos aceitos na seleção inicial foram lidos por completo, os quais foram reavaliados de acordo com os critérios de inclusão e exclusão. Restaram 13 estudos para serem analisados na síntese dos dados, ou seja, 6 estudos foram excluídos.

Figura 17 – Fluxo da fase de Condução. Fonte: Elaborado pelo autor.

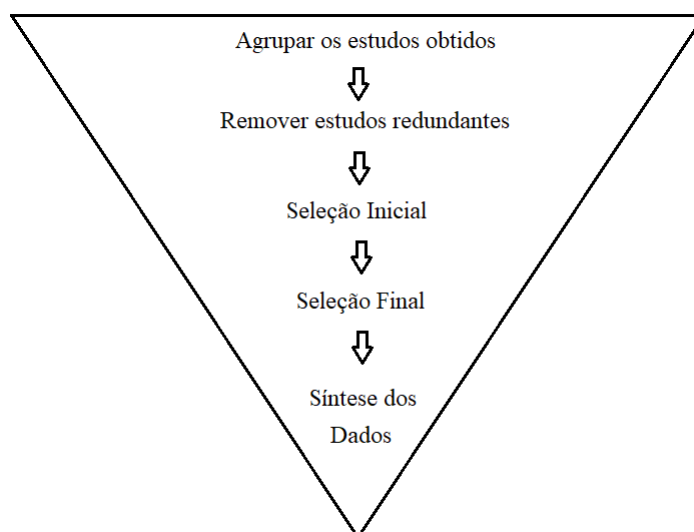
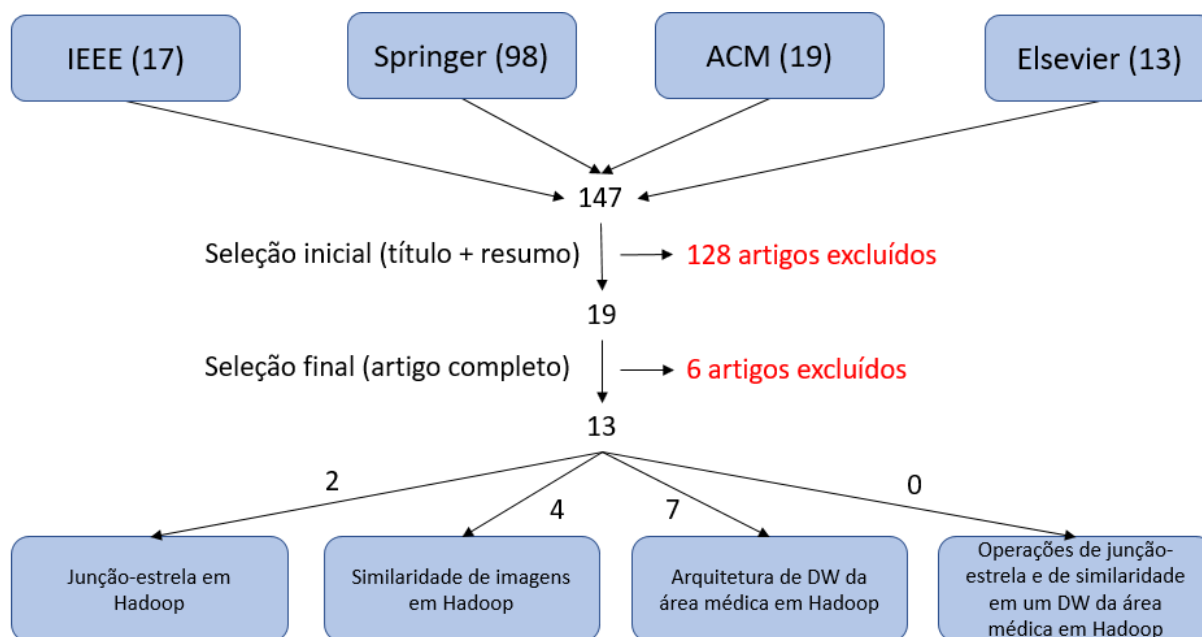


Figura 18 – Condução da revisão sistemática. Fonte: Elaborado pelo autor.



### 5.4.2 Síntese dos dados

Esta seção tem o objetivo de combinar os dados extraídos de cada estudo com foco em responder às questões de pesquisa definidas na seção 5.3.2. Como nenhum trabalho encontrado possui os objetivos específicos desta dissertação de mestrado, ou seja, nenhum artigo relacionado possui características muito semelhantes à presente dissertação, os estudos que foram aceitos na seleção final foram agrupados em temas importantes que envolvem os principais assuntos desta dissertação de mestrado (Figura 18). Esses agrupamentos são detalhados a seguir, com as contribuições dos estudos em seus respectivos temas.

#### 5.4.2.1 Junção-estrela em Hadoop

O objetivo deste grupo de estudos é identificar técnicas para o processamento eficiente de junção-estrela utilizando processamento paralelo e distribuído (BRITO *et al.*, 2016; BRITO *et al.*, 2020). Dentro deste contexto, a proposta descrita em (BRITO *et al.*, 2016) propõe duas técnicas para a otimização da junção-estrela utilizando o *framework* Spark: *Spark Bloom Filter Cascade Join* (SBFCJ) e *Spark Broadcast Join* (SBJ). SBFCJ utiliza a estrutura probabilística *bloom filter* (seção 4.6) para filtrar a tabela de fatos antes da realização da junção-estrela em cascata. Por outro lado, SBJ utiliza a estrutura exata *hash map* (seção 4.7) e assume que todos os predicados de dimensão são pequenos o suficiente para serem enviados para todos os nós do *cluster*. Com isso, a junção-estrela é realizada localmente em cada nó.

Os testes de desempenho realizados em (BRITO *et al.*, 2016) compararam as estratégias propostas com trabalhos relacionados anteriores na literatura, os quais utilizam diversas técnicas para otimizar o processamento da junção-estrela em ambientes paralelos e distribuídos (AFRATI; ULLMAN, 2010; ZHANG; WU; LI, 2013; HAN *et al.*, 2011; TAO *et al.*, 2013; BLANAS *et al.*, 2010). Tanto SBFCJ quanto SBJ demonstraram ganho de desempenho em termos de comunicação da rede (*shuffle*) e sobrecarga de disco (*disk spill*). Além disso, os experimentos de (BRITO *et al.*, 2016) também compararam SBFCJ com SBJ. Os resultados obtidos demonstraram que a utilização da técnica SBFCJ é indicada para ambientes com pouca memória primária, já que a estrutura *bloom filter* ocupa menos espaço, enquanto que a estrutura *hash map* demanda mais espaço e, com isso, pode não caber em memória primária e causar *disk spill*. Porém, ambientes que possuem memória primária razoável são adequados para a utilização de SBJ, já que *hash map* realiza a junção de modo mais direto, enquanto que SBFCJ realiza a junção-estrela em cascata.

O estudo descrito em (BRITO *et al.*, 2020) propõe uma estratégia para processar junção-estrela utilizando um índice bitmap de junção (GOLFARELLI; RIZZI, 2018), chamado *distributed Bitmap Join Index* (dBJI), no ambiente paralelo e distribuído. São propostas diferentes técnicas com o objetivo de demonstrar que, para consultas de junção-estrela que possuem predicados de alta seletividade, ou seja, predicados que possuem poucos elementos que os satisfazem, é indicado o uso de índices ao invés de se realizar a varredura completa das tabelas. Os resultados de desempenho realizados comprovaram o objetivo da pesquisa, sendo que o uso de índices em consultas de alta seletividade proveu ganho de desempenho de 59% a 88% quando comparado com a varredura completa das tabelas.

Portanto, as técnicas SBFCJ e SBJ introduzidas em (BRITO *et al.*, 2016) e as técnicas propostas em (BRITO *et al.*, 2020) proveem bons resultados para a otimização do processamento da junção-estrela. A utilização dos índices é indicada para consultas frequentes e com alta seletividade. Por um lado, o uso de índices provê melhor desempenho para essas consultas. Por outro lado, índices têm que ser armazenados e mantidos frente a cada inserção realizada no ambiente de *data warehousing*, introduzindo custos adicionais. No contexto desta dissertação

de mestrado, considera-se a flexibilidade de consultas analíticas estendidas com predicado de similaridade, ou seja, considera-se cenários nos quais é difícil de prever os tipos de consulta OLAP que são executadas e também a frequência de ocorrência de um determinado predicado de alta seletividade. Isso é usualmente comum em tarefas exploratórias de análise de dados. Nessas tarefas, dificilmente se requer o reúso de índices existentes (TRAINA *et al.*, 2019).

Nesta dissertação de mestrado, as técnicas SBFCJ e SBJ são consideradas o estado-da-arte para a otimização da junção-estrela sem o uso de índices e serão consideradas como os trabalhos relacionados usados nos testes de desempenho descritos nos capítulos 6 e 7, respectivamente.

#### 5.4.2.2 Similaridade de imagens em Hadoop

Nesta seção são agrupados os estudos que envolvem operações de similaridade de imagens utilizando processamento paralelo e distribuído (GIANGRECO; KABARY; SCHULDT, 2014; LI *et al.*, 2017; NGUYEN *et al.*, 2016; NGUYEN; HUH, 2017). A mesma possui o objetivo de resumir técnicas para executar operações de similaridade de imagens de modo eficiente em termos de desempenho computacional.

Em (GIANGRECO; KABARY; SCHULDT, 2014), utiliza-se o *framework* MapReduce para o processamento de busca por similaridade de imagens. Apesar da utilização do ambiente paralelo e distribuído oferecer bons resultados, este estudo não aborda otimizações específicas para o cálculo de similaridade, considerando os cálculos de distância para cada conjunto de características.

Em (LI *et al.*, 2017), é utilizado o método de indexação *hashing* sensível à localidade - *Locality Sensitive Hashing* (LSH) - para processar operações de similaridade de imagens no *framework* Spark. O LSH possui a característica de usar uma família de funções *hash* que atende propriedades específicas de modo que, ao mapear os objetos na função *hash*, a probabilidade de objetos similares colidirem seja alta. Com isso, o conjunto de objetos que devem ser processados para verificar a similaridade diminui. A execução desta estratégia no Spark diminui os dados transmitidos pela rede na fase de *shuffle*, aumentando o desempenho computacional das operações de similaridade.

Apesar do método de indexação LSH ser válido, é complexo definir de modo eficiente o conjunto de funções *hash* que atendam à alta probabilidade de colisão para objetos similares (NGUYEN *et al.*, 2016). Com isso, em (NGUYEN *et al.*, 2016; NGUYEN; HUH, 2017) é proposto o processamento de operações de similaridade de imagens utilizando o MapReduce e a estrutura de indexação *VP-tree*. A *VP-tree* realiza a poda no espaço de busca das consultas por similaridade. Considerando esta poda e o processamento paralelo e distribuído com o *framework* MapReduce, tal proposta possui um alto desempenho computacional. Para montar a estrutura, escolhe-se, aleatoriamente, um *vantage point*, ou seja, um objeto, no caso, uma imagem. Na sequência, calcula-se a distância deste ponto com todos os outros e define-se a mediana de todas as distâncias como o raio do *vantage point*. Como resultado, a árvore ramifica-se por meio dos

pontos que estão dentro do raio e dos que estão fora, sendo aplicada de forma recursiva para todos os objetos.

Além disso, o estudo de (NGUYEN; HUH, 2017) propõe a *distance cache*, a qual armazena em *cache* as distâncias entre objetos calculadas anteriormente. Com isso, se um cálculo futuro envolver estes objetos o valor já está armazenado, elimina-se a necessidade de realizar novamente a operação cara de cálculo de distância. O uso da *distance cache* somada ao processamento utilizando o método *VP-tree* (NGUYEN *et al.*, 2016) possui um ganho de desempenho ainda maior. Este desempenho aumenta conforme o aumento do conjunto de objetos, pois a chance do cálculo da distância já estar armazenado em *cache* é maior.

É importante pontuar que a estrutura de indexação *VP-tree* é comumente implementada para ser processada em memória principal, sendo a mesma estática. Já a estrutura de indexação *Omni-family*, utilizada em (TEIXEIRA *et al.*, 2015) é uma estrutura de indexação métrica baseada em disco e dinâmica, ganhando em termos de desempenho de seus principais concorrentes, como *R-tree* e *Slim-tree* (Filho *et al.*, 2001; TRAINA *et al.*, 2007). Desse modo, esta dissertação de mestrado investiga a utilização da técnica *Omni* para otimizar as operações de similaridade de imagens.

#### 5.4.2.3 Arquitetura de um ambiente de data warehousing da área médica processado em Hadoop

O objetivo deste agrupamento é realizar um levantamento de possíveis arquiteturas para armazenar um *data warehouse* voltado para a área médica, e a partir disto, executar consultas analíticas sobre os dados armazenados. Como a proposta desta dissertação de mestrado destina-se ao processamento paralelo e distribuído, utilizando tecnologias Hadoop, as arquiteturas levantadas estão inseridas neste contexto (DASH *et al.*, 2019; ISTEPHAN; SIADAT, 2015; ISTEPHAN; SIADAT, 2016; KUO *et al.*, 2015; RAJA; SIVASANKAR, 2014; SEBAA *et al.*, 2017; SEBAA *et al.*, 2018).

Em (DASH *et al.*, 2019) são abordadas técnicas para que diferentes dados de diferentes fontes do contexto da área médica sejam armazenados, processados e analisados. Portanto, ambientes paralelos e distribuídos são propostos com a utilização do sistema de arquivos distribuído HDFS para o armazenamento eficiente destes dados. A partir disso, são sugeridos *frameworks*, que se relacionam com o HDFS, para o processamento de dados, como o Spark. Também são listadas uma série de ferramentas, baseadas em sistemas paralelos e distribuídos, para a realização de análises específicas da área médica, como análise de dados genômicos *SparkSeq*, *SeqWare*, *CloudBurst*, *Myrna*, entre outras.

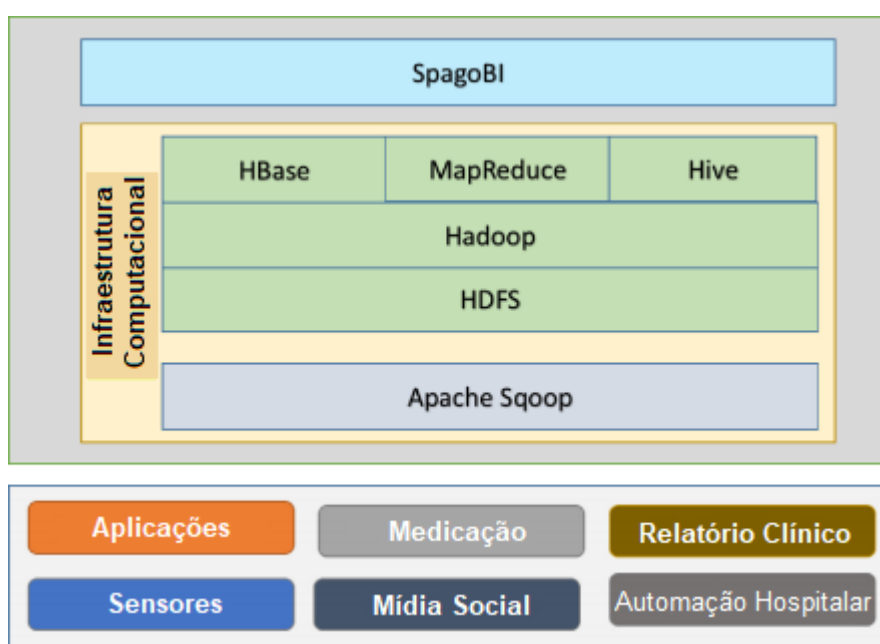
Os estudos descritos em (ISTEPHAN; SIADAT, 2015; ISTEPHAN; SIADAT, 2016) utilizam como dados estruturados os dados convencionais de um DW da área médica e consideram como dados não estruturados os dados de imagens médicas. A partir dos dados de imagens médicas, os estudos permitem a busca destas imagens baseada em seu conteúdo - CBMIR.



As buscas são realizadas utilizando MapReduce. O modo de armazenamento dos dados não é detalhado, sendo que apenas é informado o uso dos serviços da *Microsoft Azure* e o *Azure HDInsight*.

Os próximos estudos analisados utilizam tecnologias Hadoop como HBase e Hive para armazenar os dados. Como os estudos envolvem processamento paralelo e distribuído, os mesmos utilizam o sistema de arquivos distribuído HDFS e os *frameworks* MapReduce ou Spark. A tecnologia Hadoop possui seu funcionamento baseado neste sistema. A [Figura 19](#) ilustra o sistema desta arquitetura, com as tecnologias citadas e suas camadas.

Figura 19 – Exemplo de arquitetura Hadoop para DW da área médica. Fonte: Adaptado de ([RAJA; SIVASANKAR, 2014](#)).



Os estudos descritos em ([KUO et al., 2015](#); [RAJA; SIVASANKAR, 2014](#); [SEBAA et al., 2017](#)) propõem um sistema de armazenamento utilizando o HBase, sendo este um banco de dados distribuído *open-source*, NoSQL e orientado à coluna, preparado para armazenar e processar um grande volume de dados ([GEORGE, 2011](#)). O HBase é executado na estrutura do Apache Hadoop. Ao ter uma integração fácil com o Hadoop, pode ser utilizado o MapReduce ou Spark para o processamento distribuído das tarefas. Além disso, o estudo de ([KUO et al., 2015](#)) utiliza o Apache Phoenix para dar suporte às consultas *On-line Transaction Processing* (OLTP), ainda usando o HBase como base para o processamento.

Os estudos introduzidos em ([RAJA; SIVASANKAR, 2014](#); [SEBAA et al., 2017](#); [SEBAA et al., 2018](#)) propõem a utilização do Apache Hive, o qual é um sistema de *data warehouse* executado sobre o Apache Hadoop. Tal sistema oferece suporte à consulta e à análise de dados com uma interface semelhante ao SQL (*HiveQL*), abstraindo do usuário conhecimentos de MapReduce ou Spark, pois a partir da definição da estrutura e da consulta *HiveQL* o processamento paralelo é realizado. Além disso, consultas podem ser executadas a partir dos dados

armazenados no HBase. Porém, a gama de consultas para a qual o Hive oferece suporte é limitada. Adicionalmente, o estudo descrito em (SEBAA *et al.*, 2018) explora também o fato de que o Hive não possui definição de chaves primárias e estrangeiras. Dentro deste contexto, é proposto o uso de dois conceitos para tratar essa lacuna: *partitions* e *buckets*. *Partitioning* em Hive é uma técnica para criar partições de uma tabela, sendo que essas partições são definidas a partir do valor de um atributo. Com isso, os registros com o mesmo valor são armazenados na mesma partição. Na técnica de *bucketing*, o número de partições é fixo e definido por meio de uma função *hash* de acordo com uma coluna da tabela.

Em (RAJA; SIVASANKAR, 2014) é proposta a arquitetura ilustrada na Figura 19. A partir de provedores de informações médicas, é utilizado o Apache Sqoop, o qual é um aplicativo que transfere os dados de um banco de dados relacional para o ambiente Hadoop. A partir disso, no estudo é proposto tanto a utilização do HBase e o processamento dos dados realizado em MapReduce quanto o uso do Hive. São comparados os tempos de alimentação da base e execução de consultas utilizando o MapReduce *versus* um banco de dados relacional. Os resultados de desempenho demonstraram que o processamento MapReduce obteve melhor desempenho, especialmente com o aumento do número de registros envolvidos. Também foram realizados experimentos do processamento MapReduce comparado ao Hive, sendo que o Hive obteve um melhor desempenho.

Nesta seção foram detalhadas as principais tecnologias utilizadas em uma arquitetura para um DW da área médica voltado para o processamento paralelo e distribuído, segundo os estudos relacionados disponíveis na literatura, os quais mostram bons resultados para o armazenamento e processamento de dados considerando este ambiente. No contexto da proposta desta dissertação de mestrado não utiliza-se Hive, pois o processamento paralelo e distribuído envolvendo CBMIR é muito específico e não possui suporte do Hive (ISTEPHAN; SIADAT, 2016), adicionado ao fato de que as junções que Hive processa possuem limitações e provêm pior desempenho do que outras técnicas (BRITO, 2017; TAO *et al.*, 2013). Portanto, esta dissertação considera um *data warehouse* de imagens da área médica, em um ambiente paralelo e distribuído, armazenado pelo sistema de arquivos distribuído HDFS e processado pelo *framework* Spark, o qual possui melhor desempenho e permite maior controle de processamento (BRITO *et al.*, 2016). Estas definições consideram tanto o contexto flexível e genérico das consultas de interesse desta dissertação quanto à especificidade da área médica.

#### 5.4.2.4 Operações de junção-estrela e de similaridade em um ambiente de data warehousing da área médica em Hadoop

O objetivo deste agrupamento consiste em investigar estudos relacionados que mais se aproximam com a proposta desta dissertação de mestrado. De acordo com as pesquisas realizadas com base na revisão sistemática, nenhum trabalho correlato existente na literatura investiga conjuntamente operações analíticas com predicado de similaridade executados contra um DW da

área médica em Hadoop.

## 5.5 Considerações finais

Este capítulo apresentou uma revisão sistemática sobre os principais conceitos que esta dissertação de mestrado aborda. Pode-se perceber que na literatura estudada existem estudos correlatos que investigam a realização da junção-estrela em ambientes paralelos e distribuídos (seção 5.4.2.1) e a execução de consultas de similaridade sobre imagens considerando esses ambientes (seção 5.4.2.2), bem como o uso de arquiteturas Hadoop para armazenar e processar dados de DW da área médica. A Tabela 2 ilustra, nas linhas em branco, os conceitos envolvidos em cada um dos agrupamentos descritos neste capítulo.

No melhor do conhecimento do autor desta dissertação de mestrado, nenhum desses trabalhos considera todos esses aspectos conjuntamente, ou seja, o processamento eficiente da junção-estrela sobre um DW de imagem médicas que oferece suporte para similaridade, considerando ambientes paralelos e distribuídos. Na Tabela 2, destaca-se na linha em cinza a inovação e contribuição desta dissertação para o avanço do estado da arte neste contexto.

Tabela 2 – Conceitos dos temas de pesquisa abordados na revisão sistemática. Fonte: Elaborado pelo autor.

	DW	CBIR	Hadoop
Junção-estrela em Hadoop	X		X
Consultas por similaridade em Hadoop		X	X
Arquitetura DW da área médicas em Hadoop	X		X
Operações de junção-estrela e de similaridade sobre DW de imagens médica em Hadoop	X	X	X

O próximo capítulo, Capítulo 6, propõe o método *BloOmnImg*, o qual integra as técnicas *bloom filter cascade join* e *Omni* para otimizar, respectivamente, a junção-estrela e o cálculo de similaridade durante o processamento de consultas analíticas estendidas com predicado de similaridade de imagens.



