

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MÉTODO DE RECONSTRUÇÃO  
TOMOGRÁFICA DE AMOSTRAS AGRÍCOLAS  
COM O EMPREGO DE TÉCNICAS BIG DATA**

**GABRIEL MARCELINO ALVES**

**ORIENTADOR: PROF. DR. PAULO ESTEVÃO CRUVINEL**

São Carlos – SP

31 de janeiro de 2020

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

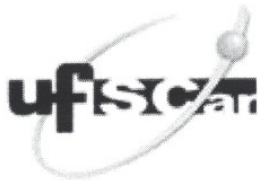
**MÉTODO DE RECONSTRUÇÃO  
TOMOGRÁFICA DE AMOSTRAS AGRÍCOLAS  
COM O EMPREGO DE TÉCNICAS BIG DATA**

**GABRIEL MARCELINO ALVES**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais e Arquiteturas de Computadores.  
Orientador: Prof. Dr. Paulo Estevão Cruvinel

São Carlos – SP

31 de janeiro de 2020



---

**Folha de Aprovação**

---

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Gabriel Marcelino Alves, realizada em 31/01/2020:

---

Prof. Dr. Paulo Estevão Cruvinel  
EMBRAPA

---

Prof. Dr. Nelson Delfino D'Avila Mascarenhas  
UFSCar

---

Prof. Dr. José Hiroki Saito  
UFSCar

---

Prof. Dr. Luciano da Fontoura Costa  
USP

---

Prof. Dr. Moacir Antonelli Ponti  
ICMC/USP

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Nelson Delfino D'Avila Mascarenhas e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

---

Prof. Dr. Paulo Estevão Cruvinel

*Dedico este trabalho a Nbia, Laura e Lucas.*



## AGRADECIMENTOS

---

---

À minha esposa Núbia e aos meus filhos Laura e Lucas, pela compreensão durante os períodos em que me ausentei para me concentrar no desenvolvimento deste trabalho. Agradeço também pelo carinho recebido, pelos momentos agradáveis que compartilhamos e pela oportunidade de crescermos juntos como família.

Aos meus pais, João e Celeste por sempre me apoiarem. Aos meus irmãos João Daniel e Alessandra pelo companheirismo. Aos meus familiares João Melo, Zélia, Magda, Ari, Márcia, Tibérius, Renato, Paula e Amanda.

Aos meus sobrinhos Arthur, Yasmin, Victor e Miguel pelas oportunidades que tivemos de brincar e de conversar, as quais sempre renovaram o ânimo do meu espírito.

Ao Dr. Paulo E. Cruvinel pela orientação neste trabalho, pelas conversas, pela disponibilidade em me ouvir, e pela compreensão nos meus momentos mais difíceis.

Ao Dr. Maurício F. Lima Pereira (UFMT) pela oportunidade de trabalharmos juntos, pelas sugestões feitas para este trabalho e pelas conversas e orientações.

Aos amigos que me acompanharam, de perto, nesta jornada: Everaldo, Kátia, Glauber, Ricardo, Natanael e Yvonne. Agradeço pelas inúmeras conversas e discussões acaloradas, pelas ajudas que recebi, pela preocupação comigo e com a minha família e, também, por respeitarem os meus momentos de silêncio. Foi muito gratificante encontrá-los e compartilhar da companhia de vocês.

A todos os amigos por compreenderem os momentos em que me distanciei para dedicar à conclusão deste projeto.

À UFSCar, sobretudo ao Programa de Pós-Graduação em Ciência da Computação, pela oportunidade de me desenvolver como profissional. Pelos conhecimentos recebidos dos professores do departamento e pela atenção do setor administrativo nas mais diversas questões que precisei tratar.

Aos colegas da UFSCar com quem tive a oportunidade de me interagir, sobretudo, aos colegas André, Alex e Gustavo pelos momentos em que trabalhamos juntos e pelas trocas de experiências.

À Embrapa Instrumentação pela oportunidade de desenvolver este trabalho. Ao analista

Paulo Orlando Lasso, pela atenção dispensada durante os trabalhos desenvolvidos no Laboratório de Técnicas Nucleares e preparação das amostras de sementes utilizadas neste trabalho.

Ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) pela oportunidade concedida para que eu pudesse realizar este trabalho. Aos colegas e alunos do campus São João da Boa Vista.

À CAPES pela bolsa concedida dentro do Programa Prodoutoral.

À Deus.

*Minha filha mal aprendera a engatinhar e já demonstrava ávido interesse pelo controle remoto. Ela observava que as teclas interferiam no comportamento da televisão, mas algo ainda lhe faltava. Ela tinha os dados, mas não o conhecimento. Sorri, ao perceber que buscávamos a mesma coisa.*

(Gabriel Marcelino Alves)

# RESUMO

É apresentado um novo método de reconstrução tomográfica de amostras agrícolas em alta resolução, o qual utiliza a densidade espectral das projeções tomográficas de Raios-X como critério para minimizar o tempo de processamento e a obtenção de imagens digitais de boa qualidade, além de ser escalável. O uso da densidade espectral das projeções tomográficas viabilizou avaliar a energia associada em cada projeção e conseqüentemente a quantidade de informação que está relacionada às suas probabilidades. Desta forma, as projeções tomográficas foram organizadas em classes de energia e aquelas portadoras de quantidades de informações mais expressivas foram selecionadas. Como parte do método, após a seleção de projeções, foi considerado a retroprojeção filtrada (FBP) e a interpolação B-spline para a obtenção das reconstruções 2D e 3D (volumétrica), etapas que foram paralelizadas considerando o ambiente Apache Spark. Para a execução do método desenvolvido foi organizado um ambiente Big Data que contou com um *cluster*, instalado na plataforma Amazon Web Services (AWS), e uma pilha de tecnologias. A avaliação da configuração do ambiente Big Data considerou quatro conjuntos de matrizes de projeções de um mesmo *phantom* heterogêneo de plexiglass totalizando 7840 matrizes (35, 63GB) os quais foram processados por 12 diferentes configurações totalizando 427, 56 GB de dados tomográficos processados. A configuração do *cluster* foi definida após a avaliação das métricas de Speedup e Eficiência para o método em execução no ambiente Big Data. Adicionalmente, um conjunto composto por um *phantom* heterogêneo de plexiglass, um *phantom* Sheep-Logan e outro homogêneo além de 33 amostras de sementes agrícolas foi preparado para fins de validar e avaliar a qualidade da reconstrução dos conjuntos de projeções tomográficas selecionadas. Neste contexto, foi organizado um banco de imagem contendo 66.642 imagens 2D de sementes agrícolas (242GB). As métricas SSIM (*Structural Similarity Index*), NRMSE (*Normalized Root Mean Square Error*) e PSNR (*Peak Signal-to-Noise Ratio*) foram utilizadas nas etapas de validação. A métrica SSIM foi calculada para cada matriz de projeções e observou-se a medida da mediana para os valores SSIM de cada amostra. Neste sentido, a análise SSIM mostrou que a reconstrução tomográfica das amostras, em duas dimensões, a partir das projeções selecionadas, levou à obtenção do valor SSIM superior a 0, 80, para todas as amostras analisadas. Os resultados mostraram que o método possibilitou a redução entre 28% e 38% no número de projeções tomográficas em cada amostra analisada, sem comprometer a qualidade das imagens reconstruídas. Finalmente, este novo método se mostrou útil para viabilizar a análise de grandes quantidades de amostras agrícolas tomando por base o uso da tomografia de Raios-X, a fim de atender os manejos baseados nos paradigmas da agricultura de precisão, onde o número crescente de análises requeridas às amostras agrícolas no processo de tomada de decisão é considerado um fator primordial.

**Palavras-chave:** reconstrução de imagens tomográficas, seleção de projeções tomográficas, big data, processamento de imagens, agricultura de precisão.

# ABSTRACT

A new method of high resolution tomographic reconstruction of agricultural samples is presented, which uses the spectral density of X-ray tomographic projections as a criterion to minimize processing time and obtain good quality digital images, besides being scalable. The use of the spectral density of the tomographic projections made it possible to evaluate the associated energy in each projection and consequently the amount of information that is related to its probabilities. Thus, the tomographic projections were organized into energy classes and those with the most expressive amounts of information were selected. As part of the method, after selecting projections, Filtered Back Projection (FBP) and B-Spline interpolation were considered to obtain 2D and 3D (volumetric) reconstruction, steps that were parallelized considering the Apache Spark environment. For the execution of the developed method was organized a Big Data environment that had a cluster, installed on the Amazon Web Services (AWS) platform and a stack of technologies. The Big Data environment configuration assessment considered four sets of projection matrices of the same plexiglass heterogeneous phantom totaling 7840 matrices (35.63 GB) which were processed for 12 different configurations totaling 427.56 GB of processed tomographic data. The cluster configuration was defined after evaluating the Speedup and Efficiency metrics for the method running in the Big Data environment. In addition, a cluster consisting of a heterogeneous plexiglass phantom, a Sheep-Logan phantom and a homogeneous sample plus 33 seed samples was prepared for the purpose of validating and evaluating the quality of cluster reconstruction of selected tomographic projections. In this context, an image dataset containing 66,642 2D images of seeds (242 GB) has been organized. The Structural Similarity Index (SSIM), Normalized Root Mean Square Error (NRMSE), and Peak Signal-to-Noise Ratio (PSNR) metrics were used in the validation steps. The SSIM metric was calculated for each projection matrix and the median measurement for the SSIM values of each sample was observed. In this sense, the SSIM analysis showed that the tomographic reconstruction of the two-dimensional samples from the selected projections led to the SSIM value exceeding 0.80 for all samples analyzed. The results showed that the method allowed a reduction between 28% and 38% in the number of tomographic projections in each sample analyzed, without compromising the quality of the reconstructed images. Finally, this new method has been shown to be useful for the analysis of large quantities of agricultural samples based on the use of X-ray tomography in order to meet the management based on precision agriculture paradigms, where the increasing number of analyzes required for agricultural samples in the decision-making process is considered a prime factor.

**Keywords:** tomographic image reconstruction, tomographic projections selection, big data, image processing, precision agriculture.

## LISTA DE FIGURAS

---

---

Figura 1 – Visão geral do método desenvolvido para reconstrução de imagens tomográficas em ambiente Big Data. . . . .	29
Figura 2 – Modelo físico da atenuação de raio-X. . . . .	41
Figura 3 – Representação da tomografia em feixes paralelos. . . . .	44
Figura 4 – Sistemas de coordenadas utilizados na tomografia. . . . .	44
Figura 5 – Diagrama esquemático de uma projeção paralela. . . . .	45
Figura 6 – Teorema das Seções de Fourier . . . . .	46
Figura 7 – Sobreposição das fatias bidimensionais para reconstrução tridimensional. Adaptado de Pereira e Cruvinel (2015). . . . .	48
Figura 8 – Exemplo interpolação utilizando <i>B-splines</i> . . . . .	50
Figura 9 – Proporção de domicílios com computador, por área (2008-2018). O percentual é sobre o total de domicílios. Fonte: adaptado de CGI.br (2019). . . . .	52
Figura 10 – Tendência de procura sobre o termo "Big Data" no período de 2004 a 2017 no mundo. Fonte: Google Trends. <a href="http://www.google.com/trends">http://www.google.com/trends</a> . . . . .	54
Figura 11 – Características do Big Data: volume, velocidade e variedade. Adaptado de Sagioglu e Sinanc (2013), Zhang e Huang (2013). . . . .	57
Figura 12 – Processo de descoberta do conhecimento. Adaptado de Chen e Zhang (2014). . . . .	61
Figura 13 – Técnicas de Ciência dos Dados. Adaptado de Chen e Zhang (2014). . . . .	62
Figura 14 – Metodologia PCAM para elaboração de algoritmos paralelos. . . . .	66
Figura 15 – Diagrama esquemático do modelo de programação <i>MapReduce</i> . . . . .	68
Figura 16 – Arquitetura do sistema HDFS. Adaptado de Gunarathne (2015). . . . .	70
Figura 17 – Diagrama esquemático do fluxo de execução de trabalho gerenciado pelo Hadoop YARN. Adaptado de Gunarathne (2015) e Hadoop (2018). . . . .	71
Figura 18 – Ecossistema Hadoop. . . . .	72
Figura 19 – Visão geral da arquitetura Spark. Fonte: <a href="https://spark.apache.org/docs/latest/cluster-overview.html">https://spark.apache.org/docs/latest/cluster-overview.html</a> . Último acesso em 16/02/2019. . . . .	74
Figura 20 – Bibliotecas adicionais do Spark. . . . .	74
Figura 21 – Amostras simuladas utilizadas para avaliação. (a) <i>phantom</i> heterogêneo Shepp-Logan. (b) <i>phantom</i> homogêneo. . . . .	78

Figura 22 – Preparação do <i>phantom</i> de plexiglass a ser utilizado para validação. (a) Diagrama de concepção de um <i>phantom</i> . (b) Imagem tomográfica do <i>phantom</i> . Adaptado de Beraldo e colaboradores (2014). . . . .	78
Figura 23 – Imagens do <i>phantom</i> inserido no tomógrafo de alta resolução SkyScan 1172.	79
Figura 24 – Tipos de sementes utilizadas neste trabalho e suas respectivas plantas. (a) Amendoim. (b) Feijão Fradinho. (c) Girassol. (d) Grão de Bico. (e) Trigo. (f) Abóbora. (g) Soja. . . . .	79
Figura 25 – Diagrama de blocos do método para reconstrução de imagens tomográficas provenientes de amostras agrícolas em ambiente Big Data. . . . .	82
Figura 26 – Fases da etapa de seleção de projeções. . . . .	83
Figura 27 – Pilha de tecnologias empregadas para a organização do ambiente Big Data. .	85
Figura 28 – Relação dos parâmetros de configuração de memória no ambiente Big Data, em configuração Apache Spark. . . . .	90
Figura 29 – Representação conceitual das classes de energia considerando a distribuição Gaussiana em cada classe. A região hachurada corresponde a região de cada classe em que se encontram as projeções mais significativas. . . . .	97
Figura 30 – Fluxograma do processo de seleção de projeções utilizando a densidade espectral. . . . .	98
Figura 31 – Método de reconstrução tomográfica sob a perspectiva do modelo de programação MapReduce. . . . .	99
Figura 32 – Diagrama de blocos da reconstrução 2D paralela. As linhas pontilhadas indicam as etapas do método sob a perspectiva do modelo de programação MapReduce. . . . .	101
Figura 33 – Diagrama de blocos da reconstrução volumétrica paralela, considerando um exemplo do processamento de um <i>phantom</i> . As linhas pontilhadas indicam as etapas do método sob a perspectiva do modelo de programação MapReduce.	103
Figura 34 – Identificação de uma região ( <i>tile</i> ) em uma fatia. No detalhe, a região e o respectivo identificador composto por oito caracteres. . . . .	103
Figura 35 – Tempos de reconstrução 2D. Na legenda foi atribuída a sigla $t_2$ para Nós $m5.xlarge$ , $t_3$ para Nós $m5.2xlarge$ e $t_4$ para Nós $m5.4xlarge$ . .	107
Figura 36 – Tempos de reconstrução 3D (volumétrica). Na legenda foi atribuída a sigla $t_2$ para Nós $m5.xlarge$ , $t_3$ para Nós $m5.2xlarge$ e $t_4$ para Nós $m5.4xlarge$ . . . . .	108
Figura 37 – Speedup para reconstrução 2D. . . . .	109
Figura 38 – Speedup para reconstrução 3D (volumétrica). . . . .	110
Figura 39 – Eficiência para reconstrução 2D. . . . .	111
Figura 40 – Eficiência para reconstrução 3D (volumétrica). . . . .	111
Figura 41 – Seleção de projeções tomográficas aplicada em um sinograma do <i>phantom</i> . .	114
Figura 42 – Análise do número de projeções tomográficas selecionadas por amostra. . .	115

Figura 43 – Fatias do <i>phantom</i> que representam os valores mínimo, mediana e máximo da medida SSIM. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	117
Figura 44 – Análise SSIM das imagens reconstruídas do conjunto de 33 amostras. . . . .	127
Figura 45 – Fatias do <i>phantom</i> que representam os valores mínimo, mediana e máximo da medida NRMSE. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	128
Figura 46 – Análise NRMSE das imagens reconstruídas do conjunto de 33 amostras de sementes agrícolas. . . . .	128
Figura 47 – Fatias do <i>phantom</i> que representam os valores mínimo, mediana e máximo da medida NRMSE. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	129
Figura 48 – Análise PSNR das imagens reconstruídas do conjunto de 33 amostras de sementes agrícolas. . . . .	130
Figura 49 – Diagrama de blocos da reconstrução 2D paralela. . . . .	131
Figura 50 – Análise de memória RAM livre durante reconstrução 2D paralela. . . . .	133
Figura 51 – Análise do tráfego de dados durante reconstrução 2D paralela. A linha cheia representa a quantidade de bytes recebidos e a linha tracejada indica a quantidade de bytes transmitidos. . . . .	134
Figura 52 – Ambiente Jupyter notebook integrado à ferramenta <i>itkwidgets</i> . . . . .	135
Figura 53 – Visualização volumétrica do <i>phantom</i> . . . . .	136
Figura 54 – Corte realizado na visualização volumétrica do <i>phantom</i> . . . . .	136
Figura 55 – Visualização volumétrica de uma amostra de Feijão Fradinho. . . . .	137
Figura 56 – Visualização volumétrica, com corte, de uma amostra de Feijão Fradinho. . . . .	137
Figura 57 – À esquerda é ilustrada uma visão geral na qual o Sistema de Suporte à Decisão basea-do em Big Data e Ciência dos Dados traduz as demandas agrícolas em propostas de soluções que dão suporte à tomada de decisão. À direita tal sistema é detalhado em quatro principais camadas. . . . .	157



## LISTA DE TABELAS

---

---

Tabela 1 – Valores dos parâmetros do tomógrafo SkyScan 1172 ajustados para aquisição das projeções do <i>phantom</i> e das amostras de sementes. . . . .	80
Tabela 2 – Capacidade dos <i>clusters</i> em função do número de Nós e da configuração individual de cada modelo de Nó. . . . .	106
Tabela 3 – Identificação das configurações de <i>clusters</i> . . . . .	106
Tabela 4 – Volume de dados processados para análise da infraestrutura Big Data . . . .	106
Tabela 5 – Tempos sequenciais obtidos para reconstrução 2D e 3D (volumétrica). . . .	109
Tabela 6 – Custo de processamento por hora, em dólares (USD). . . . .	112
Tabela 7 – Relação custo de processamento sequencial por custo de processamento paralelo nas reconstruções 2D e 3D (volumétrica). . . . .	112
Tabela 8 – Avaliação da variação do intervalo para seleção das projeções em um sinograma.	116
Tabela 9 – Comparação de intervalos para seleção das projeções em um sinograma em relação ao intervalo $\sigma = 1,00$ . . . . .	116
Tabela 10 – Valores de SSIM mínimo, mediana e máximo calculados para o <i>phantom</i> . . .	117
Tabela 11 – Resultados para a avaliação de imagens tomográficas de amostras de Feijão Fradinho. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas. . . . .	117
Tabela 12 – Resultados para a avaliação de imagens tomográficas de amostras de Girassol. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas. . .	118
Tabela 13 – Resultados para a avaliação de imagens tomográficas de amostras de Grão de Bico. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.	118
Tabela 14 – Resultados para a avaliação de imagens tomográficas de amostras de Trigo. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas. . .	119
Tabela 15 – Resultados para a avaliação de imagens tomográficas de amostras de Abóbora. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas. . .	119
Tabela 16 – Resultados para a avaliação de imagens tomográficas de amostras de Soja. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas. . .	119
Tabela 17 – Resultados para avaliação de imagens tomográficas de amostras de Amendoim. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.	120

Tabela 18 – Imagens tomográficas de amostras de Feijão Fradinho. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 11, página 117. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	120
Tabela 19 – Imagens tomográficas de amostras de sementes de Girassol. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 12, página 118. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	121
Tabela 20 – Imagens tomográficas de amostras de sementes de Grão de Bico. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 13, página 118. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	122
Tabela 21 – Imagens tomográficas de amostras de sementes de Trigo. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 14, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	123
Tabela 22 – Imagens tomográficas de amostras de sementes de Abóbora. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 15, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	124
Tabela 23 – Imagens tomográficas de amostras de sementes de Soja. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 16, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	125
Tabela 24 – Imagens tomográficas de amostras de Amendoim. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 17, página 120. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255. . . . .	126
Tabela 25 – Valores de NRMSE mínimo, mediana e máximo calculados para o <i>phantom</i> .	127
Tabela 26 – Valores de PSNR mínimo, mediana e máximo calculados para o <i>phantom</i> .	129

## LISTA DE QUADROS

---

---

Quadro 1 – Seleção de conceitos do termo <i>Big Data</i> . . . . .	55
Quadro 2 – Tecnologias de Big Data e Data Science. . . . .	63
Quadro 3 – Versões das tecnologias empregadas na organização do ambiente Big Data. . . . .	85
Quadro 4 – Tipos de instâncias utilizados na composição do <i>cluster</i> . . . . .	86
Quadro 5 – Parâmetros para a organização e configuração do <i>cluster</i> . . . . .	87
Quadro 6 – Parâmetros e valores adotados para configuração da memória do Apache Spark. . . . .	91
Quadro 7 – Parâmetros de configuração da aplicação. . . . .	93
Quadro 8 – Parâmetros e valores adotados para configuração da memória do Apache Spark para análise de viabilidade da reconstrução 2D paralela. . . . .	132

# LISTA DE CÓDIGOS

---

---

Código 1 – Criação do cluster pelo MRJob utilizando parâmetro <code>create-cluster</code> .	89
Código 2 – Criação do cluster pelo MRJob ao executar a aplicação. . . . .	89
Código 3 – Trecho do código responsável por paralelizar a reconstrução tomográfica 2D.	131
Código 4 – Arquivo de configuração MRJob . . . . .	154
Código 5 – Arquivo de configuração do método desenvolvido . . . . .	156

## LISTA DE ALGORITMOS

---

---

Algoritmo 1 – <i>Filtered BackProjection</i> (FBP) . . . . .	101
Algoritmo 2 – Reconstrução volumétrica . . . . .	102
Algoritmo 3 – Seleção de projeções baseada em densidade espectral de potência. . . . .	113
Algoritmo 4 – Calcula a energia das projeções contidas em uma matriz (sinograma). . . . .	113

## LISTA DE SIGLAS

---

---

ASiR	<i>Adaptive Statistical iterative Reconstruction</i>
ART	<i>Algebraic Reconstruction Technique</i>
AWS	<i>Amazon Web Services</i>
BM3D	<i>Block-Matching 3D</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Compressive Sensing</i>
CT	<i>Computed Tomography</i>
DSP	<i>Digital Signal Processing</i>
DWT	<i>Discrete Wavelets Transform</i>
EC2	<i>Elastic Compute Cloud</i>
EMR	<i>Elastic MapReduce</i>
FBP	<i>Filtered Back Projection</i>
FTP	<i>File Transport Protocol</i>
GFS	<i>Google File System</i>
GiB	<i>Gibibyte</i>
GPS	<i>Global Positioning System</i>
GPU	<i>Graphical Processing Unit</i>
GS	<i>Generalized Sampling</i>
HDFS	<i>Hadoop Distributed File System</i>
HTTP	<i>HyperText Transfer Protocol</i>
JDK	<i>Java Development Kit</i>

JSON	<i>JavaScript Object Notation</i>
KVM	<i>Keyboard, Video, Monitor</i>
LDA	<i>Linear Detector Arrays</i>
LTS	<i>Long Term Support</i>
MLEM	<i>Maximum Likelihood Expectation Maximum</i>
MBIR	<i>Model-Based Iterative Reconstruction</i>
MPI	<i>Message Passing Interface</i>
MSE	<i>Mean Square Error</i>
NLM	<i>Non-Local Means</i>
NIST	<i>National Institute of Standards and Technology</i>
NoSQL	<i>Not only SQL</i>
NRMSE	<i>Normalized Root Mean Squared Error</i>
ONU	Organização das Nações Unidas
OPED	<i>Orthogonal Polynomial Expansion on the Disk</i>
PCAM	Particionamento, Comunicação, Aglomeração e Monitoramento
PET	<i>Positron Emission Tomography</i>
POCS	<i>Projections Onto Convex Sets</i>
PSD	<i>Power Spectrum Density</i>
PSF	<i>Point Spread Function</i>
RBM	<i>Restricted Boltzman Machine</i>
RDD	<i>Resilient Distributed Datasets</i>
S3	<i>Simple Storage Service</i>
SART	<i>Simultaneous Algebraic Reconstruction Technique</i>
SIG	Sistema de Informação Geográfica
SIRT	<i>Simultaneous Iterative Reconstructive Technique</i>
SPECT	<i>Single Photon Emitted Computed Tomography</i>

SQL	<i>Structured Query Language</i>
SSH	<i>Secure Shell</i>
TC	Tomografia Computadorizada
vCPU	<i>Virtual CPU</i>
TIC	Tecnologias de Informação e Comunicação
VTK	<i>Visualization ToolKit</i>
YAML	<i>YAML Ain't Markup Language</i>
YARN	<i>Yet Another Resource Negotiator</i>



## LISTA DE SÍMBOLOS

---

---

- $\theta$  Ângulo de aquisição de uma projeção tomográfica, medido em graus.
- $p_\theta$  Projeção tomográfica obtida em um determinado ângulo  $\theta$ .
- $N$  Número de pontos de uma projeção tomográfica, ou número de colunas em uma matriz.
- $p_{i,\theta}$   $i$ -ésimo ponto de  $p_\theta$ , para  $i = 0, 1, \dots, N - 1$ .
- $\varsigma$  Sinograma, o mesmo que matriz de projeções tomográficas.
- $M$  Número de projeções em um sinograma  $\varsigma$ , ou número de linhas em uma matriz.
- $j$  Número imaginário,  $j = \sqrt{-1}$ .
- $Q_g$  *Spline* de grau  $g$ .
- $\Gamma$  Densidade espectral de potência.
- $\hat{\Gamma}$  Estimador por periodograma da  $\Gamma$ .
- $\xi_m$  Energia de  $p_\theta$  obtido por meio de  $\hat{\Gamma}$ , para  $m = 0, 1, \dots, M - 1$ .
- $\Xi$  Conjunto de energias  $\xi_m$  em  $\varsigma$ , tal que  $\Xi = \{\xi_0, \xi_1, \dots, \xi_{M-1}\}$ .
- $\kappa$  Número de classes de energias em  $\varsigma$ .
- $C_i$  Classe de energias em  $\varsigma$ , para  $i = 0, 1, \dots, \kappa - 1$ .
- $C_T$  Conjunto de classes de energias, tal que  $C_T = \{C_0, C_1, \dots, C_{\kappa-1}\}$ .
- $\xi_{s_i}$  Energia inicial da classe  $C_i$ .
- $\xi_{t_i}$  Energia final da classe  $C_i$ .
- $C_i^{sel}$  Conjunto de projeções selecionadas na classe  $C_i$ .
- $S_p$  Speedup para  $p$  processadores.
- $E_p$  Eficiência para  $p$  processadores.

# SUMÁRIO

---

---

<b>CAPÍTULO 1–INTRODUÇÃO</b> . . . . .	<b>23</b>
1.1 Contextualização . . . . .	23
1.2 Motivação . . . . .	27
1.3 Objetivo . . . . .	28
1.4 Visão geral do método . . . . .	28
1.5 Trabalhos correlatos . . . . .	29
1.6 Organização do documento . . . . .	36
<b>CAPÍTULO 2–TOMOGRAFIA COMPUTADORIZADA</b> . . . . .	<b>38</b>
2.1 Histórico . . . . .	38
2.2 O uso da tomografia computadorizada na agricultura . . . . .	39
2.3 Reconstrução tomográfica . . . . .	41
2.3.1 Transformada de Radon . . . . .	43
2.3.2 Teorema das seções de Fourier . . . . .	46
2.3.3 Reconstrução 3D (volumétrica) . . . . .	48
<b>CAPÍTULO 3–BIG DATA E CIÊNCIA DOS DADOS</b> . . . . .	<b>51</b>
3.1 Contextualização . . . . .	51
3.2 O conceito de Big Data . . . . .	53
3.3 Características e desafios de Big Data . . . . .	57
3.4 A ciência dos dados . . . . .	60
3.5 Técnicas e tecnologias . . . . .	62
3.6 Sistemas Big Data . . . . .	64
3.7 Programação Paralela . . . . .	65
3.7.1 Metodologia PCAM . . . . .	66
3.7.2 Modelo de programação <i>MapReduce</i> . . . . .	67
3.8 Plataforma Hadoop . . . . .	69
3.8.1 <i>Hadoop Distributed File System</i> (HDFS) . . . . .	69
3.8.2 Hadoop MapReduce . . . . .	70
3.8.3 Hadoop YARN . . . . .	71
3.8.4 Ecossistema Hadoop . . . . .	72
3.9 Apache Spark . . . . .	73
3.10 Computação em Nuvem . . . . .	75
3.10.1 Amazon Web Services (AWS) . . . . .	76

<b>CAPÍTULO 4–MÉTODO DE RECONSTRUÇÃO TOMOGRÁFICA 2D E 3D (VOLUMÉTRICA) DE AMOSTRAS AGRÍCOLAS EM AMBIENTE BIG DATA</b>	<b>77</b>
4.1 Base de dados e métricas de avaliação	77
4.1.1 Base de dados de projeções tomográficas	77
4.1.2 Métricas de avaliação	80
4.2 Visão sistêmica do método	82
4.3 Organização do ambiente Big Data	84
4.3.1 Configuração do <i>cluster</i>	86
4.3.2 Configuração do Apache Spark	89
4.4 Armazenamento das matrizes de projeções tomográficas	92
4.5 Seleção de projeções baseada em densidade espectral	93
4.5.1 Densidade espectral de potência	94
4.5.2 Modelo de seleção das projeções tomográficas	95
4.5.3 Algoritmo para seleção de projeções	98
4.6 Método de reconstrução tomográfica utilizando abordagem MapReduce	99
4.6.1 Reconstrução tomográfica bidimensional paralela	100
4.6.2 Reconstrução 3D (volumétrica) paralela	102
<b>CAPÍTULO 5–RESULTADOS E DISCUSSÕES</b>	<b>105</b>
5.1 Análise da infraestrutura para o método de reconstrução tomográfica com Big Data	105
5.1.1 Speedup	108
5.1.2 Eficiência	110
5.2 Análise do método de reconstrução tomográfica baseado na densidade espectral das projeções	112
5.2.1 Análise SSIM	116
5.2.2 Análise NRMSE	127
5.2.3 Análise PSNR	129
5.3 Reconstrução 2D paralela	130
5.3.1 Análise de viabilidade	132
5.4 Visualização 3D (volumétrica) de imagens tomográficas	135
<b>CAPÍTULO 6–CONCLUSÃO</b>	<b>138</b>
6.1 Principais contribuições	140
<b>REFERÊNCIAS</b>	<b>142</b>

<b>Apêndices</b>	<b>153</b>
<b>APÊNDICE A–ARQUIVOS DE CONFIGURAÇÃO . . . . .</b>	<b>154</b>
<b>APÊNDICE B–SISTEMA DE SUPORTE À DECISÃO BASEADO EM BIG DATA E CIÊNCIA DOS DADOS . . . . .</b>	<b>157</b>
<b>APÊNDICE C–PUBLICAÇÕES . . . . .</b>	<b>158</b>

# Capítulo 1

## INTRODUÇÃO

---

---

Este capítulo contextualiza a motivação e a definição do problema, destaca o objetivo e as justificativas do trabalho, além de apresentar a visão geral do método desenvolvido para reconstrução de imagens tomográficas com o emprego de técnicas Big Data. Ao final são apresentados os trabalhos correlatos e a organização deste documento.

### 1.1 Contextualização

A Organização das Nações Unidas (ONU) lançou, em 2015, plano de ação denominado Agenda 2030 com 17 objetivos e 169 metas de desenvolvimento sustentável. O segundo objetivo daquele plano é o de “*acabar com a fome, alcançar a segurança alimentar e melhoria da nutrição e promover a agricultura sustentável*”, que é composto por cinco metas entre as quais destaca-se a que trata de aumentar o investimento em infraestrutura rural, pesquisa e extensão de serviços agrícolas bem como desenvolvimento de tecnologias<sup>1</sup>. O fato descreve a importância da agricultura no cenário mundial e a preocupação em desenvolver novas tecnologias que contribuam para alcançar a segurança alimentar, a agricultura sustentável entre outros objetivos.

O uso das atuais e das novas tecnologias no campo, portanto, desempenham papel importante no atual cenário, tanto nacional quanto global, pois elas auxiliam o agricultor a identificar a variabilidade, a analisá-la e atuar de modo a aumentar as chances de sucesso do empreendimento agrícola. Nesse particular, encontra-se a Agricultura de Precisão que é definida como um conjunto de ferramentas e tecnologias aplicadas para permitir um sistema de gerenciamento agrícola baseado na variabilidade espacial e temporal da unidade produtiva e visa ao aumento de retorno econômico e à redução do impacto ao ambiente (GRIFFIN; LOWENBERG-DEBOER, 2005).

Pois bem, trabalhar com sistema de gerenciamento agrícola baseado na variabilidade exige lidar com diferentes tipos de dados e de informações para que seja possível elaborar mapas que auxiliem a gestão e o melhor entendimento da lavoura, como por exemplo mapa

<sup>1</sup> Fonte: <https://nacoesunidas.org/pos2015/agenda2030/>. Último acesso em 01/08/2017.

da distribuição de nutrientes, mapa da quantidade de água disponível entre outros. No caso do processo de amostragem de solo, a construção de mapas das áreas agrícolas ainda necessita de melhorias quanto a sua abordagem pois, em geral, a quantidade de amostras não é suficiente quando há casos em que se tem uma amostra de solo a cada cinco ou, pior, a cada vinte hectares (INAMASU; BERNADI, 2014). Aumentar a taxa de amostragem de solo por talhão, considerando o uso de coletores de solo, significa refinar o processo e aumentar a resolução melhorando a qualidade do mapa, que está associada ao melhor modelo de interpolação utilizado para sua geração a partir de dados medidos e especializados (VIEIRA, 2000). Significa, também, aumentar consideravelmente a quantidade de dados para se realizar uma medição direta do solo. Em geral, o que se faz para lidar com a falta de dados é utilizar técnicas que viabilizam indicações indiretas como medidas do solo (PENALOZA; CRUVINEL; OLIVEIRA; COSTA, 2017).

O exemplo relatado acima indica que aumentar a quantidade de dados pode implicar na melhoria das informações produzidas e, conseqüentemente, na melhoria do sistema de gerenciamento agrícola. Nesse sentido, o uso de sensores, de eletrônica embarcada, da robótica, de sinais de satélites *Global Positioning System* (GPS), do tratamento de imagens, da tomografia computadorizada entre outras tecnologias, possibilitaram a expansão da Agricultura de Precisão e ampliaram a quantidade e a variedade de dados. Por outro lado, há situações em que a necessidade por maior capacidade de processamento e desempenho pode superar a necessidade por maiores quantidades de dados, como é o caso da tomografia computadorizada.

A Tomografia Computadorizada (TC), como será visto em mais detalhes na seção 2.1, teve suas origens nos trabalhos pioneiros do físico Allan MacLeod Cormack e do engenheiro elétrico Godfrey Hounsfield desenvolvidos durante as décadas de 1960 e 1970. Com o passar do tempo o uso da tomografia aumentou significativamente com aplicações em diversas áreas, dentre elas a agrícola a qual empregou a tomografia como método de análise e investigação de propriedades físicas de sementes, do solo, de plantas, e de árvores (VAZ; CRESTANA; NAIME; CRUVINEL, 2014).

A evolução das técnicas tomográficas desenvolveu-se em duas principais linhas. A primeira, relacionada à instrumentação dos equipamentos tomográficos, desenvolvimento e aprimoramento dos aparelhos. A segunda, referente à elaboração e refinamento de algoritmos voltados à reconstrução e à visualização de imagens. Diferentes algoritmos (analíticos e iterativos) têm sido aplicados no processo de reconstrução bem como o emprego de variadas técnicas de filtragem, a exemplo das técnicas lineares, estatísticas e outras fundamentadas, por exemplo, no uso de Wavelets. Portanto, as reconstruções bidimensional (2D) e tridimensional (3D) configuraram-se como instrumentos relevantes, uma vez que possibilitaram a análise do interior de um corpo ou objeto de forma não invasiva e assumiram papel importante para uso em ambientes estáticos e também dinâmicos (PEREIRA, 2007).

Neste sentido, o tempo de aquisição dos dados é um parâmetro importante a ser considerado em aplicações que envolvam análises tomográficas, pois a duração influencia no processo

de reconstrução tomográfica como um todo. Deve-se observar, portanto, que reduzir o tempo de exposição à radiação também diminui o tempo de aquisição dos dados. Por outro lado, é conhecido o fato de que ao diminuir a quantidade de fótons para a obtenção das projeções, o detector recebe menos informação no processo de aquisição, o que se caracteriza por baixas contagens de fótons, podendo corromper a imagem adquirida (SALVADEO, 2013). Logo, um dos problemas da reconstrução tomográfica consiste em tratar o ruído inerente do processo de aquisição.

Neste ponto, cabe uma explicação adicional, a fim de evitar uma interpretação contraditória. Quando se fala da reconstrução do ponto de vista matemático, teoricamente seriam necessárias infinitas projeções. Na prática isso não é possível, logo trabalha-se com um conjunto finito de projeções. Portanto, um mesmo conjunto pode ser considerado pequeno do ponto de vista matemático e grande do ponto de vista computacional. É importante observar ainda que mesmo que o conjunto seja pequeno do ponto de vista computacional deve-se considerar que cada projeção possui vários pontos (na casa dos milhares) o que na composição de processo de reconstrução pode significar grande esforço computacional.

Outro problema, na reconstrução tomográfica, consiste em se lidar com um número limitado de projeções. Considerando ainda que as projeções são adquiridas entre  $0^\circ$  e  $360^\circ$ , lidar com projeções obtidas em intervalos angulares limitados também pode ser outro problema a ser considerado no processo da aquisição de projeções para a reconstrução das imagens.

De modo geral, os algoritmos de reconstrução procuram compensar tais problemas como, por exemplo, no caso de poucas projeções utilizando técnicas de interpolação. No entanto, dificilmente a reconstrução tomográfica evitará o surgimento de artefatos na imagem reconstruída ou, ainda, deixará de exibir alguma informação do objeto escaneado. Um outro aspecto associado ao processo é o fato de que tais algoritmos demandam capacidade computacional e lidam com grandes quantidades de dados (SALINA; MASCARENHAS; CRUVINEL, 2002; PIPATSRISAWAT et al., 2005).

Compreende-se, portanto, que trabalhar com reconstrução tomográfica pode implicar manusear grandes quantidades de dados e necessidade de grande capacidade de processamento. Um exemplo disso é a pesquisa realizada na Embrapa Instrumentação dedicada a análise de sementes de milho e de soja na qual o tomógrafo levou 71 minutos, na configuração 2k (matrizes de  $2000 \times 1048$  pixels), para adquirir uma única amostra e mais 60 minutos para realizar a reconstrução e visualização tridimensional do conjunto de dados adquiridos. Ao alterar a configuração do equipamento para 4k (matrizes de  $4000 \times 2300$  pixels), verificou-se que o tempo de aquisição foi superior a 2 horas (144 minutos) enquanto que o processo de reconstrução 3D ficou acima de uma hora. Vale mencionar que todo o processamento foi realizado utilizando três computadores (sendo uma máquina com 24 núcleos processadores) e dispositivo de armazenamento com capacidade de 12TB (GOMES-JUNIOR; VAZ; CICERO; JORGE, 2014).

Como foi visto, os sistemas de gerenciamento agrícola precisam lidar com grandes

volumes de dados bem como ter maior capacidade de processamento e desempenho. Logo, os métodos que constituem tais sistemas precisam estar preparados para coletar, processar e compreender os dados de modo a apoiar a tomada de decisão. Isso vai de encontro com o ciclo da Agricultura de Precisão que se dá em três etapas: leitura; interpretação, planejamento; e, atuação. A primeira etapa consiste no levantamento e na obtenção de dados, geralmente, provenientes de equipamentos eletrônicos, embora há relatos que consideram o conhecimento tácito dos agricultores como fonte adicional de dados para os sistemas (BASSOI et al., 2014; INAMASU; BERNADI, 2014). A etapa de interpretação e planejamento leva em conta o uso de Sistema de Informação Geográfica (SIG), de geoestatística, de sistemas de suporte à decisão, entre outros métodos de análises e preocupa-se, também, com as séries históricas. A terceira etapa refere-se à aplicação da decisão tomada com suporte nos dados e informações obtidos no processo.

Com efeito, o ciclo reitera a necessidade de dados e de análises adequados e suficientes para atender às demandas do campo que aumentam a cada dia. Nesse sentido, a padronização de armazenamento de dados e da arquitetura de sistemas de informação distribuídos que permitam a integração dos diferentes tipos de dados de forma simples e transparente apresenta-se como outro aspecto importante a ser considerado no desenvolvimento de novos métodos (QUEIRÓS et al., 2014). Logo, observa-se que as Tecnologias de Informação e Comunicação (TIC) aliadas às agrotecnologias desempenham papel preponderante para que a Agricultura de Precisão seja um meio efetivo e importante para lidar com os desafios da atualidade.

Particularmente, em relação as TIC, verifica-se que nos últimos vinte anos o número de dispositivos eletrônicos como sensores, computadores, *smartphones*, *tablets*, televisores (*smart TV*), entre outros dispositivos, cresceu consideravelmente. A internet possibilitou que tais dispositivos estivessem cada vez mais conectados entre si, caracterizando inclusive, o que tem sido denominada a *Internet das Coisas*. O foco do desenvolvimento dos aplicativos passou a ser, prioritariamente, para ambientes distribuídos, móveis e *on-line*. Além disso, a redução do custo de armazenamento digital permitiu o aumento significativo dos dados digitais. Os processos nas diferentes áreas passaram (e estão passando) por reformulações assim como também novas soluções têm surgido.

Os termos *Big Data* e *Ciência dos Dados* surgem nesse contexto, pois representam uma nova forma de lidar com a grande quantidade de dados heterogêneos disponíveis atualmente e que se encontram muitas vezes de forma não estruturada, ou seja, não estão presentes apenas em bancos de dados relacionais. Os dados passaram a ser considerados ativos pelas empresas e pelos pesquisadores. Antes, por exemplo, as análises consideravam pequenos conjuntos de dados e, geralmente, era dispendioso obter grandes quantidades de dados. Atualmente, em diversas áreas, os dados estão disponíveis em maior escala. Além disso, os modelos baseados em métodos de otimização e computação estatística que antes tinham dificuldade de serem implementados por limitações de capacidade de processamento encontram, nos dias de hoje, melhores condições de serem utilizados. Há de se considerar ainda a possibilidade presente de sistematizar dados,



informações e análises de modo a organizar ambientes que mantenham históricos consistentes para prever e recomendar ações que auxiliem a tomada de decisão (MAYER-SCHONBERGER; CUKIER, 2013; LEE, 2017).

Observa-se que há uma tendência em aplicar o potencial que o Big Data e a Ciência dos Dados demonstram, nas mais diversas áreas como política, biologia, astronomia, medicina, marketing, segurança, economia, entre outras. Essa tendência também ocorre na agricultura. Entretanto, há desafios a serem superados como a conectividade no campo a fim de melhorar o fluxo de informação, o procedimento de coleta de dados com o intuito de torná-lo mais confiável, entre outros. Porém, a perspectiva é que o emprego do Big Data e da Ciência dos Dados na agricultura trará ganhos em termos de otimização de recursos, maximização de produtividade, enfim de sustentabilidade.

No processo de reconstrução tomográfica, por exemplo, verifica-se que o emprego de Big Data foi recentemente iniciado como, por exemplo, na tratativa das informações a serem reconstruídas tridimensionalmente e no desenvolvimento de novos algoritmos (ZHAO; FU; TAN; CAO, 2013; DITTER; FEY; SCHON; OECKL, 2014; ALVES; CRUVINEL, 2016; WANG, 2016; ZHANG et al., 2016; ALVES; CRUVINEL, 2018).

Sob tal enfoque, a organização de modelos de reconstrução bi e tridimensionais, bem como o emprego de Big Data no processo de reconstrução são pontos ainda considerados como campo aberto para a pesquisa, necessitando de esforços que levem a resultados confiáveis e que apresentem possibilidades de aplicação em problemas reais. Considerando, portanto, a contextualização do tema ora apresentada, discorre-se na próxima seção a motivação que levou ao desenvolvimento do presente trabalho.

## 1.2 Motivação

A reconstrução tomográfica em Tomografia de Raios-X (CT) a partir de projeções é considerada na área do processamento de imagens e sinais como um problema complexo, vez que é preciso lidar com conjuntos de sinais discretos no tempo e soluções que podem depender do tipo de amostras a serem analisadas. Em situações reais, customizada à determinada aplicação, a tratativa desses sinais discretos no tempo que compõem as projeções tomográficas ainda é um desafio para a pesquisa, pois apresentam ruídos dentre os quais o de natureza estatística do processo de emissão de fótons por uma fonte e detecção por um sensor que leva a uma aleatoriedade nas medidas dessas projeções. Ademais, soma-se à este aspecto outros ruídos de outras naturezas, incluindo aqueles provenientes de etapa da aquisição das projeções tomográficas ou da geometria do sistema de varredura das amostras (ruído mecânico), da instrumentação eletrônica nuclear para a conversão de fótons de Raios-X em corrente elétrica (ruído térmico), dos algoritmos computacionais utilizados, da arquitetura computacional adotada, entre outros. Outra questão que tem despertado grande interesse nesta área é a de se reconstruir imagens

tomográficas com boa qualidade utilizando menores conjuntos de projeções tomográficas. Isto vem ocorrendo porque tem sido verificado um maior uso da tomografia de alta e super-alta resolução, onde o volume dos dados tomográficos está cada vez maior, vindo a exigir novas abordagens computacionais para se lidar com os processos da reconstrução. Baseado nestas premissas, a principal motivação desta Tese de Doutorado se estabeleceu em se buscar determinar conjuntos mínimos e adequados de projeções para a reconstrução tomográfica de amostras agrícolas, reduzindo o número de projeções, o que não somente vem a implicar na redução do tempo envolvido na reconstrução das imagens e seu processamento, como também permitir para uma mesma janela de tempo um aumento significativo no número de análises. Em CT, o tempo de processamento é determinado em parte pelo método de reconstrução tomográfica utilizado para processar as projeções tomográficas e em parte pelo ambiente computacional em que o processo de reconstrução estiver ocorrendo. Neste sentido, verifica-se também que há o interesse em se buscar novas soluções de paralelização dos algoritmos envolvidos na reconstrução, bem como a utilização de arquiteturas que permitam o processamento em hardware. Contudo, cabe observar que o emprego de *clusters* de computadores, em nuvem, para a reconstrução tomográfica é ainda pouco explorado. Logo, a principal motivação deste trabalho foi baseada na obtenção do método que permite integrar a seleção de projeções, sua reconstrução tomográfica e técnicas Big Data para se viabilizar a realização de análises de grandes quantidades de amostras agrícolas, frente ao paradigma da Agricultura de Precisão.

### 1.3 Objetivo

Este trabalho tem como principal objetivo desenvolver um método para reconstrução tomográfica 2D e 3D(volumétrica) de imagens tomográficas que viabilize selecionar as projeções mais relevantes para se reconstruir imagens com boa qualidade, além de ser escalável e estar preparado para incluir recursos associados as análises e ao processo de tomada de decisão em agricultura.

### 1.4 Visão geral do método

A Figura 1 apresenta o diagrama de blocos que ilustra a visão geral do método desenvolvido para reconstrução 2D e 3D(volumétrica) de imagens tomográficas de amostras agrícolas em ambiente Big Data. É possível observar que as amostras foram obtidas por tomógrafos agrícolas cujas projeções foram inseridas e armazenadas no ambiente Big Data. O processamento consistiu na seleção das projeções mais relevantes para as etapas de reconstrução paralela bi e tridimensional. Finalmente as imagens foram disponibilizadas para visualização e análises. Verifica-se, na figura, que o ambiente Big Data é representado pela linha tracejada, portanto, o armazenamento, o processo de seleção e filtragem das projeções, bem como as reconstruções foram realizadas dentro de uma concepção de Big Data.

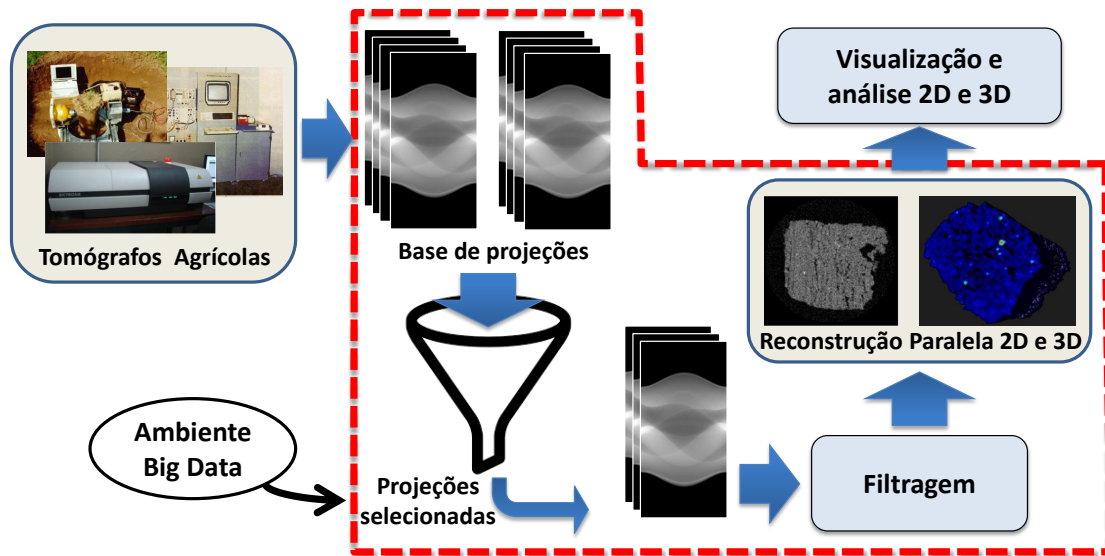


Figura 1 – Visão geral do método desenvolvido para reconstrução de imagens tomográficas em ambiente Big Data.

Na próxima seção são apresentados os trabalhos correlatos à esta pesquisa, os quais discorrem sobre métodos, algoritmos e o uso de Big Data no processo de reconstrução de imagens tomográficas.

## 1.5 Trabalhos correlatos

Determinar o conjunto mínimo e adequado de projeções para a reconstrução é uma tarefa difícil, vez que o processo de reconstruir uma imagem a partir de um conjunto de projeções exige tratar os ruídos provenientes da etapa de aquisição dos dados pelo tomógrafo que podem ser de natureza mecânica ou óptica bem como se adequar ao tipo de geometria do equipamento, além de considerar o esforço computacional e um modelo adequado para reconstrução. Uma abordagem para realizar esta tarefa está em utilizar a quantidade de informação como no *método da máxima entropia*<sup>2</sup> cujo objetivo é calcular os valores da função densidade do objeto para toda a matriz de projeções (sinograma) de modo que a entropia seja máxima e, conseqüentemente, que a informação seja maximizada (ROBERTY; REIS; CRISPIM, 1989).

Outro exemplo é o trabalho de Placidi e colaboradores (1995) que propuseram uma técnica de aquisição para seleção adaptativa de projeções que pode ser utilizada na ausência de conhecimento *a priori* da amostra. O intuito desses autores foi adaptar a seleção de projeções considerando o conjunto "mais informativo". Quando a amostra é suave ou possui simetria interna esta técnica permite a redução no número de projeções requeridas para a reconstrução da imagem.

<sup>2</sup> O valor esperado da quantidade de informação em relação a um determinado fenômeno é chamado de *entropia* e é dado pela equação:  $E[I(a_i)] = h. \sum_{j=1}^m p(a_j). \log(p(a_j))$ .

A abordagem dada por Ma (2017) considera o problema geral de reconstrução de funções de suporte compacto a partir dos coeficientes de Fourier usando o sistema *Shearlet*, que é uma extensão natural das *wavelets*, ou seja, essa modalidade de análise permite verificar o comportamento da função imagem tanto no domínio espacial quanto no domínio da frequência. O *Shearlet* permite converter características anisotrópicas em classes de problemas multivariados, ao contrário das *wavelets* que permitem a decomposição do sinal em uma família de sinais que guardam suas características, entretanto, escalonando a sua distribuição espacial. O processo de amostragem e reconstrução é analisado por meio do método *Generalized Sampling* (GS) no qual, segundo o autor, a recuperação estável de um sinal é possível a uma razão quase linear.

A análise de diagrama de fase, ferramenta padrão em *Compressive Sensing* (CS), como um método sistemático para determinar o menor número de projeções suficiente para uma reconstrução esparsa e regular com boa acurácia é um outro tratamento para a questão inicialmente apresentada de reconstruir uma imagem com poucas projeções (JORGENSEN; SIDKY, 2015). No *Compressive Sensing*, o diagrama de fase é uma maneira conveniente para se estudar e expressar certas relações entre esparsidade e amostragem suficientes. Esses autores apresentaram três estudos de caso, entre eles o emprego da técnica em imagens de grandes proporções, e mostraram que obter projeções (medidas) aleatórias não significa aumentar a performance comparado com padrões de amostragem estruturados. Eles argumentaram que diversos métodos viabilizam a obtenção de reconstruções com um número de projeções, substancialmente, menor do que o requerido por métodos tradicionais com o *Filtered Back Projection* (FBP) e o *Algebraic Reconstruction Technique* (ART). Na prática procura-se saber quantas projeções adquirir para obter uma reconstrução, esparsa e regular, de qualidade suficiente para resolver, confiavelmente, tarefas de processamento de imagens tais como detecção, classificação, segmentação, entre outras.

Outro aspecto a ser considerado no processo de reconstrução é a interpolação dos dados. Placidi e colaboradores (1996) apontaram que o *Filtered Back Projection* (FBP) é um dos algoritmos mais populares para reconstrução e é implementado como a soma de um número limitado de projeções. Na prática, na ausência de ruídos, é necessário um número  $M \approx (\pi/4) \times N$  cujas projeções possuem  $N$  pontos amostrados para que a reconstrução de uma imagem seja  $N \times N$ . No entanto, assim como Kak e Slaney (1989), eles observaram que a reconstrução introduz artefatos, mesmo com esse número de projeções ( $M$ ). A redução no número de projeções aumenta os artefatos tanto em intensidade quanto em quantidade. Nestes casos, geralmente, duas abordagens são adotadas para contornar a situação: obter mais projeções ou usar interpolação. Os autores ponderaram que a interpolação é usada em conjunto com o FBP mas que o seu uso incorreto também é fonte de artefatos. Em geral, a correlação entre as projeções de uma imagem não depende do seu espaço, pois ela muda de acordo com a forma do objeto. Por esta razão, é praticamente impossível para um sistema de interpolação, que não usa conhecimento *a priori*, recuperar exatamente as projeções ausentes. A escolha dos nós ótimos de interpolação (*optiomal interpolating knots*) é altamente dependente do objeto e tem mais efeito do que a escolha da

função de interpolação.

Em teoria, como visto, a reconstrução de uma imagem requer um número infinito de projeções e, além disso, para calcular as projeções filtradas deve-se supor que todas as componentes de Fourier são conhecidas. Na prática, um número finito de projeções,  $M$ , é coletado em coordenadas polares a um incremento angular  $\Delta_\theta$ . Para cada projeção,  $N$  pontos são adquiridos com incremento espacial  $\Delta_r$ . O algoritmo FBP consiste no cálculo (soma) depois de filtrar cada projeção o que aumenta as componentes de alta frequência proporcionalmente ao número de ondas.

O conjunto de dados gerado é formado por um número limitado de pontos e um número limitado de ângulos definido um por intervalo angular  $\Delta_\theta$ . A informação necessária (complementar) para reconstruir uma imagem por meio do FBP é extraída por interpolação. Dois tipos de interpolação são necessários: interpolação entre os pontos da projeção e interpolação entre as diferentes projeções.

A interpolação na projeção (entre os pontos) não introduz inconsistências. Uma projeção pode ser considerada, sem prejuízos, como sendo uma *função de banda limitada*. Se  $\Delta_r$  é a distância mínima de amostragem então  $1/(2\Delta_r)$  será a frequência máxima na projeção e representa o limite da resolução. As projeções medidas, neste caso, são a convolução entre as projeções reais (sinal original) e a função de espalhamento (*Point Spread Function*) e, portanto, a interpolação não aumenta o conteúdo das frequências.

A interpolação angular é um problema diferente. Ela pode gerar grandes artefatos devido a subamostragem angular. A reconstrução baseada na Transformada de Fourier não pode ser implementada sem interpolação. Neste caso, a maioria das técnicas de interpolação são consideradas locais, no sentido de que os coeficientes de Fourier desconhecidos são determinados pelos coeficientes vizinhos conhecidos. O método mais usado é o *zero-order* ou "vizinhos mais próximos" (*nearest-neighbor*), no qual o valor do coeficiente desconhecido é determinado, geralmente, como a média (simples ou ponderada) de alguns coeficientes vizinhos mais próximos a ele. Essas técnicas de interpolação local são computacionalmente eficientes, fáceis e, geralmente, apresentam boa acurácia.

