

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DAVI CARLOS DA SILVA

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXÍLIO NA AVALIAÇÃO DE
MAPAS T2 DE EXAMES DE IMAGEM POR RESSONÂNCIA MAGNÉTICA**

CURITIBA

2022

DAVI CARLOS DA SILVA

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXÍLIO NA AVALIAÇÃO DE
MAPAS T2 DE EXAMES DE IMAGEM POR RESSONÂNCIA MAGNÉTICA**

**Development of a software to aid in the evaluation of magnetic resonance
T2 maps imaging exams**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Doutor Gustavo Benvenuto Borba

Coorientador: Doutor Tiago da Silva Jornada

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

DAVI CARLOS DA SILVA

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXÍLIO NA AVALIAÇÃO DE
MAPAS T2 DE EXAMES DE IMAGEM POR RESSONÂNCIA MAGNÉTICA**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Data de aprovação: 20/abril/2022

Gustavo Benvenuti Borba
Doutorado
Universidade Tecnológica Federal do Paraná - UTFPR

Gilson Yukio Sato
Doutorado
Universidade Tecnológica Federal do Paraná

Daniel Rossato de Oliveira
Título (Doutorado)
Universidade Tecnológica Federal do Paraná

Tiago da Silva Jornada
Doutorado
Hospital das Clínicas da Universidade Federal de Pernambuco

CURITIBA

2022

Dedico esse trabalho aos meus pais e irmãos,
além da minha namorada e melhor amiga
Sthefania Darling Gonçalves de Araújo

AGRADECIMENTOS

Agradeço a minha família que sempre esteve do meu lado, independente das minhas escolhas.

Aos meus amigos que sempre me apoiaram a concluir meus projetos e seguir com meus sonhos.

Ao Dr e Professor Gustavo Benvenuti Borba pelos anos de amizade, pela orientação e pelo apoio.

Ao Dr e professor Tiago da Silva Jornada, pela inspiração do projeto e pela dedicação em orientar o projeto.

Aos demais professores pela sua paciência e cuidado nos meus anos de faculdade.

À minha namorada Sthefania Darling Gonçalves de Araújo, pelo apoio incondicional.

RESUMO

O objetivo deste trabalho foi desenvolver uma ferramenta *open source* Python para a geração, visualização e edição de mapas T2 de ressonância magnética. Essa ferramenta, além da geração do mapa T2 a partir de uma lista de arquivos Dicom, permite ao usuário analisar regiões de interesse, visualizar as imagens em diferentes mapas de cor, e exportar resultados obtidos em forma de um documento. Por fim, este documento apresenta as principais etapas de desenvolvimento do projeto, desde o estudo da teoria envolvida no processo de ressonância magnética, passando pelas tecnologias e algoritmos utilizados até os testes e resultados finais do projeto.

Palavras-chave: ressonância magnética; python; processamento de imagens.

ABSTRACT

The objective of this work was to develop a Python open source tool for generating, visualizing and editing T2 MRI maps. This tool, in addition to generating the T2 map from a list of Dicom files, allows the user to analyze regions of interest, visualize the images in different color maps, and export the results obtained as a document. Lastly, this document presents the main stages of project development, from the study of the theory involved in the magnetic resonance process, through the technologies and algorithms used, to the tests and final results of the project.

Keywords: magnetic resonance; python; image processing.

LISTA DE FIGURAS

Figura 1 – Alinhamento dos átomos sobre influência de B_0	17
Figura 2 – Movimento de precessão do átomo de hidrogênio.	18
Figura 3 – Esquema para obtenção do sinal de MR.	19
Figura 4 – Gráfico de recuperação T1.	20
Figura 5 – Gráfico de decaimento T2.	21
Figura 6 – Pulso de sincronização - analogia da Corrida.	22
Figura 7 – Sequência de <i>multiple spin echo</i> para obtenção dos valores T2.	23
Figura 8 – Estrutura da interface.	27
Figura 9 – Aplicação de pincel sobre o mapa	28
Figura 10 – Aplicação de etiqueta sobre o mapa	29
Figura 11 – Aplicação de máscaras para processamento da ROI	30
Figura 12 – Exemplo de diferentes valores limites para cada um dos mapas de cor	31
Figura 13 – Fluxo geral do algoritmo de geração do mapa T2.	32
Figura 14 – Fluxo da importação e extração da informação dos arquivos Dicom.	32
Figura 15 – Fluxo do pré-tratamento das imagens.	33
Figura 16 – Aplicação do método medfilt2d	33
Figura 17 – Fluxo do processamento dos tempos de eco.	35
Figura 18 – Fluxo do processamento final para geração do mapa T2.	36
Figura 19 – Fluxo para análise dos erros na geração do mapa T2.	37
Figura 20 – Tela completa da interface.	38
Figura 21 – Menu de edição	39
Figura 22 – Visualização das imagens Dicom e mapa T2.	40
Figura 23 – Histórico de edições.	41
Figura 24 – Informações da ROI.	42
Figura 25 – Documento exportado pelo software.	43
Figura 26 – Mapa T2 gerado - joelho	44
Figura 27 – Mapa T2 gerado - fígado	44
Figura 28 – Resultado da FDA para o mapa T2 do exame do joelho.	45
Figura 29 – Comparativo entre mapas T2 do exame do joelho.	45
Figura 30 – Resultado da FDA para o EQM do mapa T2 do exame do joelho.	46

Figura 31 – Comparativo entre matrizes de EQM do mapa T2 do exame do joelho.	46
Figura 32 – Resultado da FDA para o mapa T2 do exame do figado.	47
Figura 33 – Comparativo entre mapas T2 do exame do figado.	47
Figura 34 – Resultado da FDA para o EQM do mapa T2 do exame do figado.	48
Figura 35 – Comparativo entre matrizes de EQM do mapa T2 do exame de figado.	48

LISTA DE TABELAS

Tabela 1 – Resultado da análise de erros	49
Tabela 2 – Resultado dos testes de performance	49

LISTA DE ABREVIATURAS E SIGLAS

Siglas

DICOM	<i>Digital Imaging and Communications in Medicine</i>
RF	Radiofrequência
RM	Ressonância Magnética
PDF	Formato de Documento Portátil, do inglês <i>Portable Document Format</i>
ROI	<i>Region of interest</i>
TE	Tempo de Eco
TR	Tempo de Repetição
T1	Tempo de recuperação longitudinal
T2	Tempo de decaimento transversal

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação e justificativa	14
1.2	Objetivos	14
1.3	Organização do documento	14
2	REFERENCIAL TEÓRICO	16
2.1	Alinhamento e precessão	16
2.2	Equação de Larmor e ressonância	17
2.3	Relaxamento	18
2.3.1	Recuperação T1	18
2.3.2	Decaimento T2	20
2.4	Sequência de pulsos	22
2.5	Mecanismos de contraste	23
2.6	Aplicações de análise T2	24
2.7	Tecnologias utilizadas	24
2.7.1	Tkinter	25
2.7.2	Numpy	25
2.7.3	OpenCV	25
2.7.4	Matplotlib	25
2.7.5	PyFPDF	26
2.7.6	Pydicom	26
2.7.7	MATLAB	26
3	METODOLOGIA	27
3.1	Interface	27
3.2	Processamento de imagem	27
3.3	Documentação	30
3.4	Algoritmo para processamento do mapa T2	31
3.4.1	Importação e pré-tratamento	31
3.4.2	Tratamento dos tempos de eco	34
3.4.3	Geração do mapa T2 e Análise de erros	35
3.4.4	Testes comparativos	36

4	ANÁLISE E DISCUSSÃO DOS RESULTADOS	38
4.1	Interface	38
4.1.1	Menu de edição	39
4.1.2	Visualizador de imagens	40
4.1.3	Histórico de edições	41
4.1.4	Informações da ROI	41
4.2	Geração de relatório	42
4.3	Mapas T2 gerados	42
4.4	Performance	49
5	CONCLUSÃO	50
5.1	Dificuldades Encontradas	50
5.2	Trabalhos Futuros	50
	REFERÊNCIAS	52

1 INTRODUÇÃO

Ressonância magnética nuclear, ou imagem por ressonância magnética (IRM), é uma das técnicas mais versáteis de diagnóstico por imagem, pois é capaz de fornecer ao especialista imagens de alta definição, o que facilita a distinção de pequenos detalhes. Diversos métodos de IRM são usados atualmente, sendo os mais comuns os mapas T1 e T2.

A obtenção dos valores de T2 para um dado tecido não é o suficiente para diagnosticar doenças, pois sua função é a de apresentar a composição da área avaliada. No entanto, as informações fornecidas pela RM, quando cruzadas com outros resultados de exames clínicos e comparadas com valores de pacientes sabidamente saudáveis, podem ser cruciais para identificar doenças, localizá-las na região examinada e avaliar o comportamento das mesmas.

Mapas T1 são produzidos a partir da análise da constante de tempo T1. Essa constante de tempo diz respeito à duração do processo de recuperação de cerca de 63% da magnetização longitudinal do tecido após a aplicação do pulso de radiofrequência (WESTBROOK; ROTH; TALBOT, 2011). Para geração de imagens T1, utiliza-se de um contraste gerado pela diferença de tempos de recuperação T1 entre as diversas estruturas (por exemplo, gordura e água), presentes na região analisada.

Imagens T2 são produzidas aplicando a mesma lógica de contraste entre constantes de tempo, porém, nesse caso, a constante T2 diz respeito ao tempo de decaimento magnético transversal no tecido (WESTBROOK; ROTH; TALBOT, 2011). O decaimento T2, no entanto, possui outros fatores que influenciam no seu valor, sendo um deles as inhomogeneidades presentes no campo magnético, além da influência da precessão fora de fase dos átomos de hidrogênio. Esses fatores acabam por gerar um decaimento não ideal, convencionalmente chamado de constante T2*.

A fim de nulificar as inhomogeneidades, a obtenção do valor de T2 é feita utilizando uma série de pulsos. Aplica-se, então, a técnica *Multiple Spin Echo* (MSE), na qual um pulso de 90° em relação ao campo magnético é emitido, seguido de uma série de pulsos de 180°. O objetivo com essa série de pulsos é de compensar a diferença de fase no campo transversal, obtendo, assim, o valor T2 ao invés do valor T2*.

Os resultados obtidos através desse procedimento são armazenados, pulso a pulso, em arquivos Dicom, disponibilizados pelo próprio equipamento que registra os valores. Dentre os diversos parâmetros presentes no arquivo, pode-se extrair a matriz de valores obtidos no procedimento, que podem ser visualizados como uma imagem. A partir desses arquivos é possível obter o mapa T2, utilizando de cálculos envolvendo as matrizes e os valores de tempo de eco (TE).

Como o cálculo para obter os valores do mapa T2 deve ser feito ponto a ponto, torna-se necessário a utilização de um software capaz de processar as saídas do equipamento de ressonância (matrizes de sinal pulso a pulso) e, com o conjunto total de pulsos, obter uma matriz que pode ser convertida em imagem.

Diante dessa situação, propõe-se utilizar os conceitos de processamento digital de imagens para o desenvolvimento de um software aberto na linguagem *open source* Python. O software proposto será utilizado tanto para obtenção quanto para análises posteriores do mapa T2, através de ferramentas de edição e segmentação manual.

1.1 Motivação e justificativa

A ausência de um software com versão em língua portuguesa, desvinculado do equipamento utilizado no exame e de baixo custo (valores de mercado de aproximadamente 40.000\$) acaba limitando o potencial de uso de mapas T2 em pesquisa e no diagnóstico de doenças, tanto para estudantes da área quanto para profissionais e instituições que não possuem meios de realizar o alto investimento necessário para aquisição e atualização constante dos softwares presentes no mercado.

1.2 Objetivos

Desenvolver um software em Python para geração e edição de mapas T2 a partir de imagens de ressonância magnética.

Como objetivos específicos, pode-se citar:

- Desenvolver o algoritmo para construção da imagem utilizando mapas T2 em Python;
- Adicionar a funcionalidade de aplicação do mapa T2 sobre uma região de interesse definida pelo usuário (segmentação manual);
- Desenvolver a interface gráfica do usuário;
- Adicionar ferramentas com o objetivo de auxiliar na análise do exame, tais como:
 - Destacar regiões de interesse usando pincel;
 - Diferente níveis de zoom;
 - Criar anotações em pontos de interesse;
 - Apresentar valores de erro e desvio padrão sobre uma região de interesse.

1.3 Organização do documento

Este documento tem por função registrar os detalhes do projeto de conclusão de curso de engenharia eletrônica e está organizado conforme descrito a seguir.

No segundo capítulo serão introduzidos alguns conceitos fundamentais de ressonância magnética, partindo desde os princípios físicos básicos do comportamento do átomo de hidrogênio, passando pela teoria relativa aos tempos T1 e T2 e pelo procedimento para obtenção dos

respectivos valores, até concluir com as aplicações da análise de mapas T2 e as tecnologias utilizadas nesse projeto. No terceiro capítulo serão tratadas todas as etapas de desenvolvimento do projeto, descrevendo de forma detalhada a implementação de cada um dos módulos que constituem o software. No quarto capítulo serão apresentados e discutidos os testes e resultados obtidos com o software, utilizando exemplos reais de arquivos Dicom fornecidos e realizando também testes comparativos através dos resultados coletados com o uso do script em MATLAB desenvolvido na tese de doutorado do professor Doutor Tiago da Silva Jornada (JORNADA, 2017). Por fim, será feita uma conclusão analisando o software e os resultados obtidos como um todo, incluindo as dificuldades encontradas durante o desenvolvimento do projeto e planos para o futuro.

2 REFERENCIAL TEÓRICO

Este capítulo tem por objetivo apresentar o referencial teórico utilizado, apresentando conceitos da literatura para princípios relacionados ao processo de aquisição e processamento de imagens de ressonância magnética. Dentre esses princípios estão: precessão, ressonância, tempos de relaxamento e técnicas de obtenção de sinais, mecanismos de contraste nas imagens e cálculos necessários para a geração de imagens T2. Além dos conceitos relativos à ressonância magnética, este capítulo também contém detalhes sobre as tecnologias utilizadas para o desenvolvimento do software, apresentando as principais bibliotecas em Python referenciadas no código.

2.1 Alinhamento e precessão

A geração de imagens de ressonância magnética é possível devido a influência de um campo magnético externo B_0 nos átomos de hidrogênio presentes no corpo humano. Essa influência se dá de duas maneiras: ordenando os átomos de hidrogênio, alinhando-os à sua própria direção, e incitando os átomos a um novo movimento, conhecido por precessão.

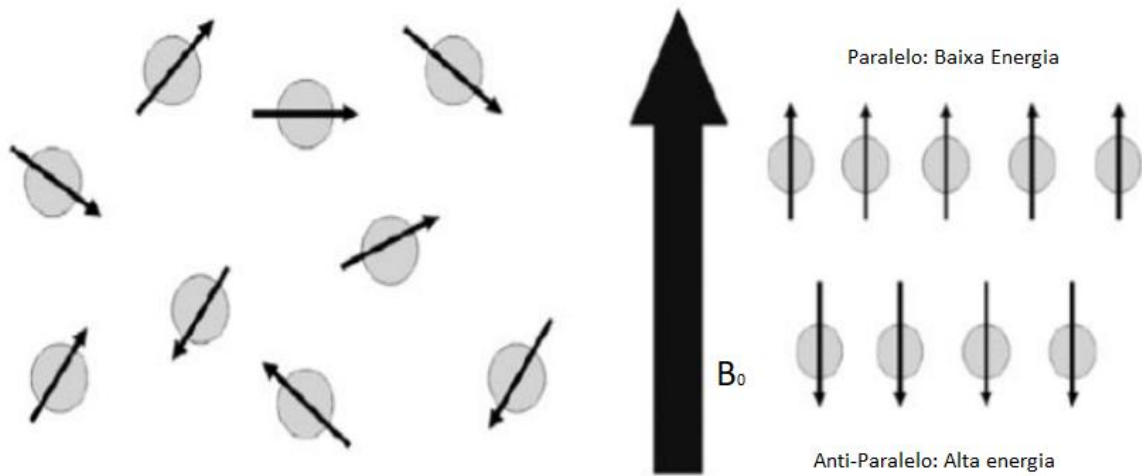
Os átomos de hidrogênio, mesmo sem influência de um campo magnético externo, ainda possuem um campo magnético devido ao seu movimento de rotação em torno do próprio eixo (spin). Esse campo atua como um dipolo, ou ímã, com um polo norte e sul, dando ao átomo um momento magnético com direção e sentido (HAGE; IWASAKI, 2009).

Quando influenciados por um campo externo, esses momentos magnéticos, antes apontando em direções aleatórias, tendem a se alinhar a essa nova influência externa. Porém, no caso do hidrogênio, esse alinhamento pode resultar em dois estados de energia distintos para os átomos: o de baixa energia, que é mais frequente e está alinhado paralelamente ao campo, e o de alta energia, que está alinhado anti-paralelamente ao campo (WESTBROOK; ROTH; TALBOT, 2011).