---

## MÉTODO *BloOmnImg*

---

### 6.1 Considerações iniciais

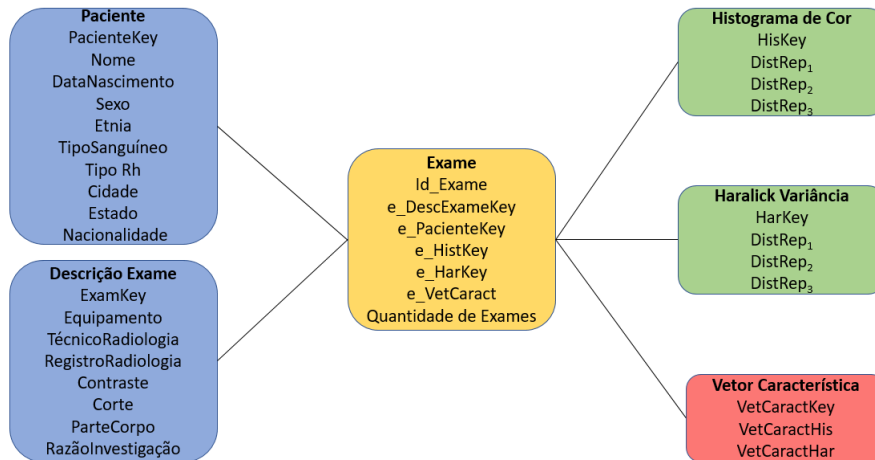
Neste capítulo é introduzido o método proposto *BloOmnImg*, o qual integra as técnicas *bloom filter cascade join* e *Omni* para processar eficientemente consultas analíticas estendidas com predicado de similaridade em Spark. Na seção 6.2 são detalhadas as características do DW de imagens e das consultas de interesse. Na seção 6.3 é detalhado o método proposto, em termos de suas características e de seu algoritmo. Na seção 6.4 é feita a validação do *BloOmnImg*. O capítulo é finalizado na seção 6.5, com as considerações finais.

### 6.2 Características do DW de imagens e consultas de interesse

Na Figura 20 é ilustrado o esquema-estrela do DW de imagens, chamado DWbase, considerado ao longo dos Capítulos 6 e 7. Este esquema consiste em uma versão simplificada do esquema-estrela da Figura 8, o qual enquadra-se no esquema EBM3 (Figura 9c). Adicionalmente, ele usa cores diferentes para facilitar o entendimento da proposta dos métodos. No projeto do DWbase, são incluídas 2 tabelas de dimensão convencional coloridas em azul: *Paciente* e *Descrição Exame*. Também são representadas 2 tabelas *Camada Perceptual* coloridas em verde: *Histograma de Cor* e *Haralick Variância*, cada qual contendo as distâncias entre cada imagem e 3 elementos representativos definidos pela técnica *Omni*. A tabela que contém o vetor de característica de cada camada perceptual considerada é chamada de *Vetor Característica*, e esta colorida em vermelho. A tabela de fatos é a tabela *Exame*, a qual está colorida em amarelo.

Com relação às consultas analíticas estendidas com predicado de similaridade de imagens, elas possuem dois tipos de predicado:

Figura 20 – Esquema-estrela do DW de imagens utilizado como base para a descrição dos métodos propostos, chamado de DWbase. Fonte: Elaborado pelo autor.



- Predicado convencional: composto por condições de seleção, sendo cada condição definida sobre um atributo armazenado em uma tabela de dimensão convencional.
- Predicado de similaridade: composto por uma operação de similaridade e pelas camadas perceptuais consideradas.

Por exemplo, na consulta “Liste a quantidade de imagens similares a uma dada imagem, para pacientes do sexo feminino diagnosticadas com câncer de mama e as camadas perceptuais de cor e textura”, tem-se:

- Condições de seleção: pacientes do sexo feminino e câncer de mama. O atributo “Sexo” (*feminino*) está presente na tabela de dimensão convencional *Paciente*, enquanto que os atributos “ParteCorpo” (*mama*) e “RazãoInvestigação” (*câncer*) são encontrados na tabela de dimensão convencional *Descrição Exame*.
- Operação de similaridade: uso da operação de *range query* para resolver a similaridade.
- Camadas perceptuais consideradas: cor e textura. Essas camadas perceptuais referem-se às tabelas de dimensão de imagens *Histograma de Cor* e *Haralick Variância*, as quais processam a etapa de filtragem da técnica Omni. Adicionalmente, as camadas perceptuais também englobam a tabela de dimensão de imagem *Vetor Característica*, a qual visa processar a etapa de refinamento da técnica Omni.

### 6.3 Descrição do método *BloOmniImg*

Nesta seção as técnicas *bloom filter cascade join* (seção 4.6) e Omni (seção 3.3.3.2) são utilizadas para otimizar a junção-estrela e o cálculo de similaridade, respectivamente. Propõe-se o método *BloOmniImg* (acrônimo para *Bloom Filter Omni for analytical Image queries*), o qual

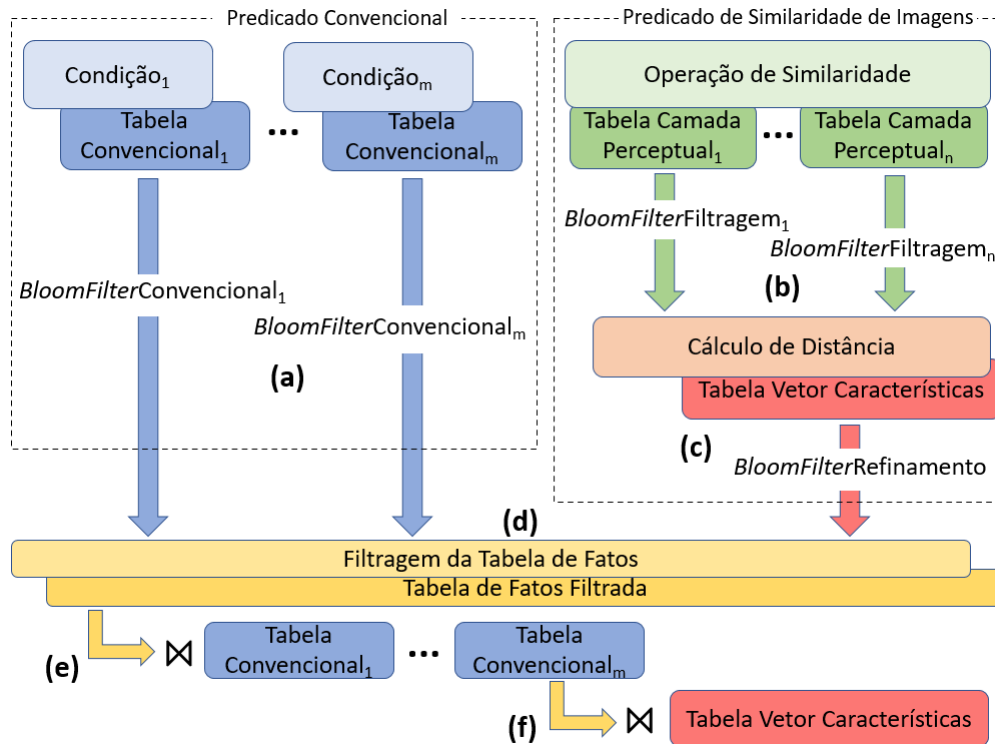
integra essas técnicas para a execução eficiente, em Spark, de consultas analíticas estendidas com predicado de similaridade sobre o DWbase ilustrado na seção 6.2.

### 6.3.1 Método

Na Figura 21 é ilustrada uma visão geral do método *BloOmnlmg* usando o mesmo esquema de cores de DWbase. *BloOmnlmg* integra as técnicas *bloom filter cascade join* e *Omni* da seguinte forma. As condições de seleção são aplicadas para cada tabela de dimensão convencional  $i$  ( $1 \leq i \leq m$ ) envolvida no predicado convencional. *BloOmnlmg* armazena e transmite, para todos os nós do *cluster*, os resultados de cada tabela convencional  $i$  nas estruturas *BloomFilterConvencional<sub>i</sub>* (Figura 21a). Para o predicado de similaridade, a etapa de filtragem da técnica *Omni* é aplicada em cada tabela  $j$  ( $1 \leq j \leq n$ ) referente a cada camada perceptual envolvida. Com isso, os elementos candidatos de cada camada perceptual são armazenados nas estruturas *BloomFilterFiltragem<sub>j</sub>*, as quais são transmitidas para todos os nós do *cluster* (Figura 21b). Na sequência, são realizadas as operações de cálculo de distância usando os resultados candidatos e a tabela *Vetor Características* para eliminar falsos positivos (Figura 21c). Os resultados gerados são aqueles que atendem ao predicado de similaridade, e são armazenados na estrutura *BloomFilterRefinamento*.

Já que a estrutura *bloom filter* é probabilística e não exata, ela não pode ser utilizada para realizar a junção-estrela diretamente. Em contrapartida, essa estrutura pode ser utilizada para filtrar a enorme tabela de fatos antes da realização da junção em cascata. Dentro desse contexto, todas as estruturas *BloomFilterConvencional<sub>i</sub>* e a estrutura *BloomFilterRefinamento* são utilizadas para filtrar a tabela de fatos (Figura 21d). Como resultado, a junção-estrela estendida é realizada, em cascata, entre a tabela de fatos filtrada e as tabelas *Convencional<sub>i</sub>* envolvidas na consulta (Figura 21e). Por fim, a última junção é realizada com a tabela *Vetor Característica* e o resultado da consulta é computado (Figura 21f).

Vale pontuar que não se faz necessária uma operação exata entre os elementos candidatos ao predicado de similaridade, armazenados em cada uma das estruturas *BloomFilterFiltragem<sub>j</sub>*, e os cálculos de similaridade, armazenados na estrutura *BloomFilterRefinamento*. Isso deve-se ao fato de que os cálculos de similaridade eliminam falsos positivos. Em detalhes, ao se utilizar a estrutura *bloom filter* para relacionar a etapa de filtragem com a etapa de refinamento, é possível que esta estrutura gere mais falsos positivos, dependendo dos parâmetros de inicialização. Isto é chamado de erro  $p$  (Equação 4.1). Porém, mesmo que haja mais falsos positivos, os mesmos são eliminados com os cálculos de similaridade. Portanto, quanto maior a probabilidade de erro  $p$ , mais falsos positivos são identificados para o cálculo de similaridade, influenciando no desempenho de processamento da consulta.

Figura 21 – Visão geral do método *BloOmniImg*. Fonte: Elaborado pelo autor.

### 6.3.2 Algoritmo

O Algoritmo 1 exemplifica como *BloOmniImg* para executa a consulta  $Q$ : “Quantas imagens são similares à uma imagem  $s_q$  com raio  $r_q$ , considerando a camada perceptual histograma de cor e pacientes do sexo feminino?”. Como entrada do algoritmo, tem-se todas as tabelas envolvidas na consulta, o centro de consulta e o raio de consulta. Como saída, o algoritmo gera a resposta de  $Q$ . Para a execução da consulta é considerado o DWbase descrito na Figura 20.

RDDs são criados para cada tabela utilizada na execução do predicado convencional (linha 1), de similaridade (linhas 4 e 7) e tabela de fatos (linha 11). O predicado convencional é filtrado e armazenado na estrutura *BloomFilterConventional*, a qual é transmitida para todos os nós do *cluster* nas linhas 2 e 3 (Figura 21a). Para processar o predicado de similaridade, a etapa de filtragem da técnica Omni é aplicada e os elementos candidatos à resposta são transmitidos para todos os nós do *cluster* na estrutura *BloomFilterFiltragem* (linhas 5 e 6 - Figura 21b). A etapa de refinamento da técnica Omni é aplicada para todos os elementos candidatos, eliminando falsos positivos. Assim, o resultado do predicado de similaridade é armazenado e transmitido pela estrutura *BloomFilterRefinamento*, nas linhas 8, 9 e 10 (Figura 21c).

*BloOmniImg* utiliza as estruturas probabilísticas *BloomFilterConvencional* e *BloomFilterRefinamento* para filtrar a tabela de fatos *Exame* e preparar as chaves estrangeiras de *Paciente* e *VetCaract* para a junção-estrela estendida (linhas 12 e 13 - Figura 21d). Com isso, as junções-estrela em cascata são realizadas com *Paciente* (linha 14 - Figura 21e) e *VetCaract* (linha 15 - Figura 21f). O número total de imagens que obedecem ao predicado convencional e de



similaridade de imagens é calculado na linha 16.

---

**Algoritmo 1** – *BloOmnImg*


---

**Input** : *Paciente, Histograma de Cor, Vetor Característica, Exame,  $s_q$  e  $r_q$*

**Output** : resultado de  $Q$

```

1  $RDD_{Paciente} = Paciente$ 
2  $RDD_{Paciente}.filter(sexo = 'feminino').mapToPair(PacienteKey, null)$ 
3  $BloomFilterConvencional = broadcast(RDD_{Paciente}.collect())$ 
4  $RDD_{His} = Histograma\ de\ Cor$ 
5  $RDD_{His}.filter(Filtragem(DistRep_1, DistRep_2, DistRep_3, s_q, r_q)).mapToPair(hisKey, null)$ 
6  $BloomFilterFiltragem = broadcast(RDD_{His}.collect())$ 
7  $RDD_{VetCaract} = Vetor\ Caracteristica$ 
8  $RDD_{VetCaract}.filter(if(HashMapFiltragem.containsKey(vetCaractKey))then$ 
    $return(Refinamento(vetCaractHis, s_q, r_q))$ 
9  $RDD_{VetCaract}.mapToPair(vetCaractKey, null)$ 
10  $BloomFilterRefinamento = broadcast(RDD_{VetCaract}.collect())$ 
11  $RDD_{Exame} = Exame$ 
12  $RDD_{Exame}.filter(BloomFilterConvencional.contains(e_PacienteKey) \cap$ 
    $BloomFilterRefinamento.contains(e_vetCaractKey))$ 
13  $RDD_{Exame}.mapToPair(e_PacienteKey, [e_vetCaractKey, quantidade\ exames])$ 
14  $RDD_{Juncao} = RDD_{Exame}.join(RDD_{Paciente}).mapToPair(e_vetCaractKey, quantidade\ exames)$ 
15  $RDD_{Juncao} = RDD_{Juncao}.join(RDD_{VetCaract}).mapToPair(1, quantidade\ exames)$ 
16  $resultadoQ = RDD_{Juncao}.reduceByKey(v_1 + v_2)$ 

```

---

*BloOmnImg* processa cada tabela do esquema-estrela como um RDD. Ele aplica os filtros convencionais e de imagem sobre os RDDs correspondentes usando a operação *filter*, e armazena os dados nas estruturas *bloom filter* por meio das operações *mapToPair* e *collect*. Adicionalmente, *BloOmnImg* filtra a tabela de fatos de acordo com o predicado convencional e de similaridade por meio da operação *filter*. Por fim, a junção-estrela estendida é realizada em cascata por meio da operação *join*, *mapToPair* e *reduceByKey*.

## 6.4 Validação

A validação do método proposto *BloOmnImg* foi realizada por meio de testes de desempenho experimentais e configurações de consultas com potencial semântico dentro do contexto da área médica. Primeiramente, é descrito o ambiente de teste e, na sequência, os resultados obtidos.

### 6.4.1 Ambiente de teste

**Dados.** O DW de imagens da área médica da Figura 20 foi criado e povoado com dados gerados pela ferramenta *ImgDW Generator* (ROCHA; CIFERRI, 2018). A ferramenta gerou dados sintéticos para as tabelas de dimensão convencionais e dados reais para as tabelas de dimensão de imagens. Os fatos da tabela de fatos foram gerados sinteticamente.

Para a realização dos experimentos de *BloOmnImg* foram definidos 2 volumes de dados distintos. O primeiro volume de dados (Volume 1) considera 300 mil pacientes relacionados com 3 milhões de imagens e exames médicos. Já o segundo volume de dados (Volume 2) considera 2 milhões de pacientes relacionados com 20 milhões de imagens e exames médicos. A Tabela 3 detalha o volume de tuplas em cada tabela da Figura 20.

Tabela 3 – Volume de dados do DW de imagens projetado para os experimentos de *BloOmnImg*. Fonte: Elaborado pelo autor.

Tabela	Volume 1	Volume 2
Paciente	300 mil	2 milhões
Descrição Exame	3 milhões	20 milhões
Histograma de Cor	3 milhões	20 milhões
Haralick Variância	3 milhões	20 milhões
Vetor Característica	3 milhões	20 milhões
Exame	3 milhões	20 milhões

**Consultas.** As consultas analíticas estendidas com predicado de similaridade foram definidas considerando até 3 condições de seleção do predicado convencional (sexo *feminino*, diagnóstico de *câncer* e parte do corpo *mama*) e 1 predicado de similaridade especificado pela operação *range query* com um raio de abrangência baseado em uma porcentagem do diâmetro de cada camada perceptual envolvida. A seletividade do predicado de similaridade também foi variada considerando a seletividade final baseada no total de imagens armazenadas.

Na Tabela 4 é exibida a dimensionalidade das camadas perceptuais consideradas nos experimentos, o número de imagens representativas de cada camada perceptual e o diâmetro das mesmas. A camada perceptual *Histograma de Cor* possui alta dimensionalidade e uma diferença grande entre a dimensionalidade e o número de imagens representativas. Já a camada perceptual *Haralick Variância* possui baixa dimensionalidade e quase nenhuma diferença entre sua dimensionalidade e o número de imagens representativas.

Tabela 4 – Características de cada camada perceptual considerada nos testes de *BloOmnImg*. Fonte: Elaborado pelo autor.

Camada Perceptual	Dimensionalidade do vetor de característica	Número de imagens representativas Omni	Diâmetro
Histograma de Cor	256	3	584,292.53
Haralick Variância	4	3	80.51

**Configuração 1.** O primeiro conjunto de configurações de desempenho considera o erro  $p$  (Equação 4.1) da estrutura *bloom filter* no processamento do cálculo de similaridade. Na Tabela 5 são mostradas as configurações geradas variando o erro da estrutura *BloomFilterFiltragem*, considerando apenas o predicado de similaridade e a camada perceptual *Histograma de Cor*. Desde que o objetivo é analisar o erro da estrutura *bloom filter* entre as etapas de filtragem e refinamento da técnica Omni, o predicado convencional foi desconsiderado. Além disso, a camada perceptual *Histograma de Cor* foi selecionada porque o desempenho da técnica Omni é mais influenciado por camadas perceptuais de alta dimensionalidade.

Tabela 5 – Configurações definidas para os testes de desempenho de *BloOmnImg*, os quais variam o erro  $p$  da estrutura *bloom filter* (Equação 4.1). Fonte: Elaborado pelo autor.

$e^{-7}$	$e^{-5}$	$e^{-3}$	$e^{-1}$	$3e$	$5e$	$7e$
E1	E2	E3	E4	E5	E6	E7

**Configuração 2.** O segundo conjunto de configurações de desempenho considera a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais. Para o predicado convencional, foram definidas as seguintes consultas: (i) sem predicado - *SemConv*; (ii) com predicado de baixa seletividade (33,33%) envolvendo os pacientes do sexo feminino - *BaixaSel*; e (iii) com predicado de alta seletividade (0,08%) envolvendo os pacientes do sexo feminino com câncer de mama - *AltaSel*. Para o predicado de similaridade, foram definidas consultas sobre: (i) a camada perceptual de alta dimensionalidade *Histograma de Cor* (256 dimensões) - *His*; a camada perceptual de baixa dimensionalidade *Haralick Variância* (4 dimensões) - *HV*; e (iii) essas duas camadas conjuntamente - *His/HV*. No total, foram geradas 9 configurações, conforme mostrado na Tabela 6.

Tabela 6 – Configurações definidas para os testes de desempenho de *BloOmnImg*, as quais variam as camadas perceptuais e a seletividade do predicado convencional. Fonte: Elaborado pelo autor.

	<b>His</b>	<b>HV</b>	<b>His/HV</b>
<b>SemConv</b>	(C1) SemConvHis	(C2) SemConvHV	(C3) SemConvHis/HV
<b>BaixaSel</b>	(C4) BaixaSelHis	(C5) BaixaSelHV	(C6) BaixaSelHis/HV
<b>AltaSel</b>	(C7) AltaSelHis	(C8) AltaSelHV	(C9) AltaSelHis/HV

**Configuração 3.** Além das Configurações 1 e 2, voltadas à análise de desempenho, também foi gerada uma terceira configuração, a qual tem como objetivo analisar o potencial semântico relacionado a agrupamentos de interesse. Estes agrupamentos permitem obter os resultados das consultas definidas agrupados por atributos convencionais de interesse, oferecendo novas possibilidades de análise. Na Tabela 7 são ilustradas as configurações de consulta com base em (C7) *AltaSelHis*, pois esta configuração possui predicado convencional e de similaridade, dado que o objetivo é considerar o potencial semântico da consulta, já que parte dos dados utilizados são sintéticos. Foram considerados os seguintes agrupamentos de interesse: tipo de sangue do paciente (G1), categoria de idade do paciente (G2) e estado natal do paciente (G3).

Tabela 7 – Configurações definidas para uma análise de semântica das consultas nos casos de teste de *BloOmnImg*, as quais variam o agrupamento final dos dados. Fonte: Elaborado pelo autor.

Configuração	Agrupamento
G1	Tipo de sangue
G2	Categoria de idade
G3	Estado natal

**Execução.** Foi usado um *cluster* com 5 nós, cada qual com, no mínimo, 3GB de RAM. Cada configuração definida foi executada 5 vezes, com remoção de *outliers*, média e desvio padrão foram calculados. Além disso, *caches* e *buffers* foram esvaziados depois de cada execução. Por fim, a linguagem Java foi considerada para a implementação.

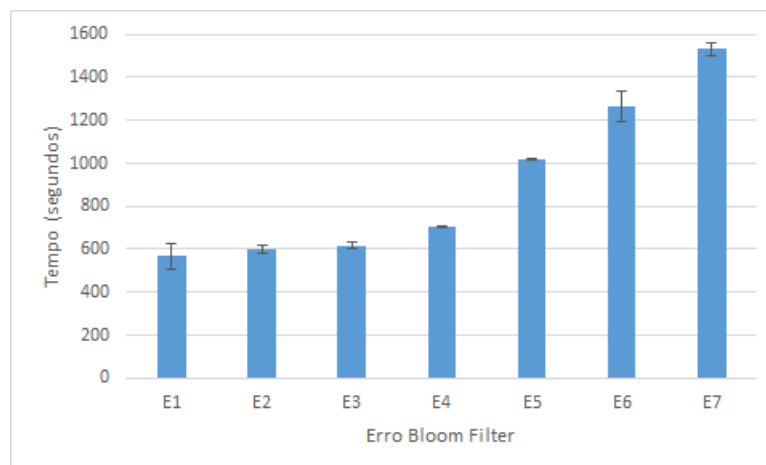
**Trabalho relacionado.** Foi considerado como trabalho relacionado o método SBFCJ, descrito na seção 5.4.2.1. Esse trabalho representa o método mais próximo existente na literatura, quando considerado o objetivo do método *BloOmnImg*. Entretanto, SBFCJ não considera imagens armazenadas em *data warehouses*. Para suprir essa limitação e usar esse trabalho nas comparações, foi usada a seguinte abordagem: o predicado de similaridade foi processado sem considerar a otimização da técnica Omni, ou seja, apenas os vetores de características, presentes na tabela de dimensão de imagens *VetCaract*, foram utilizados para calcular as distâncias do centro de consulta para todas as imagens armazenadas.