Diante dessas questões (poucas projeções, interpolação, etc) diversos métodos têm sido estudados e propostos para o problema da reconstrução de imagens tomográficas, tanto bidimensionais quanto para as tridimensionais. Salina e colaboradores (2002) compararam quatro algoritmos considerando dois fatores: presença de ruído e reconstrução a partir de ângulos limitados. Além do *Algebraic Reconstruction Technique* (ART), os autores analisaram o *Simultaneous Iterative Reconstructive Technique* (SIRT) e o *Projections Onto Convex Sets* (POCS) sequencial e paralelo. Os resultados mostraram que o uso de restrições sobre as soluções, como no caso dos métodos POCS, foi eficaz para reduzir as variações devido ao malcondicionamento do problema, sendo que o POCS paralelo apresentou resultados melhores com imagens ruidosas.

Um modelo para redução do ruído utilizando técnicas de filtragem em imagens foi proposto por Laia e colaboradores (2008). Neste modelo as projeções obtidas do tomógrafo foram submetidas a uma filtragem unidimensional por meio do *Filtro de Kalman Estendido (com estimação conjunta)*. Essa etapa de filtragem teve por objetivo remover ruídos oriundo do processo físico de aquisição. Na etapa seguinte as projeções filtradas foram reconstruídas por meio do algoritmo analítico FBP. Durante o processo de reconstrução ocorreu uma filtragem, por meio das janelas de *Hamming*, cujo intuito foi reduzir artefatos produzidos pela retroconvolução, portanto a filtragem, neste caso, foi diferente da primeira. Na terceira etapa, a imagem tomográfica bidimensional foi, novamente, filtrada por meio da *Transformada Discreta de Wavelets (DWT)* que é uma técnica não-local e utilizou a base de Haar com dois coeficientes.

Os modelos também podem ser elaborados visando atender a um determinado tipo de aplicação, uma vez que a tomografia computadorizada pode ser empregada em diferentes segmentos. As aplicações industriais, como a inspeção industrial, a metrologia e a análise de alta-resolução (microeletrônica) utilizam tomógrafos com altas energias por meio dos detectores *Linear Detector Arrays (LDA)* que proporcionam campo de visão maior, bem como colimação superior aos demais equipamentos (detectores *flat panel*) no intuito de reduzir artefatos *scatter*. Neste sentido, Herold, Tischenko, Seidl e Kurfiss (2012) argumentaram que o algoritmo convencional FBP requer raios-X igualmente espaçados, ou seja, foi projetado para geometria paralela e seu esforço computacional depende principalmente do tamanho do volume reconstruído. Portanto, os autores propuseram o algoritmo *Orthogonal Polynomial Expansion on the Disk (OPED)*. A complexidade numérica do algoritmo depende da quantidade de dados de entrada (projeções) e foi projetado para superar artefatos de *aliasing* e *streaks* sem a necessidade de filtragem espacial de modo que seja facilmente parametrizado. O algoritmo foi comparado com o FBP e obteve performance melhor de até três vezes mais do que o FBP para conjuntos que chegaram a 4096 projeções.

O autor Shtok e colaboradores (2010) propuseram um método iterativo baseado em estatística para reconstrução direta baseada em fusão local de poucas imagens estimadas, *a priori* por meio da regra da fusão não linear. A regra é espacialmente adaptativa à suavidade de regiões locais desconhecidas. O objetivo do método, de acordo com os autores, foi superar algumas desvantagens do algoritmo analítico FBP como, por exemplo, o fato de que: ele não considera os numerosos fenômenos físicos no processo de aquisição dos dados acarretando os artefatos *streak*; sofre erros de discretização; falta flexibilidade para processar dados parciais de entrada (projeções truncadas para uma região de interesse ou projeções restritas a intervalo angular limitado).

No campo da reconstrução de imagens tomográficas 3D, o trabalho de Minatel e Cruvinel (1998) apresentou um algoritmo para reconstrução e visualização tridimensional de imagens tomográficas com uso de técnicas frequenciais e *wavelets*. Os dados das projeções obtidos puderam ser filtrados por *wavelet Haar* ou *Hamming* além de ter sido possível aplicar um



modelo de restauração sobre as projeções antes da retroprojeção. Na sequência foi realizada a reconstrução bidimensional (por meio do algoritmo FBP) e técnicas de processamento de imagens (pseudo-cores, normalização de paletas, thresholds) puderam ser empregadas na imagem gerada pela reconstrução, além de suavizar eventuais ruídos por meio da filtragem *wavelets Haar* 2D (por limiar ou técnica adaptativa). Na próxima etapa, os planos tomográficos (imagens 2D) foram interpolados estimando dados intermediários a partir dos dados previamente conhecidos e foram testados Wavelets e interpoladores polinomiais. Esse processo foi a base para a reconstrução volumétrica e o resultado consistiu em uma imagem 3D dos dados obtidos do mini-tomógrafo (representação 3D do objeto real escaneado). A análise visual dos dados reconstruídos foi possível por meio de ferramentas de visualização 2D e 3D. A visualização bidimensional exibia cortes (transversais, coronais e sagitais) do objeto tridimensional (matriz 3D gerada pela interpolação das imagens 2D) e a visualização 3D exibia objetos na forma de sólidos, com o uso de iluminação e transformações geométricas (translação, rotação, escala) possibilitando a noção de perspectiva tridimensional.

Os autores [Pereira e Cruvinel \(2015\)](#) desenvolveram um modelo de reconstrução tomográfica volumétrica para amostras agrícolas com filtragem de Wiener em processamento paralelo. Os dados das projeções foram reconstruídos bidimensionalmente por meio do algoritmo FBP. O modelo permitiu ainda filtrar os dados utilizando o filtro de Wiener antes da reconstrução 2D a fim de suavizar eventuais ruídos, a exemplo do ruído Poisson. No modelo de filtragem estabelecido as projeções foram submetidas à Transformada de Anscombe, na sequência passaram pela filtragem de Wiener por predição e, por fim, foram submetidas a Transformada Inversa de Anscombe. O modelo PCAM (Particionamento; Comunicação; Aglomeração e Monitoramento) foi utilizado para modelar o algoritmo FBP de forma paralela. A reconstrução tridimensional utilizou a técnica de interpolação por B-Spline-Wavelets dos planos 2D reconstruídos previamente e a modelagem do algoritmo paralelo também utilizou o modelo PCAM. Os algoritmos paralelo FBP (para reconstrução bidimensional) e interpolação B-Spline-Wavelets (para reconstrução tridimensional), a qual foi verificada com sucesso, foram implementados na arquitetura *Digital Signal Processing* (DSP) e realizou-se a comparação com os resultados obtidos na plataforma DSP a partir do desenvolvimento de três diferentes implementações do método (algoritmos de reconstrução 2D e 3D) em ambiente computacional convencional utilizando a biblioteca de comunicação *Message Passing Interface* (MPI), a fim de examinar as potencialidades em ambientes com múltiplos núcleos ([CRUVINEL; PEREIRA; SAITO; COSTA, 2009](#)). Adicionalmente, um ambiente de visualização tridimensional utilizando a biblioteca *Visualization Toolkit* (VTK) foi desenvolvido.

A pesquisadora [Faria \(2003\)](#) apresentou um método de reconstrução 3D a partir de duas imagens tomográficas de raios-X. Tal método foi baseado em metamorfose de imagens, que geraram imagens intermediárias de seções transversais por meio de um método de deformação de imagens controlado por uma curva Bezier, assim como um método de interpolação 3D de imagens utilizando a *Teoria de Estimção Bayesiana*. O método foi desenvolvido para a

reconstrução de arcos coronais a fim de auxiliar o estudo sobre previsão de explosões solares. A implementação foi realizada em ambiente paralelo para melhorar o desempenho do algoritmo e utilizou a abordagem PCAM durante a modelagem do algoritmo.

Uma preocupação frequentemente encontrada nos métodos de reconstrução estudados e desenvolvidos é quanto a performance e tempo de processamento. Algumas estratégias são adotadas como o desenvolvimento do algoritmos em ambientes paralelos como MPI, DSP e utilizando de modelos de programação paralela a exemplo do PCAM. Recentemente, o uso de placas *Graphical Processing Units* (GPU) e/ou *clusters* abriram novas possibilidades para o desenvolvimento de métodos de reconstrução que exijam mais poder computacional. Um exemplo é o trabalho de Blas et al. (2014) que apresentou a implementação e otimização de um sistema de reconstrução tomográfica em placa GPU. Outro exemplo é o estudo comparativo realizado por Serrano, Blas e Carretero (2015) sobre o desempenho de algoritmos de reconstrução tomográfica de raios-X em placas GPU e sistemas de *cloud computing* cujo objetivo foi analisar a efetividade de aplicações de reconstrução sobre diferentes soluções de alta performance disponíveis na atualidade.

O emprego de novas tecnologias demonstra ser uma tendência para que novas soluções sejam encontradas e que, eventualmente, demandem grande poder computacional ou, ainda, lidem com grandes volumes de dados. Neste sentido é muito recente o interesse dos pesquisadores em utilizar técnicas de *Big Data* para desenvolverem novos métodos de reconstrução de imagens tomográficas.

Embora o termo *Big Data* ainda não tenha um consenso quanto à sua definição, como será discutido posteriormente, ele geralmente é associado a três características: volume, variedade e velocidade.

O *volume* relaciona-se com a capacidade do sistema processar grandes conjuntos de dados. A *variedade* diz respeito a complexidade dos dados que podem ser estruturados, semi-estruturados ou não estruturados. A *velocidade* observa a habilidade do sistema em processar diferentes tipos de informações em tempo hábil. Além disso, *Big Data* possui potencial para ser aplicado à diferentes áreas como gestão, astronomia, biologia, educação, economia, política, e agora na agricultura, entre outras. Diversas pesquisas têm sido desenvolvidas nestas áreas como, por exemplo, a adaptação de algoritmos tradicionais de mineração para o atendimento de requisitos do Big Data, ou a análise de escalabilidade de aplicações Hadoop MapReduce que visam identificar os parâmetros mais adequados para obter maior escalabilidade (ROCHA, 2013; MELLO, 2015; CARVALHO, 2016; MARQUESONE, 2017).

Na área de reconstrução de imagens tomográficas, iniciativas que empregam os conceitos e técnicas associadas ao *Big Data* começam a ser desenvolvidos. Yang e colaboradores (2015) justificaram que cerca de 100 milhões de imagens tomográficas são realizadas no mundo anualmente e que a maioria é descartada. Esses autores propuseram um projeto cujo intuito foi o de arquivar, utilizar e compartilhar uma grande base de imagens tomográficas voltada



para pesquisadores preocupados em aperfeiçoar e desenvolver técnicas de análises e reconstrução de imagens tomográficas visando reduzir as doses de radiação utilizadas atualmente na tomografia. Inicialmente, a base de imagens é composta por várias imagens obtidas de quatro cadáveres (dois masculinos, dois femininos) que foram escaneados de acordo com protocolo médico convencional. Esse artigo apresenta como os arquivos e metadados foram arquivados utilizando o protocolo *File Transport Protocol* (FTP). Dados com baixa e alta dose de radiação foram obtidos, além disso não foram obtidos artefatos provenientes de movimentação dos órgãos. Tais características permitem a análise mais precisa dos métodos de reconstrução. Os autores utilizaram o tomógrafo GE Discovery 750 HD e escanearam os corpos nos seguintes espectros de raio-X (energia): 140kVp<sup>3</sup>, 120kVp, 100kVp e 80kVp. Além disso, o equipamento utilizava um índice de ruído (NI) que é próximo do desvio padrão do número CT e definiu a qualidade das imagens. Foram utilizados os seguintes valores de NI: 10, 20, 30 e 40. Para a reconstrução das imagens o equipamento ofereceu três métodos, a saber: FBP, *Adaptive Statistical iterative Reconstruction* (ASiR) e Veo, o qual é o primeiro produto comercial baseado no *Model-Based Iterative Reconstruction* (MBIR). Para análise de filtragem das imagens reconstruídas (usaram o índice MSE - *Mean Square Error*), os autores utilizaram quatro técnicas: *Total Variation Minimization*; *Non-Local Means* (NLM); *Block-Matching 3D* (BM3D); Método baseado em *frames Wavelets*.

Outro trabalho nesta linha é o de Lee e colaboradores (2014) cujo objetivo foi implementar o algoritmo iterativo de reconstrução estatística de imagens tomográficas denominado *Maximum Likelihood Expectation Maximization* (MLEM) no ambiente Spark/GraphX. Os autores apontaram que o processamento de imagens, usando transmissão ou emissão tomográfica, exige uma quantidade significativa de tempo computacional para reconstruir imagens com boa qualidade. Eles apontaram ainda que o movimento do paciente (respiração) e do coração (batidas) produziam artefatos indesejados na imagem reconstruída o que exigiu mais capacidade de processamento dos algoritmos.

O artigo intitulado “*A perspective on Deep Imaging*” expressou opiniões e ideias de Wang (2016) sobre o uso de *Deep Learning*, *Machine Learning* e *Big Data* voltados para a reconstrução de imagens tomográficas, as quais o referido autor considerou se compor em uma mudança de paradigma. Ele considerou que o emprego dessas técnicas possibilitará uma nova geração de métodos e teorias de reconstrução de imagens e, a essa fusão, ele denominou de *Deep Imaging*. Esse autor ainda ponderou que há duas principais vertentes do imageamento médico: (i) a *formação de imagens/reconstrução de imagens*, que parte dos dados para a imagem; (ii) o *processamento de imagens/análise de imagens*, que parte de uma imagem para outra imagem melhor, a exemplo do *denoising*<sup>4</sup> e do reconhecimento de padrões. Foram também destacadas aplicações de reconhecimento de padrões em que o *Deep Learning* é empregado com

<sup>3</sup> kVp: Kilovolts pico

<sup>4</sup> O objetivo das técnicas de *denoising* é reduzir o ruído e, ao mesmo tempo, preservar as características da imagem.

sucesso, usando redes neurais como a *Restricted Boltzman Machine* (RBM). Basicamente, a entrada da rede recebe a imagem e, na saída, obtém-se as suas características. Nesse modelo, cada camada utiliza características das camadas anteriores para formar outras mais avançadas, modelo esse que vêm sendo chamado de *redes profundas de aprendizagem*. Tal autor ainda argumentou que há muitas publicações e aplicações voltadas para processamento e análise de imagens, logo isso representa uma oportunidade para explorar as novas técnicas voltadas à formação e reconstrução de imagens. Assim, sugeriu que a aplicação de reconstrução é inversa a do reconhecimento de padrões, uma vez que parte-se dos dados (características) para a imagem, logo na entrada da rede estariam os dados e na saída, a imagem. Por outro lado, quanto ao processo de reconstrução de imagens tomográficas, destacou que tanto o FBP quanto o *Simultaneous Algebraic Reconstruction Technique* (SART) podem ser facilmente formulados de forma paralela (logo em estruturas de camadas paralelas). Ponderou que se os dados estão amostrados abaixo da taxa de Nyquist, o uso dos métodos baseados na Transformada de Radon e Teorema das seções de Fourier frequentemente produzem artefatos nas imagens reconstruídas, que são estruturados e não-locais. Logo o *Deep Learning* pode lidar, a princípio, com características globais da imagem mesmo que elas estejam substancialmente distorcidas. Observou que os algoritmos iterativos de reconstrução tem se tornado, gradualmente, mais populares do que os analíticos como o FBP embora o *Deep Learning* possa ser aplicado em ambas categorias de algoritmos.

O autor Wang (2016) analisou que no passado tinha-se um conjunto completo de projeções e a reconstrução era, majoritariamente, por método analítico (FBP). Atualmente, entretanto, comentou que se busca trabalhar com conjuntos de dados provenientes de projeções de ângulos limitados e/ou obtidos com baixa dose enquanto que os métodos iterativos têm sido mais demandados. Visualizou que no futuro, a expectativa é que os conjuntos sejam obtidos com baixa dose de radiação, de modo otimizado, e que a reconstrução (analítica ou iterativa) se dará com o apoio do *Deep Imaging*.

Finalmente, a partir da bibliografia visitada nesta seção, foi possível identificar aspectos e questões relacionadas a seleção de projeções tomográficas bem como ao processo da reconstrução tomográfica ainda abertos à pesquisa e que se configuram em oportunidades tendo em vista a possibilidade de completar lacunas nesta área.

## 1.6 Organização do documento

Este documento está organizado em seis capítulos. O primeiro capítulo refere-se à introdução a qual contextualizou o trabalho bem como apresentou a motivação, o objetivo e a visão geral do método. O segundo capítulo é dedicado a Tomografia Computadorizada e apresenta o histórico, o uso na agricultura (principalmente no Brasil) e aborda os fundamentos matemáticos. O terceiro capítulo discute Big Data e Ciência dos Dados bem como as tecnologias associadas. No quarto capítulo é apresentado o método para reconstrução tomográfica de amostras agrícolas

em ambiente Big Data. O quinto capítulo apresenta os resultados e discussões referentes aos experimentos realizados. Por fim, o último capítulo apresenta as conclusões deste trabalho.

# Capítulo 2

## TOMOGRAFIA COMPUTADORIZADA

---

---

Neste capítulo é apresentado um histórico da Tomografia Computadorizada e uma discussão sobre suas aplicações, bem como os fundamentos da reconstrução tomográfica.

### 2.1 Histórico

A história da tomografia computadorizada é marcada por diversos momentos. Em 1956, o físico Allan MacLeod Cormack formulou uma matriz de coeficientes de atenuação linear ( $cm^{-1}$ ) para cortes seccionais que poderia ser obtida pela medida da transmissão de raios-X em vários ângulos através de um corpo. Em 1963, ele propôs a reconstrução de um corpo com base em um número finito de projeções. Neste ponto, cabe explicar que a tomografia utiliza diversos feixes colimados de raio-X que definem projeções que são utilizadas para gerar planos verticais. Em 1973, o engenheiro eletricitista Godfrey Newbold Hounsfield produziu o primeiro tomógrafo médico comercial. Neste contexto, se faz também importante mencionar os trabalhos realizados por Michael Faraday (1831), Wilhelm Conrad Röntgen (1895) e Johann Radon (1917). Michael Faraday foi um dos primeiros a estudar as relações entre eletricidade e magnetismo. Ele descobriu, em 1831, a indução eletromagnética e os seus estudos são considerados conceitos-chave da física atual que encontram aplicações em diversas áreas incluindo na tomografia. Wilhelm Conrad Röntgen produziu e detectou, em 1895, radiação eletromagnética em comprimentos de ondas as quais se conhece como *raio-X*<sup>1</sup>. Em 1917, Johann Radon propôs uma solução matemática das equações de reconstrução de corpos a partir de projeções, conhecida atualmente como *Transformada de Radon*. Vale destacar que Cormack veio a conhecer o trabalho de Johann Radon após ter realizado o seu trabalho em 1956. O histórico, mais detalhado, sobre o surgimento e desenvolvimento da tomografia computadorizada pode ser verificado nos trabalhos de Cruvinel (1987) e Hsieh (2009).

Cerca de quatro décadas após o desenvolvimento do primeiro tomógrafo comercial verifica-se que o uso da captura de imagens por meio da tomografia computadorizada de raio-X cresceu consideravelmente. Em 2015, na área médica, a estimativa foi que mais de 100 milhões

<sup>1</sup> Fonte: [https://en.wikipedia.org/wiki/Wilhelm\\_Röntgen](https://en.wikipedia.org/wiki/Wilhelm_Röntgen). Último acesso em 19/06/2017.

de varreduras tomográficas foram realizadas em todo o mundo. Na Inglaterra, por exemplo, dados oficiais do Serviço Nacional de Saúde (NHS) apontaram que o número de exames tomográficos no biênio 2014-2015 foi superior a 4 milhões (YANG et al., 2015; NHS, 2015).

A popularização da tomografia computadorizada, inicialmente na área médica, naturalmente fez com que ela fosse adotada em outras áreas. Na indústria, por exemplo, foi possível aprimorar as áreas de qualidade de produtos por meio de testes e ensaios não destrutivos a fim de detectar falhas, como fissuras e vazios; análise de partículas; metrologia dentre outras (CHIFFRE et al., 2014). A agricultura é outra área na qual passou a utilizar tomografia computadorizada visando aprimorar processos e análises.

## 2.2 O uso da tomografia computadorizada na agricultura

O estudo da tomografia computadorizada aplicada à agricultura começou no início da década de 80, particularmente voltada a área de ciência do solo a fim de estudar os processos de infiltração de água bem como as propriedades de densidade, umidade e porosidade cujos trabalhos pioneiros são atribuídos a Petrovic e colaboradores (1982); Hainsworth e Aylmore (1983), e; Crestana e colaboradores (1985) (SILVA et al., 2007; PIRES; BORGES; BACCHI; REICHARDT, 2010).

Petrovic e colaboradores iniciaram os estudos sobre densidade do solo por meio da tomografia computadorizada ao demonstrarem a relação linear entre densidade e atenuação de raio-X. Por sua vez, os pesquisadores Hainsworth e Aylmore, Crestana e colaboradores introduziram os estudos de medidas do conteúdo e movimento da água no solo (PETROVIC; SIEBERT; RIEKE, 1982; HAINSWORTH; AYLMORE, 1983; CRESTANA; MASCARENHAS; POZZI-MUCELLI, 1985; TAINA; HECK; ELLIOT, 2007; PIRES; BORGES; BACCHI; REICHARDT, 2010).

No Brasil, o primeiro minitomógrafo de raios X e  $\gamma$  para aplicações da ciência do solo foi construído em 1987 possibilitando a realização de medidas de amostras em laboratório constituindo, portanto, passo importante no desenvolvimento e avanço da técnica tomográfica (CRUVINEL, 1987; CRUVINEL; CESAREO; CRESTANA; MASCARENHAS, 1990; CRUVINEL; CRESTANA, 1996; NAIME et al., 2014). Posteriormente, outros tomógrafos foram desenvolvidos com o de escala milimétrica, o tomógrafo portátil de raio  $\gamma$  ou o tomógrafo de espalhamento Compton (NAIME, 1994; SILVA, 1997; NAIME, 2001; BALOGUN; CRUVINEL, 2003; SCANNAVINO, 2013).

Desde então, diversas pesquisas têm sido realizadas com o emprego da técnica tomográfica. Um exemplo, é a caracterização de solos a partir da análise de imagens tomográficas, reconstruídas computacionalmente, na qual é possível observar parâmetros como, o ROB (*Razão de Ocupação Binária*) que permite verificar a relação entre a área de sólidos e a área total da imagem ou o GA (*Grau de Anisotropia*) que permite observar o alinhamento de vazios dentro da

imagem.

A caracterização de solos por meio da análise desses parâmetros possibilita, por exemplo, quantificar as principais características que influenciam na percolação. Além disso, outros exemplos podem ser citados como, o estudo do condicionamento físico do solo ao redor das sementes visando o bom desenvolvimento inicial de uma cultura; a análise da qualidade ambiental de solo para fins de recuperação de áreas degradadas; a avaliação da densidade de solos considerando, por exemplo, a ocorrência dos processos de compactação e de adensamento de solo; a análise de mudanças na estrutura do solo como a modificação da densidade e da porosidade de amostras deformadas de solo. Outras linhas de pesquisas agrícolas também utilizam a tomografia computadorizada como na análises de plantas, de sementes, madeiras e rochas (HEERAMAN; HOPMANS; CLAUSNITZER, 1997; JUNIOR et al., 2002; PEDROTTI et al., 2003; MODOLO et al., 2008; TETZNER, 2008; PIRES; BACCHI, 2010; TSENG, 2013; BERALDO; JUNIOR; CRUVINEL, 2014; ZUBELDIA et al., 2014; PASSONI; PIRES; HECK; ROSA, 2015; PALOMBO, 2016).

Verifica-se, portanto, a importância da tomografia computadorizada para análises e testes não invasivos e não destrutivos na agricultura, embora ainda seja necessário avaliar e aperfeiçoar procedimentos (TAINA; HECK; ELLIOT, 2007; AMBERT-SANCHEZ et al., 2016). Por exemplo, o ajuste de parâmetros de aquisição, no tomógrafo, como ângulo de rotação da amostra e uso de filtros pode influenciar na relação sinal-ruído e, conseqüentemente, na qualidade de imagem. Por outro lado, pode significar o aumento no tempo de aquisição que compromete o tempo de vida útil do equipamento (LASSO; VAZ.; BACCHI, 2009).

A resolução da imagem é outro aspecto a ser considerado, pois o tamanho da resolução é inversamente proporcional ao tamanho do pixel, ou seja, a resolução maior leva a um pixel menor e vice-versa. Logo, isso pode influenciar a análise de parâmetros pois, por exemplo, em uma resolução maior é possível identificar um conjunto maior de poros do que em uma imagem com resolução menor.

É preciso considerar, ainda, outras questões como o ruído presente nas projeções que levam ao surgimento de artefatos na imagem reconstruída e o desenvolvimento de novas abordagens para a reconstrução tomográfica (SALINA; MASCARENHAS; CRUVINEL, 2002; SALVADEO, 2013; ASSIS; SALVADEO; MASCARENHAS; LEVADA, 2015; GERALDO; CURA; CRUVINEL; MASCARENHAS, 2017).

Essas questões representam um desafio para o método de reconstrução que, do ponto de vista matemático e computacional, precisa recuperar a imagem com o menor número de artefatos (oriundos da etapa de aquisição ou do próprio processo) e, ainda, é desejável que seja com o menor número de projeções.

Na próxima seção, a etapa de reconstrução de imagens tomográficas é discutida sob o ponto de vista matemático.

## 2.3 Reconstrução tomográfica

O problema central da Tomografia Computadorizada consiste em se obter uma imagem do objeto em estudo a partir da reconstrução de projeções que foram obtidas com base em transmissão, emissão, ou espalhamento de radiação ionizante. O tipo de radiação ionizante, bem como a maneira de se obter as medições das atenuações podem se dar de diferentes formas.

Neste trabalho será dado enfoque na Tomografia Computadorizada por transmissão que, de modo geral, consiste em reconstruir uma imagem por meio da obtenção de integrais de linha ao longo de retas que atravessam o objeto. Na tomografia por emissão, o corpo emite radiação por meio de radio-isótopos, de modo que as informações das emissões são utilizadas para recuperar a imagem (KAK; SLANEY, 1989; HSIEH, 2009). Na tomografia por espalhamento, a radiação ionizante espalhada é tomada para a composição das projeções, como é o caso da Tomografia Compton (*Compton Tomography*) (SCANNAVINO, 2013). Destaca-se, ainda, que as tomografias de raio-X e raio- $\gamma$  utilizam o princípio da transmissão, as tomografias PET (*Positron Emission Tomography*) e SPECT (*Single Photon Emission Tomography*) o da emissão (HSIEH, 2009).

O modelo físico da atenuação de raio-X na tomografia computadorizada por transmissão é ilustrado na Figura 2. Um feixe estreito representado por uma reta  $L$  com intensidade  $I(x)$ , parte da fonte e atravessa o objeto, que possui um determinado coeficiente de atenuação  $\mu$ . Do outro lado, o detector registra a intensidade restante do feixe cuja informação será utilizada para reconstruir a imagem bidimensional do objeto (KAK; SLANEY, 1989; RANGAYYANI, 2004; HSIEH, 2009).

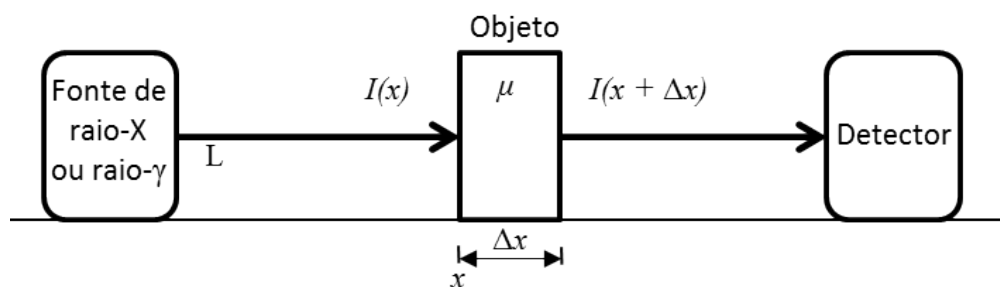


Figura 2 – Modelo físico da atenuação de raio-X.

Considerando a distância  $\Delta x$  percorrida pelo feixe de intensidade  $I(x)$  através do objeto que possui coeficiente de atenuação  $\mu(x)$ , pode-se determinar a atenuação total sofrida pelo feixe por meio da Equação 2.1.

$$I(x + \Delta x) = I(x) - \mu(x)I(x)\Delta x \quad (2.1)$$

Considerando que  $\Delta x$  seja infinitesimal, logo  $\Delta x \rightarrow 0$ , assim pode-se organizar a Equação 2.1 como:

$$\frac{I(x + \Delta x) - I(x)}{\Delta x} = -\mu(x)I(x)$$

$$\lim_{\Delta x \rightarrow 0} \frac{I(x + \Delta x) - I(x)}{\Delta x} = -\mu(x)I(x)$$

$$\frac{dI}{dx} = -\mu(x)I(x) \quad (2.2)$$

Utilizando o método de separação de variáveis e integrando-se os dois lados da Equação 2.2, tem-se:

$$\frac{dI}{I(x)} = -\mu(x)dx$$

$$\int_{I_0}^I \frac{1}{I(x)} dI = - \int_L \mu(x)dx \quad (2.3)$$

Na equação 2.3 deve-se observar os limites de integração. No lado esquerdo,  $I_0$  representa a intensidade inicial de radiação do feixe de raio-X, ou seja, representa a intensidade  $I(x)$  antes de atravessar o objeto. Por sua vez, o limite de integração  $I$  representa a intensidade final, ou seja, a intensidade  $I(x)$  após atravessar o objeto. No lado direito da equação,  $L$  representa a reta pela qual está direcionado o feixe de raio-X e que passará pelo objeto. Portanto, calculando-se a integral, do lado esquerdo, chega-se a:

$$\ln |I| - \ln |I_0| = - \int_L \mu(x)dx$$

$$\ln |I| = \ln |I_0| - \int_L \mu(x)dx$$

$$I = \exp \left( \ln |I_0| - \int_L \mu(x)dx \right)$$

$$I = \exp(\ln |I_0|) \exp \left( - \int_L \mu(x)dx \right)$$

$$I = I_0 \exp \left( - \int_L \mu(x)dx \right) \quad (2.4)$$

A Equação 2.4 é conhecida como equação de *Lambert-Beer* e expressa que a quantidade de raio-X é atenuada exponencialmente ao longo da reta  $L$ . Se for considerado o intervalo  $L = [0, x]$ , então a equação pode ser reescrita como:

$$I = I_0 \exp(-\mu(x)x) \quad (2.5)$$



Logo,  $I$  representa a intensidade do feixe de raio-X após atravessar o objeto, ou seja, a intensidade atenuada. Para fins de reconstrução tomográfica, é de interesse medir a variação desta atenuação ao longo da reta  $L$ . Isto é possível por meio da operação inversa da Equação 2.4, portanto

$$I = I_0 \exp \left( - \int_L \mu(x) dx \right)$$

$$\ln \left( \frac{I}{I_0} \right) = - \int_L \mu(x) dx$$

$$p(L) = \int_L \mu(x) dx = - \ln \left( \frac{I}{I_0} \right) \quad (2.6)$$

A Equação 2.6 é uma medida de projeção que afirma que a relação entre a intensidade do raio-X de saída sobre a de entrada, após uma operação de logaritmo, representa a linha integral dos coeficientes de atenuação ao longo de  $L$ . Um projeção pode ser entendida, portanto, como um conjunto finito de retas  $L$  medidas por essa equação.

De modo geral, busca-se estimar e calcular a distribuição de atenuação  $\mu(x)$ , dado um conjunto de medidas do objeto. Nesse sentido, a Transformada de Radon é empregada para realizar tais cálculos.

### 2.3.1 Transformada de Radon

O objetivo da Transformada de Radon é descobrir uma função  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  a partir de todas integrais de linhas em um domínio previamente determinado. Na tomografia computadorizada, ela pode ser utilizada para determinar a distribuição de atenuação  $\mu(x)$  que corresponde a densidade do objeto em estudo. O problema, portanto, é considerado um problema inverso na medida em que se busca encontrar o coeficiente de atenuação a partir dos dados disponíveis, ou seja a partir de  $I$  e  $I_0$ .

Uma maneira de compreender o processo de reconstrução tomográfica é considerar um feixe de raio-X como uma reta, que parte da *fonte* até o *detector*. Esse conjunto (fonte, detector) é rotacionado por um ângulo  $\theta \in [0, 2\pi)$  de forma que todo o objeto seja escaneado no plano em uma posição  $z$  fixa. A Figura 3 ilustra a vista de cima de um tomógrafo no qual os feixes de raio-X são emitidos paralelamente, também chamado de tomografia de geometria paralela. Na figura, o círculo menor indica a fonte e o quadrado o detector enquanto que o círculo interno (preenchido de cinza) indica o objeto.

Considerando a rotação do conjunto (fonte, detector) pode-se reescrever a Equação 2.6 em função da fonte de raio-X e do ângulo  $\theta$  de modo a adotar um sistema de coordenadas  $(\varepsilon, \eta)$

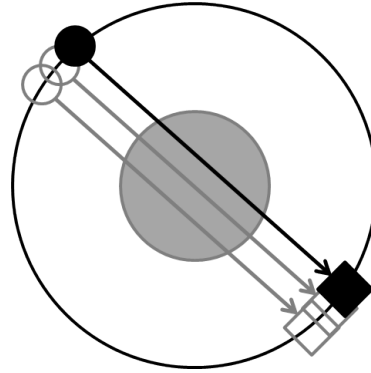


Figura 3 – Representação da tomografia em feixes paralelos.

que rotacione junto com a fonte e detector. A Equação 2.7 representa a projeção no novo sistema de coordenadas.

$$P_{\theta}(\varepsilon) = \int_L \mu(\varepsilon, \eta) d\eta \quad (2.7)$$

Observa-se que a função de densidade  $\mu$  passa a ser de duas variáveis e que a equação representa a integral de linha ao longo da reta  $L$  que descreve a posição  $\varepsilon$  do conjunto (fonte, detector) com um ângulo  $\theta$  em relação ao plano de coordenadas  $(x, y)$ . A Figura 4 ilustra os sistemas de coordenadas utilizados na tomografia.

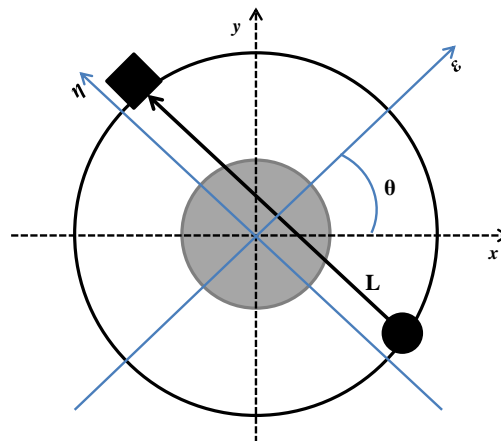


Figura 4 – Sistemas de coordenadas utilizados na tomografia.

Na prática, é ideal que os valores de coeficiente de atenuação linear sejam dados em função das coordenadas fixas  $(x, y)$ , o que é possível utilizando a relação entre as coordenadas polares e cartesianas. Desta maneira, pode-se determinar a distância perpendicular da origem até a reta  $L$  (feixe de raio-X) por meio da Equação 2.8.

$$t = x \cos(\theta) + y \sin(\theta) \quad (2.8)$$

A Figura 5 apresenta diagrama esquemático de uma projeção paralela destacando a distância  $t$ . Observa-se que uma projeção é um conjunto de integrais de linhas, representadas por  $P_\theta(t)$ , considerando o mesmo corte transversal, ou posição no eixo- $z$ , bem como o mesmo ângulo  $\theta$  do conjunto (fonte, detector) em relação às coordenadas fixas  $(x, y)$ . A matriz de rotação entre as coordenadas é dada pela Equação 2.9.

$$A = \begin{pmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{pmatrix} \quad (2.9)$$

É possível observar que  $P_\theta(t)$  é um gráfico em função de  $\varepsilon$  no qual o ângulo  $\theta$  determina a inclinação do eixo- $\varepsilon$  com relação à linha horizontal e a integral da função é efetuada sobre a reta perpendicular a esse eixo. Observa-se também que não é necessário varrer todo o intervalo  $\theta \in [0, 2\pi)$ , mas apenas o intervalo  $\theta \in [0, \pi)$  de modo a evitar redundância dos dados.

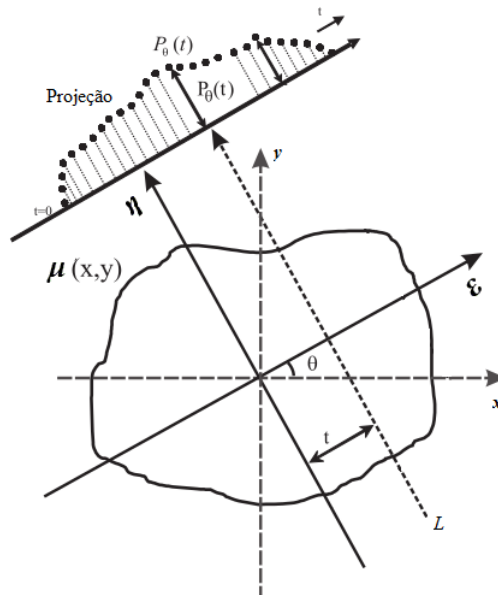


Figura 5 – Diagrama esquemático de uma projeção paralela.

Considerando o fato de que, computacionalmente, não é possível ter infinitas integrais de linhas pode-se representar o corte transversal, em um determinado ângulo, fazendo uso do operador delta de Dirac o qual possui a propriedade de amostragem e, portanto, a Equação 2.7, no caso bidimensional, pode ser reescrita como

$$P_\theta(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) \delta(x \cos \theta + y \text{sen } \theta - t) dx dy \quad (2.10)$$

A Equação 2.10 é conhecida como a *Transformada de Radon*,  $\mathcal{R}_\theta \mu(t) = P_\theta(t)$ , portanto, o problema de reconstruir uma imagem consiste em determinar  $\mu(x, y)$  a partir de  $\mathcal{R}_\theta \mu(t)$ . A Transformada de Radon mapeia o domínio espacial  $(x, y)$  no domínio  $(t, \theta)$ , no qual cada ponto no espaço  $(t, \theta)$  corresponde a uma linha no espaço  $(x, y)$ .

A Transformada Inversa de Radon,  $\mathcal{R}^{-1}$ , é utilizada para reconstruir a função  $\mu$  e pode ser obtida por meio do Teorema das seções de Fourier, ou Teorema do Corte Central. Este teorema relaciona as projeções da Transformada de Radon com a Transformada de Fourier.

### 2.3.2 Teorema das seções de Fourier

O Teorema das seções de Fourier, ilustrado pela Figura 6, declara que uma projeção paralela de uma imagem  $\mu(x, y)$  obtida em um determinado ângulo  $\theta$  fornece um corte (seção) da Transformada de Fourier bidimensional,  $F(u, v)$  no mesmo ângulo  $\theta$ , mas em relação ao eixo- $u$ . Entende-se, portanto, que a Transformada de Fourier unidimensional de  $P_\theta(t)$  fornece valores de  $F(u, v)$  ao longo da reta (feixe de raio-X) no domínio da frequência.

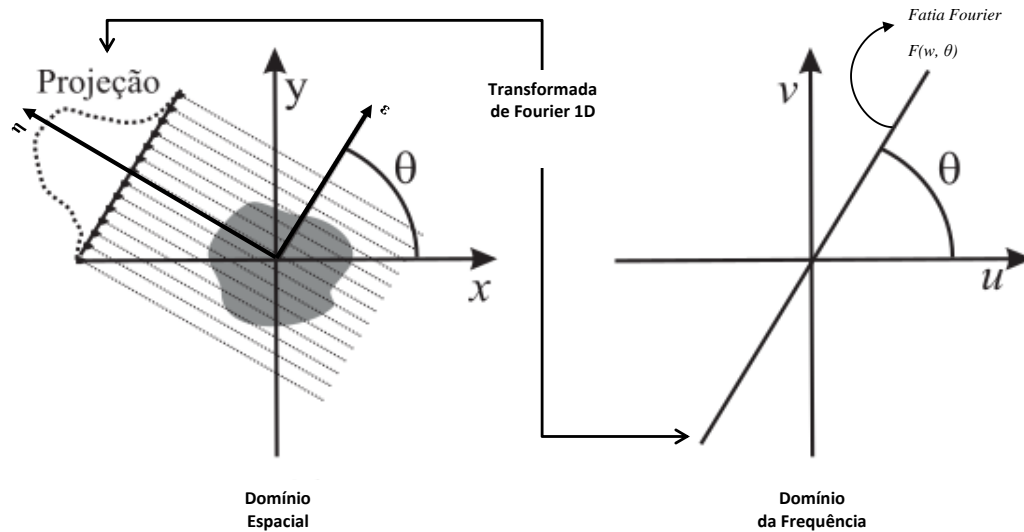


Figura 6 – Teorema das Seções de Fourier

A Transformada de Fourier de um objeto é dado pela Equação 2.11, na qual  $j = \sqrt{-1}$ .

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) e^{-j2\pi(ux+vy)} dx dy \tag{2.11}$$

Da mesma forma define-se uma projeção no ângulo  $\theta$ ,  $P_\theta(t)$ , por meio da sua Transformada de Fourier que é expressa pela Equação 2.12.

$$T_\theta(w) = \int_{-\infty}^{\infty} P_\theta(t) e^{-j2\pi wt} dt \tag{2.12}$$

O teorema pode ficar mais claro se for considerado que o sistema de coordenadas  $(\epsilon, \eta)$  é uma versão rotacionada do sistema de coordenadas  $(x, y)$  e que pode ser expresso pela Equação

## 2.13.

$$\begin{bmatrix} \varepsilon \\ \eta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.13)$$

No sistema de coordenadas  $(\varepsilon, \eta)$  uma projeção pode ser escrita como,

$$P_\theta(t) = \int_{-\infty}^{\infty} \mu(\varepsilon, \eta) d\eta \quad (2.14)$$

A Transformada de Fourier da Equação 2.14 a partir da Equação 2.12 é dada pela Equação 2.15.

$$T_\theta(w) = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} \mu(\varepsilon, \eta) d\eta \right] e^{-j2\pi w t} dt \quad (2.15)$$

Na Figura 6,  $\eta$  representa o eixo- $y$  rotacionado assim como  $\varepsilon$  o eixo- $x$ . Logo é possível reescrever a Equação 2.15 como,

$$\begin{aligned} T_\theta(w) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) dy e^{-j2\pi w t} dx \\ T_\theta(w) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) e^{-j2\pi w t} dx dy \\ T_\theta(w) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) e^{-j2\pi w (x \cos \theta + y \sin \theta)} dx dy \end{aligned} \quad (2.16)$$

Fazendo como que  $u = x \cos \theta$  e  $v = y \sin \theta$ , na Equação 2.16, e observando as Equações 2.11 e 2.12 pode-se verificar a seguinte relação:

$$T_\theta(w) = F(w, \theta) = F(u, v) = F(w \cos \theta, w \sin \theta) \quad (2.17)$$

A relação explica que  $T_\theta(w)$  representa a Transformada de Fourier bidimensional no espaço de frequências  $(u, v)$  e, portanto, pode-se determinar os valores de  $F(u, v)$  em uma linha (feixe). Logo, ao aplicar a Transformada de Fourier unidimensional nas diferentes projeções em diferentes ângulos encontra-se uma aproximação de  $F(u, v)$  que é uma aproximação de  $\mu(x, y)$  no domínio da frequência e a reconstrução, portanto, é a Transformada de Fourier inversa. Um problema associado neste processo é o fato de que o número de projeções é finito o que torna necessário interpolar os dados no domínio da frequência e, isso implica em degradações na imagem.

### 2.3.3 Reconstrução 3D (volumétrica)

A reconstrução 3D (volumétrica) é realizada a partir de um conjunto de informações bidimensionais, o qual é obtido do objeto em estudo, e que possibilita a formação de um novo conjunto de dados que representa a estrutura tridimensional do referido objeto (BRAGA NETO, 1994).

No caso da reconstrução de imagens tomográficas tridimensionais, o conjunto de informações bidimensionais refere-se às fatias obtidas em diferentes alturas da amostra. Uma fatia corresponde à reconstrução bidimensional das diferentes projeções, que foram obtidas em diferentes ângulos e na mesma altura (eixo- $z$ ). Particularmente em relação às amostras agrícolas adquiridas pelos tomógrafos da Embrapa Instrumentação, observa-se a característica de que não há deslocamento de posição durante a aquisição das projeções. Logo, a reconstrução tridimensional é possível por meio da sobreposição das fatias bidimensionais, pois não há movimentação entre elas. A Figura 7 ilustra esse processo de sobreposição e conseqüente reconstrução tridimensional.

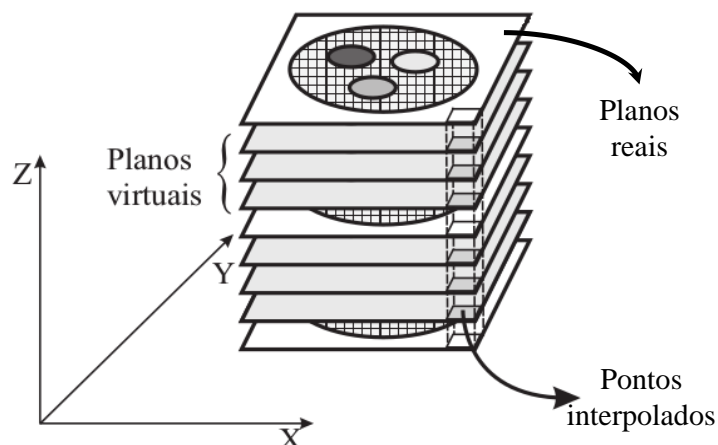


Figura 7 – Sobreposição das fatias bidimensionais para reconstrução tridimensional. Adaptado de Pereira e Cruvinel (2015).

Pode-se observar na Figura 7 que, a partir da sobreposição das fatias bidimensionais (planos reais) é possível gerar os planos virtuais, neste caso, por meio de técnicas de interpolação.

O processo de interpolar consiste em encontrar uma função aproximada a partir de pontos arbitrários os quais não são possíveis de alterar. Portanto, a interpolação consiste em estimar  $f(x)$  para determinar um  $x$ , desde que  $x_1 < x < x_n$ , a partir de um conjunto inicial de dados conhecido de modo a caracterizar a função que os representa. Conceitualmente, o processo de interpolar possui dois estágios: (i) ajustar uma função de interpolação aos pontos fornecidos; (ii) calcular a função de interpolação no ponto  $x$  (RUGGIERO; LOPES, 1996).

Observando a Figura 7, novamente, é possível verificar que os pontos arbitrários, ou *nós interpoladores*, são os pixels das fatias (planos reais) e os planos virtuais são os resultados

da técnica de interpolação. Verifica-se que há diferentes técnicas de interpolação como, por exemplo, a *polinomial* e a *spline* (GREVILLE, 1969; SCHUMAKER, 2007).

Na interpolação polinomial, a função interpoladora  $p(x)$  é um polinômio e pode ser expressa de modo geral pela Equação 2.18.

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n \quad (2.18)$$

onde  $n$  é o número de pontos,  $x$  é um ponto arbitrário,  $n - 1$  é o grau do polinômio e o objetivo consiste em encontrar os coeficientes  $(a_0, a_1, \dots, a_n)$ . A interpolação linear é um caso particular da interpolação polinomial quanto tem-se apenas dois pontos (nós interpoladores) e, portanto, a função interpoladora é um polinômio de 1ª ordem (grau 1).

A *spline* é uma função definida em um intervalo, utilizada para determinar outra função, e é composta por partes que são definidas por funções mais simples em subintervalos que, posteriormente, são reunidas entre os pontos extremos com um grau apropriado de suavidade. Logo, a interpolação *spline* é uma alternativa para trabalhar com grupos de poucos pontos e obter polinômios de grau menor e impor condições para que a função de aproximação seja contínua e tenha derivadas contínuas até uma certa ordem (RUGGIERO; LOPES, 1996). Entre os tipos de *splines*, a cúbica  $Q_3(x)$ , é mais utilizada do que as *splines* linear e quadrática, pois é uma função polinomial por partes contínuas; cada parte,  $q_g(x)$ , é um polinômio de grau 3 em  $[x_{g-1}, x_g]$ ,  $g = 1, 2, \dots, n$ ;  $Q_3(x)$  tem primeira e segunda derivadas contínuas, ou seja, não tem picos e não troca a curvatura nos nós.

No campo das *splines* existe uma particularidade que são as *B-splines* (GREVILLE, 1969). A Equação 2.19 expressa um *spline* utilizando *B-splines*.

$$Q(x) = \sum_{i=0}^{n-1} c_i B_{i,g,t}(x) \quad (2.19)$$

onde  $c_i$  são os coeficientes *splines*,  $g$  é a ordem da *B-spline*,  $t$  são os nós e  $B_{i,g}(x)$  é definida pelas Equações 2.20 e 2.21.

$$B_{i,0}(x) = \begin{cases} 1, & \text{se } t_i \leq x < t_{i+1} \\ 0, & \text{caso contrário} \end{cases} \quad (2.20)$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \quad (2.21)$$

O nome *B-splines* é oriundo de *basis spline* e indica uma função *spline* que possui suporte mínimo em relação a um determinado grau, suavidade e partição do domínio o que possibilita que uma *spline* qualquer possa ser representada como uma combinação linear de *B-splines* desde que ambas possuam o mesmo grau, suavidade e partição do domínio. A Figura 8 apresenta um exemplo da interpolação por *B-splines*.

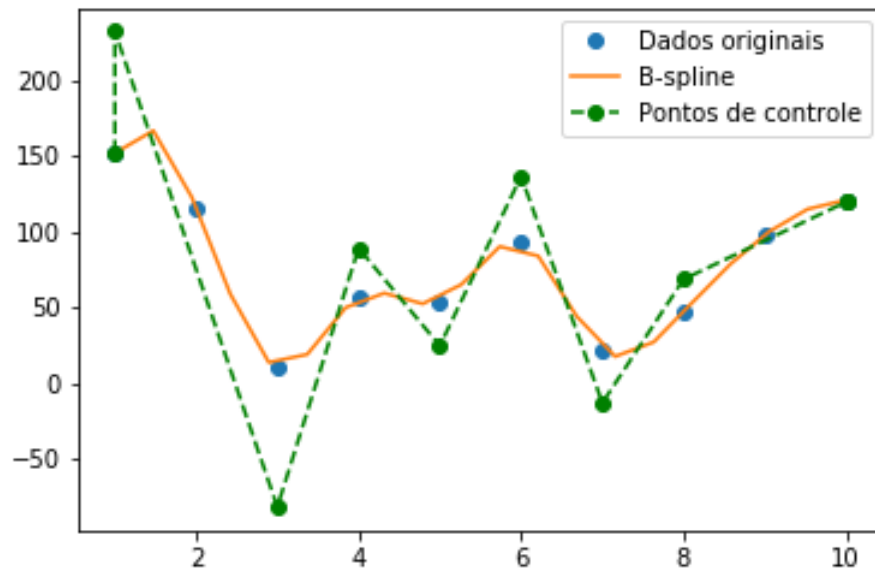


Figura 8 – Exemplo interpolação utilizando *B-splines*.

Há, ainda, a *B-Spline Wavelets*, ou *B-Wavelets*, que considera o uso das *splines* e das *wavelets* como técnica de interpolação. Do ponto de vista matemático, a *wavelet* é uma função capaz de decompor e representar outra função. O processo de decomposição por meio de uma função *wavelet* é chamado de Transformada Wavelet e consiste em decompor a função em análise como combinação linear de funções ortogonais. Portanto, uma *spline wavelet* é uma função *spline* capaz de decompor e representar outras funções o que é possível, por conta das *B-splines*. Desse modo, compreende-se que a *B-Spline Wavelet* corresponde a uma Transformada Wavelet cuja função de decomposição (*wavelet*) é uma *b-spline* (CROWLEY, 1996; UNSER, 1997; KHAN; AHMAD, 2014). De modo a otimizar o processo de cálculo, implementa-se a função denominada *blending* de acordo com a Equação 2.22.

$$B(x) = \begin{cases} \frac{1}{6}(2+x)^3, & -2 \leq x \leq -1 \\ \frac{1}{6}(4-6x^2-2x^3), & -1 \leq x \leq 0 \\ \frac{1}{6}(4-6x^2+2x^3), & 0 \leq x \leq 1 \\ \frac{1}{6}(2-x)^3, & 1 \leq x \leq 2 \\ 0, & 2 \leq |x| \end{cases} \quad (2.22)$$

Além disso, no caso da *B-Wavelet*, são adotados uso de pontos denominados “*fantasmas*” que consistem em valores antes do ponto inicial e depois do ponto final da sequência conhecida para que a curva a ser gerada passe pelos pontos extremos, de forma a melhorar o resultado da interpolação.



# Capítulo 3

## BIG DATA E CIÊNCIA DOS DADOS

---

---

Este capítulo contextualiza os temas Big Data e Ciência dos Dados e a discussão sobre o conceito dos termos associados, além de apresentar características, tecnologias, técnicas e plataformas utilizadas neste campo para o desenvolvimento de sistemas.

### 3.1 Contextualização

Observa-se ao longo das últimas décadas, que o processo de evolução e o surgimento de novas tecnologias tem alterado e influenciado de maneira significativa as relações sociais, as organizações, os negócios e até mesmo a ciência. Tal processo é motivado, em parte, pelo grau cada vez maior, de miniaturização dos componentes eletrônicos o que tem permitido a expansão dos dispositivos computacionais, tornando-os mais portáteis. Em parte, pelas redes de comunicações, particularmente a Internet, que têm possibilitado a crescente interconexão dos diferentes dispositivos bem como ampliado a troca de informações entre eles, caracterizando o que tem sido considerado a Internet das Coisas (IoT).

O Comitê Gestor da Internet no Brasil (CGI.br), apresentou em novembro de 2015, resultados da décima edição da pesquisa TIC Domicílios, a partir de dados coletados em 2014, que apontou que 54% dos domicílios brasileiros localizados na área urbana possuíam acesso à Internet o que representava cerca de 32,9 milhões de residências no Brasil (CGI.BR, 2015). A última edição desta pesquisa, realizada a partir de dados coletados em 2018, apontou um aumento em relação à pesquisa anterior, ou seja, em 2018, o percentual foi de 70% dos domicílios brasileiros, na área urbana, que possuíam acesso à Internet o que representa um aumento de 13,6 milhões de domicílios em quatro anos (CGI.BR, 2019). Logo, as edições desta pesquisa evidenciaram que as tecnologias estão cada vez mais presentes no cotidiano das pessoas e das organizações, tanto no Brasil quanto na América Latina.

A Figura 9 apresenta a proporção de domicílios com computador entre as áreas urbanas e rurais a partir da última edição da pesquisa e contempla o período de 2008 a 2018. É possível observar que a área rural apresentou ritmo mais acelerado de crescimento do número de domicílios do que na área urbana, sobretudo nos últimos três anos.

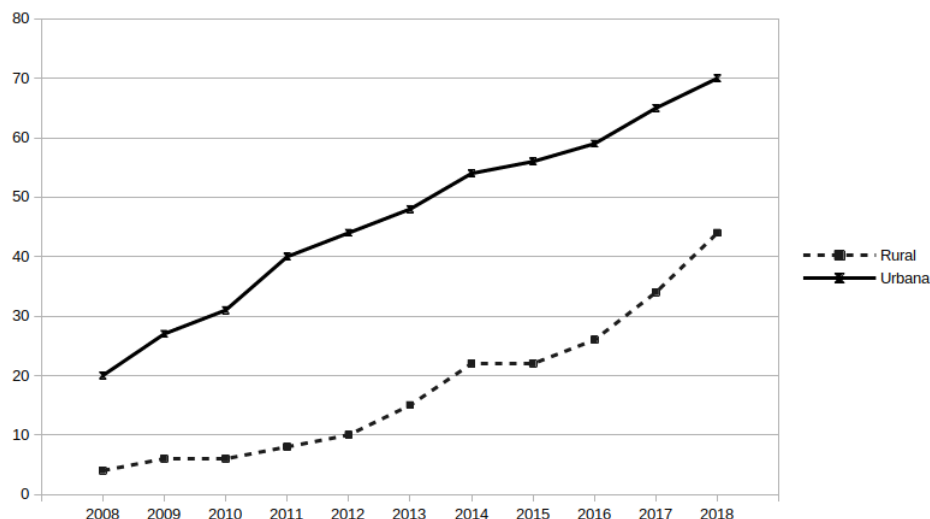


Figura 9 – Proporção de domicílios com computador, por área (2008-2018). O percentual é sobre o total de domicílios. Fonte: adaptado de [CGI.br \(2019\)](#).

A edição da pesquisa, referente ao ano de 2018, destaca que a desigualdade de acesso entre áreas urbanas e rurais tem-se mantido, embora a diferença entre as áreas tem diminuído. O percentual de domicílios com acesso à Internet, por exemplo, na área rural passou de 22% em 2014 para 44% enquanto que na área urbana passou de 50% para 70%, no mesmo período. Com relação aos aparelhos eletrônicos, a pesquisa de 2018 constatou que o telefone celular, ou *smartphone*, é utilizado por 97% dos usuários para acessar a Internet.