Como consequência desta distinção de estados, dois grupos de átomos se formam (Figura 1), sendo sua discrepância em números diretamente proporcional à intensidade do campo magnético que influenciou a sua geração. O somatório dos momentos magnéticos dos dois grupos acabam por gerar um vetor magnético resultante, de mesmo sentido do campo magnético externo.

O campo magnético externo influencia também o movimento do hidrogênio. O dipolo, além de ter sua rotação em torno do próprio eixo, tem esse mesmo eixo se movendo circularmente, de maneira semelhante a um pião. Esse novo movimento cíclico (consequentemente dotado de frequência e período) é conhecido como precessão (Figura 2) (GEUNS *et al.*, 1999).

Figura 1 – Alinhamento dos átomos sobre influência de B_0 .



Fonte: Hage e Iwasaki (2009).

2.2 Equação de Larmor e ressonância

Como B_0 gera a precessão nos átomos e como a mesma é dotada de frequência, é possível correlacionar os dois valores através da Equação 1, conhecida como equação de Larmor

$$F_0 = B_0 \cdot \frac{\lambda}{\pi \cdot 2}, \quad (1)$$

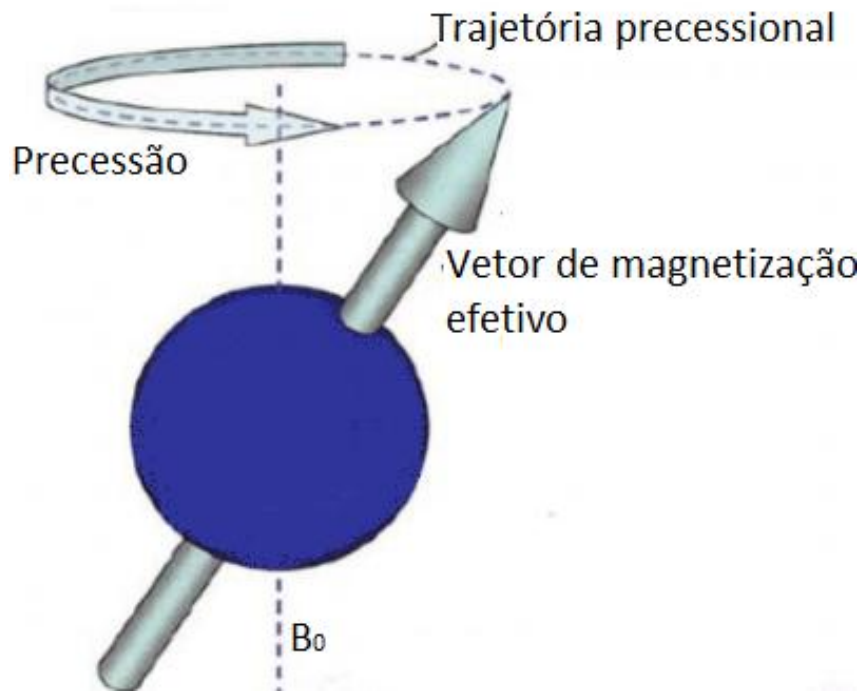
sendo F_0 a frequência de precessão e B_0 a intensidade do campo magnético. Pode-se assumir que a intensidade do campo é diretamente proporcional à frequência, sendo essa relação regida por λ , que é a razão giromagnética. A razão giromagnética (dada pela unidade Hz/T) é um valor constante para cada átomo, sendo a do hidrogênio 42.57 MHz/T.

Com essa oscilação dos átomos gerada pela precessão, é possível transferir energia para os mesmos utilizando de pulsos externos, através do fenômeno da ressonância. A ressonância se dá quando um corpo é afetado por uma perturbação oscilatória de mesma frequência que sua própria oscilação natural (WESTBROOK; ROTH; TALBOT, 2011). Como os átomos de hidrogênio apresentam essa oscilação criada pelo campo magnético, é possível excitá-los por meio de pulsos eletromagnéticos (usualmente na faixa de rádio frequência do espectro eletromagnético).

Com a excitação dos átomos, ou seja, com a transferência de energia do pulso de rádio frequência (RF) para o hidrogênio, dois fenômenos ocorrem: Um aumento da energia do conjunto de átomos e a alteração da direção do vetor resultante.

Primeiramente, vários elementos do grupo de baixa energia acabam saltando para o outro grupo, de alta energia. Essa alteração acaba por influenciar a direção do vetor de momento magnético resultante, criando um ângulo entre o vetor resultante e o campo magnético

Figura 2 – Movimento de precessão do átomo de hidrogênio.



Fonte: Westbrook, Roth e Talbot (2011).

(conhecido como plano longitudinal) e, conseqüentemente, uma componente a 90° do campo magnético (conhecido como plano transversal) (GEUNS *et al.*, 1999).

Além disso, os átomos, antes com suas precessões em diferentes fases, acabam por sincronizar seus movimentos. Segundo a lei de Faraday, a qual descreve que uma mudança no fluxo magnético causa uma força eletromotriz (KINSLER, 2020), esse momento magnético resultante acaba por induzir um sinal em bobinas receptoras presentes no equipamento (Figura 3), sendo esse o sinal mensurado para, posteriormente, serem formadas as imagens (VILLAFANA, 1988).

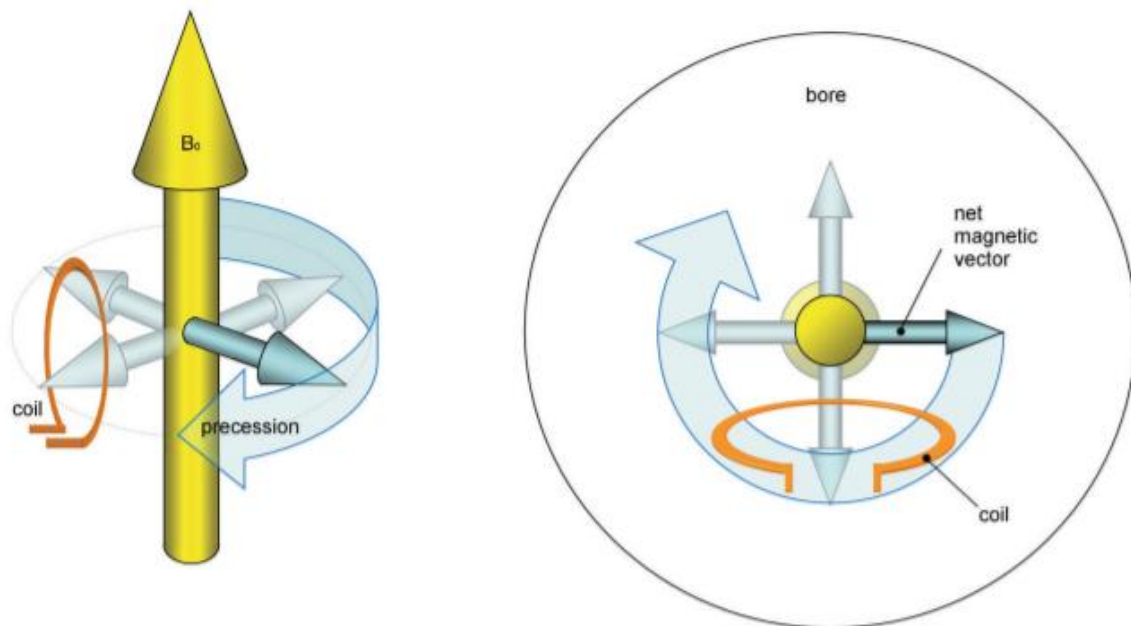
2.3 Relaxamento

A partir do instante em que o pulso RF acaba, o estado de ressonância acaba e os núcleos voltam a ser influenciados apenas pelo campo externo B_0 . Com isso, dois processos distintos se iniciam: o retorno do alinhamento dos momentos magnéticos e a dessincronização das precessões dos mesmos (JORNADA, 2017). Esses processos são denominados recuperação e decaimento.

2.3.1 Recuperação T1

"Recuperação T1, ou relaxamento T1, descreve a recuperação da magnetização longitudinal ao longo da direção do campo magnético estático B_0 logo após a aplicação de um pulso

Figura 3 – Esquema para obtenção do sinal de MR.



Fonte: Westbrook, Roth e Talbot (2011).

de RF"(JORNADA, 2017). Em outras palavras, o vetor resultante retorna a sua direção paralela ao campo magnético externo. Durante a influência do pulso de RF nos átomos de hidrogênio, como explicado anteriormente, ocorre um aumento da quantidade de átomos em estado anti-paralelo, ou seja, em estado de alta energia. A partir do momento em que essa influência sobre os átomos cessa, as interações térmicas dos átomos com suas vizinhanças causam uma perda energética, fazendo com que parte dos átomos de alta energia retornem para o grupo de baixa energia. Pelo fato dessa perda de energia ser para a rede de moléculas ao redor dos átomos, esse relaxamento é conhecido como spin-matriz (*spin-lattice*).

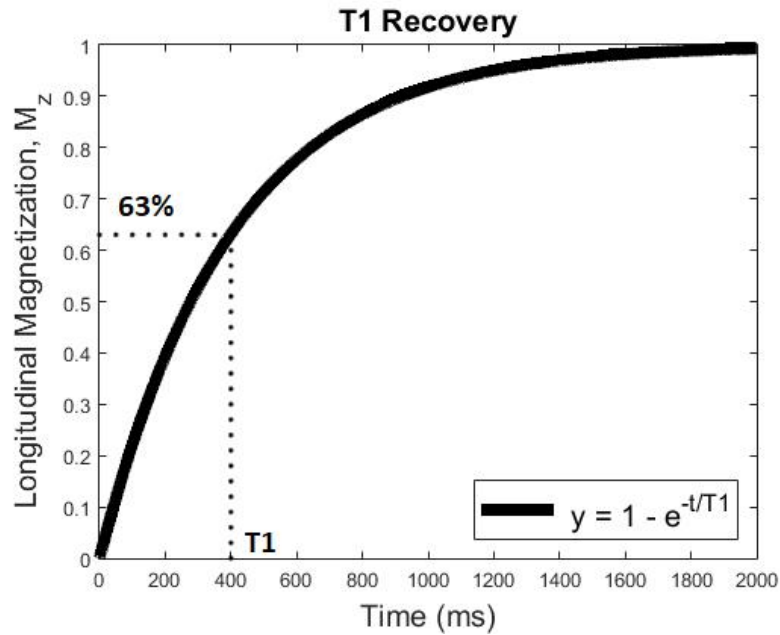
O retorno gradual do vetor resultante para a mesma direção do campo B_0 pode ser calculado através da Equação 2

$$M_Z(t) = M_0(1 - e^{-\frac{t}{T_1}}), \quad (2)$$

onde: t é tempo desde do término do pulso; T_1 é uma constante de tempo que representa o tempo necessário para o retorno de 63% da recuperação T_1 ; M_0 é o valor máximo da intensidade que o vetor resultante consegue atingir e e a constante de Euler.

Através dessa equação é possível melhor entender como ocorre o processo de recuperação: a perda de energia dos átomos começa bastante intensa (devido à quantidade de átomos em estado de alta energia), e diminui de maneira exponencial (Figura 4), conforme o tempo avança. Nunca, em teoria, os átomos retornam ao estado anterior ao pulso (após um período de $7 \cdot T_1$ ocorre um retorno de 99,9%).

Figura 4 – Gráfico de recuperação T1.



Fonte: Larsson (2015).

2.3.2 Decaimento T2

Semelhante à recuperação T1, o decaimento T2 também se dá logo ao término do pulso RF, porém, diferentemente da perda pelas interações térmicas, o decaimento T2 ocorre devido à dessincronia das precessões dos átomos. Encerrado o pulso que criava a ressonância, a influência dos próprios dipolos dos átomos sobre os seus vizinhos acaba por dessincronizar as precessões dos mesmos, dando origem a um defasamento que reduz a intensidade ao longo do plano transversal. Como a origem desse decaimento, ou relaxamento, é devida a uma relação entre spins dos átomos, dá-se o nome relaxamento *spin-spin* (HAGE; IWASAKI, 2009). De maneira análoga ao T1, o decaimento T2 também depende do instante de tempo observado após o término do pulso RF, de acordo com a Equação 4

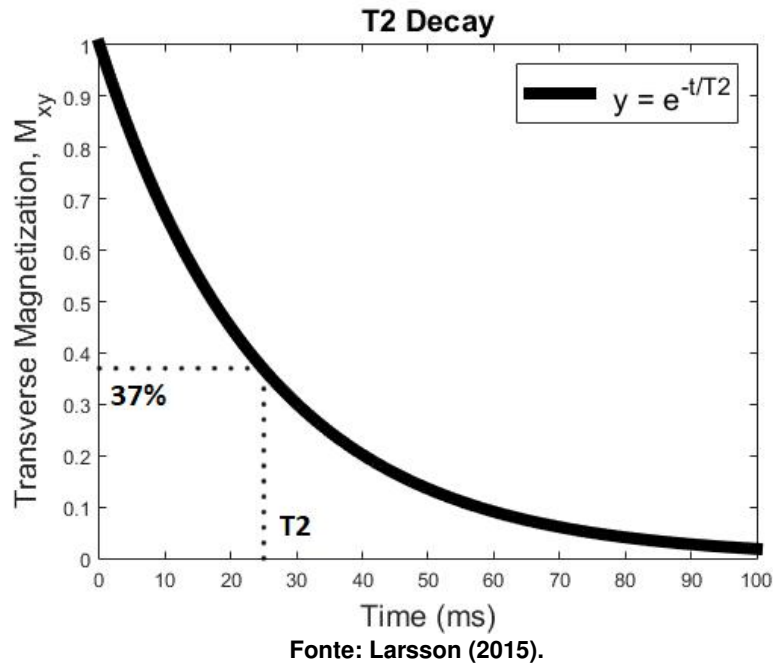
$$M_{xy_0}(t) = M_{xy}(e^{-\frac{t}{T2}}), \quad (3)$$

na qual: $M_{x,y}$ é a magnetização máxima no plano transversal, ou seja, o último instante antes do término do pulso RF; T2 é a constante de tempo para que a magnetização se reduza a 37% do seu valor máximo.

Conforme também observado na recuperação do campo longitudinal, o decaimento do campo transversal se dá em uma razão exponencial (Figura 5), sendo a velocidade desse decaimento inversamente proporcional à magnitude de T2.

Outro fator que influencia no decaimento T2 são as inhomogeneidades, ou seja, as variações e instabilidades do campo magnético B_0 (devido a essa influência, T2 sempre é menor que T1). Graças a essas inhomogeneidades, com o fim do pulso RF, a fase da precessão dos átomos

Figura 5 – Gráfico de decaimento T2.



dessincroniza com mais velocidade de acordo com a intensidade das inhomogeneidades, por se tratar de um relaxamento *spin-spin*.

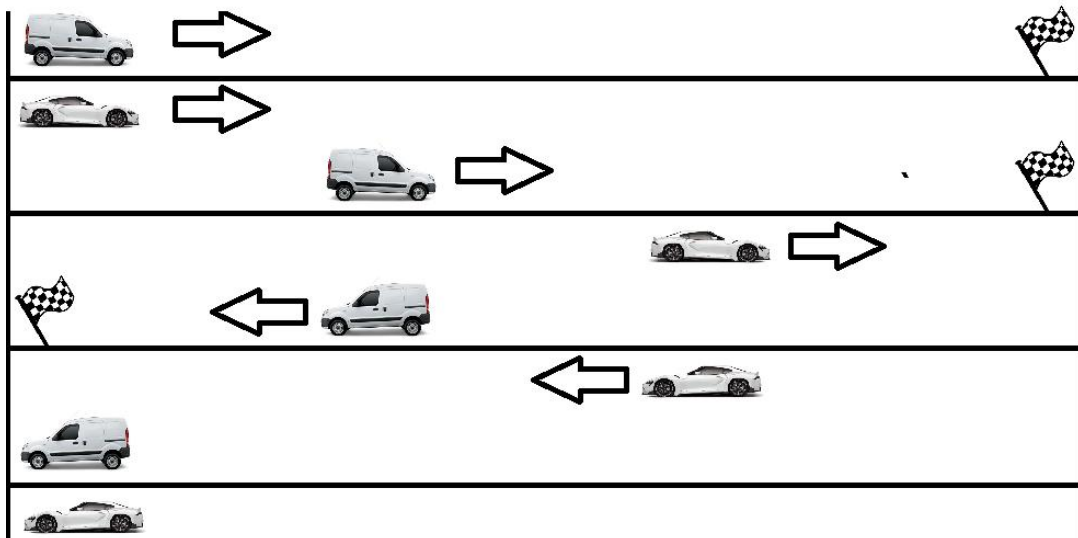
A fim de minimizar as influências das inhomogeneidades, é necessário um ajuste para que as precessões entrem novamente em fase. Para isso, utiliza-se de um pulso auxiliar, defasado em 180° em relação ao pulso anterior, criando assim um novo referencial para o momento das precessões, eliminando o impacto das inhomogeneidades. Para melhor visualização, pode-se utilizar a analogia da corrida (Figura 6), na qual têm-se, em um primeiro momento, um veículo esportivo competindo com um carro utilitário em uma pista reta. O carro esportivo, que tem maior velocidade, consegue atingir uma maior distância em um curto período de tempo, quando comparado ao utilitário, que tem um desempenho inferior. Porém, passado um certo tempo, a chegada da corrida muda para o ponto de partida, fazendo assim com que o utilitário assuma a liderança. Como o tempo para o retorno do utilitário é o mesmo para o retorno do esportivo, ambos cruzam a linha de chegada simultaneamente. Nessa analogia, tem-se ambos os veículos como átomos com diferentes precessões e, o momento de troca da chegada para o ponto de partida, é o momento do pulso de 180° , que faz com que os veículos cruzem a linha de chegada juntos, ou seja, faz com que as precessões se realinhem novamente por um momento.