## 6.4.2 Resultados

Foram definidos 4 experimentos para analisar as vantagens introduzidas pelo método proposto *BloOmnImg*. O primeiro experimento teve como objetivo investigar a **variação de erro da estrutura bloom filter**, ou seja, investigou-se o quanto a probabilidade da estrutura probabilística *bloom filter* gerar falsos positivos influencia no custo do cálculo de distância. Para tanto, considerou-se o Volume 2 de dados descrito na Tabela 3 e a Configuração 1.

Os resultados obtidos são ilustrados na Figura 22. É possível notar que, à medida que o erro cresce, o tempo de processamento de *BloOmnImg* aumenta. Isso deve-se ao fato de que a fase de filtragem da técnica Omni possui o objetivo de filtrar os elementos para que os cálculos de distância sejam realizados apenas para os elementos candidatos à resposta. Porém, *bloom filter* é uma estrutura probabilística, ou seja, há um erro associado aos parâmetros de criação da estrutura, e este erro representa a possibilidade de existência de falsos positivos. Portanto, quando há um erro baixo, *bloom filter* retorna poucos falsos positivos para os elementos candidatos à resposta e o número de cálculos de distância realizados é baixo. À medida que o erro de *bloom filter* aumenta, mais falsos positivos são retornados, aumentando a quantidade de cálculos de distância que precisam ser realizados na fase de refinamento da técnica Omni. Baseado nos resultados descritos na Figura 22, para os próximos experimentos considera-se a configuração E1 para a criação da estrutura *bloom filter* utilizada pelo método proposto *BloOmnImg*.

Figura 22 – Testes de desempenho considerando a variação do erro da estrutura *bloom filter*, utilizada no método *BloOmnImg*, de acordo com a Tabela 5. Fonte: Elaborado pelo autor.

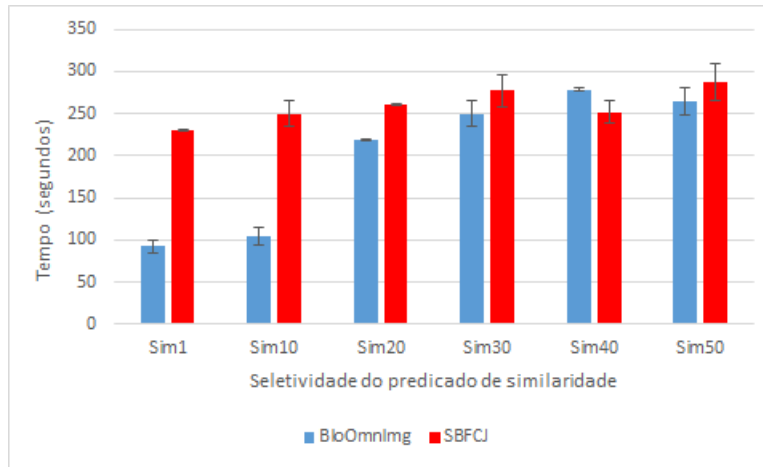


O segundo experimento teve como objetivo investigar a **variação da seletividade do predicado de similaridade**, considerando a camada perceptual de alta dimensionalidade *Histograma de Cor*. Para tanto, foi considerado o Volume 1 de dados descrito na Tabela 3. Foi considerada a camada perceptual de alta dimensionalidade pela sua influência na técnica Omni, já que sua utilização é um diferencial comparado ao método mais próximo (SBFCJ). A seletividade do predicado de similaridade variou de 1% (*Sim1*) até 50% (*Sim50*), representando a porcentagem do número total de imagens que respondem à operação de similaridade.

Na Figura 23 são mostrados os resultados de desempenho dos métodos *BloOmnImg* e SBFCJ para as consultas definidas. Para seletividades de 1% e 10% os ganhos de *BloOmnImg* sobre SBFCJ foram muito significativos, de 60,08% e 58,19%, respectivamente. Considerando a seletividade de 20%, o ganho de desempenho de *BloOmnImg* diminuiu quando comparado com SBFCJ, sendo de 16,15%. Para seletividades a partir de 30%, os tempos entre *BloOmnImg* e SBFCJ se equilibram, considerando o desvio padrão, com ganhos de *BloOmnImg* de 9,68% e 8,11% para seletividades de 30% e 50%. Considerando a seletividade de 40% SBFCJ provê um ganho de desempenho de 9,70%.

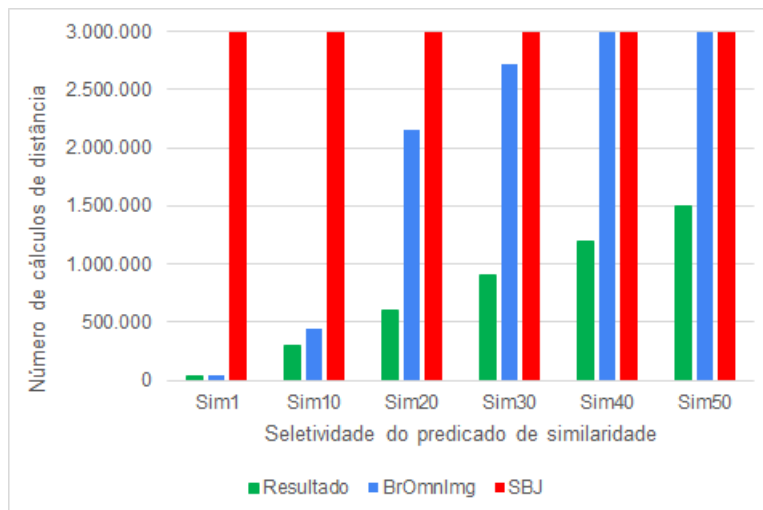
Analisando os resultados, conclui-se que, quanto maior o raio, menor é a vantagem de *BloOmnImg* sobre SBFCJ. Isso deve-se ao fato de que *BloOmnImg* é baseado na técnica Omni, sendo que o aumento da seletividade do predicado de similaridade torna a etapa de filtragem da técnica Omni ineficiente. Na Figura 24 é exibido o número de cálculos de distância realizados na execução de cada método, assim como o resultado do predicado de similaridade, ou seja, o número de imagens retornadas. Como SBFCJ não utiliza a técnica Omni, o número de cálculos de distância requeridos por esse método é igual ao número de imagens armazenadas no conjunto de dados, ou seja, 3 milhões. Para *BloOmnImg*, nota-se que as seletividades de 1% e 10% requerem a realização de um número de cálculos de distância que é próximo ao resultado retornado. As demais seletividades mostram uma diferença grande entre o número de cálculos de distância e a quantidade de imagens retornadas. Isto explica os resultados ilustrados na Figura 23, ou

Figura 23 – Testes de desempenho comparando o método proposto *BloOmnImg* com o método concorrente SBFCJ, variando a seletividade do predicado de similaridade. Fonte: Elaborado pelo autor.



seja, a partir de 20%, o desempenho de *BloOmnImg* diminui devido à ineficiência do filtro dos elementos candidatos ao predicado de similaridade somado ao custo adicional do processamento desta etapa.

Figura 24 – Número de cálculos de distância realizadas no processamento do predicado de similaridade para os métodos *BloOmnImg* e SBFCJ, além do resultado deste predicado. Fonte: Elaborado pelo autor.



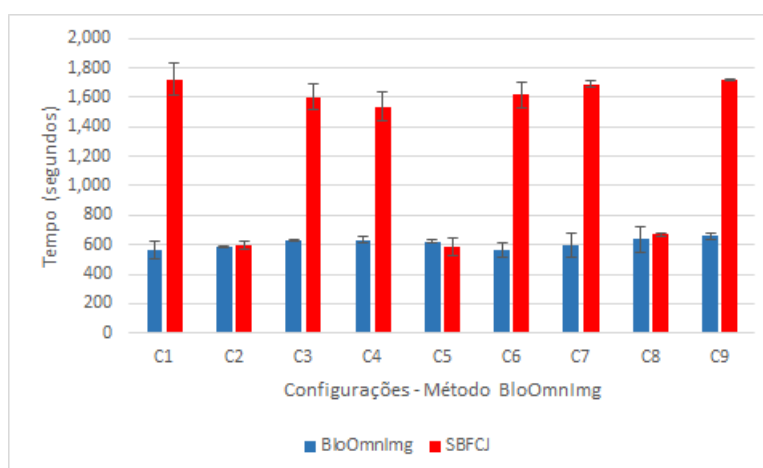
O terceiro experimento teve como objetivo investigar a **variação de seletividade das camadas perceptuais e do predicado convencional** conjuntamente, considerando o Volume 2 de dados (Tabela 3). Para cada uma das configurações *C1* a *C9* definidas na Configuração 2 (Tabela 6), o número de imagens que atenderam ao predicado convencional e de similaridade são descritas na Tabela 8. É possível notar o quanto cada configuração restringe o número de imagens que obedecem ao predicado convencional e às camadas perceptuais envolvidas. Ou seja, quanto maior a seletividade do predicado convencional e quanto mais camadas perceptuais envolvidas, maior a filtragem dos dados.

Tabela 8 – Número de imagens retornadas em cada configuração definida na Tabela 6. Fonte: Elaborado pelo autor.

	His	HV	His/HV
<b>SemConv</b>	224.060	125.460	5.620
<b>BaixaSel</b>	74.800	42.000	1.800
<b>AltaSel</b>	160	180	20

Na Figura 25 são ilustrados os resultados obtidos comparando-se *BloOmnImg* e SBFCJ. Para as configurações que possuem pelo menos uma camada perceptual de alta dimensionalidade (C1, C3, C4, C6, C7, C9), *BloOmnImg* proveu ganhos de desempenho que variaram de 58,83% a 66,93% quando comparado com SBFCJ. Isso está relacionado à complexidade dos cálculos de distância da técnica Omni, sendo dependente da dimensionalidade do conjunto de dados e do número de elementos representativos. A diferença entre o número de dimensões de *Histograma de Cor* (256) e o número de elementos representativos correspondente (3) impactou positivamente no desempenho de *BloOmnImg*. Para as configurações que possuem apenas a camada perceptual de baixa dimensionalidade (C2, C5, C8), *BloOmnImg* não obteve ganhos de desempenho significativos em relação à SBFCJ. Nessas configurações, a diferença entre o número de dimensões de *Haralick Variância* (4) e o número de elementos representativos correspondente (3) foi muito pequena. Positivamente, mesmo *BloOmnImg* sendo mais complexo do que SBCJ por integrar a técnica Omni para processar o predicado de similaridade, ou ele apresenta resultados muito próximos ou provê melhores resultados.

Figura 25 – Testes de desempenho comparando *BloOmnImg* com SBFCJ, considerando a variação da seletividade do predicado convencional e dimensionalidade das camadas perceptuais envolvidas de acordo com a Tabela 6. Fonte: Elaborado pelo autor.



O último experimento realizado teve como objetivo investigar o potencial **semântico das consultas**. Além do número de imagens que obedecem ao predicado convencional e de similaridade, este número pode ser agrupado, sobre determinado atributo, para enriquecer o suporte à tomada de decisão médica. Como estes experimentos utilizam o Volume 2 de dados (Tabela 3) e a Configuração 3 ((C7) *AltaSelHis*), a soma dos agrupamentos de cada consulta

definida deve ser 160 (Tabela 8). Na Figura 26 são ilustrados os resultados para a configuração e agrupamentos definidos na Tabela 7:

- A configuração *G1* considera o agrupamento do tipo de sangue do paciente. Isto permite uma análise de resistência ou vulnerabilidade de cada tipo de sangue para uma determinada situação. Por exemplo, no conjunto de dados considerado, os resultados das consultas realizadas mostraram que o sangue do tipo A é o mais vulnerável.
- A configuração *G2* considera o agrupamento da categoria de idade do paciente. Isto permite uma análise de resistência ou vulnerabilidade dependendo da idade do paciente para uma determinada situação. Por exemplo, no conjunto de dados considerado, os resultados das consultas realizadas mostraram que os idosos são mais vulneráveis.
- A configuração *G3* considera o agrupamento do estado natal do paciente. Isto permite uma análise de resistência ou vulnerabilidade dependendo da localidade do paciente para uma determinada situação. Por exemplo, no conjunto de dados considerado, os resultados das consultas realizadas mostraram que, considerando alguns estados da região Sul e Sudeste, o estado de SP apresenta um maior número de casos.

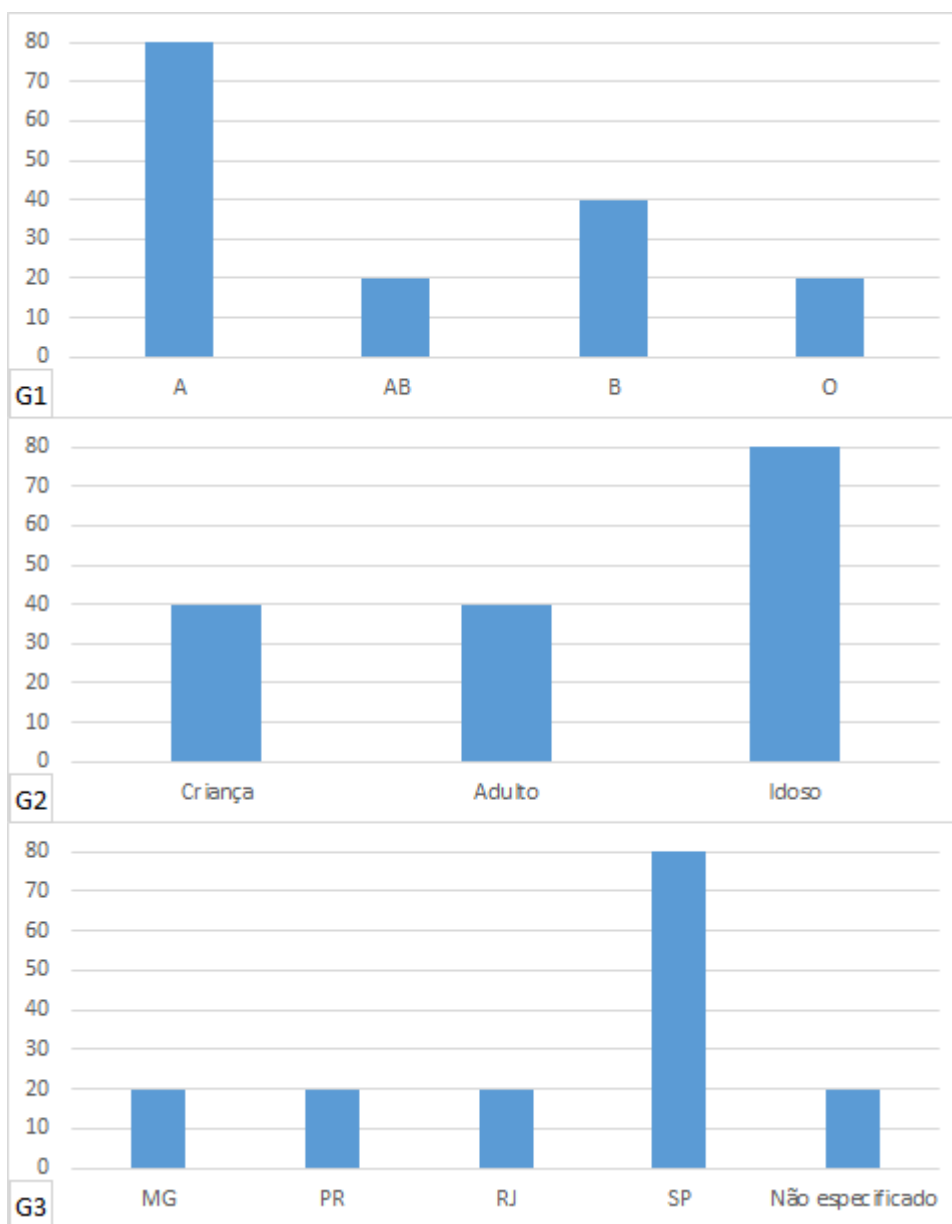
Vale lembrar que todas essas análises podem levar em consideração o número relativo e não absoluto da quantidade de imagens que obedecem ao predicado convencional e de similaridade. Os resultados apresentados são exemplos de enriquecimento semântico que pode ser incorporado às consultas no suporte à tomada de decisão da área médica. Por exemplo, questões de tipo de sangue, idade do paciente e localidade são fatores altamente relacionados ao COVID-19 (ZHAO; et al., 2020; RICHARDSON; et al., 2020; MAHASE, 2020).

## 6.5 Considerações finais

Neste capítulo foi proposto método *BloOmnImg*, o qual integra as técnicas *bloom filter cascade join* e *Omni* para o processamento eficiente de consultas analíticas estendidas com predicado de similaridade de imagens em Spark. As vantagens introduzidas por *BloOmnImg* foram demonstradas por meio de testes de desempenho experimentais. Primeiramente, foi identificado qual a taxa de erro a ser aplicada a *BloOmnImg*. Depois, usando essa taxa de erro, *BloOmnImg* foi comparado com o trabalho da literatura mais relacionado, SBFCJ. Analisando-se a variação da seletividade apenas do predicado de similaridade de imagens, *BloOmnImg* proveu ganhos de desempenho de até 60,08%. Para a variação das camadas perceptuais e predicado convencional, *BloOmnImg* proveu ganhos de até 66,93%. Por fim, também foram realizados testes para mostrar semanticamente como as consultas investigadas nesta dissertação de mestrado são relevantes para a tomada de decisão médico-analítica.



Figura 26 – Exemplos sintéticos de consultas semânticas com agrupamentos providas por *BloOmnImg*.  
Fonte: Elaborado pelo autor.



Por ser baseado em uma estrutura probabilística, *BloOmnImg* é adequado para ambientes computacionais com pouca memória primária. No próximo capítulo, Capítulo 7, investiga-se ambientes com memória primária suficiente para utilizar estruturas exatas, como *hash map*. Propõe-se o método *BrOmnImg*, o qual integra as técnicas *broadcast join* e *Omni* para otimizar, respectivamente, a junção-estrela e o cálculo de similaridade durante o processamento de consultas analíticas estendidas com predicado de similaridade de imagens em Spark.



## MÉTODOS *BrOmnImg* E *BrOmnImg<sup>CF</sup>*

---

### 7.1 Considerações iniciais

Neste capítulo são introduzidos os métodos propostos *BrOmnImg* e *BrOmnImg<sup>CF</sup>*, os quais integram as técnicas de *broadcast join* e *Omni* para processar eficientemente consultas analíticas estendidas com predicado de similaridade em Spark. *BrOmnImg* é introduzido na seção 7.2, enquanto que *BrOmnImg<sup>CF</sup>* é descrito na seção 7.3. Ambos métodos são propostos em termos de suas principais características e algoritmos. A validação dos algoritmos é feita na seção 7.4. O capítulo é finalizado na seção 7.5, com as considerações finais.

### 7.2 Descrição do método *BrOmnImg*

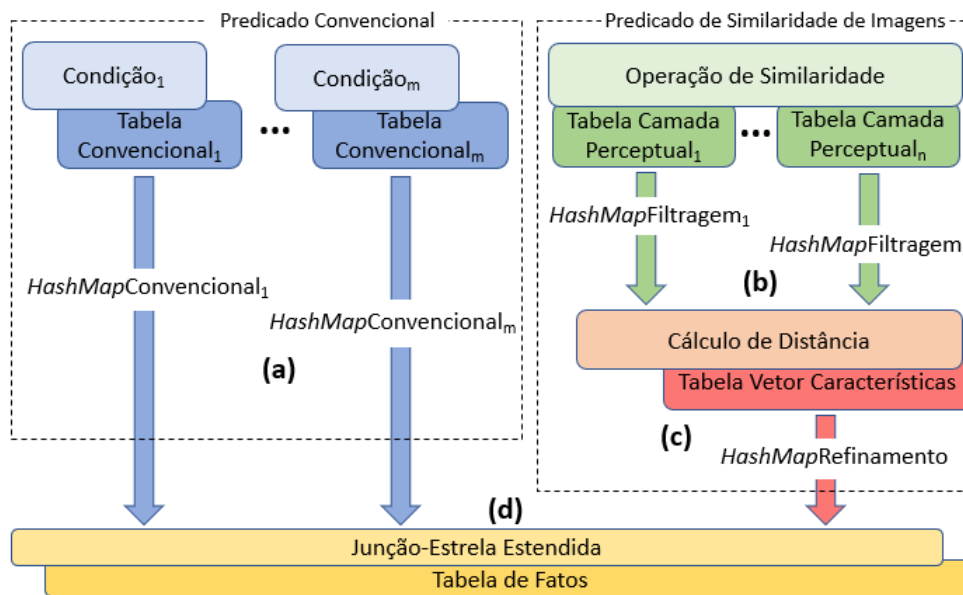
Nesta seção as técnicas *broadcast join* (seção 4.7) e *Omni* (seção 3.3.3.2) são utilizadas para otimizar a junção-estrela e o cálculo de similaridade, respectivamente. Propõe-se o método *BrOmnImg* (acrônimo para *Broadcast Omni for processing analytical Image queries*), o qual integra essas técnicas para a execução eficiente, em Spark, de consultas analíticas estendidas com predicado de similaridade sobre o DWbase ilustrado na seção 6.2.

#### 7.2.1 Método

Na Figura 27 é ilustrada uma visão geral do método *BrOmnImg* usando o mesmo esquema de cores de DWbase. *BrOmnImg* integra as técnicas *broadcast join* e *Omni* da seguinte forma. As condições de seleção são aplicadas para cada tabela de dimensão convencional  $i$  ( $1 \leq i \leq m$ ) envolvida no predicado convencional. *BrOmnImg* armazena e transmite, para todos os nós do *cluster*, os resultados de cada tabela convencional  $i$  nas estruturas *HashMapConvencional<sub>i</sub>* (Figura 27a). Para o predicado de similaridade, a etapa de filtragem da técnica *Omni* é aplicada em cada tabela  $j$  ( $1 \leq j \leq n$ ) referente a cada camada perceptual envolvida. Com isso, os elementos

candidatos de cada camada perceptual são armazenados nas estruturas *HashMapFiltragem<sub>j</sub>* e transmitidas para todos os nós do *cluster* (Figura 27b). Na sequência, são realizadas as operações de cálculo de distância usando os resultados candidatos e a tabela *Vetor Características* para eliminar falsos positivos (Figura 27c). Os resultados gerados são aqueles que atendem ao predicado de similaridade, e são armazenados na estrutura *HashMapRefinamento*. Por fim, as estruturas *HashMapConvencional<sub>k</sub>* e a estrutura *HashMapRefinamento* são transmitidas para todos os nós do *cluster* para a realização da junção-estrela estendida sobre a tabela de fatos (Figura 27d).

Figura 27 – Visão geral do método BrOmnImg. Fonte: Elaborado pelo autor.



### 7.2.2 Algoritmo

No Algoritmo 2 é exemplificado como BrOmnImg executa a consulta  $Q$ : “Quantas imagens são similares à uma imagem  $s_q$  com raio  $r_q$ , considerando a camada perceptual histograma de cor e pacientes do sexo feminino?”. As entradas do algoritmo são todas as tabelas envolvidas na consulta, o centro de consulta e o raio de consulta. Como saída, o algoritmo gera a resposta de  $Q$ . Para a execução da consulta é considerado o DWbase ilustrado na seção 6.2.

