



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

**Mosaic: Uma Ferramenta de Custo Acessível
para Digitalização de Lâminas Histopatológicas
Adequada ao Padrão DICOM**

Autor: Wesley Pereira Araujo
Orientador: Profa. Dra. Milene Serrano

Brasília, DF
2021



Wesley Pereira Araujo

**Mosaic: Uma Ferramenta de Custo Acessível para
Digitalização de Lâminas Histopatológicas Adequada ao
Padrão DICOM**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2021

Wesley Pereira Araujo

Mosaic: Uma Ferramenta de Custo Acessível para Digitalização de Lâminas Histopatológicas Adequada ao Padrão DICOM/ Wesley Pereira Araujo. – Brasília, DF, 2021-

135 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2021.

1. Palavra-chave01. 2. Palavra-chave02. I. Profa. Dra. Milene Serrano. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Mosaic: Uma Ferramenta de Custo Acessível para Digitalização de Lâminas Histopatológicas Adequada ao Padrão DICOM

CDU 02:141:005.6

Wesley Pereira Araujo

Mosaic: Uma Ferramenta de Custo Acessível para Digitalização de Lâminas Histopatológicas Adequada ao Padrão DICOM

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 21 de Maio de 2021:

Profa. Dra. Milene Serrano
Orientadora

Prof. Dr. Maurício Serrano
Coorientador

Prof. Dr. Ronni Geraldo Gomes de Amorim
Examinador

Profa. Dra. Suélia de Siqueira Rodrigues Fleury Rosa
Examinadora

Brasília, DF
2021

Dedico este trabalho aos que permanecem com a mesma curiosidade de quando era uma criança, que não possuem medo de errar na busca pelo conhecimento e não se conformam com o status quo.

Agradecimentos

Meus agradecimentos aos meus pais e meus irmãos pelo apoio em diversos momentos e também aos meus amigos Lucas Pereira e Lucas Costa que compartilharam essa jornada acadêmica.

Aos meus orientadores Milene Serrano e Maurício Serrano por terem compartilhado conhecimento, confiado em mim e me ajudado a enxergar de maneira mais ampla a Engenharia de Software.

Agradeço também ao professor Ronni Amorim pelo incentivo, desde o início da graduação, a utilizar a curiosidade e a criatividade para aprender e desenvolver ideias, além do auxílio em explorar a matemática e seu entendimento.

Aos médico patologista Thiago David, pela abertura da especialidade em anatomia patológica e compartilhamento de ideias, que possibilitou o surgimento e amadurecimento da proposta para esse trabalho.

“Todas as crianças crescem, menos uma.”

J.M. Barrie, Peter Pan

Resumo

O aumento da capacidade computacional, o ganho de armazenamento em nuvem, o aumento da velocidade de transferência de dados e os avanços em tecnologias de software possibilitam à patologia usufruir desses recursos para novas possibilidades de diagnóstico, educação e pesquisa. O surgimento da tecnologia *Whole Slide Imaging (WSI)* auxiliou na digitalização completa de lâminas histopatológicas. O custo de aquisição de equipamentos para digitalização dessas lâminas é estimado em torno de \$ 50.000 dólares, ou mais, desestimulando seu uso para implantação da patologia digital. Além do elevado custo para aquisição de um *scanner* WSI, há o impeditivo de interoperabilidade entre os equipamentos dos fabricantes, pois possuem protocolos próprios para armazenamento e comunicação dos dados. Na tentativa de estimular a a patologia digital e patologia computacional, esse Trabalho de Conclusão de Curso desenvolveu uma solução em software de custo acessível, o qual segue as diretrizes do padrão DICOM, e possui como suporte um aplicativo para *smartphone* iOS[®] e uma aplicação Web para visualização dos dados.

Palavras-chaves: Patologia Digital. Patologia Computacional. *Whole Slide Imaging*. WSI. Anatomia Patológica. DICOM.

Abstract

Increased computing power, cloud storage gain, increased data transfer speed, and advances in software technologies enable pathology to leverage these capabilities for new diagnostic, educational, and research possibilities. The emergence of Whole Slide Imaging (WSI) technology aided in the complete digitization of histopathological slides. The cost of purchasing equipment for scanning these slides is estimated at around \$ 50,000 dollars or more, discouraging their use for the implantation of the digital pathology. In addition to the high cost of acquiring a WSI scanner, there is an impediment to interoperability between manufacturers' equipment because they have their own protocols for storing and communicating the data.

In an attempt to stimulate the digital pathology and computational pathology, this final paper developed an affordable software solution that follows the DICOM standard guidelines and is supported by an application for *smartphone* iOS[®] and a Web application for visualizing the data.

Key-words: Digital Pathology. Computational Pathology. Whole Slide Imaging. WSI. Virtual Slides. Digital Slides. Anatomic Pathology. DICOM.

Lista de ilustrações

Figura 1 – Exemplo de uma lâmina histopatológica digitalizada de um carcinoma ductal ampliada em 400x.	34
Figura 2 – Diferença no mapeamento da imagem entre <i>Tile-Based Scanning</i> e <i>Line-Based Scanning</i>	36
Figura 3 – Representação de informações na patologia digital no DICOM	39
Figura 4 – Imagem binária	40
Figura 5 – Imagem em escala de cinza	40
Figura 6 – Imagem colorida com as três camadas RGB.	41
Figura 7 – Algoritmo de <i>Image Stitching</i> aplicado para gerar um panorama de Marte pelo <i>Perseverance</i>	42
Figura 8 – Segmento de 16 pontos adjacentes de teste do ponto p para classificar se é um <i>corner</i>	44
Figura 9 – Tipos de Pesquisa	53
Figura 10 – Fluxo de Atividades do Trabalho de Conclusão de Curso (TCC)	55
Figura 11 – Processo Scrum Solo, uma adaptação Scrum (PAGOTTO et al., 2016)	57
Figura 12 – Processo de desenvolvimento adaptado do Scrum e Scrum Solo.	58
Figura 13 – Processo de análise de resultados	60
Figura 14 – Lâmina histopatológica de um Carcinoma Ductal, ampliada em 400x, e utilizada no estudo comparativo.	64
Figura 15 – Lâmina histopatológica de um Carcinoma Ductal gerada a partir dos fragmentos	64
Figura 16 – Nível 1 (Contexto)	67
Figura 17 – Nível 2 (Container)	68
Figura 18 – Nível 3 (Componente)	69
Figura 19 – Representação <i>DICOM Data Model SCImage</i> . (1) Representação geral de um objeto DICOM. (2) Representação de como os relacionamentos foram utilizados no contexto do projeto Mosaic. As entidades foram mantidas em inglês a nível de código, bem como as entidades, mantendo o idioma na Figura.	70
Figura 20 – Fluxo do módulo de criação de imagem.	71
Figura 21 – Fluxo JWT no sistema de <i>software</i>	73
Figura 22 – Telas de autenticação no aplicativo iOS. (1) Tela de <i>login</i> . (2) Tela de cadastro.	74

Figura 23 – Telas de <i>upload</i> de fragmentos de imagem para o <i>Stitching</i> . (1) Tela de seleção de fragmentos de imagens. (2) Escolha da fonte dos fragmentos de imagem. (3) Escolha de fragmento de imagem a partir do álbum de fotos sendo editada para formato quadrado. (4) Tela de seleção de fragmentos de imagem com todos já selecionados.	75
Figura 24 – Tela de preenchimento dos metadados DICOM. (1) Tela sem preenchimento dos dados, impossibilitando prosseguir com o <i>upload</i> das informações. (2) Tela com os dados preenchidos, possibilitando prosseguir com o <i>upload dos dados</i>	76
Figura 25 – <i>Studies</i> no aplicativo iOS. (1) Tela de <i>studies</i> .(2) Tela do detalhe de um <i>study</i>	77
Figura 26 – Amostras utilizadas para os cenários de uso. (1) Amostra de nove imagens aplanadas em 200x. (2) Amostra da mesma lâmina (1), ampliada em 400x. (3) Amostra de lâmina diferente das anteriores, ampliada em 400x.	80
Figura 27 – Ocular do microscópio utilizado com sua marcação de ampliação e número de campo.	81
Figura 28 – Imagem composta a partir de dois fragmentos em formato quadrado.	82
Figura 29 – Ocular do microscópio utilizado com sua marcação de ampliação e número de campo.	83
Figura 30 – Quantidade de imagens para digitalizar uma lâmina completa, desconsiderando sobreposições. (1) representação de varredura de captura de imagens com ampliação em 200x. (2) representação de lâmina com varredura para captura de imagens em ampliação de 400x.	84
Figura 31 – Fluxo de captura de imagens da primeira amostra para o primeiro cenário de uso.	86
Figura 32 – Fluxo de captura dos fragmentos de imagem de uma área de duas lâminas diferentes. (1) Primeira amostra de imagens da área a ser realizada o <i>stitching</i> . (2) Segunda amostra de imagens da área a ser realizada o <i>stitching</i>	87
Figura 33 – Comparação do <i>stitching</i> gerado pelo ORB e SURF de fragmentos ampliados em 400x. (1) <i>Stitching</i> gerado pelo algoritmo ORB. (2) <i>Stitching</i> gerado pelo algoritmo SURF.	87
Figura 34 – Fragmentos de imagem e seus histogramas. (1) Fragmentos das imagens da área a ser realizada o <i>stitching</i> . (2) Histograma de cada imagem em sua respectiva ordem.	88
Figura 35 – Histogramas dos fragmentos de imagem da primeira amostra ampliada para melhor visualização.	89
Figura 36 – Saída de verificação de arquivo DICOM pelo <i>DICOM Validator</i>	90

Figura 37 – Saída de verificação de arquivo DICOM pelo <i>DICOM Validator</i>	90
Figura 38 – Adaptador de acoplamento entre <i>smartphone</i> e ocular do microscópio.	91
Figura 39 – Adaptador acoplado na ocular do microscópio em captura de imagem de uma área da lâmina.	92
Figura 40 – Saída de verificação de arquivo DICOM pelo <i>DICOM Validator</i>	92
Figura 41 – Tela de <i>upload</i> de arquivo DICOM do serviço <i>PostDICOM</i>	94
Figura 42 – Tela de exibição dos arquivos DICOM carregados no <i>PostDICOM</i>	95
Figura 43 – Tela de exibição dos metadados e imagem do arquivo DICOM selecionado.	96
Figura 44 – Arquivos locais gerados pela API. Além da imagem gerada pelo <i>stitching</i> das amostras e do arquivo DICOM, há a versão reduzida do arquivo DICOM para poder ser enviado ao <i>Cloudinary</i>	97
Figura 45 – Arquivos enviados ao <i>Cloudinary</i> acessados pela plataforma. Os nomes dos arquivos na nuvem sofrem alteração pois usou-se o comportamento padrão do <i>Cloudinary</i> em criar próprias <i>hashs</i>	98
Figura 46 – Organização dos diretórios dos arquivos enviados. Cada diretório agrupa a imagem gerada e o arquivo dicom, identificado com a hash criada pela API.	98
Figura 47 – <i>Trello</i> sendo utilizado como <i>kanban</i> para gerenciar o progresso do de- senvolvimento dos requisitos.	101
Figura 48 – <i>Trello</i> sendo utilizado como <i>kanban</i> para gerenciar <i>status</i> dos requisitos.	102
Figura 49 – Fluxograma do algoritmo de <i>Image Stitching</i>	117
Figura 50 – Notação <i>C4 Model</i> utilizando o software <i>Draw.io</i>	119
Figura 51 – Protótipos criados no <i>Marvel</i> e gerados a partir do desenvolvimento como artefatos do <i>Scrum Solo</i>	122
Figura 52 – Questionário aplicado ao especialista e suas respostas.	123
Figura 53 – Proposta de Nível 1 (Contexto).	127
Figura 54 – Proposta de Nível 2 (Container).	128
Figura 55 – Proposta de Nível 3 (Componente)	129
Figura 56 – Proposta de Nível 3 (Código). (1) Representação DICOM <i>Data Model</i> . (2) Objetos definidos na <i>VL Whole Slide Microscopy Image</i>	130
Figura 57 – Modelo DICOM (DICOM, 2019c)	133
Figura 58 – Modelo DICOM <i>Information Model</i>	135