O trabalho do CGI.br demonstra parte desse processo de crescimento das tecnologias, particularmente dos computadores e dispositivos móveis, bem como o aumento do acesso à Internet. O trabalho destacou o grande aumento dos recursos tecnológicos sobretudo na área rural, embora ainda exista uma defasagem significativa em relação a área urbana. Não fez parte do escopo da pesquisa, o estudo da evolução do uso de tecnologias em outros segmentos. Entretanto, é possível inferir que essa tendência tem se apresentado nos diversos setores, como o agrícola. No setor agrário, a Agricultura de Precisão é um exemplo que combina agrotecnologias e TIC e é vista como uma alternativa eficiente para aprimorar o processo de tomada de decisões, ampliar o conhecimento agrícola e fornecer soluções viáveis para grandes questões da atualidade, como a produção de alimentos em escala mundial ([QUEIRÓS et al., 2014](#)).

Muitos estudiosos consideram esse contexto, ora apresentado, como a sociedade da informação, pois como apontam [Mayer-Schonberger e Cukier \(2013\)](#),

"meio século depois de os computadores entrarem no meio social, os dados começaram a se acumular a ponto de algo novo e especial começar a acontecer. O mundo não apenas está mais cheio de informação como também a informação está se acumulando com mais rapidez."

Essa quantidade massiva de informação foi propiciada pela diminuição dos dispositivos eletrônicos e dos custos de armazenamento no decorrer do tempo e que tornou os dispositivos

eletrônicos e computacionais mais acessíveis e viáveis (XEXÉO, 2013), permitindo que o volume de informações também aumentasse significativamente. Segundo Sagioglu e Sinanc (2013), até 2003 foram produzidos cerca de 5 ExaBytes<sup>1</sup> pelos humanos e, segundo estimativas dos autores, em 2013 a mesma quantia seria produzida a cada dois dias, aproximadamente, e que poderia chegar a 8 ZetaBytes, ou oito trilhões de gigabytes, em 2015. E a tendência é que a quantidade produzida de dados aumente ainda mais devido, por exemplo, ao aumento de dispositivos conectados à internet. Lee (2017) destaca que em 2016 foram previstos cerca de 6.4 bilhões de dispositivos conectados à rede mundial e que este número pode ultrapassar a marca de 20 bilhões até 2020.

A quantidade expressiva de informação disponível, provocada pela coleta e armazenamento massivo de dados provenientes de enormes quantidades de dispositivos e sensores, tem despertado grande atenção de pesquisadores, organizações e governos. A essa nova fase tecnológica, em que as aplicações fazem uso intensivo de dados, tem sido atribuído o termo **Big Data**, cujo conceito é discutido na próxima seção.

## 3.2 O conceito de Big Data

O termo **Big Data** é oriundo do atual momento tecnológico em que diversas áreas produzem quantidades expressivas de dados. Segundo Ferreira (2008), *dado* é um “elemento de informação, em forma apropriada para armazenamento, processamento ou transmissão por meios automáticos”. Portanto, pode-se considerar como dados: as imagens, as mensagens de texto e de áudio, os arquivos de *log* (registro) gerados por dispositivos eletrônicos e por softwares, informações produzidas por sensores, *streaming* de vídeos, entre outros tipos.

Os autores Mayer-Schonberger e Cukier (2013) relataram que o Google recebia mais de três bilhões de pesquisas, por dia, e as armazenavam como dados da empresa. Em 2009, ela conseguiu prever, antes dos órgãos governamentais, a disseminação da gripe de inverno nos Estados Unidos, gerada pelo vírus H1N1, em âmbito nacional mas também por estados e em regiões específicas. O feito foi possível ao analisar os 50 milhões de termos de busca mais comuns dos americanos que foram comparados com dados de disseminação da gripe entre 2003 e 2008 disponibilizados pelo governo. Esse é um exemplo do potencial do Big Data, que nesse caso, forneceu apoio à área da saúde, porém o potencial dele pode ser observado em outras áreas.

Na tomografia computadorizada, por exemplo, o Big Data contribuiu em um projeto de teste não invasivo no qual o tomógrafo considerado, pelos pesquisadores, o maior do mundo possuía condições de escanear um carro inteiro e produzir matrizes de projeções da ordem de  $10000 \times 15000$  pixels, ou cerca de 5 TB de dados (DITTER; FEY; SCHON; OECKL, 2014). Ainda no campo da tomografia computadorizada, particularmente a *Single Photon Emitted Computed Tomography* (SPECT), Lee e colaboradores (2014) trabalharam na reconstrução 3D

---

<sup>1</sup> 1 ExaByte =  $10^{18}$  bytes

de amostras adquiridas com  $128^2 \times 360 \times 128^3$ , ou seja, tamanho da matriz bidimensional ( $128 \times 128$  pixels), o número de ângulos ( $360^\circ$ ) e o número de voxels ( $128^3$ ).

É importante observar que qualquer tipo de dado pode ser utilizado para extrair novos conhecimentos dependendo da aplicação e do problema em questão. Logo, um modo de compreender Big Data está na maneira como os dados são armazenados e utilizados, o que permite desenvolver uma oportunidade não identificada anteriormente (XEXÉO, 2013). O evento “Soil Big Data”<sup>2</sup>, ocorrido na Austrália em 2015, discutiu como gerar, entender e utilizar dados bem como buscou compreender o advento do Big Data como meio de transformação digital da agricultura com destaque para a questão do uso consciente dos solos. Os principais temas abordados no evento foram: inovação digital na agricultura, Big Data e Internet das Coisas na agricultura, produtividade digital dos solos, segurança e privacidade dos dados na fazenda e o futuro da agricultura.

Verifica-se, portanto, que nos últimos anos tem crescido o interesse pelo assunto e como aplicá-lo nas mais diversas áreas. A Figura 10 apresenta consulta realizada em 18/04/2017 na plataforma *Google Trends* sobre o termo "Big Data". A plataforma utiliza uma porcentagem de todas consultas realizadas no site de busca Google para compilar a popularidade de um termo indicando a tendência em um período e região. A consulta considerou o período de 2004 a 2017 (13 anos) e a tendência no mundo. Pode-se verificar que a partir de 2011 a procura pelo termo “Big Data” aumentou consideravelmente atingindo o pico de popularidade no início de 2017. Além disso, pode-se verificar que no período em que a consulta foi realizada os três países cujo termo Big Data apresentava mais popularidade eram Coréia do Sul, Índia e Taiwan. O Brasil aparecia em sétimo lugar em uma lista de 60 países.

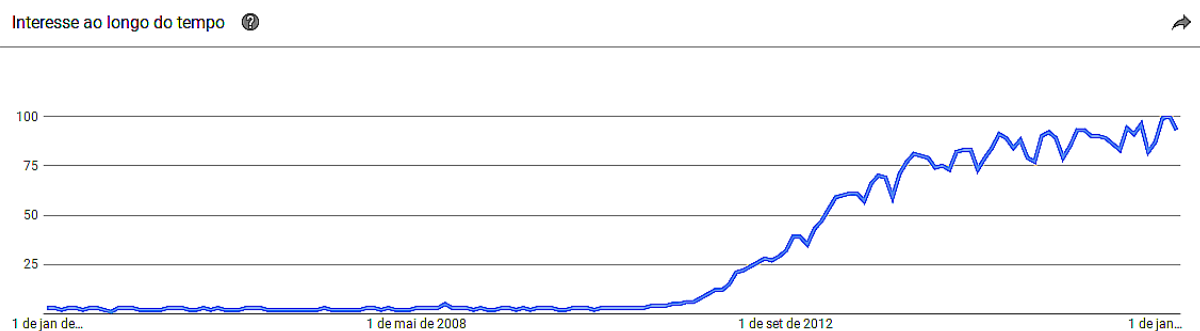


Figura 10 – Tendência de procura sobre o termo "Big Data" no período de 2004 a 2017 no mundo. Fonte: Google Trends. <http://www.google.com/trends>.

Como observado nos exemplos anteriores, é notório que Big Data tem atraído grande atenção de pesquisadores, organizações e governos e tem sido considerado, para alguns autores, como o centro da ciência moderna e dos negócios. O termo geralmente é associado para descrever grandes quantidades de dados em um sociedade dirigida pela informação mas, no entanto, não

<sup>2</sup> Disponível em: <http://soilbigdata.com.au/>. Último acesso em 07/08/2017.

possui um consenso estabelecido acerca da definição de seu conceito (SAGIROGLU; SINANC, 2013; NIST, 2015).

O trabalho de Lajara e colaboradores (2013) analisou, entre outros pontos, o Big Data como apoio à governança e a gestão da informação no ambiente organizacional e, além disso, apresentou 26 diferentes conceitos do termo, dos quais foram selecionados sete para compor o Quadro 1.

Quadro 1 – Seleção de conceitos do termo *Big Data*.

Autor(es)	Conceitos de Big Data
Chen e Zhang (2014)	“são ativos de informação em alto volume, alta velocidade e/ou alta variedade que requer <i>novas formas de processamento</i> para aperfeiçoar a tomada de decisão, a perspicácia e otimizar processos.”
Xexéo (2013)	“descreve um conjunto de problemas e suas soluções tecnológicas em <i>computação aplicada</i> com características que tornam seus dados difíceis de tratar.”
Zhang e Huang (2013)	“é o termo para coleção de conjuntos de <i>dados grandes e complexos</i> que torna-se difícil processá-los utilizando ferramentas de gerenciamento de bancos de dados, manualmente, ou aplicações tradicionais de processamento de dados.”
Lajara, Brinkhues e Maçada (2013 apud ARNOLD, 2012)	“pode ser definido como <i>quantidades massivas de conteúdo</i> armazenado (estruturado ou não) que pode ser facilmente analisado em tempo real (em uma quantidade de tempo razoável para alcançar uma resposta útil).”
Lajara, Brinkhues e Maçada (2013 apud BEGOLI; HOREY, 2012)	“refere-se a <i>prática de coletar e processar conjuntos muito grandes de dados</i> e uso de sistemas associados e algoritmos para análise desses conjuntos massivos de dados.”
Lajara, Brinkhues e Maçada (2013 apud CALLEBAUT, 2012)	“O grande desafio do Big Data não é somente de simples escala e amplitude de novos conjuntos de dados, mas também a <i>crescente complexidade</i> desses.”
Lajara, Brinkhues e Maçada (2013 apud CHEN; CHIANG; STOREY, 2012)	“descreve os conjuntos de dados e <i>técnicas analíticas</i> em aplicações que são tão grandes (de terabytes para exabytes) e complexas (de sensor a dados de mídias sociais) que eles requerem únicas e avançadas tecnologias de armazenamento de dados, administração, análise e visualização.”
Lajara, Brinkhues e Maçada (2013 apud GORDON-MURNANE, 2012)	“não é apenas aumento de quantidade de tipos de dados, é também <i>melhores ferramentas</i> para armazenar, agregar, combinar, analisar e extrair novas idéias.”
<i>continua</i>	

<i>Quadro 1 – continuação</i>	
<b>Autor(es)</b>	<b>Conceitos de Big Data</b>
Ludena e Ahrary (2013 apud MARK; LANEY, 2012)	“é a <i>análise rápida</i> de uma ampla variedade de volume de dados com o intuito de <i>encontrar padrões regulares ou comportamento dos dados</i> que permitirão tomadas de decisões rápidas e mais precisas.”
Lajara, Brinkhues e Maçada (2013 apud SINGH; SINGH, 2012)	“conjunto de dados que continuam a crescer tanto que torna difícil de administrá-los usando conceitos e ferramentas existentes de administração de base de dados.”
Lajara, Brinkhues e Maçada (2013 apud JACOBS, 2009)	“são dados cujo tamanho força-nos a <i>olhar além de métodos comprovados</i> que são predominantes na época.”
<i>fim do Quadro 1</i>	

Ao analisar o Quadro 1, verifica-se que os conceitos elaborados pelos diversos autores apresentam diferenças, embora seja possível identificar aspectos comuns entre eles. As palavras *volume*, *velocidade*, *variedade* e *complexidade* são utilizadas para caracterizar os conjuntos de dados. A busca por aprimorar o *processo de decisão*, de modo *rápido* e *preciso*, indica um objetivo almejado pelo Big Data assim como *obter novos conhecimentos* a partir dos dados. Para tal, uma preocupação que parece comum entre os autores é a necessidade de *desenvolver novas tecnologias e métodos* que superem o cenário tecnológico estabelecido e que, portanto, tenham condições de lidar com o aumento massivo de dados e às várias oportunidades provenientes do Big Data. Outro aspecto comum, possível de observar no Quadro 1, é o de atentar para que as soluções sejam capazes de se adaptarem ao aumento de dados ao longo do tempo, isto é, que os sistemas Big Data sejam escaláveis.

Ainda que o conceito de Big Data não esteja consolidado é possível deduzir que o termo possui estreita relação com a dimensão de dados, métodos e tecnologias associadas para se realizar análises e extrair informações. Contudo, o relatório do *National Institute of Standards and Technology*, NIST (2015), destaca que a falta de consenso sobre questões importantes podem dificultar o progresso do Big Data e entre elas, incluem:

1. Quais atributos definem as soluções de Big Data?
2. Quão diferente é Big Data dos ambientes de dados tradicionais?
3. Quais são as características essenciais em ambientes Big Data?
4. Como estes ambientes se integram com as arquiteturas atuais?

Portanto, se faz necessário discutir e avaliar características, bem como a organização desse novo paradigma, que é o Big Data. Nesse sentido, outros termos presentes na literatura são apresentados e discutidos a fim de se buscar ampliar a compreensão sobre o assunto.

### 3.3 Características e desafios de Big Data

Como observado no Quadro 1, da seção 3.2, diversos autores utilizam os termos *volume*, *velocidade* e *variedade* para caracterizar o Big Data e cuja Figura 11 ilustra como eles se relacionam (CHEN; ZHANG, 2014; ELGENDY; ELRAGAL, 2014; SAGIROGLU; SINANC, 2013; ZHANG; HUANG, 2013).

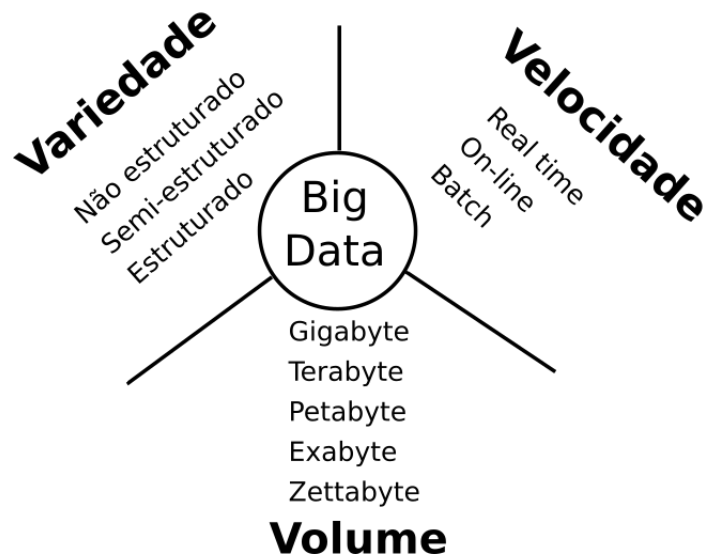


Figura 11 – Características do Big Data: volume, velocidade e variedade. Adaptado de Sagioglu e Sinanc (2013), Zhang e Huang (2013).

A primeira característica é o **volume** dos dados. Cabe observar que o relatório do NIST (2015) considera que a quantidade de dados gerada e armazenada tem crescido a uma taxa superior a Lei de Moore, pois o volume de dados tem mais do que dobrado a cada 18 meses. Portanto, tal característica indica a capacidade de processar grandes conjuntos de dados que podem variar da ordem de Gigabytes a Petabytes, ou superior. Deve-se considerar, ainda, a capacidade de lidar com dados distribuídos, ou seja, a habilidade de tratar repositórios de dados espalhados pelo ambiente de uma aplicação como sendo um único conjunto de dados.

A característica **variedade** está relacionada ao tipo ou complexidade dos dados, que podem ser: *estruturados*, *semi-estruturados* ou *não estruturados*. Tradicionalmente os sistemas utilizam dados estruturados a partir do modelo relacional, geralmente organizados em tabelas e implementados pelos Sistemas Gerenciadores de Banco de Dados (SGDB), a exemplo do MySQL e PostgreSQL, e que utilizam a *Structured Query Language* (SQL) para realizar consultas a fim de gerenciar e extrair informações das bases de dados.

No entanto, verifica-se nos últimos anos a utilização de grandes quantidades de dados não estruturados, tais como páginas web, imagens, vídeos, relações entre informações, mensagens de texto, entre outros tipos de dados. Adiciona-se também o uso de dados semi-estruturados, como: o e-mail que contém campos estruturados (remetente, destinatário, assunto, etc) e campo



não estruturado (corpo do e-mail); e os arquivos de *log* gerados pelos sistemas e que possuem registros pré estabelecidos, entretanto sem uma estrutura bem definida (KHAN; UDDIN; GUPTA, 2014; SAGIROGLU; SINANC, 2013; ZHANG; HUANG, 2013).

A terceira característica, **velocidade**, diz respeito à habilidade de processar diferentes tipos de informações, geralmente de volume imenso, em tempo hábil. Khan e colaboradores (2014) apontaram que a velocidade do fluxo de aquisição de dados é importante, assim como a rápida transmissão dos dados para grandes repositórios para processamento e análise. É importante observar que para cada aplicação pode-se determinar velocidades de aquisição, processamento e análise diferentes. Em algumas, o tempo não é crítico e os dados podem ser trabalhados em lotes, ou *batch*, enquanto que em outras esse tempo pode ser próximo do real (*on-line*) ou, nos casos críticos, os dados precisam ser trabalhados em tempo real.

Adicionalmente outras propriedades podem ser utilizadas para caracterizar Big Data, são elas: *veracidade*, *valor*, *volatilidade* (NIST, 2015; CHEN; ZHANG, 2014; KHAN; UDDIN; GUPTA, 2014). O dicionário Michaelis descreve **veracidade** como sendo qualidade ou atributo do que é verdadeiro<sup>3</sup>. Logo, entende-se que essa característica lida com os aspectos, incerteza e falta de confiabilidade, provenientes dos dados. Tais aspectos surgem devido a incompletude, a inacurácia, a inconsistência e a subjetividade dos dados (LEE, 2017). O **valor** consiste em considerar todos os dados, mesmo os considerados primários (ou crus), como relevantes de modo que possam ser reutilizados e gerar significado para a organização. A **volatilidade** preocupa-se com a tendência de que as estruturas de dados do sistema se alterem ao longo do tempo.

Ao observar esse conjunto de características nota-se que Big Data constitui uma mudança estrutural na arquitetura das aplicações o que se traduz em diversos desafios que residem, principalmente, na aquisição, no armazenamento, na busca, no compartilhamento, na análise e na visualização dos dados.

Um exemplo de desafio para a aquisição e armazenamento de dados, é a arquitetura dos computadores que, por décadas, possui processadores com alto desempenho enquanto os recursos de entrada e saída são limitados (*CPU-heavy, I/O poor*). Embora o desempenho dos processadores e dos dispositivos de entrada e saída dobrem, em média, a cada 18 meses o fato é que esse último não obteve avanços significativos em termos de velocidade. Por outro lado, a quantidade de informação cresce exponencialmente e, portanto, o desequilíbrio existente compromete as aplicações Big Data, especialmente àquelas que exigem análises e processamentos em tempo real (CHEN; ZHANG, 2014).

Outro desafio é o de lidar com base de dados eficientemente o que implica desenvolver e utilizar novos modelos, uma vez que o modelo relacional tem apresentado limitações às necessidades atuais para análises e operações em conjuntos de dados cada vez maiores e não estruturados. Contudo, cabe destacar que isso não significa que o modelo relacional tornou-se

<sup>3</sup> Fonte: <<http://michaelis.uol.com.br>>. Acesso em 1 de junho de 2017.



obsoleto. Como ponderaram [Sadalage e Fowler \(2015\)](#), os bancos de dados relacionais ainda são ferramentas poderosas que deverão ser utilizadas por décadas. O que os autores observaram foi um cenário com várias opções de armazenamento cuja tendência é que as aplicações passem a utilizar armazenamento híbrido envolvendo diferentes tecnologias e explorando o potencial de cada uma delas.

A abordagem clássica de gerenciamento de dados estruturados incluía duas partes, um esquema para armazenamento e um banco de dados relacional para recuperação da informação. Para as grandes bases utilizavam-se dos *data warehouse* e *data marts*. Hoje, bancos de dados NoSQL são empregados para gerenciamento de dados distribuídos e volumosos, pois são livres de esquema e possuem diversas abordagens, por exemplo: o modelo chave-valor para acesso a dados estruturados; o modelo de banco de dados orientados a documentos dada a relevância da análise documental em diversas aplicações; modelo orientado a objetos; modelo orientado a grafos que considera a importância da relação entre os dados.

Outro aspecto a ser considerado é quanto a transmissão de dados. Sabe-se que a capacidade da largura de banda de uma rede, muitas vezes, é o gargalo em sistemas distribuídos e, portanto, essa é mais uma questão a ser observada no desenvolvimento de projetos desta natureza.

O projeto de sistemas Big Data exige arquitetura paralela e escalável, voltada para aplicações de processamento intensivo de dados. O relatório do [NIST \(2015\)](#) apresentou dois principais métodos para tornar uma arquitetura escalável: o *vertical scaling* e o *horizontal scaling*.

O *vertical scaling* implica aumentar os parâmetros do sistemas tais como velocidade de processamento, capacidade de memória e de armazenamento. Em geral é limitada pelas capacidades máximas dos equipamentos o que torna o custo elevado quando se é necessário trocar algum dos equipamentos por outro mais sofisticado. Além disso, pode demandar mais tempo na implementação da solução.

O *horizontal scaling* é um método alternativo que consiste em utilizar um *cluster* de recursos individuais integrados para atuar como um único sistema. Como destaca o relatório do [NIST \(2015\)](#),

"o paradigma Big Data consiste na distribuição dos sistemas de dados, horizontalmente acoplados, com recursos independentes, para alcançar a escalabilidade necessária para o processamento eficiente de base de dados extensas."

Os sistemas paralelizados, fundamentados no *horizontal scaling*, iniciaram-se com as aplicações de simulações que exigiam processamento paralelo massivo (MPP, *sigla em inglês*) e, desse modo, tais sistemas precisam considerar problemas como passagem de mensagens, movimento de dados, latência, balanceamento de carga, entre outros.

Entende-se, portanto, que um sistema Big Data deve se preocupar com a escalabilidade,

o paralelismo, a transmissão e o armazenamento a um bom custo efetivo para o gerenciamento dos dados. Por outro lado, tão importante quanto a arquitetura do sistema é a análise dos dados que recebe o nome de **Data Science**, ou Ciência dos Dados, que é assunto da próxima seção.

### 3.4 A ciência dos dados

O termo *Data Science*, ou Ciência dos Dados, refere-se a condução da análise de dados como uma ciência que aprende diretamente dos dados e, para isso, combina tecnologias e métodos de diversas áreas, especialmente às relacionadas a Ciência da Computação e Estatística (MORAES; MARTÍNEZ, 2015; NIST, 2015; SLAVAKIS; GIANNAKIS; MATEOS, 2014).

Os autores Chen e Zhang (2014) observaram que na literatura há quem considere o *Data-intensive Science*, ou simplesmente *Data Science*, como o quarto paradigma científico depois das ciências empírica, teórica e computacional. Eles argumentaram que as mudanças foram significativas por conta do aumento das aplicações de uso intensivo de dados e que, por isso, tem alterado e influenciado o “mundo da ciência” sendo, portanto, considerado o quarto paradigma da ciência propício para novas descobertas.

Durante a condução da análise de dados, dois métodos podem ser empregados. O primeiro consiste na coleta de dados, seguida da constante análise e sem hipóteses preconcebidas e é denominado, às vezes, como exploração de dados. O segundo consiste na formulação da hipótese *a priori* e, na sequência, realiza-se a coleta de informações para, depois, avaliá-la a fim de que ela seja confirmada ou refutada. Em ambos os métodos, as conclusões são fundamentadas nos dados e, eventualmente, o resultado final pode assinalar a necessidade de reformular a hipótese. Desse modo, a ciência dos dados é considerada a extração de conhecimento ativo diretamente dos dados brutos por meio da descoberta, formulação e teste de hipóteses.

Nesse contexto, o termo *analytics*<sup>4</sup> pode ser compreendido como uma referência à descoberta de padrões significativos que é uma das etapas do ciclo de vida dos dados. Outro termo, comumente encontrado na literatura, é *Big Data Analytics* que pode ser definido como o uso de técnicas analíticas avançadas em aplicações Big Data (EMANI; CULLOT; NICOLLE, 2015). A princípio, o termo pode sugerir ser idêntico a ciência dos dados, no entanto há trabalhos que discutem a definição e as diferenças sutis entre eles (AGARWAL; DHAR, 2014; MORAES; MARTÍNEZ, 2015; LEE, 2017).

Por ciclo de vida dos dados, também denominado processo de descoberta do conhecimento, entende-se como o conjunto de etapas que transformam dados brutos em conhecimento ativo, como ilustrado na Figura 12, e que consiste em: coleta de dados; preparação da informação (limpeza); análise de padrões; visualização e tomada de decisão.

A primeira etapa do processo, *a coleta*, inclui as tarefas de captura e armazenamento

<sup>4</sup> O termo *analytics* não possui uma tradução literal e pode ser entendido como analítico, relacionado a ciência de análise e estatística ou, ainda, análise computacional sistemática de dados.

de dados. Um aspecto relacionado a essas tarefas é a localização dos dados que podem estar espalhados em diferentes repositórios. Outro aspecto é a complexidade dos dados adquiridos, lembrando que uma aplicação pode lidar com dados estruturados, semi-estruturados e não-estruturados. Portanto, nesta etapa, exige-se análises no sentido de obter as melhores estratégias para trabalhar em um ambiente distribuído bem como análises das diferentes abordagens de gerenciamento de dados a exemplo dos modelos relacional, chave-valor ou orientado a grafos.

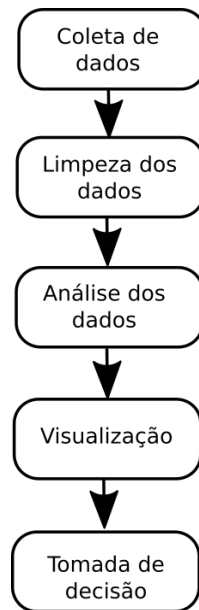


Figura 12 – Processo de descoberta do conhecimento. Adaptado de [Chen e Zhang \(2014\)](#).

Uma vez coletados os dados, a segunda etapa, denominada *limpeza*, é responsável por preparar os dados para a análise. Durante a etapa de coleta os dados podem ser obtidos na forma “bruta” ou “processada”, por exemplo: um arquivo obtido diretamente do sensor, em que cada linha apresentasse um número qualquer, poderia ser considerado um *dado bruto*; por outro lado, o conteúdo do mesmo arquivo poderia ser obtido já *processado* indicando, para cada linha do arquivo, o valor de um parâmetro do sistema como temperatura, peso, etc. Portanto, esta etapa consiste em incluir metadados, integrar ou desmembrar informações buscando a melhor representação dos dados para a etapa de análise.

Na sequência, a *análise* representa um aspecto importante, senão o mais importante, da ciência dos dados que é a extração do conhecimento. Nesta etapa diferentes técnicas são empregadas para se conseguir obter respostas para as questões estabelecidas ou descobrir novas informações que não foram previstas no início do processo. Uma preocupação é que os algoritmos e técnicas utilizadas consigam prover o resultado em um tempo de resposta considerado adequado dentro do contexto da aplicação. Portanto, muitas vezes acrescenta-se à esta etapa outras questões como otimização e paralelismo dos algoritmos de análise a fim de melhorar os tempos de respostas.

A *visualização* dos dados, penúltima etapa do processo, tem como principal objetivo

representar o conhecimento de modo mais intuitivo, conciso e eficiente, de modo a transmiti-lo com clareza. O grande desafio desta etapa reside no fato de lidar com alta dimensionalidade e alto volume dos dados, portanto é necessário utilizar, ou desenvolver, novas técnicas gráficas e de visualização que permitam representar melhor a informação.

Finalmente, a *tomada de decisão* indica o momento em que as hipóteses são validadas, refutadas ou descobertas. É o momento em que os especialistas e *stakeholders*<sup>5</sup> podem determinar rever, ou alterar as análises ou, ainda, a partir dos resultados obtidos e visualizados podem deliberar sobre assuntos pertinentes à organização que utiliza o sistema de Big Data.

Verifica-se que, inúmeras técnicas e tecnologias dão suporte à ciência dos dados. Muitas delas estão presentes há muito tempo, outras têm surgido para atender e adaptar-se às demandas atuais. Na próxima seção são apresentadas as principais técnicas e tecnologias utilizadas em ciência dos dados e, também em Big Data.

### 3.5 Técnicas e tecnologias

Novas técnicas e tecnologias têm sido desenvolvidas, ou adaptadas, no intuito de extrair novos conhecimentos da imensa quantidade de informação disponível. A Figura 13 ilustra a relação entre as diferentes disciplinas, aqui consideradas técnicas de Ciência dos Dados.

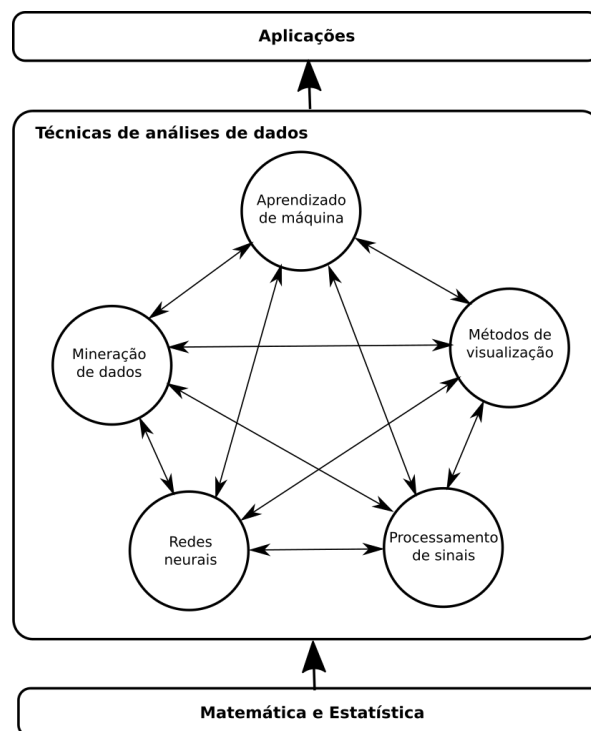


Figura 13 – Técnicas de Ciência dos Dados. Adaptado de [Chen e Zhang \(2014\)](#).

<sup>5</sup> Os *stakeholders* são elementos essenciais ao planejamento estratégico de negócios. Compreende, de modo mais amplo, todos os envolvidos (partes interessadas) em um processo, negócio ou organização. Fonte: <<https://pt.wikipedia.org/wiki/Stakeholder>>. Acesso em 26.02.2016.

Na Figura 13 podem ser observadas técnicas de análises de dados organizadas em cinco grandes áreas: aprendizado de máquina, mineração de dados, redes neurais, processamento de sinais e métodos de visualização. Destaca-se o fato que elas se inter-relacionam e são aplicadas dependendo do contexto da aplicação. Adicionalmente, no bloco "Aplicações", que aparece na figura, é possível considerar setores como: biomedicina, biotecnologia, astronomia, economia, agricultura, entre outros. Para viabilizar as aplicações de Big Data e Ciência dos Dados diversas tecnologias têm sido desenvolvidas e empregadas para tratar dos aspectos matemáticos e das técnicas de análises conforme apresentado no Quadro 2.

Quadro 2 – Tecnologias de Big Data e Data Science.

<b>Tecnologia</b>	<b>Uso específico</b>
Apache Hadoop	Plataforma e infraestrutura
Dryad	Plataforma e infraestrutura
Apache Mahout	Aprendizado de máquina voltado para negócios
Jaspersoft BI Suite	Business intelligence
Pentaho Business Analytics	Plataforma de análise de negócios
Skytree server	Aprendizado de máquina e análise avançada
Tableau	Visualização de dados, análise de negócios
Karmashpere studio and analyst	Big Data Workspace
Talend open studio	Gerenciamento de dados e integração de aplicações
Storm	Sistema de computação em tempo real
S4	Processamento contínuo de fluxo de dados
SQLStream s-Server	Sensores e aplicações telemáticas
Apache Spark	Processamento em cluster de dados em larga escala
Splunk	Coleta e proteção de dados de máquina
Apache Kafka	Sistema distribuído de mensagens
Apache Drill	Sistema distribuído de análise interativa
OpenRefine	Limpeza e transformação de dados
Cloudera Impala	Mecanismo de busca em ambiente distribuído
Elastic search	Servidor de busca em ambiente distribuído
Microsoft Azure	Plataforma e infraestrutura
Bancos NoSQL	Utilizam dados sem esquema e, geralmente, são executados em clusters
Plot.ly	Plataforma para visualização de dados
Google Chart	Criação de gráficos
Blockspring	Integração de dados
Caffe	Framework para Deep Learning
Sci-kit learn	Biblioteca, escrita em Python, para aprendizado de máquina
TensorFlow	Software open-source para aprendizado de máquina

Fonte: adaptado de [Chen e Zhang \(2014\)](#).

O Quadro 2 apresenta um conjunto de tecnologias cujo intuito é fornecer um panorama geral das principais ferramentas utilizadas e não o de esgotar o assunto a respeito das tecnologias disponíveis na atualidade. A segunda coluna do quadro procura descrever o uso de cada tecnologia.

Observa-se que estas técnicas e tecnologias permeiam diferentes disciplinas, a exemplo da ciência da computação, matemática, estatística, processamento de sinais entre outras. As

ferramentas matemáticas incluem, além dos fundamentos matemáticos, a estatística e os métodos de otimização. A estatística trata dos meios para coletar, organizar e interpretar dados, no entanto, no âmbito do Big Data há o desafio de adaptar as técnicas tradicionais, ou criar novos métodos. Uma abordagem é a de desenvolver algoritmos mais eficientes buscando a escalabilidade e o paralelismo das técnicas estatísticas, por isso, as sub-áreas de *computação estatística* e *aprendizado estatístico* tem despertado grande interesse dos pesquisadores. Os métodos de otimização, por sua vez, são aplicados para solucionar inúmeros problemas em diferentes áreas, contudo frequentemente possuem alta complexidade e consumo de memória o que torna sua utilização um desafio, até mesmo, em aplicações de Big Data. Novamente, as técnicas de paralelismo além de técnicas de redução da dimensionalidade dos dados são abordagens alternativas para problemas de otimização.

Além das tecnologias apresentadas, as linguagens de programação Python, Java e R são, geralmente, as mais citadas e empregadas no desenvolvimento de aplicações e, particularmente, essa última é voltada para a computação estatística.

Na próxima seção são apresentados princípios para o desenvolvimento de sistemas Big Data bem como exemplos de aplicações.

## 3.6 Sistemas Big Data

Os projetos de sistemas Big Data são de alta complexidade, pois em geral demandam intensivo processamento de dados além de exigir eficiência e escalabilidade. [Chen e Zhang \(2014\)](#) propõem sete princípios para o desenvolvimento de tais sistemas que são relacionados a seguir:

**Princípio 1.** *Boas arquiteturas e frameworks são necessários.* O uso das arquiteturas e tecnologias tradicionais, a exemplo dos bancos de dados relacionais, podem não ser suficientes para lidar com as necessidades que os sistemas Big Data exigem, por isso empregar arquiteturas de processamento paralelo e distribuído bem como estratégias diferentes como NoSQL são boas alternativas.

**Princípio 2.** *Suportar uma variedade de métodos analíticos.* O conjunto de tarefas complexas exige que diversos métodos de diferentes áreas sejam, muitas vezes, articulados de modo a preparar a solução mais apropriada e, portanto, é importante que os sistemas Big Data estejam preparados para uma grande variedade de métodos analíticos.

**Princípio 3.** *Não há tamanho que se ajuste a todas aplicações.* As ferramentas possuem limitações, portanto saber usá-las apropriadamente pode significar benefícios para lidar com a quantidade de informação que tem crescido constantemente.

**Princípio 4.** *Leve a análise aos dados.* Esse princípio considera a necessidade de levar a análise até os dados em problemas de computação intensiva. O princípio sugere que é inviável transportar a grande quantidade de dados, que geralmente é processada, até as unidades de processamento para se realizar as análises.

**Princípio 5.** *O processamento em memória deve ter condições de ser distribuído.* Considerando o princípio anterior, o processamento em memória dos sistemas Big Data deve ter condições de ser distribuído.

**Princípio 6.** *Armazenamento de dados em memória deve ter condições de ser distribuído.* Os dados gerados e acumulados em centros de dados também precisam ser particionados para análise em memória.

**Princípio 7.** *Coordenação é necessária entre unidades de dados e de processamento.* Esse princípio tem por objetivo aprimorar a escalabilidade, a eficiência e a tolerância a falhas em sistemas Big Data.

Os problemas de Big Data estão associados as mais diversas áreas e, portanto, pode-se considerar que é possível desenvolver sistemas desta natureza para qualquer campo. Geralmente as aplicações voltadas para o comércio, especialmente, o eletrônico e para as redes sociais são as mais discutidas. Além disso, é possível encontrar aplicações voltadas à outras áreas, como gestão pública e agricultura de precisão. Outro exemplo é a pesquisa sobre tomografia computadorizada em larga-escala realizada no *Argonne National Laboratory*<sup>6</sup>.

Ainda no campo da pesquisa científica o interesse por obter conhecimento a partir de dados produzidos de simulações em larga escala pode ser considerado um problema de Big Data e, neste caso, possui uma estreita relação com *e-Science*. O termo está relacionado a ciência computacionalmente intensiva que geralmente é implementada em sistemas distribuídos e, além disso, é um conceito amplo que engloba outras subáreas (CHEN; ZHANG, 2014).

Nas próximas seções são discutidos recursos que viabilizam a construção de sistemas Big Data. Primeiro, será abordada a programação paralela, metodologia PCAM, modelo de programação *MapReduce*, na sequência, as plataformas *Hadoop* e *Spark* e, por fim, computação em nuvem.

### 3.7 Programação Paralela

A programação paralela pode ser empregada visando reduzir o tempo necessário para encontrar a solução de um problema ou para tratar questões mais complexas e que envolvam quantidades maiores de dados. Além disso, a programação paralela pode otimizar o uso de recursos computacionais disponíveis, tais como memória e processadores.

<sup>6</sup> Mais informações em: <[http://www.aps.anl.gov/Xray\\_Science\\_Division/Big-Data/Overview.html](http://www.aps.anl.gov/Xray_Science_Division/Big-Data/Overview.html)>. Acesso em: 18/04/2017.



Por outro lado, a construção de algoritmos paralelos não é uma tarefa fácil como observaram os autores [Mattson, Sanders e Massingill \(2004\)](#) no livro "*Patterns for Parallel Programming*". Os autores enfatizaram a necessidade de explorar a concorrência, ou seja, decompor o problema em subproblemas, gerenciar a dependência entre as tarefas bem como definir e utilizar padrões como alguns dos principais desafios ao elaborar programas paralelos.

Nesta mesma linha de raciocínio, [Foster \(1995\)](#) já havia proposto a metodologia PCAM (*Partitioning, Communication, Agglomeration, Mapping*) a qual é apresentada a seguir.

### 3.7.1 Metodologia PCAM

A metodologia PCAM (*Partitioning, Communication, Agglomeration, Mapping*) foi proposta com o intuito de auxiliar os desenvolvedores a tratarem problemas por meio de uma abordagem de programação paralela que minimizasse o retrabalho e que possibilitasse identificar situações que seriam observadas somente após a construção do sistema ([FOSTER, 1995](#); [PEREIRA; CRUVINEL, 2015](#)). A Figura 14 ilustra as quatro etapas da metodologia PCAM para elaboração de algoritmos e sistemas paralelos.

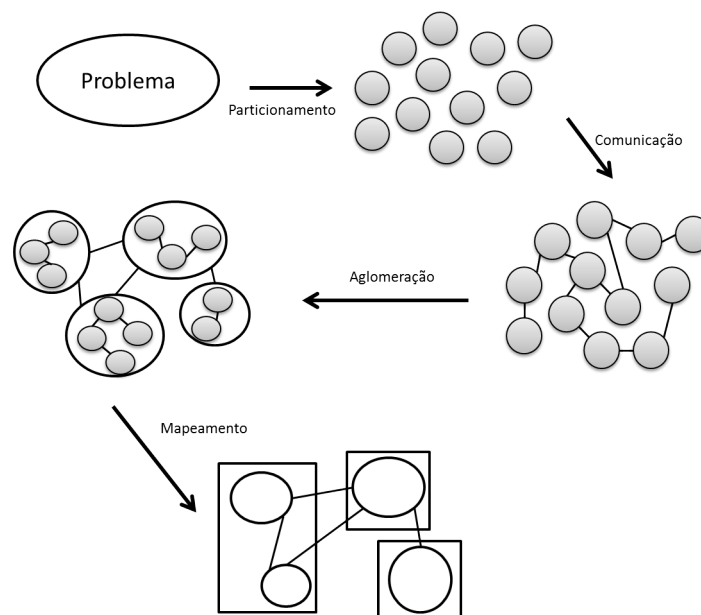


Figura 14 – Metodologia PCAM para elaboração de algoritmos paralelos.

A primeira etapa da metodologia PCAM consiste em particionar (*partitioning*) o problema em pequenas tarefas buscando aproveitar oportunidades de paralelismo, ou seja, não há uma preocupação com o número de processadores nesta etapa. O intuito é obter uma boa partição do problema seja baseada na decomposição do domínio (dados do problema) ou na decomposição funcional, que consiste em separar o problema em tarefas que possuam pouca dependência entre si e que necessitem de pouca comunicação ou replicação de dados. Na segunda etapa, há uma preocupação com a comunicação (*communication*) necessária para coordenar o fluxo de informações e de execução das tarefas que compõe o algoritmo bem como com as



estruturas e algoritmos associados. Uma vez definidas as tarefas e as estruturas de comunicação, a terceira etapa irá se preocupar com a aglomeração (*agglomeration*). O objetivo desta etapa é otimizar custos de computação a fim de melhorar a performance mantendo um equilíbrio entre granularidade e flexibilidade da aplicação. Por exemplo, tarefas podem ser agrupadas em uma tarefa maior para reduzir tempo de comunicação e transferência de dados. Nesta etapa o número de processadores passa a ser considerada na elaboração do algoritmo. Por fim, a última etapa consiste no mapeamento (*mapping*) das tarefas para os recursos computacionais disponíveis. Portanto nesta etapa, pode se adotar a estratégia de alocar tarefas concorrentes em processadores diferentes ao passo que tarefas que se comunicam mais podem ser alocadas no mesmo processador a fim de reduzir custos de comunicação e de aumentar a localidade.

Vale ressaltar que durante a concepção da paralelização de sistemas é possível determinar o nível da granularidade<sup>7</sup>, que pode ser grossa, média ou fina. O nível é determinado em função do grau de agrupamento de tarefas que compõem um determinado processo computacional (GRAMA; KARYPIS; KUMAR, 2003). Portanto, a quantidade e o tamanho das tarefas em que um problema é decomposto determina o nível da granularidade. A granularidade fina é caracterizada quando um problema é decomposto em um grande número de pequenas tarefas, ao passo que a granularidade grossa é caracterizada pela decomposição em um pequeno número de grandes tarefas. A granularidade média busca o equilíbrio entre os níveis fino e grosso.

### 3.7.2 Modelo de programação *MapReduce*

O aumento do volume de dados e de informações exigiu que os sistemas computacionais ampliassem a quantidade de serviços disponibilizados e em janelas de tempo cada vez menores. Esse cenário, considerado por alguns como “dilúvio de dados”, fez com que as aplicações computacionais explorassem, ainda mais, os paradigmas de programação paralela e processamento distribuído. Por outro lado, programar para ambientes distribuídos é complexo, pois existem diversas questões que precisam ser tratadas, tais como concorrência, tolerância a falhas, distribuição de dados e balanceamento de carga (GRAMA; KARYPIS; KUMAR, 2003).

Visando simplificar esse processo de desenvolvimento para ambientes distribuídos, inicialmente para as aplicações da empresa Google Inc., foi proposto em 2004 o modelo de programação paralela para processamento distribuído denominado *MapReduce* (DEAN; GHEMAWAT, 2008; WHITE, 2012; ROCHA, 2013). Tal modelo é voltado para computação paralela, pois distribui o processamento de dados entre os computadores organizados em cluster e possibilita abstrair dos desenvolvedores, a complexidade da paralelização permitindo que eles se dediquem à elaboração das aplicações.

O modelo *MapReduce* deriva da concepção de dividir uma tarefa em outras menores (*map*) e, posteriormente, reuni-las e consolidá-las (*reduce*). Esse novo paradigma de programação, inspirado pelas linguagens de programação funcionais, mostrou-se apropriado para lidar com

<sup>7</sup> A granularidade não é uma medida, mas uma ideia qualitativa da razão entre computação e comunicação.

problemas que podem ser particionados em subproblemas de modo que várias funções *map* possam ser executadas paralelamente com conjuntos de dados diferentes para que, na sequência, operações de agregação (*reduce*) sejam aplicadas aos resultados gerados pelas funções (LIN; DYER, 2010).

Aplicações cujas tarefas são do tipo *Bag-of-tasks* mostram-se mais apropriadas para este modelo, pois tais tarefas não possuem dependências entre si (SILVA; SENGER, 2009). A Figura 15 apresenta o diagrama esquemático do modelo *MapReduce* aplicado a um conjunto de computadores organizados em *cluster*.

Na Figura 15, os arquivos de entrada, que compõem o conjunto de dados, são divididos em várias partes. Essas partes, que são blocos de dados, são mapeadas para os “Nós trabalhadores” que executam as tarefas *map* sobre elas. Os resultados oriundos das tarefas de mapeamento produzem arquivos intermediários. Tais arquivos são enviados para “Nós trabalhadores” responsáveis por executar as tarefas de consolidação (*reduce*) e gerar os arquivos de saída que correspondem ao resultado final do processo.

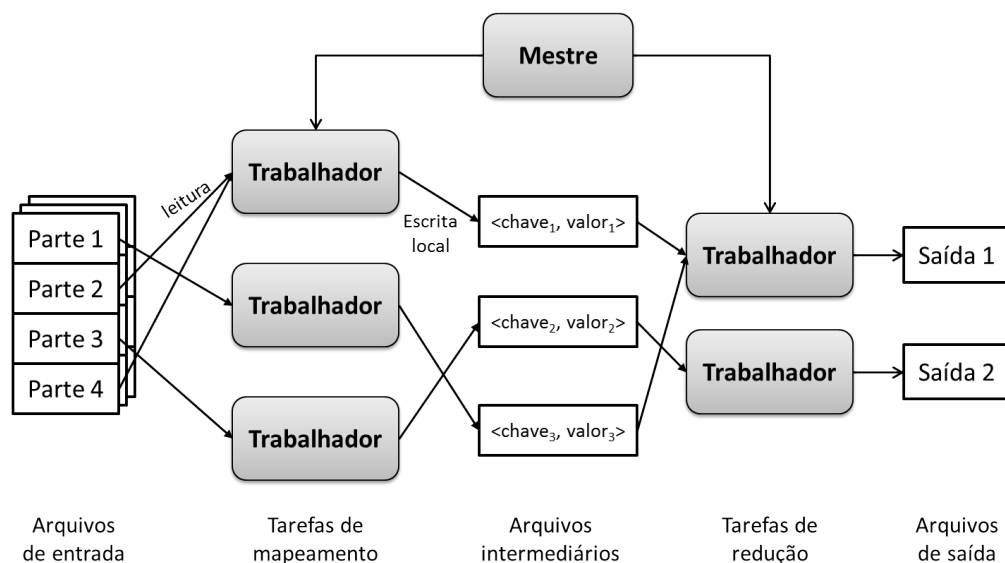


Figura 15 – Diagrama esquemático do modelo de programação *MapReduce*.

Além disso, três pontos precisam ser destacados na Figura 15. O primeiro refere-se à arquitetura *master-workers* do *cluster*, no qual um Nó mestre (*master*) gerencia os demais Nós trabalhadores (*workers*). O Nó mestre é responsável, entre outras coisas, a distribuir as tarefas *map* e *reduce* entre os demais Nós. O segundo ponto é que o modelo *MapReduce* utiliza um sistema de arquivos distribuído. O terceiro ponto reside no fato de que o conjunto de dados de entrada é mapeado em blocos de dados (*chunks*) formando uma coleção de tuplas *<chave, valor>* cujas tuplas que possuem a mesma *chave* são, posteriormente, consolidadas produzindo a saída final do processamento.

Há diversas implementações desse modelo como, por exemplo, Google MapReduce, que utiliza linguagem C++; Greenplum, desenvolvido em Python, entre outras. No entanto, a

implementação em `Java`, conhecida como Hadoop MapReduce é, talvez, a mais conhecida e difundida por ser *open-source* e fazer parte da plataforma *Hadoop*.

## 3.8 Plataforma Hadoop

O projeto Hadoop foi criado por Doug Cutting, no início dos anos 2000, inspirado pelos recursos *Google File System* (GFS) e MapReduce, ambos criados pela empresa Google. Posteriormente, o projeto passou a pertencer à Fundação Apache na qual encontra-se hospedado<sup>8</sup>. O Apache™ Hadoop<sup>©</sup> é uma plataforma para processamento de grandes quantidades de dados em *clusters* computacionais (WHITE, 2012). Atualmente as versões mais recentes, consideradas estáveis, são 3.1.2 e 3.2.0. O núcleo do Hadoop é composto por quatro módulos, a saber:

- **Hadoop Common:** é composto por softwares utilitários que dão suporte à plataforma;
- **Hadoop Distributed File System:** é o sistema de arquivos distribuído;
- **Hadoop MapReduce:** o módulo tem por objetivo facilitar o desenvolvimento de aplicações paralelas;
- **Hadoop YARN:** é responsável pelo gerenciamento de recursos do *cluster*.

### 3.8.1 Hadoop Distributed File System (HDFS)

Um sistema de arquivos distribuído deve realizar operações de forma transparente como um sistema de arquivos convencional e possibilitar o armazenamento e o compartilhamento dos arquivos em diferentes dispositivos interconectados em uma rede de computadores. Além disso, ele deve tratar a segurança de acesso, oferecer desempenho similar a um sistema de arquivos tradicionais, prover escalabilidade. No Hadoop, o principal sistema de arquivos distribuído utilizado em aplicações Big Data é conhecido como *Hadoop Distributed File System*, ou HDFS, cuja arquitetura *master-worker* é ilustrada pela Figura 16.

Um único Nó da rede (mestre) deve ser configurado com o processo "Nome" (NameNode), que é responsável por gerenciar o sistema de arquivos e controlar o acesso aos arquivos. Além disso, é possível configurar um processo "Nome Secundário" (SecondaryNameNode) que opera como assistente do NameNode. Os demais Nós (trabalhadores) são configurados com o processo "Dados" DataNode, geralmente um por Nó, que é responsável por gerenciar os dados armazenados localmente na unidade computacional em que o processo está sendo executado. O Nó com o processo "Dados" (DataNode) comunica-se regularmente com o NameNode e realiza operações de leitura e escrita sobre os dados. Antes de iniciar o processamento de uma tarefa, o conjunto de dados é copiado para o sistema HDFS e subdividido em blocos de dados (*chunks*) os quais são distribuídos e replicados aos processos DataNode. Cada bloco HDFS

<sup>8</sup> Disponível em: <http://hadoop.apache.org/>. Último acesso em 15/02/2019.

possui tamanho padrão de 64 MB, que pode ser ajustado de acordo com a aplicação; além disso, também é possível configurar a replicação dos blocos, cujo valor padrão é 3, de modo a permitir a recuperação de dados em caso de falhas (SHVACHKO; KUANG; RADIA; CHANSLER, 2010; WHITE, 2012; ROCHA, 2013).

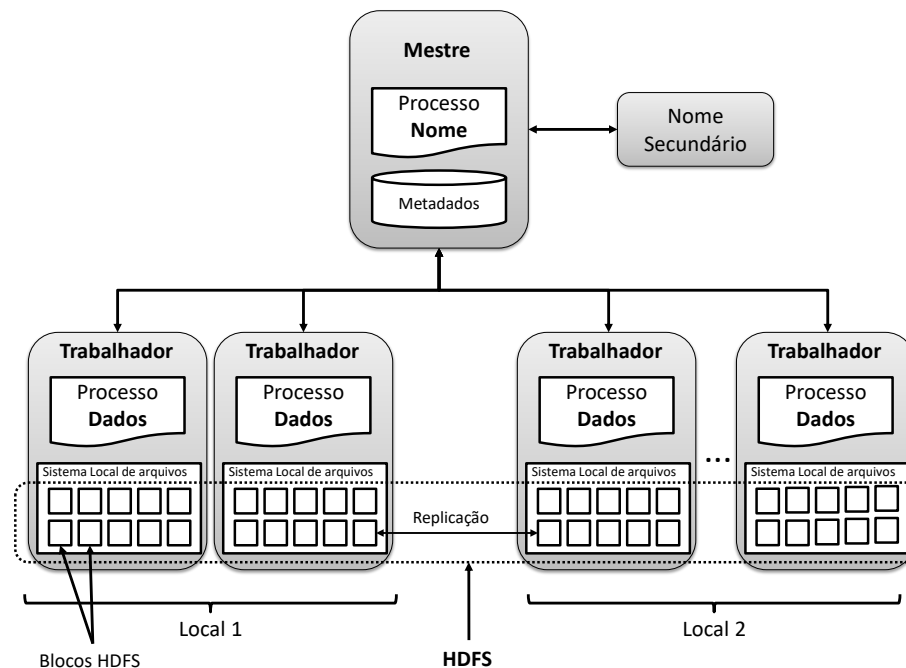


Figura 16 – Arquitetura do sistema HDFS. Adaptado de Gunarathne (2015).

O sistema HDFS é tolerante a falhas e projetado para trabalhar em redes compostas por computadores de baixo custo. O objetivo central da arquitetura é detectar falhas e recuperá-las automaticamente no menor tempo possível<sup>9</sup>.

### 3.8.2 Hadoop MapReduce

O modelo de programação MapReduce proposto por Dean e Ghemawat (2008) é implementado no Hadoop por meio de dois processos: JobTracker e TaskTracker. O primeiro, geralmente, é associado ao Nó mestre e possui as responsabilidades de agendar e gerenciar as submissões de tarefas, além de monitorar o processamento dos vários Nós que estão executando processos TaskTracker. O segundo processo ocorre nos Nós trabalhadores (*workers*) e fornece condições para a execução das tarefas *map* e *reduce* submetidas pelo processo JobTracker. Todo processo TaskTracker é configurado com um número de tarefas que ele pode executar concorrentemente (ROCHA, 2013).

Considerando que a largura de banda da rede é um recurso crítico, o MapReduce procura aproveitar o fato dos dados de entrada estarem armazenados nos Nós trabalhadores. Para isso, o JobTracker obtém a localização de um determinado bloco de dados, por exemplo, e procura

<sup>9</sup> Disponível em: [Arquitetura HDFS](#). Último acesso em 15/02/2019.

submeter a tarefa *map*, associada ao bloco, para o processo `TaskTracker` que está no mesmo Nó em que se encontram os dados. Esse procedimento pode ser definido como *data locality optimization* (WHITE, 2012).

É importante observar, também, que durante a configuração do Hadoop em um *cluster*, cuja arquitetura deve ser *master-worker*, o Nó mestre recebe os processos: `NameNode`, referente ao sistema HDFS, e `JobTracker`, referente ao MapReduce. Do mesmo modo, os demais Nós do *cluster* recebem os processos: `DataNode`, referente ao sistema HDFS, e `TaskTracker`, referente ao MapReduce.

### 3.8.3 Hadoop YARN

O YARN (*Yet Another Resource Negotiator*) é um gerenciador de recursos introduzido na segunda versão do Hadoop, que permite a diversos frameworks de processamento distribuído, a exemplo do Apache Spark, compartilhar dos recursos de um *cluster* Hadoop bem como dos dados armazenados no HDFS. Antes do surgimento do YARN, o Hadoop suportava apenas a execução do framework MapReduce. A Figura 17 ilustra o fluxo de execução de trabalho gerenciado pelo Hadoop YARN.

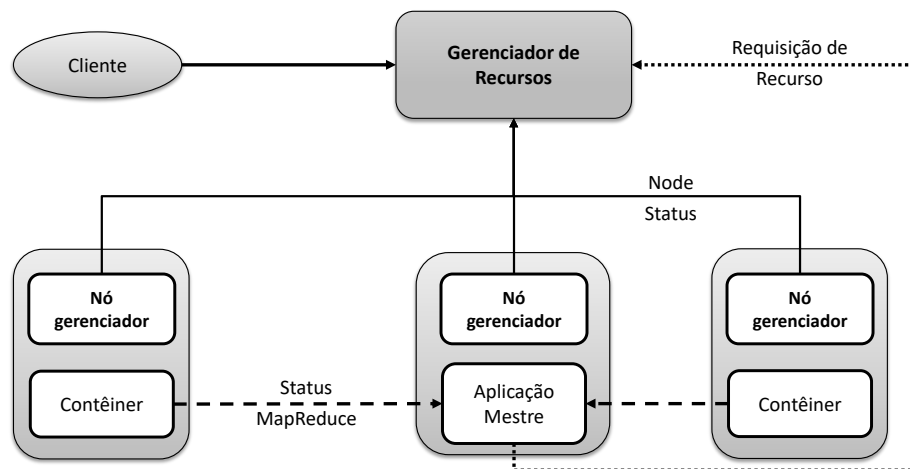


Figura 17 – Diagrama esquemático do fluxo de execução de trabalho gerenciado pelo Hadoop YARN. Adaptado de Gunarathne (2015) e Hadoop (2018).