RDDs são criados para cada tabela utilizada na execução do predicado convencional (linha 1), de similaridade (linha 4 e 7) e junção-estrela estendida (linha 11), sendo essa última operação realizada na tabela de fatos *Exame*. O predicado convencional é filtrado e armazenado na estrutura *HashMapConventional*, a qual é transmitida para todos os nós do *cluster* nas linhas 2 e 3 (Figura 27a). Para processar o predicado de similaridade, a etapa de filtragem da técnica Omni é aplicada e os elementos candidatos à resposta são transmitidos para todos os nós do *cluster* na estrutura *HashMapFiltragem* (linhas 5 e 6 - Figura 27b). A etapa de refinamento da técnica Omni é aplicada para todos os elementos candidatos, eliminando falsos positivos. Assim, o resultado do predicado de similaridade é armazenado e transmitido pela estrutura

$HashMapRefinamento$ , nas linhas 8, 9 e 10 (Figura 27c). Finalmente, a junção-estrela estendida é realizada utilizando as estruturas que armazenam os resultados do predicado convencional e de similaridade ( $HashMapConventional$  e  $HashMapRefinamento$ , respectivamente), além do número de imagens ser calculado (linhas 12, 13 e 14 - Figura 27d).

---

**Algoritmo 2 –  $BrOmnImg$** 


---

**Input** :  $Paciente$ ,  $Histograma\ de\ Cor$ ,  $Vetor\ Característica$ ,  $Exame$ ,  $s_q$  e  $r_q$

**Output** : resultado de  $Q$

- 1  $RDD_{Paciente} = Paciente$
  - 2  $RDD_{Paciente}.filter(sexo = 'feminino').mapToPair(PacienteKey, null)$
  - 3  $HashMapConventional = broadcast(RDD_{Paciente}.collect())$
  - 4  $RDD_{His} = Histograma\ de\ Cor$
  - 5  $RDD_{His}.filter(Filtragem(DistRep_1, DistRep_2, DistRep_3, s_q, r_q)).mapToPair(hisKey, null)$
  - 6  $HashMapFiltragem = broadcast(RDD_{His}.collect())$
  - 7  $RDD_{VetCaract} = Vetor\ Característica$
  - 8  $RDD_{VetCaract}.filter(if(HashMapFiltragem.hasKey(vetCaractKey))then$   
 $\quad return(Refinamento(vetCaractHis, s_q, r_q))$
  - 9  $RDD_{VetCaract}.mapToPair(vetCaractKey, null)$
  - 10  $HashMapRefinamento = broadcast(RDD_{VetCaract}.collect())$
  - 11  $RDD_{Exame} = Exame$
  - 12  $RDD_{Exame}.filter(HashMapConventional.hasKey(e_PacienteKey) \cap$   
 $\quad HashMapRefinamento.hasKey(e_vetCaractKey))$
  - 13  $RDD_{Exame}.mapToPair(1, quantidade\ de\ exames)$
  - 14  $resultadoQ = RDD_{Exame}.reduceByKey(v_1 + v_2)$
- 

$BrOmnImg$  processa cada tabela do esquema-estrela como um RDD. Ele aplica os filtros convencionais e de imagem sobre os RDDs correspondentes usando a operação *filter*, e armazena os dados nas estruturas *hash map* por meio das operações *mapToPair* e *collect*. Adicionalmente,  $BrOmnImg$  realiza a junção-estrela estendida aplicando as operações *filter*, *mapToPair* e *reduceByKey*.

### 7.3 Descrição do método $BrOmnImg^{CF}$

Nesta seção é proposto o método  $BrOmnImg^{CF}$  (acrônimo para *Broadcast Omni for processing analytical Image queries with Conventional Filter*), o qual integra as técnicas de *broadcast join* e *Omni* para a execução eficiente, em Spark, de consultas analíticas estendidas com predicado de similaridade de imagens.  $BrOmnImg^{CF}$  estende o método  $BrOmnImg$  de

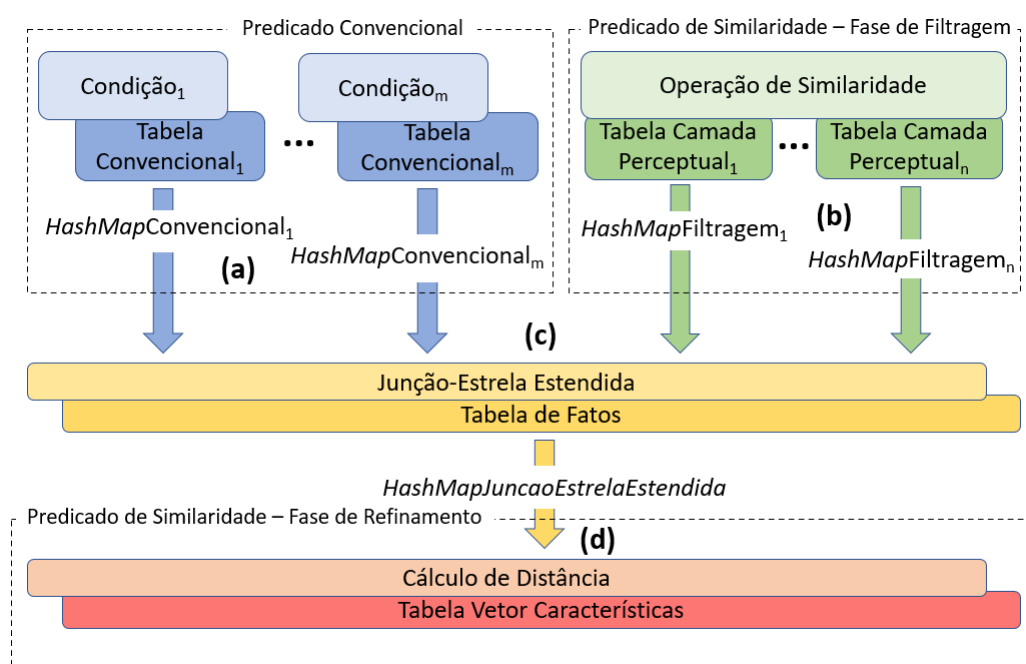
forma a utilizar o predicado convencional para a filtragem dos elementos candidatos ao predicado de similaridade. Em *BrOmnImg*, o processamento do predicado convencional somente ocorre depois que os cálculos de distância já foram processados.

### 7.3.1 Método

Na Figura 28 é ilustrada uma visão geral do método *BroOmnImg<sup>CF</sup>* usando o mesmo esquema de cores de DWbase. Do mesmo modo que *BrOmnImg*, *BrOmnImg<sup>CF</sup>* processa o predicado convencional para cada tabela de dimensão convencional  $i$  ( $1 \leq i \leq m$ ) envolvida na consulta e armazena seus resultados nas estruturas *HashMapConvencional<sub>i</sub>*, transmitindo-as para todos os nós do *cluster* (Figura 28a). Na etapa de filtragem da técnica Omni cada tabela de dimensão relacionada com as  $j$  ( $1 \leq j \leq n$ ) camadas perceptuais envolvidas, os elementos candidatos ao predicado de similaridade são armazenados e transmitidos nas estruturas *HashMapFiltragem<sub>j</sub>* (Figura 28b).

A diferença de *BrOmnImg<sup>CF</sup>* para *BrOmnImg* é que, ao invés de executar a etapa de refinamento logo após a etapa de filtragem, *BrOmnImg<sup>CF</sup>* realiza a junção-estrela estendida considerando o predicado convencional e os elementos candidatos ao predicado de similaridade. Com isso, os fatos que obedecem tanto ao predicado convencional quanto aos elementos candidatos ao predicado de similaridade são armazenados e transmitidos na estrutura *HashMapJuncaoEstrelaEstendida* (Figura 28c). A etapa de refinamento é realizada considerando os elementos presentes na estrutura *HashMapJuncaoEstrelaEstendida*. Calcula-se, finalmente, a resposta da consulta (Figura 28d).

Figura 28 – Visão geral do método *BrOmnImg<sup>CF</sup>*. Fonte: Elaborado pelo autor.



Para que seja possível processar a etapa de refinamento depois da junção-estrela, como

proposto por  $BrOmnImg^{CF}$ , a tabela de  *fatos*  tem que possuir um relacionamento unitário com a tabela  *Vetor Característica* . Tendo como exemplo o DWbase, todo exame possui uma imagem e toda imagem é referente a um exame. Isto torna  $BrOmnImg^{CF}$  mais restrito do que  $BrOmnImg$ .

### 7.3.2 Algoritmo

No Algoritmo 3 é exemplificado como  $BrOmnImg^{CF}$  executa a mesma consulta  $Q$  descrita na seção 7.2.2: “*Quantas imagens são similares à uma imagem  $s_q$  com raio  $r_q$ , considerando a camada perceptual histograma de cor e pacientes do sexo feminino?*”. Como entradas do algoritmo, devem ser definidas as tabelas envolvidas na consulta, o centro de consulta e o raio de consulta. Como saída, o algoritmo gera a resposta de  $Q$ . Para a execução da consulta é considerado o DWbase ilustrado na Figura 20.

Inicialmente, RDDs são criados para cada tabela utilizada na execução do predicado convencional (linha 1), de similaridade (linha 4 e 7) e junção-estrela estendida (linha 11). O predicado convencional é processado e transmitido em  $HashMapConventional$  nas linhas 2 e 3 (Figura 28a). A etapa de filtragem da técnica Omni é aplicada e os elementos candidatos à resposta são transmitidos em  $HashMapFiltragem$  (linhas 5 e 6 - Figura 28b). A junção-estrela estendida entre o predicado convencional e os elementos candidatos ao predicado de similaridade é realizada e seu resultado é transmitido na estrutura  $HashMapJuncaoEstrelaEstendida$  (linhas 8, 9 e 10 - Figura 28c). Por fim, nas linhas 12, 13 e 14 a etapa de refinamento da técnica Omni é executada para todos os elementos candidatos ao predicado de similaridade filtrados pelo predicado convencional, presentes em  $HashMapJuncaoEstrelaEstendida$  (Figura 28d).

Como discutido anteriormente, para que seja possível realizar a etapa de refinamento depois da junção-estrela, como proposto por  $BrOmnImg^{CF}$ , a tabela  *Exame*  deve possuir um relacionamento unitário com a tabela  *Vetor Característica* . Além disso, as funções utilizadas para filtragem, transmissão e cálculo final dos resultados são as mesmas do algoritmo  $BrOmnImg$ :  *filter* ,  *mapToPair* ,  *broadcast*  e  *reduceByKey* .

**Algoritmo 3** – BrOmnImg<sup>CF</sup>**Input** : *Paciente*, *Histograma de Cor*, *Vetor Característica*, *Exame*,  $s_q$  e  $r_q$ **Output** : resultado de  $Q$ 


---

```

1  $RDD_{Paciente} = Paciente$ 
2  $RDD_{Paciente}.filter(sexo = 'feminino').mapToPair(PacienteKey, null)$ 
3  $HashMapConvencional = broadcast(RDD_{Paciente}.collect())$ 
4  $RDD_{His} = Histograma\ de\ Cor$ 
5  $RDD_{His}.filter(Filtragem(DistRep_1, DistRep_2, DistRep_3, s_q, r_q)).mapToPair(hisKey, null)$ 
6  $HashMapFiltragem = broadcast(RDD_{His}.collect())$ 
7  $RDD_{Exame} = Exame$ 
8  $RDD_{Exame}.filter(HashMapConvencional.hasKey(e_PacienteKey) \cap$ 
    $HashMapFiltragem.hasKey(e_hisKey))$ 
9  $RDD_{Exame}.mapToPair(examKey, quantidade\ de\ exames)$ 
10  $HashMapJuncaoEstrelaEstendida = broadcast(RDD_{Exame}.collect())$ 
11  $RDD_{VetCaract} = Vetor\ Caracteristica$ 
12  $RDD_{VetCaract}.filter(if(HashMapJuncaoEstrelaEstendida.hasKey(vetCaractKey)) then$ 
    $return(Refinamento(vetCaractHis, s_q, r_q))$ 
13  $RDD_{VetCaract}.mapToPair(vetCaractKey, quantidade\ de\ exames)$ 
14  $resultadoQ = RDD_{VetCaract}.reduceByKey(v_1 + v_2)$ 

```

---

## 7.4 Validação

A validação dos métodos propostos *BrOmnImg* e *BrOmnImg<sup>CF</sup>* foi realizada por meio de testes de desempenho experimentais dentro do contexto da área médica. Primeiramente, é descrito o ambiente de teste e, na sequência, os resultados obtidos.

### 7.4.1 Ambiente de teste

**Dados.** Os dados foram gerados da mesma forma que descrito na seção 6.4.1. Também foram considerados 2 volumes de dados: (i) Volume 1, com 3 milhões de exames; e (ii) Volume 2, com 20 milhões de exames (Tabela 3).

**Consultas 1.** A primeira classe de consultas analíticas estendidas com predicado de similaridade foi a mesma que a introduzida na seção 6.4.1.

**Consultas 2.** A segunda classe de consultas analíticas estendidas com predicado de similaridade foi definida considerando até 4 condições de seleção do predicado convencional (paciente com tipo Rh do sangue *positivo*, paciente do sexo *feminino*, diagnóstico de *câncer* e parte do corpo



sendo *mama*) e 1 predicado de similaridade especificado pela operação *range query* com um raio de abrangência de 20% do diâmetro de cada camada perceptual envolvida. A seletividade do predicado de similaridade também foi variada considerando a seletividade final baseada no total de imagens armazenadas. Essa segunda classe de consultas foi introduzida para oferecer suporte para a avaliação de desempenho de *BrOmnImg<sup>CF</sup>*, desde que esse método é direcionado ao uso do predicado convencional como filtro. As características das camadas perceptuais são as mesmas que as definidas na [Tabela 4](#).

**Configuração 1.** O primeiro conjunto de configurações de desempenho foi o mesmo que o definido na seção [6.4.1](#) na Configuração 2. Ele está descrito na [Tabela 6](#).

**Configuração 2.** O segundo conjunto de configurações de desempenho considerou as seguintes características para a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais. Para o predicado convencional, foram definidas consultas: (i) sem predicado - *SemConv*; (ii) seletividade de 50% envolvendo os pacientes com tipo Rh do sangue positivo - *Conv50*; (iii) seletividade de 33,33% envolvendo os pacientes do sexo feminino - *Conv33*; e (iv) com seletividade de 0,08% envolvendo os pacientes do sexo feminino com câncer de mama - *Conv0.08*. Para o predicado de similaridade, foram definidas consultas sobre: (i) a camada perceptual de alta dimensionalidade *Histograma de Cor* (256 dimensões) - *His*; a camada perceptual de baixa dimensionalidade *Haralick Variância* (4 dimensões) - *HV*; e (iii) essas duas camadas conjuntamente - *His/HV*. No total, foram geradas 12 configurações, conforme mostrado na [Tabela 9](#).

Tabela 9 – Configurações definidas para os testes de desempenho de *BrOmnImg*, as quais variam a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor.

	<b>His</b>	<b>Har</b>	<b>His/Har</b>
<b>SemConv</b>	(C1) SemConvHis	(C2) SemConvHar	(C3) SemConvHis/Har
<b>Conv50</b>	(C4) Conv50His	(C5) Conv50Har	(C6) Conv50His/Har
<b>Conv33</b>	(C7) Conv33His	(C8) Conv33Har	(C9) Conv33His/Har
<b>Conv0,08</b>	(C10) Conv0,08His	(C11) Conv0,08Har	(C12) Conv0,08His/Har

**Configuração 3.** O terceiro conjunto de configurações de desempenho considerou as configurações *C4*, *C7* e *C10* definidas na [Tabela 9](#). As configurações que não consideram predicado convencional foram descartadas porque o objetivo dessa configuração é avaliar o impacto do filtro do predicado convencional. Para a seletividade do predicado de similaridade foram consideradas valores de seletividade de 1% (*Sim1*), 10% (*Sim10*), 20% (*Sim20*), 30% (*Sim30*), 40% (*Sim40*) e 50% (*Sim50*). Portanto, permutando as variações descritas, 18 configurações foram definidas, as quais são exibidas na [Tabela 10](#).

**Execução.** Foi realizada da mesma forma que a descrita na seção [6.4.1](#).

**Trabalho relacionado.** Para o método *BrOmnImg*, foi considerado como trabalho relacionado o método SBJ, descrito na seção [5.4.2.1](#). Esse trabalho representa o método mais próximo existente

Tabela 10 – Configurações definidas para os testes de desempenho de *BrOmnImg<sup>CF</sup>*, as quais variam a seletividade do predicado convencional e de similaridade. Fonte: Elaborado pelo autor.

	<b>Sim1</b>	<b>Sim10</b>	<b>Sim20</b>	<b>Sim30</b>	<b>Sim40</b>	<b>Sim50</b>
<b>(C4)</b> <b>Conv50</b> <b>His</b>	(C4S1) Conv50 Sim1	(C4S10) Conv50 Sim10	(C4S20) Conv50 Sim20	(C4S30) Conv50 Sim30	(C4S40) Conv50 Sim40	(C4S50) Conv50 Sim50
<b>(C7)</b> <b>Conv33</b> <b>His</b>	(C7S1) Conv33 Sim1	(C7S10) Conv33 Sim10	(C7S20) Conv33 Sim20	(C7S30) Conv33 Sim30	(C7S40) Conv33 Sim40	(C7S50) Conv33 Sim50
<b>(C10)</b> <b>Conv0.08</b> <b>His</b>	(C10S1) Conv0.08 Sim1	(C10S10) Conv0.08 Sim10	(C10S20) Conv0.08 Sim20	(C10S30) Conv0.08 Sim30	(C10S40) Conv0.08 Sim40	(C10S50) Conv0.08 Sim50

na literatura, quando considerado o método proposto *BrOmnImg*. Entretanto, SBJ não considera imagens armazenadas em *data warehouses*. Para suprir essa limitação e usar esse trabalho nas comparações, foi usada a seguinte abordagem: o predicado de similaridade foi processado sem considerar a otimização da técnica Omni, ou seja, apenas os vetores de características, presentes na tabela de dimensão de imagens *Vetor Característica*, foram utilizados para calcular as distâncias do centro de consulta para todas as imagens armazenadas. Para o método *BrOmnImg<sup>CF</sup>*, foi considerado como trabalho relacionado o método proposto *BrOmnImg*.

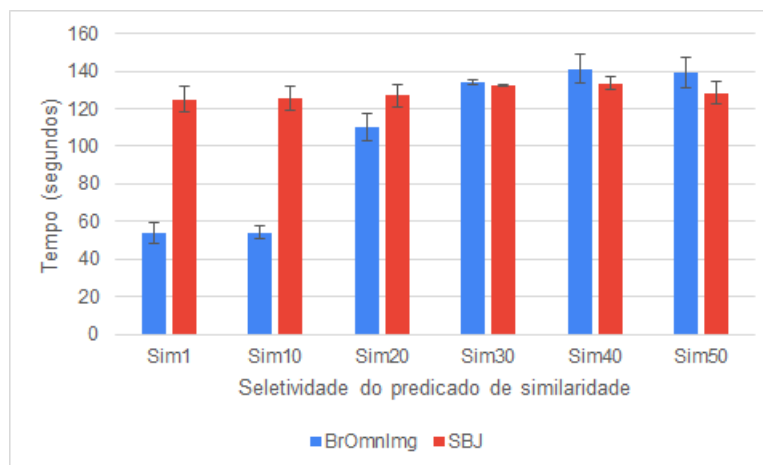
#### 7.4.2 Resultados de *BrOmnImg*

Foram definidos 3 experimentos para analisar as vantagens introduzidas pelo método proposto *BrOmnImg*. O primeiro experimento teve como objetivo investigar a **variação da seletividade do predicado de similaridade**. Para tanto, foi considerado o Volume 1 de dados descrito na [Tabela 3](#) e a camada perceptual de alta dimensionalidade *Histograma de Cor* pela sua influência na técnica Omni, já que sua utilização é um diferencial comparado ao método mais próximo (SBJ). A seletividade do predicado de similaridade variou de 1% (*Sim1*) até 50% (*Sim50*), representando a porcentagem do número total de imagens que respondem à operação de similaridade.

Na [Figura 29](#) são ilustrados os resultados dos métodos *BrOmnImg* e SBJ. Para valores de seletividade de 1% e 10% os ganhos de *BrOmnImg* sobre SBJ foram significativos, de 56,75% e 56,94%, respectivamente. Para a seletividade de 20%, o ganho de desempenho de *BrOmnImg* em relação à SBJ diminuiu para 13,31%. Para seletividades com 30%, 40% e 50%, SBJ demonstrou ligeira vantagem com ganho de desempenho de 1,27%, 5,40% and 7,72%, respectivamente. Portanto, quanto maior o raio, menor é a vantagem de *BrOmnImg* sobre SBJ. Isso deve-se ao fato de que *BrOmnImg* utiliza a técnica Omni. Assim, o aumento da seletividade do predicado de similaridade torna a etapa de filtragem da técnica Omni ineficiente, conforme discutido na seção [6.4.2](#).

O segundo experimento teve como objetivo investigar a **variação da seletividade do**

Figura 29 – Testes de desempenho comparando o método proposto *BrOmnImg* com o método concorrente SBJ, variando a seletividade do predicado de similaridade. Fonte: Elaborado pelo autor.



**predicado convencional e da dimensionalidade das camadas perceptuais**, considerando o Volume 2 de dados definido na Tabela 3, a classe de Consulta 1 e o conjunto de Configuração 1, descrita na Tabela 6. Na Figura 30 são ilustrados os resultados obtidos. Para as configurações que possuem, pelo menos, uma camada perceptual de alta dimensionalidade (C1, C3, C4, C6, C7, C9), *BrOmnImg* proveu ganhos de desempenho que variaram de 60,01% a 65,49% quando comparado com SBJ. Isso está relacionado à complexidade dos cálculos de distância da técnica Omni, sendo dependente da dimensionalidade do conjunto de dados e do número de elementos representativos. A diferença entre o número de dimensões de *Histograma de Cor* (256) e o número de elementos representativos correspondente (3) impactou positivamente no desempenho de *BrOmnImg*. Para as configurações que possuem apenas a camada perceptual de baixa dimensionalidade (C2, C5, C8), *BrOmnImg* empatou com SBJ ou proveu ganhos de desempenho de até 10,50%. Nessas configurações, a diferença entre o número de dimensões de *Haralick Variância* (4) e o número de elementos representativos correspondente (3) foi muito pequena. Positivamente, mesmo *BrOmnImg* sendo mais complexo do que SBJ por integrar a técnica Omni para processar o predicado de similaridade, ou ele apresenta resultados próximos ou provê melhores resultados.