Caso o decaimento seja mensurado unicamente a partir do pulso inicial, ou seja, sem o pulso à 180° , tem-se o chamado decaimento $T2^*$. A partir da Equação 4

$$\frac{1}{T2^*} = \frac{1}{T2} + \gamma \Delta B_0, \quad (4)$$

onde γ é a razão giromagnética do átomo de hidrogênio e ΔB_0 é a intensidade da inhomogeneidade no campo. Pode-se perceber que $T2^*$ é inversamente proporcional à intensidade

Figura 6 – Pulso de sincronização - analogia da Corrida.



Fonte: Autoria própria.

das inhomogeneidades no campo magnético B_0 , ou seja, quanto maior as inhomogeneidades, menor é o tempo de decaimento e, inversamente, caso o B_0 seja totalmente homogêneo, tem-se o valor de $T2^*$ igual a $T2$.

2.4 Sequência de pulsos

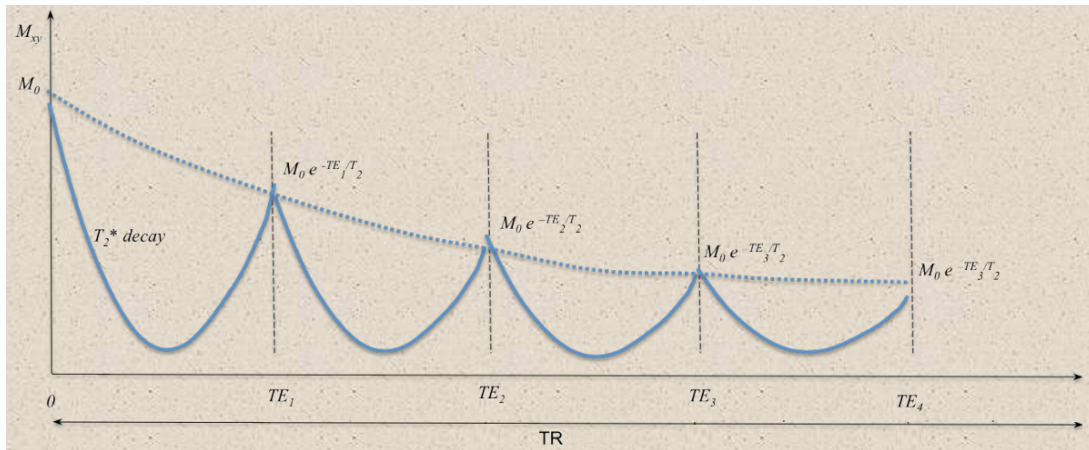
Considerando a necessidade de pulsos auxiliares para o ajuste da fase para a geração do sinal $T2$, define-se uma sequência de pulsos para a obtenção dos sinais necessários durante o procedimento. Para definição dessa sequência, uma parametrização se faz necessária, devido as técnicas do *hardware* e a padronização do exame.

Dentro desses parâmetros, dois tem maior destaque, sendo eles o tempo de repetição (TR) e o tempo de eco (TE) (JORNADA, 2017). O TR define o período de tempo entre cada um dos pulsos RF, e seu valor é definido com base no tempo de relaxamento $T1$. Um TR muito curto resultaria em um realinhamento antes de permitir que todos os tecidos tenham relaxado da posição imposta pelo pulso, podendo impactar negativamente os resultados obtidos. Já TE define o tempo entre o pulso RF e a coleta do sinal, podendo o mesmo ser logo após o pulso RF (para obtenção dos valores $T1$), ou após o pulso de ajuste de fase a 180° (para obtenção dos valores $T2$). Para o caso da análise com foco em $T2$, o TE necessita ser longo a fim de maximizar o tempo de decaimento, dando maior contraste para os diferentes tecidos.

A sequência de pulsos com apenas um pulso RF com um pulso de ajuste de fase é conhecida como *spin echo* ou spin eco (SE), na qual o pulso RF inicial tem mais intensidade e o pulso de ajuste tem apenas por objetivo realinhar as fases. Uma possível variação dessa sequência é a *multiple spin echo*, ou multi spin eco (MSE) (Figura 7). Ela permite que, utilizando uma série de pulsos de 180° , vários sinais sejam captados sem a interferência do desalinha-

mento das fases, possibilitando ambos TR e TE longos para a obtenção dos sinais para a análise T2.

Figura 7 – Sequência de *multiple spin echo* para obtenção dos valores T2.



Fonte: Stikov (2012).

2.5 Mecanismos de contraste

O fator comparativo é crucial para tirar conclusões a respeito do sinal mensurado durante a ressonância magnética, ou seja, a análise apenas faz sentido quando se comparados os sinais recebidos entre diferentes tecidos (WESTBROOK; ROTH; TALBOT, 2011). Como se tratam de imagens, esse contraste se dá através dos níveis de cinza da área, sendo áreas com maior intensidade de sinal traduzidas em áreas brancas na imagem e, conforme diminui a intensidade, mais escura se torna a área até ficar completamente preta na ausência de qualquer sinal.

Diferentes sinais podem ser obtidos através da ressonância magnética, sendo cada um deles determinado por uma série de fatores (WESTBROOK; ROTH; TALBOT, 2011). Para os contrastes obtidos com os sinais de relaxamento T1 e decaimento T2, três fatores são centrais, sendo eles:

- **A energia inerente do tecido** influencia na capacidade da malha de absorver energia, impactando diretamente no relaxamento T1, que depende da dissipação de energia através a absorção da mesma pela malha no entorno do átomo;
- **A densidade do tecido** afeta a proximidade dos átomos de hidrogênio, alterando assim as interações entre os campos magnéticos de cada átomo, impactando diretamente o decaimento T2, que depende dessas interações para dissipar energia;
- **A semelhança entre a vibração da molécula e a frequência de Larmor do hidrogênio** afeta a difusão de energia dos átomos para a matriz, assim como a energia do tecido. Essa relação ocorre por conta de um fenômeno semelhante ao da ressonância, que facilita a transferência de energia, impactando assim os valores de T1.

Imagens com base em T1 dependem da intensidade do sinal de recuperação magnética paralela a B_0 , ou seja, baseiam-se majoritariamente na recuperação T1. Devido à rápida recuperação do sinal, utilizam-se valores curtos de TR e TE, a fim de evitar o realinhamento completo dos prótons de hidrogênio, o que deixaria a imagem totalmente homogênea e sem contraste (JONES; HAOUIMI, 2021).

Para imagens com base em análise T2, o sinal obtido diz respeito à intensidade do vetor no plano transversal ao campo B_0 , e necessita de um longo TR e um curto TE no caso do uso da sequência MSE. Isto ocorre pois os TEs são determinados no instante de reajuste das fases e, como a sequência conta com uma quantidade grande de ecos, o próximo pulso RF principal terá um grande intervalo de tempo em relação ao seu anterior, ou seja, um longo TR (JONES; BABA, 2021).

2.6 Aplicações de análise T2

Devido à sua natureza não invasiva e a não necessidade de irradiar a área com raios X, a RM tem seu uso difundido para diversas aplicações, como caracterização do miocárdio, avaliação de cartilagem, entre outros (FEGGER, 2021).

As imagens obtidas através da análise de RM utilizando o resultado T2 são extremamente valiosas, pois apresentam detalhes morfológicos e detalhes de composição química dos tecidos que podem ser utilizados para o diagnóstico de doenças. No entanto, as imagens por si só não são o suficiente, pois são necessários tanto uma referência saudável para comparação, quanto outros resultados de exames, a fim de cruzar informações e poder chegar a uma conclusão sobre o caso.

Algumas das patologias que podem ser identificadas com o auxílio da RM são:

- Síndrome coronariana aguda/infarto do miocárdio;
- Miocardite;
- Sarcoidose cardíaca;
- Cardiomiopatia de Takotsubo;
- Rejeição de transplante de coração;
- Amiloidose cardíaca.

2.7 Tecnologias utilizadas

Com objetivo de desenvolver um algoritmo capaz de gerar o mapa T2, assim como um software capaz de visualizá-lo e editá-lo, optou-se por utilizar um conjunto de bibliotecas já disponíveis para uso público.

Essas bibliotecas foram selecionadas por serem orientadas para a execução de operações matemáticas, simplicidade de uso e capacidade de atender os requerimentos básicos para importação, tratamento e exportação dos arquivos Dicom utilizados.

2.7.1 Tkinter

A biblioteca *Tkinter* conta com um conjunto de objetos que tem por objetivo criar interfaces utilizando estruturas chamadas *widgets*. Através desses *widgets* é possível criar tanto elementos que recebem *inputs* do usuário (caixas de texto, botões, etc), quanto elementos que retornam informações ao mesmo (janelas de *log*, imagens, tabelas, etc). Suas ferramentas ainda permitem o uso de processamento paralelo, através de threads, recurso este fundamental para softwares com intensa interação com o usuário (PYTHON SOFTWARE FOUNDATION, 2022). Adicionalmente ao *Tkinter*, tem-se também o *TtkThemes*, que expande as possibilidades de *widgets* e permite a aplicação de temas visuais, a fim de padronizar esteticamente as interfaces geradas.

2.7.2 Numpy

Numpy é uma biblioteca que conta com um objeto de matriz (*array* multidimensional) como sua estrutura central, disponibilizando ao desenvolvedor métodos otimizados de interação com essa mesma estrutura, permitindo cálculos, análises lógicas, redimensionamento, entre outros (HARRIS *et al.*, 2020). Sua aplicação é central para diversas bibliotecas populares no meio científico (como *Matplotlib*, *Scipy*, entre outras), devido ao seu rápido processamento e fácil uso.

2.7.3 OpenCV

OpenCV é uma biblioteca que possibilita aplicações de processamento digital de imagens, com foco em visão computacional, permitindo, inclusive, o desenvolvimento de soluções utilizando aprendizado de máquina (BRADSKI, 2000).

2.7.4 Matplotlib

A biblioteca *Matplotlib* dispõe de ferramentas para visualização de listas e matrizes em forma gráfica. Com a capacidade de criação de imagens e visualização de dados, seu uso se assemelha ao software MATLAB, contando até mesmo com métodos idênticos, com a vantagem de ser de uso livre (HUNTER, 2007).

2.7.5 PyFPDF

PyFPDF é uma biblioteca que tem sua origem em uma biblioteca em PHP chamada FPDF (*Free PDF*) e é utilizada para geração de arquivos pdf. A biblioteca faz uso de um sistema de coordenadas para inserir textos e imagens nos documentos, além de outras funções que facilitam a geração e edição de documentos (FENNIAK, 2010).

2.7.6 Pydicom

A biblioteca *Pydicom* disponibiliza recursos para interação com arquivos Dicom. Suas principais funcionalidades são a extração das informações codificadas neste formato, permitindo ao desenvolvedor acesso às informações do paciente (nome, idade, etc), do exame (horário, parâmetros do equipamento) e, principalmente, a matriz de valores gerada a partir dos sinais obtidos durante o processo de imageamento. Além disso, a biblioteca permite a construção de arquivos Dicom e a conversão dos mesmos para outros formatos (MASON *et al.*, 2021).

2.7.7 MATLAB

No MATLAB (nome originado de *Matrix Laboratory*), tanto o ambiente de desenvolvimento quanto a linguagem utilizada no mesmo apresentam soluções para manipulação e visualização de dados (MATLAB, 2010). No MATLAB, as bibliotecas para aplicações específicas são denominadas *toolboxes*, havendo, por exemplo, *toolboxes* para processamento de imagens, estatística, aprendizagem de máquina, entre várias outras.

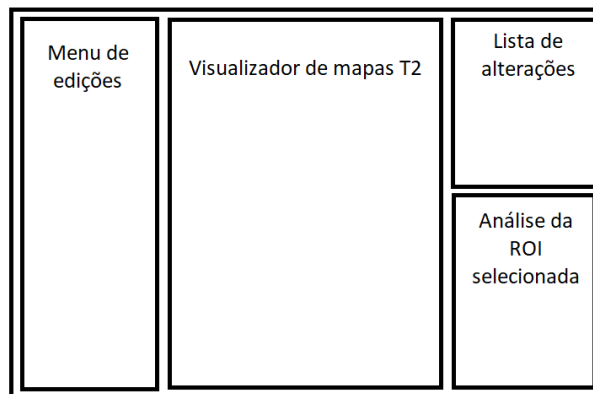
3 METODOLOGIA

Este capítulo tem por objetivo apresentar a metodologia utilizada, detalhando as bibliotecas usadas no desenvolvimento do software, com maior destaque para o processo de geração dos mapas T2. Para isso, o capítulo será dividido quatro partes, sendo elas: interface, com aprofundamento no uso da biblioteca *Tkinter*, processamento de imagem, com as aplicações das bibliotecas *Numpy*, *Matplotlib* e *OpenCV*, documentação, através da biblioteca *PyFPDF* e, por fim, geração dos mapas T2.

3.1 Interface

O desenvolvimento da interface foi realizado inteiramente em *Tkinter*, com o objetivo de criar uma interface simples e funcional tanto para o usuário, quanto para o posterior desenvolvimento de novas funções. Sendo assim, utilizou-se de uma estrutura modular (Figura 8), dividida em quatro blocos: (i) um menu de edições, no qual pode-se importar, editar e exportar os resultados; (ii) um visualizador para os mapas T2, sendo possível a visualização simultânea do mapa T2 e das imagens Dicom utilizadas; (iii) uma área de edições, na qual é possível listar as edições realizadas; (iv) uma área com informações da análise da região de interesse (ROI - *Region of interest*), determinada pelo usuário, do mapa T2.

Figura 8 – Estrutura da interface.



Fonte: Autoria própria.

3.2 Processamento de imagem

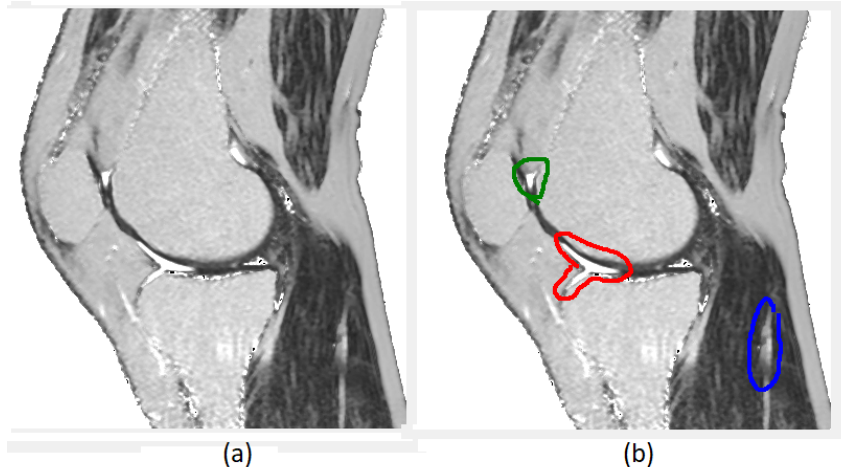
O processamento de imagem é realizado, na sua maior parte, dentro dos métodos do menu de edição. Como a maioria dos métodos de edição depende da seleção de posições dentro das fronteiras da imagem, utilizou-se o objeto canvas da biblioteca *Tkinter*, que dispõe de três eventos que auxiliam na obtenção dessas posições. Os eventos reconhecem o pressionar do botão esquerdo do mouse, o deslocamento do cursor sobre o canvas e o liberar do mesmo

botão, disparando, em cada uma dessas situações, um método distinto para o tratamento da posição do cursor. Assim que cada um dos métodos que os utilizam são finalizados, os eventos são desvinculados do canvas, possibilitando o uso de outras ferramentas posteriormente.

Dentre as funções desenvolvidas, estão:

- **Pincel:** A função pincel, quando selecionada, dispara ao pressionar do botão esquerdo do mouse um método para coleta da posição do cursor, assim como também adiciona a edição na lista de edições realizadas sobre o mapa. Em seguida, conforme o usuário desloca o cursor sobre o canvas, outro método é disparado, coletando, a cada deslocamento identificado, um novo par de coordenadas. Além disso, a cada execução do método, é realizado o cálculo para identificar a diferença entre as últimas coordenadas e as atuais. Caso o cálculo supere um limiar de distância, um novo par de coordenadas é adicionado entre os dois pontos, com objetivo de suavizar o traçado sobre a imagem (Figura 9).