O YARN contém o `ResourceManager` que trata-se de um gerenciador global de recursos, está localizado no Nó mestre do *cluster* e possui dois principais componentes: escalonador e gerente de aplicações. O escalonador é responsável por alocar e monitorar os recursos das diversas aplicações. O gerente de aplicações, por sua vez, aceita os trabalhos recebidos dos clientes, como ilustrado no diagrama esquemático da Figura 17, além de negociar o primeiro contêiner de uma aplicação. Além disso, o `ResourceManager` trabalha em conjunto com o `NodeManager` que localiza-se nos Nós trabalhadores e é responsável por monitorar os recursos da máquina trabalhadora em que ele está.

O YARN também contém um `ApplicationMaster` (Aplicação Mestre) para cada aplicação que localiza-se em uma das máquinas trabalhadoras. Ele é responsável solicitar recursos ao `ResourceManager` para disponibilizar um ou mais contêineres necessários para atender os requisitos da aplicação.

### 3.8.4 Ecosystema Hadoop

Além dos sistemas HDFS, YARN e MapReduce, diversos outros projetos compõem o Hadoop de modo a fornecerem serviços relacionados a computação distribuída em larga escala, formando o que é considerado o *Ecosystema Hadoop*. Uma lista desses projetos relacionados ao Hadoop pode ser obtida no site <http://hadoop.apache.org><sup>10</sup>.

É importante reiterar que existem inúmeros projetos que podem ser organizados em um amplo ecossistema Hadoop o que torna, praticamente, impossível retratá-lo em sua totalidade. No entanto, a Figura 18 busca ilustrar uma visão de parte desse ecossistema e enfatiza os projetos listados na página da internet em que o Hadoop está hospedado atualmente.

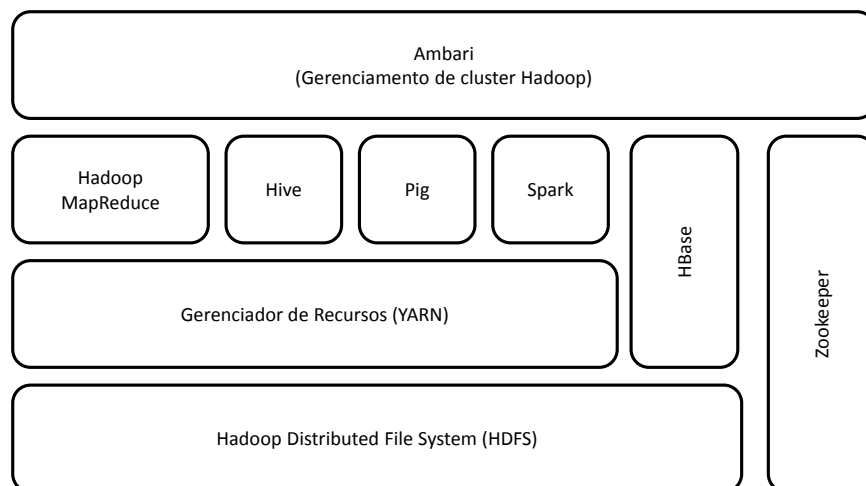


Figura 18 – Ecosystema Hadoop.

As seções 3.8.1 e 3.8.2 apresentaram o sistema HDFS e o Hadoop MapReduce, respectivamente. O gerenciador de recursos YARN, introduzido na versão 2 do Hadoop, fornece interfaces para requisição e utilização dos recursos do *cluster*. No entanto, tais interfaces não são, geralmente, usadas diretamente pelo código da aplicação que utiliza outras interfaces consideradas de *alto nível*, pois abstraem detalhes da plataforma, que visam facilitar o desenvolvimento da aplicação (WHITE, 2012). Esses três itens (HDFS, MapReduce e YARN) compõem o núcleo do Hadoop conforme já mencionado.

Além deles, observam-se outros elementos na Figura 18. O *Hive* é um *data warehouse* distribuído e gerencia arquivos no HDFS, além de prover uma linguagem de consulta inspirada

<sup>10</sup> Último acesso em 09/08/2017.

no SQL. *Pig* é uma linguagem de procedimentos de alto nível para grandes bases de dados e pode ser executada em *clusters* que utilizam HDFS e MapReduce. O *HBase* é um banco de dados distribuído orientado a colunas, portanto é considerado um banco NoSQL. O *Zookeeper* é um coordenador de serviços para aplicações distribuídas. O *Ambari* é uma ferramenta web para provisionamento, gerenciamento e monitoramento de *clusters* que utilizam o Hadoop. Destaca-se, ainda, o *Spark* que é um módulo para processamento rápido em *clusters*, para aplicações de propósito geral.

Vale mencionar que atualmente há diversas implementações comerciais da plataforma Hadoop. Isso ocorre pelo fato que uma aplicação Big Data geralmente necessita de um grande conjunto de ferramentas que formam um complexo ecossistema. Logo, preparar, configurar e instalar um ambiente desse, exige amplo e considerável conhecimento técnico por parte dos especialistas. Por conta disso, diversas empresas prepararam suas próprias versões do Hadoop de modo a tornar mais fácil o processo de instalação e utilização das ferramentas de Big Data. Algumas das principais empresas que fornecem tal tipo de serviço são: IBM, Cloudera, HortonWorks, Amazon Web Services.

### 3.9 Apache Spark

O projeto Apache Spark iniciou-se em 2009 na Universidade da Califórnia, Berkeley. O grupo que, originalmente, começou o desenvolvimento do Spark observou que a natureza do Big Data é muito diversificada e não possui uma organização clara e única. Os pesquisadores do grupo ponderaram que há processos que podem ser modelados pelo *MapReduce* e outros que utilizam consultas no estilo SQL, por exemplo. Eles observaram ainda a existência de diferentes ferramentas especializadas como o *MapReduce* para processamento em lote (*batch*), *Dremel* para consultas SQL interativas e o *Pregel* voltado para algoritmos em grafos e que, geralmente, as aplicações Big Data necessitam combinar vários recursos para os diferentes tipos de processamentos (ZAHARIA et al., 2016).

O Spark, portanto, foi projetado como um framework unificado para processamento de dados distribuídos, de fácil compreensão e que tenha capacidade de lidar com dados de diversas naturezas e diferentes origens. As aplicações são executadas no *cluster* como conjuntos de processos independentes coordenados pelo que é denominado objeto de contexto no programa principal (*driver program*), conforme ilustrado pela Figura 19.

O Apache Spark pode trabalhar com três gerenciadores de recursos (*cluster manager*): Hadoop YARN, Apache Mesos e *Standalone* (gerenciador nativo do Spark usado principalmente para testes em máquinas individuais). Além disso, ele utiliza o sistema HDFS para armazenamento de dados e funciona com qualquer fonte de dados compatível com Hadoop e as aplicações podem ser desenvolvidas utilizando as seguintes linguagens de programação: Scala, Java e Python.



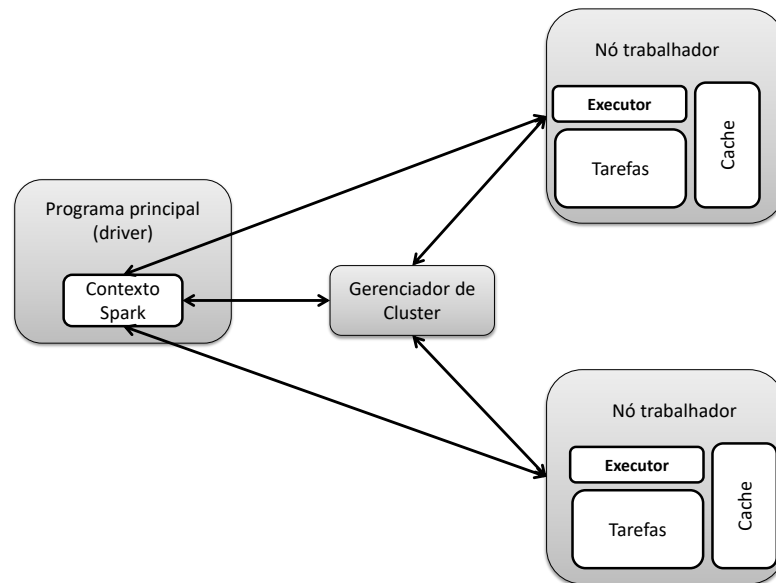


Figura 19 – Visão geral da arquitetura Spark. Fonte: <https://spark.apache.org/docs/latest/cluster-overview.html>. Último acesso em 16/02/2019.

É importante mencionar que existem bibliotecas adicionais que complementam o Spark e fornecem capacidades extras para as áreas de análise de Big Data e aprendizado de máquina. A Figura 20 ilustra a organização do Spark com as bibliotecas adicionais.

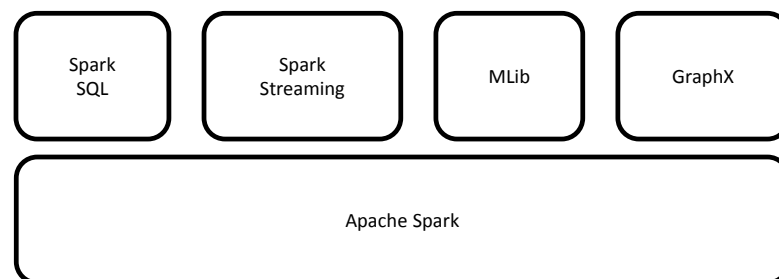


Figura 20 – Bibliotecas adicionais do Spark.

A biblioteca *Spark SQL* permite realizar consultas sobre os dados armazenados no estilo SQL. A *Spark Streaming* pode ser utilizada para processar dados de fluxo em tempo real. *MLib* é uma biblioteca de aprendizado de máquina e *GraphX* permite realizar computação baseada em grafos.

O Apache Spark possui o modelo de programação semelhante ao MapReduce porém ele é estendido, pois utiliza o conceito de compartilhamento de dados por meio do *Resilient Distributed Datasets (RDD)*, que provê armazenamento de dados em memória e evita movê-los durante o processamento. É importante observar que atualmente o Apache Spark não permite aninhar RDD como observado no trabalho dos autores [Katsogridakis, Papagiannaki e Pratikakis \(2017\)](#), que propuseram um mecanismo para lidar com situações em que algoritmos não são facilmente expressados de acordo com o modelo MapReduce.



### 3.10 Computação em Nuvem

A demanda por recursos computacionais tem aumentado em função da maior disponibilidade de acesso à Internet bem como do aumento da complexidade e da quantidade de dados envolvidas nas aplicações. A ideia de *Computação em Nuvem* geralmente está associada à concepção de recursos ilimitados e disponíveis a qualquer momento. Além disso, o conceito de utilização da *computação como serviço* reforça a ideia de computação em nuvem uma vez que diminui os recursos instalados e consumidos por parte do usuários que passam a dedicar mais esforços aos aspectos de seus negócios e menos com aspectos de infraestrutura (VELTE; VELTE; ELSENPETER, 2009).

Observa-se que a computação em nuvem representa a sinergia de conceitos e fundamentos das áreas de virtualização de servidores, computação em grade, computação em *cluster*, segurança, software orientado a serviços, gestão de centros de dados; e trata-se de um modelo, como observam Mell e Grance (2011), que permite acesso via rede a um conjunto compartilhado de recursos que podem ser rapidamente fornecidos e liberados. Tal modelo possui cinco características essenciais:

- serviço por demanda: o usuário consome os recursos da nuvem que estão à sua disposição de acordo com a necessidade e de forma automática;
- amplo acesso a rede: o usuário pode acessar os recursos da nuvem por meio de dispositivos padrões, tais como celulares e notebooks, de forma transparente e independente de plataforma;
- *pool* de recursos: o provedor deverá ser capaz de fornecer os mais variados recursos e distribuí-los dinamicamente de acordo com a demanda do consumidor;
- elasticidade rápida: a capacidade dos recursos deve ser provisionada e liberada de forma rápida, automática e transparente de acordo com a necessidade do usuário o que passa a ideia de "recursos ilimitados";
- serviço medido: o provedor deve ter a capacidade de controlar e otimizar automaticamente o uso dos recursos por meio de métricas que possibilitem ao usuário monitorar os custos gerados em função dos serviços utilizados.

Atualmente há diferentes provedores, a exemplo das empresas Google, Microsoft e Amazon, que atendem aos conceitos e características da computação em nuvem. Na próxima seção, discorre-se sobre a estrutura do provedor da Amazon.

### 3.10.1 Amazon Web Services (AWS)

A empresa multinacional Amazon<sup>11</sup> considerada uma das maiores do mercado de varejo online no mundo, criou em 2006 uma empresa denominada *Amazon Web Services (AWS)* que passou a oferecer sua infraestrutura de servidores como uma plataforma de serviços para desenvolvimento de aplicações em nuvem.

A nuvem AWS possui uma infraestrutura organizada por regiões, zonas de disponibilidade e pontos de presença que abrange 61 zonas de disponibilidades em 20 regiões geográficas em todo o mundo<sup>12</sup>. Uma região representa um local físico do mundo, enquanto que a zona de disponibilidade representa um *datacenter* isolado que fica dentro de uma região e atua de forma independente, entretanto conectadas com todas as outras zonas da mesma região. Os pontos de presença referem-se a servidores especializados para prover conteúdos com alta velocidade e baixa latência.

Entre os diversos serviços que a AWS oferece, três são apresentados neste documento: EC2 (*Elastic Compute Cloud*), S3 (*Simple Storage Service*) e EBS (*Elastic Block Store*). O EC2 disponibiliza meios para criação de máquinas virtuais (VM) dentro do provedor da AWS que são denominadas instâncias. Cada VM pode ser configurada de acordo com a necessidade levando em consideração que o preço se altera em função da configuração escolhida. A configuração consiste na definição do tipo e o número de processadores, quantidade de núcleos por processador, quantidade de memória, capacidade de armazenamento, rede, sistema operacional, arquitetura (32 ou 64 bits), entre outros requisitos. Além disso, é necessário configurar a segurança de acesso a VM por meio de chave de autenticação e grupos de segurança. Ainda é possível por meio do AMI (*Amazon Machine Image*) criar imagens virtuais das instâncias de modo a facilitar a replicação de instâncias de mesma configuração em um *cluster*.

O S3 é um serviço de armazenamento de dados, na qual um *bucket* se assemelha à uma pasta em um diretório de arquivos e os objetos são os conteúdos que o usuário deseja armazenar. Além disso, a AWS disponibiliza uma API por meio da qual o usuário pode acessar o serviço.

O EBS, por sua vez, refere-se ao armazenamento persistente de dados da instância. Quando se cria uma máquina virtual estipula-se uma capacidade de armazenamento que ela irá utilizar, análogo a um disco rígido primário, e que é fixa. Pode-se associar o EBS à analogia de um disco rígido externo que pode aumentar a sua capacidade de acordo com a necessidade. Isso possibilita, por exemplo, que várias instâncias acessem um mesmo volume EBS onde os dados da aplicação poderão estar armazenados.

---

<sup>11</sup> <<https://www.amazon.com/>>

<sup>12</sup> <<https://aws.amazon.com/pt/about-aws/global-infrastructure/>>

# Capítulo 4

## MÉTODO DE RECONSTRUÇÃO TOMOGRÁFICA 2D E 3D (VOLUMÉTRICA) DE AMOSTRAS AGRÍCOLAS EM AMBIENTE BIG DATA

---

---

Este capítulo apresenta em detalhes o método desenvolvido para a reconstrução de imagens tomográficas, 2D e 3D (volumétrica), de amostras agrícolas em ambiente Big Data.

### 4.1 Base de dados e métricas de avaliação

Nesta seção são apresentados o conjunto de dados bem como as métricas utilizadas para avaliação do ambiente Big Data, da seleção de projeções baseada em densidade espectral e das reconstruções tomográficas 2D e 3D (volumétrica) de amostras de sementes agrícolas.

#### 4.1.1 Base de dados de projeções tomográficas

Para avaliação do processo de seleção de projeções foram utilizadas amostras simuladas dos *phantoms* heterôgeneo e homogêneo. O *phantom* heterôgeneo conhecido como Shepp-Logan refere-se a uma imagem de teste, preparada em laboratório, criada por Larry Shepp e Benjamin Logan em 1974 e é amplamente utilizada para testes durante o desenvolvimento de algoritmos de reconstrução de imagens (SHEPP; LOGAN, 1974).

A imagem simulada do *phantom* heterogêneo consiste de 10 elipses superpostas e, neste trabalho, foi gerada a partir da biblioteca de Processamento de Imagens em Python, *scikit-image*. Já a imagem do *phantom* homogêneo simula um cilindro formado unicamente por um elemento, a exemplo do Nylon, e consiste de uma circunferência cujo interior possui a mesma intensidade do tom de dinza. A Figura 21 apresenta os *phantoms* heterogêneo e homogêneo utilizados durante a avaliação do trabalho.

Adicionalmente, foram preparadas amostras de *phantoms* heterogêneos que se referem a corpos de ensaios construídos com parâmetros conhecidos ou de referência utilizados para a calibração ou caracterização de padrões. A Figura 22 ilustra o diagrama de concepção de um

*phantom* heterogêneo com diâmetro de 100mm e altura de 150mm bem como apresenta uma imagem tomográfica obtida a partir do *phantom* preparado.

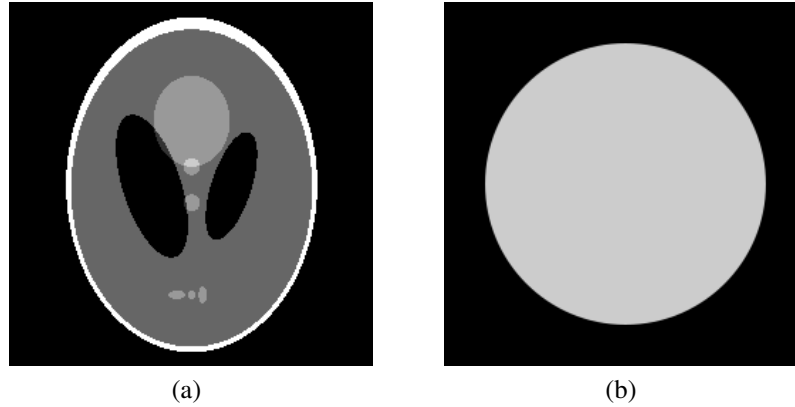


Figura 21 – Amostras simuladas utilizadas para avaliação. (a) *phantom* heterogêneo Shepp-Logan. (b) *phantom* homogêneo.

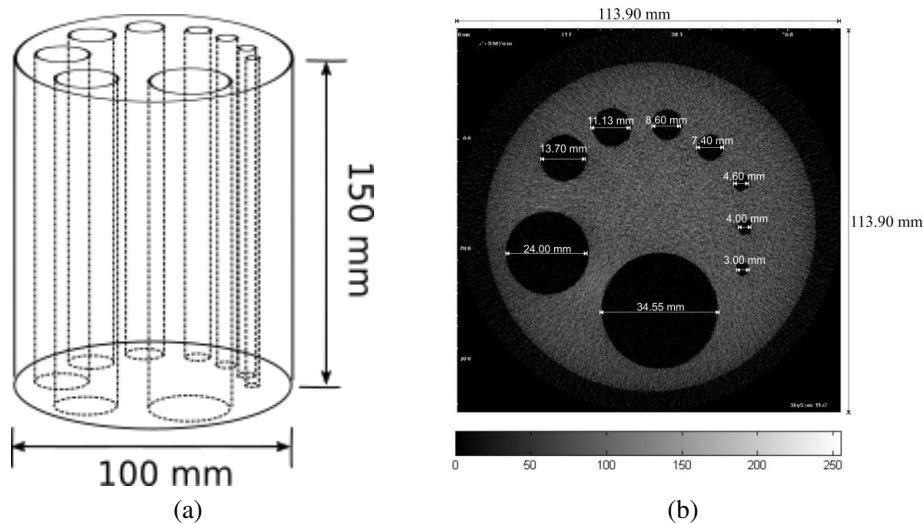


Figura 22 – Preparação do *phantom* de plexiglass a ser utilizado para validação. (a) Diagrama de concepção de um *phantom*. (b) Imagem tomográfica do *phantom*. Adaptado de Beraldo e colaboradores (2014).

O *phantom* de plexiglass com nove furos de diversos diâmetros<sup>1</sup> foi submetido ao tomógrafo de alta resolução. A Figura 23 apresenta o *phantom* utilizado como amostra nesta atividade.

A imagem à esquerda, Figura 23(a), apresenta o tomógrafo e a imagem à direita, Figura 23(b), destaca o *phantom* preparado para ser escaneado. As projeções foram adquiridas pelo tomógrafo de alta resolução SkyScan 1172.

<sup>1</sup> Os diâmetros dos furos são: 3,00mm; 4,00mm; 4,60mm; 7,40mm; 8,60mm; 11,13mm; 13,70mm; 24,00mm e 34,55mm



Figura 23 – Imagens do *phantom* inserido no tomógrafo de alta resolução SkyScan 1172.

Neste trabalho, além dos *phantoms*, foram utilizadas amostras de sete tipos de semente: Amendoim (*Arachis hypogaea*), Feijão Fradinho (*Vigna unguiculata*), Girassol (*Helianthus annuus*), Grão de Bico (*Cicer arietinum*), Trigo (*Triticum*), Abóbora (*Cucurbita*) e Soja (*Glycine max*). A Figura 24 apresenta os tipos de sementes e suas respectivas plantas.

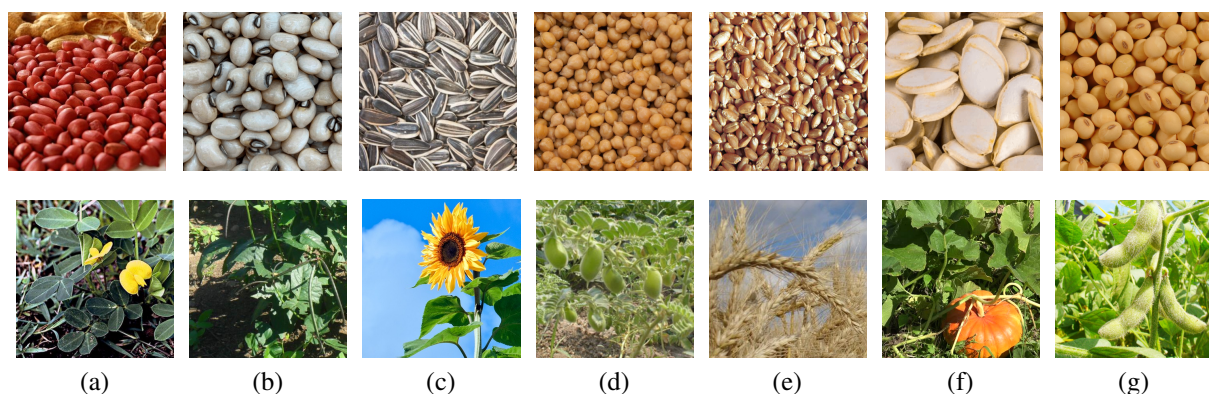


Figura 24 – Tipos de sementes utilizadas neste trabalho e suas respectivas plantas. (a) Amendoim. (b) Feijão Fradinho. (c) Girassol. (d) Grão de Bico. (e) Trigo. (f) Abóbora. (g) Soja.

Esses tipos de semente foram escolhidos, pois são relevantes para a economia brasileira, tanto para o consumo interno quanto para exportação. Logo, entende-se que a técnica tomográfica é uma alternativa para viabilizar a análise destas sementes e, conseqüentemente, auxiliar produtores com o cultivo de lavouras mais resilientes às condições climáticas atuais.

Foram preparadas cinco amostras para cada tipo de semente, exceto para os tipos Feijão Fradinho e Amendoim em que foram preparadas quatro amostras, o que totalizou 33 amostras.

A aquisição das matrizes de projeções das amostras de sementes bem como das matrizes de projeções do *phantom* foi realizada no tomógrafo SkyScan 1172. Para isto, foi necessário ajustar o equipamento que utilizou a mesma configuração tanto para as amostras quanto para o *phantom*.

A Tabela 1 apresenta os valores dos parâmetros ajustados no tomógrafo para a aquisição das projeções.



Tabela 1 – Valores dos parâmetros do tomógrafo SkyScan 1172 ajustados para aquisição das projeções do *phantom* e das amostras de sementes.

Parâmetro	Valor
Número de arquivos	976 (.tif)
Número de seções	2096
Passo angular (graus)	0, 20
Tempo de exposição	790 ms
Intervalo de rotação	0° a 180°
Tamanho do pixel da imagem	8,54 $\mu$ m
Voltagem	100kV
Corrente	100 $\mu$ A
Primeira seção	68
Última seção	1968
Variação angular de reconstrução (graus)	195, 20

As amostras foram escaneadas no intervalo de 0° a 195, 2° com passo angular de 0, 2°. Logo, uma matriz de projeções, ou sinograma, contém 976 projeções e 2000 pontos por projeção. O número de projeções (976) foi determinado em função a parte experimental que está relacionada com o tamanho da amostra e a configuração dos parâmetros do tomógrafo como, por exemplo, ângulo de rotação e resolução do pixel que influenciam na visibilidade da amostra dentro do tomógrafo. Cada ponto de projeção possui 2 bytes, portanto uma amostra corresponde a:  $1960 \times 2000 \times 976 \times 2 \approx 7, 13\text{GB}$ . Além das 33 amostras foi preparado um *phantom* com as mesmas configurações das sementes, portanto foram utilizadas 34 amostras com 7, 13GB cada totalizando cerca de 242GB de dados de entrada.

### 4.1.2 Métricas de avaliação

Neste trabalho, foram utilizadas as métricas de avaliação NRMSE (*Normalized Root Mean Squared Error*), SSIM (*Structural Similarity*) e PSNR (*Peak Signal-to-Noise*) que são medidas baseadas em diferenças entre pixels. Para o cálculo dessas medidas é necessário a imagem de referência, também chamada de *ground-truth*, e a imagem que se quer avaliar.

Para o cálculo da medida NRMSE, primeiro calcula-se a medida MSE (*Mean Squared Error*), dada pela Equação 4.1, que é computada como a média da intensidade ao quadrado entre as imagens original e resultante, ambas de tamanho  $M \times N$ .

$$MSE(x, y) = \frac{1}{MN} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} e(n, m)^2 \quad (4.1)$$

onde  $e(m, n)$  é a diferença (erro) entre a imagem original e a resultante. Na sequência, a Equação 4.2 fornece o cálculo para a medida NRMSE.

$$NRMSE(x, y) = \frac{\sqrt{MSE(x, y)}}{\sqrt{\|x\|^2 - \|y\|^2}} \quad (4.2)$$

O índice SSIM, dado pela Equação 4.3, considera a degradação da imagem como mudança percebida na informação estrutural, ao mesmo tempo que incorpora fenômenos como luminância e contraste. A informação estrutural consiste na ideia que os pixels possuem forte interdependência especialmente quando estão próximos espacialmente. Estas dependências carregam importantes informações sobre a estruturas dos objetos na cena.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.3)$$

onde  $x$  e  $y$  são, respectivamente, as imagens original e resultante;  $\mu_x$  é a média de  $x$ ;  $\mu_y$  é a média de  $y$ ;  $\sigma_x^2$  é a variância de  $x$ ;  $\sigma_y^2$  é a variância de  $y$ ;  $\sigma_{xy}$  é a covariância de  $x$  e  $y$ ;  $c_1 = (k_1L)^2$  e  $c_2 = (k_2L)^2$  são variáveis que estabilizam a divisão sendo que  $L$  é o intervalo dos valores dos pixels (geralmente  $2^{bits} - 1$ ) e  $k_1 = 0,01$  e  $k_2 = 0,03$  por padrão.

A medida PSNR, dada pela Equação 4.4, é baseada na relação sinal-ruído que é uma estimativa da imagem reconstruída comparada com a imagem original.

$$PSNR(x, y) = 10 \log \frac{s^2}{MSE(x, y)} \quad (4.4)$$

onde  $s = 255$  para imagens com 256 tons de cinza.

Para avaliação do modelo proposto, foi reconstruída uma imagem a partir do conjunto completo de projeções de uma amostra a qual foi considerada a imagem de referência, ou *ground-truth*. Posteriormente, as imagens foram reconstruídas a partir de subconjuntos de projeções da mesma amostra. O cálculo do NRMSE informou o erro entre a imagem de referência em relação às demais e, nesse caso, quanto menor o valor da medida menor foi o erro. Já o cálculo do SSIM indicou a similaridade entre a imagem de referência em relação às demais, pois ele considera outros aspectos não observados pelo NRMSE. Nesse caso, quanto mais próximo de 1 o valor do SSIM maior foi considerada a proximidade entre as imagens avaliadas. A medida PSNR obteve a relação sinal-ruído entre a imagem reconstruída com menos projeções em relação à reconstruída com todas as projeções. Logo, os valores obtidos pelas métricas puderam auxiliar na análise do modelo fornecendo indicativos, por exemplo, de qual subconjunto de projeções foi o mais adequado para reconstrução.

A fim de avaliar o ambiente paralelo foi utilizada a métrica Speedup, calculada pela Equação 4.5, que determina o aumento de velocidade obtido para a execução de um programa utilizando  $p$  processadores, em relação a sua execução sequencial, usando um único processador.

Na equação,  $T_{seq}$  e  $T_{par}$  são os tempos sequencial e paralelo, respectivamente, para executar o mesmo programa.

$$S_p = \frac{T_{seq}}{T_{par}} \quad (4.5)$$

Outra medida empregada para avaliação do ambiente Big Data é a Eficiência, cujo valor é obtido por meio da Equação 4.6.

$$E_p = \frac{S_p}{p} \quad (4.6)$$

A medida de Eficiência avalia quanto o paralelismo foi explorado no algoritmo e quantifica a utilização do processador. Geralmente, o valor da medida fica no intervalo  $[0, 1]$  e, neste caso, quanto mais próximo de 1 for o valor de  $E_p$  maior a eficiência. Por outro lado, há situações em que o valor da Eficiência pode ser superior a 1 (KONTOGHIORGHES, 2005).

## 4.2 Visão sistêmica do método

O intuito desta seção é descrever os principais elementos que compõe o método de reconstrução de imagens tomográficas em ambiente Big Data e como eles se relacionam de modo a prover uma visão sistêmica. A Figura 25 apresenta o diagrama de blocos que descreve o método desenvolvido. A linha tracejada delimita as etapas que foram implementadas em ambiente Big Data.

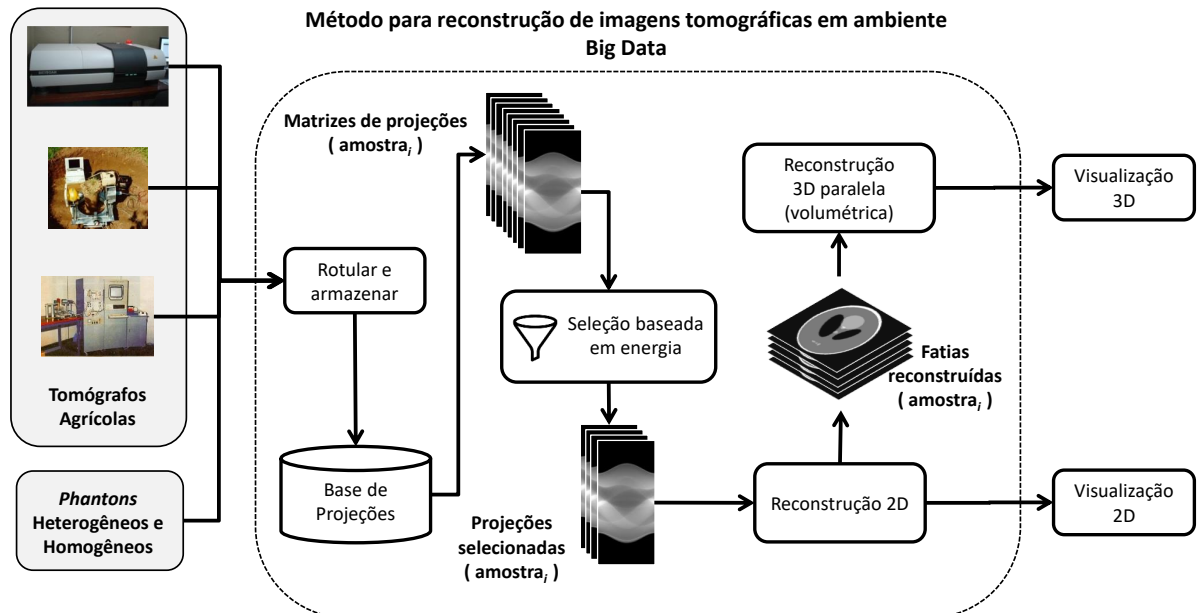


Figura 25 – Diagrama de blocos do método para reconstrução de imagens tomográficas provenientes de amostras agrícolas em ambiente Big Data.

Uma amostra ao ser escaneada, pelo tomógrafo selecionado, produz um conjunto de projeções (matriz), dito completo. Tais conjuntos de projeções tomográficas são oriundos de



amostras agrícolas tais como sementes, plantas, madeira e amostras de solo entre outras. Além disso, amostras simuladas (*phantoms* heterogêneo e homogêneo) são utilizadas para apoiar a calibração e validação do método.

A primeira etapa do método, portanto, consiste em rotular e *armazenar* essas amostras em uma base de projeções no ambiente Big Data. O rótulo consiste em um conjunto de metadados a fim de descrever o tipo e as características das amostras, bem como para identificá-las e facilitar o acesso a elas dentro do referido ambiente. As amostras são armazenadas em um sistema de arquivos distribuídos que facilita o acesso pelas demais etapas no ambiente Big Data.

Na sequência, o intuito da etapa de *seleção de projeções* é a partir do conjunto completo, o de identificar um subconjunto de projeções que também possam representar a amostra. A Figura 26 descreve esta etapa, em mais detalhes, apresentando as fases de como ocorre a seleção das projeções.

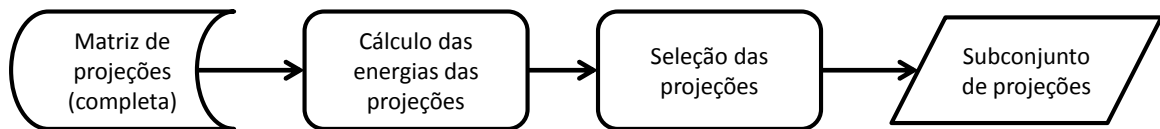


Figura 26 – Fases da etapa de seleção de projeções.

A primeira fase recupera a matriz completa de projeções de uma determinada amostra que foi previamente armazenada na base de projeções. Na sequência é realizado o cálculo das energias de todas as projeções da matriz por meio da densidade espectral. Na terceira fase, ocorre a seleção propriamente dita das projeções. Entende-se que encontrar o conjunto de projeções tomográficas plausível significa alcançar a relação que possa maximizar a qualidade das reconstruções tomográficas ao mesmo tempo que minimiza o conjunto de dados necessários para serem processados. Por melhor conjunto entende-se que é aquele que fornece a maior quantidade de informação com a menor quantidade de dados possível.

Na etapa seguinte, as projeções selecionadas são submetidas à reconstrução bidimensional por meio do algoritmo FBP (*Filtered Back Projection*) que foi paralelizado utilizando a estrutura RDD (*Resilient Distributed Dataset*) e o conceito de *map* e *reduce*, conforme discutido no Capítulo 3. Neste ponto é importante mencionar que, desde a implantação, a Embrapa Instrumentação vem trabalhando com o tradicional algoritmo FBP nos estudos sobre a tomografia na física de solos (PIRES; BORGES; BACCHI; REICHARDT, 2010). Portanto, a utilização do algoritmo FBP permite avaliar a concepção do método desenvolvido neste trabalho, bem como a integração e utilização com outros métodos já existentes. Após à reconstrução, as fatias reconstruídas são então submetidas à etapa da reconstrução volumétrica paralela. De modo geral, as fatias são divididas em regiões, ou *tiles*, que posteriormente são reunidas e interpoladas por meio da interpolação *B-spline*. A partir daí, os dados gerados pelas reconstruções 2D e 3D (volumétrica) ficam disponíveis no ambiente distribuído para visualização, bem como para

compor uma base de conhecimento.

Na seção seguinte é apresentado o conjunto de tecnologias empregadas para a organização do ambiente Big Data. Posteriormente, as etapas de seleção das projeções baseada em densidade espectral e as reconstruções 2D e 3D (volumétrica) de modo paralelo são apresentadas em detalhes.

### 4.3 Organização do ambiente Big Data

A concepção do método de reconstrução de imagens tomográficas exigiu a organização de um ambiente Big Data que estivesse preparado para o processamento de grandes quantidades de projeções tomográficas de modo paralelo e distribuído, além de considerar que a análise fosse realizada o mais próximo dos dados, bem como ter condições de processamento em memória e que possuísse coordenação entre as unidades de dados e de processamento visando aprimorar a escalabilidade. O intuito, portanto, foi a concepção de um ambiente que considerasse os princípios elencados na seção 3.6 para o projeto de sistemas Big Data.

Logo, a organização do ambiente Big Data foi considerada em duas perspectivas: infraestrutura e aplicação. A construção destas duas perspectivas se deu por meio de um rol de tecnologias. Vale mencionar que há um vasto conjunto de tecnologias disponíveis, conforme observado no Capítulo 3 e, portanto, foi necessário selecionar as mais apropriadas à aplicação.

A Figura 27 ilustra a pilha de tecnologias empregada na organização do ambiente Big Data para este trabalho e enfatiza, por meio das linhas tracejadas, o *cluster* que representa a infraestrutura e o método de reconstrução de imagens tomográficas, desenvolvido em linguagem Python.

Ao observar a pilha de tecnologias, verifica-se que a primeira camada refere-se ao armazenamento dos dados cuja tecnologia utilizada foi o sistema de arquivos distribuídos da Amazon, S3. A plataforma *Amazon Elastic MapReduce* (EMR) permitiu estruturar o *cluster* cujos computadores, ou Nós do *cluster*, são instâncias do *Amazon Elastic Compute Cloud* (EC2). O Apache Spark é o framework utilizado para processamento de dados distribuídos e no qual o método de reconstrução de imagens tomográficas é executado. Além disso, o Hadoop YARN atua em conjunto com o Spark e tem a função de gerenciar os recursos do *cluster*.

Do ponto de vista da aplicação, a camada referente à biblioteca `MRJob`<sup>2</sup> é responsável por integrar aplicações escritas em Python com diversos serviços de computação em nuvem, a exemplo dos oferecidos pela Amazon. Além disso, a biblioteca possui a vantagem de tornar a organização e configuração do *cluster* mais transparente para o desenvolvedor.

O método de reconstrução de imagens tomográficas 2D e 3D (volumétrica) representado pela última camada na Figura 27 foi escrito em linguagem Python, cujo módulo foi denominado

<sup>2</sup> Disponível em: <<https://github.com/Yelp/mrjob>>. Último acesso em 11/07/2019.

ctrecon, e utilizou diversas bibliotecas que estão representadas na penúltima camada, a exemplo do PySpark<sup>3</sup>, Numpy<sup>4</sup> bem como diversas bibliotecas auxiliares representadas pelo bloco *utils*.

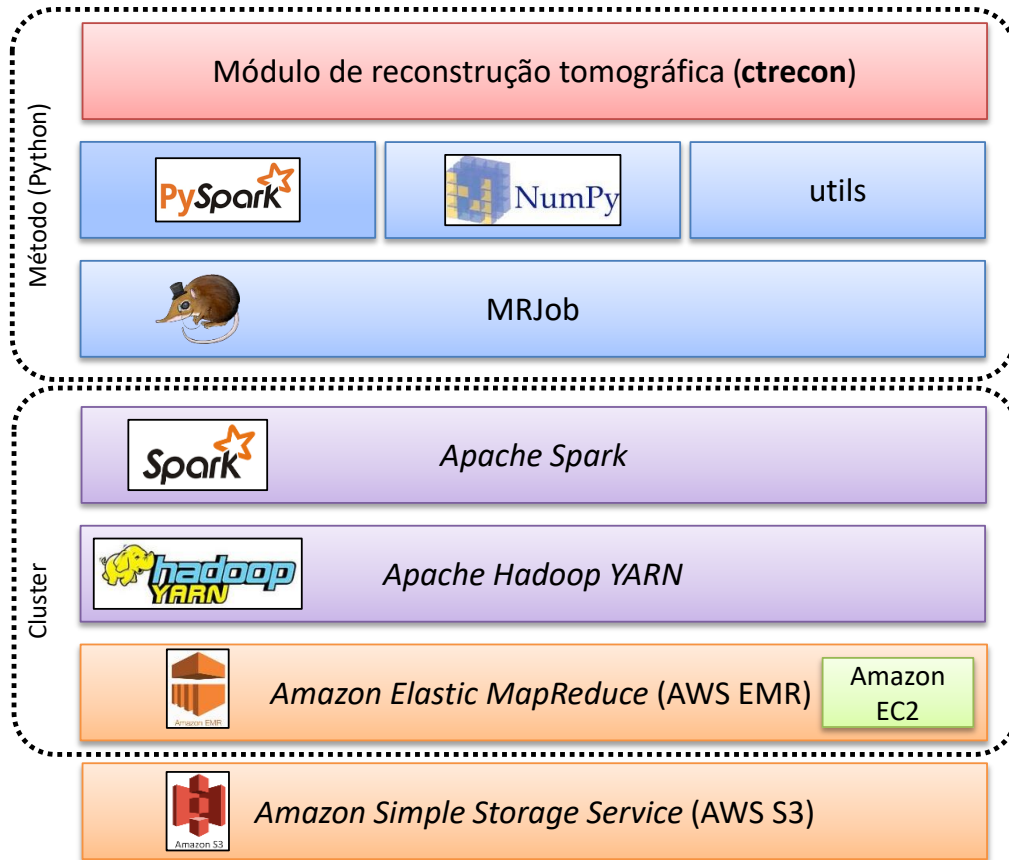


Figura 27 – Pilha de tecnologias empregadas para a organização do ambiente Big Data.

O Quadro 3 apresenta as versões das tecnologias utilizadas para a organização do ambiente Big Data.

Quadro 3 – Versões das tecnologias empregadas na organização do ambiente Big Data.

Tecnologia	Versão
PySpark	2.4.2
Numpy	1.16.4
MRJob	0.6.9
Apache Spark	2.4.2
AWS EMR	emr-5.24.0
Apache Hadoop YARN	2.8.5
Java (OpenJDK)	1.8.0_201
Python	3.7.6

Conforme mencionado anteriormente, um *cluster* organizado no AWS EMR é composto por instâncias do Amazon EC2. Uma instância representa um computador com configurações

<sup>3</sup> Disponível em: <<https://spark.apache.org/docs/latest/api/python/index.html>>. Último acesso em 13/07/2019.

<sup>4</sup> Disponível em: <<https://numpy.org/>>. Último acesso em 08/07/2019

específicas de memória, armazenamento e processamento. A Amazon possui ampla diversidade de configurações que são classificadas nos seguintes tipos de instâncias<sup>5</sup>:

1. **uso geral**: modelos a1, t3, t3a, t2, m5, m5a, m4;
2. **otimizadas para computação**: modelos c5, c5n, c4;
3. **otimizadas para memória**: modelos r5, r5a, r4, x1e, x1, z1d;
4. **computação acelerada**: modelos p3, p2, g3, f1;
5. **otimizadas para armazenamento**: modelos i3, i3en, d2, h1.

Cada tipo está associado a um conjunto de modelos de instâncias, no qual cada modelo possui um código precedido, geralmente, por uma letra e um número. Além disso, um modelo ainda pode conter as seguintes variações: *nano*, *micro*, *small*, *medium*, *large*, *xlarge* e *nlarge*. Tais variações fornecem uma ideia geral de capacidade, sendo a *nano* a de menor capacidade e a *nlarge* a de maior capacidade. Importante observar que nesta última variação, *n* é um fator multiplicador da variação *xlarge*, por exemplo, *2xlarge*, *8xlarge* ou *24xlarge*.

Além das configurações de memória, armazenamento e processamento, outro aspecto importante que deve ser levado em consideração, durante a definição das instâncias, é a disponibilidade das máquinas na região em que o *cluster* é montado. A informação de disponibilidade encontra-se no painel do EC2, opção *Limits*. Para a região "São Paulo" (código *sa-east-1*) o limite total foi de 20 máquinas, logo várias configurações não estavam disponíveis por padrão. Portanto, para este trabalho buscou-se utilizar instâncias que possuem equilíbrio entre processamento, memória e disponibilidade na região *sa-east-1* e, por isso, optou-se por instâncias de uso geral. O Quadro 4 apresenta os tipos de instâncias utilizados para avaliação do ambiente preparado para execução do método desenvolvido.

Quadro 4 – Tipos de instâncias utilizados na composição do *cluster*.

Modelo	vCPU	Memória (GiB)
m5.xlarge	4	16
m5.2xlarge	8	32
m5.4xlarge	16	64

É importante destacar que neste trabalho foi organizado um *cluster* homogêneo, ou seja, todas as máquinas possuem a mesma configuração, de modo que a avaliação considerou a quantidade de instâncias de mesmo modelo.

### 4.3.1 Configuração do *cluster*

A biblioteca MRJob auxiliou na preparação do *cluster* por meio do arquivo de configuração denominado `.mrjob.conf`. Este arquivo consiste de informações que seguem o padrão

<sup>5</sup> A relação completa dos tipos de instâncias pode ser obtida em: [<https://aws.amazon.com/pt/ec2/instance-types/>](https://aws.amazon.com/pt/ec2/instance-types/)

(chave: valor). As chaves, neste caso, referem-se aos parâmetros de configuração. O Quadro 5 descreve a funcionalidade dos parâmetros utilizados na organização do *cluster*.

Quadro 5 – Parâmetros para a organização e configuração do *cluster*.

Parâmetro	Descrição
label	Rótulo utilizado para identificar o <i>cluster</i> no AWS EMR.
max_mins_idle	Tempo máximo, em minutos, que o <i>cluster</i> permanece ligado e ocioso. Após este tempo ele desliga.
instance_groups	Define grupos de instâncias.
InstanceRole	Especifica o tipo do grupo de instância: <i>master</i> ou <i>core</i> .
Name	Nome do grupo.
InstanceCount	Determina o número de instâncias que será utilizada no grupo.
InstanceType	Especifica o tipo da instância que será utilizada no grupo.
region	Informa a região em que o <i>cluster</i> será montado.
cloud_log_dir	Informa o diretório em que serão gravados os arquivos de <i>log</i> .
cloud_tmp_dir	Informa o diretório em que serão gravados os arquivos temporários.
aws_access_key_id	Credencial de acesso ao AWS.
aws_secret_access_key	Credencial de acesso ao AWS.
ec2_key_pair	Nome do par de chave de segurança do EC2.
ec2_key_pair_file	Local em que está armazenado o arquivo de chave privada (.pem)
ssh_tunel	Determina a criação de um túnel SSH se valor for <i>true</i> .
release_label	Refere-se a versão do Amazon EMR que será instalado no <i>cluster</i> . Cada versão corresponde a um conjunto de aplicativos de código aberto no ecossistema de big data <sup>6</sup> .
bootstrap_spark	Informa se o Apache Spark deve ser instalado no cluster.
emr_configurations	Define configurações específicas do <i>cluster</i> EMR.
Classification	Associa a um arquivo de configuração no <i>cluster</i> .
Properties	Relaciona as propriedades do arquivo de configuração definido no campo <i>Classification</i> .
bootstrap	Inclui comandos específicos como instalação de pacotes adicionais e configuração de variáveis de ambiente no <i>cluster</i> .
python_bin	Informa o local em que está o arquivo executável do Python.
py_files	Relaciona arquivos pythons que deverão ser copiados para o <i>cluster</i> .
upload_dirs	Informa diretórios que deverão ser copiados para o <i>cluster</i> .
setup	Permite configurações adicionais no cluster após o processo de instalação ( <i>bootstrapping</i> ).

É importante observar que diversas tarefas precisam ser realizadas para configurar e preparar o *cluster* adequadamente para a execução do método de reconstrução de imagens tomográficas. Tais tarefas podem ser organizadas considerando as seis principais etapas abaixo determinadas:

1. reservar máquinas;
2. configurar o acesso e comunicação;
3. instalar Apache Spark e Hadoop YARN;
4. configurar Apach Spark e Hadoop YARN;

<sup>6</sup> Mais informações em: <[https://docs.aws.amazon.com/pt\\_br/emr/latest/ReleaseGuide/emr-release-components.html](https://docs.aws.amazon.com/pt_br/emr/latest/ReleaseGuide/emr-release-components.html)>

5. instalar bibliotecas adicionais;
6. realizar configurações adicionais no *cluster*.

Visando tratar cada uma das seis etapas, o arquivo de configuração `.mrjob.conf` foi preparado. Ele foi codificado no formato YAML que é o formato principal adotado pela biblioteca. O elemento raiz do arquivo é denominado *runners* que, neste contexto, é responsável por empacotar e enviar o trabalho para ser executado além de reportar os resultados obtidos. Os principais tipos de *runners* disponibilizados pela biblioteca são: Amazon Elastic MapReduce (*emr*), Google Cloud Dataproc (*dataproc*) e *local*.

Durante a elaboração do arquivo de configuração foi adotado o elemento raiz *emr* que indica quais configurações serão aplicadas no Amazon Elastic MapReduce. Na etapa 1, *reservar máquinas*, definiu-se o rótulo do *cluster* em "AgReconSparkCluster", o que facilitou localizá-lo no console AWS. Na sequência, foi configurado o elemento *instance\_groups*, que é composto por outros elementos, que permitiu indicar o tipo de instância utilizada no *cluster*, o papel que determinado grupo de instâncias assumiu, além de definir a quantidade de instâncias do grupo. Para cada configuração de *cluster* foi criado um arquivo `.mrjob.conf`, no qual continha dois grupos: um *master* (*InstanceRole*) contendo uma máquina e o grupo *core* cujo número de máquinas variou em função do tamanho do *cluster*.

Ainda na etapa 1, foram configurados os elementos `max_mins_idle`, `region`<sup>7</sup>, `cloud_log_dir` e `cloud_tmp_dir` definem, respectivamente, tempo máximo em que o *cluster* permanece ligado de modo ocioso, região em que os Nós deverão estar e diretórios para gravação de arquivos temporários e de *logs*. Os elementos `aws_access_key_id`, `aws_secret_access_key`, `ec2_key_pair`, `ec2_key_pair_file`, `ssh_tunnel` trataram da configuração de acesso e comunicação (etapa 2). Por sua vez, a etapa 3 foi determinada pelo elemento `bootstrap_spark`, que quando *True* executou a instalação do Apache Spark e do Hadoop YARN cujas versões são definidas pelo elemento `release_label` que também define a versão do Java JDK. O elemento `bootstrap` indicou os pacotes adicionais que foram instalados (etapa 5) bem como a configuração da variável de ambiente (etapa 6). Por fim, na última etapa, foi incluído, no arquivo de configuração, elementos que permitiram copiar arquivos Python para o *cluster* (`py_files`), além de configurar a variável de ambiente responsável por tornar tais arquivos disponíveis para a execução do método. Vale ressaltar que o conjunto de arquivos copiados para o *cluster* referiam-se ao módulo `ctrecon`. O Código 4, listado no Apêndice A, apresenta o arquivo de configuração `.mrjob.conf` para um *cluster* com três instâncias *core*.

O arquivo de configuração, depois de elaborado, precisa ser acionado para preparar e configurar o *cluster* por meio de duas maneiras possíveis: a primeira por meio do comando `mrjob` que organiza ("cria") o *cluster*; ou, a segunda, ao executar a aplicação. O Código 1 apresenta a

<sup>7</sup> O valor `sa-east-1` indica a região Brasil.

primeira maneira que consiste em criar o *cluster*. No código a informação `/caminho/para` representa o local em que o arquivo `.mrjob.conf` está armazenado.

Código 1 – Criação do cluster pelo MRJob utilizando parâmetro `create-cluster`.

```
1 mrjob create-cluster --conf-path /caminho/para/.mrjob.conf
```

O comando `mrjob` possui o parâmetro `create-cluster` e a opção `-conf-path` na qual é possível indicar a localização do arquivo `.mrjob.conf`. Caso o `mrjob` consiga criar o *cluster* será retornado o identificador, como por exemplo, `j-2ZPNT7F1NH6CM`.

O identificador é importante pois, posteriormente, ele deve ser utilizado para submeter o trabalho para o local correto por meio do parâmetro `-cluster-id`. O Código 2 apresenta a segunda maneira de acionar a criação e preparação do *cluster*. No código a informação `/caminho/para` representa o local em que o arquivo `.mrjob.conf` está armazenado.

Código 2 – Criação do cluster pelo MRJob ao executar a aplicação.

```
1 python ctrecon_spark.py --conf-path /caminho/para/.mrjob.conf -r emr --ctparams=fantoncomRM.
   yml s3://agrecon/dados/fanton
```

Neste caso, a aplicação é acionada primeiro pelo comando `python` seguido pelo código `ctrecon_spark.py`. No entanto, como não foi fornecido o parâmetro `-cluster-id`, a biblioteca MRJob entende que é necessário primeiro criar o *cluster* a partir do arquivo de configuração informando em `-conf-path` para posteriormente executar a aplicação.

### 4.3.2 Configuração do Apache Spark

Na seção anterior discutiu-se a organização e configuração do *cluster* do ponto de vista das instâncias que o compõe bem como das configurações associadas a memória, processamento e armazenamento. Destacou-se, ainda, o emprego da biblioteca MRJob e a elaboração do arquivo de configuração utilizados na preparação do *cluster*. Entretanto, outro aspecto que merece detalhamento são as configurações do framework Apache Spark, cujos parâmetros precisam ser definidos em função dos recursos disponíveis no ambiente Big Data. Portanto, nesta seção, são apresentadas as configurações realizadas em tal framework.

A documentação oficial do Apache Spark disponibiliza informações sobre a configuração<sup>8</sup> e ajustes específicos do framework<sup>9</sup>. O amplo conjunto de parâmetros permite variar a configuração do Apache Spark a fim de preparar o ambiente da maneira mais apropriada para a aplicação. O referido conjunto de parâmetros pode ser organizado em propriedades para diferentes aspectos do ambiente, como: gerenciamento de memória, gerenciador de *cluster* (por exemplo, YARN); segurança; propriedades da aplicação; entre outras.

<sup>8</sup> Disponível em: <<https://spark.apache.org/docs/latest/configuration.html>>

<sup>9</sup> Disponível em: <<https://spark.apache.org/docs/latest/tuning.html>>



Neste trabalho, o objetivo das configurações foi o de otimizar o uso de memória considerando que a reconstrução de imagens tomográficas é entendida como um processo de computação intensiva com alta demanda de memória. Logo, é necessário conhecer a relação entre os parâmetros de configuração associados à memória do framework de modo a definir valores mais apropriados. A Figura 28 apresenta a relação entre os principais parâmetros associados ao gerenciamento de memória na configuração atribuída pelo uso do Apache Spark.

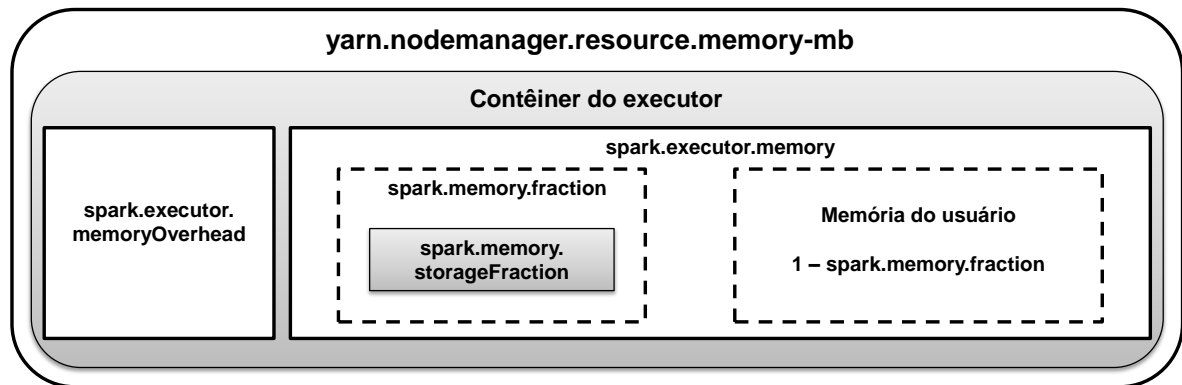


Figura 28 – Relação dos parâmetros de configuração de memória no ambiente Big Data, em configuração Apache Spark.

O parâmetro `yarn.nodemanager.resource.memory-mb` informa ao gerenciador de *cluster*, neste caso o Hadoop YARN, qual o valor total de memória RAM disponível em cada Nó, que é expresso em megabytes. Um Nó, por sua vez, contém um ou mais executores e, portanto, é necessário definir a quantidade de memória que cada executor terá a disposição. A quantidade de memória disponível para cada executor é a composição dos valores definidos pelos parâmetros `spark.executor.memory` e `spark.executor.memoryOverhead`. A memória disponível para execução das tarefas e realização de operações corresponde a uma fração de `spark.executor.memory` cujo valor é definido em `spark.memory.fraction`. Ainda é possível, definir espaço para realizar operações de *cache* e *broadcast*, por meio do parâmetro `spark.memory.storageFraction` que corresponde a uma parcela da fração de memória definida para execução das tarefas.