Lista de tabelas

Tabela 1 – Ferramentas de suporte tecnológico.	51
Tabela 2 – Cronograma de Atividades para o TCC 1.	61
Tabela 3 – Cronograma de Atividades para o TCC 2.	61
Tabela 4 – <i>Endpoints</i> que a aplicação API dispõe para operacionalização de aplicações cliente.	72
Tabela 5 – Histórias de Usuário com suas respectivas tarefas e pontuação	99
Tabela 6 – Critérios de aceitação das Histórias de Usuário.	100

Lista de abreviaturas e siglas

WSI	<i>Whole Slide Imaging</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
API	<i>Application Programming Interface</i>
WADO-RS	<i>Web Access to DICOM Objects RESTful</i>
HE	Hematoxilina e Eosina
QUIDO-RS	<i>Query based on ID for DICOM Objects RESTful</i>
STOW-RS	<i>Store Over the Web RESTful</i>
IOD	<i>Information Object Definition</i>
UML	<i>Unified Modeling Language</i>
RGB	<i>Red, Green, Blue</i>
LIS	<i>Laboratory Information System</i>
BPMN	<i>Business Process Model and Notation</i>
IDE	<i>Integrated Development Environment</i>
UnB	Universidade de Brasília
FGA	Faculdade UnB Gama
FAST	Features from Accelerated Segment Test
BRIEF	<i>Binary Robust Independent Elementary Features</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SURF	<i>Speeded Up Robust Features</i>
ORB	<i>Oriented FAST and Rotated BRIEF</i>
VL	<i>Visible Light</i>
CDN	<i>Content Delivery Network</i>
TCC1	Trabalho de Conclusão de Curso 1

TCC2 Trabalho de Conclusão de Curso 2

SC Image *Secondary Capture Image*

UID *Universal Identifier*

Lista de símbolos

\in	Pertence
τ	Representação de uma imagem como matriz
C	Centróide de uma imagem
O	Ponto central de um <i>corner</i>
θ	Ângulo theta formado a partir do eixo horizontal
S	Matriz de descritores formado pelo algoritmo BRIEF
H	Matriz de homografia
I	Intensidade em um ponto da imagem
p	<i>Patch</i> de uma imagem

Sumário

1	INTRODUÇÃO	29
1.1	Contextualização	29
1.2	Questão de Desenvolvimento	31
1.3	Justificativa	31
1.4	Objetivo Geral	31
1.4.1	Objetivos Específicos	31
1.5	Organização do Documento	32
2	REFERENCIAL TEÓRICO	33
2.1	Anatomia Patológica	33
2.1.1	Lâminas Histopatológicas	33
2.1.2	Telepatologia	33
2.1.3	Patologia Digital	34
2.1.4	Patologia Computacional	35
2.1.5	<i>Whole Slide Imaging</i>	35
2.2	DICOM	36
2.2.0.1	Definição de Objetos no DICOM	37
2.2.0.2	DICOMWeb™	37
2.2.0.3	DICOM na Patologia Digital	38
2.3	Adaptadores de Smartphone para Microscópios	39
2.4	Processamento de Imagem	39
2.4.1	Representação de Imagem Digital	39
2.4.1.1	Imagem Binária	39
2.4.1.2	Imagem em Escala de Cinza	40
2.4.1.3	Imagens Coloridas - RGB	40
2.4.2	Intensidade de Imagens	41
2.4.3	<i>Image Stitching</i>	41
2.4.4	Registro de Imagem	41
2.4.4.1	Deteção e Descrição de Características	42
2.4.4.2	<i>Features from Accelerated Segment Test (FAST)</i>	43
2.4.4.3	<i>Binary Robust Independent Elementary Features (BRIEF)</i>	44
2.4.4.4	<i>Oriented FAST and Rotated BRIEF (ORB)</i>	45
2.4.4.5	Matriz de Homografia	46
2.4.4.6	Composição da Imagem	46
2.5	Resumo do Capítulo	47

3	SUPORE TECNOLÓGICO	49
3.1	<i>Xcode 12.4</i>	49
3.2	<i>OpenCV 3.4.5</i>	49
3.3	<i>Git 2.14</i>	49
3.4	<i>GitHub</i>	49
3.5	<i>PostDICOM</i>	50
3.6	<i>Pydicom 2.1.2</i>	50
3.7	<i>DICOM Validator</i>	50
3.8	<i>Draw.io</i>	50
3.9	Resumo do Capítulo	50
4	METODOLOGIA	53
4.1	Classificação de Pesquisa	53
4.1.1	Abordagem	53
4.1.2	Natureza	54
4.1.3	Objetivos	54
4.1.4	Procedimentos Técnicos	54
4.2	Fluxo de Atividades	55
4.3	Metodologia de Desenvolvimento	56
4.4	Metodologia de Coleta e Análise de Resultados	59
4.5	Cronograma	60
4.6	Resumo do Capítulo	61
5	PROVA DE CONCEITO	63
5.1	<i>Image Stitching</i>	63
5.2	Resumo do Capítulo	64
6	MOSAIC	65
6.1	Contextualização	65
6.2	Arquitetura do sistema	66
6.2.1	Contexto	67
6.2.2	<i>Container</i>	67
6.2.3	Componente	68
6.2.4	Código	70
6.3	Módulo de <i>Stitching</i>	70
6.4	Módulo de Criação Imagem	71
6.4.1	Aplicação API	71
6.5	Segurança	72
6.6	Aplicativo iOS	73
6.7	Configuração e Integração das Tecnologias	77

6.7.1	Flask Rest API	77
6.7.2	Aplicativo iOS	78
6.8	Resumo do Capítulo	78
7	ANÁLISE DOS RESULTADOS	79
7.1	Cenários de Uso	79
7.1.1	Cenário de Uso 1	80
7.1.1.1	Coleta e Análise de Dados	81
7.1.1.2	Plano de Ação	81
7.1.1.3	Resultados da Ação	82
7.1.2	Cenário de Uso 2	83
7.1.2.1	Coleta e Análise de Dados	84
7.1.2.2	Plano de Ação	84
7.1.2.3	Resultados da Ação	85
7.1.3	Cenário de Uso 3	85
7.1.3.1	Coleta e Análise de Dados	85
7.1.3.2	Plano de Ação	85
7.1.3.3	Resultados da Ação	86
7.1.4	Cenário de Uso 4	88
7.1.4.1	Coleta e Análise de Dados	89
7.1.4.2	Plano de Ação	89
7.1.4.3	Divulgação dos Resultados	91
7.1.5	Cenário de Uso 5	91
7.1.5.1	Coleta e Análise de Dados	91
7.1.6	Cenário de Uso 6	92
7.1.6.1	Plano de Ação	93
7.1.6.2	Coleta de Análise de Dados	93
7.1.7	Cenário de Uso 7	97
7.1.7.1	Coleta e Análise de Dados	97
7.1.7.2	Plano de Ação	97
7.1.7.3	Resultados da Ação	97
7.2	Engenharia de produto	98
7.3	Validação pelo Especialista	102
7.4	Resumo do Capítulo	103
8	CONCLUSÃO	107
8.1	Trabalhos Futuros	108
	REFERÊNCIAS	109

APÊNDICES	115
APÊNDICE A – FLUXOGRAMA DO ALGORITMO DE IMAGE STITCHING	117
APÊNDICE B – NOTAÇÃO <i>C4 MODEL</i>	119
APÊNDICE C – PROTÓTIPO DO MOSAIC	121
APÊNDICE D – ANÁLISE DOS RESULTADOS	123
APÊNDICE E – ARTEFATOS DE PROPOSTA	127
ANEXOS	131
ANEXO A – MODELO ENTIDADE RELACIONAMENTO DICOM PARA ESPÉCIMES	133
ANEXO B – DICOM <i>INFORMATION MODEL</i>	135

1 Introdução

Pretendeu-se, com esse trabalho, contribuir com o domínio da saúde, mais especificamente com a área de patologia. Trata-se de uma área bastante atuante em diagnósticos médicos. Portanto, também propícia à aplicação de soluções computacionais que tanto agilizem os diagnósticos, quanto busquem por diagnósticos mais precisos.

No intuito de conferir informações que permitam definir o escopo de atuação desse trabalho, nesse capítulo, têm-se: a contextualização, com foco em particularidades da área de patologia; a questão de desenvolvimento, pontuando exatamente o que se pretende responder ao final desse trabalho; as justificativas para realização do mesmo, e os objetivos geral e específicos, visando o cumprimento do trabalho como um todo. Por fim, tem-se a organização dessa monografia.

1.1 Contextualização

O *College of American Pathologists* estima que entre 60% a 70% de todos os diagnósticos são dependentes da patologia (AUGUSTE; PALSANA, 2015). Durante anos, o único método disponível para essa observação diagnóstica era o método convencional utilizando um microscópio (AL-JANABI; HUISMAN; DIEST, 2012). Entretanto, o processo de observação está gradualmente mudando com os microscópios que possuem câmeras digitais, os quais permitem o compartilhamento das imagens histopatológicas com outros patologistas (AL-JANABI; HUISMAN; DIEST, 2012). O trabalho de um patologista que compreende o exame à distância é definido como telepatologia (KAYSER et al., 2004). Porém, a utilização de câmeras acopladas nos microscópios produz apenas imagens estáticas de um determinado campo visual, dependente da ampliação. Esse campo visual propicia limitações clínicas, uma vez que não consegue apresentar a imagem histopatológica por completo (AL-JANABI; HUISMAN; DIEST, 2012).

O aumento da capacidade computacional, a velocidade de transferência de dados, os avanços em tecnologia de software, e o ganho no armazenamento de serviços na nuvem, fizeram com que a patologia pudesse usufruir de novas possibilidades para aplicações digitais. Essas possibilidades estão ganhando aceitabilidade em diagnóstico, educação e pesquisa (AL-JANABI; HUISMAN; DIEST, 2012), além de aplicações em visão computacional e aprendizado de máquina (HERRMANN et al., 2018). Dentre esses ganhos tecnológicos, surgiu a tecnologia *Whole Slide Imaging* (WSI) (AL-JANABI; HUISMAN; DIEST, 2012), a qual, diferentemente dos microscópios que possuem uma câmera acoplada e que capturam as imagens em um campo visual limitado, tem a capacidade de apresentar toda a lâmina histopatológica digitalizada (AL-JANABI; HUISMAN; DIEST,

2012).

A tecnologia WSI possui dois processos para captura da lâmina histopatológica (AL-JANABI; HUISMAN; DIEST, 2012). Para o processo de captura de imagem, é utilizado um *scanner* de lâminas, que produz a imagem digital, também conhecida por lâmina digital e um software responsável pela visualização e/ou análise dessas lâminas digitais (AL-JANABI; HUISMAN; DIEST, 2012). Para o escaneamento das lâminas, a maioria dos fabricantes de soluções WSI utiliza o método de capturar múltiplas imagens em alta resolução, considerando pequenas em formato quadrado; ou ainda utilizando sistema *line scanning*, o qual consiste em fazer um escaneamento linear da lâmina, resultando em listras (BANCROFT; GAMBLE, 2008). Ambos os métodos utilizam algoritmos de composição de imagem para gerar a lâmina digital por completo (BANCROFT; GAMBLE, 2008), técnica conhecida como *image stitching* em visão computacional (BROWN; LOWE, 2007).