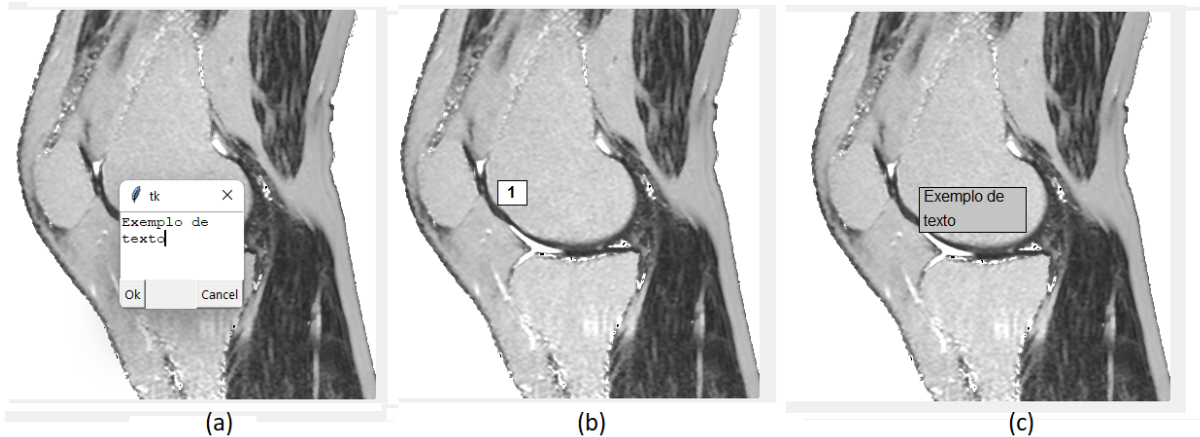
Figura 9 – Antes (a) e depois (b) da aplicação do pincel para destaque de possíveis áreas de interesse.



Fonte: Autoria própria.

- **Zoom:** A função Zoom, quando selecionada altera o valor de zoom ativo registrado no programa, multiplicando (para o *zoom in*), ou dividindo (para o *zoom out*), o valor atual do zoom por um fator fixo. Somado a isso, o zoom também altera os elementos de rolagem vertical e horizontal do visualizador da página, utilizando as opções de rolagem disponíveis no elemento canvas utilizado.
- **Etiqueta:** A função etiqueta utiliza apenas o evento inicial de pressionamento do botão do mouse, com objetivo de coletar a posição do cursor. Com essa informação, uma entrada de texto para o conteúdo da etiqueta é solicitada ao usuário, em forma de uma nova janela. Preenchido o campo, é adicionada à imagem uma etiqueta numerada que, quando interagida, revela o texto inserido, apresentando também a mesma informação na lista de edições (Figura 10).

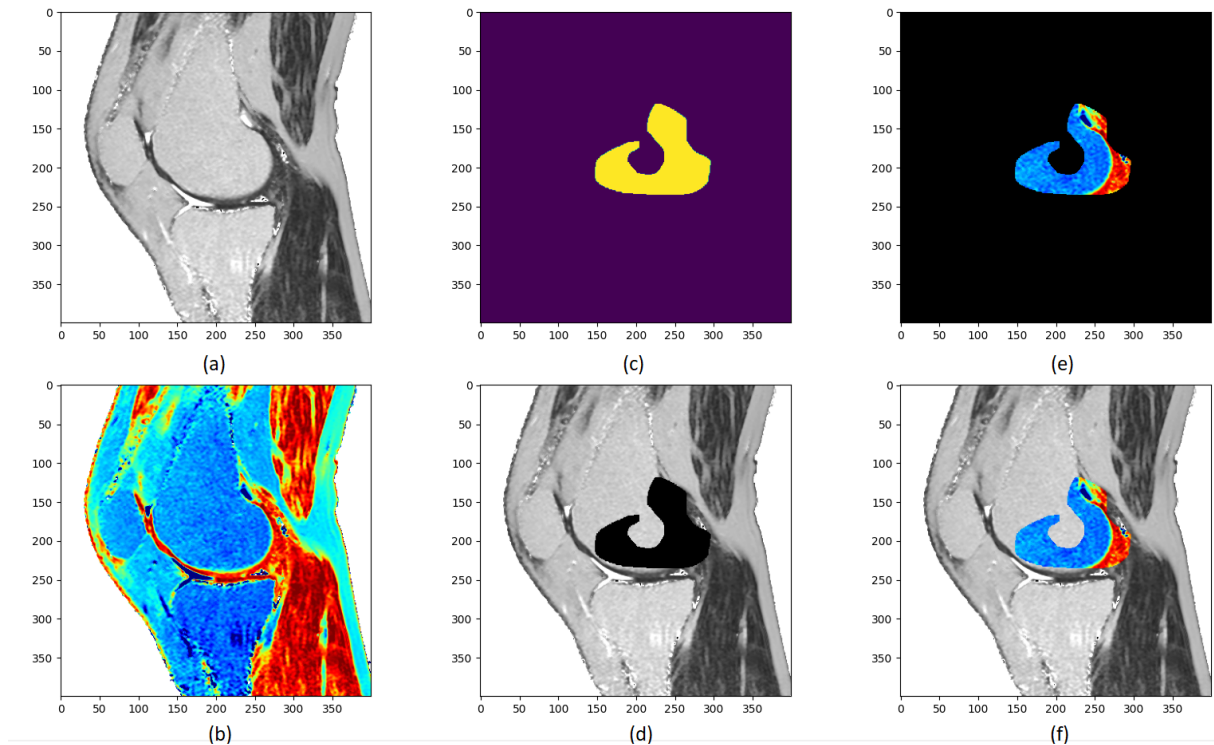
Figura 10 – (a) Janela para entrada do texto da etiqueta, (b) etiqueta sobre a imagem no visualizador e (c) etiqueta apresentando o texto salvo.



Fonte: Autoria própria.

- **ROI:** A função ROI é dividida em três diferentes morfologias, sendo elas: retangular, circular e mão-livre. Todas as funções, ao serem selecionadas, disparam a criação do objeto ROI, porém, a partir da criação, cada morfologia tem um processamento distinto. A função retangular, inicialmente, coleta a posição do cursor e cria um retângulo de perímetro de cor preta para indicar ao usuário os limites da ROI. Conforme o cursor se desloca, a função retangular atualiza os limites do retângulo, atualizando também o retângulo desenhado. A função circular, apesar de também ser limitada por uma forma retangular, difere da anterior, pois cria um círculo para a visualização das fronteiras da ROI. Além disso, no deslocamento, é necessário identificar o quadrante da posição atual do cursor em relação à referência inicial e identificar qual é a menor dimensão do retângulo delimitante, porque, dependendo do quadrante em que se localiza o cursor e da menor dimensão, altera-se o cálculo do retângulo que contém o círculo que delimita a ROI. A função mão-livre difere totalmente dos métodos anteriores, se assemelhando muito mais ao método de pincel, porque, a cada deslocamento identificado, adiciona um novo par de coordenadas, além de desenhar em preto sobre o desenho o traço realizado enquanto o botão não é liberado. Assim que o usuário finaliza a seleção e solta o botão, cada uma das opções cria uma máscara associada ao mapa (Figura 11), utilizando os pontos fornecidos para delimitar a região válida da mesma (no caso da mão livre, os pontos determinam um polígono criado com a função *fillPoly*, da biblioteca *OpenCV*). Essa máscara é então aplicada sobre as duas versões do mapa: um espaço vazio (preto) é recortado na imagem em escala de cinza, e o inverso aplicado sobre a colorida, deixando apenas o espaço selecionado e todo o resto vazio. As versões são então somadas e, o espaço vazio na versão em escala de cinza é preenchida pela imagem colorida.

Figura 11 – A partir das duas imagens iniciais (a) e (b), quando aplicada a máscara (c), são criadas as imagens (d) e (e) que, quando somadas, resultam na imagem (f).



Fonte: Autoria própria.

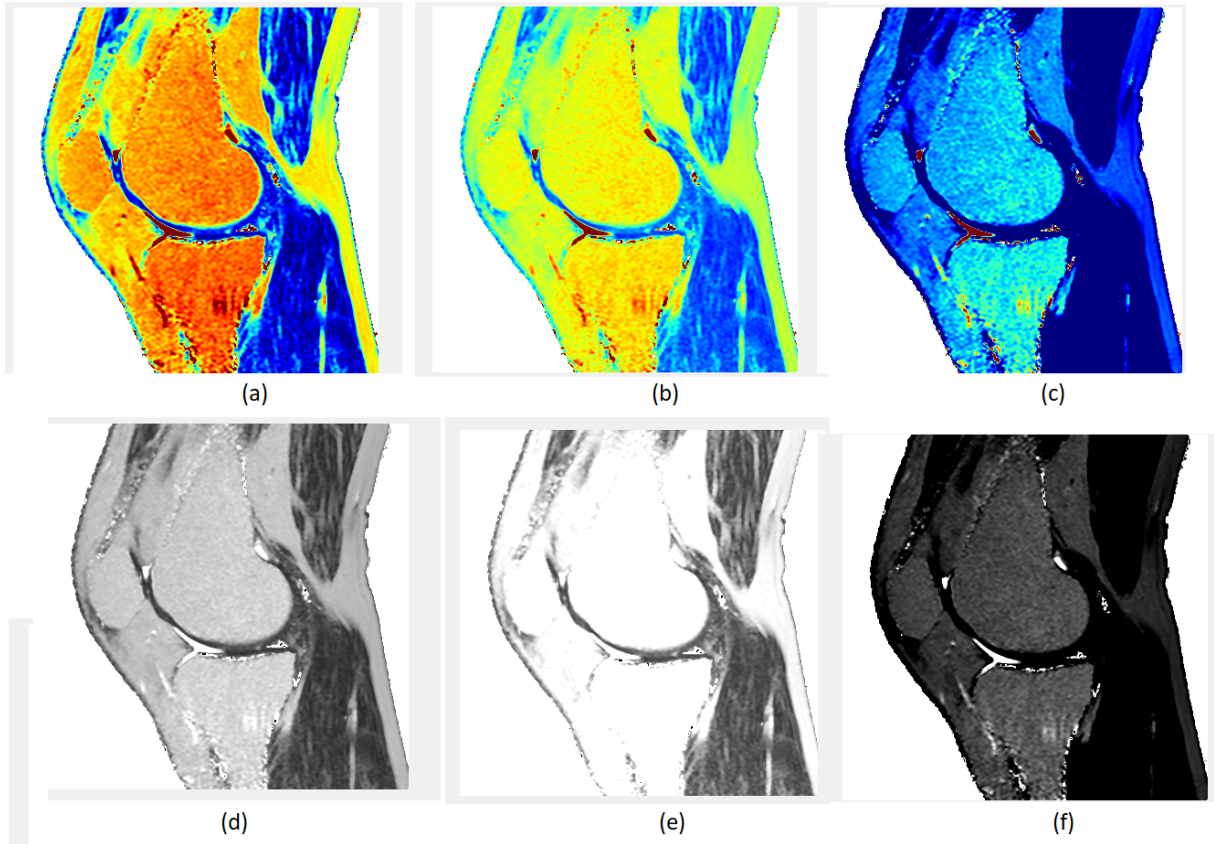
- **Mapas de cor:** A função Mapas de cor altera os valores limites de cada um dos dois mapas utilizados, sendo eles o mapa *jet* (usado para dar destaque nos elementos dentro da ROI) e a escala de cinza. Conforme o usuário altera os valores limites de cada um dos mapas (tanto limite superior quanto inferior), um método é disparado para redefinir as correspondências entre valores e o mapa, atualizando as cores do mapa *jet* ou a intensidade da escala de cinza (Figura 12).

3.3 Documentação

A geração de documentação se dá através do uso da biblioteca PyFPDF, utilizando seus métodos para posicionar as informações obtidas no mapa T2, assim como as informações de ROI e textos das etiquetas inseridas.

Para organização do documento, o mesmo foi dividido em três partes, sendo elas: informações gerais, na qual serão inseridas informações relativas ao mapa gerado, como o mapa representado com um mapa de cor *jet* e o resultado final do exame pós edições; regiões de interesse, com informações das ROIs definidas; e etiquetas inseridas, com a lista de etiquetas posicionadas na edição, com seu respectivo texto. Devido à possibilidade de criação de diversas ROIs e etiquetas, ambas regiões não possuem tamanho fixo, podendo se expandir para múltiplas páginas, se necessário, para abrigar todas as edições realizadas.

Figura 12 – (a) Mapa jet, limites 0-146. (b) Mapa jet, limites 0-250. (c) Mapa jet, limites 120-250. (d) Escala de cinza, limites 0-146. (e) Escala de cinza, limites 0-250. (f) Escala de cinza, limites 126-250.



Fonte: Autoria própria.

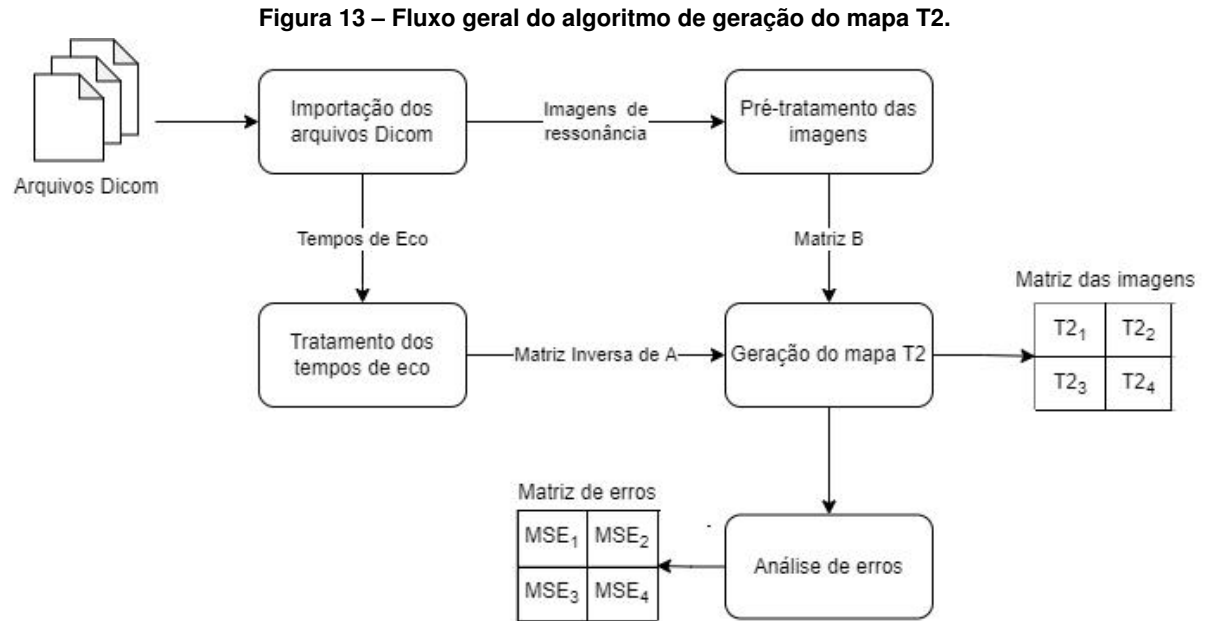
3.4 Algoritmo para processamento do mapa T2

O processo de geração de um mapa T2 teve seu desenvolvimento baseado em análises matriciais através do uso da biblioteca Numpy. Com isso, pode-se dar um foco maior em velocidade de processamento, devido às otimizações já existentes na biblioteca para tratamento de matrizes com grandes quantidades de elementos. No entanto, como cada passo não resulta mais em uma matriz que tenha valor de análise se interpretada como imagem, limitou-se a capacidade de análise do processo passo a passo.

As próximas seções abordam em detalhes cada uma das etapas do algoritmo ilustrado na Figura 13.

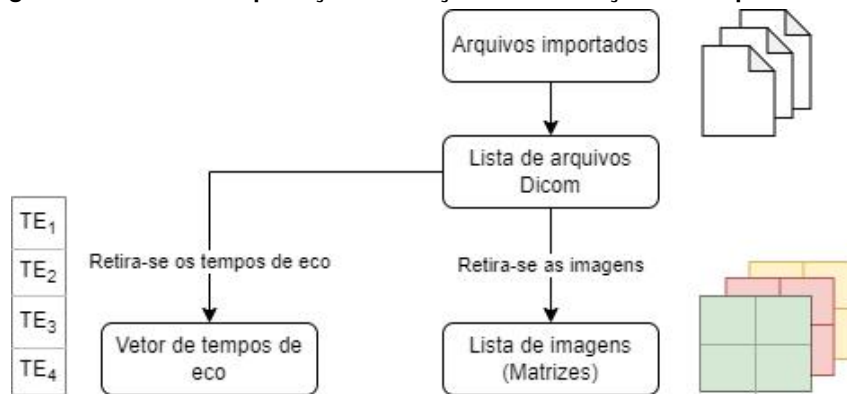
3.4.1 Importação e pré-tratamento

Inicialmente, os arquivos Dicom importados tem suas informações extraídas, sendo as imagens obtidas como matrizes onde cada elemento da matriz reflete um pixel na imagem, e os valores de TE como variáveis que são agrupadas em um vetor (Figura 14).