Neste trabalho foram realizadas diversas configurações de *clusters*, a fim de definir um ambiente mais apropriado para a reconstrução de imagens tomográficas. Um exemplo de configuração é o *cluster* que foi organizado com cinco máquinas, cujo tipo de instância adotado foi o *m5.xlarge*, na qual cada máquina continha 8 vCPU e 32GiB<sup>10</sup> de memória RAM. Do quantitativo de máquinas, uma foi reservada para o Nó mestre (*driver*) e quatro para os Nós trabalhadores. É importante mencionar que neste trabalho adotou-se que 1GiB é igual a 1GB, para efeitos de configuração, de modo que após ajustar os parâmetros do ambiente ainda se tenha

<sup>10</sup> O Gibibyte (GiB) é uma medida de **armazenamento** eletrônico de informação, estabelecida pela Comissão Eletrotécnica Internacional, e representa  $2^{30}$ , ou 1.073.741.824 bytes. Por outro lado, Gigabyte (GB) é uma unidade de medida de **informação** segundo o Sistema Internacional de Unidades e equivale a  $10^9$ , ou 1.000.000.000 bytes. Portanto, 1 GiB representa 1,07374 GB.



reserva de bytes para eventuais condições não previstas. O Quadro 6 relaciona parâmetros e valores associados à configuração de memória do Apache Spark.

Quadro 6 – Parâmetros e valores adotados para configuração da memória do Apache Spark.

Parâmetro	Valor
<code>spark.executor.cores</code>	5
<code>spark.executor.instances</code>	4
<code>spark.executor.memory</code>	28 GB
<code>spark.executor.memoryOverhead</code>	4 GB
<code>spark.memory.fraction</code>	0.9
<code>spark.memory.storageFraction</code>	0.1
<code>yarn.nodemanager.resource.memory-mb</code>	32768 MB
<code>spark.default.parallelism</code>	40

O primeiro passo para realizar a configuração de memória foi definir o número de *cores* (vCPU) para o parâmetro `spark.executor.cores` que indica o número máximo de tarefas concorrentes que um executor pode executar. Não foi encontrado na literatura um método objetivo para determinar o número de vCPU para cada executor, mas foi observado que na prática tem se adotado o valor 5.

Na sequência foi definido o número de executores por instância obtido pelo cálculo  $\lfloor (vCPU - 1) \div cores \rfloor$ , no qual reserva pelo menos uma vCPU para outros processos do ambiente. O valor do parâmetro `spark.executor.instances` foi definido pela multiplicação do número de Nós trabalhadores pelo número de executores por instância. O valor do parâmetro `spark.executor.memory` foi obtido a partir do total de memória dividido pelo número de executores por instância multiplicado por 90%, ou seja,  $\lfloor (32 \div 1) \times 0,90 \rfloor = 28$ , logo chegando ao valor do parâmetro em 28GB.

O valor do parâmetro `spark.executor.memoryOverhead` foi calculado como a diferença entre o total de memória e o valor definido em `spark.executor.memory`, garantindo que seja no mínimo 384 MB. A partir dos parâmetros definidos derivou a configuração do número total de partições (`spark.default.parallelism`), calculado como o dobro do número de *cores* multiplicado pelo total de executores ( $4 \times 5 \times 2$ ).

O parâmetro `spark.dynamicAllocation.enabled` que é utilizado para realizar a alocação dinâmica dos recursos, bem como o parâmetro `maximizeResourceAllocation`, foram desabilitados para que os ajustes definidos fossem aplicados ao ambiente Big Data. Além disso, as seguintes opções de compressão foram ativadas, visando melhorar os aspectos da comunicação de dados entre os Nós do *cluster*:

- `spark.rdd.compress`;
- `spark.shuffle.compress`;
- `spark.shuffle.spill.compress`, e;

- `mapreduce.map.output.compress`.

Já o parâmetro `spark.storage.level` foi alterado para `MEMORY_AND_DISK_SER`, visando priorizar e manter os dados em memória e, se necessário, utilizar o disco para concluir o processamento.

## 4.4 Armazenamento das matrizes de projeções tomográficas

As matrizes de projeções foram obtidas utilizando um tomógrafo SkyScan 1172 cuja geometria é *cone-beam*. Por outro lado, o algoritmo de reconstrução utilizado neste trabalho (FBP - *Filtered BackProjection*) considera a geometria paralela. Na geometria *cone-beam* o objeto é escaneado em todo o eixo-*z*, além disso ele é rotacionado no intervalo 0° a 180°, ou 0° a 360°, de acordo com um passo angular pré-determinado. Na geometria paralela ocorre a rotação similar à *cone-beam*, no entanto há um deslocamento no eixo-*z* (HSIEH, 2009).

Portanto, as matrizes passaram por um pré-processamento visando prepará-las para a execução do algoritmo FBP que consistiu em: (i) organizar as projeções para geometria paralela; (ii) corrigir o alinhamento das projeções. Na primeira etapa, as matrizes denominadas "matrizes *cone-beam*" foram gravadas em arquivos no formato `.tif` e, assim, a quantidade de arquivos indicavam o número de rotações realizadas em função do passo angular. O número de linhas de cada matriz obtidas na geometria *cone-beam* indicavam quantos sinogramas foram obtidos no eixo-*z*. Portanto, para organizar as projeções para geometria paralela, as matrizes *cone-beam* foram lidas e as projeções que estavam na mesma posição nas diferentes matrizes foram copiadas para uma nova matriz.

É importante mencionar que a matriz gerada pelo tomógrafo ao ser salva registrou, no nome do arquivo, uma numeração que, ao ser associada ao passo angular, tornou possível identificar o ângulo em que a matriz foi escaneada. Esta informação foi importante para reposicionar as projeções na nova matriz, denominada "matriz paralela". Portanto, a tarefa consistiu em organizar  $A$  matrizes de  $(F \times N)$  em  $F$  matrizes de  $(A \times N)$ , onde  $A$  representa o número de ângulos escaneados, determinado em função do passo angular,  $F$  representa o número de sinogramas e  $N$  é o número de pontos por projeção. Na segunda tarefa foi realizada uma filtragem nas projeções visando minimizar os efeitos do desalinhamento produzido pelo fato das projeções, originalmente, terem sido adquiridas na geometria *cone-beam*.

Após preparadas, as matrizes paralelas foram armazenadas no sistema de arquivos distribuído da Amazon, AWS S3. As matrizes foram armazenadas em pastas cujos nomes se referiam às amostras as quais elas pertenciam. Além disso, o nome do arquivo de cada matriz continha o nome da amostra e o sinograma a qual ela representava.

As matrizes foram enviadas para o AWS S3 por meio do console AWS. Além da informação presente no nome do arquivo de cada matriz, foi preparado um modelo de arquivo de

configuração para cada amostra com informações adicionais utilizadas pelo método desenvolvido. O Quadro 7 relaciona os parâmetros elaborados para o método.

Quadro 7 – Parâmetros de configuração da aplicação.

Parâmetro	Descrição
Nome	nome da amostra.
Passo	passo angular utilizado durante aquisição da amostra.
Primeira seção	primeira seção a ser reconstruída.
Última seção	última seção a ser reconstruída.
Dimensão	tupla (linhas, colunas) indicando a dimensão das matrizes de projeções.
Taxa	taxa de seleção das projeções.
Seleção por classe	indica se a seleção das projeções será por classes ou geral.
Output size	fornece o tamanho da matriz quadrada correspondente a seção reconstruída.
Intervalo de reconstrução	tupla (início, fim) que define o intervalo de reconstrução a partir dos ângulos inicial e final
Tamanho da janela (tam_janela)	número de pontos que define o tamanho da janela quadrada utilizada para a reconstrução volumétrica.
njanelas_linhas	número de janelas por linha.
njanelas_colunas	número de janelas por coluna.
Interpolação B-spline	indica quantos planos virtuais serão criados durante a reconstrução volumétrica. Por exemplo, 2 indica que a quantidade de planos reais será dobrado.

O arquivo foi preparado no formato YAML e procurou seguir o mesmo formato utilizado pela biblioteca MRJob. O Código 5, listado no Apêndice A apresenta um arquivo de configuração da amostra denominada *Phanton*.

O arquivo de configuração armazena o nome da amostra, o passo angular, a primeira e última seção, a taxa a ser adotada na tarefa de seleção das projeções, se a seleção será por classes, o tamanho (`output_size`) da matriz quadrada resultante da reconstrução bidimensional. Além disso, são registrados o intervalo, tipo de interpolação e filtro adotados na reconstrução 2D; o tamanho da janela e o número de janelas por linha e coluna para formar os *tiles* na reconstrução volumétrica e o valor utilizado na interpolação B-spline para determinar o número de pontos a serem interpolados.

Os parâmetros do arquivo foram definidos considerando informações obtidas da aquisição feita pelo tomógrafo bem como de informações do próprio método.

## 4.5 Seleção de projeções baseada em densidade espectral

Nesta seção é realizada uma discussão sobre densidade espectral de potência que foi utilizada para calcular a energia das projeções. Na sequência, é apresentado o algoritmo que realiza a seleção das projeções.

### 4.5.1 Densidade espectral de potência

Estimar a densidade espectral de potência (PSD) de um sinal, geralmente, é resolvido estimando-se a função de autocorrelação com os dados disponíveis, à qual se aplica em seguida a Transformada de Fourier para obter a descrição espectral desejada. No entanto, há diferentes abordagens para realizar a estimação espectral as quais podem ser classificadas como: métodos paramétricos ou não-paramétricos.

O primeiro tipo, em geral, é mais simples de se calcular porém necessita de conhecimento *a priori* do sinal, enquanto que o segundo tipo não assume nenhuma estrutura particular por trás dos dados disponíveis (OPPENHEIM; SCHAFER, 1975; DINIZ; SILVA; NETTO, 2010).

Dado um sinal aleatório no domínio do tempo,  $\mathcal{X}(t)$ , assume-se que ele está amostrado sobre um intervalo finito de tempo  $(-T/2, T/2)$  e é denotado por  $X_T(t)$ . Ao aplicar a transformada de Fourier tem-se:

$$\tilde{X}_T(f) = F\{X_T(t)\} = \int_{-\infty}^{\infty} X_T(t)e^{-2\pi jft} dt = \int_{-T/2}^{T/2} \mathcal{X}(t)e^{-2\pi jft} dt \quad (4.7)$$

Da Equação 4.7 tem-se que o módulo e o argumento de  $\tilde{X}_T$  são, respectivamente, o *espectro de amplitude* e o *espectro de fase*. Por sua vez, a *densidade espectral de energia* é calculada a partir de  $\tilde{X}_T$  por meio do valor esperado do quadrado do espectro de amplitude, como indicado na Equação 4.8.

$$\Gamma(f) = \mathcal{E}\{|\tilde{X}_T(f)|^2\} \quad (4.8)$$

Observa-se que  $\Gamma(f)$  tende ao infinito quando  $T$  tende ao infinito. Logo, dividir a Equação 4.8 pelo intervalo de  $T$  limita o crescimento e fornece a *densidade do espectro de potência* expressa pela Equação 4.9, que é real, não negativa. Esta definição é válida e existe para todos processos estacionários com média zero e variância finita. Para a tomografia agrícola, as amostras a serem ensaiadas não sofrem movimentos em relação à mesa tomográfica, permanecem estacionárias durante o processo envolvido na aquisição das projeções, portanto trata-se de um processo onde se pode utilizar essa teoria (BEUTLER; LENEMAN, 1966). Adicionalmente, como o ruído de Poisson é prioritário no processo tomográfico, é considerado também que o mesmo tem comportamento estacionário ao longo do processo tomográfico.

$$\Gamma(f) = \lim_{T \rightarrow \infty} \mathcal{E} \left\{ \frac{1}{T} \left| \int_{-T/2}^{T/2} \mathcal{X}(t)e^{-2\pi jft} dt \right|^2 \right\} \quad (4.9)$$

No caso discreto, considerando a sequência  $x[n]$ , tem-se a Equação 4.10 na qual  $\hat{\Gamma}$  é o estimador por periodograma<sup>11</sup>. Isso é equivalente a aplicar uma janela retangular sobre o intervalo  $0 \leq n \leq (T - 1)$  da sequência  $x[n]$ , elevar ao quadrado o módulo da transformada de

<sup>11</sup> Periodograma é um método de estimativa da densidade espectral de um sinal.

Fourier da sequência truncada e normalizar o resultado por um fator  $T$ , para obter uma medida de densidade espectral de potência.

$$\hat{\Gamma}(e^{jf}) = \frac{1}{T} \left| \sum_{n=0}^{T-1} x[n]e^{-jfn} \right|^2 \quad (4.10)$$

A seguir é apresentado o modelo desenvolvido para seleção das projeções tomográficas utilizado em uma das etapas do método de reconstrução tomográfica.

#### 4.5.2 Modelo de seleção das projeções tomográficas

Tomando por base a densidade espectral de cada projeção tomográfica presente em um sinograma considerado, foi avaliada a energia de cada projeção de forma a se buscar reconhecer aquelas portadoras de um conjunto mais relevante de informações para a obtenção da reconstrução tomográfica em duas dimensões.

Neste contexto, a informação sobre a densidade espectral de cada projeção tomográfica pode ser obtida considerando o espectro de potência a ela relacionado.

Considerando, por outro lado, um sinal  $s = s(t)$ , contínuo no tempo, como uma função que representa um sinal aleatório e  $\Gamma = \Gamma(\omega)$ , uma função que representa o periodograma do referido sinal é possível decompor  $\Gamma$  na forma:

$$\Gamma = \Gamma_r + j\Gamma_i \quad (4.11)$$

onde  $\Gamma_r$  e  $\Gamma_i$  são as partes real e imaginária, respectivamente, e  $j = \sqrt{-1}$ . Esta equação ainda pode ser escrita na forma polar:

$$\Gamma = |\Gamma|e^{j\theta(p)} \quad (4.12)$$

Portanto, as amplitudes no espectro traduzidas pela Equação 4.12 podem ser dadas por:

$$|\Gamma| = \sqrt{\Gamma_r^2 + \Gamma_i^2} \quad (4.13)$$

Desta forma, a partir daí é possível considerar a própria Equação 4.10 quando se trabalha com uma projeção tomográfica de raio-X, a qual é tratada como uma sequência  $s[n]$ , uma vez que o sinal é discretizado, ou seja:

$$\xi_m = \hat{\Gamma}(e^{jf}) = \frac{1}{N} \left| \sum_{n=0}^{N-1} s[n]e^{-jfn} \right|^2 \quad (4.14)$$

onde  $n$  está no intervalo  $0 \leq n \leq (N - 1)$  e representa o número de amostras na sequência  $s[n]$ .

Assim, com base na Equação 4.14, as energias das projeções tomográficas ( $\xi_m$ ) podem ser calculadas, considerando que  $m$  está no intervalo  $0 \leq m < M$  e que  $M$  representa o número de projeções do sinograma.

Neste contexto, o conjunto de projeções tomográficas que compõe um sinograma é entendido com sendo um conjunto de energias  $\Xi = \{\xi_0, \xi_1, \xi_2, \dots, \xi_{M-1}\}$ , em que cada energia  $\xi_m$ , onde  $m = 0, 1, 2, \dots, M - 1$ , representa uma projeção.

A partir do conjunto de energias  $\Xi$ , são definidas as classes de modo a constituir o conjunto de classes  $C_T = \{C_0, C_1, C_2, \dots, C_{\kappa-1}\}$ , em que uma determinada classe  $C_i$ , para  $i = 0, 1, \dots, \kappa - 1$ , representa um subconjunto de energias contidas em  $\Xi$ .

A partir do conjunto de energias  $\Xi$  e do número de projeções contidas no sinograma em questão é utilizada a Equação 4.15 para se determinar o número de classes,  $\kappa$ , contidas nesse conjunto de energias (MORETTIN; BUSSAB, 2017).

$$\kappa = \lfloor \sqrt{M} \rfloor \quad (4.15)$$

onde  $M$  representa o número de projeções tomográficas contidas em um sinograma. A função piso, denotada por  $\lfloor x \rfloor$  converte um número real  $x$  no maior número inteiro menor ou igual a  $x$ , que neste caso refere-se ao número de classes que serão definidas para o sinograma considerado.

O intervalo<sup>12</sup>  $\Delta$ , de energias associado a cada classe, é expresso pela Equação 4.16 que leva em consideração a maior e a menor energia encontradas no conjunto de energias  $\Xi$ , bem como o número de classes definido pela Equação 4.15.

$$\Delta = \frac{\max \Xi - \min \Xi}{\kappa} \quad (4.16)$$

Logo, cada classe possui uma energia inicial,  $\xi_{s_i}$ , e uma energia final,  $\xi_{t_i}$ , de modo que  $C_i = [\xi_{s_i}, \xi_{t_i})$ . A energia inicial de uma classe é dada pela Equação 4.17.

$$\xi_{s_i} = \begin{cases} \min \Xi & \text{se } i = 0, \\ \xi_{t_{i-1}} & \text{c.c.} \end{cases} \quad (4.17)$$

A energia final de uma classe, por sua vez, é dada pela Equação 4.18.

$$\xi_{t_i} = \begin{cases} \xi_{s_i} + \Delta & \text{se } i < (\kappa - 1), \\ \max \Xi & \text{c.c.} \end{cases} \quad (4.18)$$

Após definidas as classes e os intervalos de energias, as projeções tomográficas são classificadas de acordo com os valores de energia. Foi considerado no desenvolvimento do

<sup>12</sup> O intervalo de energia também pode ser chamado de *amplitude* da classe.

modelo a distribuição Gaussiana. Neste contexto, assumiu-se que as classes encontradas seguiram este modelo de distribuição. Portanto, após a classificação das projeções, são calculadas as médias ( $\mu_0, \mu_1, \dots, \mu_{\kappa-1}$ ) e o desvio padrão ( $\sigma$ ) de cada classe.

Shannon mostrou que a informação é algo que pode ser quantificado e que a quantidade de informação está relacionada à sua probabilidade (SHANNON, 1948; SHANNON, 1949; VERDU, 1998). Neste sentido, o critério de seleção consistiu em escolher dentro de cada classe de energia as probabilidades mais significativas, as quais traduzem as projeções tomográficas que apresentam maior quantidade de informações. Logo, são consideradas significativas as projeções que contém energia dentro do intervalo de um desvio padrão ( $[-\sigma, \sigma]$ ) em cada classe, o que leva a formação dos conjuntos  $C_i^{sel}$ , para  $i = 0, 1, \dots, \kappa - 1$ , os quais contém as projeções selecionadas para cada faixa de energia associada às classes. Portanto, são consideradas as projeções tomográficas que carregam maior densidade espectral de energia ou maior densidade de informações sobre o conjunto das projeções que compõe um determinado sinograma.

Na sequência, as projeções tomográficas identificadas como mais significativas em cada classe, tomando por base a informação da energia, são reunidas para formar um novo sinograma composto por um número menor de projeções em relação ao sinograma original considerado. O novo sinograma refere-se ao conjunto  $\hat{\zeta} = \{C_0^{sel}, C_1^{sel}, \dots, C_{\kappa-1}^{sel}\}$ . Além disso, as projeções tomográficas contidas no novo sinograma,  $\hat{\zeta}$ , são organizadas em função do ângulo em que foram adquiridas a fim de se preparar para a etapa de reconstrução em duas dimensões. A Figura 29 ilustra a representação conceitual das classes de energia.

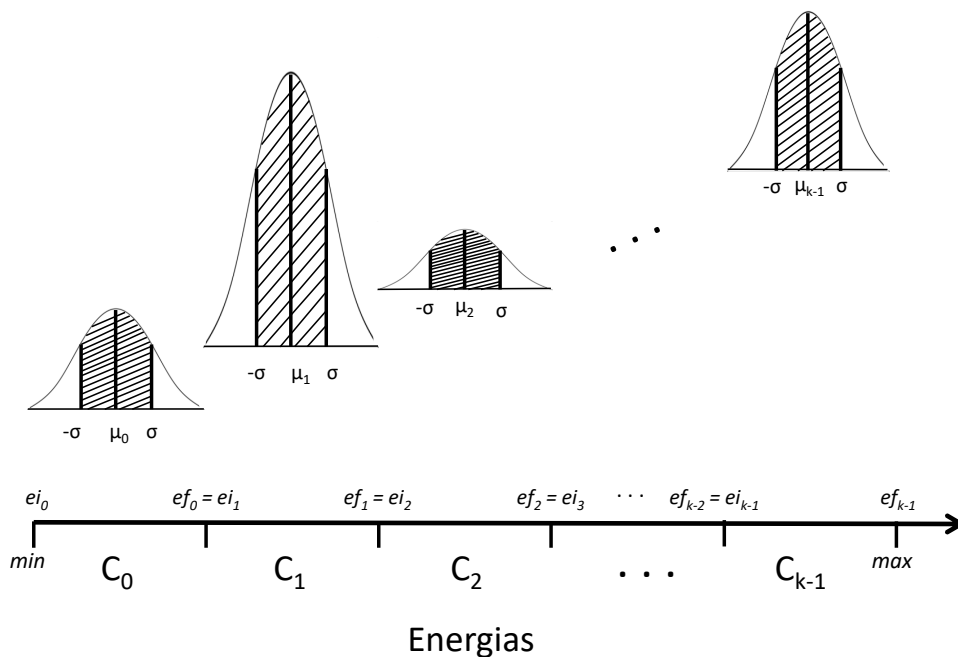


Figura 29 – Representação conceitual das classes de energia considerando a distribuição Gaussiana em cada classe. A região hachurada corresponde a região de cada classe em que se encontram as projeções mais significativas.

Na Figura 29 é possível observar as classes  $C_i$  e as energias inicial e final, bem como a distribuição Gaussiana associada a cada classe. A região hachurada indica a região em que serão identificadas as projeções mais significativas de cada classe.

Na próxima seção é apresentado o algoritmo para seleção de projeções tomográficas, o qual considera a energia das projeções para determinar quais serão escolhidas.

### 4.5.3 Algoritmo para seleção de projeções

A Figura 30 apresenta o fluxograma do algoritmo responsável pelo processo de seleção de projeções. Uma matriz  $N \times M$  que contém um conjunto de projeções  $p_{\theta_1}, p_{\theta_2}, \dots, p_{\theta_N}$ , na qual cada projeção contém  $M$  pontos, foi submetida ao algoritmo que se inicia com a leitura das projeções.

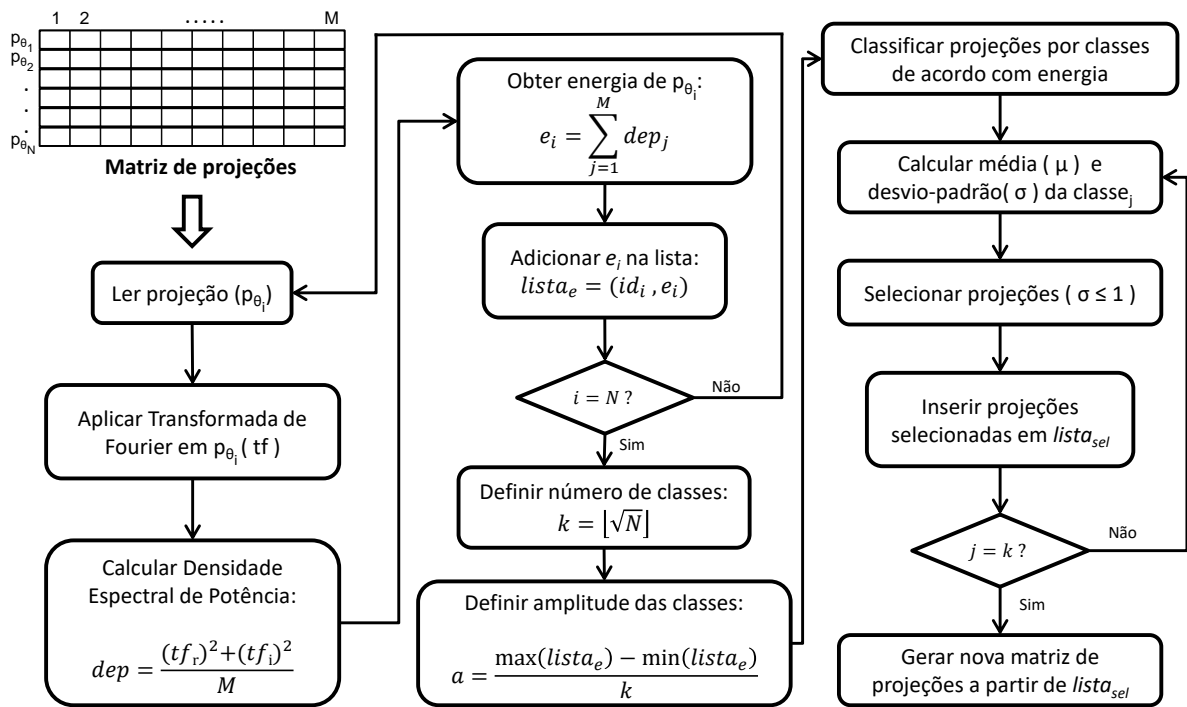


Figura 30 – Fluxograma do processo de seleção de projeções utilizando a densidade espectral.

Para cada projeção, calculou-se a Transformada de Fourier, a densidade espectral de potência  $e$ , conseqüentemente, a energia. Posteriormente, a energia calculada foi armazenada em um lista juntamente com o identificador da projeção, que neste caso é o ângulo a qual ela se referia. O valor do ângulo foi obtido por meio da multiplicação da posição da projeção na matriz pelo passo angular da aquisição. Por exemplo, uma matriz de projeções escaneada no intervalo de  $0^\circ$  a  $180^\circ$  ao passo angular de  $0,2^\circ$  gerou 90 projeções, logo a projeção armazenada na linha 2 da matriz corresponde ao ângulo  $0,4^\circ$ .

Como observado na Figura 30, após calcular a energia espectral para todas as projeções e gerar a lista de energias ( $lista_e$ ), foi definido o número de classes bem com a amplitude de



energias de cada classe. Na sequência as projeções foram classificadas de acordo com a energia. Para cada classe de energia, calculou-se a média ( $\mu$ ) e o desvio-padrão ( $\sigma$ ) e as projeções cuja energia estivessem no intervalo de um desvio-padrão foram selecionadas e uma nova matriz de projeções foi gerada a partir do subconjunto das projeções selecionadas.

Finalmente, vale ressaltar que o tempo de processamento gasto para seleção compõe o tempo de processamento gasto para realizar a reconstrução bidimensional. É importante mencionar ainda que a seleção das projeções tomográficas está inserida em um contexto em que as matrizes são compostas por projeções, na ordem de milhares, e que a amostra em análise possui matrizes, também na ordem de milhares. Portanto, a seleção das projeções busca uma boa relação custo-benefício, na qual busca-se reduzir o tempo de processamento bem como a quantidade de dados e garantindo boa qualidade na reconstrução.

## 4.6 Método de reconstrução tomográfica utilizando abordagem MapReduce

Nesta seção é apresentado o método para reconstrução tomográfica 2D e 3D (volumétrica) de amostras agrícolas, estruturado e desenvolvido sob a perspectiva do modelo de programação MapReduce, conforme ilustra o diagrama da Figura 31.

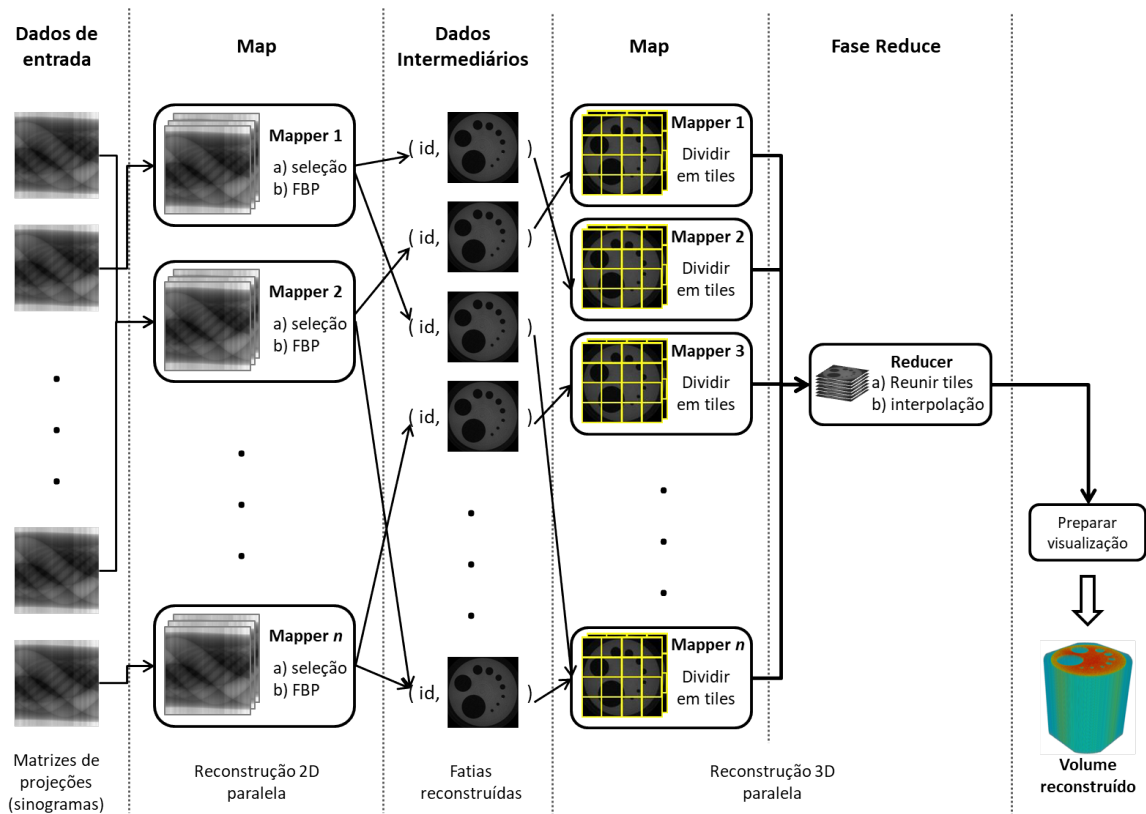


Figura 31 – Método de reconstrução tomográfica sob a perspectiva do modelo de programação MapReduce.

Considere  $\varsigma_i$  uma matriz de projeções tomográficas, também denominada de sinograma, onde  $i = 1, 2, \dots, T$ , e  $A = \{\varsigma_1, \varsigma_2, \dots, \varsigma_T\}$  o conjunto de sinogramas de uma amostra agrícola. Cada sinograma,  $\varsigma_i$ , possui  $M$  projeções tomográficas na qual cada projeção possui  $N$  pontos, ou seja, a dimensão de um sinograma é  $M \times N$ .

A primeira etapa do método consiste na leitura dos dados de entrada, que correspondem as matrizes de projeções tomográficas  $\varsigma_i$ . Os sinogramas são então distribuídos para as tarefas *mappers*, na segunda etapa que trata da primeira fase de mapeamento. Os *mappers*, nesta etapa, são responsáveis por realizarem a seleção das projeções dos sinogramas e realizarem a reconstrução bidimensional. Como resultado, são gerados na terceira etapa pares (*chave, valor*) em que a chave corresponde a posição da fatia no eixo- $z$  e o valor refere-se a fatia reconstruída.

A quarta etapa corresponde a segunda fase de mapeamento em que as fatias reconstruídas são divididas em regiões (*tiles*). Por fim, a última etapa refere-se a redução que consiste em reunir os *tiles* e realizar a interpolação B-Spline de modo a obter os subvolumes que formarão o volume. É importante mencionar que a preparação da visualização do volume reconstruído ocorre fora do ambiente Big Data.

A seguir as reconstruções bidimensional e volumétrica que compõem o método são apresentadas em detalhes.

#### 4.6.1 Reconstrução tomográfica bidimensional paralela

Após a leitura dos dados de entrada inicia-se a etapa em que ocorre o mapeamento das tarefas de seleção e de reconstrução tomográfica bidimensional das matrizes de projeções tomográficas, cujo resultado são as imagens tomográficas reconstruídas. Considere, portanto,  $I_i$  a imagem reconstruída que é obtida por meio do algoritmo de retroprojeção filtrada (FBP) a partir do sinograma  $\varsigma_i$  onde  $i = 1, 2, \dots, T$ .

De forma a paralelizar o processo de reconstrução são mapeados subconjuntos de matrizes  $\varsigma_i$ , contidas no conjunto  $A$ , tal que  $B_k = \{\varsigma_1, \varsigma_2, \dots, \varsigma_{k/T}\}$ . A Figura 32 apresenta o diagrama de blocos que elucida o processo de paralelização da reconstrução bidimensional.

Conforme ilustrado na Figura 32, o processo de paralelização consiste em dividir o conjunto  $A$  em subconjuntos,  $B_k$ , os quais são distribuídos para os Nós do *cluster*. Cada Nó, inicialmente seleciona as projeções tomográficas de cada sinograma  $\varsigma_i$ , com base na Equação 4.14, o que possibilita a geração do sinograma  $\hat{\varsigma}_i$  com as projeções selecionadas.

Na sequência, os sinogramas  $\hat{\varsigma}_i$  são reconstruídos com base na Transformada de Radon, expressa pela Equação 2.10. Conforme discutido no Capítulo 2, a reconstrução tomográfica bidimensional a partir de um conjunto de projeções com base no Teorema das Seções de Fourier ocorre ao aplicar a Transformada de Fourier unidimensional nas projeções em diferentes ângulos para encontrar uma aproximação da Transformada de Fourier bidimensional do objeto.

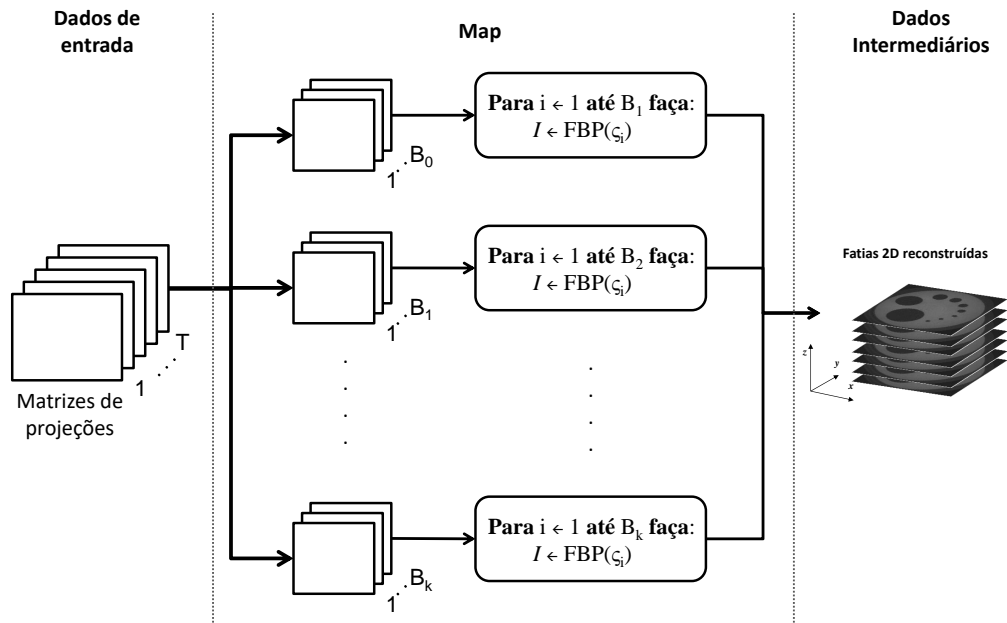


Figura 32 – Diagrama de blocos da reconstrução 2D paralela. As linhas pontilhadas indicam as etapas do método sob a perspectiva do modelo de programação MapReduce.

O Algoritmo 1 apresenta o FBP (*Filtered Back Projection*) que é uma alternativa de implementação bastante utilizada.

---

**Algoritmo 1:** *Filtered BackProjection* (FBP)

---

**Entrada:** Matriz de projeções  $\hat{\xi}_i$

**Saída:** Imagem reconstruída  $I$

- 1  $I \leftarrow \emptyset$  ;
  - 2 **para cada**  $P_\theta \in S$  **faça**
  - 3      $P_{filtrada} \leftarrow \text{filtrar}( P_\theta )$  ;
  - 4      $tmp \leftarrow \text{interpolar}( P_{filtrada} )$  ;
  - 5      $I \leftarrow I + tmp$  ;
  - 6 **fim**
- 

Após a filtragem das projeções que compõem a matriz  $\hat{\xi}_i$ , o processo de reconstrução pode ser iniciado e a interpolação no domínio do espaço é realizada. Na sequência, a etapa de retroprojeção é responsável pela somatória das projeções filtradas e interpoladas a fim de computar a contribuição de cada projeção no pixel da imagem reconstruída. A imagem reconstruída  $I_i$ , a partir da matriz  $\hat{\xi}_i$ , também pode ser referida como fatia.

Ao final do processo os dados intermediários são organizados em uma lista de pares  $\langle chave, valor \rangle$ , em que cada *chave* refere-se à posição no eixo- $z$  (também tratada como identificador) da imagem reconstruída  $I$  armazenada no campo *valor* do par. A lista é disponibilizada para a etapa seguinte do método responsável pela reconstrução 3D (volumétrica).

## 4.6.2 Reconstrução 3D (volumétrica) paralela

A etapa de reconstrução volumétrica, desenvolvida neste trabalho, utiliza o conjunto  $F = \{I_1, I_2, \dots, I_T\}$  de imagens reconstruídas, também denominado de conjunto de fatias reais. Além disso, é gerado o conjunto  $\hat{F} = \{\hat{I}_1, \hat{I}_2, \dots, \hat{I}_U\}$  de fatias virtuais que são obtidas por meio de interpolação B-spline, expressa pela Equação 2.19. O Algoritmo 2 estrutura as principais tarefas desta etapa.

---

### Algoritmo 2: Reconstrução volumétrica

---

**Entrada:** Conjunto de fatias  $F$

**Saída:** Volume reconstruído  $V$

```

1 para cada posição  $(x, y) \in F$  faça
2     pontos  $\leftarrow I_0, I_1, \dots, I_T$  ;
3     novos_pontos  $\leftarrow$  bspline( pontos ) ;
4      $V(x, y) \leftarrow$  novos_pontos ;
5 fim
    
```

---

De modo geral, as fatias são empilhadas e para cada posição forma-se um conjunto de pontos que serão interpolados gerando o conjunto de voxels. Portanto, o volume final,  $V = F \cup \hat{F}$ , é formado a partir da união das fatias reais e fatias virtuais. Vale lembrar que o total de fatias reais  $T$ , no conjunto  $F$ , representa o número de sinogramas obtidos de uma amostra agrícola. Por outro lado, o total de fatias virtuais  $U$ , no conjunto  $\hat{F}$ , é dado em função de  $T$  e do parâmetro de configuração  $P_b$ , que indica o número de planos (fatias) virtuais a serem geradas entre duas fatias reais, portanto  $U = (T - 1) \times P_b$ . Logo, a organização da fatias reais e virtuais no conjunto  $V$  é dada pela Equação 4.19.

$$V = \{I_1, \hat{I}_{1+\frac{U}{T-1} \times 0}, \dots, \hat{I}_{\frac{U}{T-1} + \frac{U}{T-1} \times 0}, I_2, \dots, I_{T-1}, \hat{I}_{1+\frac{U}{T-1} \times (T-2)}, \dots, \hat{I}_{\frac{U}{T-1} + \frac{U}{T-1} \times (T-2)}, I_T\} \quad (4.19)$$

Considerando que as fatias reconstruídas, neste trabalho, podem ter dimensões superiores a  $1000 \times 1000$  pixels, adotou-se a estratégia de paralelização que consistiu em dividir as fatias em pequenas regiões (*tiles*) e aplicar o processo de interpolação a um subconjunto de pontos. Vale lembrar que uma imagem reconstruída  $I_i$  possui dimensão  $M \times N$ , onde  $M$  é o número de linhas e  $N$  o número de colunas da imagem. Portanto, seja  $\tau_i$  um *tile* de dimensão tal que  $\tau_i = \frac{M}{L} \times \frac{N}{C}$ , onde  $L$  indica o número de linhas e  $C$  é o número de colunas, pode-se reescrever  $I_i$  em função dos *tiles*, de modo que  $I_i = \{\tau_{i;1,1}, \tau_{i;1,2}, \dots, \tau_{i;L,C}\}$ . Logo, a partir da Equação 4.19 é possível definir o subvolume  $V_{lc}$  como o volume dos *tiles* que estejam na mesma linha  $l$  e coluna  $c$ , como expressa a Equação 4.20.

$$V_{lc} = \{\tau_{1;lc}, \dots, \tau_{T-1;lc}, \hat{\tau}_{1+\frac{U}{T-1} \times (T-2);lc}, \dots, \hat{\tau}_{\frac{U}{T-1} + \frac{U}{T-1} \times (T-2);lc}, \tau_{T;lc}\} \quad (4.20)$$

Neste caso o volume final pode ser reescrito como  $V = \{V_{11} \cup V_{12} \cup \dots \cup V_{LC}\}$ . A Figura 33 apresenta o diagrama de blocos da reconstrução volumétrica paralelizada utilizada neste trabalho.

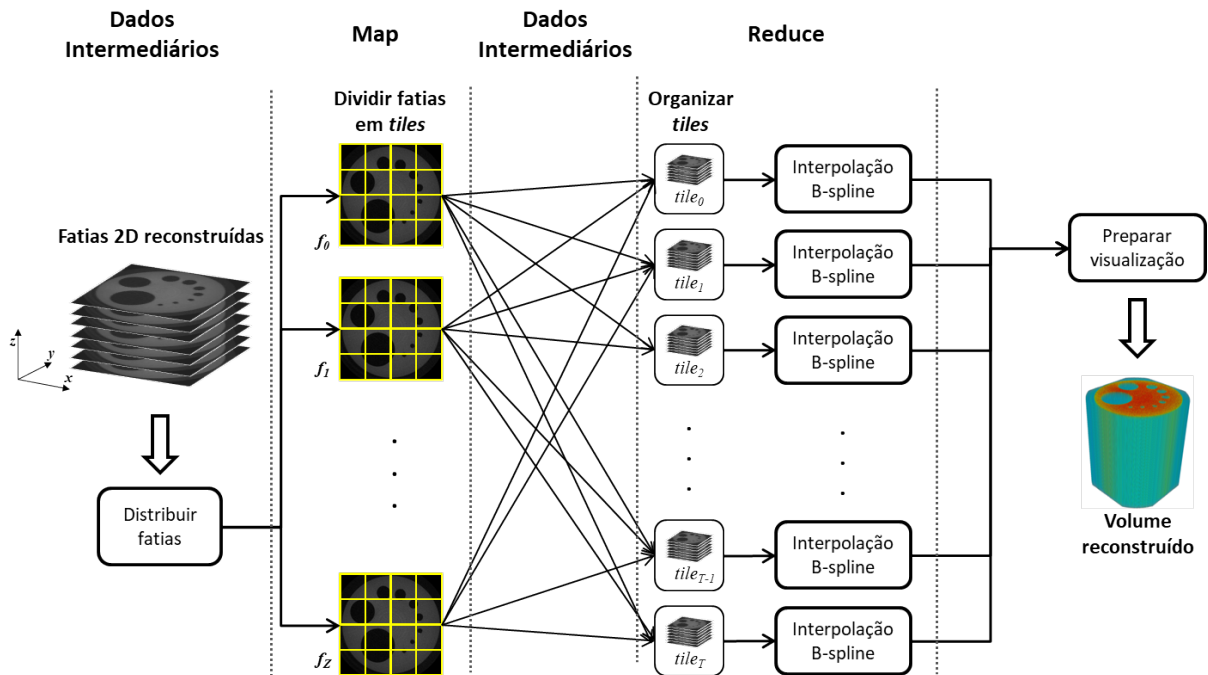


Figura 33 – Diagrama de blocos da reconstrução volumétrica paralela, considerando um exemplo do processamento de um *phantom*. As linhas pontilhadas indicam as etapas do método sob a perspectiva do modelo de programação MapReduce.

A primeira tarefa, após distribuir as fatias, consiste em dividir cada fatia em regiões, ou *tiles*. Uma região recebe uma identificação que será necessária para a reconstrução final do volume. Logo, cada fatia é dividida em linhas e colunas cujos identificadores são utilizados para identificar uma determinada região conforme ilustra a Figura 34.

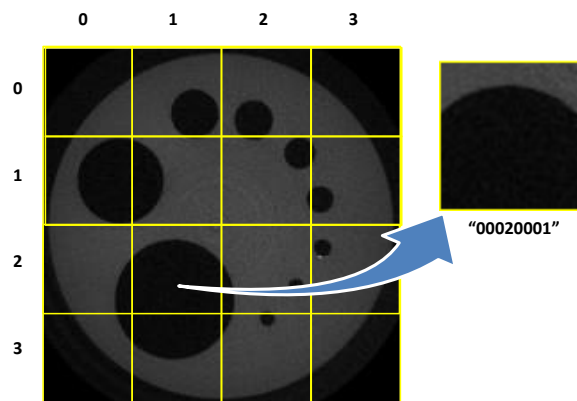


Figura 34 – Identificação de uma região (*tile*) em uma fatia. No detalhe, a região e o respectivo identificador composto por oito caracteres.

O identificador é formado por oito caracteres, no qual os quatro primeiros indicam a linha e os demais a coluna. Além do identificador é necessário incluir a seção (posição no eixo-*z*) e o conjunto de pixels associados a região. O resultado retornado pela tarefas de dividir a fatia em regiões possui o seguinte formato: (<id-regiao>, (<secao>, <tile>)). O formato é uma tupla, na qual o primeiro valor é o identificador da região e o segundo valor é uma tupla

contendo a seção e o conjunto de pixels. Em termos de implementação o conjunto de pixels da região foi armazenado em um objeto da classe `DenseMatrix`, disponibilizada pelo *PySpark* que otimiza a transmissão no ambiente Spark.

A próxima tarefa da reconstrução volumétrica consiste em agrupar as regiões por identificador, desse modo cada subconjunto contém todas as fatias de uma determinada região. Portanto, esta tarefa pode ser considerada como uma operação de redução intermediária. Em termos de implementação foi utilizado o método `combineByKey` que permitiu realizar tal agrupamento.

A penúltima etapa consiste em realizar a interpolação B-spline de cada subconjunto e, finalmente, na última etapa os dados são disponibilizados para visualização tridimensional. Considerando que, neste trabalho, a visualização foi planejada para ser realizada fora do ambiente Big Data, os dados finais da reconstrução volumétrica são salvos em arquivos para posterior utilização.

No próximo capítulo são apresentados os resultados e discussões obtidos pelo método para reconstrução de imagens tomográficas de amostras agrícolas com o emprego de Big Data.

# Capítulo 5

## RESULTADOS E DISCUSSÕES

---

---

Neste capítulo são apresentados os resultados obtidos e discussões visando a validação do método desenvolvido. A primeira parte do capítulo refere-se à análise da infraestrutura Big Data, em que se buscou identificar a configuração mais apropriada para a execução da reconstrução tomográfica. Na segunda parte, são apresentados os resultados obtidos no processo da seleção de projeções baseadas na densidade espectral. Na última parte do capítulo é apresentada a avaliação de uma segunda abordagem de paralelização para reconstrução 2D bem como a visualização 3D (volumétrica) das amostras de sementes agrícolas.

### 5.1 Análise da infraestrutura para o método de reconstrução tomográfica com Big Data

Para efeitos de análise da infraestrutura Big Data, buscou-se avaliar a configuração mais apropriada de *cluster* para o método de reconstrução tomográfica 2D e 3D (volumétrica). Neste sentido foram considerados os seguintes aspectos:

1. **seleção de *phantom* para qualificação da infraestrutura:** foram preparados quatro conjuntos de projeções tomográficas para o *phantom*, em que cada conjunto continha 1960 sinogramas, ou matrizes de projeções. Cada conjunto referia-se a uma das seguintes resoluções: 1000 pontos (1k), 2000 pontos (2k), 3000 pontos (3k) e 4000 pontos (4k);
2. **resolução das matrizes de projeções:** a resolução consiste no número de pontos de cada projeção. Foram observadas matrizes cujas projeções continham 1000 pontos (1k), 2000 pontos (2k), 3000 pontos (3k) e 4000 pontos (4k). Importante mencionar que o número de projeções (976) permaneceu igual em todas resoluções, logo foram analisadas matrizes com dimensões de  $976 \times 1000$ ,  $976 \times 2000$ ,  $976 \times 3000$  e  $976 \times 4000$ .
3. **número de Nós do cluster:** foram organizados quatro tipos de clusters, no qual cada tipo continha um determinado número de Nós. O intuito foi avaliar o comportamento do método no *cluster* em função do número de máquinas. Vale ressaltar que do total de Nós,

em todos os tipos, um Nó foi configurado como mestre e os demais como trabalhadores. Nesta análise foram considerados quatro tipos de clusters: 4, 6, 8 e 10 Nós.

4. **modelo dos Nós do cluster:** consistiu em selecionar três configurações de máquinas para compor o *cluster* conforme apresentado anteriormente pelo Quadro 4 (página 86). A Tabela 2, por sua vez, apresenta as capacidades dos *clusters* em função do número de vCPU e memória RAM. No intuito de facilitar a compreensão da tabela foram incluídas as configurações individuais de cada modelo de Nó.

Tabela 2 – Capacidade dos *clusters* em função do número de Nós e da configuração individual de cada modelo de Nó.

Modelo	Configuração do Nó		Capacidade do cluster (vCPU/Memória)			
	vCPU	Memória (GB)	4 Nós	6 Nós	8 Nós	10 Nós
m5.xlarge	4	16	16/64	24/96	32/128	40/160
m5.2xlarge	8	32	32/128	48/192	64/256	80/320
m5.4xlarge	16	64	64/256	96/384	128/512	160/640

É possível observar na Tabela 2 que foram organizadas 12 configurações diferentes de *clusters*, sendo a de menor capacidade contendo 16 vCPU e 64 GB de RAM e a configuração de maior capacidade com 180 vCPU e 640 GB de memória RAM. A Tabela 3 estabelece uma identificação para cada uma das 12 configurações de modo a facilitar a indicação dela ao longo do texto.

Tabela 3 – Identificação das configurações de *clusters*.

Modelo	Configuração do Nó		Capacidade do cluster (vCPU/Memória)			
	vCPU	Memória (GB)	4 Nós	6 Nós	8 Nós	10 Nós
m5.xlarge	4	16	t2-3	t2-5	t2-7	t2-9
m5.2xlarge	8	32	t3-3	t3-5	t3-7	t3-9
m5.4xlarge	16	64	t4-3	t4-5	t4-7	t4-9

A Tabela 4 apresenta o volume de dados que cada configuração processou a fim de obter as medidas. Considerando que cada configuração processou 35,63 GB de dados, conclui-se que foram processados 427,56 GB de modo paralelo e distribuído.

Tabela 4 – Volume de dados processados para análise da infraestrutura Big Data

Conjunto	Resolução	Volume (GB)
1	1000 (1k)	3,56
2	2000 (2k)	7,13
3	3000 (3k)	10,69
4	4000 (4k)	14,25
<b>Volume total</b>		35,63

A Figura 35 apresenta o gráfico com os tempos de reconstrução 2D para os quatro conjuntos de matrizes de projeções do *phantom*. Observa-se que, a medida que os recursos disponíveis nas configurações dos *clusters* aumentaram, o tempo de reconstrução 2D diminuiu.



A menor configuração avaliada ( $t_{2-3}$ ) levou 27 horas e 37 minutos para reconstruir o conjunto de projeções 4k. No entanto, foram observadas as seguintes variações: A configuração  $t_{2-5}$  levou mais tempo para realizar a reconstrução do conjunto de projeções 1k do que a configuração  $t_{3-3}$ , porém a partir do conjunto de projeções 2k a configuração  $t_{2-5}$  apresentou menores tempos de reconstrução comparados a configuração  $t_{3-3}$ . Situação semelhante ocorreu entre as configurações  $t_{2-9}$  e  $t_{3-5}$ . A configuração  $t_{2-9}$  levou a tempos de reconstrução maiores que  $t_{3-5}$  para os conjuntos de projeções 1k e 2k, mas melhores para os conjuntos 3k e 4k. O mesmo ocorreu com as configurações  $t_{4-3}$  e  $t_{3-7}$ .

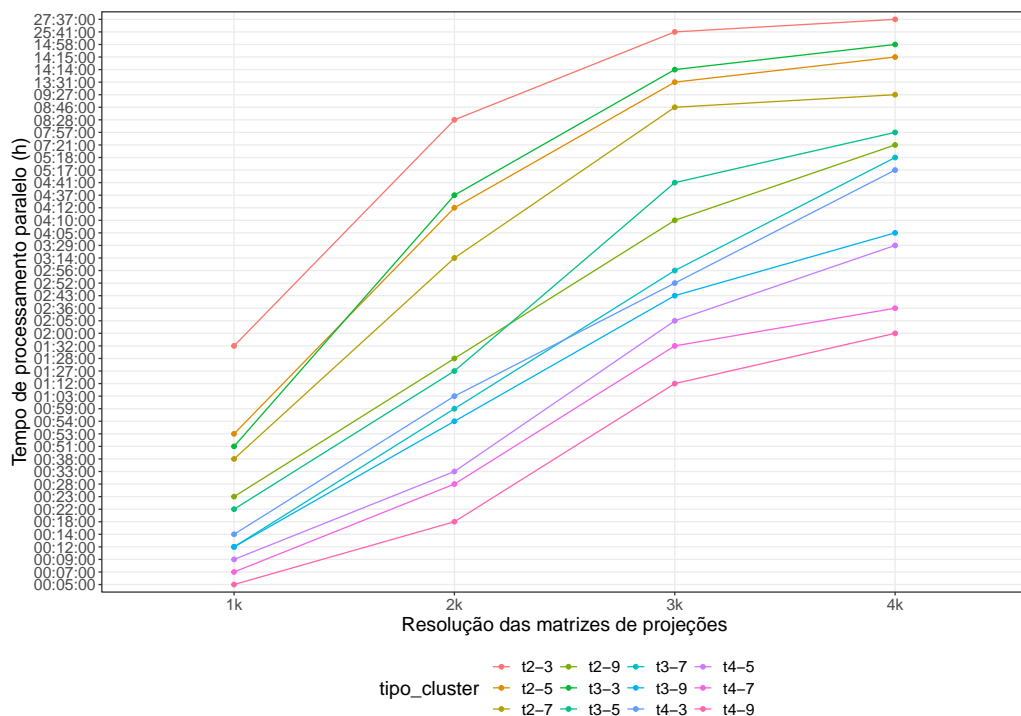


Figura 35 – Tempos de reconstrução 2D. Na legenda foi atribuída a sigla  $t_2$  para Nós  $m5.xlarge$ ,  $t_3$  para Nós  $m5.2xlarge$  e  $t_4$  para Nós  $m5.4xlarge$ .

A Figura 36 apresenta o gráfico com os tempos de reconstrução 3D (volumétrica) para os quatro conjuntos de matrizes de projeções do *phantom*.

A partir dos resultados apresentados na Figura 35 foi possível observar que a reconstrução 2D levou mais tempo que a reconstrução 3D (volumétrica), cujos resultados foram apresentados na Figura 36. A reconstrução 3D (volumétrica) levou menos tempo, pois os blocos reconstruídos foram armazenados em disco e não carregados em memória para serem visualizados. Verifica-se que não há informação para a reconstrução 3D (volumétrica) para matrizes  $976 \times 4000$  (4k) para a configuração de *cluster*  $t_{2-3}$  (3 Nós trabalhadores, 1 mestre, modelo  $m5.xlarge$ ). Isto porque tal configuração não suportou o processo de reconstrução 3D (volumétrica) para matrizes de projeções 4k ocorrendo em erro de "falta de memória".

Particularmente para o caso da reconstrução 2D, observou-se que os conjuntos de projeções 3k e 4k levaram mais de uma hora para reconstruir as matrizes, mesmo considerando a

configuração com maior quantidade de recursos ( $t_{4-9}$ ). Portanto, diante dos tempos observados nas Figuras 35 e 36 decidiu-se utilizar matrizes de projeções com dimensões  $976 \times 2000$  (2k) para se realizar os cálculos de Speedup e Eficiência bem como para realizar a análise da seleção de projeções.