O terceiro experimento também investiga a **variação da seletividade do predicado convencional e da dimensionalidade das camadas perceptuais**. Porém, este experimento usa o Volume 1 de dados definido na Tabela 3, a classe de Consulta 2 e o conjunto de Configuração 2 definido na Tabela 9. Na Figura 31 são mostrados os resultados obtidos para as configurações C1 a C12. Corroborando com os resultados mostrados na Figura 30, para consultas que envolvem camadas perceptuais de alta dimensionalidade (C1, C3, C4, C6, C7, C9, C10 e C12) *BrOmnImg* proveu ganhos de desempenho de 57,32% a 64,47% quando comparado com SBJ. Porém, para consultas que envolvem apenas camadas perceptuais de baixa dimensionalidade (C2, C5, C8 e C11), não foram identificadas diferenças significativas de desempenho entre os métodos. Concluiu-se que o novo volume de dados e a nova seletividade do predicado convencional considerada (*Conv50*) não influenciaram na análise dos resultados.

Figura 30 – Testes de desempenho comparando o método proposto *BrOmnImg* com o método concorrente SBJ, para o Volume 2 de dados, variando a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor.

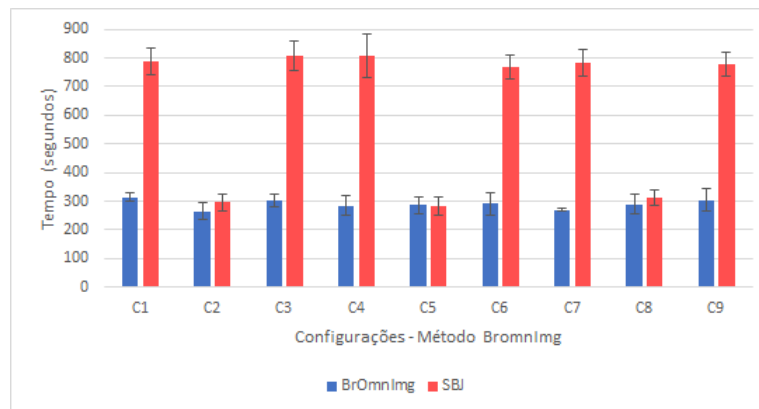
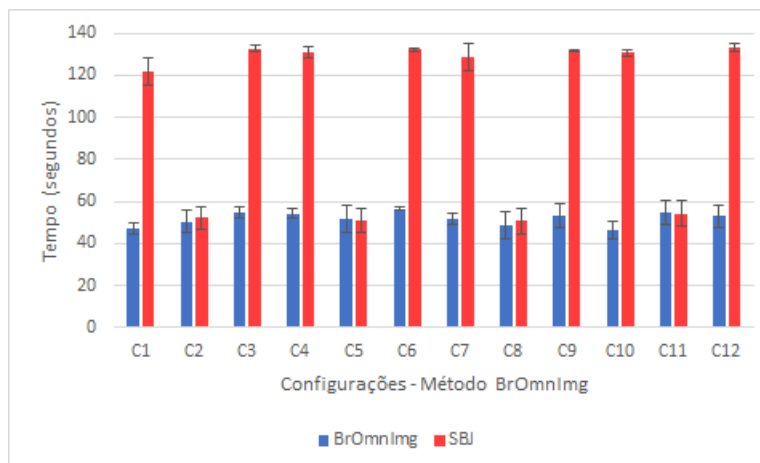


Figura 31 – Testes de desempenho comparando o método proposto *BrOmnImg* com o método concorrente SBJ, para o Volume 1 de dados, variando a seletividade do predicado convencional e a dimensionalidade das camadas perceptuais envolvidas. Fonte: Elaborado pelo autor.



Como pode ser observado na [Figura 30](#) e [Figura 31](#), a variação da seletividade do predicado convencional não possui relevância nos resultados providos pelo método *BrOmnImg*. Isso deve-se ao fato de que esse método realiza o processamento do predicado convencional somente depois que os cálculos de distância já foram processados. Ou seja, o predicado convencional não é utilizado para ajudar a filtrar os cálculos de similaridade.

### 7.4.3 Resultados de *BrOmnImg<sup>CF</sup>*

Os testes de desempenho experimentais de *BrOmnImg<sup>CF</sup>* tiveram como objetivo investigar a **variação da seletividade do predicado convencional e de similaridade**, a fim de analisar o quanto o predicado convencional pode otimizar o cálculo de similaridade. Tais experimentos foram definidos com base no Volume 1 de dados ([Tabela 3](#)), na classe de Consultas 2 e no conjunto de Configuração 3. Para cada uma das configurações definidas na [Tabela 10](#), o número de imagens que atenderam ao predicado convencional e de similaridade é descrito na [Tabela 11](#).

Nota-se a influência da seletividade do predicado de similaridade em detrimento da seletividade do predicado convencional.

Tabela 11 – Número de imagens retornadas para as configurações definidas para validação do método *BrOmnImg<sup>CF</sup>* (Tabela 10). Autor: Elaborado pelo autor.

	Sim1	Sim10	Sim20	Sim30	Sim40	Sim50
(C4) <b>Conv50His</b>	16761	157314	300528	454380	610413	758187
(C7) <b>Conv33His</b>	11220	104908	200220	302955	406653	505398
(C10) <b>Conv0.08His</b>	24	327	543	735	969	1152

Na [Figura 32](#), [Figura 33](#) e [Figura 34](#) são exibidos os resultados de desempenho para consultas com predicado convencional de 50% (C4), 33% (C7) e 0,08% (C10), respectivamente. Para todas as configurações com valores de seletividade do predicado de similaridade de 1% e 10% não houve diferença significativa de desempenho entre *BrOmnImg<sup>CF</sup>* e *BrOmnImg*. Isso deve-se ao fato de que o predicado de similaridade atua como um filtro eficiente para os dados sendo processados.

*BrOmnImg<sup>CF</sup>* proveu ganhos de desempenho mais significativos sobre *BrOmnImg* para os demais valores de seletividade do predicado de similaridade de imagens (20%, 30%, 40% e 50%). Estes ganhos foram de 29,50% a 33,12% para C4 ([Figura 32](#)); 38,01% a 44,92% para C7 ([Figura 33](#)); e 50,24% a 54,21% para C10 ([Figura 34](#)). Para estas configurações, a etapa de filtragem do predicado de similaridade não foi eficiente. Como resultado, o predicado convencional atuou como um filtro eficiente para os elementos candidatos ao predicado de similaridade. Observando-se os resultados de desempenho, conclui-se que, quanto mais restrito o predicado convencional, maior é o ganho de desempenho de *BrOmnImg<sup>CF</sup>* ([Figura 35](#)).

Figura 32 – Testes de desempenho comparando o método proposto *BrOmnImg<sup>CF</sup>* com o método proposto *BrOmnImg*, para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 50% (C4) de seletividade do predicado convencional. Fonte: Elaborado pelo autor.

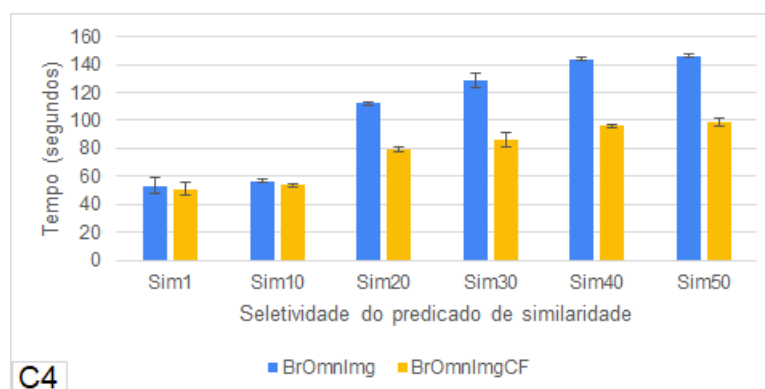


Figura 33 – Testes de desempenho comparando o método proposto *BrOmnImg<sup>CF</sup>* com o método proposto *BrOmnImg*, para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 33% (C7) de seletividade do predicado convencional. Fonte: Elaborado pelo autor.

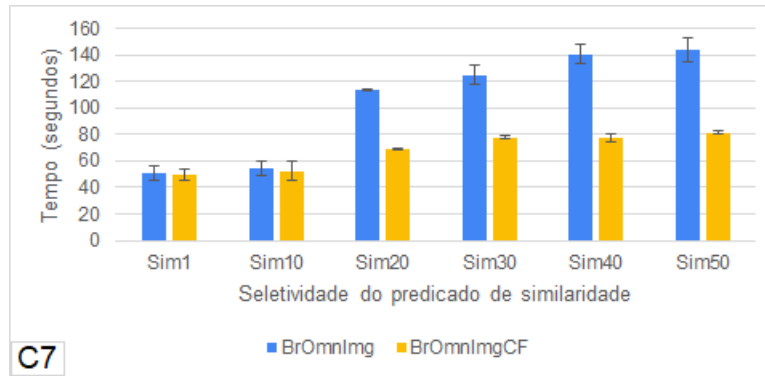
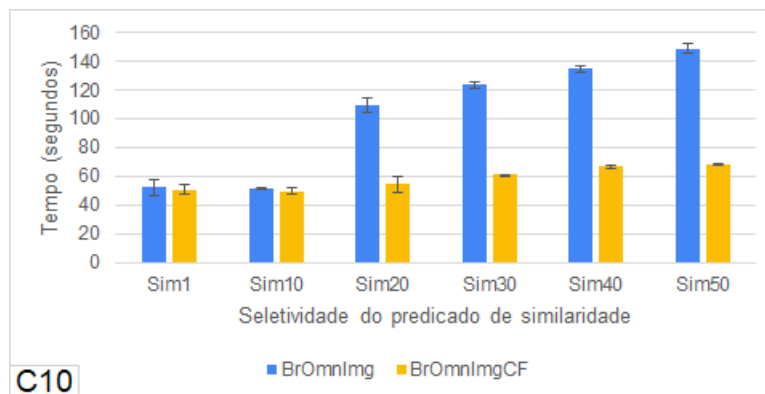


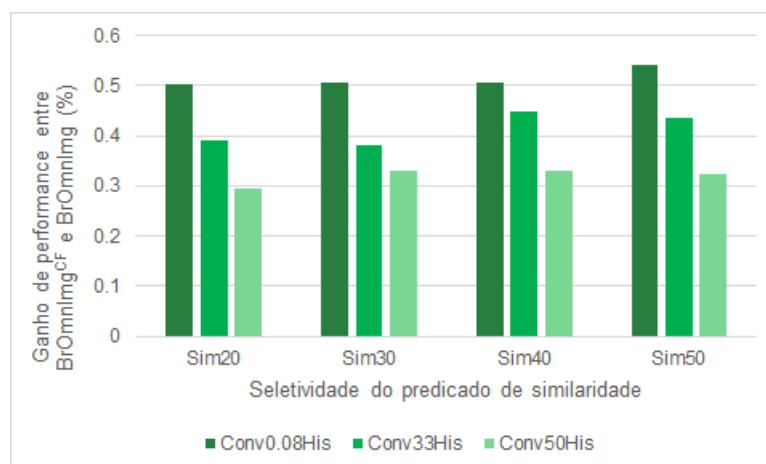
Figura 34 – Testes de desempenho comparando o método proposto *BrOmnImg<sup>CF</sup>* com o método proposto *BrOmnImg*, para o Volume 1 de dados, variando a seletividade do predicado de similaridade, considerando 0,08% (C10) de seletividade do predicado convencional. Fonte: Elaborado pelo autor.



Na Figura 35 são sumarizados os resultados de desempenho da Figura 32 a Figura 34 considerando uma visualização diferente desses resultados. Nessa figura, compara-se conjuntamente o desempenho de *BrOmnImg<sup>CF</sup>* com *BrOmnImg* para as configurações *Sim20*, *Sim30*, *Sim40* e *Sim50*, e a seletividade do predicado convencional de 0,08%, 33% e 50%. Nota-se que a diferença de desempenho é muito similar entre as seletividades do predicado de similaridade, porém, esta diferença é variada considerando a seletividade do predicado convencional.

Nesta seção foram descritos resultados de desempenho comparando *BrOmnImg<sup>CF</sup>* com *BrOmnImg*, enquanto que na seção 7.4.2 foram descritos resultados comparando *BrOmnImg* com o trabalho relacionado disponível na literatura SBJ. Considerando os testes de desempenho que foram realizados considerando o mesmo cenário, ou seja, Volume 1 de dados, Consulta 2 e Configuração 3, é possível realizar a seguinte análise cruzada. *BrOmnImg* obteve ganho de desempenho sobre SBJ para valores de seletividade do predicado de similaridade de 1%, 10% e 20% (Figura 29) e *BrOmnImg<sup>CF</sup>* proveu resultados próximos ou melhores do que *BrOmnImg* para as mesmas configurações (Figura 32, Figura 33 e Figura 34). É possível afirmar, portanto, que

Figura 35 – Porcentagem dos ganhos de performance para os testes de desempenho comparando  $BrOmnImg^{CF}$  com  $BrOmnImg$ , variando a seletividade do predicado convencional e de similaridade. Fonte: Elaborado pelo autor.



$BrOmnImg^{CF}$  também provê ganho de desempenho sobre SBJ para esses valores de similaridade. Para os valores de seletividade de 30%, 40% e 50% do predicado de similaridade de imagens, SBJ proveu ligeira vantagem sobre  $BrOmnImg$ , com ganho de desempenho variando de 1,27% a 7,72%. Entretanto, considerando esses valores de seletividade,  $BrOmnImg^{CF}$  proveu ganho de desempenho sobre  $BrOmnImg$  que variou de 29,50% a 54,21%. Pode-se concluir, portanto, que  $BrOmnImg^{CF}$  também provê ganho de desempenho sobre SBJ para valores de seletividade maiores.

## 7.5 Considerações finais

Neste capítulo foram propostos os métodos  $BrOmnImg$  e  $BrOmnImg^{CF}$ , os quais integram as técnicas *broadcast join* e *Omni* para o processamento eficiente de consultas analíticas estendidas com predicado de similaridade de imagens em Spark. O método  $BrOmnImg$  é mais genérico, podendo ser aplicado a qualquer cenário no qual existe memória primária suficiente para utilizar estruturas exatas. O método  $BrOmnImg^{CF}$  estende o método  $BrOmnImg$  para usar o predicado convencional para filtrar o predicado de similaridade. Entretanto, o método  $BrOmnImg^{CF}$  é mais restrito no sentido que requer que a tabela de fatos tenha um relacionamento unário com a tabela que armazena os vetores de características das imagens.

As vantagens introduzidas por  $BrOmnImg$  e  $BrOmnImg^{CF}$  foram demonstradas por meio de testes de desempenho experimentais.  $BrOmnImg$  foi comparado com o trabalho da literatura mais relacionado, SBJ. Analisando-se a variação da seletividade apenas do predicado de similaridade de imagens,  $BrOmnImg$  proveu ganhos de desempenho de até 56,94%. Para a variação do predicado convencional e da dimensionalidade das camadas perceptuais envolvidas,  $BrOmnImg$  proveu ganhos de desempenho similares para diferentes volumes de dados, os quais foram de até 65,49%.  $BrOmnImg^{CF}$  foi comparado com  $BrOmnImg$ . Analisando-se o impacto

do predicado convencional sobre o predicado de similaridade, *BrOmnImg<sup>CF</sup>* proveu ganhos de desempenho de até 54,21%.

Consultas semânticas, como as descritas na seção 6.4.2, também podem ser realizadas usando-se *BrOmnImg* e *BrOmnImg<sup>CF</sup>*. Elas foram omitidas neste capítulo porque os resultados retornados pelos métodos *BloOmnImg*, *BrOmnImg* e *BrOmnImg<sup>CF</sup>* são exatamente os mesmos. A diferença entre os métodos é o uso da estrutura probabilística ou exata subjacente, o que possibilita a execução dos algoritmos em diferentes configurações de memória primária disponível.

No próximo capítulo, Capítulo 8, o método *BrOmnImg* é estendido para processar eficientemente consultas analíticas estendidas não somente com fatores de similaridade de imagens, mas também com fatores de similaridade de dados geográficos e socioeconômicos. Propõe-se o método *SimSparkOLAP* para esse fim.



---

## MÉTODO *SimSparkOLAP*

---

### 8.1 Considerações iniciais

Neste capítulo é introduzido o método proposto *SimSparkOLAP*, o qual estende o método *BrOmnImg* para realizar o processamento eficiente de consultas analíticas estendidas com predicados de similaridade de imagens, geográficos e socioeconômicos. Na seção 8.2 é detalhada a motivação para a proposta do método. Na seção 8.3, *SimSparkOLAP* é introduzido. Na seção 8.4 é feita a validação do método. O capítulo é finalizado na seção 8.5, com as considerações finais.

### 8.2 Motivação

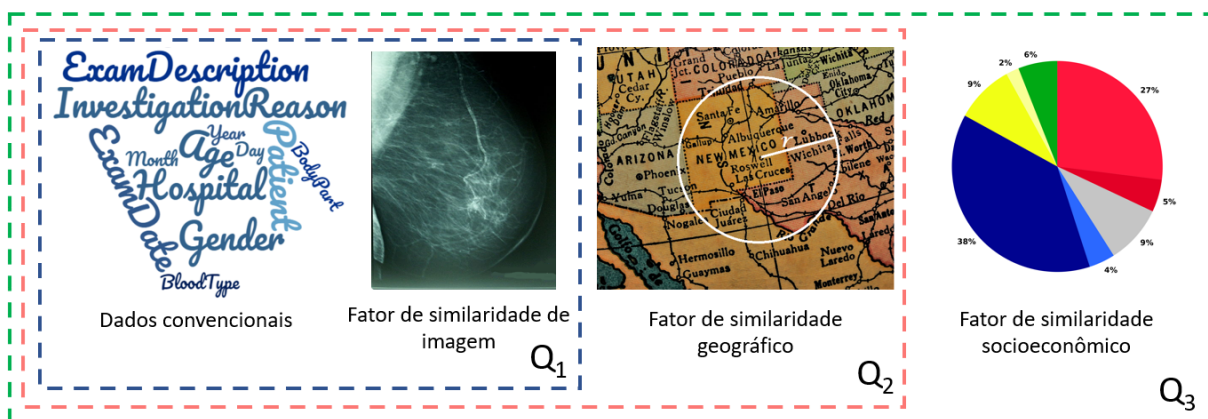
A organização de um DW pode beneficiar a investigação de vários assuntos de interesse, considerando diferentes fatores de análise. Na área médica, diferentes tipos de dados podem ser considerados: (i) dados convencionais (por exemplo, numéricos, alfanuméricos e datas); (ii) dados de imagens (por exemplo, vetores de características que descrevem características intrínsecas de imagens); (iii) dados geográficos (por exemplo, cidades representadas por um ponto que é definido pela latitude e longitude); e (iv) dados socioeconômicos (por exemplo, faixa de idade da população e faixa salarial). Neste capítulo, esses diferentes tipos de dados são considerados no contexto da área médica.

Na [Figura 36](#) são ilustradas consultas de interesse envolvendo diferentes fatores de similaridade para a área médica. Considere como base o DW descrito na [Figura 8](#), o qual integra dados relacionados a exames. Dados convencionais e de imagens destes exames são coletados ao longo de anos, os quais pertencem ao tratamento de pacientes em diferentes hospitais. Os hospitais estão localizados em muitas cidades, sendo que cada uma pode ser adicionalmente descrita por dados convencionais, geográficos (latitude e longitude) e socioeconômicos. As

seguintes consultas podem ser realizadas sobre esse DW estendido:

- $Q_1$ . “Quantos exames realizados entre 1970 e 2010 possuem imagens similares a uma dada imagem de câncer de mama?”
- $Q_2$ . “Quantos exames possuem imagens similares a uma dada imagem de câncer de mama, pertencem a pacientes do sexo feminino que realizaram os exames em hospitais privados localizados em torno de até 15 km de New York?”.
- $Q_3$ . “Quantos exames possuem imagens similares a uma dada imagem de câncer de mama, para pacientes com diferentes faixas de idade e estados geográficos, os quais foram realizados em hospitais localizados em torno de até 1.500 km de New York cuja população tenha uma faixa de idade similar à faixa etária da população de New York?”.

Figura 36 – Dados de imagens, geográficos e socioeconômicos como fatores de similaridade. Fonte: Elaborado pelo autor.



A consulta  $Q_1$  contém atributos convencionais de exames e datas. Ela requer a busca por similaridade de imagens, conforme feito nos Capítulos 6 e 7. A consulta  $Q_2$  possui tanto atributos convencionais de exames, pacientes e hospitais quanto inclui busca por similaridade de dados de imagens e dados geográficos. A consulta  $Q_3$  estende  $Q_2$  definindo também uma busca por similaridade de dados socioeconômicos (Figura 36).

Portanto, considerar dados de imagens, geográficos e socioeconômicos em consultas analíticas estendidas com predicado de similaridade enriquece a gama de consultas que pode ser realizada na tomada de decisão da área médica.

### 8.3 Descrição do método SimSparkOLAP

Nesta seção, propõe-se SimSparkOLAP (acrônimo para *efficiently process OLAP queries extended with geographic, socioeconomic, and image similarity search predicates in Spark*), um

método que visa processar de modo eficiente consultas analíticas estendidas com predicados de similaridade de imagens, geográficos e socioeconômicos, utilizando o *framework* Spark.

Antes de descrever o método proposto na seção 8.3.2, na seção 8.3.1 são introduzidos diferentes esquemas-estrela para investigar os fatores de similaridade de dados geográficos e socioeconômicos.

### 8.3.1 Esquemas

Nesta seção, propõe-se três esquemas-estrela que contêm fatores de similaridade de imagens, geográficos e socioeconômicos (Figura 37). Para este fim, estende-se o DW de imagens da área médica, descrito no capítulo 3. Ou seja, são reutilizadas as seguintes tabelas do DW de imagens: (i) tabela de fatos *Exame*; (ii) tabelas de dimensão convencional *DataExame*, *DescriçãoExame*, *Paciente* e *Idade*; (iii) tabela dos vetores de características *VetCaractImg*; e (iv) tabelas das camadas perceptuais *Histograma de Cor* e *Haralick Variância*.

Para considerar dados geográficos e socioeconômicos, armazena-se também no DW de imagens estendido dados relacionados com as cidades nas quais hospitais que realizam os exames estão localizados. O fator de similaridade de dados geográficos é definido por um ponto e retratado pelo pictograma ●, sendo representado por um vetor de características composto pelos atributos *Latitude* e *Longitude*. Já o fator de similaridade de dados socioeconômicos possui as características dessas cidades, como distribuição de idade, etnia e lares, sendo representado por um vetor de características composto pelas atributos *SocioCaract*<sub>1</sub>, ..., *SocioCaract*<sub>q</sub>. Os esquemas-estrela propostos diferem em como armazenam os fatores de similaridade geográficos e socioeconômicos.