Fonte: Autoria própria.

Figura 14 – Fluxo da importação e extração da informação dos arquivos Dicom.



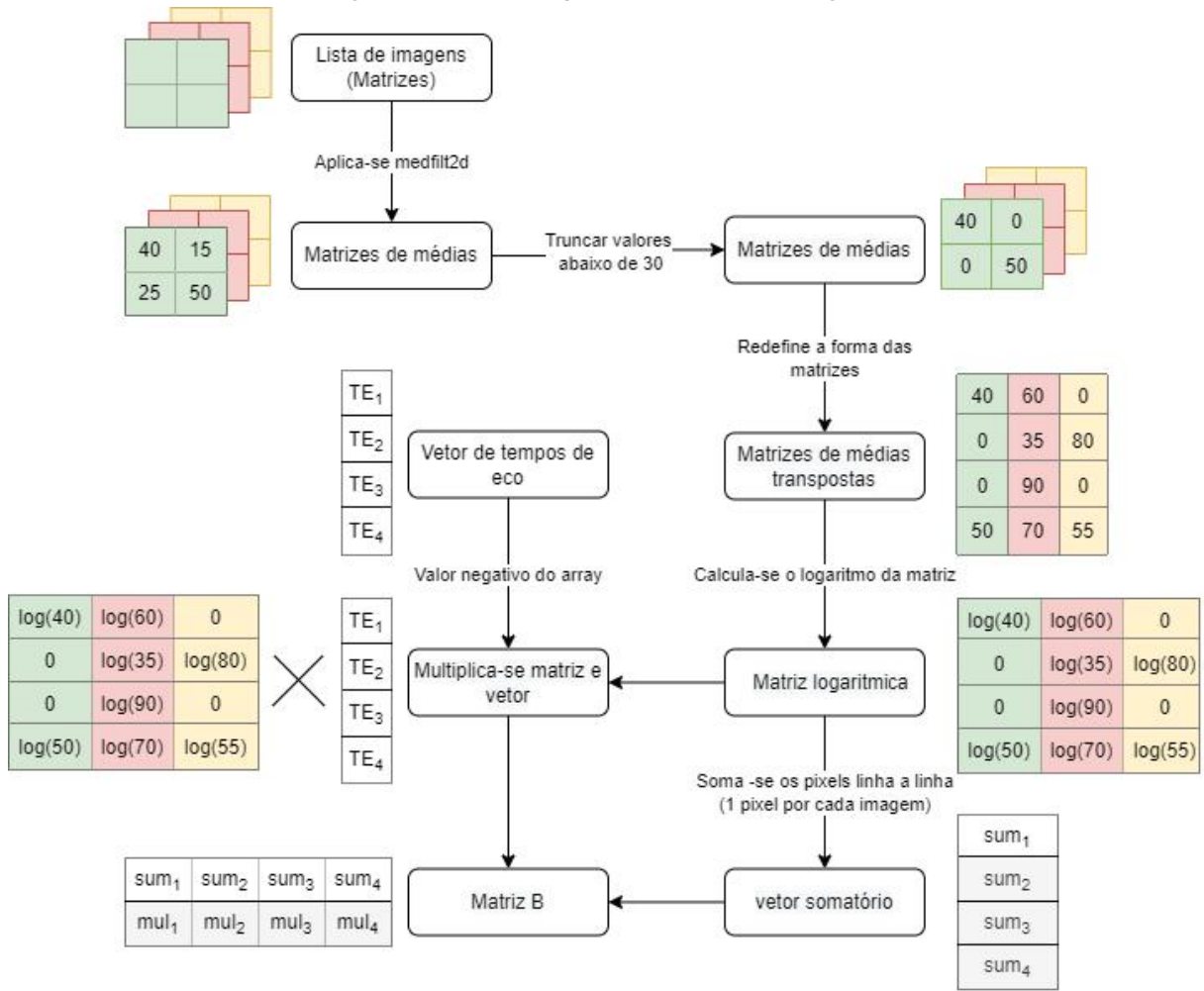
Fonte: Autoria própria.

Com as matrizes das imagens, é feito o pré-tratamento das informações para o cálculo do mapa final (Figura 15). Para isso, primeiro retira-se os possíveis ruídos gerados pelo hardware e presentes nas imagens, fazendo uso da função `medfilt2d`.

O método `medfilt2d` percorre a imagem agrupando todas as vizinhanças de pixels possíveis em uma dimensão 3×3 e, a cada vizinhança, calcula a média dos nove elementos presentes, e o salva no ponto central da matriz. Em contrapartida, o método acaba por eliminar os elementos nas margens da matriz, pois não é possível substituir o valor central quando utilizada uma vizinhança 3×3 (Figura 16).

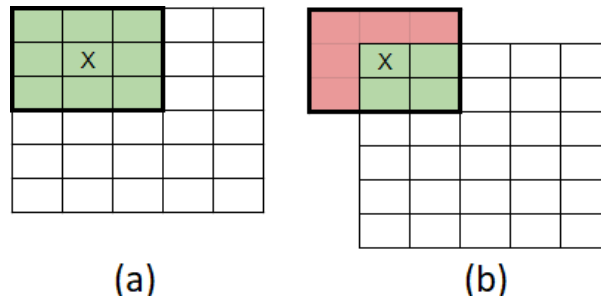
A partir das matrizes dos valores médios (Figura 15), substituem-se todos os valores abaixo de 30 na matriz por zero e converte-se todas as matrizes em vetores de dimensões $M \cdot N \times 1$. Em sequência, concatenam-se todos os vetores em uma única matriz de dimensões $M \cdot N \times L$, sendo M e N as dimensões originais da imagem, e L a quantidade de arquivos Dicom utilizados na geração.

Figura 15 – Fluxo do pré-tratamento das imagens.



Fonte: Autoria própria.

Figura 16 – O medfilt2d aplicado a uma matriz depende de que toda a vizinhança esteja dentro da matriz (a), caso contrário, o método não consegue calcular a média (b).



Fonte: Autoria própria.

Com a matriz consolidada a partir de todos esses vetores, aplica-se o logaritmo ponto a ponto na matriz. Como vários dos valores são zero, a fim de evitar o limite ao infinito do logaritmo de zero, mantém-se os valores como zero mesmo após a aplicação.

Segue-se, então, dois diferentes passos: primeiro soma-se a matriz linha a linha, fazendo com que a soma seja feita sobre todos os valores que referenciam a mesma posição em cada imagem, ou seja, a soma é feita pixel a pixel com todas as imagens, resultando em um vetor de

dimensões $M \cdot N \times 1$ (a dimensão L torna-se um). Segundo, multiplica-se a mesma matriz que tem os resultados dos logaritmos com o vetor que armazena os valores de TE. Como trata-se de uma multiplicação de matrizes, o produto deve ser feito seguindo o modelo $A \cdot B$ aonde A é a matriz e B o vetor. Isso é necessário devido às dimensões de cada um dos termos da multiplicação, pois a matriz tem dimensões $M \cdot N \times L$ e o vetor dimensões $L \times 1$, o que torna impossível a multiplicação no sentido oposto ($B \cdot A$). O resultado do produto é também um vetor, porém com dimensões $M \cdot N \times 1$, semelhante ao outro resultado sobre a mesma matriz. A partir de ambos os resultados, pode-se criar a matriz chamada B , concatenando ambos os resultados. Para criação de B , utilizou-se o método `stack` da biblioteca NumPy, resultando assim não em uma matriz de dimensões $M \cdot N \times 2$, porém na sua transposta ($2 \times M \cdot N$), a fim de facilitar cálculos futuros.

3.4.2 Tratamento dos tempos de eco

Além do processamento das imagens dos arquivos Dicom, são processados também os tempos de eco de cada um dos arquivos (Figura 17). A partir dos TE obtém-se dois valores: o somatório de todos os TE e o somatório dos quadrados dos TE.

Para otimizar o processamento do somatório dos quadrados, utilizou-se da técnica de multiplicar pela direita o vetor dos TE pelo seu transposto. Com isso, o produto das matrizes resulta em apenas um valor escalar, já consolidado a partir das somas dos respectivos produtos elemento a elemento. Essa técnica evita o uso de laços explícitos no código, executando todo processo através das soluções otimizadas de cálculo matricial da biblioteca NumPy.

Como conclusão desse bloco do algoritmo, organiza-se a matriz A , que é uma matriz quadrada de dimensões 2×2 , na qual seus elementos são:

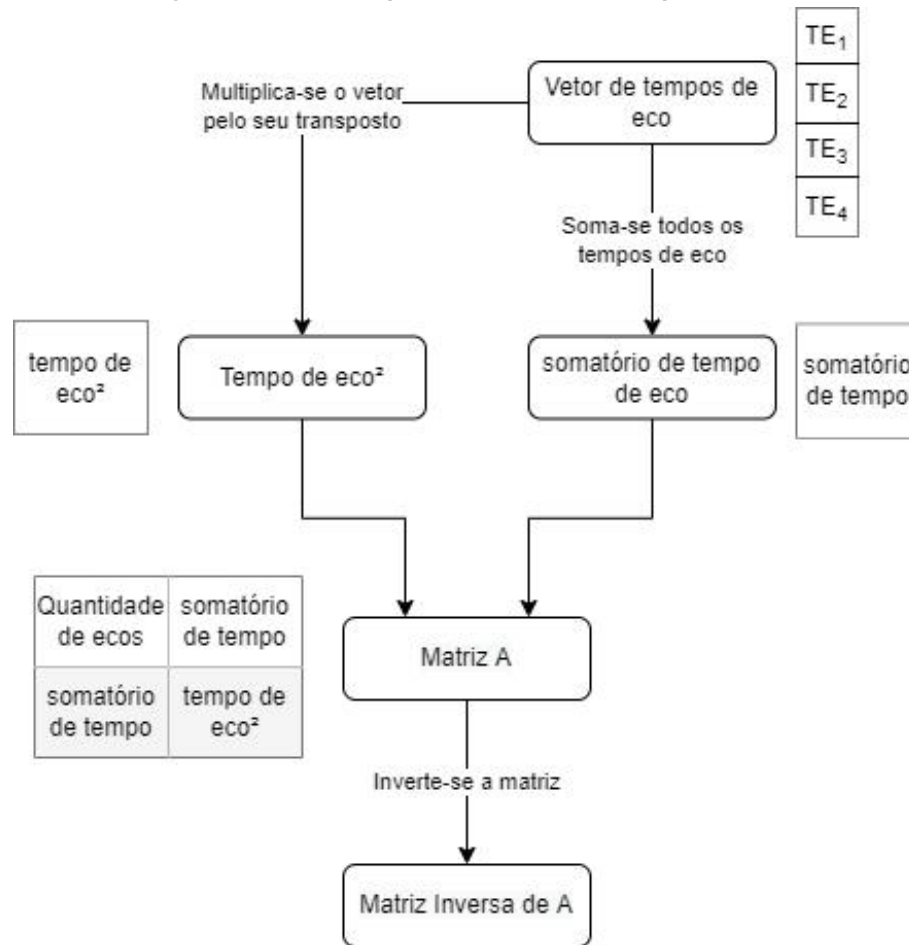
- (1,1): Quantidade de ecos utilizados para geração do mapa;
- (1,2) e (2,1): Somatório dos TE;
- (2,2): Somatório dos quadrados dos TE;

Finalizada a criação da matriz A , inverte-se a mesma, obtendo-se A^{-1} , utilizando o método `inv` do módulo de álgebra linear da biblioteca NumPy. Pode-se garantir que A sempre será inversível, pois o determinante de A , dado pela Equação 5

$$\det(A) = N * \sum_{n=1}^N TE_n^2 - 2 * \sum_{n=1}^N TE_n, \quad (5)$$

sempre será diferente de zero, pois $TE \neq 0$ e os valores de TE não se repetem ao longo do exame, impedindo assim que a equação tenha apenas os valores que são raízes como valores de TE.

Figura 17 – Fluxo do processamento dos tempos de eco.



Fonte: Autoria própria.

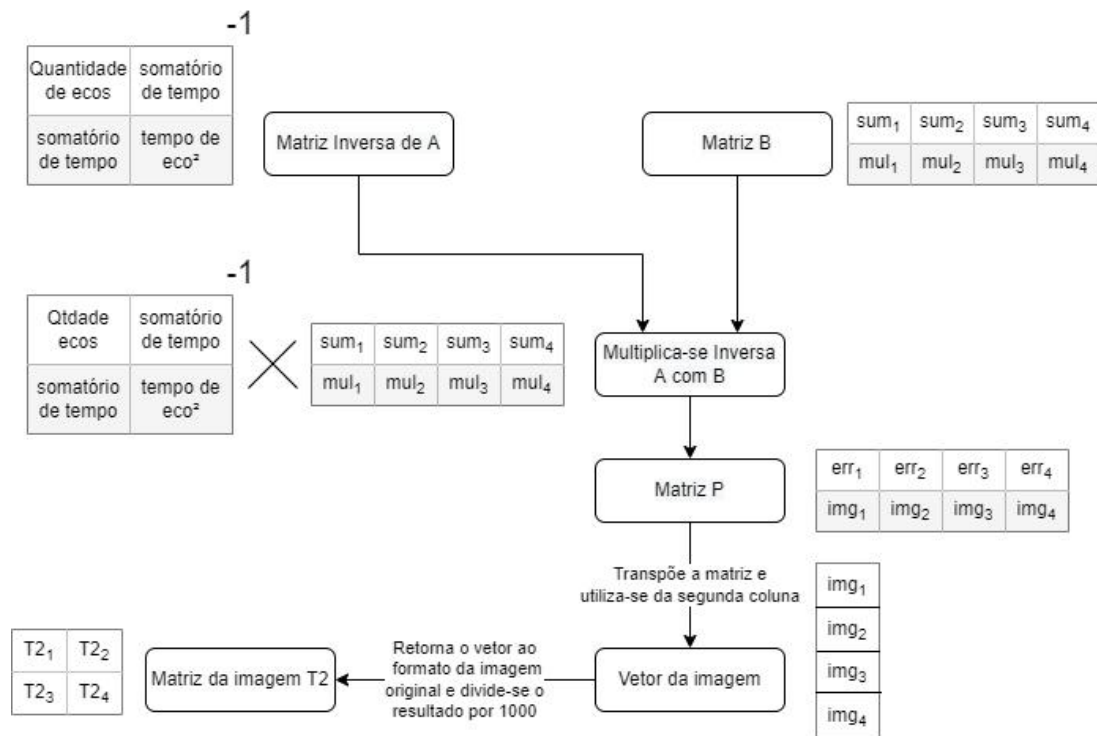
3.4.3 Geração do mapa T2 e Análise de erros

Conforme ilustrado na Figura 18, com ambas as matrizes A^{-1} e B , pode-se realizar o produto de ambas no sentido $A^{-1} \times B$. O resultado, chamado de matriz P , tem dimensões $2 \times M \cdot N$ e pode ser dividida de maneira imediata em dois vetores, sendo eles o vetor com as informações para o cálculo do erro envolvido no processo de aproximação, chamado P_0 , e o vetor dos valores que se tornarão a matriz do mapa T2, chamado P_1 .

Como passo final no algoritmo para criação da imagem, transpõe-se P_1 , diminui-se o resultado em três ordens de grandeza (divide-se por 1000) e redefine-se a forma do vetor para a mesma forma da imagem original, ou seja, de $1 \times M \cdot N$ para $M \times N$.

Já com P_0 , segue-se para o cálculo do EQM (erro médio quadrático), conforme ilustrado na Figura 19. Para isso, é realizado o produto de P_1 com o vetor dos tempos de eco. Como ambos são vetores de diferentes tamanhos, o produto só pode ser realizado de forma que o resultado terá ambas as dimensões diferentes de um dos vetores, gerando assim uma matriz de dimensões $L \times M \cdot N$. Com a matriz resultante, calcula-se a exponencial elemento a elemento da matriz e, em seguida, cada linha da matriz exponencial é multiplicada elemento a elemento pelo vetor P_0 .

Figura 18 – Fluxo do processamento final para geração do mapa T2.



Fonte: Autoria própria.

Com a matriz resultante do último produto, aplica-se o EQM utilizando como estimativa inicial a matriz transposta das médias, ou seja, é feita a subtração das duas matrizes e cada elemento do resultado elevado ao quadrado. Com o resultado transposto, de dimensões $M \cdot N \times L$, é realizada a soma coluna a coluna, obtendo assim um vetor de $M \cdot N$ de comprimento. Concluindo, os valores desse vetor são divididos pela quantidade de arquivos Dicom utilizados e, assim como no vetor P_1 , é redefinida a forma do vetor para coincidir com a do mapa T2 gerado.