O custo de aquisição de soluções WSI é estimado em torno de \$ 50 000 dólares, ou mais, segundo Ma e Wang (2017). O alto valor pode ser algo impeditivo para implantação da patologia digital em países em desenvolvimento (MA; WANG, 2017). Entretanto, dado esse alto custo de aquisição, têm-se o registro de estudos visando o desenvolvimento de ferramentas com menor custo. Entre as ferramentas que foram desenvolvidas, encontram-se o sWSI (MA; WANG, 2017) e o mWSI (AUGUSTE; PALSANA, 2015), os quais possuem propostas similares de utilizar um *smartphone* acoplado no microscópio, aproveitando-se de adaptadores que estão surgindo no mercado (ROY et al., 2014) bem como sendo confeccionados via impressoras 3D (MA; WANG, 2017). O uso desse *smartphone* permite capturar imagens e utilizar um serviço disponível na nuvem para o processamento de composição completa da lâmina digital (MA; WANG, 2017). Outra ferramenta que propõe um custo significativamente menor, se comparado às soluções WSI convencionais, é o *InstantScope* que faz uso de recursos ópticos e mecânicos de um microscópio convencional comumente encontrados em laboratórios de anatomia patológica para a construção de um dispositivo que possui também o escaneamento do plano focal (GUO et al., 2015).

Com o advento de imagens digitais conferindo apoio ao acompanhamento clínico e de diagnóstico, diversos fabricantes de equipamentos de produção de imagens médicas estabeleceram seu próprio padrão para tratar imagens bem como seu próprio protocolo de operabilidade (armazenamento de dados, imagens, recuperação e transferência). Nesse contexto, observou-se, então, a necessidade de estabelecer um padrão "universal", com objetivo de criar um protocolo de operabilidade dessas imagens, único também. Corroborando ainda mais com essa questão de padronização, priorizou-se também a padronização de imagens digitais, resultando no padrão *Digital Imaging and Communication in Medicine* (DICOM) (MILDENBERGER; EICHELBERG; MARTIN, 2002). Esse padrão fornece diretrizes bem definidas de padronização para a patologia digital (HERRMANN

et al., 2018), confirmando-se como um padrão de fato.

1.2 Questão de Desenvolvimento

Como motivação para esse Trabalho de Conclusão de Curso (TCC), e com base no exposto na Contextualização, pretendeu-se responder à seguinte questão de desenvolvimento: Como desenvolver uma solução de baixo custo que consiste em um aplicativo para *smartphone*, utilizando a tecnologia WSI, sendo esse aplicativo auxiliado por um visualizador, orientado às diretrizes do padrão DICOM?

1.3 Justificativa

Com base na literatura investigada (HERRMANN et al., 2018), acordou-se que um dos equívocos comumente encontrados sobre o uso de DICOM consiste em considerá-lo apenas como um formato aberto de arquivo, o qual armazena pixels de imagens, desprezando aspectos relevantes, tais como: integração, comunicação e troca de dados e metadados.

A ferramenta mWSI propõe o armazenamento da imagem guiando-se pelo DICOM. Nesse caso, são de fato armazenados os pixels da imagem e os metadados (AUGUSTE; PALSANA, 2015). Entretanto, não fica claro como a solução mWSI usufrui de outros protocolos do DICOM, os quais foram contextualizados anteriormente.

Diante das colocações, tem-se a identificação de um *gap* tecnológico, sendo esse focado em propor uma solução computacional em cumprimento à questão de desenvolvimento, procurando deixar claro o uso da tecnologia WSI e de um visualizador orientado às diretrizes do padrão DICOM como um todo.

1.4 Objetivo Geral

No intuito de prover uma solução computacional WSI, acessível em termos de custo, utilizando-se de dispositivos comuns (i.e. *smartphones* e microscópios de laboratórios de anatomia patológica), e facilitando a troca de dados entre outras soluções WSI, o objetivo geral desse trabalho foi o desenvolvimento de um suporte orientado ao padrão DICOM para o domínio de patologia digital.

1.4.1 Objetivos Específicos

1. Desenvolvimento de um aplicativo iOS® para captação dos fragmentos da imagem ampliada da lâmina;

2. Estabelecimento de um servidor em nuvem para o processamento de imagem responsável pela composição;
3. Identificação de um algoritmo para a composição da imagem;
4. Adequação da lâmina digital ao DICOM;
5. Desenvolvimento de uma API Restful ([FIELDING, 2000](#)) com base no DICOMWeb™ ([DICOMWEB, 2019](#)), e
6. Utilização de um software para visualizar o arquivo DICOM gerado pela solução.

1.5 Organização do Documento

Esse documento possui a seguinte organização de capítulos:

1. Referencial Teórico: apresenta e descreve os conceitos relevantes para o objetivo desse trabalho, com base em outros estudos realizados;
2. Suporte Tecnológico: define as tecnologias adotadas para o desenvolvimento do trabalho e dos objetivos especificados;
3. Metodologia: apresenta a metodologia adotada para o desenvolvimento desse trabalho;
4. Prova de Conceito: apresenta uma conceitualização prática a partir da teoria definida, que auxiliou no desenvolvimento do objetivo do trabalho;
5. Mosaic: define e detalha o software desenvolvido como objetivo do trabalho;
6. Análise dos Resultados: apresenta a coleta dos resultados e suas análises, seguindo a metodologia, e
7. Conclusão: apresenta a conclusão do trabalho.

2 Referencial Teórico

Esse capítulo apresenta os principais conceitos para embasamento teórico, importantes para entendimento do contexto desse trabalho. A Seção 2.1 descreve conceitos da especialidade médica de anatomia patológica para o contexto desse Trabalho de Conclusão de Curso. Na Seção 2.2, são descritos conceitos chave do padrão de imagens médicas DICOM. Conceitos de processamento de imagem são abordados na Seção 2.4 e, na Seção, 2.4.3, são apresentados os conceitos da técnica de composição de imagens *Image Stitching* utilizando algoritmos de processamento de imagem. Por fim, é apresentado o Resumo do Capítulo.

2.1 Anatomia Patológica

A Anatomia Patológica é a especialidade médica responsável pelo estudo e diagnóstico de doenças a partir da análise histopatológica e citopatológica de extrações cirúrgicas (UCLA, 2019). À essa extração de amostras cirúrgicas, dá-se o nome de exames de biópsia, nas quais são submetidas a diagnóstico ao médico patologista (NCI, 2019). Esse trabalho irá ater-se a imagens histopatológicas.

A Anatomia Patológica também possui a responsabilidade de auxílio ao prognóstico do paciente por meio de determinação do grau de malignidade e estágio de um tumor maligno (UCLA, 2019).

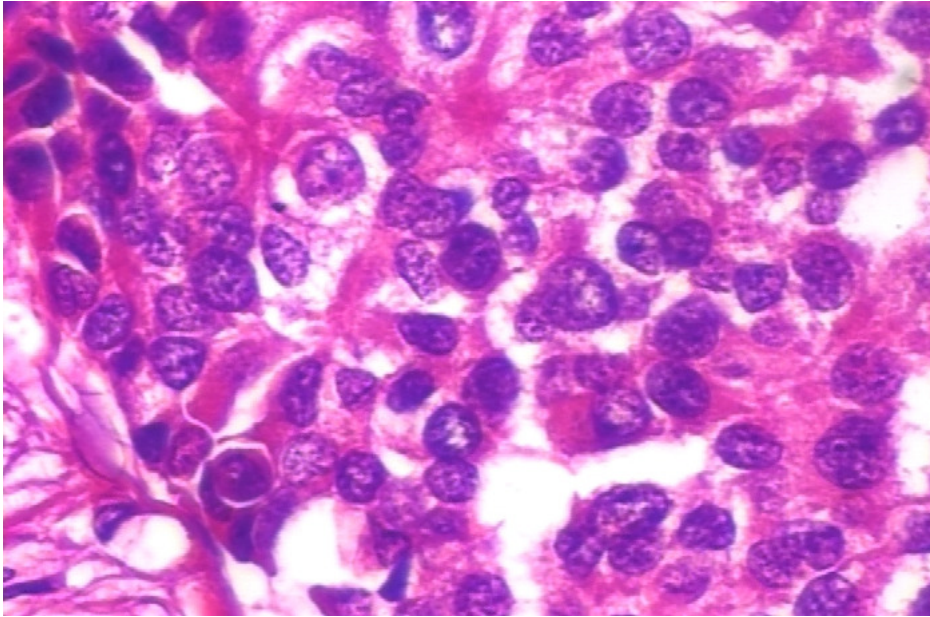
2.1.1 Lâminas Histopatológicas

A partir das amostras cirúrgicas, todo o processo deverá resultar em uma lâmina para a análise microscópica. Para que seja possível a observação, é necessária a coloração da amostra tecidual utilizando Hematoxilina e Eosina (HE) (CHAN, 2014). A coloração HE possibilita revelar detalhes celulares numa amostra tecidual de interesse do médico patologista para o diagnóstico. (CHAN, 2014).

2.1.2 Telepatologia

A Telepatologia é o trabalho de um patologista executado à distância (KAYSER et al., 2004), utilizando microscópios controlados remotamente e recursos de compartilhamento, como a internet, tanto para diagnósticos primários quanto diagnósticos secundários (KAYSER et al., 2004).

Figura 1 – Exemplo de uma lâmina histopatológica digitalizada de um carcinoma ductal ampliada em 400x.



Fonte: (SPANHOL et al., 2016).

O recurso para o exercício da telepatologia é um microscópio equipado com câmeras para aquisição de imagem estática ou vídeo para o exame à distância da lâmina. (AL-JANABI; HUISMAN; DIEST, 2012).

2.1.3 Patologia Digital

A patologia digital consiste em armazenamento, compartilhamento e troca de informação, incluindo dados e imagens, utilizando como apoio a tecnologia da informação para o gerenciamento do fluxo de trabalho na anatomia patológica (DANIEL et al., 2011). Segundo Cross et al. (2018), a patologia digital pode ser conceituada de forma objetiva como uso de imagem digital na patologia.

Segundo o *The Leeds Guide to Digital Pathology*, a patologia digital possui três tipos de tecnologia de imagem (TREANOR et al., 2018):

1. Imagem Macroscópica: aquisição e armazenamento das imagens macroscópicas para auxílio de diagnóstico histopatológico e agregação de *dataset* de câncer;
2. Telepatologia: imagens obtidas a partir de microscópios controlados remotamente e que não possuem a tarefa de armazená-las. A definição de telepatologia está descrita na subseção anterior 2.1.2, e
3. WSI: obtida a partir de *scanners* WSI e descrita na subseção 2.1.5.