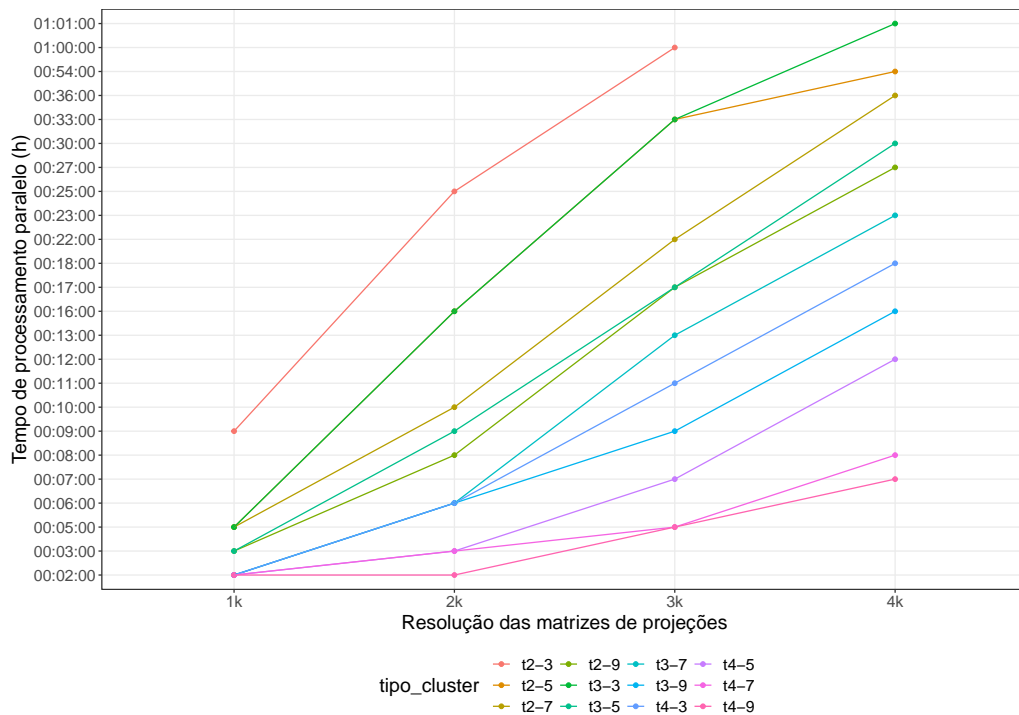


Figura 36 – Tempos de reconstrução 3D (volumétrica). Na legenda foi atribuída a sigla  $t_2$  para Nós  $m5.xlarge$ ,  $t_3$  para Nós  $m5.2xlarge$  e  $t_4$  para Nós  $m5.4xlarge$ .

Vale destacar que, neste trabalho, a estratégia adotada para paralelização da reconstrução bidimensional considerou a distribuição das matrizes de projeções pelos Nós do *cluster* de modo que a granularidade foi considerada média, pois tratou-se de dividir o volume total das matrizes de projeções. Na seção 5.3 é apresentada outra estratégia de paralelização da etapa de reconstrução bidimensional e a análise de viabilidade que buscou observar o custo da comunicação e o uso de memória. A estratégia considerou granularidade mais fina, pois ao invés de distribuir as matrizes pelo *cluster* distribuiu as projeções individuais das matrizes.

Nos tópicos a seguir, os resultados obtidos para as medidas de Speedup e Eficiência são apresentados a fim de definir a configuração de *cluster* mais apropriada para execução do método.

### 5.1.1 Speedup

O cálculo do Speedup, como indicado na Equação 4.5 (pág. 82), consiste na razão entre o tempo para operação sequencial e o tempo para operação em paralelo.

Portanto, o conjunto de matrizes de projeções tomográficas com dimensão  $976 \times 2000$  (2k), indicado na Tabela 4, foi submetido a uma única máquina de cada modelo indicado na

Tabela 2. Os tempos sequenciais obtidos para a reconstrução 2D e 3D (volumétrica) estão indicados na Tabela 5.

Tabela 5 – Tempos sequenciais obtidos para reconstrução 2D e 3D (volumétrica).

modelo	Reconstrução 2D (h)	Reconstrução 3D (volumétrica) (h)	Total (h)
m5.xlarge	31h 41min	8h 52min	40h 33min
m5.2xlarge	31h 23min	8h 32min	39h 55min
m5.4xlarge	31h 05min	8h 29min	39h 34min

As Figuras 37 e 38 apresentam a medida de Speedup para a reconstrução 2D e reconstrução 3D, respectivamente, calculadas a partir dos resultados observados na Tabela 5.

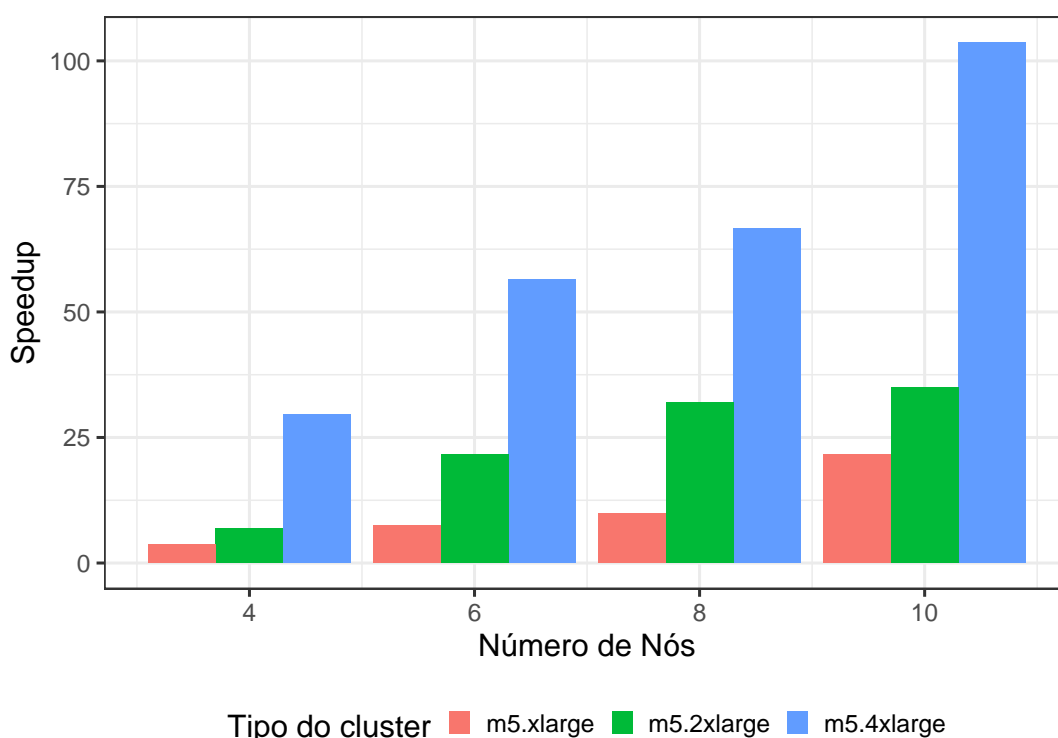


Figura 37 – Speedup para reconstrução 2D.

É possível observar que os *clusters* que utilizaram máquinas `m5.4xlarge` obtiveram maiores valores de Speedup em função do maior número de processadores (vCPU) e de memória RAM. Observa-se que o Speedup para os *clusters* que utilizaram máquinas `m5.xlarge` e `m5.2xlarge` obtiveram melhores resultados no caso da reconstrução 3D (volumétrica). O motivo está associado ao fato que após reconstruir o volume, o método salva os dados em disco direto do Nó trabalhador ao invés de enviá-los para o Nó mestre o que reduz tempo de comunicação.

Sob a perspectiva de análise da infraestrutura de Big Data, destaca-se que todos os sinogramas (1960) foram reconstruídos considerando todas as projeções (matrizes completas). Este procedimento mostrou-se adequado, pois foi possível considerar a avaliação do ambiente

na condição de maior demanda, ou seja, quando todos os sinogramas de uma amostra são processados.

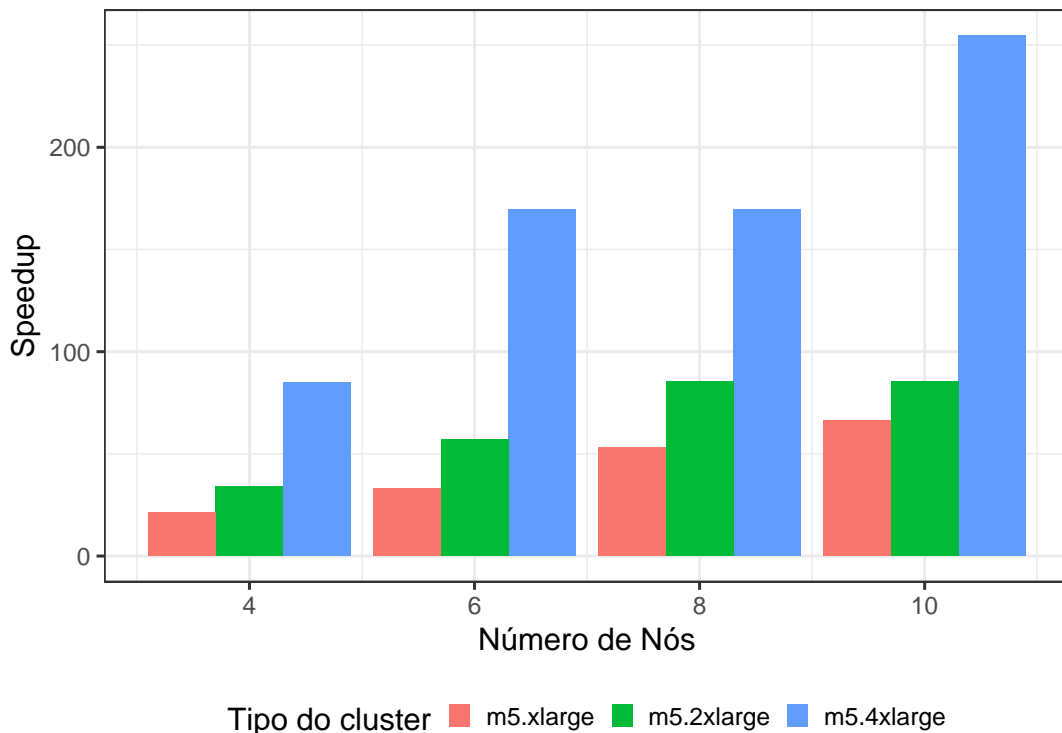


Figura 38 – Speedup para reconstrução 3D (volumétrica).

A seguir é discutida a medida de Eficiência para análise da execução do método em diferentes configurações de *clusters*.

### 5.1.2 Eficiência

A medida de Eficiência consiste na razão do Speedup pelo número de processadores e avalia quanto o paralelismo foi explorado no algoritmo bem como quantifica a utilização do processador. A Figura 39 apresenta o cálculo da Eficiência para a reconstrução 2D.

No gráfico, apresentado na Figura 39, é possível observar que os *clusters* com máquinas *m5.4xlarge* apresentaram melhores eficiência como era de se esperar, no entanto, cabe observar que o *cluster* com seis Nós obteve eficiência idêntica ao *cluster* com 10 Nós. A Figura 40 apresenta a medida de eficiência para a reconstrução 3D (volumétrica).

No gráfico é possível observar que o *cluster* com seis máquinas *m5.4xlarge* obteve a maior eficiência em comparação as demais configurações, inclusive sobre *clusters* com máquinas de mesmo modelo mas com maior número de Nós. Tal resultado pode implicar também no custo de processamento.

Considerando as configurações de *clusters* que obtiveram melhores eficiências e para efeito da discussão sobre a relação de custo denominou-se de A a configuração *m5.4xlarge*

com 6 Nós e denominou-se de B a configuração m5 . 4xlarge com 10 Nós.

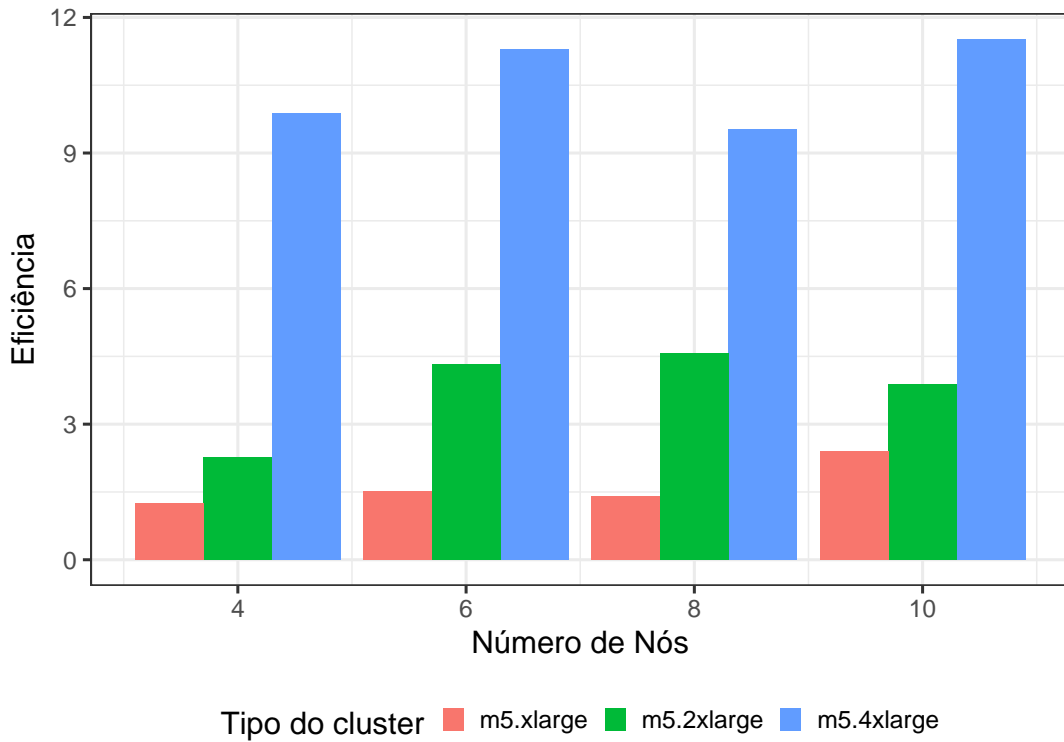


Figura 39 – Eficiência para reconstrução 2D.

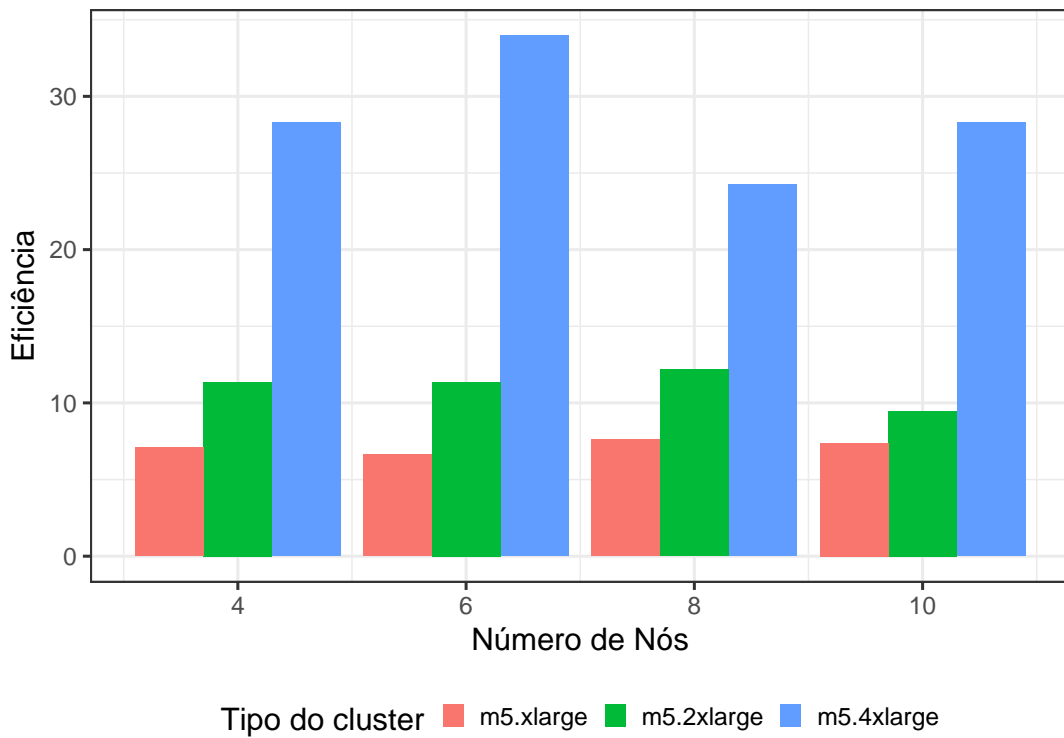


Figura 40 – Eficiência para reconstrução 3D (volumétrica).

Considerando o custo de processamento de cada máquina do *cluster* foi possível observar

a relação custo de processamento sequencial e custo de processamento paralelo. A Tabela 6 apresenta o custo, em dólares americanos/hora de processamento, de uma máquina de acordo com o modelo de configuração.

Tabela 6 – Custo de processamento por hora, em dólares (USD).

modelo	Custo/hora (USD)
m5.xlarge	0,192
m5.2xlarge	0,384
m5.4xlarge	0,768

Na Tabela 7 foi possível observar que a relação de custo no caso da reconstrução 2D para a configuração A foi de 11,31, ou seja, o processamento paralelo foi superior na ordem de 11 vezes e 1,82% abaixo da configuração B. Por outro lado, a reconstrução 3D (volumétrica) para a configuração A foi 20% superior à configuração B. Por este motivo, decidiu-se utilizar o cluster de configuração A (m5.4xlarge com 6 Nós, sendo 1 Nó mestre e 5 Nós trabalhadores) para processar as amostras e avaliar o processo de seleção das projeções.

Tabela 7 – Relação custo de processamento sequencial por custo de processamento paralelo nas reconstruções 2D e 3D (volumétrica).

Modelo	4 Nós		6 Nós		8 Nós		10 Nós	
	Rec. 2D	Rec. 3D	Rec. 2D	Rec. 3D	Rec. 2D	Rec. 3D	Rec. 2D	Rec. 3D
m5.xlarge	1,25	7,09	1,51	6,65	1,40	7,60	2,40	7,39
m5.2xlarge	2,27	11,37	4,33	11,37	4,56	12,19	3,87	9,48
m5.4xlarge	9,87	28,29	<b>11,31</b>	<b>33,95</b>	9,52	24,25	<b>11,52</b>	<b>28,29</b>

Vale observar que o tempo de processamento do método de reconstrução tomográfica no ambiente Big Data foi em torno de 35 minutos para uma única amostra de semente agrícola composta por 1960 sinogramas o que equivale a 1.912.960 projeções, ou ainda 7,13 GB de dados tomográficos. O tempo de processamento considerou o carregamento das projeções no ambiente, a seleção das projeções nos sinogramas e as reconstruções 2D e 3D (volumétrica).

Na próxima seção é apresentada a análise das reconstruções tomográficas a partir das matrizes de projeções selecionadas tomando por base a densidade espectral.

## 5.2 Análise do método de reconstrução tomográfica baseado na densidade espectral das projeções

Na seção 4.5 se discutiu o processo de seleção de projeções baseada em densidade espectral e foi apresentado o fluxograma do processo de seleção das projeções tomográficas. Com base naquele fluxograma, o Algoritmo 3 a seguir apresenta as tarefas necessárias para a realização de tal processo.

Observa-se que, no algoritmo, é dado uma matriz de projeções  $S$  com dimensão  $N \times M$ , em que  $N$  indica o número de projeções e  $M$  o número de pontos por projeção. Na sequência,

realiza-se o cálculo da energia, define-se as classes a partir do valor de energia das projeções, seleciona-se as projeções e um novo sinograma  $S'$ , com dimensão  $R \times M$ , é gerado. Neste caso, o número de projeções  $R$  é um subconjunto de  $N$ .

---

**Algoritmo 3:** Seleção de projeções baseada em densidade espectral de potência.

---

**Entrada:** Matriz de projeções ( $S = N \times M$ )

**Saída:** Novo sinograma ( $S' = R \times M$ )

```

1  $lista_e \leftarrow \text{calcula\_energia}(S)$ ;
2  $projecoes\_selecionadas \leftarrow \emptyset$ ;
3  $k \leftarrow \lfloor \sqrt{N} \rfloor$ ;
4  $amplitude \leftarrow \lfloor (\max(lista_e) - \min(lista_e)) \div k \rfloor$ ;
5 para cada classe  $c \in 1, 2, \dots, k$  faça
6   |  $\mu \leftarrow \text{calcula média da classe } c$ ;
7   |  $\sigma \leftarrow \text{calcula desvio padrão da classe } c$ ;
8   |  $projecoes\_selecionadas \leftarrow \text{projecoes da classe } c \text{ com } \sigma \leq 1$ ;
9 fim
10  $S' \leftarrow \text{gera novo sinograma a partir das } projecoes\_selecionadas$ ;
```

---

A respeito do cálculo da energia, a linha 1 do Algoritmo 3, invoca a função *calcula\_energia* que é apresentada em detalhes no Algoritmo 4. A referida função recebe uma matriz de projeções  $S$  e, para cada projeção, calcula a Transformada de Fourier, calcula o espectro de potência, bem como a energia. Ao final da função, uma lista é gerada contendo o índice e a energia associada a cada projeção.

---

**Algoritmo 4:** Calcula a energia das projeções contidas em uma matriz (sinograma).

---

```

1 Função  $calcula\_energia(S)$ 
   | Entrada: Matriz de projeções ( $S = N \times M$ )
   | Saída: Lista de energias das projeções ( $lista$ )
2   para cada  $projecao p_i \in 1, 2, \dots, N$  faça
3     |  $tf \leftarrow \text{calcula a Transformada de Fourier de } p_i$ ;
4     |  $dep \leftarrow \text{calcula o Espectro de potência}$ ;
5     |  $e_i \leftarrow \text{calcula a Energia}$ ;
6     |  $lista \leftarrow (\text{índice de } p_i, \text{ energia } e_i)$ ;
7   fim
8   retorna  $lista$ 
9 fim
```

---

A Figura 41 ilustra os processos da seleção de projeções tomográficas em uma matriz de projeções de um *phantom*. A linha tracejada ilustra as etapas do processo de seleção de projeções tomográficas de acordo com os algoritmos apresentados. O processo foi aplicado em um sinograma do *phantom*. Vale destacar o gráfico em que são apresentadas as classes de energia das quais foram selecionadas as projeções e que geraram a nova matriz de projeções ( $S'$ ). Na

sequência, ainda na Figura 41, é apresentada a reconstrução bidimensional que foi realizada após a seleção das projeções.

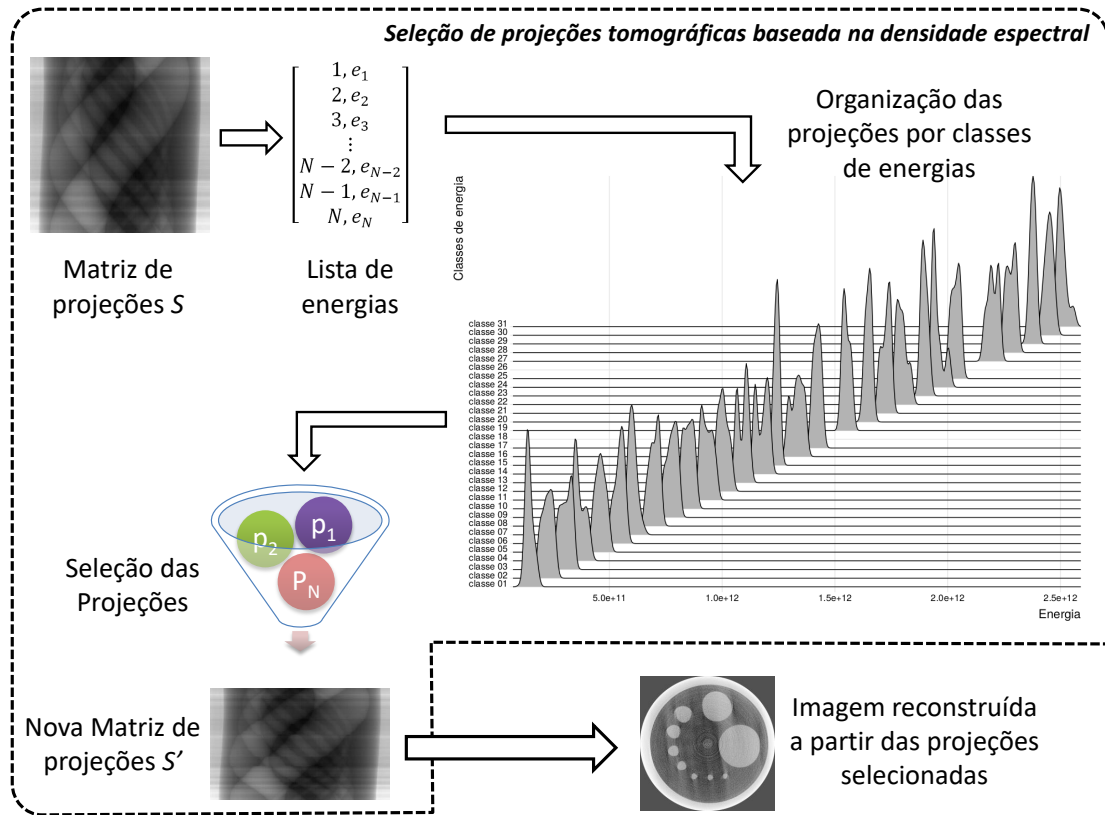


Figura 41 – Seleção de projeções tomográficas aplicada em um sinograma do *phantom*.

Para a análise da seleção das projeções tomográficas, visando a reconstrução bidimensional, foram processadas 33 amostras de sementes que totalizou 235, 29 GB de dados. A análise consistiu em calcular e observar as métricas SSIM, NRMSE e PSNR. A imagem de referência (*ground truth*), para cada fatia analisada, foi preparada a partir da reconstrução tomográfica bidimensional considerando todas as projeções, ou seja, 976 projeções. Além disso, foi definida uma região de interesse (ROI) com dimensões  $1200 \times 1200$  pixels, posicionada no centro das imagens reconstruídas, que forneceu os dados para a análise das medidas.

Cada amostra agrícola (semente) continha 1960 sinogramas, que representava a população, e para o cálculo das métricas foi selecionado um subconjunto do total de sinogramas, ou seja, a amostra representativa da população. A Equação 5.1 foi aplicada para determinar o número de sinogramas do subconjunto, onde foi considerado o intervalo de confiança de 99%, margem de erro de 5% e proporção  $p = 0,50$ , uma vez que não foi considerada nenhuma informação *a priori* dos sinogramas.

$$n = \frac{N.p.(1 - p).z^2}{p.(1 - p).z^2 + N.e^2} \tag{5.1}$$



onde  $N = 1960$ ,  $p = 0,50$ ,  $e = 0,05$ ,  $z = 2,58$ . Logo, o valor obtido foi de  $n \approx 498$ , ou seja, para a análise da seleção de projeções tomográficas foram utilizadas 498 matrizes de projeções em cada amostra de semente.

A Figura 42 apresenta o gráfico do número de projeções tomográficas selecionadas em cada uma das fatias escolhidas nas 33 amostras analisadas.

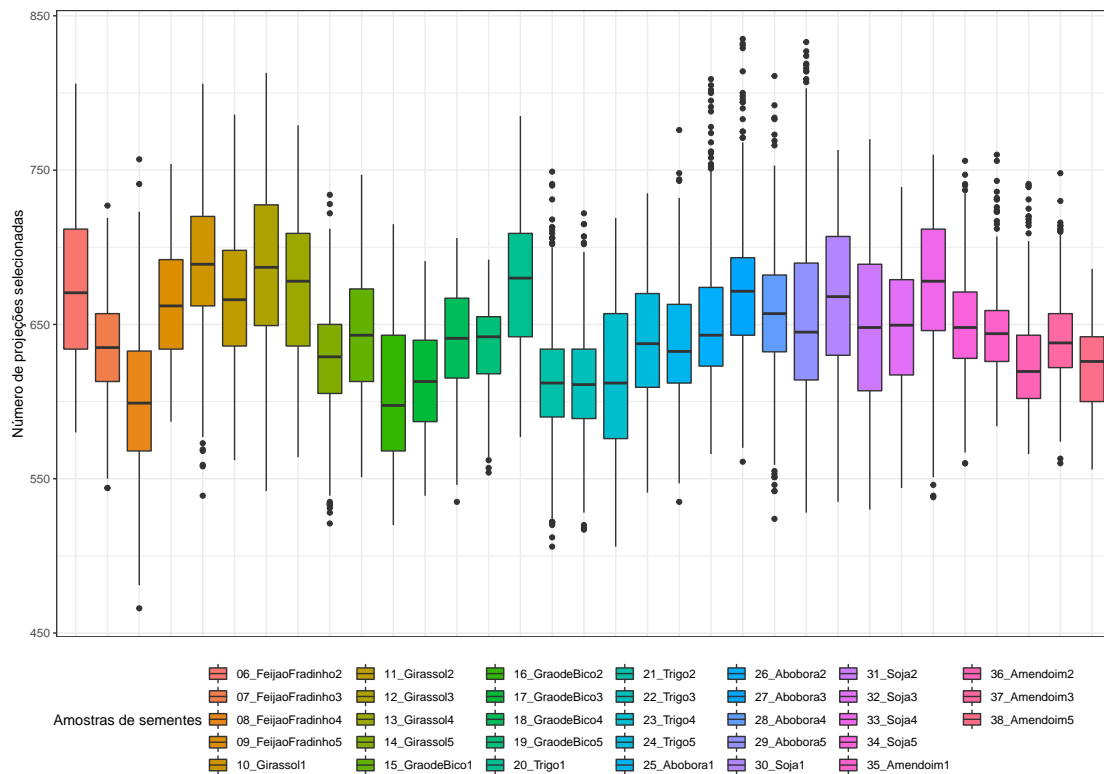


Figura 42 – Análise do número de projeções tomográficas selecionadas por amostra.

A estratégia de escolha dos sinogramas considerou a posição no eixo- $z$  de modo a obter um arranjo uniformemente espaçado, por entender ser possível avaliar melhor o processo de seleção das projeções tomográficas em diferentes localizações da amostra. Portanto, a cada quatro sinogramas um foi escolhido, sendo que próximo ao sinograma central reduziu-se o intervalo de escolha a fim de completar o total de 498 sinogramas. O termo *sinograma central*, neste contexto, refere-se ao sinograma cujo feixe de raio-X incide à  $90^\circ$  graus.

Observa-se que, em geral, o algoritmo selecionou entre 600 e 700 projeções por sinograma com mediana em torno de 650. Considerando que todas matrizes continham 976 projeções, tem-se que o algoritmo selecionou em torno de 61,47% a 71,72% das projeções, com frequência maior em torno de 66,60%. Vale observar, no entanto, que ocorreram casos em que a taxa de seleção foi mais alta, a exemplo da amostra "27\_Abobora3", onde se obteve em um dos sinogramas taxa de 85,55%, enquanto em outras foram obtidas taxas menores de seleção a exemplo da amostra "08\_FeijaoFradinho4" na qual um dos sinogramas levou à obtenção da taxa de 47,75%. Nos tópicos a seguir as medidas SSIM, NRMSE e PSNR são observadas após reconstruir as matrizes com as projeções selecionadas pelo método desenvolvido.

### 5.2.1 Análise SSIM

A medida SSIM foi observada e permitiu avaliar a informação estrutural das imagens reconstruídas a partir de um subconjunto de projeções. Assim foi possível verificar a qualidade da reconstrução bidimensional ao reduzir a quantidade de dados. Quanto mais próximo de 1 for o valor da medida SSIM entende-se que mais idêntica é a imagem reconstruída com menor número de projeções em relação à imagem reconstruída com todas as projeções disponíveis.

Inicialmente foi realizada a análise experimental da medida SSIM para o *phantom*, indicado na Figura 22 (pág. 78), considerando 498 fatias reconstruídas a partir dos sinogramas escolhidos conforme estratégia supracitada. Além disso, com relação a seleção das projeções, foi mencionado na seção 4.5.2, que foram consideradas significativas as projeções que continham energia dentro do intervalo de um desvio-padrão ( $\sigma$ ). Portanto, antes de analisar todas as amostras agrícolas, observou-se a variação do intervalo no qual as projeções do sinograma foram consideradas significativas e, portanto, selecionadas para a reconstrução. Além disso, durante o processo de reconstrução, preencheu-se com zeros a matriz com as projeções selecionadas devido o fato dela não ser quadrada o que caracteriza mal condicionamento do sistema. A Tabela 8 apresenta as três variações de intervalo observadas neste trabalho considerando os valores mínimo, mediana e máximo para a medida SSIM. durante o processo de reconstrução,

Tabela 8 – Avaliação da variação do intervalo para seleção das projeções em um sinograma.

	$\sigma = 1,00$		$\sigma = 1,50$		$\sigma = 2,00$	
<b>Valor</b>	<b>SSIM</b>	<b>Seleção</b>	<b>SSIM</b>	<b>Seleção</b>	<b>SSIM</b>	<b>Seleção</b>
Mínimo	0,774	61,78%	0,932	86,58%	0,976	94,88%
Mediana	0,813	62,50%	0,948	88,42%	0,984	96,11%
Máximo	0,931	63,83%	0,974	89,04%	0,994	97,03%

A Tabela 9 apresenta a comparação dos intervalos  $\sigma = 1,50$  e  $\sigma = 2,00$  tomando por base os valores de SSIM e Seleção do intervalo  $\sigma = 1,00$ .

Tabela 9 – Comparação de intervalos para seleção das projeções em um sinograma em relação ao intervalo  $\sigma = 1,00$ .

	$\sigma = 1,50$		$\sigma = 2,00$	
<b>Valor</b>	<b>SSIM</b>	<b>Seleção</b>	<b>SSIM</b>	<b>Seleção</b>
Mínimo	20,14%	40,14%	26,10%	53,58%
Mediana	16,60%	41,47%	21,03%	53,78%
Máximo	4,62%	39,49%	6,77%	52,01%

Observa-se na Tabela 9, para o valor mínimo quando  $\sigma = 1,50$ , que a taxa de seleção de projeções tomográficas aumentou em 40,14% ao passo que o aumento do valor do SSIM foi de 20,41%. Quando  $\sigma = 2,00$ , a taxa de seleção aumentou em 53,58% para um aumento de 26,10% no SSIM. Situação semelhante ocorreu para o valor de mediana. Já para o valor de máximo quando  $\sigma = 1,50$ , por exemplo, houve aumento de 39,49% na taxa de seleção para aumento de 4,62% no SSIM. Portanto, decidiu-se manter o intervalo de seleção em um desvio

padrão ( $\sigma = 1,00$ ), vez que o valor do SSIM não aumenta na mesma proporção que a taxa de seleção. A Tabela 10 apresenta os valores mínimo, mediana e máximo para a medida SSIM indicando ainda a fatia bem como taxa de seleção das projeções, considerando  $\sigma = 1,00$ .

Tabela 10 – Valores de SSIM mínimo, mediana e máximo calculados para o *phantom*.

Valor	Fatia	SSIM	Seleção
Mínimo	728	0,774	61,78%
Mediana	1276	0,813	62,50%
Máximo	988	0,931	63,83%

Observa-se que a taxa de seleção foi de 62,50% para o SSIM de 0,813, ou seja, com redução de 37,50% do conjunto inicial de dados ainda foi possível obter um valor SSIM em torno de 0,800. A Figura 43 apresenta as imagens reconstruídas das fatias mencionadas na Tabela 10. Visualmente é possível observar que as principais informações das fatias foram preservadas mesmo com a redução do número de projeções.

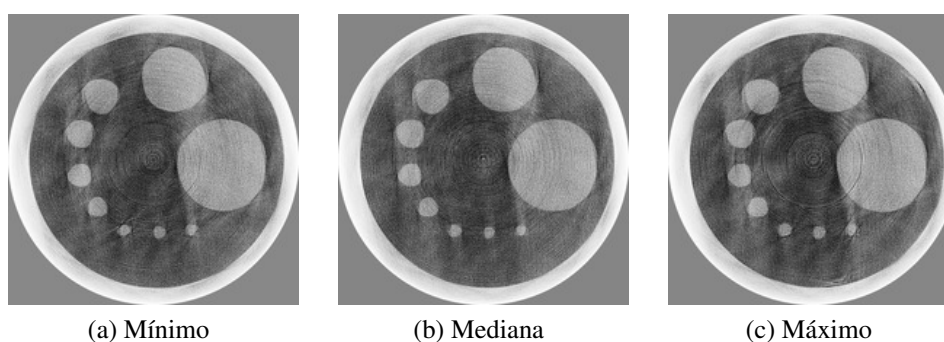


Figura 43 – Fatias do *phantom* que representam os valores mínimo, mediana e máximo da medida SSIM. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

Na sequência, o primeiro conjunto de dados analisado refere-se a semente de Feijão Fradinho cujos valores obtidos de SSIM estão reportados na Tabela 11.

Tabela 11 – Resultados para a avaliação de imagens tomográficas de amostras de Feijão Fradinho. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
06_FeijaoFradinho2	1696	0,687	61,27%	1152	0,854	69,47%	700	0,947	82,27%
07_FeijaoFradinho3	1924	0,753	57,89%	1256	0,828	66,70%	484	0,922	71,31%
08_FeijaoFradinho4	444	0,702	49,59%	1080	0,809	61,68%	832	0,893	68,24%
09_FeijaoFradinho5	360	0,789	62,19%	876	0,846	69,26%	1580	0,928	72,23%

Entre as amostras de Feijão Fradinho, observa-se que a maior taxa de seleção (82,27%) resultou no maior valor SSIM (0,947), enquanto que a menor taxa de seleção (49,59%) não produziu o menor valor SSIM. Neste caso, destaca-se que a fatia que obteve o menor valor SSIM (0,687) apresentou taxa de seleção das projeções de 61,27% que é idêntica às taxas de seleção das fatias cujos valores SSIM correspondem a mediana de cada conjunto.

A Tabela 12 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções, do segundo conjunto de amostras referente à semente de Girassol. No conjunto das amostras de semente de Girassol, observou-se que a imagem reconstruída da fatia 1368 obteve valor SSIM de 0,910 com taxa de seleção das projeções de 65,98%. Interessante observar que a fatia 1020 obteve o maior valor SSIM do conjunto de amostras, porém selecionou 21% de projeções do que a fatia 1368 para obter um valor SSIM superior em 2,5%. Por outro lado, a imagem reconstruída que obteve o menor valor SSIM observado (0,734) selecionou mais projeções (73,98%). Vale observar ainda que a fatia 500 selecionou menos de 60% e obteve um valor de SSIM superior a 0,800.

Tabela 12 – Resultados para a avaliação de imagens tomográficas de amostras de Girassol. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
10_Girassol1	268	0,745	58,20%	620	0,854	71,62%	1236	0,928	77,05%
11_Girassol2	1656	0,773	71,72%	704	0,854	68,65%	1708	0,922	75,72%
12_Girassol3	1856	0,734	73,98%	500	0,840	58,71%	1020	0,933	83,20%
13_Girassol4	1284	0,746	59,12%	680	0,848	65,06%	1328	0,923	77,25%
14_Girassol5	520	0,735	53,38%	1156	0,822	66,60%	1368	0,910	65,98%

A Tabela 13 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções, do conjunto de amostras referente à semente de Grão de Bico.

Tabela 13 – Resultados para a avaliação de imagens tomográficas de amostras de Grão de Bico. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
15_GraodeBico1	1296	0,760	56,97%	968	0,832	65,27%	652	0,914	69,26%
16_GraodeBico2	1868	0,744	54,30%	700	0,809	61,58%	1324	0,902	71,31%
17_GraodeBico3	188	0,752	58,50%	140	0,817	59,12%	488	0,876	69,77%
18_GraodeBico4	1464	0,744	54,82%	112	0,831	64,45%	1056	0,922	70,90%
19_GraodeBico5	320	0,746	59,22%	176	0,828	64,86%	1836	0,897	69,47%

No grupo das amostras de Grão de Bico, observa-se que os valores SSIM foram superiores a 0,700 com taxas de seleção entre 54,30% e 71,31%. No caso da menor taxa de seleção (54,30%) foram selecionadas 530 projeções que produziu o valor SSIM de 0,744. Por outro lado, a maior taxa de seleção no grupo (71,31%) obteve SSIM de 0,902, ou seja, valor SSIM 21,23% superior em relação a menor taxa de seleção, no entanto, utilizando 31,32% a mais de projeções. Neste grupo, ao aumentar a taxa de seleção mesmo que superior a 30% obteve-se um ganho no SSIM superior a 20% que pode ser entendido um resultado expressivo.

A Tabela 14 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções, do conjunto de amostras referente à semente de Trigo. O conjunto de amostras de sementes de Trigo apresentou taxas de seleção maiores do que a das amostras de Grão de Bico, sendo a maior taxa de seleção em 74,90%. Por outro lado, observou-se valor SSIM de 0,912

com 63,11% das projeções selecionadas. Na coluna da mediana foi possível observar que para a fatia 264 se obteve valor SSIM de 0,819 com taxa de seleção menor que 60%.

Tabela 14 – Resultados para a avaliação de imagens tomográficas de amostras de Trigo. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
20_Trigo1	340	0,757	63,63%	636	0,859	71,31%	856	0,926	74,08%
21_Trigo2	1356	0,754	51,84%	148	0,821	62,81%	468	0,897	74,90%
22_Trigo3	324	0,751	57,07%	664	0,818	63,63%	1492	0,912	63,11%
23_Trigo4	1336	0,743	51,84%	264	0,819	57,17%	1784	0,910	68,14%
24_Trigo5	328	0,775	58,09%	552	0,832	65,98%	1784	0,906	71,31%

O próximo conjunto refere-se às amostras de semente de Abóbora. A Tabela 15 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções.

Tabela 15 – Resultados para a avaliação de imagens tomográficas de amostras de Abóbora. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
25_Abóbora1	1104	0,751	59,43%	388	0,821	63,01%	988	0,935	71,52%
26_Abóbora2	1768	0,758	59,73%	668	0,834	67,93%	1144	0,964	80,74%
27_Abóbora3	368	0,760	67,62%	444	0,847	67,83%	196	0,935	76,84%
28_Abóbora4	1356	0,758	53,69%	172	0,839	66,80%	992	0,926	71,21%
29_Abóbora5	1084	0,739	55,23%	72	0,831	67,21%	920	0,925	83,40%

Na Tabela 15, observa-se que todas as fatias obtiveram valor SSIM superior a 0,750. A Tabela 16 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções, do conjunto de amostras referente à semente de Soja.

Tabela 16 – Resultados para a avaliação de imagens tomográficas de amostras de Soja. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
30_Soja1	1664	0,751	57,48%	324	0,827	68,85%	848	0,910	76,43%
31_Soja2	100	0,718	55,33%	180	0,830	63,01%	1252	0,907	78,18%
32_Soja3	1804	0,701	66,29%	356	0,821	69,98%	1992	0,905	72,03%
33_Soja4	980	0,625	62,40%	268	0,855	69,67%	1108	0,922	72,85%
34_Soja5	140	0,768	60,96%	228	0,837	67,11%	948	0,912	69,88%

Na Tabela 16 é possível observar que a fatia 948 obteve valor SSIM superior ao da fatia 268 embora ambas fatias tenham taxa de seleção próximas. O último conjunto refere-se às amostras de semente de Amendoim. A Tabela 17 apresenta os valores obtidos de SSIM, bem como de taxa de seleção das projeções.

Na Tabela 17 observa-se que para a fatia 992 se obteve maior valor SSIM (0,959) com taxa de seleção em 74,08% enquanto que as projeções que obtiveram valor SSIM em torno de 0,80 a taxa de seleção ficou em torno de 63%. As Tabelas 18, 19, 20, 21, 22, 23 e 24 apresentam conjuntos de imagens reconstruídas.



Tabela 17 – Resultados para avaliação de imagens tomográficas de amostras de Amendoim. Valores de SSIM mínimo, mediana e máximo para as amostras analisadas.

Amostra	Mínimo			Mediana			Máximo		
	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção	Fatia	SSIM	Seleção
35_Amendoim1	1640	0,762	60,86%	116	0,829	64,55%	992	0,959	74,08%
36_Amendoim2	244	0,760	59,22%	124	0,817	61,48%	492	0,911	73,57%
37_Amendoim3	1560	0,781	58,81%	104	0,829	64,45%	988	0,916	67,83%
38_Amendoim5	1988	0,745	56,97%	316	0,826	65,37%	1436	0,905	66,39%

Tabela 18 – Imagens tomográficas de amostras de Feijão Fradinho. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 11, página 117. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

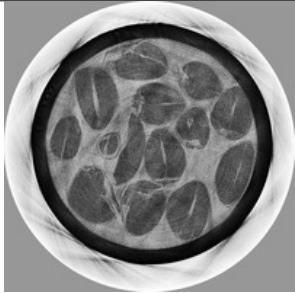
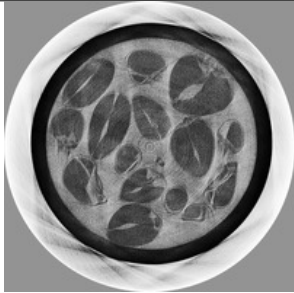
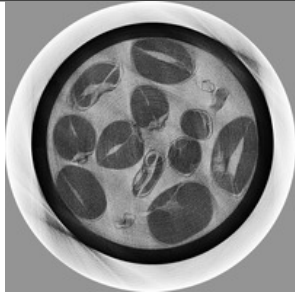
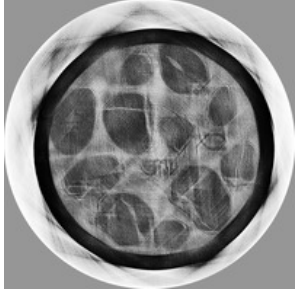
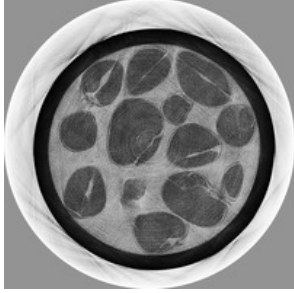
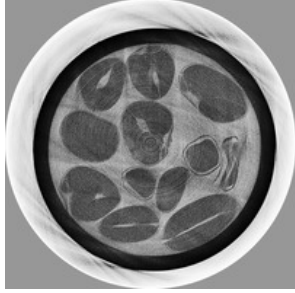
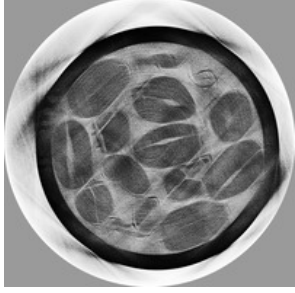
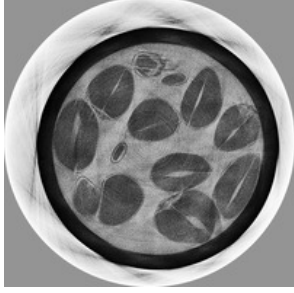
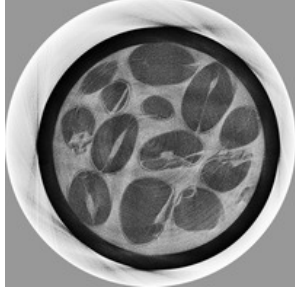
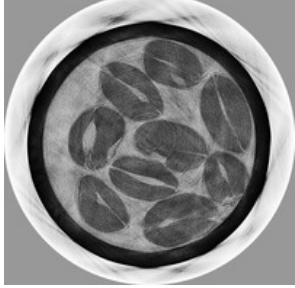
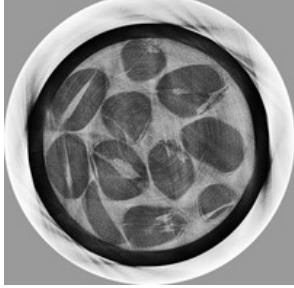
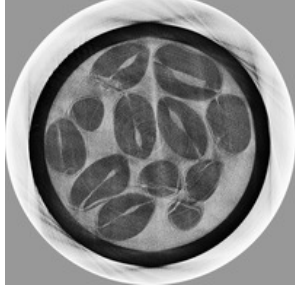
Amostra	Mínimo	Mediana	Máximo
06_FeijaoFradinho2			
07_FeijaoFradinho3			
08_FeijaoFradinho4			
09_FeijaoFradinho5			

Tabela 19 – Imagens tomográficas de amostras de sementes de Girassol. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 12, página 118. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

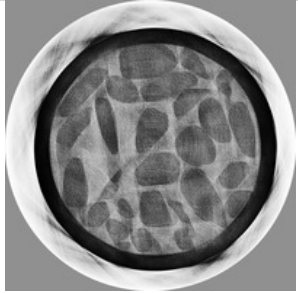
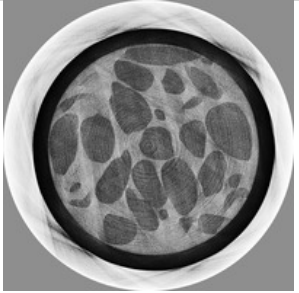
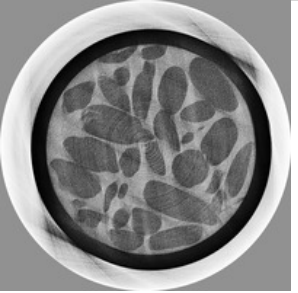
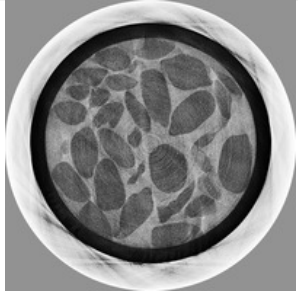
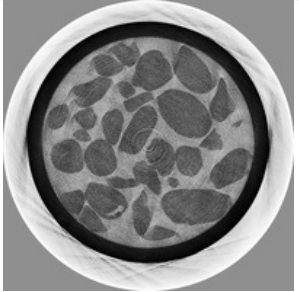
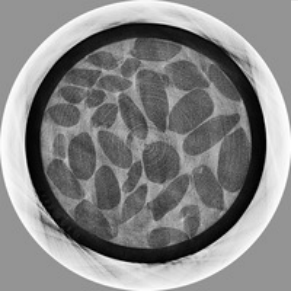
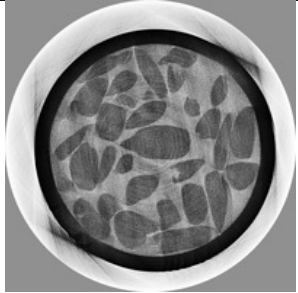
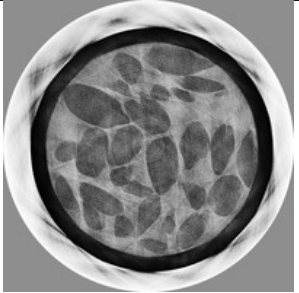
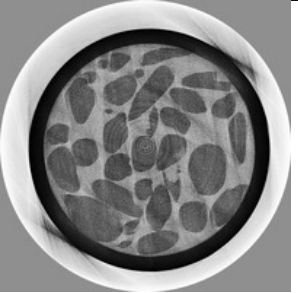
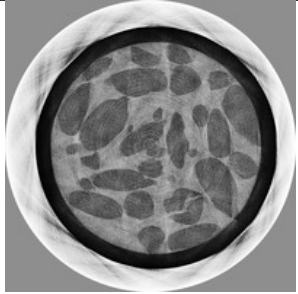
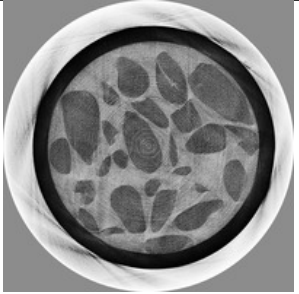
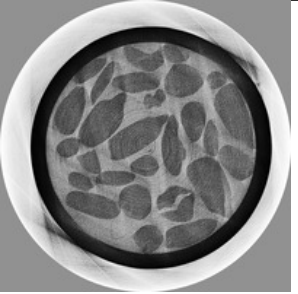
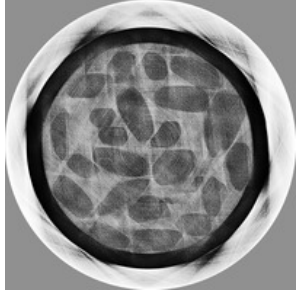
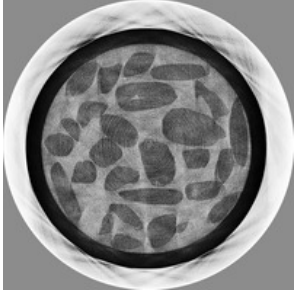
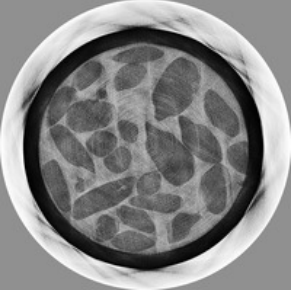
Amostra	Mínimo	Mediana	Máximo
10_Girassol1			
11_Girassol2			
12_Girassol3			
13_Girassol4			
14_Girassol5			

Tabela 20 – Imagens tomográficas de amostras de sementes de Grão de Bico. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 13, página 118. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

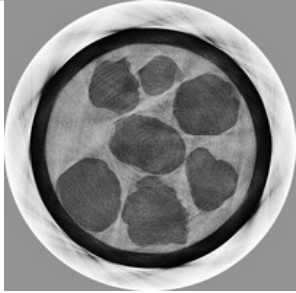
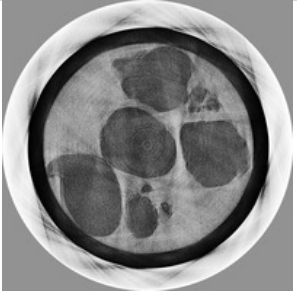
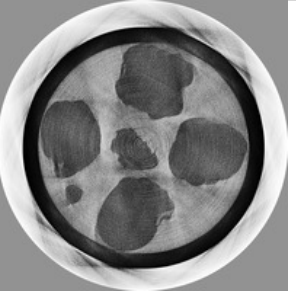
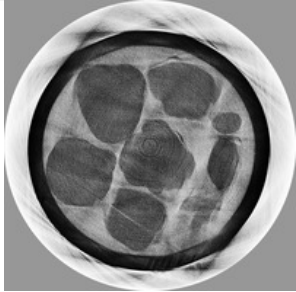
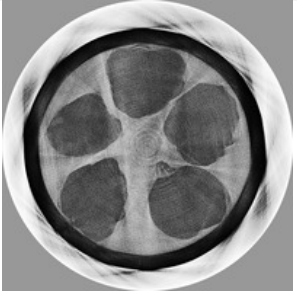
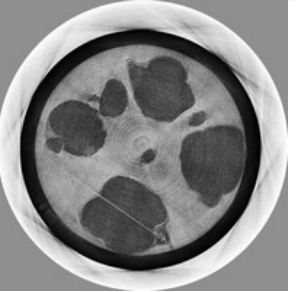
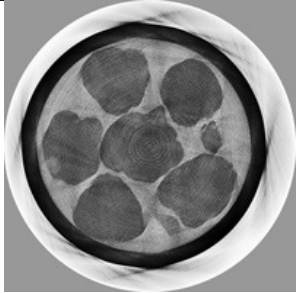
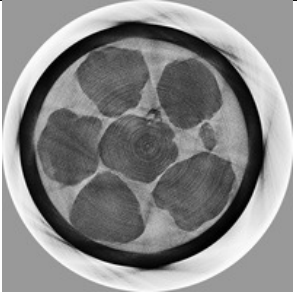
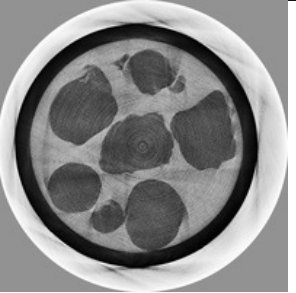
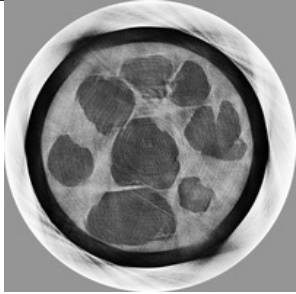
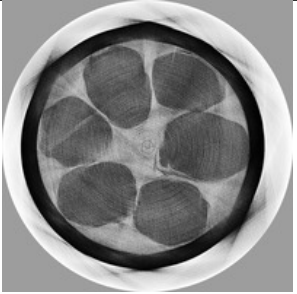
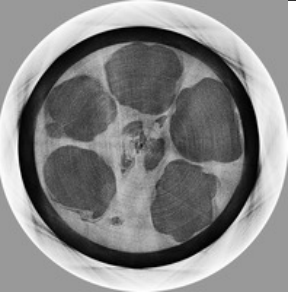
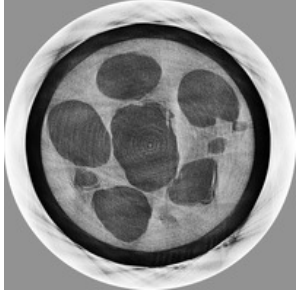
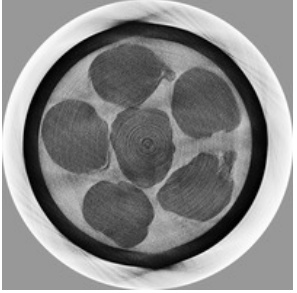
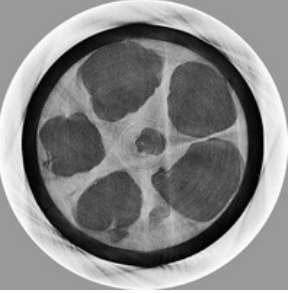
Amostra	Mínimo	Mediana	Máximo
15_Graodebico1			
16_Graodebico2			
17_Graodebico3			
18_Graodebico4			
19_Graodebico5			



Tabela 21 – Imagens tomográficas de amostras de sementes de Trigo. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 14, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.