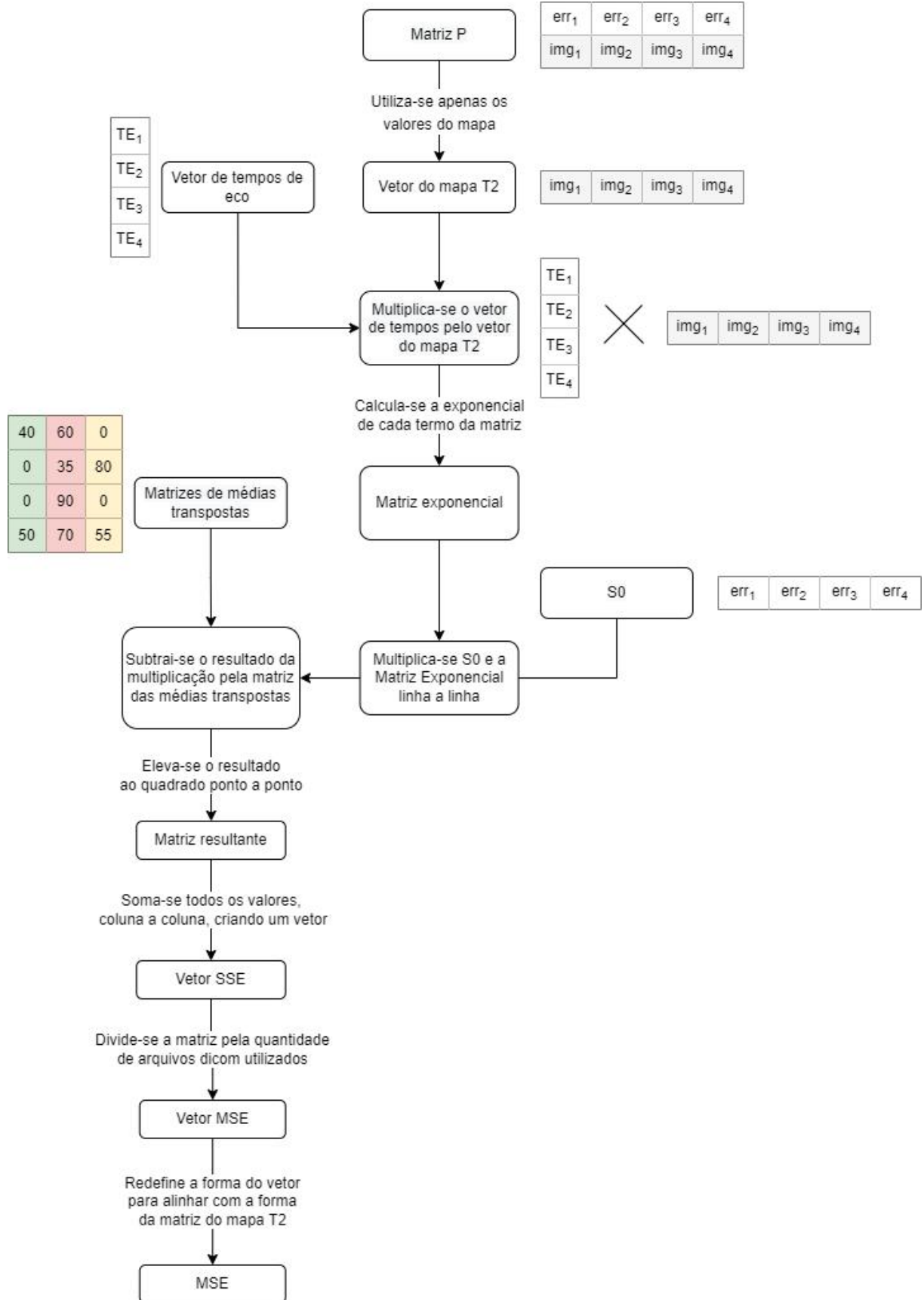
3.4.4 Testes comparativos

Com objetivo de avaliar a performance e os resultados do projeto desenvolvido, utilizou-se arquivos Dicom, com autorização do Comitê de Ética do Hospital Universitário de São Paulo (CAAE 16837213.1.000.5505/Parecer N^o 2.131.280), para dois exames de RM: um joelho esquerdo e um fígado.

A fim de se ter um comparativo dos resultados, foram utilizados os resultados obtidos através da execução do código desenvolvido pelo professor doutor Tiago Jornada Silva (JORNADA, 2017) na plataforma MATLAB.

Além disso, testes de performance também foram executados utilizando o *software* MATLAB para execução do script, e o ambiente de desenvolvimento Python, para execução do software.

Figura 19 – Fluxo para análise dos erros na geração do mapa T2.



Fonte: Autoria própria.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

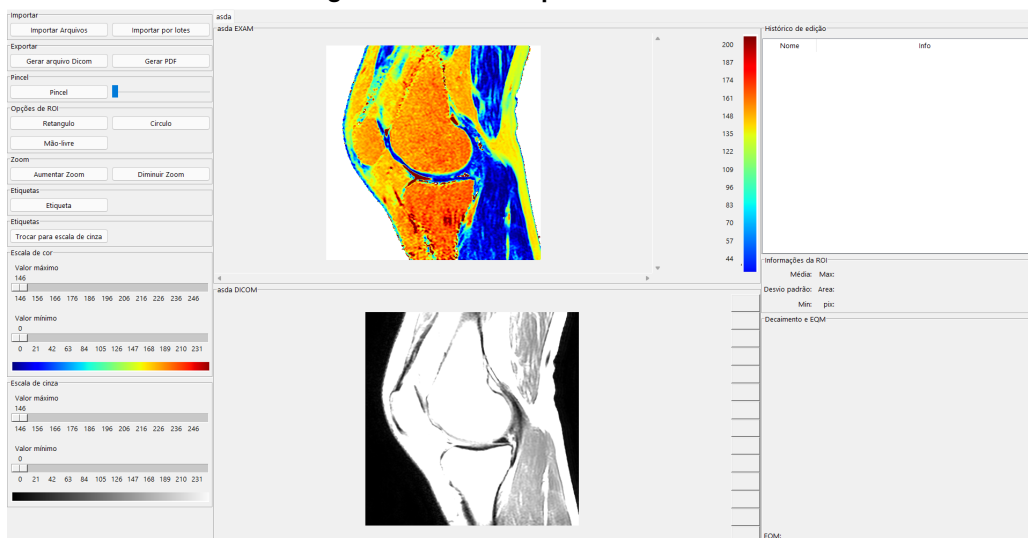
Este capítulo tem por objetivo apresentar os resultados obtidos, detalhando as funcionalidades desenvolvidas e os mapas T2 gerados. Além disso, os mapas T2 foram comparados com os mapas gerados através do código desenvolvido pelo professor doutor Tiago Jornada Silva (JORNADA, 2017) na plataforma MATLAB.

O código, assim como uma versão do software pronta para execução, estão disponíveis em um repositório na plataforma Github.

4.1 Interface

Para o desenvolvimento da interface foram usadas as bibliotecas *Tkinter* e *TtkThemes*. A *Tkinter*, além de ser nativa do Python, disponibiliza uma lista de elementos, chamados *widgets*, para geração de interfaces simples e funcionais. Já a *TtkThemes* acrescenta novas funcionalidades e um conjunto de temas visuais para a aplicação e compatíveis com a *Tkinter*. A aplicação da biblioteca *Tkinter* foi dada através de uma janela principal *mainFrame* dividida em diversas áreas delimitadas, chamadas *frames*, cada uma com seus respectivos *widgets* (botões, campos de texto, etc), conforme a Figura 20. A divisão das áreas foi definida agrupando funcionalidades e replicando interfaces populares de editores de imagem, com objetivo de trazer familiaridade para o usuário. Essas áreas são: menu de edição, visualizador de imagens, histórico de edições e análise gráfica da ROI.

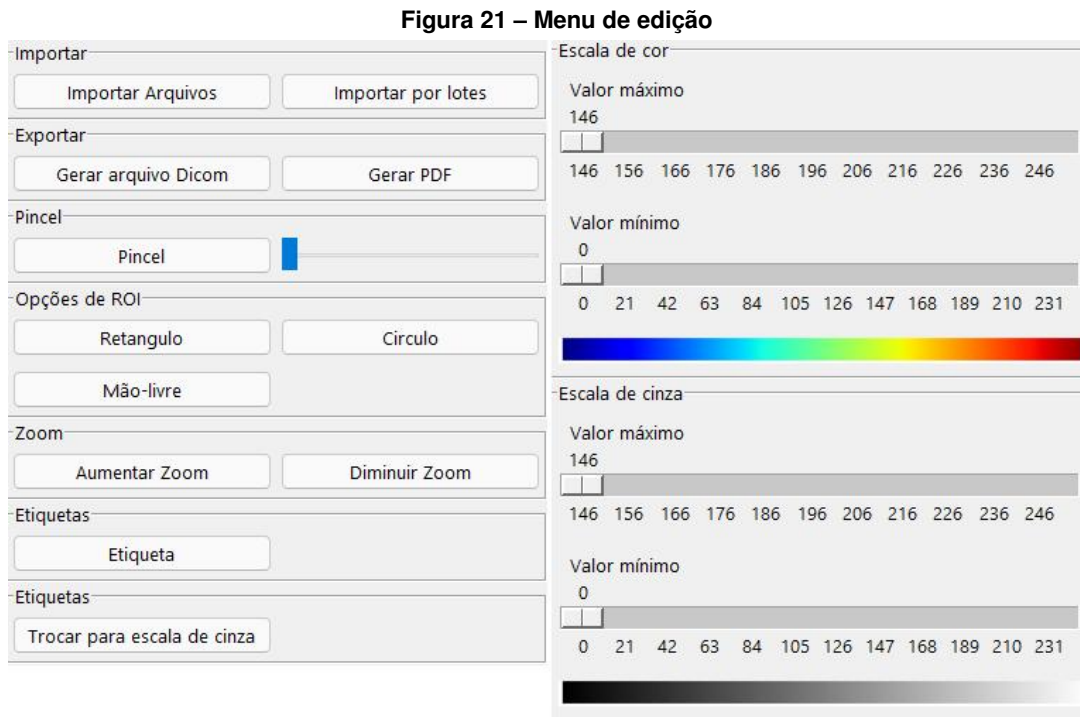
Figura 20 – Tela completa da interface.



Fonte: Autoria própria.

4.1.1 Menu de edição

O menu de seleção de edição (Figura 21) abriga todas as funcionalidades para tratamento do mapa T2 gerado e a seleção dos arquivos Dicom para sua criação.



Fonte: Autoria própria.

O menu é constituído por nove diferentes blocos, sendo eles:

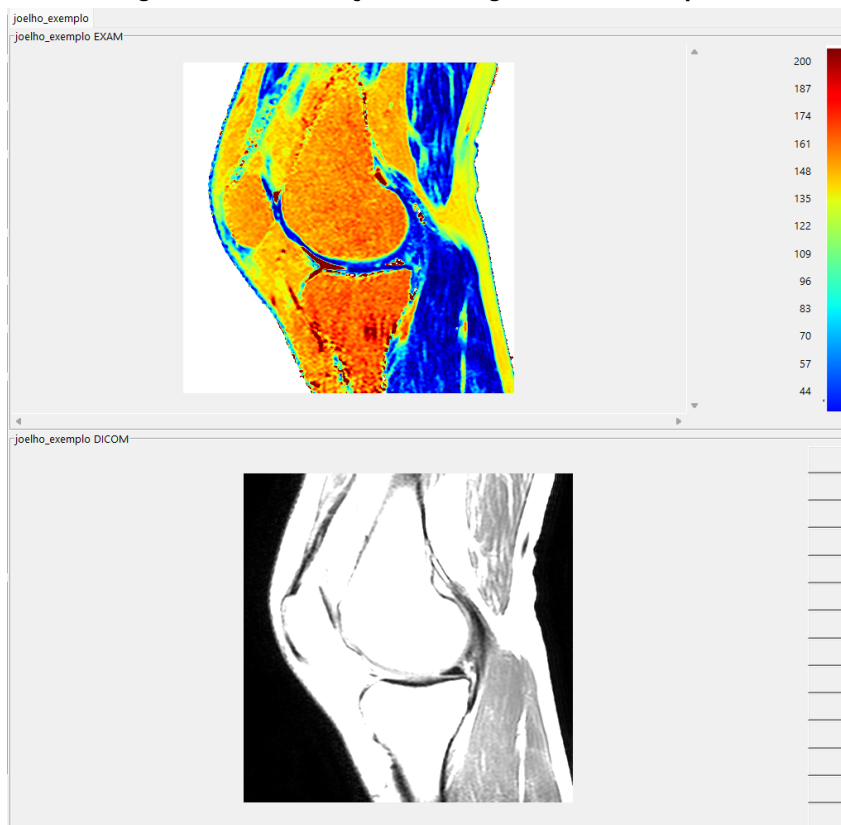
- **Importar:** Comporta as duas opções de importação dos arquivos Dicom: importação normal e importação por lotes. Na importação normal, o usuário pode selecionar uma lista de arquivos a fim de gerar o mapa T2. Já na importação por lotes, o usuário pode carregar diversos exames simultaneamente e, a partir de um valor de quantidade de ecos fornecido pelo usuário, o programa gera um mapa T2 para cada divisão.
- **Exportar:** Comporta as duas opções de exportação de informações: Gerar arquivo Dicom (gera um arquivo Dicom a partir do mapa T2) e Gerar PDF (gera um PDF com as imagens do mapa T2 e com as informações de cada uma das ROIs geradas).
- **Pincel:** Comporta a opção de desenhar sobre o exame. Quando selecionado, o menu apresenta um grupo de cores para seleção, assim como uma escala para escolha da espessura do traço.
- **ROI:** Comporta as opções de ROI (retangular, circular e livre). As diferentes opções diferem exclusivamente em morfologia, sendo todas as análises em cada opção iguais. Além das análises numéricas, a seleção da ROI também determina uma região colorida para análise quando o mapa de cores selecionado é a escala de cinza.

- **Anotações:** Comporta o botão para geração de notas sobre a imagem. As notas permitem ao usuário registrar informação textual em pontos específicos do mapa T2.
- **Mapa de cor:** Comporta o botão para troca do mapa de cor, sendo possível alterar entre escala de cinza (na qual as ROIs limitam uma região colorida) e *jet*.
- **Escala de cor T2:** Comporta as escalas para alteração dos limites dos valores de máximo e mínimo para o mapa de cor *jet*.
- **Escala de cor cinza:** Comporta as escalas para alteração dos limites dos valores de máximo e mínimo para a escala de cinza.

4.1.2 Visualizador de imagens

A área de visualização das imagens (Figura 22) apresenta as informações principais do programa, como as imagens Dicom usadas para a geração da imagem e o mapa T2 gerado, juntamente com todas as edições realizadas. A estrutura do visualizador segue o formato de abas, sendo que cada mapa T2 gerado cria uma nova aba dentro da estrutura, permitindo a manipulação de múltiplos mapas simultaneamente.

Figura 22 – Visualização das imagens Dicom e mapa T2.



Fonte: Autoria própria.

4.1.3 Histórico de edições

A janela de histórico de edições (Figura 23) apresenta a lista de edições já realizadas, em ordem cronológica. Cada elemento adicionado na edição carrega informações distintas e relevantes para facilitar a identificação do elemento (por exemplo, para a ROI, a lista carrega a informação da morfologia da ROI, assim como os pontos iniciais e finais definidos pela ferramenta), para que o usuário possa, caso seja necessário, deletar a alteração feita.

Figura 23 – Histórico de edições.