2.1.4 Patologia Computacional

Louis et al. (2014) define a patologia computacional como

A abordagem diagnóstica que incorpora a aquisição de dados de diversas fontes para a extração de informações biológicas e clínicas em níveis moleculares, individuais e populacionais para gerar inferências diagnósticas e predições para então utilizar essas informações em conhecimentos clinicamente acionáveis para os consumidores por meio de laudos dinâmicos e integrados, permitindo que médicos, laboratórios e outros interessados do sistema de saúde tomem suas decisões de forma mais segura. (Tradução do autor)

São apontados quatro fatores para a definição emergente de Patologia Computacional, de acordo com (LOUIS et al., 2014):

1. Os dados patológicos digitais dos pacientes estão crescendo e estão mais detalhados e estruturados em bases de dados;
2. Os *Laboratory Information System* (LIS) estão ficando mais flexíveis, integrados e crescendo em funcionalidades. Essas características permitem análises mais complexas e melhor integração com outros sistemas diagnósticos;
3. Patologistas possuem acesso aos dados digitais dos pacientes, permitindo relacionar os dados laboratoriais e clínicos para o desenvolvimento de base de conhecimento médico e biológico, e
4. Em larga escala clínica, cresce a coleta de dados fenotípicos e o armazenamento em bases de dados estruturados, o que permite aos patologistas buscarem e integrarem informações para análise populacional.

2.1.5 *Whole Slide Imaging*

A combinação entre câmeras acopladas em microscópios, utilizadas na telepatologia para o acesso de partes da lâmina, tecnologia conhecida como *Whole Slide Access* e a alta resolução de câmeras digitais, possibilitou o desenvolvimento de *scanners* de lâminas (AL-JANABI; HUISMAN; DIEST, 2012) para o surgimento do WSI. O WSI, também podendo ser chamado de lâmina digital, lâmina virtual (AL-JANABI; HUISMAN; DIEST, 2012) ou microscópio virtual (FARAHANI; PARWANI; PANTANOWITZ, 2015), é explorado a partir de um software que possui um visualizador de imagens (AL-JANABI; HUISMAN; DIEST, 2012).

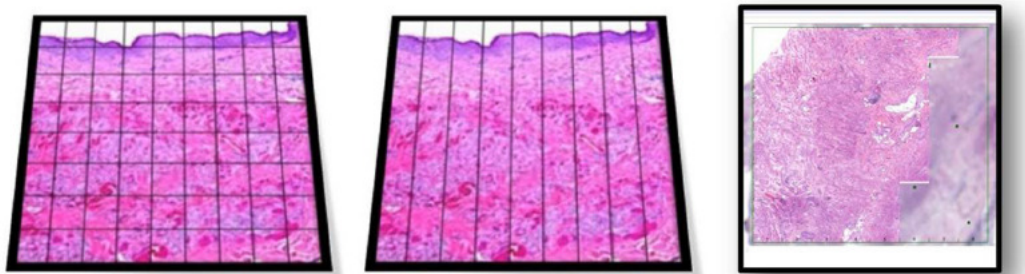
Segundo Al-Janabi, Huisman e Diest (2012), o exame de uma lâmina digital pode ser comparado ao método convencional de exame via microscópio de três formas:

1. Ajuste de magnitude da imagem;
2. Navegação em qualquer direção na lâmina digital, e
3. Alteração de plano focal.

Comumente, scanners WSI utilizam de duas técnicas para captura de toda a imagem da lâmina, baseando-se em algoritmos de *Image Stitching*, os quais serão mais bem detalhados na seção 2.4.3. Essas duas técnicas são: *Tile-Based Scanning* e *Line-Based Scanning* (FARAHANI; PARWANI; PANTANOWITZ, 2015).

- ***Tile-Based Scanning***: são capturadas imagens em formatos quadrados da lâmina para a composição da imagem completa. Comumente com 2% a 5% de sobreposição entre elas (FARAHANI; PARWANI; PANTANOWITZ, 2015).
- ***Line-Based Scanning***: captura imagem em um movimento linear, em um só eixo, formando listras (FARAHANI; PARWANI; PANTANOWITZ, 2015) que são agrupadas para a composição da lâmina digital.

Figura 2 – Diferença no mapeamento da imagem entre *Tile-Based Scanning* e *Line-Based Scanning*



Fonte: (FARAHANI; PARWANI; PANTANOWITZ, 2015)

Segundo Farahani, Parwani e Pantanowitz (2015), a técnica linear de captura das imagens que a lâmina digital é composta é menos complexa do que a *Tile-Based Scanning*, pois o processo de alinhamento é simplificado devido à redução do número de listras e graus de liberdade associados a cada uma.

2.2 DICOM

O padrão *Digital Imaging and Communications in Medicine* (DICOM) tem o intuito de estabelecer protocolos para imagens médicas, armazenamento, busca e troca desses dados para viabilizar a troca aberta de informações sobre imagens médicas e seus

metadados, auxiliando no acompanhamento clínico e diagnóstico, bem como na comunicação de equipamentos de diferentes fabricantes (MILDENBERGER; EICHELBERG; MARTIN, 2002).

O DICOM surgiu com a iniciativa da *American College of Radiology* (AMC) e a *National Electrical Manufacturers Association* (NEMA) em 1983, emergindo o primeiro padrão, chamado ACR-NEMA, em 1985. Esse padrão confirmou-se como uma forma padronizada para armazenar dados de mídia e a comunicação não proprietária (MILDENBERGER; EICHELBERG; MARTIN, 2002). Em 1988, surge uma nova versão, ACR-NEMA 2.0, que traz terminologias, estrutura dos dados e *encoding* (MILDENBERGER; EICHELBERG; MARTIN, 2002). Na versão 3.0, o padrão, por fim, é nomeado DICOM, lançado em 1993, no qual se apresenta a especificação de protocolo de rede dependente do modelo ISO/OSI, utilizando o TCP/IP para não ter dependência de soluções proprietárias (MILDENBERGER; EICHELBERG; MARTIN, 2002).

Atualmente, o DICOM está além da transferência de dados de imagem. Esse também estabelece protocolos de objetos de interesse de outros dados clínicos relacionados como dados do paciente e laudos (MILDENBERGER; EICHELBERG; MARTIN, 2002).

2.2.0.1 Definição de Objetos no DICOM

O DICOM representa as entidades e seus relacionamentos abstraindo elementos reais. o DICOM *Data Model* apresenta essas entidades de forma normalizada. Entretanto, as classes dos objetos DICOM (*DICOM Objects*) são compostas por módulos de diferentes entidades. Essas classes no DICOM são chamadas de *Service-Object Pair* (SOP) *Classes DICOM* (2019a).

Ainda segundo a especificação DICOM (2019a), as SOP *Classes* são definidas pelo *Information Object Definitions* (IOD's), que são uma coleção de módulos que representa uma coleção de elementos, também chamados de atributos.

O modelo entidade-relacionamento das informações definidas pelo DICOM é apresentado na Figura 58, no Anexo B.

2.2.0.2 DICOMWeb™

O DICOMWeb é um protocolo para APIs RESTful e possui serviços de operação baseados nos padrões de serviço definidos pelo próprio DICOM (DICOM, 2019b) e que possuem seus próprios *endpoints*. Esse trabalho utilizou o serviço de armazenamento de imagem, denominado no DICOMWeb™ como *Store Over the Web*(STOW-RS).

2.2.0.3 DICOM na Patologia Digital

Segundo [Herrmann et al. \(2018\)](#), a implementação de um padrão comum entre diferentes fabricantes e especialistas, tanto de imagem quanto de protocolo de comunicação, facilita a interoperabilidade de informações clínicas e diagnósticas.

A necessidade de interoperabilidade da tecnologia WSI foi notada pela *Digital American Association* (DAP) e pelo *DICOM Working Group 26* ([HERRMANN et al., 2018](#)). Diversos fabricantes participaram desses grupos e discutiram a importância, pela primeira vez, da troca de imagens utilizando padronização DICOM, bem como outros obstáculos (ex. propriedade intelectual) foram superados ([HERRMANN et al., 2018](#)).

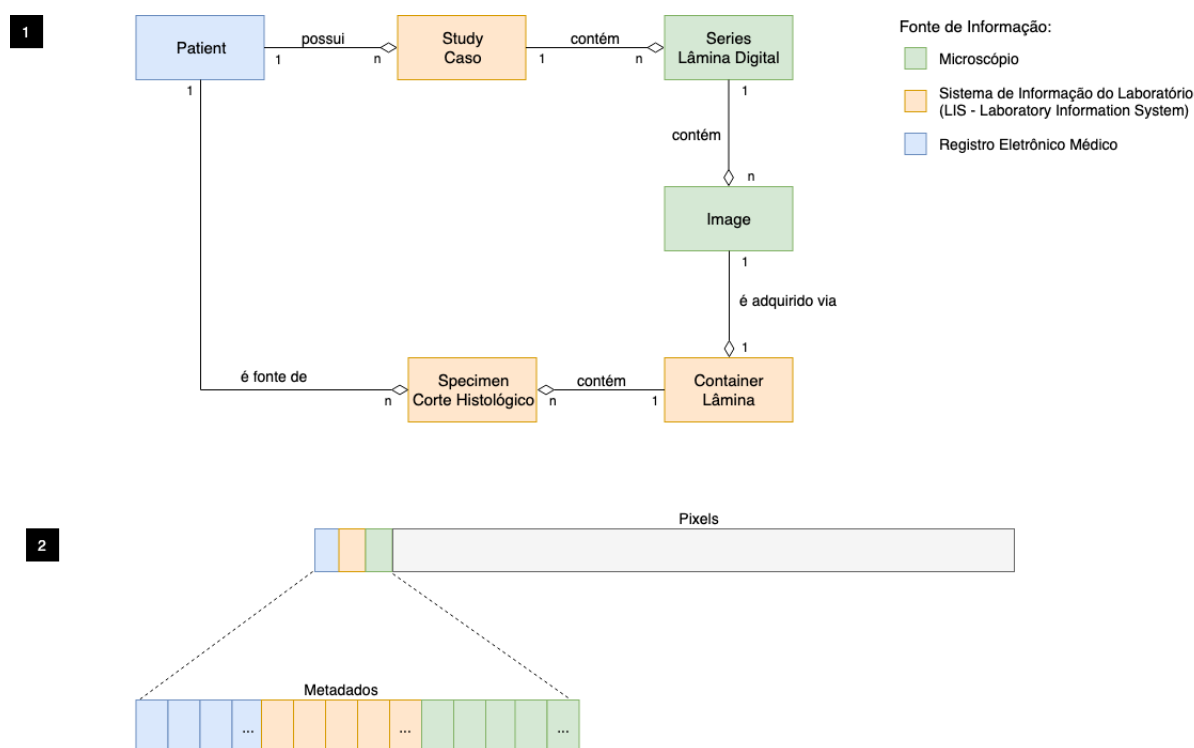
O estudo de [Herrmann et al. \(2018\)](#) concentrou-se em utilizar informações clínicas obtidas a partir de duas soluções computacionais da categoria LIS e lâminas virtuais WSI, obtidas de diversos fabricantes de *scanners* WSI e, com isso, avaliar se o DICOM tem um sentido prático para a patologia.

A Figura 3 representa um diagrama de classes, em um contexto real, com suas entidades e relacionamentos, orientando-se pelo DICOM, sendo esse especializado para a patologia digital, segundo [Herrmann et al. \(2018\)](#), e o agrupamento dos metadados com os pixels da imagem. As entidades modeladas por [Herrmann et al. \(2018\)](#) são:

- **Patient:** dados do paciente dispostos no padrão DICOM;
- **Study:** dados do caso do paciente;
- **Series:** contém os dados da lâmina digital, que é o objeto de estudo;
- **Image:** imagem WSI adquirida via lâmina histopatológica;
- **Container:** lâmina física que contém o corte histológico, e
- **Speciment:** corte histológico preparado na coloração HE.