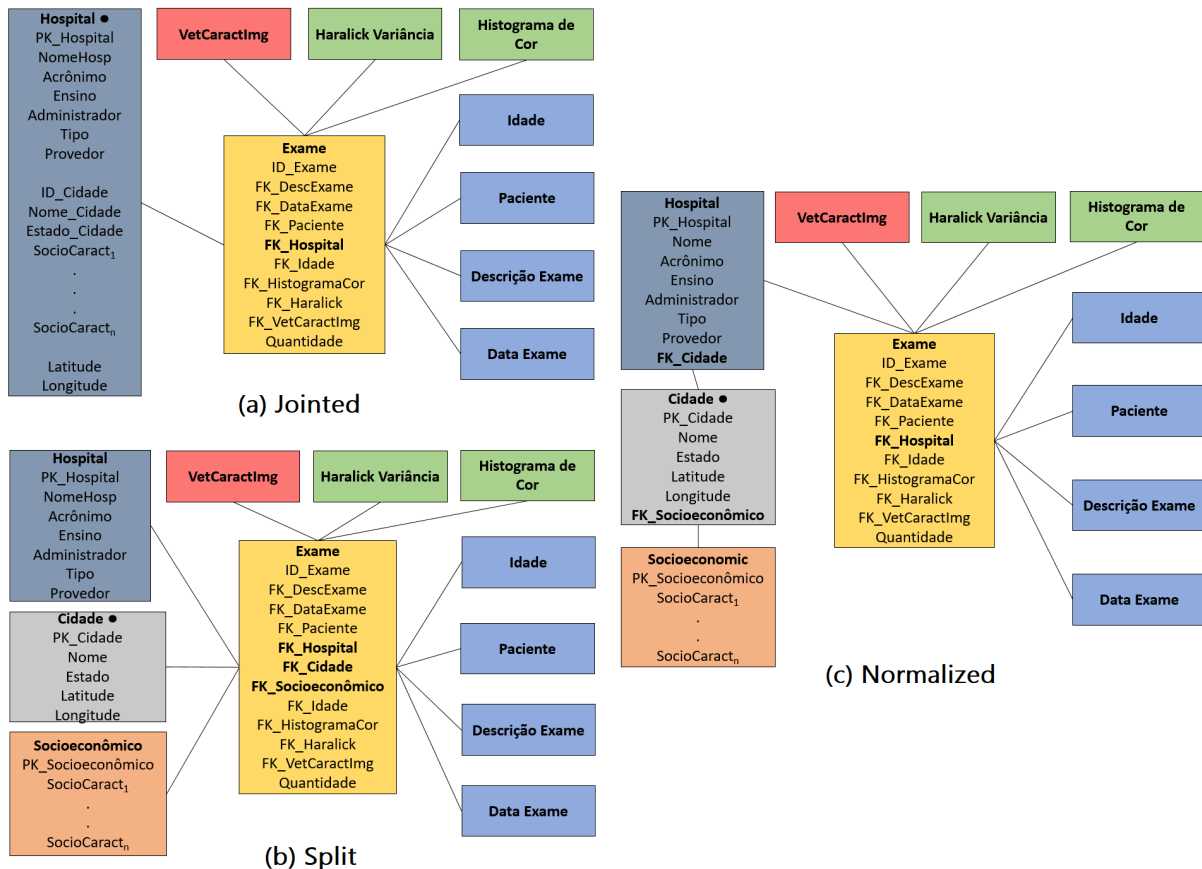
**Esquema *jointed*** (Figura 37a). Neste esquema, os dados geográficos e socioeconômicos são armazenados na tabela de dimensão *Hospital*. Além desses atributos convencionais, esta tabela possui os vetores de características geográficos e socioeconômicos das cidades. Apenas *Hospital* é relacionada com a tabela de fatos *Exame*.

**Esquema *split*** (Figura 37b). Neste esquema, os dados geográficos e socioeconômicos são armazenados nas tabelas de dimensão *Cidade* e *Socioeconomico*, respectivamente. Assim, a tabela de dimensão *Hospital* armazena apenas os atributos convencionais; a tabela de dimensão de similaridade *Cidade* armazena o vetor de características geográficas das cidades; e a tabela de dimensão de similaridade *Socioeconomico* armazena o vetor de características socioeconômicas das cidades. Cada uma destas tabelas (*Hospital*, *Cidade* e *Socioeconomico*) é relacionada com a tabela de fatos *Exame*.

**Esquema *normalized*** (Figura 37c). Neste esquema, os dados geográficos e socioeconômicos são armazenados nas tabelas de dimensão *Cidade* e *Socioeconomico*, respectivamente. Assim, cada tabela armazena apenas seus atributos correspondentes. Porém, a tabela de dimensão de similaridade *Socioeconomico* é relacionada com a tabela de dimensão de similaridade *Cidade*,

a qual é relacionada com a tabela de dimensão convencional *Hospital*, que por sua vez é relacionada com a tabela de fatos *Exame*. O relacionamento entre as tabelas é feito respeitando a granularidade dos atributos.

Figura 37 – Três diferentes esquemas-estrela para um DW com fatores de similaridade de imagens, geográficos e socioeconômicos: *jointed*, *split* e *normalized*. Fonte: Elaborado pelo autor.



### 8.3.2 Método

As consultas de interesse para *SimSparkOLAP* consistem em uma extensão das consultas definidas na seção 8.2. Ou seja, elas são compostas pelos seguintes predicados: (i) convencional, baseado em condições de seleção; (ii) fator de similaridade de imagens, composto por uma operação de similaridade e uma ou mais camadas perceptuais; (iii) fator de similaridade de dados geográficos, especificado em termos de uma operação de similaridade; (iv) fator de similaridade de dados socioeconômicos, composto por uma operação de similaridade e uma ou mais características socioeconômicas. Por exemplo, para a consulta  $Q_3$  definida na Figura 36: (i) *câncer de mama* representa o predicado convencional, (ii) operação de similaridade *range query* e camada perceptual *Histograma de Cor* representam o predicado de imagem; (iii) operação de similaridade *range query* representa o predicado geográfico; e (iv) operação de similaridade *range query* e *faixa de idade*, como fator de dados socioeconômicos envolvido, representam o predicado de dados socioeconômicos.

Na **Figura 38** é ilustrada uma visão geral do método *SimSparkOLAP* usando o mesmo esquema de cores da **Figura 37**. *SimSparkOLAP* realiza as seguintes tarefas para processar as consultas de interesse. Cada tabela de dimensão convencional  $i$  ( $1 \leq i \leq m$ ) envolvida no predicado convencional é acessada para processar as seleções convencionais aplicadas sobre essas tabelas. *SimSparkOLAP* armazena cada conjunto  $i$  do filtro convencional na estrutura *HashMapConvencional<sub>i</sub>* (**Figura 38a**).

Para o fator de similaridade de imagens, a operação de similaridade é aplicada sobre todas as camadas perceptuais envolvidas. Na etapa de filtragem, os elementos de cada tabela  $j$  ( $1 \leq j \leq n$ ) são filtrados utilizando a técnica Omni, gerando um conjunto  $j$  de elementos candidatos ao predicado de imagem para cada camada perceptual, os quais são armazenados na estrutura *HashMapFiltragem<sub>j</sub>* (**Figura 38b**). Assim, os elementos candidatos de cada camada perceptual são analisados na etapa de refinamento para eliminar falsos positivos. Para isso, a tabela *VetCaractImg* é acessada e as distâncias são calculadas para determinar o resultado do predicado de imagem. O conjunto final dos resultados é armazenado na estrutura *HashMapRefinamento* (**Figura 38c**).

*SimSparkOLAP* processa os fatores de similaridade de dados geográficos e socioeconômicos como descrito a seguir, de acordo com cada esquema proposto (**Figura 37**).

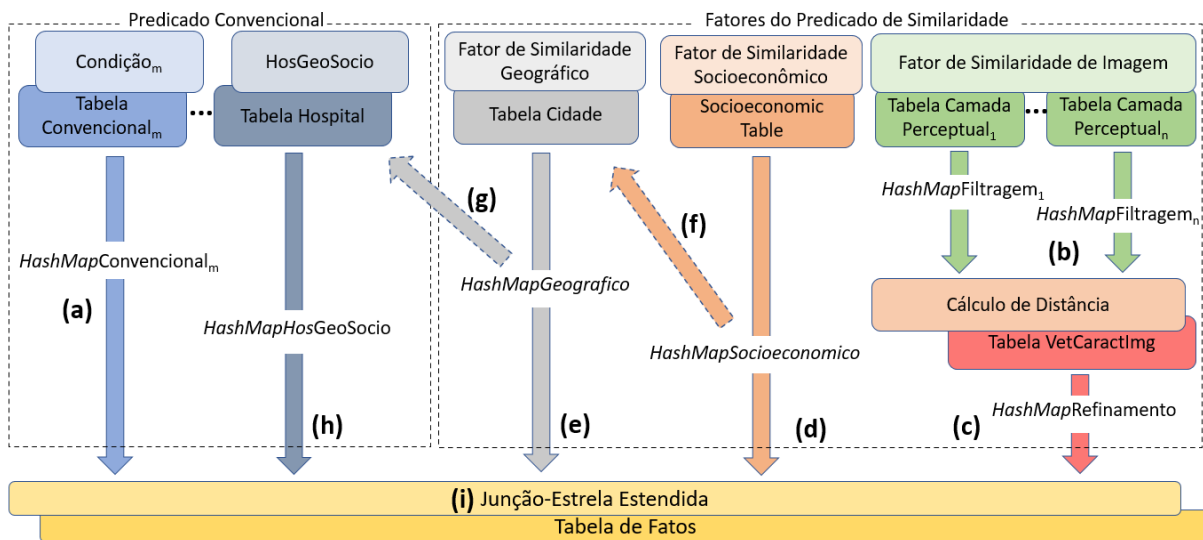
**Esquema *jointed*** (**Figura 37a**). Os fatores de similaridade de dados geográficos e socioeconômicos são processados a partir da tabela de dimensão *Hospital*, além das condições de seleção do predicado convencional que podem ser aplicados nesta tabela. Os resultados são armazenados na estrutura *HashMapHosGeoSocio* (**Figura 38h**). Finalmente, todas as estruturas são transmitidas para todos os nós do *cluster* e, com isso, a junção-estrela estendida é realizada na tabela de fatos *Exame* (**Figura 38i**).

**Esquema *split*** (**Figura 37b**). Os fatores de similaridade de dados geográficos e socioeconômicos são processados a partir das tabelas *Cidade* e *Socioeconomico*, respectivamente. O conjunto de resultados desses fatores são armazenados em suas respectivas estruturas, *HashMapGeografico* (**Figura 38e**) e *HashMapSocioeconomico* (**Figura 38d**). Além disso, o predicado convencional, definido sobre a tabela de dimensão convencional *Hospital*, é processado e seu resultado armazenado na estrutura *HashMapHosGeoSocio* (**Figura 38h**). Finalmente, todas as estruturas são transmitidas para todos os nós do *cluster* e, com isso, a junção-estrela estendida é realizada na tabela de fatos *Exame* (**Figura 38i**).

**Esquema *normalized*** (**Figura 37c**). O fator de similaridade de dados socioeconômicos é processado a partir da tabela de dimensão de similaridade *Socioeconomico* e seus resultados são armazenados na estrutura *HashMapSocioeconomico*. Assim, esta estrutura é relacionada com a tabela *Cidade* (**Figura 38f**), onde o fator de similaridade de dados geográficos é processado, gerando resultados considerando os fatores de similaridade de dados geográficos e socioeconômicos, os quais, são armazenados na estrutura *HashMapGeografico*. A partir disso, essa estrutura é associada à tabela convencional *Hospital* (**Figura 38g**), onde as condições de seleção sobre

esta tabela são aplicadas e seus resultados, considerando estes fatores de similaridade e predicado convencional em *Hospital*, são armazenados em *HashMapHosGeoSocio* (Figura 38h). Finalmente, essa estrutura é transmitida para todos os nós do *cluster* e, com isso, a junção-estrela estendida é realizada na tabela de fatos *Exame* (Figura 38i).

Figura 38 – Visão geral do método SimSparkOLAP. Fonte: Elaborado pelo autor.



## 8.4 Validação

A validação do método proposto foi realizada por meio de testes de desempenho experimentais e consultas semânticas dentro do contexto da área médica. Primeiramente, é descrito o ambiente de teste e, na sequência, os resultados obtidos.

### 8.4.1 Ambiente de teste

**Dados.** Os experimentos foram conduzidos baseados em um DW da área médica com o objetivo de avaliar as vantagens das propostas dos esquemas e métodos apresentados. Estes objetivos foram divididos em: (a) determinar o efeito da utilização de diferentes esquemas no processamento de consultas analíticas estendidas com predicado de fatores de similaridade; e (b) investigar a importância da nova gama de análises que estas consultas permitem ser realizadas na tomada de decisão da área médica.

Utilizou-se a ferramenta *ImgDW Generator* (ROCHA; CIFERRI, 2018) para povoar a tabela de fatos, tabelas de dimensão convencionais e tabelas relacionadas com o fator de similaridade de imagens. A ferramenta gerou dados reais de imagens médicas e dados sintéticos para as demais tabelas. O número de tuplas geradas para cada tabela é descrito na Tabela 12. Utilizou-se também dados reais de 25.000 cidades, correspondente ao censo dos Estados Unidos

do ano de 2000<sup>1</sup>, para povoar as tabelas relacionadas aos fatores de similaridade de dados geográficos e socioeconômicos. Estes dados consideram um país grande, com um grande número de cidades e uma média de 4 hospitais por cidade. No contexto médico, esta média é uma boa representação da realidade. Na [Tabela 12](#), o número de tuplas nas tabelas *Hospital*, *Cidade* e *Socioeconômico* são referentes ao esquema *normalized*.

Tabela 12 – Volume de dados para a realização dos experimentos de consultas analíticas estendidas com predicado de fatores de similaridade. Fonte: Elaborado pelo autor.

<b>Tabela</b>	<b>Tuplas</b>
Exame	30 milhões
DataExame	18.268
DescriçãoExame	3 milhões
Paciente	300.000
Idade	121
Histograma de Cor	3 milhões
Haralick Variância	3 milhões
VetCaractImg	3 milhões
Hospital	100.000
Cidade	25.000
Socioeconômico	25.000

**Consultas 1.** A primeira classe de consultas analíticas estendidas com fatores de similaridade de dados geográficos e socioeconômicos foi definida segundo os seguintes predicados: (i) predicado convencional em *Hospital*; (ii) predicado de similaridade geográfico: *cidades dentro de um dado raio*; e (iii) predicado de similaridade socioeconômico: *distribuição da faixa de idade*. Na operação *range query* foi utilizada a função de distância *Euclidiana* para os fatores de imagens e de dados socioeconômicos; e a função *GCDist* para o fator de dados geográficos. O fator de similaridade de imagens não foi investigado porque *SimSparkOLAP* estende *BrOmnImg* e esse fator já foi considerado nos testes de desempenho descritos na seção 7.4.

**Consultas 2.** A segunda classe de consultas analíticas estendidas com fatores de similaridade de dados geográficos e socioeconômicos foi definida segundo as consultas  $Q_1$ ,  $Q_2$  e  $Q_3$  descritas na seção 8.2.

**Configurações.** O conjunto de configurações considerou as seguintes variações: (i) predicado convencional e fator de similaridade geográfico; (ii) fatores de similaridade de dados geográficos e socioeconômicos; e (iii) predicado convencional e fatores de similaridade de dados geográficos e socioeconômicos. Para cada configuração, foram variados os valores de seletividade de cada condição apresentada conjuntamente: 1%, 25% e 50% ([Tabela 13](#)). Quanto menor a seletividade, maior o número de tuplas como resultado. Para obter estes valores de seletividade, configurou-se o centro de consulta *New York* com raios de 35 km, 900 km e 1.500 km, respectivamente. Para

<sup>1</sup> <https://www.census.gov/programs-surveys/decennial-census/decade.2000.html>

os outros fatores, os valores de seletividade foram controlados limitando o número de tuplas retornadas.

Tabela 13 – Configurações para a realização de experimentos, com foco em desempenho, de consultas analíticas estendidas com predicado de fatores de similaridade de dados geográficos e socioeconômicos. Fonte: Elaborado pelo autor.

	1%	25%	50%
<b>HosGeo</b>	HosGeo (1%)	HosGeo (25%)	HosGeo (50%)
<b>GeoSocio</b>	GeoSocio (1%)	GeoSocio (25%)	GeoSocio (50%)
<b>HosGeoSocio</b>	HosGeoSocio (1%)	HosGeoSocio (25%)	HosGeoSocio (50%)

**Execução.** Foi usado um cluster com 5 nós, cada qual com, no mínimo, 3GB de RAM. Cada configuração definida foi executada 10 vezes, com remoção de *outliers*. Além disso, *caches* e *buffers* foram esvaziados depois de cada execução. Os resultados são ilustrados em segundos. Também são representados a média e o desvio padrão. Por fim, a linguagem Java foi considerada para a implementação.

**Trabalho relacionado.** Dado que o objetivo desta proposta é investigar o processamento dos fatores de similaridade de imagens, geográficos e socioeconômicos, relacionado com o esquema necessário para processá-los, foi considerado como trabalho relacionado os próprios métodos propostos.

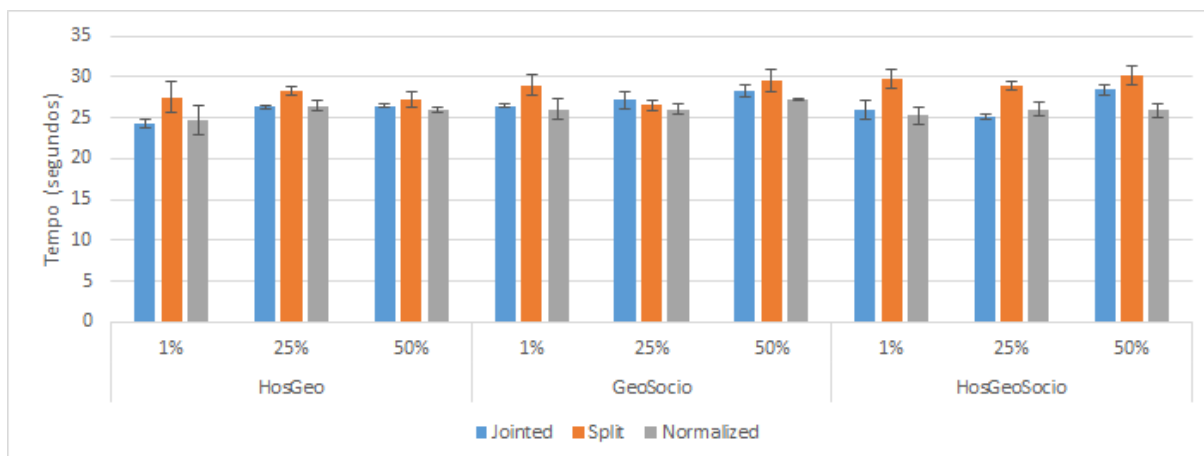
### 8.4.2 Resultados

Foram definidos 2 experimentos para analisar o método proposto. O primeiro experimento teve como objetivo **investigar o desempenho do método** considerando os diferentes esquemas-estrela introduzidos na seção 8.3.1. Foram considerados o volume de dados descrito na Tabela 12, classe de Consultas 1 e o conjunto de Configurações descrito na tabela Tabela 13. Na Figura 39 são apresentados os resultados obtidos para cada configuração de consulta e cada esquema proposto. Para a maioria das configurações, *SimSparkOLAP* proveu melhor desempenho sobre o esquema *normalized*, seguido de perto pelo esquema *jointed*, e depois pelo esquema *split*. Comparado com o esquema *split*, o esquema *normalized* possibilitou ganhos de desempenho de até 15,89%. Isso deve-se ao fato de que o esquema *normalized* permite que *SimSparkOLAP* realize junções entre tabelas significativamente menores (*Socioeconomico*, *Cidade* e *Hospital*), antes de processar a junção-estrela com a enorme tabela de fatos *Exame*. Comparado com o esquema *split*, o esquema *jointed* possibilitou ganhos de desempenho de até 13,68%. Isto ocorre por conta do esquema *jointed* armazenar todos os atributos dos fatores de similaridade de dados geográficos e socioeconômicos em *Hospital*. Como resultado, *SimSparkOLAP* realiza apenas uma junção com *Exame* para processar a consulta. Por outro lado, o esquema *split* requer mais junções-estrela no processamento da consulta.

O segundo experimento teve como objetivo **investigar o potencial semântico das consultas**, já que parte dos dados considerados são sintéticos. Foram considerados o volume de



Figura 39 – Testes experimentais de *SimSparkOLAP* focado na performance das consultas, considerando os métodos propostos. Fonte: Elaborado pelo autor.



dados descrito na Tabela 12 e a classe de Consultas 2. *SimSparkOLAP* foi utilizado para investigar exemplos de consultas semânticas e o quanto elas podem ser relevantes no suporte à tomada de decisão na área médica. Para isso, considerou-se as consultas  $Q_1$ ,  $Q_2$  e  $Q_3$  descritas na Figura 36. Definiu-se a camada perceptual *Histograma de Cor* como fator de similaridade de imagens. Baseado nos resultados providos pelo primeiro experimento, utilizou-se, o esquema *normalized* para o processamento destas consultas. Na Figura 40 são exibidos os resultados obtidos, os quais são discutidos a seguir.