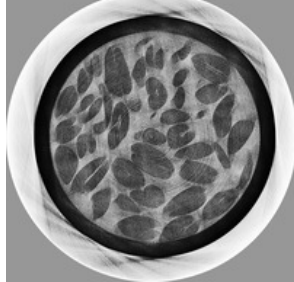
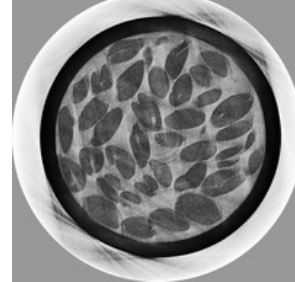
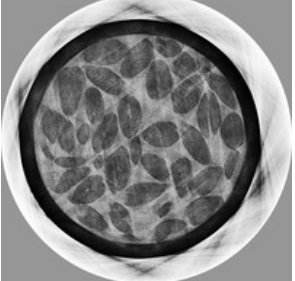
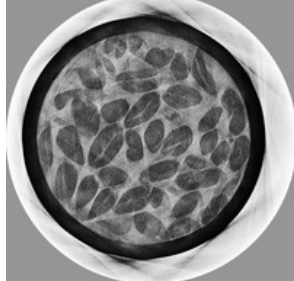
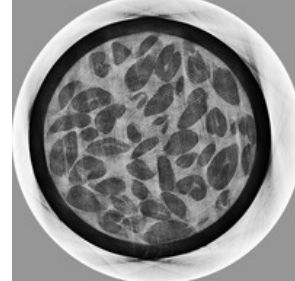
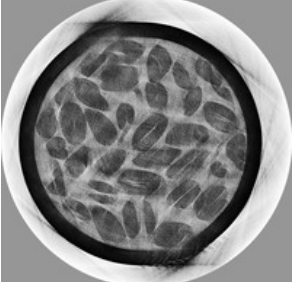
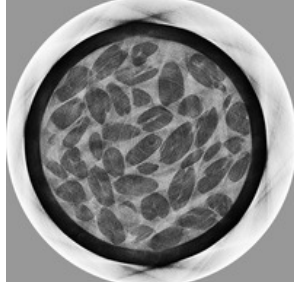
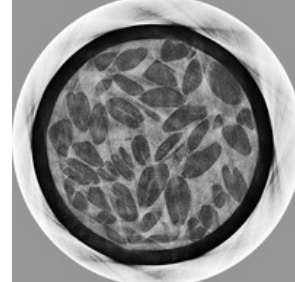
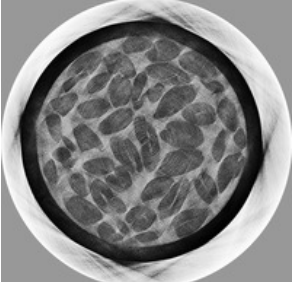
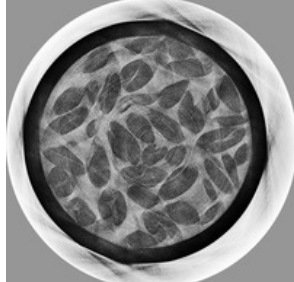
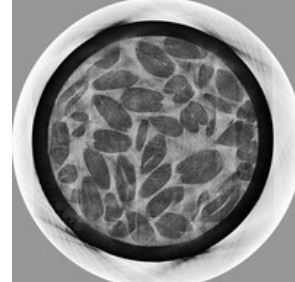
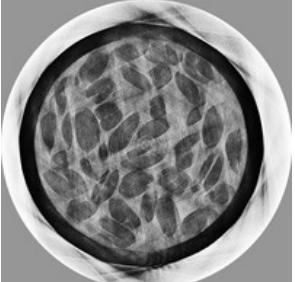
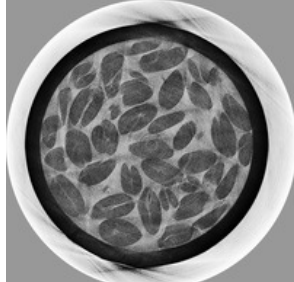
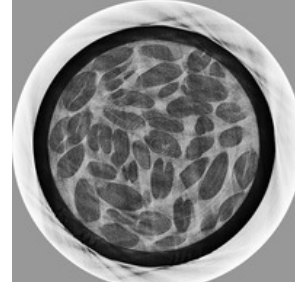
Amostra	Mínimo	Mediana	Máximo
20_Graodetrigo1			
21_Graodetrigo2			
22_Graodetrigo3			
23_Graodetrigo4			
24_Graodetrigo5			

Tabela 22 – Imagens tomográficas de amostras de sementes de Abóbora. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 15, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.


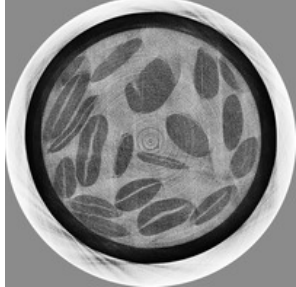
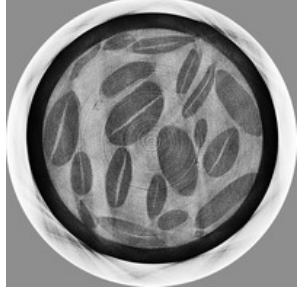

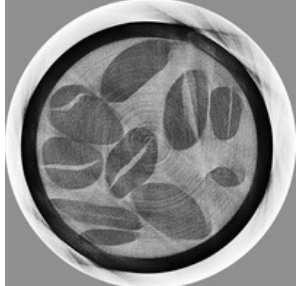
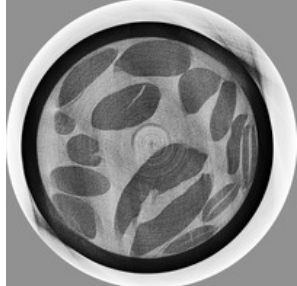
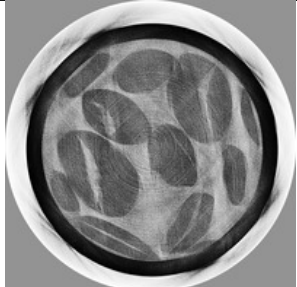
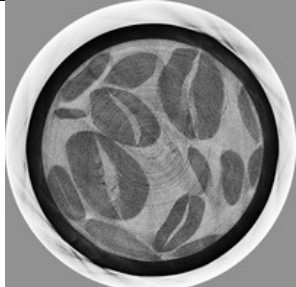
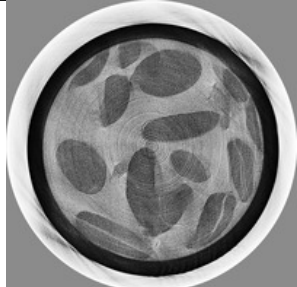
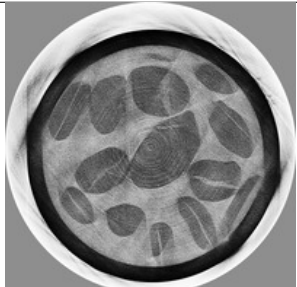
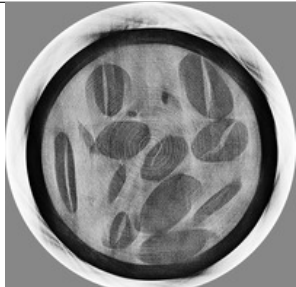

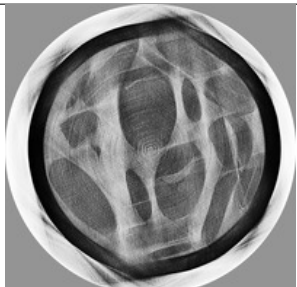

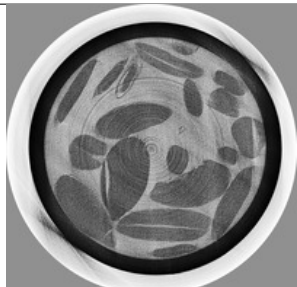
Amostra	Mínimo	Mediana	Máximo
25_SementeAbobora1			
26_SementeAbobora2			
27_SementeAbobora3			
28_SementeAbobora4			
29_SementeAbobora5			



Tabela 23 – Imagens tomográficas de amostras de sementes de Soja. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 16, página 119. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

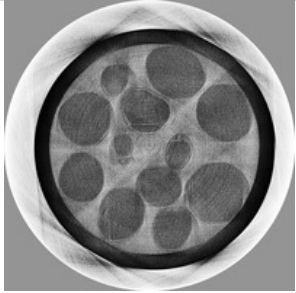
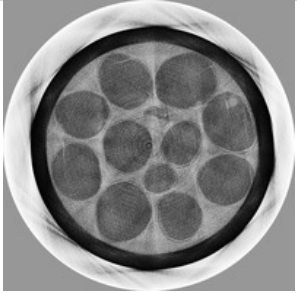
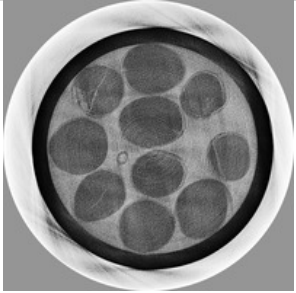
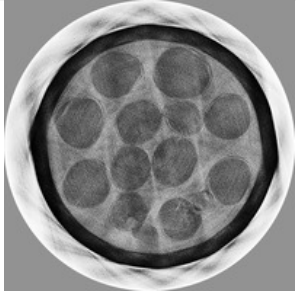
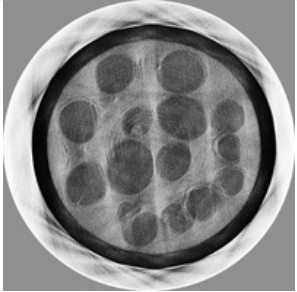
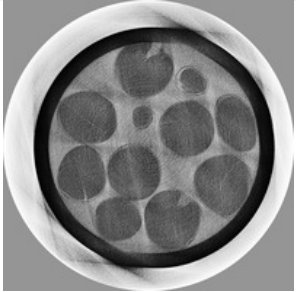
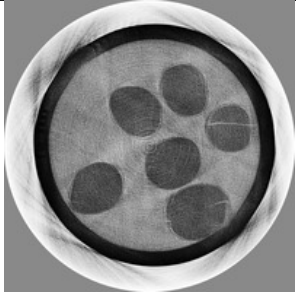
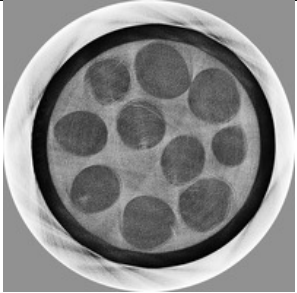
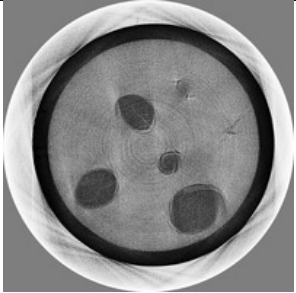
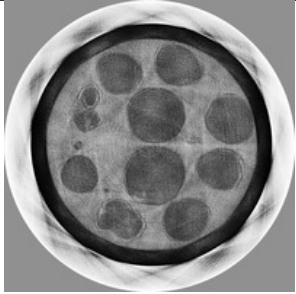
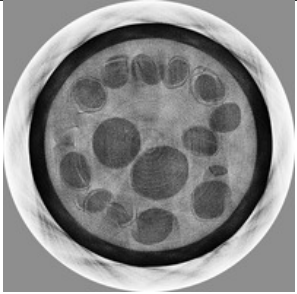
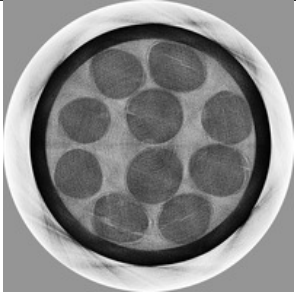
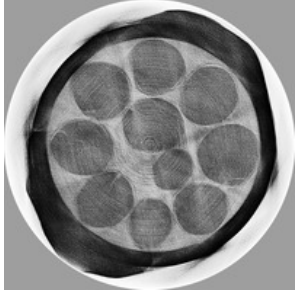
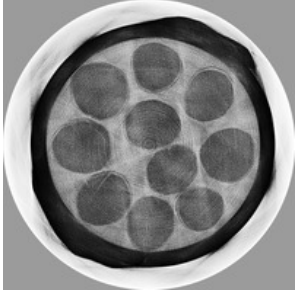
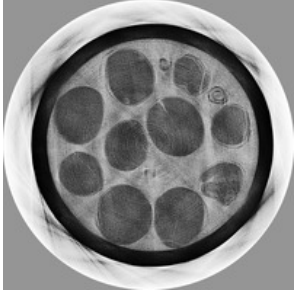


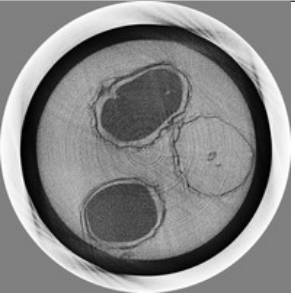


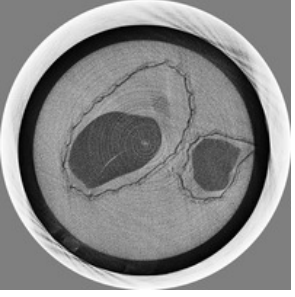


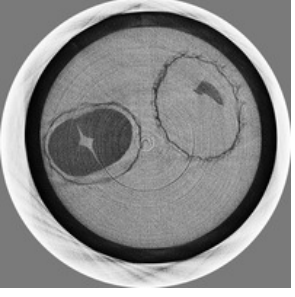



Amostra	Mínimo	Mediana	Máximo
30_Soja1			
31_Soja2			
32_Soja3			
33_Soja4			
34_Soja5			

Tabela 24 – Imagens tomográficas de amostras de Amendoim. Cada linha representa uma amostra com as fatias reconstruídas cujos valores SSIM foram respectivamente, mínimo, mediana e máximo. Os valores SSIM foram apresentados na Tabela 17, página 120. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

Amostra	Mínimo	Mediana	Máximo
35_Amendoim1			
36_Amendoim2			
37_Amendoim3			
38_Amendoim5			

No intuito de fornecer uma visão completa da medida SSIM para a base de imagens utilizadas nesta avaliação, após a análise de cada conjunto de amostras de semente, é apresentado na Figura 44, em gráfico *boxplot*, o cálculo da medida SSIM para as fatias das 33 amostras de sementes, que totalizou a reconstrução e análise de 16.434 fatias.

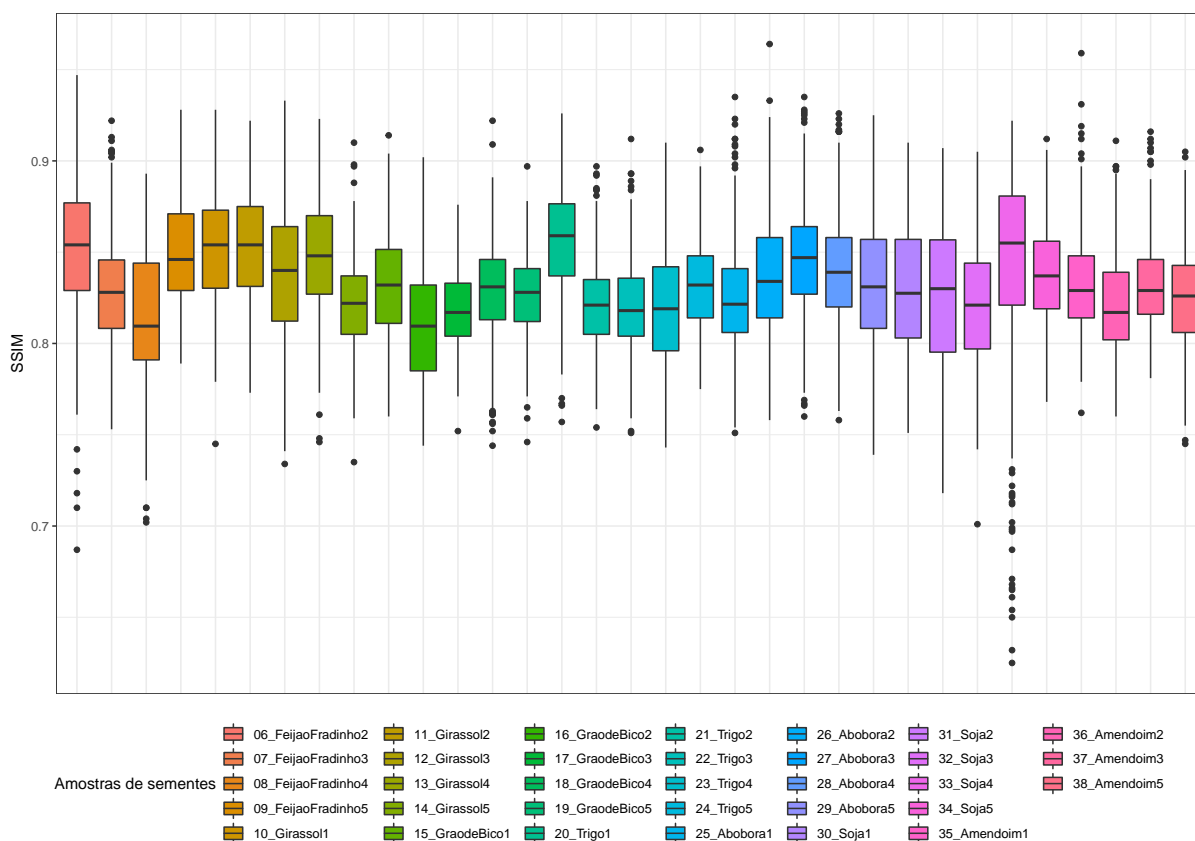


Figura 44 – Análise SSIM das imagens reconstruídas do conjunto de 33 amostras.

Um outro aspecto com relação a medida SSIM é que ao comparar a Figura 44 com a Figura 42, observa-se uma evidência experimental de que a medida SSIM é sensível a variação da taxa de seleção de projeções no sinograma. Na próxima seção são apresentados os resultados obtidos a partir da análise da medida NRMSE para o conjunto de 33 amostras e suas respectivas matrizes de projeções tomográficas.

### 5.2.2 Análise NRMSE

No caso da medida NRMSE, quanto mais próximo de zero menor a diferença entre a imagem de referência e a de análise. A Tabela 25 apresenta os valores NRMSE calculados para o *phantom*.

Tabela 25 – Valores de NRMSE mínimo, mediana e máximo calculados para o *phantom*.

Valor	Fatia	NRMSE	Seleção
Mínimo	988	0,018	63,83%
Mediana	1396	0,081	63,42%
Máximo	1936	0,333	64,45%

A Figura 45 apresenta as fatias 988, 1396 e 1936 que geraram, respectivamente, os valores mínimo, mediana e máximo para a medida NRMSE.

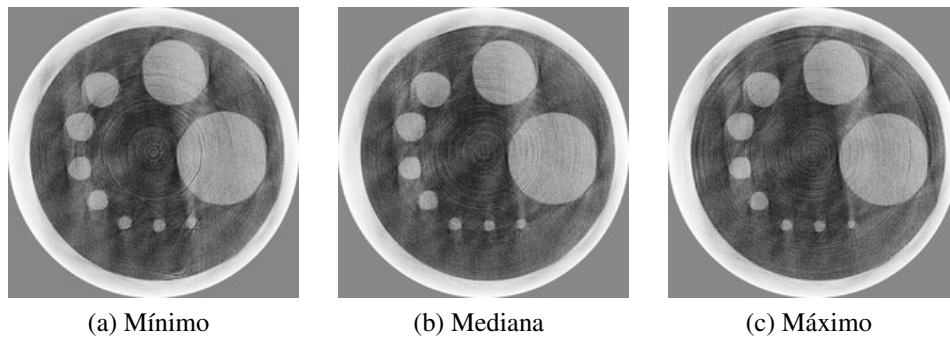


Figura 45 – Fatias do *phantom* que representam os valores mínimo, mediana e máximo da medida NRMSE. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

A Figura 46 apresenta o cálculo do NRMSE para as imagens avaliadas, o que totalizou 16.434 fatias de imagens distribuídas entre 33 amostras de sementes agrícolas.

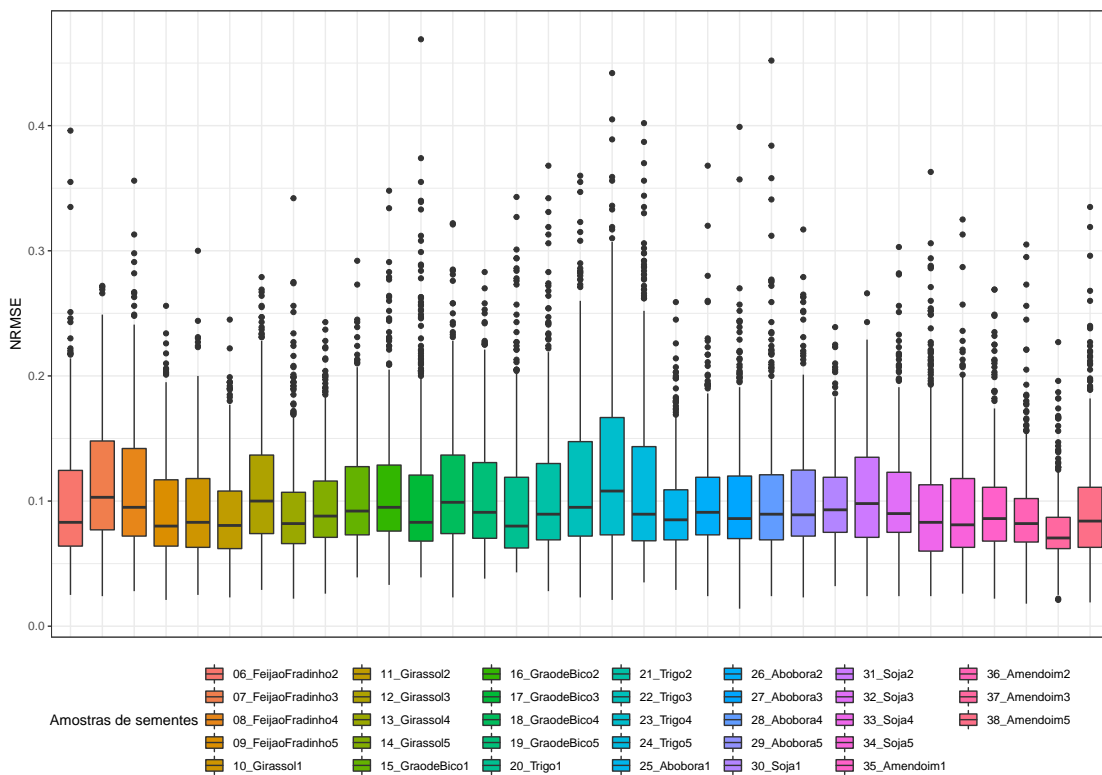


Figura 46 – Análise NRMSE das imagens reconstruídas do conjunto de 33 amostras de sementes agrícolas.

Na Figura 46 é possível verificar que o valor da mediana ficou em torno de 0,100. Apesar do número de *outliers* poder ser considerado alto, observa-se que ficaram com valores abaixo de 0,500. A análise da medida NRMSE buscou complementar a análise do SSIM, pois a medida verifica o erro entre a imagem reconstruída com todas projeções disponíveis e a imagem reconstruída com menos projeções. Ao contrário da medida SSIM, o valor mínimo neste caso representa a fatia que obteve menor erro e, portanto, pode ser considerada a melhor fatia dentre



as fatias escolhidas para análise do *phantom*. Observa-se que a fatia 988 obteve NRMSE de 0,018 para uma taxa de seleção de projeções próxima à da fatia 1396 que obteve o valor 0,081 para a medida NRMSE.

### 5.2.3 Análise PSNR

A análise da medida PSNR teve por objetivo observar a relação sinal-ruído da imagem reconstruída comparada com a imagem de referência de modo a verificar se a seleção de projeções implicou no aumento do ruído comprometendo a etapa de reconstrução bidimensional dos sinogramas.

Logo, tal análise foi motivada pelo fato que ao selecionar projeções ocorreu uma redução na quantidade dos dados, logo a seleção pode ter degradado a imagem gerando artefatos durante a reconstrução tomográfica.

Primeiramente, a Tabela 26 apresenta os valores PSNR calculados para o *phantom* e, posteriormente, os cálculos para as amostras são apresentados.

Tabela 26 – Valores de PSNR mínimo, mediana e máximo calculados para o *phantom*.

Valor	Fatia	PSNR (dB)	Seleção
Mínimo	1066	16,953	64,14%
Mediana	1928	27,027	63,63%
Máximo	988	39,873	63,83%

A Figura 47 apresenta as fatias 1066, referente ao valor mínimo do PSNR; 1928, referente ao valor da mediana do PSNR; 988, referente ao valor máximo do PSNR.

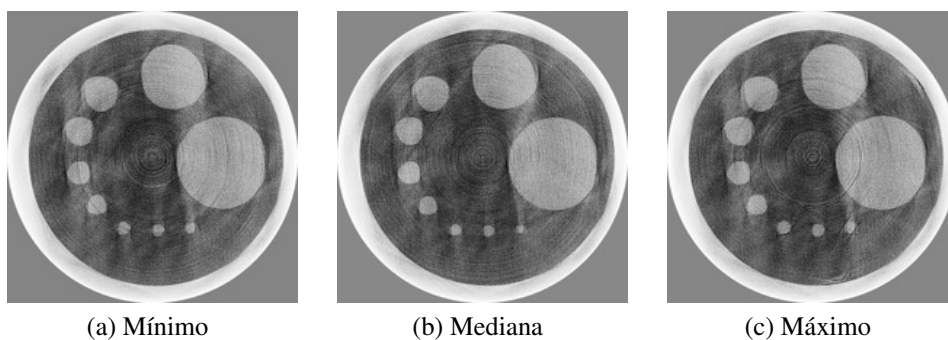


Figura 47 – Fatias do *phantom* que representam os valores mínimo, mediana e máximo da medida NRMSE. Todas as imagens estão representadas pela escala de tons de cinza variando de 0 a 255.

Foi observado que a fatia 988, do *phantom*, gerou os maiores valores para as medidas PSNR e SSIM, enquanto que para a medida NRMSE a fatia 988 gerou o menor valor, o que comprova tal resultado. Tal situação levou ao entendimento de que a redução do número de projeções não comprometeu a qualidade da reconstrução como confirmado pelas medidas SSIM e NRMSE.

A Figura 48 apresenta o cálculo do PSNR para fins de análise da base de dados utilizada nesta avaliação que totalizou 16.434 fatias distribuídas entre 33 amostras de sementes agrícolas.

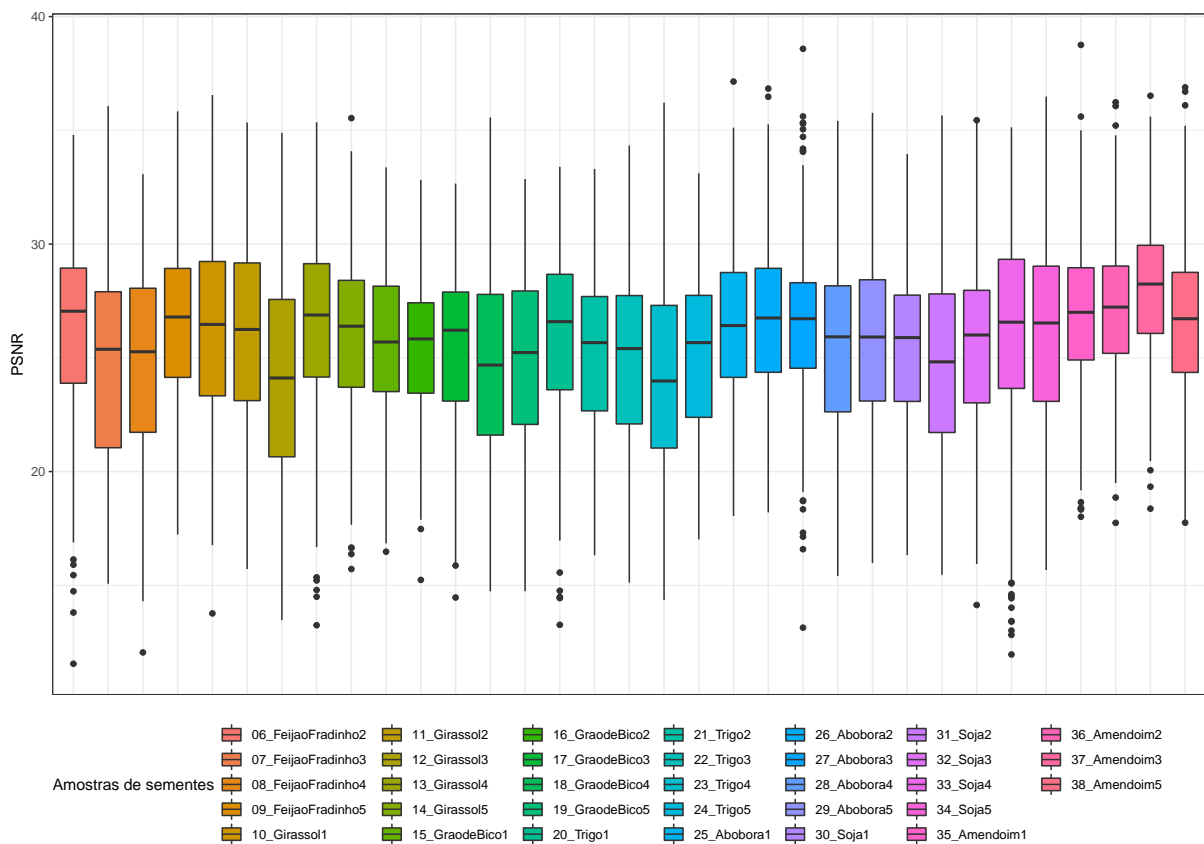


Figura 48 – Análise PSNR das imagens reconstruídas do conjunto de 33 amostras de sementes agrícolas.

Como pode ser observado na Figura 48, a mediana do PSNR para todas as amostras ficou acima de 25 dB, o que indica que o processo de seleção de projeções, não comprometeu a qualidade da reconstrução tomográfica bidimensional.

### 5.3 Reconstrução 2D paralela

Nesta seção é apresentada uma outra abordagem de paralelização da reconstrução bidimensional paralela, implementada no ambiente Apache Spark, diferente do processo de paralelização apresentado na seção 4.6.1. A estratégia de paralelização consistiu em particionar o problema de modo a tornar mais fina a granularidade do paralelismo entre as tarefas de filtragem e reconstrução. A Figura 49 apresenta o diagrama de blocos que elucida o processo de paralelização do algoritmo FBP.

Uma matriz formada por  $N$  projeções, na qual cada projeção contém  $M$  pontos, é submetida a filtragem no domínio da frequência que ocorre em um único processo. Após obter as projeções filtradas, inicia-se o processo de distribuir as projeções pelos diversos processos



disponíveis nos diferentes Nós do *cluster*, no qual cada processo fica responsável por interpolar um conjunto de projeções. Cada projeção recebe uma identificação da amostra e do sinograma a qual ela representa. Posteriormente as projeções interpoladas são retroprojetadas para formar a fatia reconstruída. Neste caso, as retroprojeções são realizadas nos Nós trabalhadores e enviadas para o Nó mestre, que reúne as retroprojeções de uma mesma fatia e as somam, a fim de concluir o processo de reconstrução da imagem tomográfica.

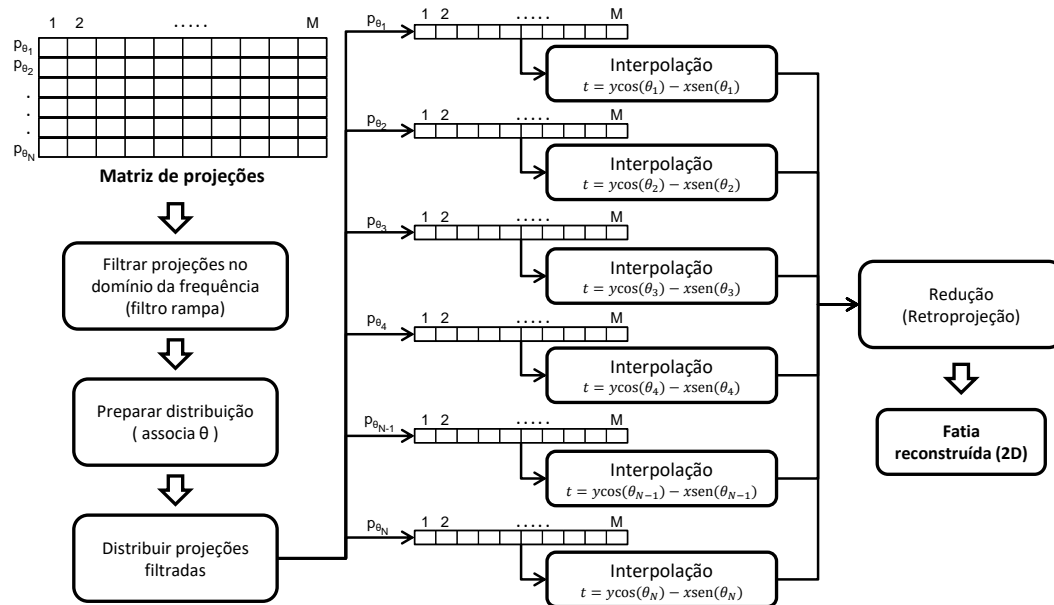


Figura 49 – Diagrama de blocos da reconstrução 2D paralela.

O Código 3 apresenta trecho do código, utilizando PySpark, no qual ocorre a paralelização da reconstrução tomográfica bidimensional.

Código 3 – Trecho do código responsável por paralelizar a reconstrução tomográfica 2D.

```

1 m_projecoes, theta, secao = tupla
2 lin, col = m_projecoes.shape
3
4 # prepara matriz que sera interpolada
5 matriz = zeros( (lin, col+1) )
6 matriz[:,0:col] = m_projecoes
7 matriz[:, -1] = array( theta ).T
8
9 rows = spark.sparkContext.parallelize( matriz )
10
11 # matriz usada na agregacao
12 mZeros = DenseMatrix( self._output_size, self._output_size,
13                       zeros( self._output_size * self._output_size ) )
14
15 fatia = rows.map( lambda r: self.interpolar(r, col) ).treeAggregate(
16                 mZeros, self.adicionar, self.adicionar)

```

O valor *tupla* na linha 1 contém o resultado da etapa de seleção das projeções baseada em energia. Tal resultado consiste na matriz com as projeções selecionadas e filtradas (*m\_projecoes*), o vetor com os ângulos das projeções selecionadas (*theta*) e a seção

(`secao`) na qual a matriz se refere. Na sequência, a preparação para a distribuição das projeções pelo `cluster` é realizada ao adicionar a informação do ângulo da projeção na última posição (linha 7). Na linha 9 ocorre a distribuição da matriz pelo método `parallelize` que envia cada projeção (linha da matriz) para os Nós trabalhadores.

A reconstrução consiste em duas etapas e está codificada na linha 15. A primeira é a interpolação de  $t = y * \cos(\theta) - x * \sin(\theta)$  que ocorre no método `interpolate` e a segunda etapa consiste em somar todas as interpolações em uma única matriz pelo método `treeAggregate`.

O método `treeAggregate` é uma operação de ação no Spark e realiza a agregação de modo diferente ao realizado pelos métodos `aggregate` ou `reduce`, pois tais métodos enviam todos os valores computados para o Nó mestre o que pode provocar um gargalo quando o número de partições e dados é grande. Por outro lado, o `treeAggregate` possui uma estratégia de agregação em árvore, ou seja, multi-nível. Neste caso, os dados são combinados parcialmente entre os executores de modo a reduzir a quantidade de dados enviados para o Nó mestre.

### 5.3.1 Análise de viabilidade

A análise de viabilidade foi realizada em um `cluster` com cinco máquinas do tipo `m5.2xlarge`, sendo uma máquina destinada como `core` e as demais como `workers`. O Quadro 8 apresenta os parâmetros ajustados para a configuração de memória do Apache Spark.

Quadro 8 – Parâmetros e valores adotados para configuração da memória do Apache Spark para análise de viabilidade da reconstrução 2D paralela.

Parâmetro	Valor
<code>spark.executor.cores</code>	5
<code>spark.executor.instances</code>	4
<code>spark.executor.memory</code>	28 GB
<code>spark.executor.memoryOverhead</code>	4 GB
<code>spark.memory.fraction</code>	0.9
<code>spark.memory.storageFraction</code>	0.1
<code>yarn.nodemanager.resource.memory-mb</code>	32768 MB
<code>spark.default.parallelism</code>	40

O total de memória RAM do `cluster` foi de 160 GB, dos quais 140 GB dedicados para o processamento e 20 GB dedicados para `overhead` de memória, conforme indicado nas configurações apresentadas pelo Quadro 8. Os dados, para esta análise, foram obtidos a partir do software Ganglia<sup>1</sup>, que trata-se de um sistema de monitoramento de `clusters` e `grids` computacionais.

Foram avaliados dois atributos: memória RAM e comunicação da rede. Com relação ao uso de memória RAM buscou-se observar a quantidade de memória livre no `cluster`, neste caso, quanto maior a quantidade de memória livre durante o processamento melhor é a indicação de que a estratégia de paralelização foi adequada. Com relação a comunicação da rede buscou-se avaliar a quantidade de bytes que trafegavam pelo `cluster`. Neste caso, entendeu-se que quanto

<sup>1</sup> Disponível em: <<http://ganglia.sourceforge.net/>>

menor a quantidade de bytes transferidos entre as instâncias do cluster, melhor tendeu a ser o desempenho do algoritmo, uma vez que diminuiu o custo de comunicação entre os Nós.

A Figura 50 apresenta a análise de memória RAM livre no *cluster* durante a reconstrução 2D paralela para uma amostra de 1960 sinogramas, em que cada sinograma era composto por 976 projeções de 2000 pontos cada, portanto um sinograma correspondia a uma matriz  $976 \times 2000$  (2k). Além disso, é importante observar que o tempo de processamento foi de cinco horas, iniciado às 10:00 e encerrado às 15:00.

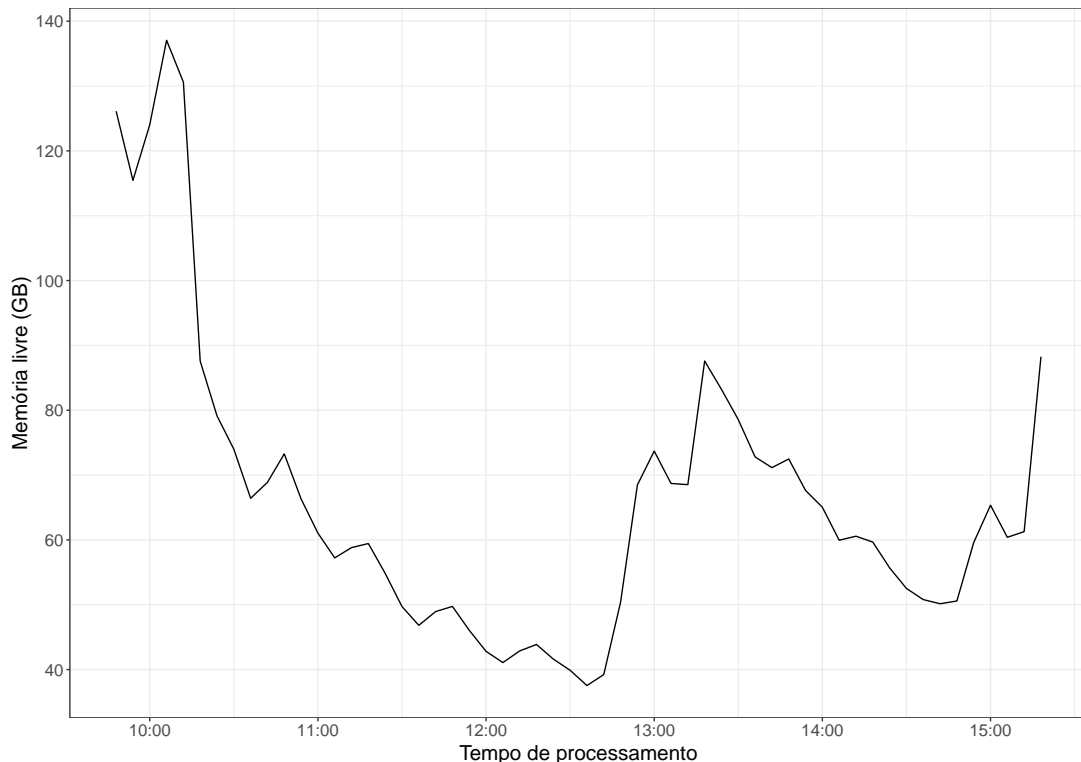


Figura 50 – Análise de memória RAM livre durante reconstrução 2D paralela.

Observou-se que por volta de 12:30 ocorreu a situação em que a quantidade memória RAM livre no *cluster* foi inferior à 40 GB, ou seja muito próximo de ocorrer *overhead* de memória.

Com relação a estratégia de paralelização, observou-se que os recursos `reduce` e `aggregate` enviavam os dados de todas as partições para o *driver* o que exigia dele maior quantidade de memória além de aumentar o tráfego entre as instâncias do *cluster* que precisavam transmitir os dados para o *driver*. Portanto, utilizou-se o recurso `treeAggregate` que buscou minimizar tais custos, uma vez que o recurso oferece um modelo de comunicação de agregação baseada em multi-níveis, no qual os dados são combinados parcialmente em partições inferiores de modo a reduzir o custo de envio e de memória dos dados que chegam ao *driver*. No entanto, devido a característica do problema, o uso do `treeAggregate` não foi suficiente para otimizar o custo de envio e de uso de memória.

Outro aspecto observado, relativo a estratégia, é que a retroprojeção consiste em duas

principais operações, a interpolação e a soma das interpolações que resultará, ao final do processo, na fatia bidimensional reconstruída. A segunda operação, portanto, precisou somar os resultados de todas as interpolações em uma única matriz que, ao final, seria a fatia reconstruída. No entanto, os resultados das interpolações (matriz), estavam nos Nós *worker* e, portanto, fez-se necessário transferir a matriz para o *driver*. A questão é que isso gerou um alto tráfego de comunicação e, principalmente, custo de memória global. No intuito de avaliar este aspecto, observou-se a taxa de transmissão e recepção de bytes durante o processamento da amostra cujo gráfico é apresentado pela Figura 51.

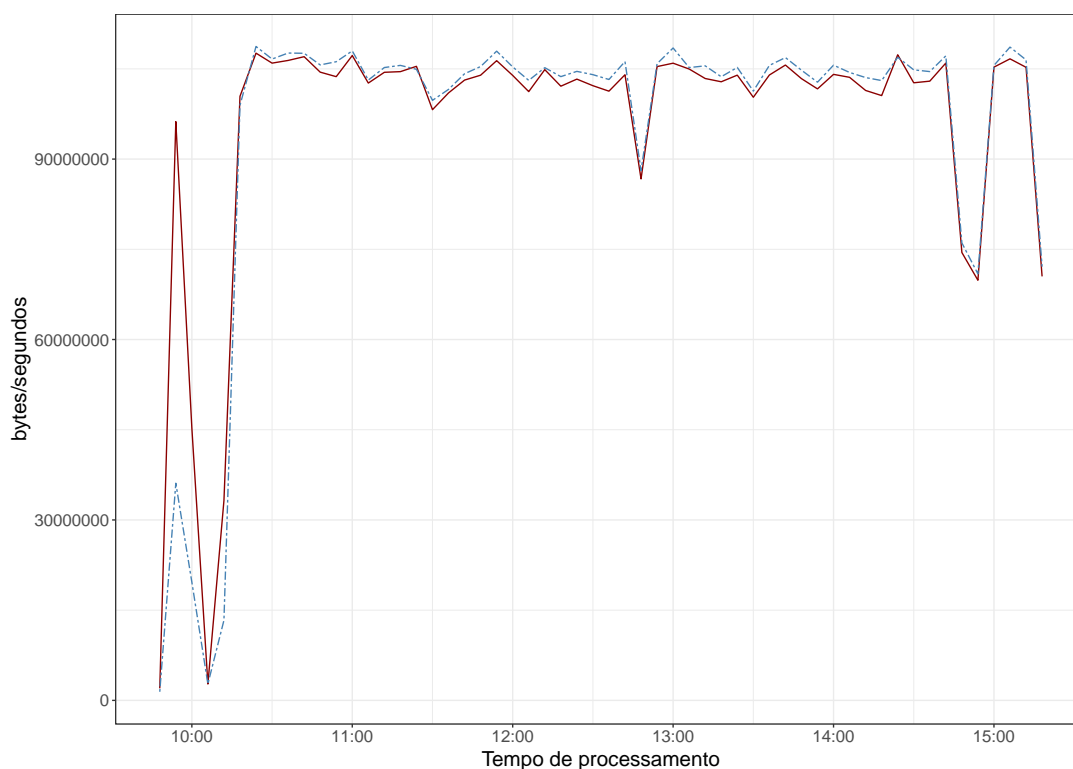


Figura 51 – Análise do tráfego de dados durante reconstrução 2D paralela. A linha cheia representa a quantidade de bytes recebidos e a linha tracejada indica a quantidade de bytes transmitidos.

As altas taxas de transmissão e recepção observadas reforçam o entendimento que o envio da interpolação para o *driver* não se mostrou apropriado. No caso da implementação local, o cálculo da interpolação é realizado sempre na mesma matriz e adicionada em outra matriz resultante que será a fatia reconstruída, logo o consumo de memória é menor. Todavia, o Apache Spark não permite o uso de variáveis compartilhadas para fins de escrita, apenas para fins de leitura (denominadas variáveis *broadcast*).

Dado que uma instância recebe uma parcela do total de projeções por meio de uma estrutura RDD, uma alternativa para otimizar o processamento seria a partir da instância (Nó *worker*) gerar um novo RDD que paralelizaria a interpolação daquela parcela de projeções e ao fim teria-se parte da fatia reconstruída que ao integrar com as demais chegaria-se a fatia total

reconstruída. Todavia, como foi observado durante a revisão da literatura, atualmente o Apache Spark não permite aninhar RDD.

## 5.4 Visualização 3D (volumétrica) de imagens tomográficas

Nesta seção são apresentados os resultados da visualização 3D (volumétrica) para as amostras reconstruídas no ambiente Big Data. Conforme discutido no tópico 4.6.2, as fatias das amostras foram divididas em regiões e organizadas em blocos que foram interpolados para gerar uma parte do volume, ou subvolume, da amostra. Na sequência, os subvolumes foram arquivados no ambiente da Amazon (AWS S3) para depois serem visualizados fora da infraestrutura Big Data organizada neste trabalho. É importante mencionar que tal situação ocorreu a fim de reduzir a taxa de comunicação necessária para o envio dos dados dos subvolumes para o Nó mestre. Portanto, o processo de visualizar uma amostra consistiu em duas etapas: 1) Reunir os subvolumes para organizar o volume completo da amostra; 2) Visualizar o volume completo a partir da ferramenta *itkwidgets*<sup>2</sup>.

Ambas etapas ocorreram fora do ambiente Big Data estruturado neste trabalho e foram utilizados os seguintes recursos: linguagem Python e o ambiente de desenvolvimento Jupyter notebook<sup>3</sup> integrado à biblioteca de visualização VTK da Kitware, por meio do *plugin* *itkwidgets*, conforme apresentado na Figura 52. A visualização foi realizada em um computador com 32 GB de memória RAM e processador Intel core i9.

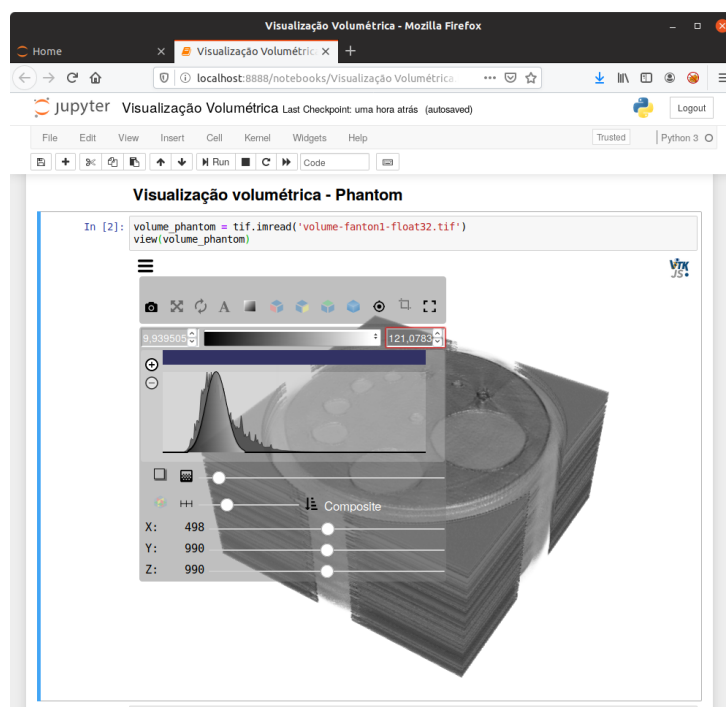


Figura 52 – Ambiente Jupyter notebook integrado à ferramenta *itkwidgets*.

<sup>2</sup> Disponível em: <<https://github.com/InsightSoftwareConsortium/itkwidgets>>

<sup>3</sup> Disponível em: <<https://jupyter.org/>>

Na primeira etapa foi preparado um *script* em Python que recuperou os subvolumes armazenados no AWS S3. Cada subvolume possuía uma identificação de posição, logo foi possível organizar o volume completo com tal informação. A segunda etapa consistiu em carregar o volume completo em memória utilizando os recursos do *plugin itkwidgets* para visualizá-la. A Figura 53 apresenta a visualização volumétrica do *phantom*.

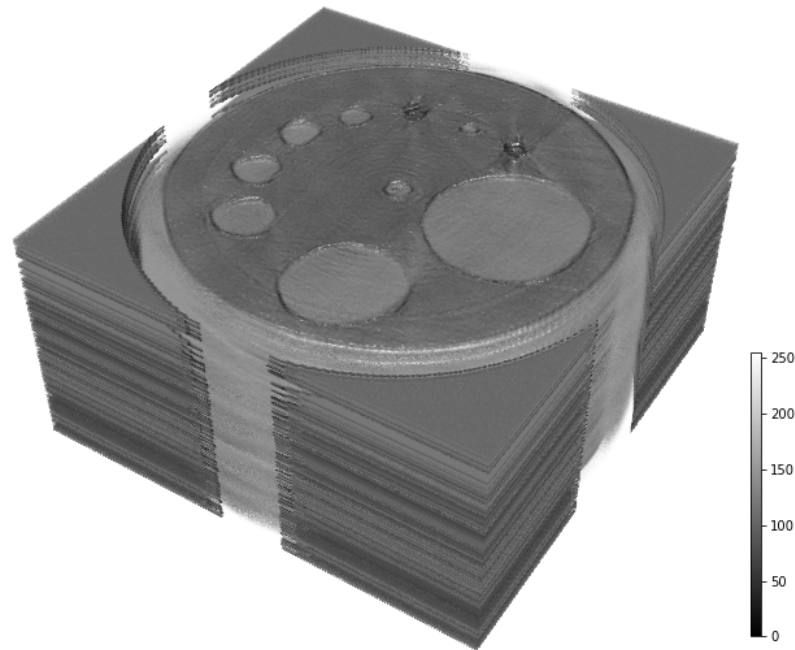


Figura 53 – Visualização volumétrica do *phantom*.

O *plugin itkwidgets* permitiu aplicar cortes, a fim de visualizar o interior do volume. A Figura 54 apresenta corte realizado na visualização volumétrica do *phantom*.

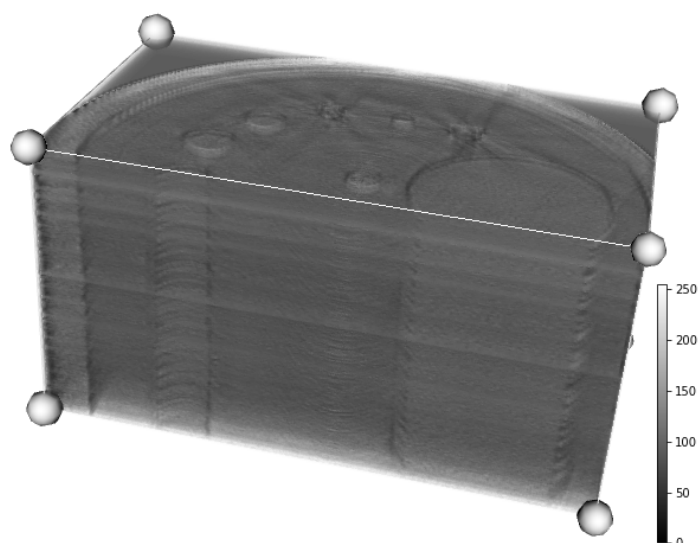


Figura 54 – Corte realizado na visualização volumétrica do *phantom*.

O *phantom* foi reconstruído com 996 fatias com dimensão de  $2000 \times 2000$  pixels. Do total de fatias, 498 eram reais e 498 eram virtuais, geradas pelo processo de interpolação, logo o

volume completo produziu  $2000 \times 2000 \times 996$  voxels o que totalizou 15 GB de dados. A Figura 55 apresenta a visualização volumétrica de uma amostra de Feijão Fradinho.

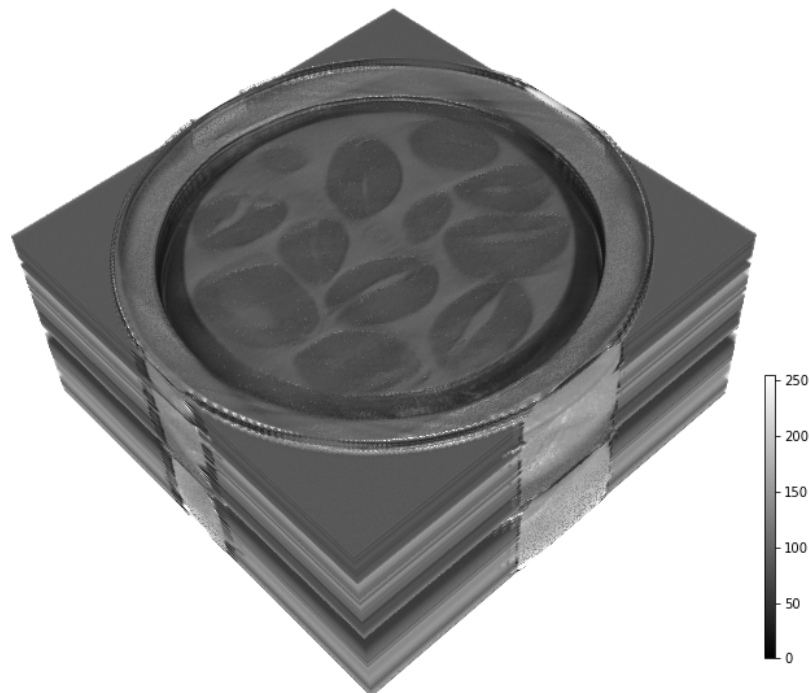


Figura 55 – Visualização volumétrica de uma amostra de Feijão Fradinho.

A Figura 56 apresenta corte realizado na visualização volumétrica de uma amostra de Feijão Fradinho utilizando o *plugin* itkwidgets.



Figura 56 – Visualização volumétrica, com corte, de uma amostra de Feijão Fradinho.

Do mesmo modo que o *phantom*, o volume de uma amostra de Feijão Fradinho foi reconstruído com dimensão  $2000 \times 2000 \times 996$  voxels que resultou em 15 GB de dados.

# Capítulo 6

## CONCLUSÃO

---

---

Este trabalho apresentou um método de reconstrução 2D e 3D (volumétrica) de imagens tomográficas de amostras agrícolas com o emprego de técnicas de Big Data. A revisão da literatura permitiu, entre outros pontos, verificar o interesse pelos temas Big Data e Ciência dos Dados, particularmente no ambiente acadêmico e científico. Além disso, considerando que o conceito Big Data ainda não está bem estabelecido, o estudo possibilitou avaliar as diferentes definições visando caracterizá-lo adequadamente dentro de um contexto atual e customizado para aplicações na área da agricultura digital.

Verificou-se que na agricultura tem ocorrido um significativo aumento de dados e informações provenientes de sementes, solos, plantas, clima, sistemas de manejo, insumos utilizados, entre outros. Portanto, o termo Big Data pode ser associado a este cenário como a capacidade de extrair informações de diferentes fontes inferindo novas linhas de raciocínio e integração entre sistemas visando aprimorar os processos de análise e de auxílio à tomada de decisão. Nesse sentido, a tomografia computadorizada pode ser considerada uma ferramenta útil para análises de amostras agrícolas que produz grandes quantidades de dados o que demanda grande capacidade de processamento. Logo, o método desenvolvido e apresentado neste trabalho viabiliza a reconstrução de imagens tomográficas em ambiente Big Data e permite maior número de análises tomográficas em um modelo escalável.

Para a execução do método desenvolvido foi necessário estruturar um *cluster* de computadores. A infraestrutura utilizada para tal fim foi provida por meio do serviço da Amazon AWS. Neste contexto, foram avaliadas 12 diferentes configurações de *clusters* em que se definiu a configuração que permitiu utilizar a maior quantidade de dados tomográficos, com a maior eficiência. A configuração que prevaleceu foi a que continha 6 Nós, pois apresentou eficiência superior a configuração que continha 10 Nós, embora o valor de Speedup tenha sido inferior. Logo, foi possível constatar que maior valor de Speedup não, necessariamente, implica em maior eficiência. É importante observar que a eficiência apresentou valores maiores que 1. O entendimento foi que para a obtenção do tempo de processamento sequencial, a cada iteração do método, apenas uma matriz de projeção era carregada em memória o que demandou maior tempo para a reconstrução. Por outro lado, para a obtenção do tempo de processamento em paralelo, o



Apache Spark carregou as matrizes de projeções em memória, ou grande parte delas, as quais foram distribuídas pelos Nós trabalhadores. Adicionalmente, a análise do custo de processamento paralelo/sequencial corroborou o entendimento de que a medida de eficiência representa uma informação mais apurada sobre a arquitetura do ambiente.