Esse trabalho utilizou o IOD *Secondary Capture Image (SC Image)* por ser focado na conversão de imagens que não são DICOM para um formato DICOM independente e sua especificação consta na seção A.8 do padrão DICOM ([DICOM, 2013](#)), sendo a representação de imagem DICOM mais simples ([RONI, 2011](#)), contendo menos atributos que o *Visible Light Whole Slide Microscopy Image* representado por [Herrmann et al. \(2018\)](#), já que também as imagens capturadas não utilizaram o atributo do ponto focal no eixo z. Uma representação em notação *Unified Modeling Language (UML)* simplificada para as entidades do *SC Image* é apresentado na Figura 19, Seção 6.2.4 do capítulo sobre o Mosaic.

Figura 3 – Representação de informações na patologia digital no DICOM



Fonte: (HERRMANN et al., 2018)

2.3 Adaptadores de Smartphone para Microscópios

Diversos estudos estão sendo desenvolvidos analisando a qualidade da captura de imagens microscópicas a partir das câmeras de smartphone (ROY et al., 2014). Com a alta qualidade de imagem adquirida por smartphones atualmente, foram desenvolvidos adaptadores que os suportem nos microscópios para a aquisição das imagens (ROY et al., 2014).

O estudo de Roy et al. (2014) apresenta diversos adaptadores de microscópio, que variam de modelos únicos, de acordo com o *smartphone*, a modelos universais, os quais atendem ao acoplamento de diversos modelos de *smartphone*.

2.4 Processamento de Imagem

2.4.1 Representação de Imagem Digital

2.4.1.1 Imagem Binária

Imagens binárias possuem pixels com apenas dois níveis de intensidade: 0 e 1, onde 0 representa a cor preta e 1 a cor branca. Um exemplo de imagem binária é apresentada na Figura 4.

Figura 4 – Imagem binária



Fonte: Autor

2.4.1.2 Imagem em Escala de Cinza

A escala em cinza apresenta um canal em escala de cinza, que varia em intensidade de 0 a 255, representando uma variedade na escala comparada a imagens binárias. A Figura 5 apresenta uma imagem em escala de cinza.

Figura 5 – Imagem em escala de cinza



Fonte: Autor

2.4.1.3 Imagens Coloridas - RGB

As imagens coloridas são representadas digitalmente em três canais de cor, sendo estes vermelho, verde e azul, constituindo o padrão *Red, Green, Blue* (RGB) (GONZALES

RAFAEL; WOODS RICHARD, 2006). A intensidade de cada matriz que representa o canal de cor varia de 0 a 255. Na Figura 6, é apresentada uma imagem colorida e os canais de cor RGB, elaborada no estudo de Deng e Song (2014).

Figura 6 – Imagem colorida com as três camadas RGB.



Fonte: (DENG; SONG, 2014)

2.4.2 Intensidade de Imagens

Como abordado nas três subseções anteriores sobre os tipos de imagem, a intensidade em imagens binárias é representada com valores de 0 ou 1. Entretanto, as imagens em escalas de cinza e RGB possuem um intervalo de intensidade que varia de 0 a 255 em cada pixel. Em alguns estudos, a intensidade é representada como peso, o qual é a divisão entre a intensidade e o valor máximo suportado, dando um intervalo de 0 a 1, com números fracionados, representando a intensidade (JYOTHI; SUSHMA; VEERESH, 2015).

2.4.3 *Image Stitching*

A técnica de *Image Stitching* consiste em algoritmos que tem a função de compor imagens a partir de pequenas amostragens com sobreposição entre elas. Esses algoritmos são largamente utilizados em composição de fotografias de mapas e satélites, bem como em fotos panorâmicas de diversos ângulos (SZELISKI et al., 2007). O estudo QuickTime (1995) utiliza a abordagem de *Image Stitching* para construção de campo visual para a navegação em realidade virtual. A Figura 7 apresenta o panorama do horizonte de Marte a partir das imagens coletadas pelo rover Perseverance (NASA, 2021).

Essa seção tem o objetivo de apresentar conceitos importantes para o desenvolvimento do *Image Stitching*.

2.4.4 Registro de Imagem

Os algoritmos de *Image Stitching* utilizam implementações de registro de imagem (encontrado na literatura em inglês como *Image Registration* ou *Image Align*), para que

Figura 7 – Algoritmo de *Image Stitching* aplicado para gerar um panorama de Marte pelo *Perseverance*.



Fonte: (NASA, 2021)

seja possível transformar todas as imagens que a compõem em uma mesma coordenada (SZELISKI et al., 2007) e, assim, gerar como saída a imagem completa em uma mesma perspectiva.

Algumas das abordagens de registro de imagem incluem a de forma direta e a de utilização de características da imagem (SZELISKI et al., 2007). A abordagem direta compara a relação de pixels entre as imagens, verificando-se a quantidade de pixels correspondentes (SZELISKI et al., 2007); enquanto que o método de utilização de características da imagem utiliza algoritmos de detecção e descrição de características para que ocorra a comparação das imagens e, assim, obtendo as características correspondentes entre as imagens.

2.4.4.1 Detecção e Descrição de Características

As características de uma imagem ou *Features* são pontos que ela possui, sendo esses distinguíveis dentre os outros pontos da imagem (conhecidos também como *Key-points*, *Feature Points* ou *Interest Points*). Para a detecção desses pontos, são utilizados algoritmos de detecção de características (CULJAK et al., 2012). Além da detecção, é necessário saber o que caracteriza esses pontos como distinguíveis, aplicando-se algoritmos descritores (OPENCV, 2019).

Os *corners* de uma imagem são comumente referenciados como pontos de características (CULJAK et al., 2012), sendo citados na documentação do OpenCV como exemplo de que os *corners* podem ser mais facilmente diferenciados de pontos planares mais simples (OPENCV, 2019). Um dos algoritmos utilizados para detecção de *corners* é o *Harris Detector* (CULJAK et al., 2012).

Algoritmos detectores de características invariantes à escala e à rotação foram desenvolvidos como evolução. Estes são importantes para, por exemplo, correspondência de pontos de características (*Feature Matching*) entre duas imagens, independentes da distância entre determinado objeto contido nela (CULJAK et al., 2012). Alguns dos algoritmos de detecção invariantes à escala e também à rotação são:

- *Scale Invariant Feature Transform* (SIFT): algoritmo tanto detector quanto descritor de características, com o propósito de ser invariante à escala, rotação, intensidade e mudanças de ponto de visão da imagem (KARAMI; PRASAD; SHEHATA, 2017).
- *Speed up Robust Feature* (SURF): variante do algoritmo SIFT, concentrando-se na otimização do tempo, sendo esse importado pelo SIFT, não reduzindo a qualidade dos pontos de características detectados (CULJAK et al., 2012)
- *Oriented FAST and Rotated BRIEF* (ORB): proposto por Rublee et al. (2011), como o nome referencia, utiliza o detector de características *Features from Accelerated Segment Test*(FAST) e o descritor BRIEF, sendo uma alternativa eficiente ao SIFT e SURF (RUBLEE et al., 2011). Um melhor detalhamento do algoritmo ORB é apresentado na 2.4.4.4.

Existem diversos estudos de *Image Stitching* que utilizam algoritmos baseados em detecção e descrição de características em imagens biomédicas (PYARWIN; KITJAI-DURE, 2018), como por exemplo: a solução sWSI de (MA; WANG, 2017) que utiliza o SURF. Entretanto, o algoritmo ORB é pouco explorado segundo PyarWin e Kitjaidure (2018). Pyarwin afirma também que é o primeiro estudo utilizando ORB para imagens biomédicas. Pelos motivos de pouca exploração do algoritmo ORB para *Image Stitching*; por ser um algoritmo aberto, diferentemente do SIFT e SURF que possuem patente; por apresentar melhor desempenho (KARAMI; PRASAD; SHEHATA, 2017), permitindo uma otimização de tempo para aplicações em tempo real, bem como robustez comparável ao SIFT (ADEL; ELMOGY; ELBAKRY, 2015), escolheu-se o ORB para a etapa de detecção e descrição de características. Esse utiliza o algoritmo *Features from Accelerated Segment Test* (FAST), descrito na próxima subseção 2.4.4.2, para a detecção de pontos de características, e o algoritmo *Binary Robust Independent Elementary Features* (BRIEF) como descritor, apresentado na 2.4.4.3. O algoritmo ORB está detalhado na subseção 2.4.4.4.

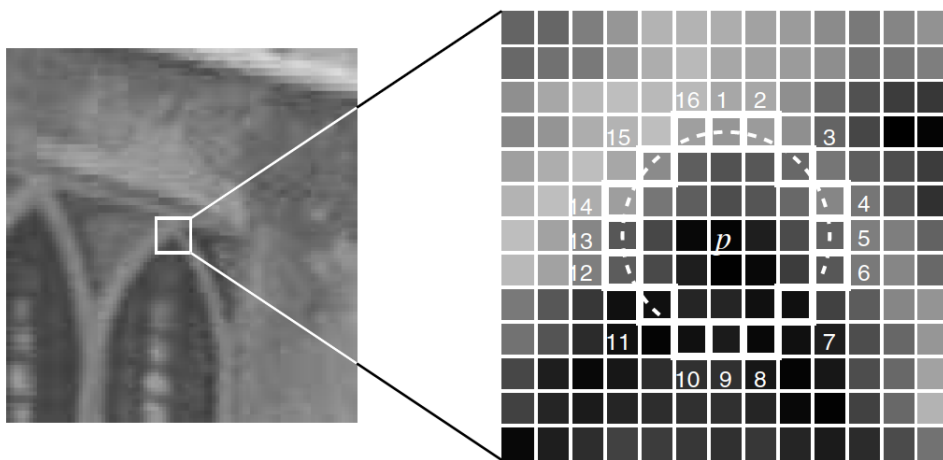
2.4.4.2 *Features from Accelerated Segment Test* (FAST)

O detector de características FAST foi proposto por Rosten e Drummond (2006), com o objetivo de ter um desempenho superior a outros algoritmos detectores, tornando-se uma alternativa de interesse em aplicação que necessita de detecção em tempo real ou com poder computacional limitado.

Para expressar o FAST, considera-se um ponto que representa um *corner* p e um conjunto n de pixels ao redor desse ponto p , formando um círculo e definindo um limite (*threshold*) t . A intensidade dos pixels adjacentes ao ponto p deverá ter o brilho mais alto $I_p + t$ ou mais escuro $I_p - t$ que p . Se nenhum desses dois critérios forem satisfeitos, o ponto p não será considerado um *corner*.

A Figura 8 apresenta a ampliação de uma amostra de imagem que ilustra o algoritmo FAST.

Figura 8 – Segmento de 16 pontos adjacentes de teste do ponto p para classificar se é um *corner*.



Fonte: (OPENCV, 2019a)

2.4.4.3 Binary Robust Independent Elementary Features (BRIEF)

O BRIEF é um descritor de características baseado em *strings* binárias, que consiste em atribuir um conjunto de descritores em *string* em um vetor de dimensão pré determinada (CALONDER et al., 2010). Por ser um descritor que se concentra em apresentar de *strings* binárias, o BRIEF tem um desempenho superior a outros algoritmos descritores como SIFT e SURF (CALONDER et al., 2010).

Sendo dada uma imagem τ com uma amostra dessa imagem de dimensão $S \times S$, a descrição binária será dada por:

$$\tau(p; x, y) = \begin{cases} 1, & \text{se } I(x) < I(y) \\ 0, & \text{caso contrário} \end{cases}$$

Em que p representa o *patch*, ou seja, uma região amostral e quadrada da imagem, $I(x)$ representa a intensidade do pixel no ponto x e $I(y)$ no ponto y . Escolhendo um conjunto n_d de localização dos pontos (x, y) , o vetor de descritores do BRIEF é dado por:

$$\sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i)$$

O estudo de (CALONDER et al., 2010), autor do BRIEF, define n_d com dimensões de 128, 256 ou 512.