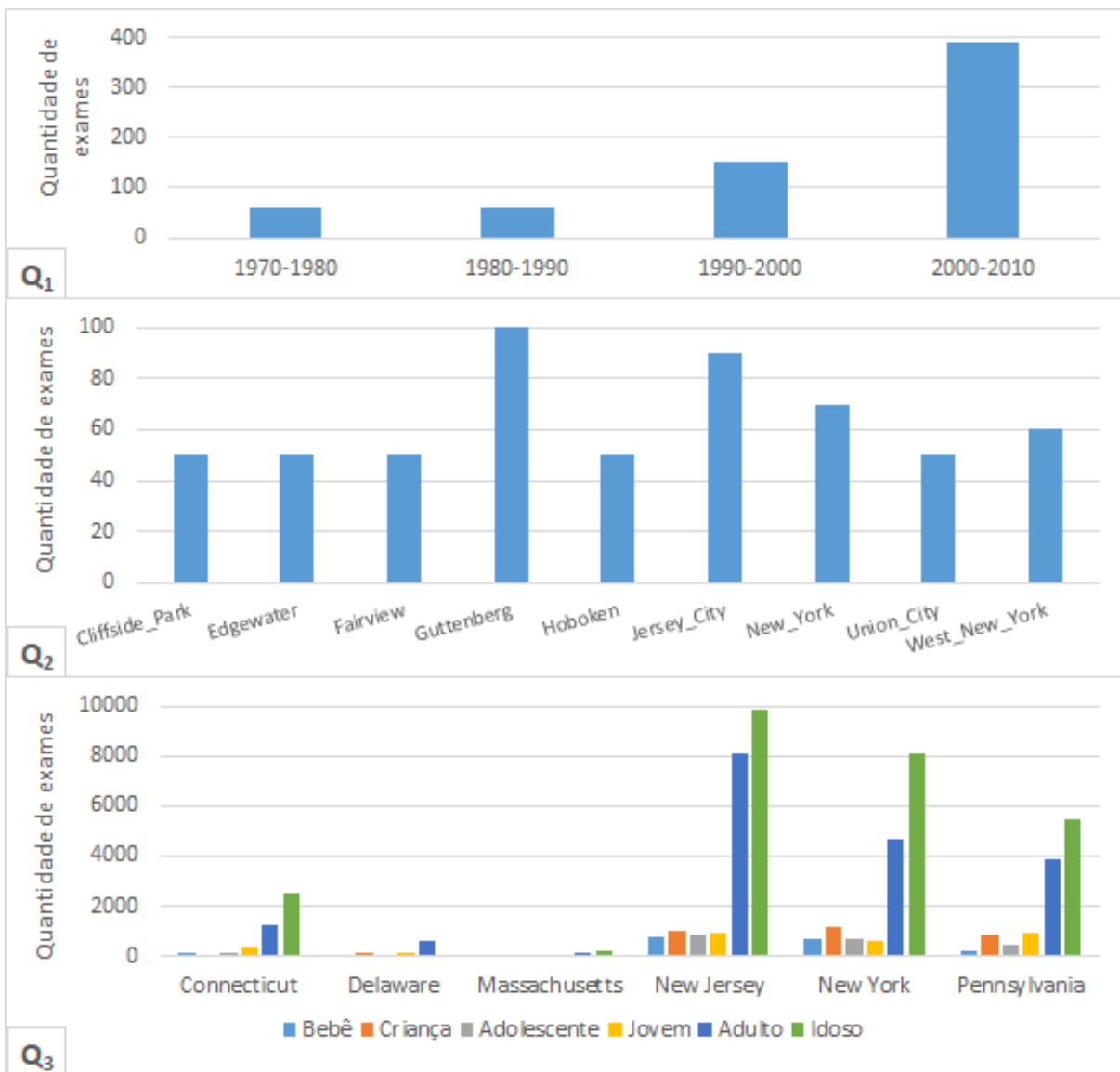
Para prover uma melhor visualização da consulta  $Q_1$ , ou seja, “Liste a quantidade de exames realizados de 1970 até 2010 que são similares a uma dada imagem de câncer de mama”, os anos foram agrupados em décadas. Na tomada de decisão, este tipo de consulta é importante para analisar a evolução da curva de uma determinada doença sobre o tempo, considerando dados convencionais e de imagens. Análises similares podem ser realizadas utilizando outros aspectos, como o tipo do sangue do paciente. Por exemplo, existem estudos sobre o COVID-19 que apontam o tipo “A” como o mais suscetível e o tipo “O” como o mais resistente (ZHAO; et al., 2020).

No resultado da consulta  $Q_2$  é exibido, para cada cidade, quantos exames possuem imagens similares a uma dada imagem de câncer de mama para pacientes do sexo feminino que realizaram exames em hospitais privados localizados em cidades em torno de até 15 km de *New York*. Na tomada de decisão da área médica, investigar áreas geográficas em torno de um ponto de interesse pode revelar o avanço de uma doença no espaço ou identificar epicentros. Também é possível explorar o quanto uma doença pode influenciar em outras, tal como comorbidades relacionadas ao COVID-19 (RICHARDSON; et al., 2020).

Para cada faixa de idade e o estado da localização do paciente, na consulta  $Q_3$  é retornada a quantidade de exames que possuem imagens similares a uma determinada imagem de câncer de mama, realizados em hospitais localizados em cidades em torno de até 1.500 km de *New*

*York* e que possuem uma distribuição de faixa de idade similar à de *New York*. Analisar dados socioeconômicos é muito importante no contexto da tomada de decisão da área médica. Assim, é possível identificar pontos de interesse com alto/baixo número de casos e como uma dada doença afeta pessoas a partir de diferentes faixas de idade, faixa salarial e níveis de educação. Por exemplo, é conhecido que há uma alta taxa de letalidade do COVID-19 para idosos (MAHASE, 2020).

Figura 40 – Exemplos sintéticos de consultas semânticas providas por *SimSparkOLAP*. Fonte: Elaborado pelo autor.



## 8.5 Considerações finais

Neste capítulo foi proposto o método *SimSparkOLAP*, o qual estende o método *BrOm-nImg* para o processamento eficiente de consultas analíticas estendidas com predicado de simi-

laridade geográficos, socioeconômicos e de imagens em Spark. As contribuições introduzidas foram: (i) proposta de três esquemas-estrela que investigam diferentes formas de projeto para os fatores de dados geográficos e socioeconômicos; (ii) validação de *SimSparkOLAP* considerando experimentos voltados à análise de desempenho e à análise semântica das consultas investigadas e sua relevância para a tomada de decisão médico-analítica; e (iii) proposta do método *SimSparkOLAP* e adequação desse método em termos de cada esquema-estrela proposto.

Para os fatores de similaridade geográficos e socioeconômicos, os resultados mostraram que armazenar estes dados de modo normalizado é vantajoso em relação aos outros esquemas. Porém, essa vantagem não é tão significativa, quando se considera também o desvio padrão. Além disso, o potencial das consultas semânticas analisadas foram relacionados com estudos em relação à pandemia do COVID-19, ressaltando a importância no suporte à tomada de decisão das mesmas, já que parte dos dados considerados são sintéticos.

No próximo capítulo, Capítulo 9, são descritas as conclusões desta dissertação de mestrado, as quais incluem os resultados obtidos, as dificuldades encontradas e possíveis trabalhos futuros.



---

## CONCLUSÃO

---

### 9.1 Considerações iniciais

Neste capítulo são descritas as conclusões finais da presente dissertação de mestrado. Na seção 9.2 são sumarizados os principais resultados obtidos. Na seção 9.3 são listados os artigos científicos produzidos durante o período de execução da dissertação. As dificuldades encontradas no desenvolvimento da presente dissertação são discutidas na seção 9.4. Por fim, na seção 9.5 são descritas as possibilidades de trabalhos futuros.

### 9.2 Contribuições

Nesta dissertação de mestrado foram considerados ambientes paralelos e distribuídos, os quais possuem alto poder de armazenamento e processamento. Esses requisitos são importantes dentro do contexto da área médica, na qual existe a necessidade de gerenciamento de grandes volumes de dados e do compartilhamento desses dados para a tomada de decisão médico-analítica. Nesse contexto, um DW de imagens da área médica oferece subsídios para a execução de consultas analíticas estendidas com predicado de similaridade de imagens. Isso permite que uma vasta gama de consultas analíticas envolvendo predicado convencional e predicado de similaridade de imagens possa ser realizada. Entretanto, essas consultas envolvem duas operações de alto custo computacional: junção-estrela e cálculo de similaridade. O desafio consiste em prover métodos que processem eficientemente essas consultas em ambientes paralelos e distribuídos, mais especificamente, em Spark. Assim, o presente trabalho foi motivado pela seguinte hipótese: *É possível processar eficientemente consultas analíticas estendidas com predicado de similaridade sobre um data warehouse de imagens em ambientes paralelos e distribuídos.*

Para a investigação dessa hipótese, foi feita uma revisão sistemática da literatura, a qual permitiu realizar um levantamento das principais técnicas e tecnologias que permeiam os

conceitos de interesse da dissertação. Por meio da revisão sistemática, foi identificado que não existem trabalhos na literatura que considerem conjuntamente todos esses conceitos, ou seja, o processamento eficiente da junção-estrela sobre um DW de imagem médicas que oferece suporte para similaridade, considerando ambientes paralelos e distribuídos.

Nesta dissertação de mestrado, essa lacuna existente na literatura foi investigada por meio da proposta de diferentes métodos, os quais representam contribuições do trabalho desenvolvido e são sumarizados nas seções 9.2.1 a 9.2.4.

### 9.2.1 Método *BloOmnImg*

O método proposto *BloOmnImg* (acrônimo para *Bloom Filter Omni for analytical Image queries*) integra as técnicas *bloom filter cascade join* e *Omni* para otimizar a junção-estrela e o cálculo de similaridade, respectivamente, no processamento das consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens. *BloOmnImg* é indicado para ambientes computacionais com pouca memória primária disponível. Isso está relacionado ao fato dele ser baseado no uso de *bloom filter*, que é uma estrutura probabilística e que, portanto, demanda menor espaço de armazenamento.

As vantagens introduzidas por *BloOmnImg* foram investigadas por meio de vários testes de desempenho experimentais. Inicialmente, foi investigada a variação de erro da estrutura do *bloom filter*, de forma a identificar qual erro deveria ser aplicado na criação dessa estrutura quando integrada ao *BloOmnImg*. Os testes de desempenho demonstraram que o menor erro deveria ser aplicado. Quanto menor o erro, menor o número de falsos positivos retornados.

Na sequência, foi investigado o desempenho do método proposto considerando apenas a variação da seletividade do predicado de similaridade. *BloOmnImg* foi comparado com o método mais próximo disponível na literatura, SBFCJ. Os testes de desempenho demonstraram que, quanto menor o raio da consulta, maior a vantagem de *BloOmnImg* sobre SBFCJ. O ganho de desempenho de *BloOmnImg* foi de 60,08% e 58,19% para raios menores e variou de 8,11% a 16,15% para raios menores.

Considerando a variação conjunta da seletividade das camadas perceptuais e do predicado convencional, os testes de desempenho demonstraram que consultas que possuem pelo menos uma camada perceptual de alta dimensionalidade favorecem a vantagem de *BloOmnImg* sobre SBFCJ. O ganho de desempenho de *BloOmnImg* variou de 58,83% a 66,93% para essas consultas. Os testes de desempenho também demonstraram que, para consultas que possuem uma camada perceptual de baixa dimensionalidade, *BloOmnImg* não obteve ganho de desempenho significativo em relação a SBFCJ.

Por fim, também foram realizados testes para mostrar semanticamente como as consultas investigadas nesta dissertação de mestrado são relevantes para a tomada de decisão da área médica. Foram realizadas consultas com potencial semântico, incluindo agrupamentos de atributos de

interesse. Como exemplo de utilidade, essas consultas foram relacionadas com a pandemia do COVID-19.

### 9.2.2 Método *BrOmnImg*

O método proposto *BrOmnImg* (acrônimo para *Broadcast Omni for processing analytical Image queries*) integra as técnicas *broadcast join* e *Omni* para otimizar a junção-estrela e o cálculo de similaridade, respectivamente, no processamento das consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens. *BrOmnImg* é indicado para ambientes computacionais com memória primária suficiente para utilizar estruturas exatas, as quais demandam maior espaço de armazenamento.

Nos testes de desempenho realizados para validar as vantagens introduzidas por *BrOmnImg*, ele foi comparado com SBJ, que é o método mais próximo disponível na literatura. Os resultados obtidos demonstraram que *BrOmnImg* possui comportamento muito similar a *BloOmnImg*. Ou seja, quanto menor o raio da consulta, maior a vantagem de *BrOmnImg* sobre SBJ. O ganho de desempenho de *BrOmnImg* foi de 56,75% e 56,94% para raios menores e variou de 1,27% a 13,31% para raios maiores. Adicionalmente, consultas que possuem pelo menos uma camada perceptual de alta dimensionalidade favoreceram a vantagem de *BrOmnImg* sobre SBJ, cujo ganho de desempenho variou de 57,32% a 64,47% para um DW de imagens com 3 milhões de exames e de 60,01% a 65,49% para 20 milhões de exames. Para consultas que possuem a camada perceptual de baixa dimensionalidade, não houve diferença significativa de desempenho entre *BrOmnImg* e SBJ.

### 9.2.3 Método *BrOmnImg<sup>CF</sup>*

O método proposto *BrOmnImg<sup>CF</sup>* (acrônimo para *Broadcast Omni for processing analytical Image queries with Conventional Filter*) é baseado no método *BrOmnImg*. Portanto, ele também integra as técnicas *broadcast join* e *Omni* para otimizar a junção-estrela e o cálculo de similaridade, respectivamente, no processamento das consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens. Entretanto, ele se difere do método *BrOmnImg* porque considera a seletividade do predicado convencional para ajudar a filtrar o predicado de similaridade. *BrOmnImg*, por sua vez, realiza o processamento do predicado convencional somente depois que os cálculos de distância já foram processados.

Nos testes de desempenho, *BrOmnImg<sup>CF</sup>* foi comparado com *BrOmnImg* para analisar as vantagens introduzidas pelo uso do predicado convencional como filtro. Foram investigadas diferentes variações de seletividade do predicado convencional e de similaridade. Os resultados demonstraram que, quando o predicado de similaridade retorna poucos resultados intermediários (alta seletividade), não há diferença de desempenho significativa entre *BrOmnImg<sup>CF</sup>* e *BrOmnImg*. Em contrapartida, quanto mais resultados intermediários são retornados pelo

processamento do predicado de similaridade, maior o ganho de desempenho de  $BrOmnImg^{CF}$ . Adicionalmente, quanto menor a seletividade do predicado convencional, maior o ganho de desempenho de  $BrOmnImg^{CF}$ . O ganho de desempenho de  $BrOmnImg^{CF}$  sobre  $BrOmnImg$  variou de 29,50% a 33,12%, 38,01% a 44,92% e 50,24% a 54,21% para a seletividade do predicado convencional igual a 50%, 33% e 0,08%, respectivamente.

Apesar das vantagens introduzidas por  $BrOmnImg^{CF}$ , ele é mais restrito do que  $BrOmnImg$ , podendo ser utilizado somente em situações nas quais existe uma correspondência única entre a tabela de fatos e a tabela de dimensão que armazena os vetores de características das imagens. Em contrapartida,  $BrOmnImg$  é mais genérico, pois não há restrição neste sentido.

### 9.2.4 Método SimSparkOLAP

O método proposto *SimSparkOLAP* (acrônimo para *efficiently process OLAP queries extended with geographic, socioeconomic, and image similarity search predicates in Spark*) integra as técnicas *broadcast join* e *Omni* para prover o processamento eficiente de consultas analíticas estendidas com predicado de similaridade de imagens, geográficos e socioeconômicos.

Foram propostos três esquemas-estrela para um DW da área médica estendido com tabelas voltadas para o armazenamento de dados geográficos e socioeconômicos. O método proposto *SimSparkOLAP* estendeu  $BrOmnImg$  para também processar fatores de similaridade geográficos e socioeconômicos, considerando os três esquemas-estrela proposto. Em geral, o esquema que armazena dados geográficos e socioeconômicos de forma normalizada possibilitou que *SimSparkOLAP* provesse os melhores resultados, com ganhos de desempenho de até 15,89%.

Por fim, foram realizados testes para indicar a relevância de consultas envolvendo predicado convencional e predicados de imagem, geográfico e socioeconômico. Com foco na tomada de decisão da área médica, os potenciais semânticos dessas consultas foram relacionados com possíveis análises semânticas para o COVID-19.

## 9.3 Divulgação de resultados

Durante o desenvolvimento desta dissertação de mestrado, os seguintes artigos científicos foram produzidos:

- Artigo intitulado “*The SimilarQL framework: similarity queries in plain SQL*” (TRAINA *et al.*, 2019), publicado em *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC)*, 2019. Neste artigo investiga-se o processamento de consultas que consideram diferentes operações de similaridade, como *range query*, *K-nn query*, *join range query*, *join K-nn query* e diferentes fatores de similaridade, como geográficos e socioeconômicos.



- Artigo intitulado “**Processamento Eficiente de Consultas Analíticas Estendidas com Predicado de Similaridade em Spark**” (ROCHA; CIFERRI, 2019a), publicado na sessão de *short papers* do XXXIV Simpósio Brasileiro de Banco de Dados (SBBD), 2019. Neste artigo, é descrita uma versão preliminar do método *BrOmnImg*. O artigo foi classificado entre os 5 melhores artigos do evento, tendo sido convidado para ser estendido e submetido no *Journal of Information and Data Management*.
- Artigo intitulado “**Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens em ambientes paralelos e distribuídos**” (ROCHA; CIFERRI, 2019b), publicado no Workshop de Teses e Dissertações em Banco de Dados (WTDBD) do XXXIV Simpósio Brasileiro de Banco de Dados (SBBD), 2019. O artigo descreve o objetivo desta dissertação de mestrado, bem como os resultados preliminares obtidos.
- Artigo intitulado “**Healthcare decision-making over a geographic, socioeconomic, and image data warehouse**”, aceito no *International Workshop on Intelligent Data – From Data to Knowledge*, 2020. Neste artigo é descrito o método *SimSparkOLAP*.

Adicionalmente, o seguinte artigo científico encontra-se em fase final de preparação:

- Artigo intitulado “***Efficient Processing of Analytic Queries Extended with Similarity Search Predicates over Images in Spark***”, a ser submetido para o *Journal of Information and Data Management*. Neste artigo está sendo detalhado o método *BrOmnImg*, bem como está sendo proposto o método *BrOmnImg<sup>CF</sup>*.

## 9.4 Dificuldades encontradas

As seguintes dificuldades foram encontradas no desenvolvimento dessa dissertação de mestrado:

- **Dados.** Esta dissertação de mestrado lida com diversos tipos de dados, como dados convencionais e dados complexos de imagens, geográficos e socioeconômicos. Coletar estes dados de diferentes fontes e armazená-los considerando ainda grandes volumes de dados, foi um desafio encontrado em diversas etapas da dissertação.
- **Utilização do *cluster* de computadores.** Lidar com o sistema de arquivos distribuídos HDFS e *framework* Spark foi complexo no início do desenvolvimento da dissertação, pela dificuldade em depurar problemas nos algoritmos. Além disso, o *cluster* disponível para experimentos possui limitações de memória, as quais limitaram algumas configurações de consulta.

- **Baseline.** Por um lado, em (TEIXEIRA *et al.*, 2015) foram propostas soluções para o processamento eficiente de consultas analíticas estendidas com predicado de similaridade de imagens, porém considerando apenas ambientes de *data warehousing* convencionais. Por outro lado, em (BRITO *et al.*, 2016) foram propostas soluções para o processamento eficiente da junção-estrela em Spark, porém não foram considerados *data warehouses de imagens*. Dessa forma, definir um *baseline* para comparação de desempenho dos métodos propostos foi alvo de discussão em todas as etapas da dissertação. Por fim, optou-se por ter um *baseline* mais próximo do estado da arte, que foram os métodos descritos em (BRITO *et al.*, 2016).
- **Background.** Esta dissertação lida com diversos assuntos complexos, como *data warehouse* de imagens, buscas por similaridade entre imagens, entre dados geográficos e entre dados socioeconômicos, e processamento paralelo e distribuído. Portanto, um primeiro desafio enfrentado foi relacionado ao aprendizado desses assuntos, com foco em absorver as características intrínsecas dos conceitos usados na dissertação. Outro desafio está relacionado ao fato de que os assuntos tratados estão muito relacionados a tecnologias. Portanto, foi um desafio descrever as contribuições desta dissertação de mestrado de modo a ser o mais independente possível dessas tecnologias.

## 9.5 Trabalhos futuros

Trabalhos futuros relacionados à presente dissertação são:

- **Aplicação real para a área médica.** Considerar um conjunto de dados real e volumoso para que as consultas analíticas estendidas com predicado de similaridade possam ser processadas usando os métodos propostos de forma a obter resultados semânticos e de relevância para a aplicação real. Por exemplo, as propostas feitas nesta dissertação de mestrado podem ser aplicadas na análise dos dados do COVID-19.
- **Considerar outras operações de similaridade.** Além da operação de similaridade *range query* considerada nos testes de desempenho dos métodos propostos, trabalhos futuros podem investigar também consultas analíticas estendidas com predicado de similaridade de imagens, geográficos e socioeconômicos definidos em termos da operação *K-nn*.
- **Considerar outras operações OLAP.** As consultas analíticas utilizadas na proposta dos métodos e nos testes de desempenho são caracterizadas por serem equivalentes às consultas *slice and dice*, *drill-down* e *roll-up*. Trabalhos futuros também podem considerar a operação *drill-across*, possibilitando uma nova abordagem semântica envolvendo medidas numéricas de tabelas de fatos diferentes.
- **Avaliar outros esquemas-estrela para o armazenamento de dados de imagens.** Avaliar os desempenhos dos métodos propostos *BloOmnImg* e *BrOmnImg* para os diferentes

esquemas-estrela ilustrados na (Figura 9), os quais se diferem no armazenamento dos dados referentes aos vetores de características das camadas perceptuais consideradas (ANNIBAL, 2011).

- **Configuração do cluster.** Trabalhos futuros também podem variar o número de nós considerados no *cluster* de computadores, avaliando o impacto do número de nós para os métodos propostos.
- **Fatores de similaridade.** Além dos fatores de similaridade geográficos, socioeconômico e de imagens, outros fatores podem ser considerados, como áudios, textos e vídeos. Nesse sentido, os métodos propostos devem ser estendidos para também serem capazes de processar esses outros fatores de similaridade.
- **Avaliação de especialista.** Desde que esta dissertação contextualiza-se na área médica, outro trabalho futuro consiste na realização de entrevistas com especialistas dessa área com o objetivo de avaliar a utilidade prática dos diversos tipos de consultas investigadas nos testes de desempenho, como feito em (TEIXEIRA *et al.*, 2015).
- **Uso de índices.** Os métodos propostos nesta dissertação de mestrado vislumbram a flexibilidade das consultas analíticas estendidas com predicado de similaridade, ou seja, consideram cenários nos quais é difícil prever os tipos de consulta OLAP que são executadas e também a frequência de ocorrência de um determinado predicado de alta seletividade. Um último trabalho futuro é estender os métodos propostos considerando outros cenários, os quais requerem o uso de índices, como feito em (BRITO *et al.*, 2020).



## REFERÊNCIAS

---

---

AFRATI, F. N.; ULLMAN, J. D. Optimizing joins in a map-reduce environment. In: **Proceedings of the 13th International Conference on Extending Database Technology**. New York, NY, USA: ACM, 2010. (EDBT '10), p. 99–110. ISBN 978-1-60558-945-9. Disponível em: <<http://doi.acm.org/10.1145/1739041.1739056>>. Citado nas páginas 32 e 76.

ANNIBAL, L. P. **Istar : um esquema estrela otimizado para Image Data Warehouses baseado em similaridade**. Dissertação (Mestrado em Ciências Exatas e da Terra) — Universidade Federal de São Carlos, São Carlos, 2011. Citado nas páginas 15, 31, 50, 52 e 129.

Annibal, L. P.; Felipe, J. C.; Ciferri, C. D. A.; Ciferri, R. R. icube: A similarity-based data cube for medical images. In: **2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.: s.n.], 2010. p. 321–326. ISSN 1063-7125. Citado nas páginas 45 e 52.

ARIYACHANDRA, T.; WATSON, H. J. Key organizational factors in data warehouse architecture selection. **Decision Support Systems**, v. 49, n. 2, p. 200–212, 2010. Disponível em: <<https://doi.org/10.1016/j.dss.2010.02.006>>. Citado na página 30.

ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R. H.; KONWINSKI, A.; LEE, G.; PATTERSON, D. A.; RABKIN, A.; ZAHARIA, M. **Above the Clouds: A Berkeley View of Cloud Computing**. [S.l.], 2009. Citado na página 58.

BATISTA, N. A.; SOUSA, G. A.; BRANDÃO, M. A.; SILVA, A. P. C. da; MORO, M. M. Tie strength metrics to rank pairs of developers from github. **Journal of Information and Data Management**, v. 9, n. 1, p. 69–83, 6 2018. Citado na página 70.