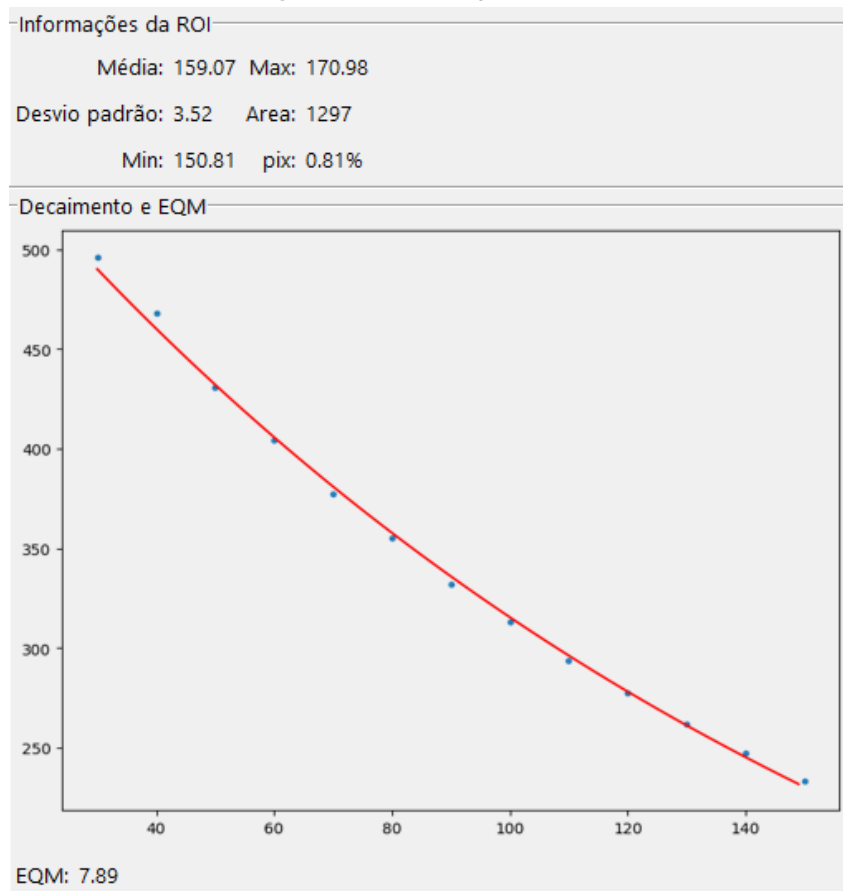
Histórico de edição	
Nome	Info
ROI	RECTANGLE(146,117),(284,273)
DrawnLines	RED(242,87)
DrawnLines	BLACK(230,72)
Tag	1-Teste 1(148,122)
Tag	2-Teste 2(301,294)
ROI	FREE(266.0,53.0),(271.0,56.0)

Fonte: Autoria própria.

4.1.4 Informações da ROI

A área de informações da ROI (Figura 24) apenas contém informações quando está selecionado uma ROI no histórico de edições. Através dessa área, é possível verificar os valores de máximo, mínimo, desvio padrão, área da ROI, assim como o percentual da imagem contida dentro da ROI (pix). Além disso, pode-se avaliar também os valores obtidos nas imagens Dicom utilizadas, presentes no gráfico como pontos azuis, e a curva de ajuste exponencial, em vermelho no gráfico. A diferença entre o ajuste e os pontos obtidos dos arquivos é apresentado como o erro quadrático médio, sob o gráfico.

Figura 24 – Informações da ROI.



Fonte: Autoria própria.

4.2 Geração de relatório

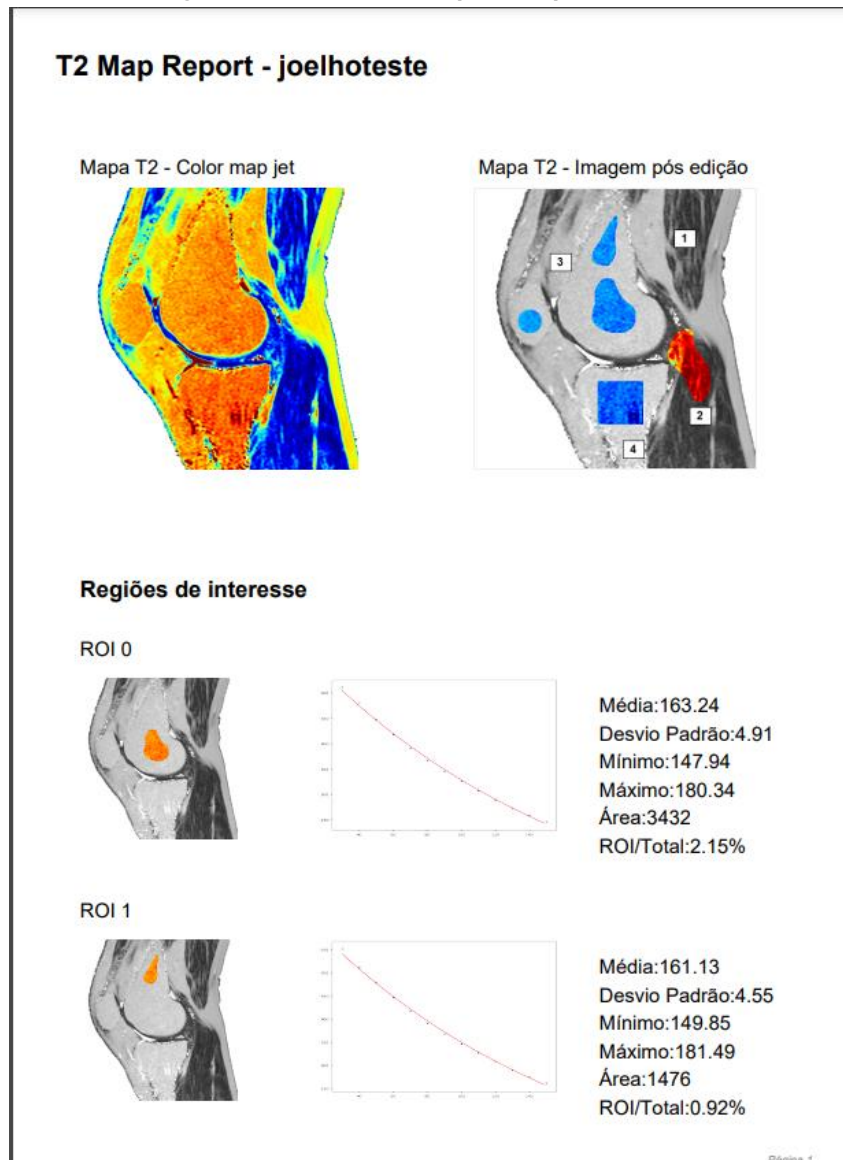
A geração da documentação pelo *software* (Figura 25) pode ser feita a qualquer momento pelo usuário. O *software* exporta a lista de ROIs e a lista de etiquetas inseridas no mapa T2, assim como as imagens com mapa de cor *jet* e o mapa em escala de cinza com as ROIs e etiquetas destacadas.

Cada ROI do documento é apresentada com o mapa em escala de cinza com apenas a ROI em questão destacada (em escala de cor *jet*), o gráfico gerado a partir da análise dos valores médios das imagens comparados com a aproximação exponencial da série e os valores de máximo, mínimo, área, percentual da imagem dentro da ROI e desvio padrão.

4.3 Mapas T2 gerados

O resultado do mapa T2 gerado pelo *software* desenvolvido para o exame do joelho esquerdo pode ser visualizado na Figura 26. O contraste da imagem é mais intenso quando colocada a escala com valores maiores (80 a 210 na figura em questão). Pode-se perceber

Figura 25 – Documento exportado pelo software.



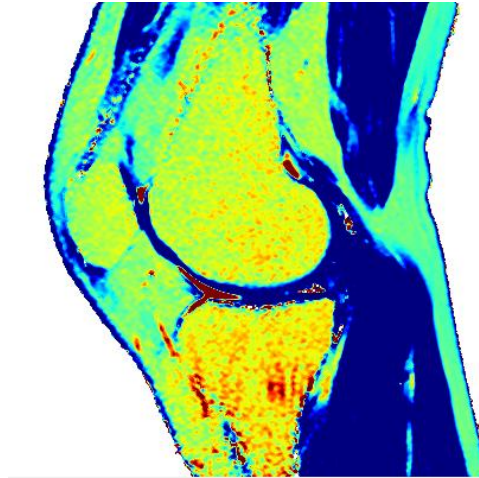
Fonte: Autoria própria.

uma coloração mais escura na região dos músculos, quando comparado à região óssea do paciente, ou até mesmo certas regiões da cartilagem.

O resultado do mapa T2 gerado pelo *software* desenvolvido para o exame do fígado pode ser visualizado na Figura 27. O contraste da imagem é mais intenso quando colocada a escala com valores menores (0 a 150 na figura em questão). Pode-se perceber uma coloração mais clara em alguns pontos do fígado, porém, em geral, a imagem resultante é bastante uniforme.

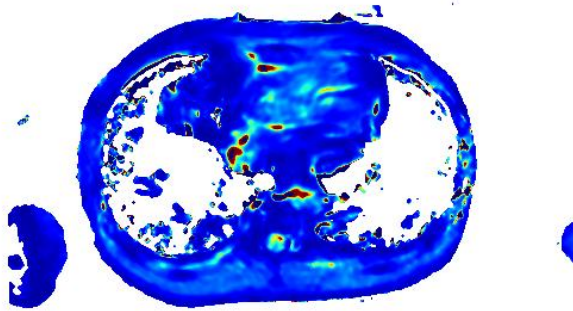
As comparações entre os valores obtidos para ambos os exames pelo *software* desenvolvido e pelo script em MATLAB (JORNADA, 2017) foram feitas pixel a pixel, avaliando dois aspectos distintos: o erro percentual em relação ao percentual de área da imagem, através do gráfico de função distribuição acumulada (FDA); e a distribuição do erro ao longo da imagem, normalizando o resultado para o uso do mapa de cor *jet*.

Figura 26 – Mapa T2 resultante de um exame RM de um joelho esquerdo com 14 ecos, limites do mapa de cor 80-210.



Fonte: Autoria própria.

Figura 27 – Mapa T2 resultante a partir de um exame RM de um fígado com 16 ecos, limites do mapa de cor 0-150.



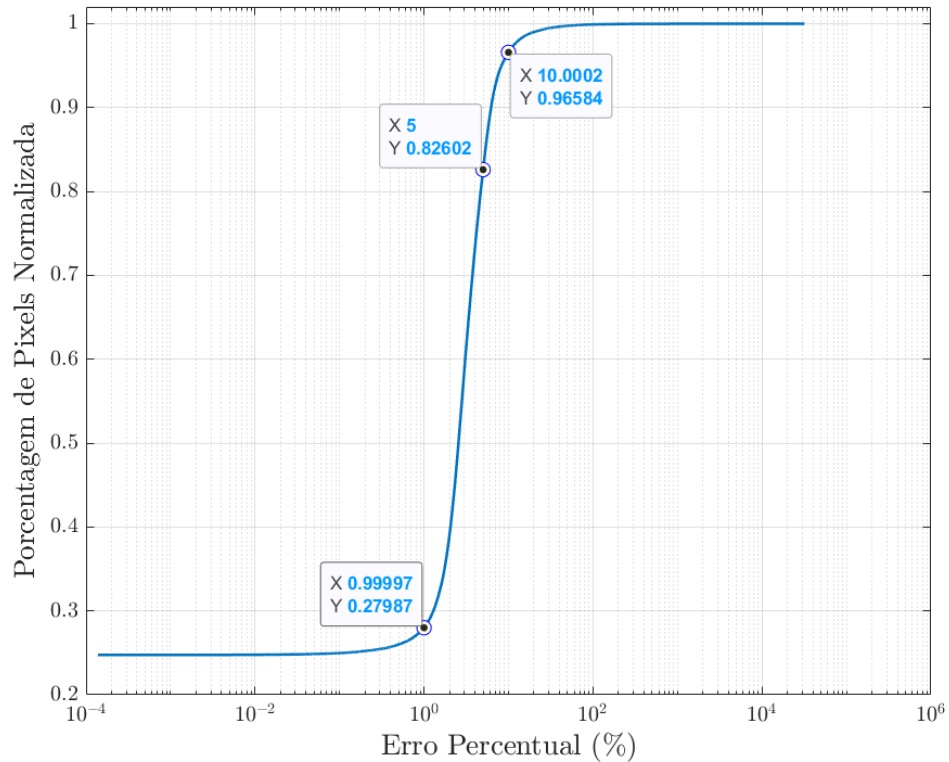
Fonte: Autoria própria.

Para a primeira análise do mapa T2 do joelho esquerdo (Figura 28), pode-se observar que grande parte da imagem (82% da área) possui um erro de até 5% em relação ao valor obtido pelo script em MATLAB (JORNADA, 2017), enquanto 96% da área possui um erro de até 10%, abrangendo assim quase toda a área da imagem. Ainda nessa análise, tem-se 27% da área com um valor de 1% de erro, devido a parte externa à região do joelho, que é uniforme e sem informações relevantes ao exame.

Para a segunda análise do mapa T2 do joelho esquerdo (Figura 29), pode-se perceber que os erros se concentram em algumas regiões, principalmente na região entre tecidos. Esse resultado é possivelmente devido a grande variação de valores na região, por se tratar de uma fronteira entre áreas com resultados distintos no exame.

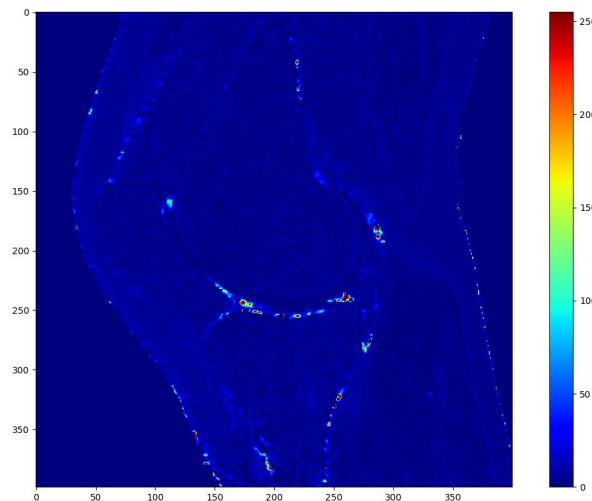
Já para a primeira análise do EQM do mapa T2 do joelho esquerdo (Figura 30), foram obtidos resultados próximos ao da análise do próprio mapa, sendo que, aproximadamente 94% da área possui um erro de até 10% e 73% da área possui um erro de até 5%. Para os erros de até 1%, tem-se apenas 2% da área.

Figura 28 – Resultado da FDA para o mapa T2 do exame do joelho.



Fonte: Autoria própria.

Figura 29 – Comparativo entre mapas T2 do exame do joelho.

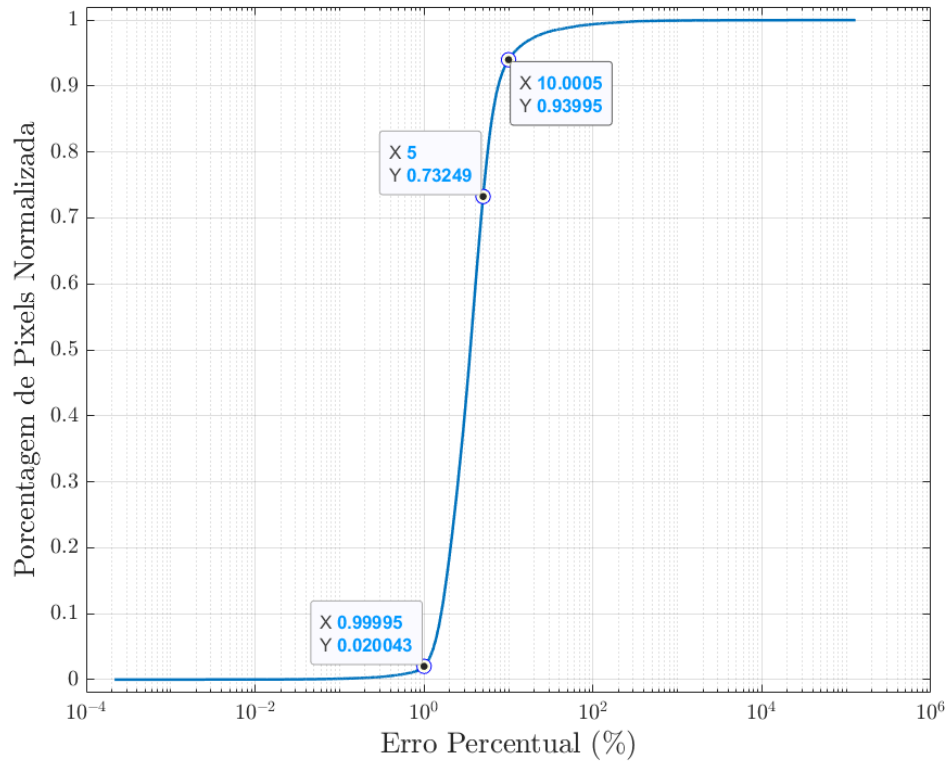


Fonte: Autoria própria.

Para a segunda análise do EQM do mapa T2 do joelho esquerdo (Figura 31), pode-se perceber que a diferença se deu principalmente na região externa à região de interesse (região do joelho).