A intensidade é dada após a aplicação do *Smoothing Filter* (CALONDER et al., 2010), que é utilizado para redução de ruídos e (*blurring*) da imagem (GONZALES RAFAEL; WOODS RICHARD, 2006).

2.4.4.4 Oriented FAST and Rotated BRIEF (ORB)

O algoritmo ORB é dado como uma alternativa ao SURF e SIFT, como mencionado anteriormente na subseção 2.4.4.1. O ORB é um detector e descritor de características invariante à escala e rotação que utiliza os algoritmos FAST e BRIEF. Entretanto, o FAST não possui a informação do quanto uma região R possui um ponto representando um *corner*, medida denominada *Cornerless Measure* e escala; enquanto o algoritmo detector BRIEF não é invariante à rotação (RUBLEE et al., 2011). Por conta dessas limitações, o algoritmo ORB aplica otimizações para que seja possível ser invariante às escala e rotação (RUBLEE et al., 2011).

O estudo de Rublee et al. (2011) utiliza o *Harris Corner Measure* para obter informações de quantidade de pontos que representam um *corner* *cornerless measure*. Para que seja possível o ORB ser invariante à escala, em Rublee et al. (2011), uma pirâmide com diversos níveis de escala é acordada, aplicando a cada nível o algoritmo FAST para detecção de características, utilizando o filtro de Harris.

Para obter a propriedade de orientação de um *corner*, o ORB utiliza o *intensity centroid*. Esse define a intensidade do *corner* como o deslocamento do *corner* de seu centro, sendo esse vetor usado para atribuir uma orientação (RUBLEE et al., 2011). Os momentos de uma imagem amostral, segundo Rosin (1999), são definidos como:

$$m_{qp} = \sum_{x,y} x^p y^q I(x, y)$$

O centróide é determinado como:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

Por fim, obtém-se a orientação a partir do vetor do centro do *corner* que é a origem O , ao centróide C , formando o vetor \overrightarrow{OC} e calculando o ângulo:

$$\theta = \arctan2(m_{01}, m_{10})$$

O ORB é invariante à rotação. Para que seja possível, essa propriedade [Ruble et al. \(2011\)](#) otimiza o algoritmo BRIEF definindo uma matriz S de dimensão $2 \times n$ das descrições binárias:

$$S = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \end{bmatrix}$$

Utilizando a descrição da direção θ e a matrix de rotação correspondente R_θ , gera-se uma matriz S_θ a partir de S :

$$S_\theta = R_\theta S$$

Consequentemente, o descritor que possui a orientação:

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta$$

2.4.4.5 Matriz de Homografia

A matriz de homografia é uma matriz 3×3 responsável por uma transformação linear que auxilia no alinhamento das imagens componentes, permitindo que as imagens sejam dispostas em suas coordenadas ([ADEL; ELMOGY; ELBAKRY, 2015](#)).

Seja uma imagem x_1 e outra imagem x_2 e a matriz de homografia H , a relação de perspectiva será dada por:

$$x_1 = Hx_2$$

2.4.4.6 Composição da Imagem

Após o registro de todas as imagens que componentes, segue-se para a etapa de composição, na qual irá combinar as imagens utilizando as sobreposições ([SZELISKI et al., 2007](#)).

Segundo [Szeliski et al. \(2007\)](#), a decisão do melhor algoritmo para a composição de imagem têm como parâmetros:

1. a superfície final da imagem (plana, esférica ou cilíndrica);

2. uma imagem de referência;
3. a seleção de pixels que contribuirão para a composição, e
4. informações quanto às otimizações visando reduzir resultados insatisfatórios na imagem final (ex. visibilidade das junções das imagens componentes, *blur* e fantasmas).

Para campos visuais grandes, que excedem 90° , construir uma imagem final em uma superfície plana geralmente gera distorções nas bordas (SZELISKI et al., 2007), sendo uma boa escolha para esses casos as projeções cilíndricas ou esféricas (SZELISKI et al., 2007). A seleção de uma imagem como parâmetro de composição para as outras pode seguir sendo a imagem mais central geometricamente, tanto para superfícies planas quanto para panoramas que utilizem o método cilíndrico ou esférico.

Com a seleção da imagem componente de referência e a superfície, com base em Szeliski et al. (2007), as coordenadas precisam ser mapeadas entre as imagens componentes e a saída da imagem esperada. Caso a imagem final esteja em uma superfície plana, sem distorções radiais, como no caso desse trabalho, a matriz de homografia poderá ser utilizada para este fim. Para formas cilíndricas e esféricas, os pixels devem ser convertidos em um ponto 3D e, então, mapeadas de acordo com a projeção (SZELISKI et al., 2007).

2.5 Resumo do Capítulo

Nesse capítulo, foram apresentados conceitos importantes para a conceitualização teórica para o desenvolvimento desse trabalho. Apresentou-se conceitos da área de anatomia patológica, que é a área no qual se aplica a solução desenvolvida nesse trabalho. Conceitualizou-se o padrão DICOM para imagens médicas, apresentando-se um importante padrão para a interoperabilidade dessas imagens. Definiu-se todo o conceito e descrição matemática para desenvolver o algoritmo de *Image Stitching* para a composição da imagem WSI a partir da captura via câmera de *smartphone*. Adicionalmente, foi definido e descrito o algoritmo ORB para detecção e descrição de características das imagens.

3 Suporte Tecnológico

Nesse capítulo, são apresentadas as tecnologias utilizadas para o desenvolvimento desse trabalho em que as seções descrevem cada tecnologia aplicada.

3.1 *Xcode* 12.4

Xcode[®] (APPLE, 2019) é uma *ntegrated Development Environment* (IDE) desenvolvida pela empresa *Apple*[™] com o principal objetivo de fornecer um ambiente para desenvolvedores de suas plataformas, incluindo o sistema operacional iOS[®] dos produtos iPhone[®] e iPad, bem como para os sistemas operacionais OSX.

3.2 *OpenCV* 3.4.5

OpenSource Computer Vision Library (*OpenCV*) é uma biblioteca de visão computacional lançada em 1999 pela Intel[®] (CULJAK et al., 2012), e possui interfaces disponíveis para as linguagens *Python*, C++, Java e MATLAB (OPENCV, 2019b).

OpenCV possui mais de 2500 algoritmos otimizados de visão computacional, que incluem reconhecimento de objetos, processamento de vídeo, processamento de imagem, algoritmos de detecção e descrição de características, dentre outros (OPENCV, 2019b). Segundo o site oficial do OpenCV, empresas consolidadas como Google[®], IBM[®], Microsoft[®], a própria Intel[®], entre outras utilizam o OpenCV em seus produtos e serviços, bem como *startups*.

3.3 *Git* 2.14

O gerenciador de controle de versão distribuído *Git* (GIT, 2019) foi desenvolvido por Linus Torvalds para projeto na utilização do kernel Linux e depois expandido para a comunidade. Atualmente, o *Git* é uma das ferramentas de gerenciamento de controle de versão mais utilizadas por desenvolvedores em vários projetos *opensource* e comerciais.

3.4 *GitHub*

O *GitHub* (GITHUB, 2019) é uma versão remota para produtos de software que utilizam o sistema de controle de versão *Git*. Com o *GitHub*, é possível utilizar o *GIT* de forma escalável e interoperável, pois não limita o seu uso apenas localmente.

3.5 *PostDICOM*

O *PostDICOM* é um *Picture Archiving and Communication System* (PACS ([POST-DICOM, 2021](#))), sendo esse um serviço que tem por objetivo facilitar o armazenamento de imagem e a comunicação entre setores em um ambiente de diagnóstico de imagem ([CHOPLIN; 2ND; MAYNARD, 1992](#)).

Com o *PostDICOM*, é possível obter funcionalidades de visualização de arquivos DICOM, e também possui função de colaboração entre usuários. Sua versão gratuita oferece 50GB de armazenamento disponíveis ([POSTDICOM, 2021](#)).

3.6 *Pydicom 2.1.2*

O *Pydicom* é uma biblioteca *open source* escrita em *Python* com o intuito de ler e gerar arquivos DICOM ([MASON, 2011](#)). Lançada a primeira versão em 2008 ([MASON, 2011](#)) e está na versão estável 2.1.2.

3.7 *DICOM Validator*

DICOM Validator é uma ferramenta em linha de comando, que permite validar arquivos DICOM e verificar se estão de acordo com o protocolo, gerando *logs* de erro e *warnings*, caso encontre inconsistências com o padrão DICOM ([CLUNIE, 2018](#)).

3.8 *Draw.io*

Draw.io ([DRAW.IO, 2019](#)) é uma ferramenta em nuvem para criação de diagramas de diversos tipos, incluindo *Unified Modeling Language* (UML), *Business Process Model and Notation* (BPMN), fluxogramas, entre outros. Nesse trabalho, o *Draw.io* foi utilizado para a modelagem de diagramas BPMN para processos de metodologia, para a arquitetura de software e diagramas de entidade relacionamento.

3.9 Resumo do Capítulo

Esse capítulo apresentou as principais tecnologias que foram utilizadas tanto para o desenvolvimento da solução de acordo com os objetivos propostos por esse Trabalho de Conclusão de Curso, quanto para o desenvolvimento da monografia. A Tabela [3.9](#) apresenta as ferramentas e suas descrições.

Tabela 1 – Ferramentas de suporte tecnológico.

Xcode	IDE com principal objetivo de atender desenvolvedores para plataformas <i>Apple</i>
OpenCV	Biblioteca para aplicação de processamento de imagem e visão computacional.
Git	Sistema de controle de versões utilizada principalmente em desenvolvimento de software.
<i>GitHub</i>	Serviço em nuvem para armazenar e gerenciar repositórios Git.
<i>PostDICOM</i>	Serviço PACS em nuvem para arquivos DICOM.
<i>DICOM Validator</i>	Ferramenta de validação de conformidade de arquivos DICOM.
<i>Draw.io</i>	Ferramenta em nuvem para criação de diagramas de diversos tipos.

4 Metodologia

Esse capítulo visa descrever toda a metodologia utilizada para o desenvolvimento desse trabalho, tanto em pesquisa, quanto no desenvolvimento da solução WSI descrita no Capítulo 6. Na seção 4.1, é descrita a classificação da pesquisa. O fluxo de atividades que envolve todo o processo para o desenvolvimento do Trabalho de Conclusão de Curso 1 e Trabalho de Conclusão de Curso 2 é descrito na seção 4.2. A metodologia de desenvolvimento é descrita na seção 4.3. A metodologia de análise de resultados é apresentada na seção 4.4. Por fim, tem-se o Resumo do Capítulo.

4.1 Classificação de Pesquisa

Uma pesquisa científica, segundo Gerhardt e Silveira (2009), é classificada de acordo com os pontos de vista de sua abordagem, sua natureza, seu objetivo e seus procedimentos. Em procedimentos, Prodanov e Freitas (2013) menciona também a pesquisa operacional.

A Figura 9 apresenta os tipos de pesquisa mencionados por Gerhardt e Silveira (2009) e Prodanov e Freitas (2013), e a classificação desse trabalho está evidenciada em cor azul.