Um outro aspecto importante foi a estratégia de paralelização adotada para a reconstrução bidimensional. Neste ponto vale mencionar o estudo realizado que buscou diminuir a granularidade de modo a encontrar um equilíbrio entre custo de comunicação e de memória. A estratégia adotada foi a de distribuir as projeções e, neste caso, a granularidade foi considerada fina. Tal estratégia foi implementada no Apache Spark e os resultados obtidos e apresentados para as análises de memória RAM livre e tráfego de dados tomográficos mostraram que ao reduzir a granularidade aumentou-se tanto o custo de comunicação quanto o custo de memória. A solução encontrada para o problema foi adotar a estratégia que consistiu em paralelizar as matrizes de projeções, ao invés das projeções individuais. Neste caso, a granularidade foi considerada média, uma vez que lidou com blocos maiores de dados tomográficos. A solução reduziu o custo de comunicação, ou seja, de transferir os resultados da reconstrução tomográfica dos Nós trabalhadores para o Nó mestre. No entanto, ao reduzir o custo de comunicação aumentou-se a demanda de memória RAM nos Nós trabalhadores. Um entendimento decorrente do estudo das estratégias de paralelização é sobre a importância da modelagem do problema em ambiente Big Data, que exige compreender a natureza dos dados e suas propriedades únicas bem como das ferramentas que serão utilizadas, a exemplo do Apache Spark, a fim de preparar modelos adequados às demandas de processamento de dados.

Um ponto de relevância apresentado neste trabalho refere-se a seleção de projeções tomográficas que buscou selecionar, em cada sinograma, o menor número de projeções que melhor representasse o objeto na imagem reconstruída. Foi considerada a informação de energia de cada projeção de forma a se buscar reconhecer aquelas mais relevantes para a obtenção da reconstrução tomográfica em duas dimensões. Além disso, a organização em classes de energia das projeções tomográficas, mostrou-se uma alternativa adequada por considerar todo o espectro das energias contidas em um sinograma. Foi considerado um conjunto de 33 amostras de sementes e um *phantom* heterogêneo de plexiglass que totalizou 66.640 matrizes de projeções, ou 242 GB de dados tomográficos. Deste conjunto foram consideradas 16.932 matrizes de projeções, ou 498 matrizes por amostra, para fins de avaliação da seleção das projeções tomográficas e da qualidade da reconstrução bidimensional. Para as 16.932 matrizes, o algoritmo selecionou 61,47% a 71,72% das projeções, ou seja, houve uma redução em torno de 28% a 38% por matriz de projeções analisada. A métrica SSIM foi calculada para cada matriz de projeções e observou-se a medida da mediana para os valores SSIM de cada amostra. Neste sentido, a análise mostrou que a reconstrução tomográfica das amostras, em duas dimensões, a partir das projeções selecionadas, levou à obtenção do valor SSIM superior a 0,800, para todas as amostras analisadas. Além disso, foi possível observar situações como a da amostra de Girassol (*14\_Girassol5*) que obteve a taxa de 65,98% de seleção das projeções ao passo que a reconstrução levou ao SSIM

de 0,910, ou a da amostra de Soja (*33\_Soja4*) que obteve a taxa de 62,40% ao passo que a reconstrução levou ao SSIM de 0,625. Vale destacar, ainda, que as análises NRMSE e PSNR corroboraram com os resultados obtidos pela análise SSIM. Portanto, os resultados obtidos e apresentados no Capítulo 5, mostraram que a redução do número de projeções para as amostras de sementes (Amendoim, Feijão Fradinho, Girassol, Grão de Bico, Trigo, Abóbora e Soja) não comprometeu a estrutura das informações contidas nas imagens reconstruídas conforme observado pelas análises do SSIM, NRMSE e PSNR. A redução do número de projeções levou a uma melhora do Speedup e, conseqüentemente, do desempenho viabilizando um maior número de análises tomográficas. Logo, uma conclusão que se pode chegar é que o processo de seleção de projeções tomográficas é importante, inclusive em um cenário de Big Data, pois permite lidar com tomografia de alta resolução.

Com relação a reconstrução 3D (volumétrica) destaca-se que o tempo de processamento foi menor do que o tempo de processamento observado para a reconstrução 2D, pois os dados tomográficos dos volumes reconstruídos foram armazenados em disco. No entanto, verificou-se que a visualização de grandes volumes de dados demandou grande capacidade computacional.

A principal contribuição deste trabalho foi a de preparar um arcabouço para reconstrução tomográfica de imagens de alta resolução para amostras agrícolas preparado para ser executado em ambiente Big Data. Neste sentido, o método desenvolvido mostrou-se desafiador ao lidar com processamento de imagens de alta resolução em ambiente distribuído e paralelo e destacou a importância de se buscar preparar os atuais algoritmos, bem como a oportunidade para se trabalhar com grandes quantidades de imagens a fim de atender a crescente demanda agrícola por análises de amostras no processo de tomada de decisão.

Como proposta de trabalhos futuros, pretende-se avaliar outras estratégias de paralelização das matrizes de projeções bem como avaliar outras distribuições, a exemplo das distribuições Gama, Exponencial e Qui-Quadrado, que podem vir a ser empregadas na seleção de projeções tomográficas. Além disso, avalia-se estabelecer a estruturação de um Sistema de Suporte à Decisão baseado em Big Data e Ciência dos Dados conforme indicado no Apêndice B, no qual este trabalho seria umas das camadas de tal sistema.

## 6.1 Principais contribuições

Nesta tese trabalhou-se com a reconstrução tomográfica de amostras agrícolas, em alta resolução, o que exigiu o desenvolvimento de métodos e uso de técnicas capazes de lidar com a grande quantidade de dados tomográficos. Portanto, a seguir, são citadas as principais contribuições deste trabalho:

- **Concepção de método de reconstrução tomográfica utilizando a abordagem MapReduce:** a fim de lidar com a grande quantidade de dados tomográficos foi concebido um

método de reconstrução tomográfica 2D e 3D (volumétrica) que levou em consideração a abordagem de programação MapReduce, o que permitiu estruturar as etapas do método de modo paralelo e distribuído. Além disso, o método concebido foi implementado utilizando o Apache Spark. Os resultados mostraram que o método mostrou-se útil para viabilizar a análise tomográfica de grandes quantidades de amostras agrícolas, além de possibilitar a redução no número de projeções tomográficas sem comprometer a qualidade das imagens reconstruídas.

- **Mineração de projeções tomográficas com base na densidade espectral:** o intuito da mineração das projeções tomográficas em um sinograma foi a de selecionar as projeções que carregavam maior quantidade de informação de modo a não comprometer a qualidade da reconstrução. Portanto, foi apresentado o método de seleção que levou em consideração a energia de cada projeção, calculada a partir da densidade espectral. A partir disso foi possível estabelecer um critério de seleção que permitiu identificar as projeções com maior quantidade de informação. Os valores calculados para a métrica SSIM possibilitaram observar que a qualidade da reconstrução tomográfica não foi comprometida com a mineração das projeções.
- **Estruturação de ambiente Big Data propício para reconstrução tomográfica:** para viabilizar a execução do método de reconstrução tomográfica foi necessário o planejamento, organização e estruturação de um ambiente Big Data cuja infraestrutura foi composta por um *cluster*, instalado na plataforma AWS, além de um conjunto de tecnologias que possibilitou o funcionamento do ambiente Big Data. A definição da configuração mais adequada para o ambiente foi obtida por meio da avaliação de 12 diferentes configurações de *clusters* e o processamento de 427, 56 GB de dados tomográficos.

## REFERÊNCIAS

---

---

AGARWAL, R.; DHAR, V. Editorial: Big data, data science, and analytics: The opportunity and challenge for its research. *Information Systems Research*, v. 25, n. 3, p. 443–448, 2014. Disponível em: <<http://dx.doi.org/10.1287/isre.2014.0546>>. Citado na página 60.

ALVES, G. M.; CRUVINEL, P. E. Big data environment for agricultural soil analysis from CT digital images. In: *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. [S.l.]: IEEE, 2016. Citado na página 27.

ALVES, G. M.; CRUVINEL, P. E. Big data infrastructure for agricultural tomographic images reconstruction. *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, IEEE, Jan 2018. Disponível em: <<http://dx.doi.org/10.1109/ICSC.2018.00071>>. Citado na página 27.

AMBERT-SANCHEZ, M.; MICKELSON, S. K.; AHMED, S. I.; GRAY, J. N.; WEBBER, D. Evaluating soil tillage practices using x-ray computed tomography and conventional laboratory methods. *Transactions of the ASABE*, American Society of Agricultural and Biological Engineers (ASABE), v. 59, n. 2, p. 455–463, apr 2016. Citado na página 40.

ARNOLD, E. The big deal about big data. In: *INFORMATION TODAY*. [S.l.], 2012. Citado na página 55.

ASSIS, V. C.; SALVADEO, D. H.; MASCARENHAS, N. D.; LEVADA, A. L. Double noise filtering in CT: Pre- and post-reconstruction. In: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <<http://dx.doi.org/10.1109/SIBGRAPI.2015.42>>. Citado na página 40.

BALOGUN, F.; CRUVINEL, P. Compton scattering tomography in soil compaction study. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier BV, v. 505, n. 1-2, p. 502–507, jun 2003. Citado na página 39.

BASSOI, L. H.; MIELE, A.; JUNIOR, C. R.; GEBLER, L.; FLORES, C. A.; ALBA, J. M. F.; GREGO, C. R.; TERRA, V. S. S.; TIMM, L. C.; NASCIMENTO, P. S. Agricultura de precisão em fruticultura. In: \_\_\_\_\_. *Agricultura de Precisão: resultados de um novo olhar*. [S.l.: s.n.], 2014. Citado na página 26.

BEGOLI, E.; HOREY, J. Design principles for effective knowledge discovery from big data. In: *Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*. [S.l.: s.n.], 2012. Citado na página 55.

BERALDO, J. M. G.; JUNIOR, F. de A. S.; CRUVINEL, P. E. Application of x-ray computed tomography in the evaluation of soil porosity in soil management systems. *Engenharia Agrícola*, FapUNIFESP (SciELO), v. 34, n. 6, p. 1162–1174, 12 2014. Disponível em: <<http://dx.doi.org/10.1590/S0100-69162014000600012>>. Citado 3 vezes nas páginas 9, 40 e 78.

- BERNADI, A. C. C.; NAIME, J. ao M.; RESENDE, A. V.; BASSOI, L. H.; INAMASU, R. Y. *Agricultura de Precisão: resultados de um novo olhar*. [S.l.: s.n.], 2014. Nenhuma citação no texto.
- BEUTLER, F. J.; LENEMAN, O. A. Random sampling of random processes: Stationary point processes. *Information and Control*, Elsevier BV, v. 9, n. 4, p. 325–346, aug 1966. Citado na página 94.
- BLAS, J. G.; ABELLA, M.; ISAILA, F.; CARRETERO, J.; DESCO, M. Surfing the optimization space of a multiple-gpu parallel implementation of a x-ray tomography reconstruction algorithm. *Journal of Systems and Software*, Elsevier BV, v. 95, p. 166–175, 9 2014. ISSN 0164-1212. Citado na página 34.
- BRAGA NETO, U. M. *Reconstrução Volumétrica e Análise de Imagens Tridimensionais por Morfologia Matemática*. Dissertação (Mestrado) — Unicamp, 1994. Citado na página 48.
- CALLEBAUT, W. Scientific perspectivism: a philosopher of science’s response to the challenge of big data biology. *Studies in history and philosophy of biological sciences*, v. 43, 2012. Citado na página 55.
- CARVALHO, D. C. *Obtenção de padrões sequenciais em data streams atendendo requisitos de big data*. Dissertação (Mestrado) — UFSCar, 2016. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/8280>>. Citado na página 34.
- CGI.BR. *Pesquisa sobre o uso das tecnologias da informação e comunicação nos domicílios brasileiros: TIC Domicílios 2014*. São Paulo: Comitê Gestor da Internet no Brasil, 2015. Citado na página 51.
- CGI.BR. *Pesquisa sobre o uso das tecnologias da informação e comunicação nos domicílios brasileiros: TIC Domicílios 2018*. [S.l.: s.n.], 2019. Citado 3 vezes nas páginas 8, 51 e 52.
- CHEN, C. L. P.; ZHANG, C. Data-intensive applications, challenges, techniques and technologies: a survey on Big Data. *Information Sciences*, v. 275, p. 314–347, 8 2014. Citado 10 vezes nas páginas 8, 55, 57, 58, 60, 61, 62, 63, 64 e 65.
- CHEN, H.; CHIANG, R. H. L.; STOREY, V. C. Business intelligence and analytics: from big data to big impact. *MIS Quartely*, v. 36, n. 4, 2012. Citado na página 55.
- CHIFFRE, L. D.; CARMIGNATO, S.; KRUTH, J.-P.; SCHMITT, R.; WECKENMANN, A. Industrial applications of computed tomography. v. 63, n. 2, p. 655–677, jan. 2014. ISSN 0007-8506. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0007850614001930>>. Citado na página 39.
- CRESTANA, S.; MASCARENHAS, S.; POZZI-MUCELLI, R. S. STATIC AND DYNAMIC THREE-DIMENSIONAL STUDIES OF WATER IN SOIL USING COMPUTED TOMOGRAPHIC SCANNING. *Soil Science*, Ovid Technologies (Wolters Kluwer Health), v. 140, n. 5, p. 326–332, nov 1985. Citado na página 39.
- CROWLEY, P. J. *The B-spline Wavelet recurrence relation and B-spline Wavelet Interpolation*. 1996. Honors Projects. Paper 9. Disponível em: <[http://digitalcommons.iwu.edu/math\\_honproj/9](http://digitalcommons.iwu.edu/math_honproj/9)>. Citado na página 50.

CRUVINEL, P.; CESAREO, R.; CRESTANA, S.; MASCARENHAS, S. X- and gamma -rays computerized minitomograph scanner for soil science. *IEEE Transactions on Instrumentation and Measurement*, Institute of Electrical and Electronics Engineers (IEEE), v. 39, n. 5, p. 745–750, 1990. Citado na página 39.

CRUVINEL, P.; CRESTANA, S. The use of x- and  $\gamma$ -rays dedicated computerized minitomography scanner in agriculture due to the limitations imposed by medical computerized tomography scanners. In: *Proceedings II Workshop on Cybernetic Vision*. Institute of Electrical & Electronics Engineers (IEEE), 1996. p. 208–212. Disponível em: <<http://dx.doi.org/10.1109/CYBVIS.1996.629465>>. Citado na página 39.

CRUVINEL, P.; PEREIRA, M.; SAITO, J.; COSTA, L. da F. Performance improvement of tomographic image reconstruction based on DSP processors. *IEEE Trans. Instrum. Meas.*, Institute of Electrical & Electronics Engineers (IEEE), v. 58, n. 9, p. 3295–3304, 9 2009. Disponível em: <<http://dx.doi.org/10.1109/TIM.2009.2022378>>. Citado na página 33.

CRUVINEL, P. E. *Minitomógrafo de raio-X e raio- $\gamma$  computadorizado para aplicações multidisciplinares*. phdthesis — Unicamp, 1987. Disponível em: <<http://repositorio.unicamp.br/handle/REPOSIP/260480>>. Citado 2 vezes nas páginas 38 e 39.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 51, n. 1, p. 107, jan 2008. Citado 2 vezes nas páginas 67 e 70.

DINIZ, P. S. R.; SILVA, E. A. B.; NETTO, S. L. *Digital Signal Processing*. Cambridge University Press, 2010. ISBN 0521887755. Disponível em: <[http://www.ebook.de/de/product/11090272/paulo\\_s\\_r\\_universidade\\_federal\\_do\\_rio\\_de\\_janeiro\\_diniz\\_eduardo\\_a\\_b\\_da\\_universidade\\_federal\\_do\\_rio\\_de\\_janeiro\\_silva\\_sergio\\_l\\_universidade\\_federal\\_do\\_rio\\_de\\_janeiro\\_netto\\_digital\\_signal\\_processing.html](http://www.ebook.de/de/product/11090272/paulo_s_r_universidade_federal_do_rio_de_janeiro_diniz_eduardo_a_b_da_universidade_federal_do_rio_de_janeiro_silva_sergio_l_universidade_federal_do_rio_de_janeiro_netto_digital_signal_processing.html)>. Citado na página 94.

DITTER, A.; FEY, D.; SCHON, T.; OECKL, S. On the way to big data applications in industrial computed tomography. In: *2014 IEEE International Congress on Big Data*. Institute of Electrical & Electronics Engineers (IEEE), 2014. p. 792–793. Disponível em: <<http://dx.doi.org/10.1109/BigData.Congress.2014.125>>. Citado 2 vezes nas páginas 27 e 53.

ELGENDY, N.; ELRAGAL, A. Big data analytics: A literature review paper. *Perner P. (eds) Advances in Data Mining*, 2014. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-319-08976-8\\_16](https://link.springer.com/chapter/10.1007/978-3-319-08976-8_16)>. Citado na página 57.

EMANI, C. K.; CULLOT, N.; NICOLLE, C. Understandable Big Data: A Survey. *Computer Science Review*, v. 17, p. 70–81, 2015. Citado na página 60.

FARIA, L. N. *Reconstrução 3D de imagens tomográficas de raios-x de arcos coronais em ambiente paralelo*. Dissertação (Mestrado) — Universidade Federal de São Carlos, 2003. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/541>>. Citado na página 33.

FERREIRA, A. B. H. *Miniaurélio: o minidicionário da língua portuguesa*. 7. ed. Curitiba: Ed. Positivo, 2008. Citado na página 53.

FOSTER, I. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. [S.l.]: Pearson, 1995. ISBN 0201575949. Citado na página 66.



- GERALDO, R. J.; CURA, L. M. V.; CRUVINEL, P. E.; MASCARENHAS, N. D. A. Low dose CT filtering in the image domain using MAP algorithms. *IEEE Transactions on Nuclear Science*, Institute of Electrical and Electronics Engineers (IEEE), v. 64, n. 6, p. 1506–1517, jun 2017. Citado na página 40.
- GOMES-JUNIOR, F. G.; VAZ, C. M. P.; CICERO, S. M.; JORGE, L. A. C. Procedimentos para avaliação da estrutura de sementes de soja e milho por microtomografia computadorizada de raios X. In: *Simpósio Nacional de Instrumentação Agropecuária*. [S.l.: s.n.], 2014. p. 621–624. Citado na página 25.
- GORDON-MURNANE, L. *Big data: a big opportunity for librarians*. 2012. Online. Citado na página 55.
- GRAMA, A.; KARYPIS, G.; KUMAR, V. *Introduction to Parallel Computing*. ADDISON WESLEY PUB CO INC, 2003. ISBN 0201648652. Disponível em: <[http://www.ebook.de/de/product/3255091/ananth\\_grama\\_george\\_karypis\\_vipin\\_kumar\\_introduction\\_to\\_parallel\\_computing.html](http://www.ebook.de/de/product/3255091/ananth_grama_george_karypis_vipin_kumar_introduction_to_parallel_computing.html)>. Citado na página 67.
- GREVILLE, T. N. E. *Theory and applications of spline functions*. New York, Academic Press, 1969. Disponível em: <<https://lcn.loc.gov/69013490>>. Citado na página 49.
- GRIFFIN, T.; LOWENBERG-DEBOER, J. Worldwide adoption and profitability of precision agriculture implications for brazil. *Revista de Política Agrícola*, n. 4, 2005. Citado na página 23.
- GUNARATHNE, T. *Hadoop MapReduce v2 Cookbook - Second Edition*. Packt Publishing, 2015. ISBN 978-1-78328-547-1. Disponível em: <<https://www.amazon.com/Hadoop-MapReduce-v2-Cookbook-Second-ebook/dp/B00U1D9WT6?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00U1D9WT6>>. Citado 3 vezes nas páginas 8, 70 e 71.
- HADOOP, A. *Apache Hadoop YARN*. 2018. Último acesso em 15/02/2019. Disponível em: <<https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>>. Citado 2 vezes nas páginas 8 e 71.
- HAINSWORTH, J.; AYLMOORE, L. The use of computer assisted tomography to determine spatial distribution of soil water content. *Australian Journal of Soil Research*, CSIRO Publishing, v. 21, n. 4, p. 435, 1983. Citado na página 39.
- HEERAMAN, D.; HOPMANS, J.; CLAUSNITZER, V. Three dimensional imaging of plant roots in situ with x-ray computed tomography. *Plant and Soil*, Springer Nature, v. 189, n. 2, p. 167–179, feb 1997. Citado na página 40.
- HEROLD, F.; TISCHENKO, O.; SEIDL, C.; KURFISS, M. Fast and analytical exact reconstruction if large CT-Volumes. In: *Proceedings of 18th World Conference on Non-Destructive Testing*. [s.n.], 2012. Disponível em: <<http://www.ndt.net/article/wcndt2012/toc.htm>>. Citado na página 32.
- HSIEH, J. *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*. [S.l.]: JOHN WILEY & SONS INC, 2009. ISBN 0470563532. Citado 3 vezes nas páginas 38, 41 e 92.
- INAMASU, R. Y.; BERNADI, A. C. de C. Agricultura de precisão. In: \_\_\_\_\_. *Agricultura de Precisão: resultados de um novo olhar*. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 24 e 26.



JACOBS, A. The pathologies of big data. *Communications of the ACM.*, v. 52, n. 8, 2009. Citado na página 56.

JORGENSEN, J. S.; SIDKY, E. Y. How little data is enough? phase-diagram analysis of sparsity-regularized x-ray computed tomography. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society, v. 373, n. 2043, p. 20140387–20140387, may 2015. Citado na página 30.

JUNIOR, L. F.; OLIVEIRA, J. C. M.; BASSOI, L. H.; VAZ, C. M. P.; MACEDO, A.; BACCHI, O. O. S.; REICHARDT, K.; CAVALCANTI, A. C.; SILVA, F. H. B. B. Tomografia computadorizada na avaliação da densidade de um solo do semi-árido brasileiro. *Revista Brasileira de Ciência do Solo*, FapUNIFESP (SciELO), v. 26, n. 4, p. 835–842, dec 2002. Citado na página 40.

KAK, A. C.; SLANEY, M. *Principles of Computerized Tomographic Imaging*. [S.l.]: IEEE Press, 1989. ISBN 0879421983. Citado 2 vezes nas páginas 30 e 41.

KATSOGRIDAKIS, P.; PAPAGIANNAKI, S.; PRATIKAKIS, P. Execution of recursive queries in apache spark. In: *Lecture Notes in Computer Science*. [S.l.]: Springer International Publishing, 2017. p. 289–302. Citado na página 74.

KHAN, M. A.; UDDIN, M. F.; GUPTA, N. Seven Vs of Big Data. In: *American Society for Engineering Education, Proceedings of 2014 Zone 1 Conference of the*. [S.l.: s.n.], 2014. Citado na página 58.

KHAN, S.; AHMAD, M. K. A study on b-spline wavelets and wavelet packets. *Applied Mathematics*, Scientific Research Publishing, Inc., v. 05, n. 19, p. 3001–3010, 2014. Citado na página 50.

KONTOGHIORGHES, E. J. *Handbook of Parallel Computing and Statistics*. Chapman and Hall/CRC, 2005. ISBN 082474067X. Disponível em: <[https://www.ebook.de/de/product/4024818/erricos\\_john\\_kontoghiorghes\\_handbook\\_of\\_parallel\\_computing\\_and\\_statistics.html](https://www.ebook.de/de/product/4024818/erricos_john_kontoghiorghes_handbook_of_parallel_computing_and_statistics.html)>. Citado na página 82.

LAIA, M. A. M.; LEVADA, A. L. M.; BOTEAGA, L. C.; PEREIRA, M. F. L.; CRUVINEL, P. E.; MACEDO, Á. A novel model for combining projection and image filtering using Kalman and discrete wavelet transform in computerized tomography. In: *2008 11th IEEE International Conference on Computational Science and Engineering*. Institute of Electrical & Electronics Engineers (IEEE), 2008. Disponível em: <<http://dx.doi.org/10.1109/CSE.2008.37>>. Citado na página 32.

LAJARA, T. T.; BRINKHUES, R.; MAÇADA, A. C. G. Investimentos em TI e sua influência no desempenho organizacional: através da governança da informação e capacidades de gestão da informação, moderado pelo big data. In: *IV Encontro de Administração da Informação*. Bento Gonçalves, RS: EnADI, 2013. Citado 2 vezes nas páginas 55 e 56.

LASSO, P. R. O.; VAZ., C. M. C. M. P.; BACCHI, O. O. S. Otimização de parâmetros para aquisição de imagens tomográficas de amostras de solo. In: *XXXII Congresso brasileiro de ciência do solo*. [S.l.: s.n.], 2009. Citado na página 40.

LEE, I. Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*, v. 60, n. 3, p. 293 – 303, 2017. ISSN 0007-6813. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0007681317300046>>. Citado 4 vezes nas páginas 27, 53, 58 e 60.

- LEE, J. H.; YAO, Y.; SHRESTHA, U.; GULLBERG, G. T.; SEO, Y. Handling big data in medical imaging: Iterative reconstruction with large-scale automated parallel computation. In: *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. Institute of Electrical & Electronics Engineers (IEEE), 2014. Disponível em: <<http://dx.doi.org/10.1109/NSSMIC.2014.7430758>>. Citado 2 vezes nas páginas 35 e 53.
- LIN, J.; DYER, C. Data-intensive text processing with MapReduce. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers LLC, v. 3, n. 1, p. 1–177, jan 2010. Citado na página 68.
- LUDENA, R. D. A.; AHRARY, A. A big data approach for a new ICT agriculture application development. In: *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. Institute of Electrical & Electronics Engineers (IEEE), 2013. Disponível em: <<http://dx.doi.org/10.1109/CyberC.2013.30>>. Citado na página 56.
- MA, J. Generalized sampling reconstruction from fourier measurements using compactly supported shearlets. *Applied and Computational Harmonic Analysis*, Elsevier BV, v. 42, n. 2, p. 294–318, mar 2017. Citado na página 30.
- MARK, A. B.; LANEY, D. The importance of big data: a definition. *Gartner*, 6 2012. Citado na página 56.
- MARQUESONE, R. *Big data: técnicas e tecnologias para extração de valor dos dados*. [S.l.: s.n.], 2017. Citado na página 34.
- MATTSON, T. G.; SANDERS, B. A.; MASSINGILL, B. L. *Patterns for Parallel Programming*. [S.l.]: Addison-Wesley Professional, 2004. ISBN 0321228111. Citado na página 66.
- MAYER-SCHONBERGER, V.; CUKIER, K. *Big Data: como extrair volume, variedade, velocidade e valor da avalanche de informação cotidiana*. 1. ed. Rio de Janeiro: Elsevier, 2013. Tradução Paulo Polzonoff Junior. Citado 3 vezes nas páginas 27, 52 e 53.
- MELL, P.; GRANCE, T. Special Publication 800-145, *The NIST definition of Cloud Computing*. 2011. Último acesso em 04/02/2019. Disponível em: <<https://csrc.nist.gov/publications/detail/sp/800-145/final>>. Citado na página 75.
- MELLO, R. G. S. *Utilização de big data analytics nos sistemas de medição de desempenho: estudos de caso*. Dissertação (Mestrado) — UFSCar, 2015. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/3784>>. Citado na página 34.
- MINATEL, E.; CRUVINEL, P. Three-dimensional reconstruction and visualization of tomographic images system using frequential techniques and wavelets. In: *Proceedings SIB-GRAPI'98. International Symposium on Computer Graphics, Image Processing, and Vision (Cat. No.98EX237)*. Institute of Electrical & Electronics Engineers (IEEE), 1998. p. 38–45. Disponível em: <<http://dx.doi.org/10.1109/sibgra.1998.722731>>. Citado na página 32.
- MODOLO, A. J.; FERNANDES, H. C.; NAIME, J. de M.; SCHAEFER, C. E. G.; SANTOS, N. T.; SILVEIRA, J. C. M. da. Avaliação do ambiente solo-semente por meio da tomografia computadorizada. *Revista Brasileira de Ciência do Solo*, FapUNIFESP (SciELO), v. 32, n. 2, p. 525–532, apr 2008. Citado na página 40.
- MORAES, R. M.; MARTÍNEZ, L. Editorial: Computational Intelligence Applications for Data Science. *Knowledge-Based Systems*, v. 87, 2015. Citado na página 60.

MORETTIN, P. A.; BUSSAB, W. O. *Estatística básica*. 9. ed. [S.l.: s.n.], 2017. ISBN 8547220224. Citado na página 96.

NAIME, J. ao M.; MATTOSO, L. H. C.; SILVA, W. T. L.; CRUVINEL, P. E.; NETO, L. M.; CRESTANA, S. *Conceitos e aplicações da instrumentação para o avanço da agricultura*. [S.l.: s.n.], 2014. Citado na página 39.

NAIME, J. M. *Projeto e construção de um Tomógrafo portátil para estudos de ciência do solo e plantas, em campo*. Dissertação (Mestrado) — USP, 1994. Citado na página 39.

NAIME, J. M. *Um novo método para estudos dinâmicos, in situ, da infiltração da água na região não-saturada do solo*. Tese (Doutorado) — USP, 2001. Citado na página 39.

NHS. *Diagnostic Imaging Dataset Annual Statistical Release 2014/15*. [S.l.], 2015. Disponível em: <<https://www.england.nhs.uk/statistics/statistical-work-areas/diagnostic-imaging-dataset/>>. Citado na página 39.

NIST. *NIST Big Data Interoperability Framework: volume 1, definitions. Final version 1*. Gaithersburg, MD, 2015. Citado 6 vezes nas páginas 55, 56, 57, 58, 59 e 60.

OPPENHEIM, A. V.; SCHAFER, R. W. *Digital Signal Processing*. PRENTICE HALL, 1975. ISBN 0132146355. Disponível em: <[http://www.ebook.de/de/product/3624721/alan\\_v\\_oppenheim\\_ronald\\_w\\_schafer\\_digital\\_signal\\_processing.html](http://www.ebook.de/de/product/3624721/alan_v_oppenheim_ronald_w_schafer_digital_signal_processing.html)>. Citado na página 94.

PALOMBO, L. *A microtomografia de raios X e a porosimetria por intrusão de mercúrio na determinação de porosidade e densidade de rochas reservatório*. Dissertação (Mestrado) — USP, 2016. Citado na página 40.

PASSONI, S.; PIRES, L. F.; HECK, R.; ROSA, J. A. Three Dimensional Characterization of Soil Macroporosity by x-Ray Microtomography. *Rev. Bras. Ciênc. Solo*, FapUNIFESP (SciELO), v. 39, n. 2, p. 448–457, 4 2015. Disponível em: <<http://dx.doi.org/10.1590/01000683rbcs20140360>>. Citado na página 40.

PEDROTTI, A.; PAULETTO, E. A.; CRESTANA, S.; CRUVINEL, P. E.; VAZ, C. M. P.; NAIME, J. de M.; SILVA, A. M. da. Tomografia computadorizada aplicada a estudos de um planossolo. *Pesq. agropec. bras.*, FapUNIFESP (SciELO), v. 38, n. 7, 7 2003. Disponível em: <<http://dx.doi.org/10.1590/S0100-204X2003000700005>>. Citado na página 40.

PENALOZA, E. A. G.; CRUVINEL, P. E.; OLIVEIRA, V. A.; COSTA, A. G. F. Database and knowledge base integration method to support decision-making related to quality of application and use of agricultural sprayers. *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, IEEE, 2017. Disponível em: <<http://dx.doi.org/10.1109/ICSC.2017.41>>. Citado na página 24.

PEREIRA, M.; CRUVINEL, P. A model for soil computed tomography based on volumetric reconstruction, wiener filtering and parallel processing. *Computers and Electronics in Agriculture*, Elsevier BV, v. 111, p. 151–163, 2 2015. Disponível em: <<http://dx.doi.org/10.1016/j.compag.2014.12.006>>. Citado 4 vezes nas páginas 8, 33, 48 e 66.

PEREIRA, M. F. L. *Um modelo de reconstrução tomográfica 3D para amostras agrícolas com filtragem de Wiener em processamento paralelo*. Tese (Doutorado) — USP, 2007. Disponível em: <<http://dx.doi.org/10.11606/T.76.2007.tde-17092007-205738>>. Citado na página 24.

PETROVIC, A. M.; SIEBERT, J. E.; RIEKE, P. E. Soil bulk density analysis in three dimensions by computed tomographic scanning1. *Soil Science Society of America Journal*, Soil Science Society of America, v. 46, n. 3, p. 445, 1982. Citado na página 39.

PIPATSRISAWAT, T.; GASIC, A.; FRANCHETTI, F.; PUESCHEL, M.; MOURA, J. Performance analysis of the filtered backprojection image reconstruction algorithms. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Institute of Electrical & Electronics Engineers (IEEE), 2005. Disponível em: <<http://dx.doi.org/10.1109/ICASSP.2005.1416263>>. Citado na página 25.

PIRES, L. F.; BACCHI, O. O. S. Mudanças na estrutura do solo avaliada com uso de tomografia computadorizada. *Pesquisa Agropecuária Brasileira*, FapUNIFESP (SciELO), v. 45, n. 4, p. 391–400, apr 2010. Citado na página 40.

PIRES, L. F.; BORGES, J. A.; BACCHI, O. O.; REICHARDT, K. Twenty-five years of computed tomography in soil physics: A literature review of the brazilian contribution. *Soil and Tillage Research*, Elsevier BV, v. 110, n. 2, p. 197–210, nov 2010. Citado 2 vezes nas páginas 39 e 83.

PLACIDI, G.; ALECCI, M.; SOTGIU, A. Theory of adaptive acquisition method for image reconstruction from projections and application to EPR imaging. *Journal of Magnetic Resonance, Series B*, Elsevier BV, v. 108, n. 1, p. 50–57, 7 1995. Disponível em: <<http://dx.doi.org/10.1006/jmrb.1995.1101>>. Citado na página 29.

PLACIDI, G.; ALECCI, M.; SOTGIU, A. Angular space-domain interpolation for filtered back projection applied to regular and adaptively measured projections. *Journal of Magnetic Resonance, Series B*, v. 110, n. 1, p. 75 – 79, 1996. ISSN 1064-1866. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1064186696900102>>. Citado na página 30.

QUEIRÓS, L. R.; JUNIOR, A. L.; CAMARGO NETO, J.; MASSRUHÁ, S. M. F. S.; INAMASU, R. Y.; SPERANZA, E. A.; EVANGELISTA, S. R. M. *Análise das possibilidades e tendências do uso das tecnologias da informação e comunicação em Agricultura de Precisão*. [S.l.]: Embrapa, 2014. 97-108 p. Citado 2 vezes nas páginas 26 e 52.

RANGAYYANI, R. M. *Biomedical Image Analysis (Biomedical Engineering)*. CRC Press, 2004. ISBN 0849396956. Disponível em: <<https://www.amazon.com/Biomedical-Image-Analysis-Engineering/dp/0849396956?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0849396956>>. Citado na página 41.

ROBERTY, N. C.; REIS, M. L.; CRISPIM, V. R. Método da máxima entropia para reconstrução de imagens. In: *II Simpósio Brasileiro de Computação Gráfica*. [S.l.: s.n.], 1989. Citado na página 29.

ROCHA, F. G. *Análise de escalabilidade de aplicações hadoop mapreduce por meio de simulação*. Dissertação (Mestrado) — UFSCar, 2013. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/534>>. Citado 3 vezes nas páginas 34, 67 e 70.

RUGGIERO, M. A. G.; LOPES, V. L. R. *Cálculo Numérico. Aspectos Teóricos e Computacionais*. Pearson, 1996. ISBN 9788534602044. Disponível em: <<https://www.amazon.com/Num%C3%A9rico-Aspectos-Te%C3%B3ricos-Computacionais-Portuguese/dp/8534602042?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=8534602042>>. Citado 2 vezes nas páginas 48 e 49.



SADALAGE, P. J.; FOWLER, M. *NoSQL essencial: um guia conciso para o mundo emergente da persistência poliglota*. [S.l.: s.n.], 2015. 220 p. ISBN 978-85-7522-338-3. Citado na página 59.

SAGIROGLU, S.; SINANC, D. Big data: A review. In: *2013 International Conference on Collaboration Technologies and Systems (CTS)*. Institute of Electrical & Electronics Engineers (IEEE), 2013. p. 42–47. Disponível em: <<http://dx.doi.org/10.1109/cts.2013.6567202>>. Citado 5 vezes nas páginas 8, 53, 55, 57 e 58.

SALINA, F.; MASCARENHAS, N.; CRUVINEL, P. A comparison of POCS algorithms for tomographic reconstruction under noise and limited view. In: *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*. Institute of Electrical & Electronics Engineers (IEEE), 2002. Disponível em: <<http://dx.doi.org/10.1109/SIBGRA.2002.1167164>>. Citado 3 vezes nas páginas 25, 31 e 40.

SALVADEO, D. H. P. *Filtragem de ruído em imagens tomográficas com baixa taxa de contagem utilizando uma abordagem bayseiana contextual*. Tese (Doutorado) — UFSCar, 2013. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/284>>. Citado 2 vezes nas páginas 25 e 40.

SCANNAVINO, F. A. *Tomógrafo de espalhamento Compton para estudos da física de solos agrícolas em ambiente de campo*. Tese (Doutorado) — USP, 2013. Citado 2 vezes nas páginas 39 e 41.

SCHUMAKER, L. *Spline functions : basic theory*. New York: Cambridge University Press, 2007. ISBN 9780511334047. Citado na página 49.

SERRANO, E.; BLAS, J. G.; CARRETERO, J. A comparative study of an x-ray tomography reconstruction algorithm in accelerated and cloud computing systems. *Concurrency and Computation: Practice and Experience*, Wiley-Blackwell, v. 27, n. 18, p. 5538–5556, 8 2015. Disponível em: <<http://dx.doi.org/10.1002/cpe.3599>>. Citado na página 34.

SHANNON, C. Communication in the presence of noise. *Proceedings of the IRE*, Institute of Electrical and Electronics Engineers (IEEE), v. 37, n. 1, p. 10–21, Jan 1949. ISSN 0096-8390. Disponível em: <<http://dx.doi.org/10.1109/JRPROC.1949.232969>>. Citado na página 97.

SHANNON, C. E. A mathematical theory of communication. *Bell System Technical Journal*, Institute of Electrical and Electronics Engineers (IEEE), v. 27, n. 3, p. 379–423, Jul 1948. ISSN 0005-8580. Disponível em: <<http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>>. Citado na página 97.

SHEPP, L. A.; LOGAN, B. F. The Fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, Institute of Electrical & Electronics Engineers (IEEE), v. 21, n. 3, p. 21–43, 6 1974. Disponível em: <<http://dx.doi.org/10.1109/TNS.1974.6499235>>. Citado na página 77.

SHTOK, J.; ZIBULEVSKY, M.; ELAD, M. Spatially-adaptive reconstruction in computed tomography based on statistical learning. 2010. Disponível em: <<http://arxiv.org/abs/1004.4373>>. Citado na página 32.

SHVACHKO, K.; KUANG, H.; RADIA, S.; CHANSLER, R. The hadoop distributed file system. In: *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. [S.l.]: IEEE, 2010. Citado na página 70.

SILVA, A. M. *Construção e uso de um tomógrafo com resolução micrométrica para aplicações em ciências do solo e ambi.* Tese (Doutorado) — USP, 1997. Citado na página 39.

SILVA, A. M.; NAIME, J. ao M.; VAZ, C. M. P.; CRESTANA, S.; PAULO. Instrumentação avançada em ciência do solo. In: \_\_\_\_\_. [S.l.]: Embrapa Instrumentação Agropecuária, 2007. cap. Tomografia computadorizada de raios X e gama para investigação não invasiva do solo. ISBN 85-86463-14-0. Citado na página 39.

SILVA, F. A. B. da; SENGER, H. Improving scalability of bag-of-tasks applications running on master-slave platforms. *Parallel Computing*, Elsevier BV, v. 35, n. 2, p. 57–71, feb 2009. Citado na página 68.

SINGH, S.; SINGH, N. Big data analytics. In: *International conference on Communication, information & computing technology*. [S.l.: s.n.], 2012. Citado na página 56.

SLAVAKIS, K.; GIANNAKIS, G. B.; MATEOS, G. Modeling and Optimization for Big Data Analytics. *IEEE Signal Processing Magazine*, p. 18–31, 2014. Citado na página 60.

TAINA, I. A.; HECK, J.; ELLIOT, T. R. Application of x-ray computed tomography to soil science: a literature review. *Canadia Journal Soil Science*, 2007. Citado 2 vezes nas páginas 39 e 40.

TETZNER, G. C. *Aplicação da tomografia computadorizada industrial na análise de rochas.* Tese (Doutorado) — USP, 2008. Citado na página 40.

TSENG, C. L. *Tomografia computadorizada de raios-X aplicada à análise da qualidade ambiental de solo no entorno da Usina Hidrelétrica de Ilha Solteira - SP.* Tese (Doutorado) — USP, 2013. Citado na página 40.

UNSER, M. A. Ten good reasons for using spline wavelets. In: ALDROUBI, A.; LAINE, A. F.; UNSER, M. A. (Ed.). *Wavelet Applications in Signal and Image Processing V*. [S.l.]: SPIE, 1997. Citado na página 50.

VAZ, C. M. P.; CRESTANA, S.; NAIME, J. de M.; CRUVINEL, P. E. Tomografia computadorizada de raios x ou gamma. 2014. Citado na página 24.

VELTE, T.; VELTE, A.; ELSENPETER, R. C. *Cloud Computing, A Practical Approach*. [S.l.]: McGraw-Hill Education, 2009. ISBN 9780071626941. Citado na página 75.

VERDU, S. Fifty years of shannon theory. *IEEE Transactions on Information Theory*, Institute of Electrical and Electronics Engineers (IEEE), v. 44, n. 6, p. 2057–2078, 1998. ISSN 0018-9448. Disponível em: <<http://dx.doi.org/10.1109/18.720531>>. Citado na página 97.

VIEIRA, S. R. Geoestatística em estudo de variabilidade espacial do solo. In: \_\_\_\_\_. *Tópicos em ciência do solo*. [S.l.: s.n.], 2000. v. 1, p. 1–54. Citado na página 24.

WANG, G. A perspective on deep imaging. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 4, p. 8914–8924, 2016. Citado 3 vezes nas páginas 27, 35 e 36.

WHITE, T. *Hadoop: The Definitive Guide*. O'Reilly Media, 2012. ISBN 9781449311520. Disponível em: <<https://www.amazon.com/Hadoop-Definitive-Guide-Tom-White/dp/1449311520?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1449311520>>. Citado 5 vezes nas páginas 67, 69, 70, 71 e 72.

- XEXÉO, G. Big data: computação para uma sociedade conectada e digitalizada. In: *Revista Ciência Hoje*. [S.l.: s.n.], 2013. v. 306, p. 18–23. Citado 3 vezes nas páginas 53, 54 e 55.
- YANG, Q.; KALRA, M. K.; PADOLE, A.; LI, J.; HILLIARD, E.; LAI, R.; WANG, G. Big data from CT scanning. *SM Biomed Imaging Data Papers*, 2015. Citado 2 vezes nas páginas 34 e 39.
- ZAHARIA, M.; FRANKLIN, M. J.; GHODSI, A.; GONZALEZ, J.; SHENKER, S.; STOICA, I.; XIN, R. S.; WENDELL, P.; DAS, T.; ARMBRUST, M.; DAVE, A.; MENG, X.; ROSEN, J.; VENKATARAMAN, S. Apache spark. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 59, n. 11, p. 56–65, oct 2016. Citado na página 73.
- ZHANG, H.; LI, L.; QIAO, K.; WANG, L.; YAN, B.; LI, L.; HU, G. Image Prediction for Limited-angle Tomography via Deep Learning with Convolutional Neural Network. *ArXiv e-prints*, jul. 2016. Citado na página 27.
- ZHANG, J.; HUANG, M. L. 5Ws model for big data analysis and visualization. In: *2013 IEEE 16th International Conference on Computational Science and Engineering*. Institute of Electrical & Electronics Engineers (IEEE), 2013. p. 1021–1028. Disponível em: <<http://dx.doi.org/10.1109/CSE.2013.149>>. Citado 4 vezes nas páginas 8, 55, 57 e 58.
- ZHAO, J.; FU, Y.; TAN, Y.; CAO, F. A reduction algorithm for the big data in 3D surface reconstruction. In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. Institute of Electrical & Electronics Engineers (IEEE), 2013. Disponível em: <<http://dx.doi.org/10.1109/SMC.2013.824>>. Citado na página 27.
- ZUBELDIA, E. H.; OZELIM, L. C. S. M.; TSENG, C. L.; CRESTANA, S.; CAVALCANTE, A. L. B. Uso da tomografia computadorizada de raios x para a geração de meios porosos artificiais. In: *Simpósio Nacional de Instrumentação Agropecuária*. [S.l.: s.n.], 2014. p. 577–580. Citado na página 40.



# **Apêndices**

# APÊNDICE A

## ARQUIVOS DE CONFIGURAÇÃO

---

---

Código 4 – Arquivo de configuração MRJob

```
1 runners:
2   emr:
3     # etapa 1 – reservar maquinas
4     label: AgReconSparkCluster
5     max_mins_idle: 30
6     instance_groups:
7     - InstanceRole: MASTER
8       Name: AgReconAwsMaster
9       InstanceCount: 1
10      InstanceType: m5.xlarge
11     - InstanceRole: CORE
12       Name: AgReconAwsCore
13       InstanceCount: 3
14       InstanceType: m5.xlarge
15     region: sa-east-1
16     cloud_log_dir: s3://agrecon/tmp/logs/
17     cloud_tmp_dir: s3://agrecon/tmp/
18
19     # etapa 2 – configurar acesso e comunicacao
20     aws_access_key_id: <valor da chave>
21     aws_secret_access_key: <valor da chave>
22     ec2_key_pair: agrecon-key-pair
23     ec2_key_pair_file: /home/doutorado/.ssh/agrecon-key-pair.pem
24     ssh_tunnel: true
25
26     # etapa 3 – instalar apache spark, yarn
27     release_label: emr-5.24.0
28     bootstrap_spark: True
29
30     # etapa 4 – configurar spark, yarn
31     emr_configurations:
32     - Classification: spark-defaults
33       Properties:
34         # driver
35         spark.driver.memory: 27g
36         spark.driver.cores: 5
37         spark.driver.memoryOverhead: 4g
38         spark.driver.maxResultSize: 0
39
40     # executor
```

```
41     spark.executor.memory: 27g
42     spark.executor.cores: 5
43     spark.executor.instances: 4
44     spark.executor.memoryOverhead: 4g
45     spark.executor.heartbeatInterval: 60s
46
47     spark.rdd.compress: true
48     spark.shuffle.compress: true
49     spark.shuffle.spill.compress: true
50     spark.default.parallelism: 40
51     spark.sql.shuffle.partitions: 40
52     spark.network.timeout: 800s
53
54     spark.dynamicAllocation.enabled: false
55     spark.memory.fraction: 0.9
56     spark.memory.storageFraction: 0.1
57     spark.yarn.scheduler.reporterThread.maxFailures: 5
58     spark.storage.level: MEMORY_AND_DISK_SER
59
60 - Classification: yarn-site
61   Properties:
62     yarn.nodemanager.resource.memory-mb: 32768
63     yarn.scheduler.maximum-allocation-mb: 32768
64     yarn.nodemanager.vmem-check-enabled: false
65     yarn.nodemanager.pmem-check-enabled: false
66
67 - Classification: spark
68   Properties:
69     maximizeResourceAllocation: false
70
71 - Classification: mapred-site
72   Properties:
73     mapreduce.map.output.compress: true
74
75 # etapa 5 - instalar bibliotecas adicionais
76 bootstrap:
77 - sudo pip-3.6 install scikit-image scikit-learn pandas boto3 imageio tiffio pyyaml
78   pyspark
79 - echo 'export PYSPARK_PYTHON=/usr/bin/python3' >> /home/hadoop/.bashrc && source /home/
80   hadoop/.bashrc
81
82 # etapa 6 - configuracoes adicionais
83 python_bin: python3
84 py_files: /home/doutorado/Documentos/030-doutorado/65_metodo/src/ctrecon.zip
85 upload_dirs: /home/doutorado/Documentos/030-doutorado/65_metodo/src/ctrecon/
86 setup:
87 - export PYTHONPATH=$PYTHONPATH:./ctrecon
```

## Código 5 – Arquivo de configuração do método desenvolvido

```
1 # amostra
2 nome: 'Phanton'
3 passo: 0.2
4 primeira_secao: 259
5 ultima_secao: 269
6 dimensao: (1000, 976)
7
8 # selecao
9 taxa: 1.0
10 selecao_por_classe: True
11
12 # recon2d
13 output_size: 1000
14 intervalo_reconstrucao: (0.0, 195.2)
15
16 # divisao (da imagem reconstruida)
17 tam_janela: 100
18 njanelas_linhas: 10
19 njanelas_colunas: 10
20
21 # recon3d
22 interpolacao_bspline: 2.0
```

# APÊNDICE B

## SISTEMA DE SUPORTE À DECISÃO BASEADO EM BIG DATA E CIÊNCIA DOS DADOS

---

---

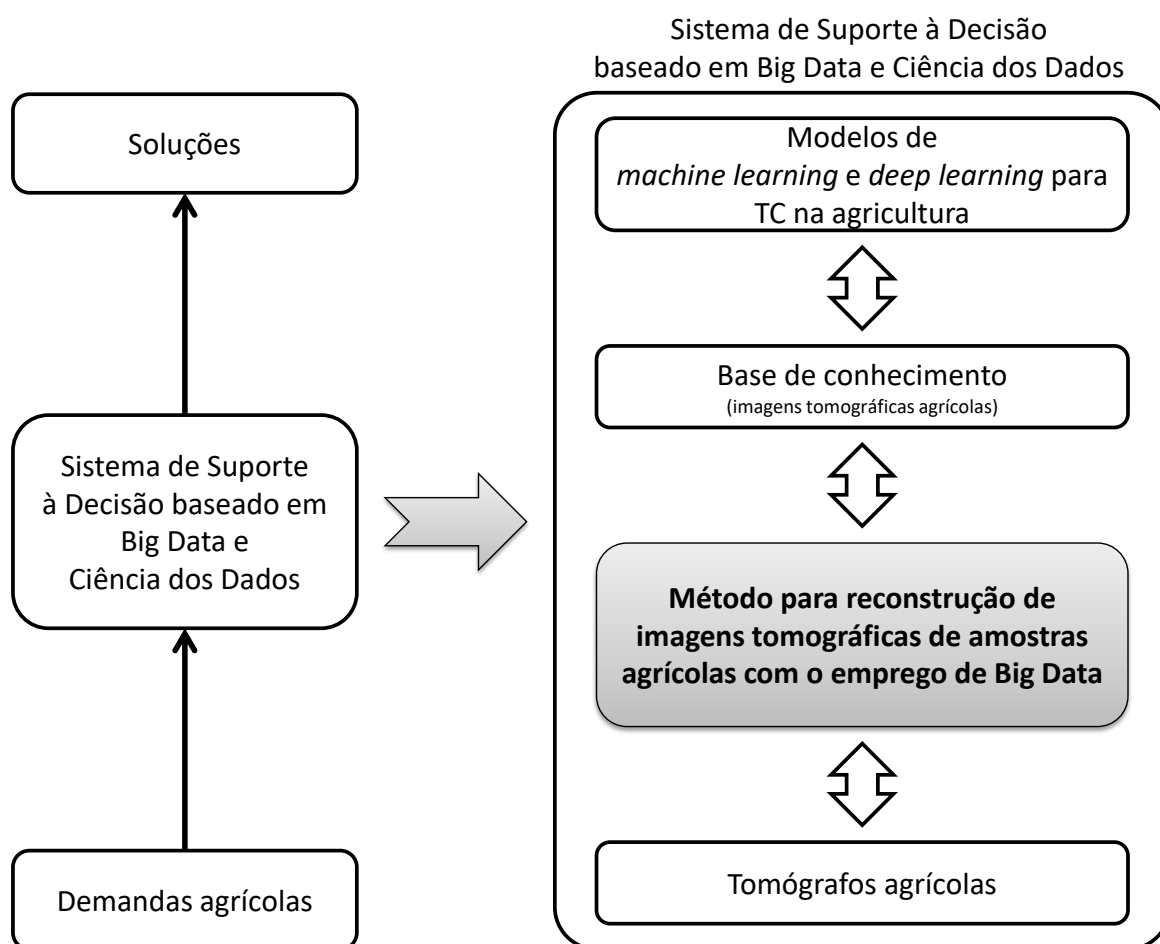


Figura 57 – À esquerda é ilustrada uma visão geral na qual o Sistema de Suporte à Decisão baseado em Big Data e Ciência dos Dados traduz as demandas agrícolas em propostas de soluções que dão suporte à tomada de decisão. À direita tal sistema é detalhado em quatro principais camadas.

# APÊNDICE C

## PUBLICAÇÕES

---

---

A seguir são relacionados trabalhos científicos os quais foram elaborados, submetidos e apresentados em congressos nacionais e internacionais durante o doutorado.

- 2019

- **Parallel Computational Structure and Semantics for Soil Quality Analysis based on LoRa and Apache Spark.**

- \* Maurício F. L. Pereira, Paulo E. Cruvinel, Gabriel M. Alves, José M. G. Beraldo
    - \* The Fifth International Workshop on Semantic Computing and Knowledge Creation based on Needs Engineering on 14th IEEE International Conference on Semantic Computing.
    - \* *Artigo submetido e aprovado.*

- **Organização de uma estrutura de processamento paralelo baseado em Apache Spark para gerenciamento de risco agrícola.**

- \* Maurício F. L. Pereira, Gabriel M. Alves, José M. G. Beraldo, Paulo E. Cruvinel
    - \* Simpósio Nacional de Instrumentação Agropecuária (Siagro 2019)

- **Método baseado em Grafos para segmentação de sementes oleaginosas em imagens tomográficas de alta resolução.**

- \* André R. Brito, Gabriel M. Alves, Paulo E. Cruvinel
    - \* Simpósio Nacional de Instrumentação Agropecuária (Siagro 2019)

- **Big Data infrastructure for agricultural tomographic images reconstruction.**

- \* Gabriel M. Alves, Paulo E. Cruvinel
    - \* 3º Encontro Paulista dos Pós-Graduandos em Computação

- 2018

- **Big Data infrastructure for agricultural tomographic images reconstruction.**

- \* Gabriel M. Alves, Paulo E. Cruvinel

- \* 2º Encontro Paulista dos Pós-Graduandos em Computação
- **Big Data infrastructure for agricultural tomographic images reconstruction.**
  - \* Gabriel M. Alves, Paulo E. Cruvinel
  - \* 12th IEEE International Conference on Semantic Computing (ICSC2018)
  - \* Qualis B1
- 2017
  - **Big Data environment for agricultural soil analysis from CT digital images.**
    - \* Gabriel M. Alves, Paulo E. Cruvinel
    - \* 1º Encontro Paulista dos Pós-Graduandos em Computação
  - **A new approach for plant phenotyping and image segmentation based on contextual information.**
    - \* Gabriel M. Alves, Paulo E. Cruvinel, Gustavo B. Souza, Aparecido N. Marana, Alexandre L. M. Levada
    - \* II Latin-American Conference on Plant Phenotyping and Phenomics for Plant Breeding
- 2016
  - **Desenvolvimento de novo modelo de reconstrução tomográfica 3D de amostras agrícolas com emprego de Big Data.**
    - \* Gabriel M. Alves, Paulo E. Cruvinel
    - \* I Workshop do PPGCC/UFSCar
    - \* *Eleito entre os cinco melhores trabalhos.*
  - **A graph-based approach for contextual image segmentation**
    - \* Gustavo B. Souza, Gabriel M. Alves, Alexandre L. M. Levada, Paulo E. Cruvinel, Aparecido N. Marana
    - \* XXIX Conference on Graphics, Patterns and Images (SIBGRAP' 16).
    - \* Qualis B1
  - **Customized computer vision and sensor system for colony recognition and live bacteria counting in agriculture.**
    - \* Gabriel M. Alves, Paulo E. Cruvinel
    - \* Journal of Sensors & Transducers.
    - \* GIF 0.987
  - **Using customized computer vision and charge-coupled device (CCD) sensor for the recognition of colonies formation and counting of alive bacteria in agricultural industry.**



- \* Gabriel M. Alves, Paulo E. Cruvinel
- \* The First International Conference on Advances in Sensors, Actuators, Metering and Sensing
- \* *Best paper awards.*
- **Big Data environment for agricultural soil analysis from CT digital images**
  - \* Gabriel M. Alves, Paulo E. Cruvinel
  - \* 10th IEEE International Conference on Semantic Computing (ICSC2016).
  - \* Qualis B1
- 2015
  - **Estruturação de um algoritmo baseado em Big Data e técnica de tomografia para análise da amostra de solo agrícola**
    - \* Gabriel M. Alves, Paulo E. Cruvinel
    - \* XLIV Congresso Brasileiro de Engenharia Agrícola (CONBEA)