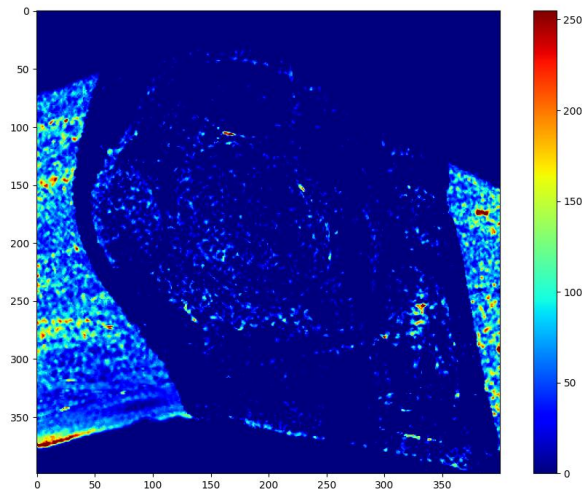
Para a primeira análise do mapa T2 do fígado (Figura 32), é perceptível o melhor desempenho em relação ao resultado obtido para o exame do joelho esquerdo, devido aos valores de percentual de erro para os diferentes percentuais de área. 82% da área possui um erro de

Figura 30 – Resultado da FDA para o EQM do mapa T2 do exame do joelho.



Fonte: Autoria própria.

Figura 31 – Comparativo entre matrizes de EQM do mapa T2 do exame do joelho.

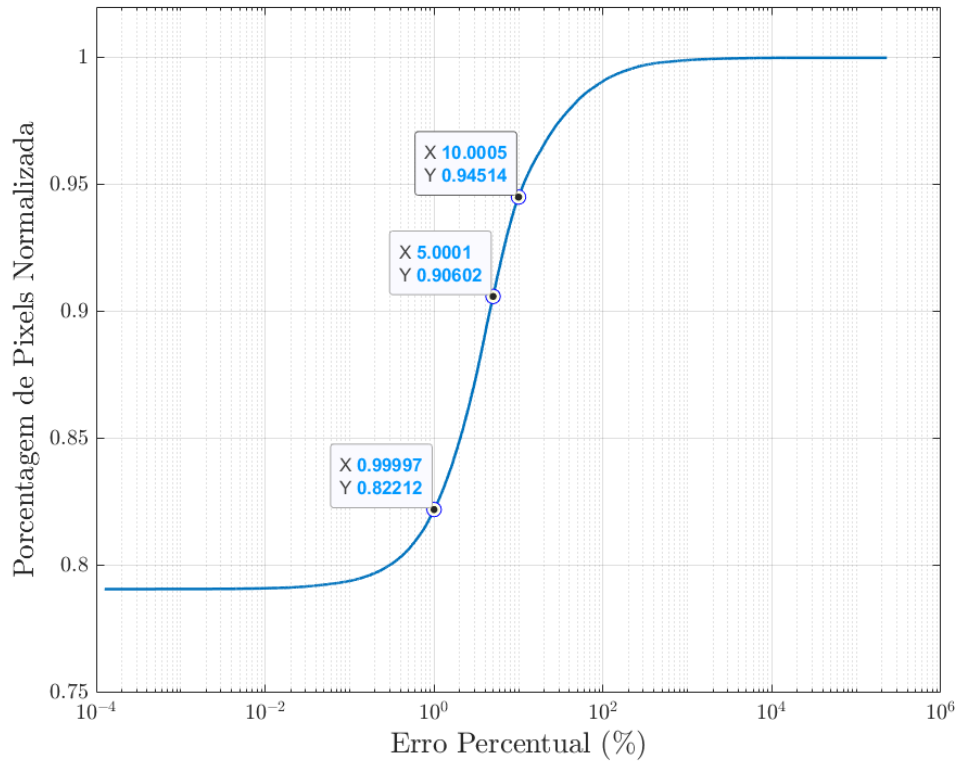


Fonte: Autoria própria.

até 1%, 90% da área possui um erro de até 5% e, por fim, 94% da área, um erro de até 10%. Esses valores revelam uma concentração de erros bastante intensos em certos pontos, porém com um bom desempenho na maior parte da imagem.

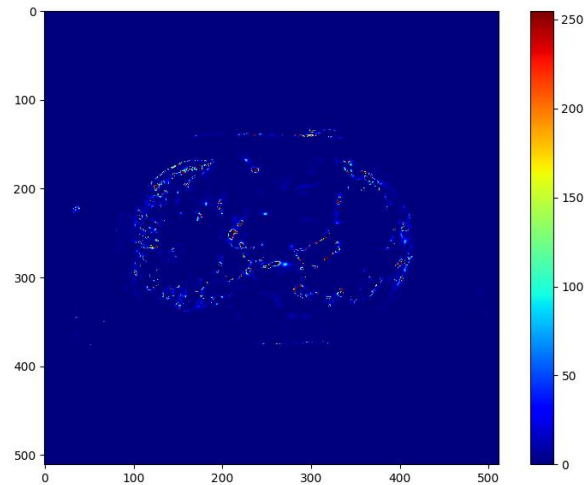
Para a segunda análise do mapa T2 do fígado (Figura 33), pode-se perceber que a diferença foi bastante semelhante à segunda análise do mapa T2 do joelho esquerdo (Figura 29), pois ambas possuem uma concentração de erros em regiões de fronteira.

Figura 32 – Resultado da FDA para o mapa T2 do exame do fígado.



Fonte: Autoria própria.

Figura 33 – Comparativo entre mapas T2 do exame do fígado.

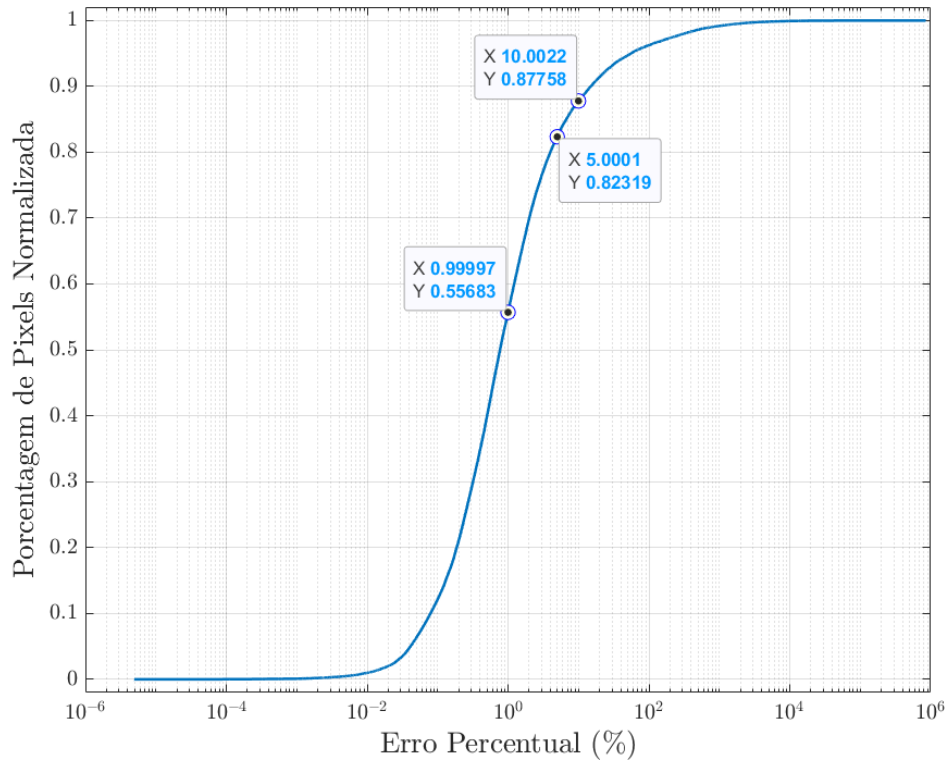


Fonte: Autoria própria.

Para a análise da FDA do EQM do exame do fígado (Figura 34), tem-se uma queda na performance, com valores de percentual de área de 56%, 82% e 87% para os erros de até 1%, 5% e 10% respectivamente.

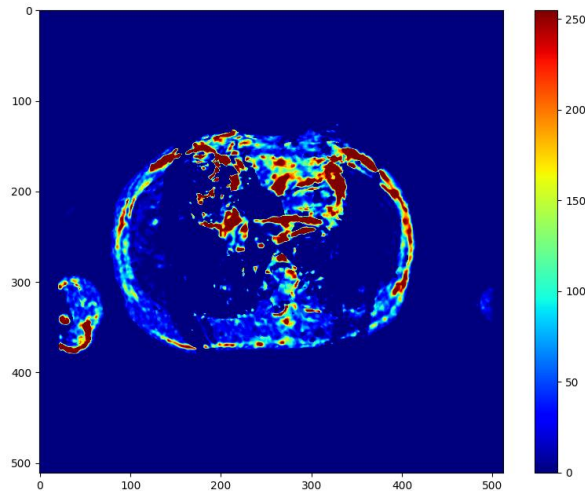
Já para a análise de imagem do EQM do fígado (Figura 35), pode-se perceber áreas de intenso erro bem maiores. Novamente, esse resultado é consequência da quantidade de valores infinitos no resultado final, que acabam por distorcer a análise EQM.

Figura 34 – Resultado da FDA para o EQM do mapa T2 do exame do fígado.



Fonte: Autoria própria.

Figura 35 – Comparativo entre matrizes de EQM do mapa T2 do exame de fígado.



Fonte: Autoria própria.

Por fim, pode-se concluir que ambas as imagens tiveram resultados satisfatórios na geração do mapa T2, apresentando poucos erros. Porém, quando analisados os resultados da análise EQM, diferenças mais acentuadas podem ser vistas entre os resultados.

Outro aspecto que pode ser observado é em relação à redução das diferenças de áreas nas análises FDA conforme aumenta-se o valor de erro observado (Tabela 1). Para os erros de até 5%, tem-se valores já bastante próximos, variando de 73% a 91% e com um desvio padrão

de 7,34%, enquanto para 10% os valores variam de 88% a 96%, com um desvio padrão de 3,59%. Já para os erros de 1%, a variação é bem mais acentuada, com uma faixa de 2% a 82%, o que leva a um desvio padrão de 34,60%.

Tabela 1 – Resultado da análise de erros - Comparação dos percentuais de erros entre diferentes exames.

Exame	Objeto de análise	Erro percentual de até	Percentual de Área
Joelho esquerdo	Mapa T2	1% 5% 10%	28% 82% 96%
	Matriz EQM	1% 5% 10%	2% 73% 94%
Fígado	Mapa T2	1% 5% 10%	82% 91% 95%
	Matriz EQM	1% 5% 10%	56% 82% 88%

Fonte: Autoria própria.

4.4 Performance

Analisando os resultados obtidos (Tabela 2), é possível verificar um ganho de performance de 96,5% quando comparado com os resultados do código em MATLAB.

Tabela 2 – Resultado dos testes de performance - Comparação do benchmark e projeto.

	Geração de mapa pelo script MATLAB (s)	Geração de mapa <i>software</i> desenvolvido (s)
1	10,88	0,55
2	10,14	0,35
3	9,86	0,33
4	11,18	0,34
5	12,06	0,33
Média	10,82	0,38
Desvio Padrão	0,87	0,09

Fonte: Autoria própria.

A partir desses valores pode-se concluir que a aplicação de uma solução mais otimizada no que diz respeito a uso de cálculo matricial impacta diretamente no tempo de execução do código. A solução desenvolvida utiliza de diversos métodos presentes em bibliotecas com otimizações desenvolvidas em linguagem de programação C para apresentar um ganho de tempo de processamento, aspecto esse não apreciado no código em MATLAB, que preza por uma execução mais transparente e de fácil entendimento, utilizando no local de cálculos matriciais laços para iteração sobre os elementos presentes em cada vetor e matriz.

5 CONCLUSÃO

Este documento apresentou o desenvolvimento completo de um software de criação e edição de mapas T2 de ressonância magnética. O desenvolvimento foi feito inteiramente em Python, utilizando uma série de bibliotecas de livre acesso para o desenvolvimento da interface, processamento de imagem e geração de documentação.

No que diz respeito aos resultados, pode-se dividir em diferentes tópicos:

- A interface teve um resultado bastante satisfatório. Todos os elementos propostos foram desenvolvidos e operam de maneira simples e efetiva. As ferramentas de edição do mapa funcionam de maneira consistente, não sendo encontrados erros de funcionamento na versão final do software.
- O processamento para geração do mapa T2 teve resultados bastante semelhantes ao comparado, pixel a pixel, com o mapa obtido pelo script em MATLAB. Além disso, o software apresentou uma performance bastante superior em relação ao MATLAB, sendo quase 30 vezes mais rápido.
- A análise de erros apresentou valores bastante distintos para cada exame testado, resultando em um aspecto não tão satisfatório na versão final do software.

5.1 Dificuldades Encontradas

Dentre as dificuldades encontradas, pode-se listar a natureza clínica do software, a dificuldade de comparar resultados e a complexidade da interface.

O entendimento do procedimento clínico e as aplicações da análise T2 acabaram por se tornar aspectos desafiadores, muito disso devido à complexidade do tema e a distância entre as áreas de atuação (medicina e engenharia).

Além disso, a limitada lista de exames disponíveis, somada à pouca variedade de softwares para comparação (apenas o script em MATLAB), vieram a dificultar as análises dos resultados intermediários, tanto durante o desenvolvimento quanto durante o refino da otimização para geração de mapas.

Concluindo, a interface trouxe consigo um desafio de programação distinto ao da geração de mapas, pois a necessidade de responder a eventos causados pelo usuário de maneira rápida e clara exigiu um aprofundamento no funcionamento das bibliotecas envolvidas.

5.2 Trabalhos Futuros

Uma possível sugestão de trabalho futuro seria a melhoria do sistema para ser executado em um modelo *web*. Além da maior facilidade de uso e de acesso, um sistema com servidor

e cliente poderia armazenar análises e, a longo prazo, utilizar essa base de dados para treinar uma inteligência artificial para auxiliar profissionais em análises futuras.

REFERÊNCIAS

- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000. Acesso em: 02 de março de 2022.
- FEGER, J. T2 mapping - myocardium. **Radiopaedia**, 11 2021. Acesso em: 02 de março de 2022.
- FENNIK, M. **PyPDF 1.13: A Pure-Python library built as a PDF toolkit**. [s.n.], 2010. Disponível em: <https://pypi.org/project/pyPdf/#description>. Acesso em: 02 de março de 2022.
- GEUNS, R. J. M. *et al.* Basic principles of magnetic resonance imaging. **Progress in Cardiovascular Diseases**, v. 42, n. 2, p. 149–156, 1999. ISSN 0033-0620. Advances in Coronary Imaging. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0033062099700149>. Acesso em: 02 de março de 2022.
- HAGE, M. C. F. N. S.; IWASAKI, M. Imagem por ressonância magnética: princípios básicos. **Ciência Rural**, 2009. Acesso em: 02 de março de 2022.
- HARRIS, C. R. *et al.* Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>. Acesso em: 02 de março de 2022.
- HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Acesso em: 02 de março de 2022.
- JONES, J.; BABA, Y. T2 weighted image. **Radiopaedia**, 05 2021. Acesso em: 02 de março de 2022.
- JONES, J.; HAQUIMI, A. T1 weighted image. **Radiopaedia**, 02 2021. Acesso em: 02 de março de 2022.
- JORNADA, T. da S. **Desenvolvimento de uma ferramenta para auxílio na avaliação de lesões cartilaginosas do joelho**. 2017. 6,7,11,12 f. Dissertação (Doutorado em Radiologia Clínica e Ciências Radiológicas) — Universidade Federal de São Paulo, São Paulo, 2017.
- KINSLER, P. Faraday's law and magnetic induction: Cause and effect, experiment and theory. **Physics**, v. 2, p. 148–161, 05 2020. Acesso em: 02 de março de 2022.
- LARSSON, E. **Evaluation of the Dual-Modal usage of contrast agents by means of Synchrotron X-ray Computed Microtomography and Magnetic Resonance Imaging using Macrophages loaded with Barium Sulfate and Gadolinium Nanoparticles for Detection and Monitoring in Animal Disease Models**. 12 2015. Tese (Doutorado), 12 2015. Acesso em: 02 de março de 2022.
- MASON, D. *et al.* **pydicom/pydicom: pydicom 2.2.2**. Zenodo, 2021. Disponível em: <https://doi.org/10.5281/zenodo.5543955>. Acesso em: 02 de março de 2022.
- MATLAB. **version 7.10.0 (R2010a)**. Natick, Massachusetts: The MathWorks Inc., 2010.
- PYTHON SOFTWARE FOUNDATION. **tkinter — Python interface to Tcl/Tk**. [S.l.], 2022. Disponível em: <https://docs.python.org/3/library/tkinter.html#module-tkinter>. Acesso em: 02 de março de 2022.

STIKOV, N. **Clinical Applications of MR Relaxometry**. 2012. Disponível em: https://www.bic.mni.mcgill.ca/uploads/Seminars/2012-04-02_-_Clinical_MR_Relaxometry_-_Nikola_Stikov.pdf. Acesso em: 02 de março de 2022.

VILLAFANA, T. Fundamental physics of magnetic resonance imaging. **Radiologic clinics of North America**, v. 26, n. 4, p. 701—715, July 1988. ISSN 0033-8389. Disponível em: <http://europepmc.org/abstract/MED/3380941>.

WESTBROOK, C.; ROTH, C. K.; TALBOT, J. **MRI in Practice**. Chichester, Reino Unido: Wiley-Blackwell, 2011.