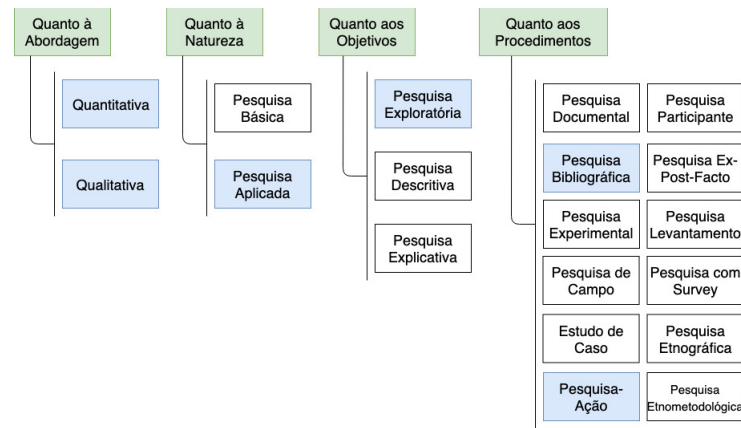


Figura 9 – Tipos de Pesquisa

4.1.1 Abordagem

A pesquisa qualitativa, segundo Gerhardt e Silveira (2009), concentra-se em uma compreensão e explicação da dinâmica de relações sociais, as quais possuem realidades que não são quantificadas. Esse trabalho utiliza a abordagem qualitativa por desenvolver uma solução em software para digitalização de imagens histopatológicas, conscientizando

a importância dessa aplicação para profissionais e entendendo as relações desses com a tecnologia. Dessa forma, por serem elementos subjetivos, utilizou-se uma abordagem qualitativa.

4.1.2 Natureza

Segundo [Gerhardt e Silveira \(2009\)](#), a pesquisa aplicada caracteriza-se por gerar conhecimentos para aplicação prática para uma solução de problemas específicos, envolvendo verdades e interesses locais. Esse trabalho utiliza pesquisa aplicada, pois desenvolve uma solução baseada em software acessível, em um contexto específico - sendo esse a digitalização de lâminas histopatológicas - e a partir de um *smartphone* acoplado a um microscópio.

4.1.3 Objetivos

A pesquisa exploratória utiliza estudos anteriormente tratados para embasar o desenvolvimento do trabalho, possibilitando maior familiaridade com o problema ([GERHARDT; SILVEIRA, 2009](#)), e utilizando o levantamento bibliográfico para esse fim. Segundo [Prodanov e Freitas \(2013\)](#), o estudo bibliográfico também é utilizado nas pesquisas para a formação do referencial teórico, como especificado nesse trabalho.

4.1.4 Procedimentos Técnicos

Na pesquisa-ação, os pesquisadores desempenham um papel atuante na pesquisa em cooperação com os participantes da situação ou do problema para estabelecer uma ação ou resolução de um problema coletivo, com apoio em base empírica ([PRODANOV; FREITAS, 2013](#)).

[Tripp \(2005\)](#) afirma que a pesquisa-ação é uma das formas de investigação-ação, sendo um termo genérico que possui um ciclo básico de ação:

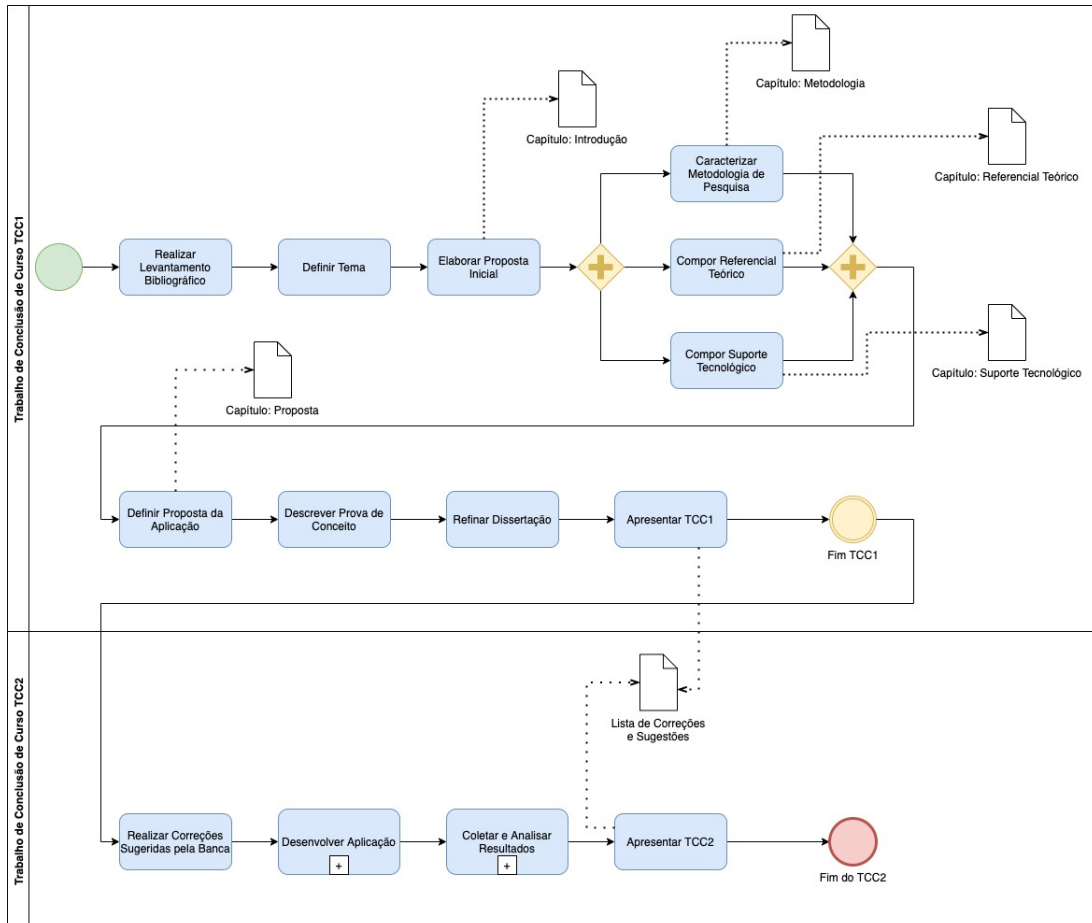
1. Planejar uma melhora para a prática;
2. Implementar a ação planejada;
3. Descrever os efeitos da ação, e
4. Avaliar os resultados da ação, retornando ao planejamento e executando o ciclo.

Devido à característica empírica e cíclica da pesquisa-ação, adotou-se, nesse trabalho, por esse tipo de procedimento, o qual se adequa ao contexto de metodologia de desenvolvimento de software iterativo e incremental.

4.2 Fluxo de Atividades

Seguem as principais atividades, as quais estão sendo realizadas visando:

Figura 10 – Fluxo de Atividades do Trabalho de Conclusão de Curso (TCC)



Fonte: Autor

1. **Realizar Levantamento Bibliográfico:** essa atividade objetivou levantar informações para familiarização com o tema pré definido na área de anatomia patológica, e definir um problema a ser solucionado;
2. **Delimitar Tema:** com o levantamento bibliográfico realizado, o tema foi delimitado a fim de se ter um objetivo para o trabalho, que é o desenvolvimento de um software de custo acessível de digitalização de lâminas histopatológicas de acordo com o padrão DICOM;
3. **Elaborar Proposta Inicial:** definida uma proposta, no intuito de atender a questão chave desse trabalho, sendo essa proposta um aplicativo para *smartphones iPhone*[®] bem como um software visualizador de dados;

4. **Caracterizar Metodologia de Pesquisa:** a metodologia do trabalho é definida, com base em abordagem, natureza, objetivos e procedimentos técnicos, bem como as coleta e análise de resultados, sendo a saída definida na seção 4.1;
5. **Compor Referencial Teórico:** com o objetivo definido, foi acordado o referencial teórico, com os principais conceitos que embasam esse trabalho. Esse levantamento foi realizado por meio de uma pesquisa bibliográfica. O referencial teórico está no capítulo 2;
6. **Compor Suporte Tecnológico:** nessa atividade, foram estabelecidas as tecnologias utilizadas para o desenvolvimento da aplicação, bem como desse trabalho. O suporte tecnológico está descrito no Capítulo 3;
7. **Definir Proposta da Aplicação:** atividade realizada em que se descreveu a aplicação, desde a modelagem, padrões de projeto e protótipos para apresentar o software a ser desenvolvido. (adicionar antigos artefatos da proposta em apêndice?).
8. **Descrever Prova de Conceito:** nessa atividade, descreveu-se uma forma prática de ilustrar o embasamento teórico e que possibilitou avaliar a possibilidade do desenvolvimento do software final (enviar para apêndice?).
9. **Refinar Dissertação:** melhorias que foram aplicadas na dissertação com auxílio dos orientadores de forma contínua e cíclica;
10. **Apresentar TCC1:** apresentação do TCC1 para a avaliação da banca, sendo essa uma atividade já realizada;
11. **Realizar Correções Sugeridas pela Banca:** sugestões da banca avaliadora foram implementadas ao trabalho;
12. **Desenvolver Aplicação:** atividade que possui um sub-processo que visou desenvolver a aplicação de acordo com o processo de desenvolvimento definido na seção 4.3;
13. **Coletar e Analisar Resultados:** atividade que possui um sub-processo implementado no decorrer do TCC2 e descrito na seção 4.4, e
14. **Apresentar TCC2:** apresentação do trabalho concluído para a banca avaliadora.

4.3 Metodologia de Desenvolvimento

Para o desenvolvimento da solução proposta, é utilizada uma metodologia de desenvolvimento adaptada do *framework Scrum*, em conjunto com metodologias ágeis para os requisitos criados em forma de histórias de usuário.

O estudo de [Pagotto et al. \(2016\)](#) adapta o *framework Scrum* para o desenvolvimento no contexto de apenas um desenvolvedor, chamado de Scrum Solo. As principais atividades do Scrum Solo, de acordo com [Pagotto et al. \(2016\)](#) são:

- **Requisitos:** os requisitos do software são elicitados para gerar o *product backlog*, definindo assim o escopo do produto e os clientes da solução;
- **Sprint:** a partir da seleção de alguns requisitos do *product backlog*, é planejado o ciclo de desenvolvimento do software. Têm como entradas o *product backlog* e o protótipo do software. [Pagotto et al. \(2016\)](#) define o ciclo da *sprint* no Scrum Solo como sendo de uma semana;
- **Entrega:** é a atividade que disponibiliza o produto para o cliente. Possui como entrada a planta de desenvolvimento que é constituída de artefatos que especificam as funcionalidades, e
- **Gestão:** atividade de planejamento, monitoramento e desenvolvimento do produto. Possui o artefato *product backlog* como entrada e gera os artefatos: estrutura analítica do projeto (EAP), planilha de custos, cronograma e planilha de controle.

A Figura 11 apresenta o processo do Scrum Solo. Comparativamente ao Scrum convencional, possui algumas diferenças como a falta de reuniões diárias e, quando necessário, reunião de validação com os clientes ou usuários finais para validação ([PAGOTTO et al., 2016](#)).