BHOWMIK, S. **Cloud Computing**. Cambridge University Press, 2017. ISBN 9781316638101. Disponível em: <<https://books.google.com.br/books?id=jeTFDgAAQBAJ>>. Citado na página 58.

BIOLCHINI, J. C. de A.; MIAN, P. G.; NATALI, A. C. C.; CONTE, T. U.; TRAVASSOS, G. H. Scientific research ontology to support systematic review in software engineering. **Advanced Engineering Informatics**, v. 21, n. 2, p. 133 – 151, 2007. ISSN 1474-0346. Ontology of Systems and Software Engineering; Techniques to Support Collaborative Engineering Environments. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S147403460600070X>>. Citado na página 69.

BLANAS, S.; PATEL, J. M.; ERCEGOVAC, V.; RAO, J.; SHEKITA, E. J.; TIAN, Y. A comparison of join algorithms for log processing in mapreduce. In: **Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2010. (SIGMOD '10), p. 975–986. ISBN 978-1-4503-0032-2. Disponível em: <<http://doi.acm.org/10.1145/1807167.1807273>>. Citado nas páginas 32, 33, 62 e 76.

BRITO, J. J. **Data Warehouses na era do Big Data: processamento eficiente de Junções Estrela no Hadoop**. Tese (Doutorado) — University of São Paulo, Brazil, 2017. Disponível

em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-23072018-111356/>>. Citado nas páginas 15, 59 e 80.

BRITO, J. J.; MOSQUEIRO, T.; CIFERRI, R. R.; CIFERRI, C. D. de A. Faster cloud star joins with reduced disk spill and network communication. **Procedia Computer Science**, v. 80, p. 74 – 85, 2016. ISSN 1877-0509. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050916306500>>. Citado nas páginas 30, 32, 76, 80 e 128.

BRITO, J. J.; MOSQUEIRO, T.; CIFERRI, R. R.; CIFERRI, C. D. Random access with a distributed bitmap join index for star joins. **Heliyon**, v. 6, n. 2, p. e03342, 2020. ISSN 2405-8440. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2405844020301870>>. Citado nas páginas 32, 76 e 129.

CARÉLO, C. C.; POLA, I. R.; CIFERRI, R. R.; TRAINA, A. J.; TRAINA-JR., C.; CIFERRI, C. D. A. The onion-tree: Quick indexing of complex data in the main memory. In: **Proceedings of the 13th East European Conference on Advances in Databases and Information Systems**. Berlin, Heidelberg: Springer-Verlag, 2009. (ADBIS '09), p. 235–252. ISBN 978-3-642-03972-0. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-03973-7\\_18](http://dx.doi.org/10.1007/978-3-642-03973-7_18)>. Citado na página 47.

CARÉLO, C. C. M.; POLA, I. R. V.; CIFERRI, R. R.; TRAINA, A. J. M.; TRAINA JR, C.; CIFERRI, C. D. d. A. Slicing the metric space to provide quick indexing of complex data in the main memory. **Inf. Syst.**, Elsevier Science Ltd., Oxford, UK, UK, v. 36, n. 1, p. 79–98, mar. 2011. ISSN 0306-4379. Disponível em: <<http://dx.doi.org/10.1016/j.is.2010.06.004>>. Citado na página 47.

CHAUDHURI, S.; DAYAL, U. An overview of data warehousing and OLAP technology. **SIGMOD Record**, v. 26, n. 1, p. 65–74, 1997. Disponível em: <<https://doi.org/10.1145/248603.248616>>. Citado nas páginas 29, 38 e 40.

CHAUDHURI, S.; DAYAL, U.; NARASAYYA, V. R. An overview of business intelligence technology. **Commun. ACM**, v. 54, n. 8, p. 88–98, 2011. Disponível em: <<https://doi.org/10.1145/1978542.1978562>>. Citado na página 30.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. **MONET**, v. 19, n. 2, p. 171–209, 2014. Disponível em: <<https://doi.org/10.1007/s11036-013-0489-0>>. Citado nas páginas 31 e 58.

CHÁVEZ, E.; NAVARRO, G.; BAEZA-YATES, R.; MARROQUÍN, J. L. Searching in metric spaces. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 33, n. 3, p. 273–321, set. 2001. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/502807.502808>>. Citado na página 49.

CIACCIA, P.; PATELLA, M. Searching in metric spaces with user-defined and approximate distances. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 27, n. 4, p. 398–437, dez. 2002. ISSN 0362-5915. Disponível em: <<http://doi.acm.org/10.1145/582410.582412>>. Citado na página 49.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. In: **Proceedings of the 23rd International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. (VLDB '97), p. 426–435. ISBN 1-55860-470-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=645923.671005>>. Citado na página 46.

CIFERRI, C. D. de A. **Distribuição dos Dados em Ambientes de Data Warehousing: O Sistema WebD2W e Algoritmos Voltados à Fragmentação Horizontal dos Dados**. Tese (Doutorado em Ciência da Computação) — Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Brasil, 2002. Citado nas páginas 29 e 31.

CIFERRI, C. D. de A.; CIFERRI, R. R.; GÓMEZ, L. I.; SCHNEIDER, M.; VAISMAN, A. A.; ZIMÁNYI, E. Cube algebra: A generic user-centric model and query language for OLAP cubes. **IJDWM**, v. 9, n. 2, p. 39–65, 2013. Disponível em: <<https://doi.org/10.4018/jdwm.2013040103>>. Citado nas páginas 15, 38 e 39.

DASH, S.; SHAKYAWAR, S.; SHARMA, M.; KAUSHIK, S. Big data in healthcare: management, analysis and future prospects. **Journal of Big Data**, v. 6, 12 2019. Citado nas páginas 32 e 78.

DATTA, R.; JOSHI, D.; LI, J.; WANG, J. Z. Image retrieval: Ideas, influences, and trends of the new age. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 40, n. 2, p. 5:1–5:60, maio 2008. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1348246.1348248>>. Citado na página 47.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. **Commun. ACM**, v. 51, n. 1, p. 107–113, 2008. Disponível em: <<http://doi.acm.org/10.1145/1327452.1327492>>. Citado nas páginas 31 e 60.

DERBEKO, P.; DOLEV, S.; GODES, E.; SHARMA, S. Security and privacy aspects in mapreduce on clouds: A survey. **Computer Science Review**, v. 20, p. 1–28, 2016. Disponível em: <<https://doi.org/10.1016/j.cosrev.2016.05.001>>. Citado na página 60.

FELIPE, J. C. **Desenvolvimento de métodos para extração, comparação e análise de características intrínsecas de imagens médicas, visando à recuperação perceptual por conteúdo**. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2005. Citado na página 48.

Filho, R. F. S.; Traina, A.; Traina, C.; Faloutsos, C. Similarity search without tears: the omnifamily of all-purpose access methods. In: **Proceedings 17th International Conference on Data Engineering**. [S.l.: s.n.], 2001. p. 623–630. ISSN 1063-6382. Citado na página 78.

GEORGE, L. **HBase - The Definitive Guide: Random Access to Your Planet-Size Data**. O'Reilly, 2011. ISBN 978-1-449-39610-7. Disponível em: <<http://www.oreilly.de/catalog/9781449396107/index.html>>. Citado na página 79.

GIANGRECO, I.; KABARY, I. A.; SCHULDT, H. Adam: A system for jointly providing ir and database queries in large-scale multimedia retrieval. In: **Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2014. (SIGIR '14), p. 1257–1258. ISBN 9781450322577. Disponível em: <<https://doi.org/10.1145/2600428.2611182>>. Citado nas páginas 32 e 77.

GOLFARELLI, M.; RIZZI, S. A survey on temporal data warehousing. **IJDWM**, v. 5, n. 1, p. 1–17, 2009. Disponível em: <<https://doi.org/10.4018/jdwm.2009010101>>. Citado na página 30.

\_\_\_\_\_. From star schemas to big data: 20+ years of data warehouse research. In: \_\_\_\_\_. **A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years**. [s.n.], 2018. p. 93–107. Disponível em: <[https://doi.org/10.1007/978-3-319-61893-7\\_6](https://doi.org/10.1007/978-3-319-61893-7_6)>. Citado nas páginas 37, 38, 41 e 76.

GONZALES, M. L.; BAGCHI, K. K.; UDO, G. J.; KIRS, P. Diffusion of business intelligence and data warehousing: An exploratory investigation of research and practice. In: **44th Hawaii International International Conference on Systems Science (HICSS-44 2011), Proceedings, 4-7 January 2011, Koloa, Kauai, HI, USA**. [S.l.: s.n.], 2011. p. 1–9. Citado na página 30.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Disponível em: <<http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>>. Citado na página 47.

HAN, H.; JUNG, H.; EOM, H.; YEOM, H. Y. Scatter-gather-merge: An efficient star-join query processing algorithm for data-parallel frameworks. **Cluster Computing**, v. 14, n. 2, p. 183–197, Jun 2011. ISSN 1573-7543. Disponível em: <<https://doi.org/10.1007/s10586-010-0144-5>>. Citado nas páginas 32 e 76.

Haralick, R. M. Statistical and structural approaches to texture. **Proceedings of the IEEE**, v. 67, n. 5, p. 786–804, May 1979. ISSN 0018-9219. Citado na página 47.

Haralick, R. M.; Shanmugam, K.; Dinstein, I. Textural features for image classification. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472. Citado na página 47.

HARINARAYAN, V.; RAJARAMAN, A.; ULLMAN, J. D. Implementing data cubes efficiently. In: **Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996**. [s.n.], 1996. p. 205–216. Disponível em: <<https://doi.org/10.1145/233269.233333>>. Citado nas páginas 38 e 39.

INMON, W. **Building the Data Warehouse**. Wiley, 2005. (Timely, practical, reliable). ISBN 9780764599446. Disponível em: <<https://books.google.com.br/books?id=rnG3vjy7iPoC>>. Citado na página 37.

ISTEPHAN, S.; SIADAT, M. Extensible query framework for unstructured medical data – a big data approach. In: **2015 IEEE International Conference on Data Mining Workshop (ICDMW)**. [S.l.: s.n.], 2015. p. 455–462. ISSN 2375-9259. Citado nas páginas 32 e 78.

ISTEPHAN, S.; SIADAT, M.-R. Unstructured medical image query using big data – an epilepsy case study. **Journal of Biomedical Informatics**, v. 59, p. 218 – 226, 2016. ISSN 1532-0464. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1532046415002853>>. Citado nas páginas 32, 78 e 80.

KIMBALL, R.; ROSS, M. **The data warehouse toolkit: the complete guide to dimensional modeling, 2nd Edition**. Wiley, 2002. ISBN 9780471200246. Disponível em: <<http://www.worldcat.org/oclc/49284159>>. Citado nas páginas 30, 31 e 41.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007. Citado nas páginas 67 e 68.



KUO, M.; CHRIMES, D.; MOA, B.; HU, W. Design and construction of a big data analytics framework for health applications. In: **2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)**. [S.l.: s.n.], 2015. p. 631–636. Citado nas páginas 32, 78 e 79.

LI, D.; ZHANG, W.; SHEN, S.; ZHANG, Y. Ses-lsh: Shuffle-efficient locality sensitive hashing for distributed similarity search. In: **2017 IEEE International Conference on Web Services (ICWS)**. [S.l.: s.n.], 2017. p. 822–827. Citado nas páginas 32, 61 e 77.

MAHASE, E. Covid-19: death rate is 0.66% and increases with age, study estimates. **BMJ**, v. 369, 2020. Citado nas páginas 94 e 120.

MCAFEE, A.; BRYNJOLFSSON, E. Big data: The management revolution. **Harvard business review**, v. 90, p. 60–6, 68, 128, 10 2012. Citado na página 57.

MOHANIA, M. K.; SAMTANI, S.; RODDICK, J. F.; KAMBAYASHI, Y. Advances and research directions in data-warehousing technology. **Australasian J. of Inf. Systems**, v. 7, n. 1, 1999. Citado na página 29.

NAKAGAWA, E. Y.; SCANNAVINO, K. R. F.; FABBRI, S. C. P. F.; FERRARI, F. C. **Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática**. [S.l.]: Elsevier Brasil, 2017. Citado nas páginas 16, 68 e 69.

NGUYEN, D.-T.; YONG, C. H.; PHAM, X.-Q.; NGUYEN, H.-Q.; LOAN, T. T. K.; HUH, E.-N. An index scheme for similarity search on cloud computing using mapreduce over docker container. In: **Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication**. New York, NY, USA: ACM, 2016. (IMCOM '16), p. 60:1–60:6. ISBN 978-1-4503-4142-4. Disponível em: <<http://doi.acm.org/10.1145/2857546.2857607>>. Citado nas páginas 32, 77 e 78.

NGUYEN, T. D. T.; HUH, E.-N. An efficient similar image search framework for large-scale data on cloud. In: **Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication**. New York, NY, USA: ACM, 2017. (IMCOM '17), p. 65:1–65:8. ISBN 978-1-4503-4888-1. Disponível em: <<http://doi.acm.org/10.1145/3022227.3022291>>. Citado nas páginas 32, 77 e 78.

PATGIRI, R.; NAYAK, S.; BORGHAIN, S. K.; XU, Z. Hunting the pertinency of bloom filter in computer networking and beyond: A survey. **J. Comput. Netw. Commun.**, Hindawi Limited, London, GBR, v. 2019, jan. 2019. ISSN 2090-7141. Disponível em: <<https://doi.org/10.1155/2019/2712417>>. Citado nas páginas 16 e 63.

POLA, I. R.; TRAINA, A. J.; TRAINA JR., C. Easing the dimensionality curse by stretching metric spaces. In: **Proceedings of the 21st International Conference on Scientific and Statistical Database Management**. Berlin, Heidelberg: Springer-Verlag, 2009. (SSDBM 2009), p. 417–434. ISBN 978-3-642-02278-4. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-02279-1\\_30](http://dx.doi.org/10.1007/978-3-642-02279-1_30)>. Citado nas páginas 46 e 47.

PONNIAH, P. **Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals**. Wiley, 2004. ISBN 9780471463894. Disponível em: <<https://books.google.com.br/books?id=n2nIM011TQ0C>>. Citado na página 42.

RAJA, P. V.; SIVASANKAR, E. Modern framework for distributed healthcare data analytics based on hadoop. In: LINAWATI; MAHENDRA, M. S.; NEUHOLD, E. J.; TJOA, A. M.; YOU, I. (Ed.). **Information and Communication Technology**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 348–355. ISBN 978-3-642-55032-4. Citado nas páginas 16, 32, 78, 79 e 80.

RICHARDSON, S.; et al. Presenting characteristics, comorbidities, and outcomes among 5700 patients hospitalized with COVID-19 in the New York City area. **JAMA**, 2020. ISSN 0098-7484. Citado nas páginas 94 e 119.

ROCHA, G. M.; CIFERRI, C. D. A. Processamento eficiente de consultas analíticas estendidas com predicado de similaridade em Spark. In: **Anais da Sessão de Short Papers do 34o Simpósio Brasileiro de Banco de Dados**. [S.l.: s.n.], 2019. p. 1–6. Citado nas páginas 34 e 127.

\_\_\_\_\_. Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um data warehouse de imagens em ambientes paralelos e distribuídos. In: **Anais do XVIII Workshop de Teses e Dissertações em Banco de Dados do 34o Simpósio Brasileiro de Banco de Dados**. [S.l.: s.n.], 2019. p. 1–7. Citado nas páginas 34 e 127.

ROCHA, G. M. da; CIFERRI, C. D. de A. Imgdw generator: a tool for generating data for medical image data warehouses. In: **XXXIII Simpósio Brasileiro de Banco de Dados: Demos e WTDBD, SBBD 2018 Companion, Rio de Janeiro, RJ, Brazil, August 25-26, 2018**. [S.l.: s.n.], 2018. p. 23–28. Citado nas páginas 15, 54, 88 e 116.

Sadashiv, N.; Kumar, S. M. D. Cluster, grid and cloud computing: A detailed comparison. In: **2011 6th International Conference on Computer Science Education (ICCSE)**. [S.l.: s.n.], 2011. p. 477–482. Citado na página 58.

SEBAA, A.; CHIKH, F.; NOUICER, A.; TARI, A. Medical big data warehouse: Architecture and system design, a case study: Improving healthcare resources distribution. **Journal of Medical Systems**, v. 42, n. 4, p. 59, Feb 2018. ISSN 1573-689X. Disponível em: <<https://doi.org/10.1007/s10916-018-0894-9>>. Citado nas páginas 31, 32, 78, 79 e 80.

SEBAA, A.; NOUICER, A.; CHIKH, F.; TARI, A. Big data technologies to improve medical data warehousing. In: **Proceedings of the 2Nd International Conference on Big Data, Cloud and Applications**. New York, NY, USA: ACM, 2017. (BDCA'17), p. 21:1–21:5. ISBN 978-1-4503-4852-2. Disponível em: <<http://doi.acm.org/10.1145/3090354.3090376>>. Citado nas páginas 32, 78 e 79.

SHI, J.; QIU, Y.; MINHAS, U. F.; JIAO, L.; WANG, C.; REINWALD, B.; ÖZCAN, F. Clash of the titans: Mapreduce vs. spark for large scale data analytics. **PVLDB**, v. 8, n. 13, p. 2110–2121, 2015. Disponível em: <<http://www.vldb.org/pvldb/vol8/p2110-shi.pdf>>. Citado nas páginas 32 e 61.

SHVACHKO, K.; KUANG, H.; RADIA, S.; CHANSLER, R. The hadoop distributed file system. In: **IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST 2012, Lake Tahoe, Nevada, USA, May 3-7, 2010**. [s.n.], 2010. p. 1–10. Disponível em: <<https://doi.org/10.1109/MSST.2010.5496972>>. Citado na página 59.

SKOPAL, T.; HOKSZA, D. Improving the performance of m-tree family by nearest-neighbor graphs. In: **Proceedings of the 11th East European Conference on Advances in Databases**

**and Information Systems**. Berlin, Heidelberg: Springer-Verlag, 2007. (ADBIS'07), p. 172–188. ISBN 3-540-75184-X, 978-3-540-75184-7. Disponível em: <http://dl.acm.org/citation.cfm?id=1780119.1780139>. Citado na página 46.

SKOPAL, T.; LOKOČ, J. Nm-tree: Flexible approximate similarity search in metric and non-metric spaces. In: . [S.l.: s.n.], 2008. v. 5181, p. 312–325. Citado na página 46.

TAO, Y.; ZHOU, M.; SHI, L.; WEI, L.; CAO, Y. Optimizing multi-join in cloud environment. In: **2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing**. [S.l.: s.n.], 2013. p. 956–963. Citado nas páginas 32, 76 e 80.

TARKOMA, S.; ROTHENBERG, C. E.; LAGERSPETZ, E. Theory and practice of bloom filters for distributed systems. **IEEE Communications Surveys and Tutorials**, v. 14, n. 1, p. 131–155, 2012. Disponível em: <https://doi.org/10.1109/SURV.2011.031611.00024>. Citado nas páginas 33 e 62.

TEIXEIRA, J. W. **Processamento de consultas analíticas com predicados de similaridade entre imagens em ambientes de data warehousing**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2015. Citado nas páginas 15, 42, 48 e 53.

TEIXEIRA, J. W.; ANNIBAL, L. P.; FELIPE, J. C.; CIFERRI, R. R.; CIFERRI, C. D. de A. A similarity-based data warehousing environment for medical images. **Computers in Biology and Medicine**, v. 66, p. 190 – 208, 2015. ISSN 0010-4825. Disponível em: <http://www.sciencedirect.com/science/article/pii/S001048251500308X>. Citado nas páginas 15, 30, 32, 52, 55, 70, 78, 128 e 129.

TRAINA, C.; FILHO, R. F. S.; TRAINA, A. J. M.; VIEIRA, M. R.; FALOUTSOS, C. The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. **The VLDB Journal**, v. 16, n. 4, p. 483–505, Oct 2007. ISSN 0949-877X. Disponível em: <https://doi.org/10.1007/s00778-005-0178-0>. Citado nas páginas 15, 31, 33, 46, 47, 49, 50, 51 e 78.

TRAINA, C.; MORIYAMA, A.; ROCHA, G. M.; CORDEIRO, R.; CIFERRI, C. D. A.; TRAINA, A. J. M. The similarql framework: similarity queries in plain sql. In: **Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC 2019) Poster Paper**. [S.l.: s.n.], 2019. p. 1–4. Citado nas páginas 34, 77 e 126.

TRAINA JR., C.; TRAINA, A.; FALOUTSOS, C.; SEEGER, B. Fast indexing and visualization of metric data sets using slim-trees. **IEEE Trans. on Knowl. and Data Eng.**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 14, n. 2, p. 244–260, mar. 2002. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/69.991715>. Citado nas páginas 46 e 47.

VAISMAN, A. A.; ZIMÁNYI, E. **Data Warehouse Systems - Design and Implementation**. Springer, 2014. (Data-Centric Systems and Applications). ISBN 978-3-642-54654-9. Disponível em: <https://doi.org/10.1007/978-3-642-54655-6>. Citado nas páginas 29, 37, 39, 40 e 41.

VIEIRA, M. R.; JR, C.; CHINO, F. J. T.; TRAINA, A. Dbm-tree: A dynamic metric access method sensitive to local density data. In: . [S.l.: s.n.], 2004. v. 1, p. 163–177. Citado nas páginas 46 e 47.

WILSON, D. R.; MARTINEZ, T. R. Improved heterogeneous distance functions. **J. Artif. Int. Res.**, AI Access Foundation, USA, v. 6, n. 1, p. 1–34, jan. 1997. ISSN 1076-9757. Disponível em: <<http://dl.acm.org/citation.cfm?id=1622767.1622768>>. Citado na página 48.

YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. In: **Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1993. (SODA '93), p. 311–321. ISBN 0-89871-313-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=313559.313789>>. Citado na página 46.

ZAHARIA, M.; CHOWDHURY, M.; FRANKLIN, M. J.; SHENKER, S.; STOICA, I. Spark: Cluster computing with working sets. In: **2nd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'10, Boston, MA, USA, June 22, 2010**. [S.l.: s.n.], 2010. Citado nas páginas 32 e 61.

ZHANG, C.; WU, L.; LI, J. Efficient processing distributed joins with bloomfilter using mapreduce †. In: . [S.l.: s.n.], 2013. Citado nas páginas 32 e 76.

ZHANG, H.; CHEN, G.; OOI, B. C.; TAN, K.; ZHANG, M. In-memory big data management and processing: A survey. **IEEE Trans. Knowl. Data Eng.**, v. 27, n. 7, p. 1920–1948, 2015. Disponível em: <<https://doi.org/10.1109/TKDE.2015.2427795>>. Citado na página 59.

ZHAO, J.; et al. Relationship between the ABO blood group and the COVID-19 susceptibility. **medRxiv**, 2020. Citado nas páginas 94 e 119.

ZHOU, X.; WANG, G.; YU, J. X.; YU, G. M+-tree: A new dynamical multidimensional index for metric spaces. In: **Proceedings of the 14th Australasian Database Conference - Volume 17**. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003. (ADC '03), p. 161–168. ISBN 0-909-92595-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=820085.820118>>. Citado na página 46.