Figura 11 – Processo Scrum Solo, uma adaptação Scrum ([PAGOTTO et al., 2016](#))

[Pagotto et al. \(2016\)](#) define os papéis do Scrum Solo, os quais são:

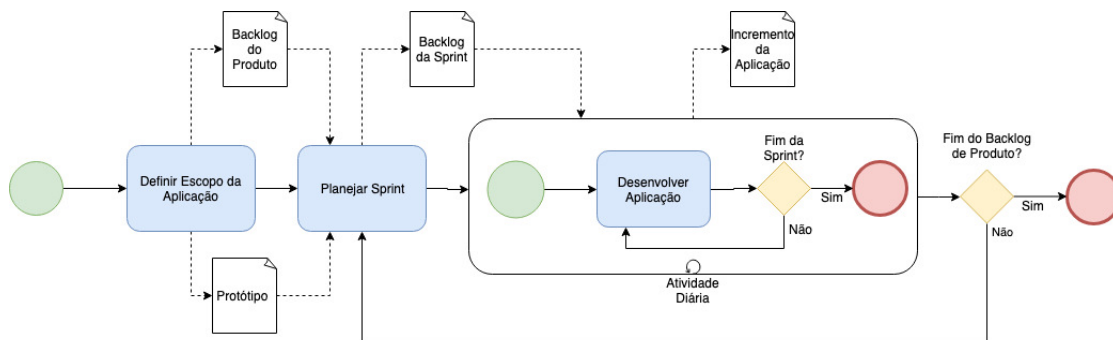
- **Product Owner:** interage diretamente com a solução. Nesse trabalho, o *product owner* tem relação com o grupo de validação e o autor desse trabalho à partir dos

levantamentos bibliográficos. Por isso, o *product owner* é constituído pelo grupo de validação e o autor dessa monografia. Pagotto et al. (2016) afirma que o *product owner* pode ser constituído como um grupo de indivíduos;

- **Desenvolvedor Individual:** responsável pelo desenvolvimento do software e é constituído pelo autor desse trabalho;
- **Orientador:** possui uma ampla visão do processo, bem como tem conhecimento das tecnologias e do escopo. Esse papel é exercido pela Orientadora e pelo Coorientador desse trabalho, e
- **Grupo de Validação:** usuários do produto gerado participam da validação do produto desenvolvido. Nesse trabalho, são caracterizados como os médicos patologistas, especialistas em anatomia patológica.

Com base no Scrum e no Scrum Solo, foi definida uma adaptação de processo de desenvolvimento para esse trabalho, apresentado na Figura 12.

Figura 12 – Processo de desenvolvimento adaptado do Scrum e Scrum Solo.



Fonte: Autor.

As atividades que constituem o processo de desenvolvimento são descritas como:

1. **Definir Escopo da Aplicação:** a definição do escopo da aplicação caracteriza-se pelo levantamento bibliográfico para a definição dos objetivos do software, bem como as opiniões de médicos patologistas para o desenvolvimento de um software de digitalização de lâminas histopatológicas. Tem como saídas os artefatos *backlog do produto* e o protótipo do software;
2. **Planejar Sprint:** nessa atividade, foram selecionados itens do *product backlog* para o desenvolvimento da *sprint* de uma semana, e
3. **Desenvolver Aplicação:** o desenvolvimento da aplicação aconteceu em um ciclo diário durante a *sprint* de duas semanas. As funcionalidades desenvolvidas na *sprint*

vieram das histórias de usuário planejadas no *sprint backlog*. Essa atividade gera um incremento do software.

Os artefatos da metodologia de desenvolvimento adotada são:

- **Product Backlog:** é constituído pelos requisitos gerais do software. Nesse trabalho, os requisitos são especificados, como anteriormente mencionado, em histórias de usuário;
- **Sprint Backlog:** seleção das histórias de usuário que serão desenvolvidas durante a *sprint* de uma semana, e
- **Incremento da Aplicação:** o software funcional com as funcionalidades anteriormente desenvolvidas e com as da *sprint* recém finalizada.

4.4 Metodologia de Coleta e Análise de Resultados

A coleta de resultados e a análise desses resultados visaram responder à pergunta inicial da pesquisa (GERHARDT; SILVEIRA, 2009). Esse trabalho possui um caráter aplicável, no qual o levantamento bibliográfico teve o intuito de familiarização com o tema para uma prática, que é o desenvolvimento de um software.

Pesquisa-ação: como discutido na subseção 4.1.4, a pesquisa-ação é adotada como procedimento técnico, por meio de cenários de uso, visando utilizar do seu caráter cíclico para:

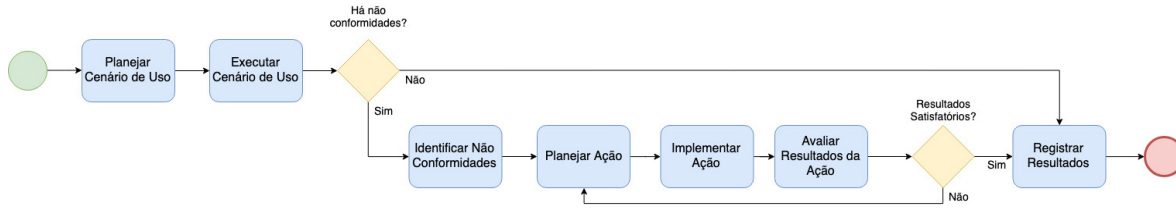
1. Identificar não conformidades;
2. Planejar ação para a melhoria das não conformidades;
3. Implementar ação planejada, e
4. Avaliar os efeitos da ação;

As atividades do processo de análise de resultados são apresentadas na Figura 13.

As atividades para o processo de análise de resultados são descritas como:

- **Planejar Cenário de Uso:** planejar o objetivo do cenário de uso e a forma como será executado;
- **Executar Cenário de Uso:** executar o cenário de uso planejado para identificação de não conformidades;

Figura 13 – Processo de análise de resultados



Fonte: Autor

- **Identificar Não Conformidades:** identificar não conformidades com o intuito de realizar ações para melhoria;
- **Planejar Ação:** levantar possíveis causas da não conformidade e planejar a ação para melhoria, também executada como replanejamento de uma ação que não há os resultados esperados;
- **Implementar Ação:** executar ação para a melhoria da não conformidade;
- **Avaliar Resultados da Ação:** verificar se os resultados são satisfatórios como o planejado, e
- **Registrar Resultados:** caso os resultados sejam os esperados, são então registrados no trabalho.

4.5 Cronograma

O TCC, segundo as regras da UnB no campus FGA, é realizado em duas fases: TCC 1 e TCC 2. O TCC 1 tem por objetivo estabelecer a base teórica do trabalho, enquanto o TCC 2 tem como foco atingir o objetivo estabelecido que é “*prover uma solução computacional WSI, acessível em termos de custo, utilizando-se de dispositivos comuns (i.e. smartphones e microscópios de laboratório-rios de anatomia patológica), e facilitando a troca de dados entre outras soluções WSI, o objetivo geral desse trabalho é desenvolver um suporte orientado ao padrão DICOM para o domínio de patologia digital*” baseando-se nas fundamentações teóricas.

A Tabela 2 apresenta o cronograma de atividades realizadas para o desenvolvimento do TCC 1.

Por sua vez, a Tabela 3 apresenta o cronograma para o desenvolvimento do TCC 2.

Tabela 2 – Cronograma de Atividades para o TCC 1.

Atividades	Abr/2019	Mai/2019	Jun/2019	Jul/2019
Introdução	X			
Referencial Teórico	X			
Suporte Tecnológico		X		
Metodologia		X		
Prova de Conceito			X	
Proposta			X	
Status do Trabalho			X	
Finalização Geral da Monografia			X	
Prévia			X	
Apresentação à Banca				X

Fonte: Autor.

Tabela 3 – Cronograma de Atividades para o TCC 2.

Atividades	Mar/2021	Abr/2021	Mai/2021
Coletar Amostras de Imagens Histopatológicas	X		
Implementar Algoritmo de Stitching	X		
Desenvolver Aplicativo e API	X	X	
Analisar Resultados			X
Apresentar TCC 2			X

Fonte: Autor

4.6 Resumo do Capítulo

Esse capítulo teve o intuito de apresentar toda a metodologia do presente trabalho, sendo classificado como:

- **Abordagem:** qualitativa;
- **Natureza:** pesquisa aplicada;
- **Objetivos:** pesquisa exploratória e,
- **Procedimentos Técnicos:** pesquisa-ação e pesquisa bibliográfica.

Também apresentou o fluxo de atividades do trabalho, a metodologia de desenvolvimento da aplicação, sendo uma adaptação do Scrum e Scrum Solo, e a metodologia de análise de resultados como pesquisa-ação aplicada a cenários.

5 Prova de Conceito

Esse capítulo tem como objetivo demonstrar a aplicação prática do conceito de *Image Stitching* apresentado na Seção 2.4.3 do capítulo 2 de Referencial Teórico, apresentando a possibilidade do desenvolvimento da solução em software para composição de imagens histopatológicas. Esse capítulo contém apenas uma Seção, 5.1, sendo essa focada em *Image Stitching*. Por fim, tem-se o Resumo do Capítulo.

5.1 *Image Stitching*

Para demonstrar a prática do *Image Stitching* em imagens histopatológicas, comparou-se a composição de imagens entre a imagem composta por meio dos algoritmos SURF, SIFT e ORB e a imagem original. O método para essa comparação seguiu as seguintes fases:

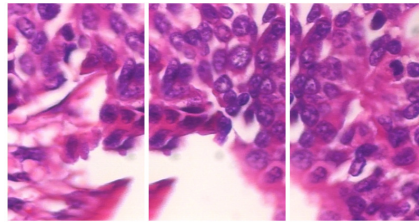
1. **Seleção de Imagens:** nessa primeira fase, foram selecionadas cinco imagens histopatológicas do banco BreakHis (SPANHOL et al., 2016);
2. **Recorte das Imagens:** as imagens escolhidas foram recortadas de modo aleatório, horizontalmente e verticalmente;
3. **Execução do Algoritmo de Image Stitching:** as imagens recortadas são executadas pelo algoritmo desenvolvido na linguagem python, utilizando a biblioteca de visão computacional OpenCV com os algoritmos SURF, SIFT e ORB (OPENCV, 2019b), e
4. **Comparação:** as imagens construídas a partir dos recortes são comparadas com as imagens originais usando distância euclidiana.

A Figura 14 apresenta três fragmentos de uma imagem histopatológica recortada. A seguir, a Figura 15 apresenta a imagem composta a partir dos fragmentos anteriormente citados.

Como considerações finais, demonstrou-se que a acurácia entre os três algoritmos não revelou diferenças significantes. Entretanto, no tempo de execução, o algoritmo ORB obteve melhor desempenho, o que concorda com o que foi apresentado na subseção 2.4.4.1 do Capítulo de Referencial Teórico.

A Figura 49 apresentada no apêndice A representa o fluxograma do algoritmo de *Image Stitching* utilizado e referenciado nesse capítulo. Tal algoritmo baseia-se no estudo de Vyas (2019)

Figura 14 – Lâmina histopatológica de um Carcinoma Ductal, ampliada em 400x, e utilizada no estudo comparativo.



Fonte: (SPANHOL et al., 2016)

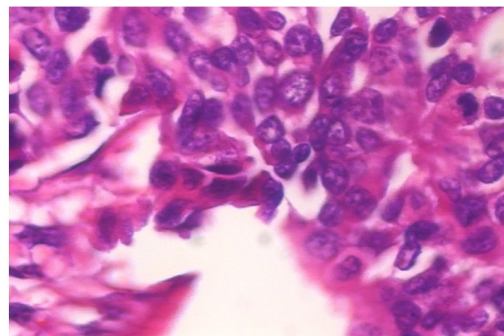


Figura 15 – Lâmina histopatológica de um Carcinoma Ductal gerada a partir dos fragmentos

5.2 Resumo do Capítulo

Nesse capítulo, foi apresentada uma aplicação que procurou explorar os conceitos teóricos apresentados na seção 2.4.3. Essa aplicação teve como objetivo a comparação de desempenho dos algoritmos SIFT, SURF e ORB para a técnica de *Image Stitching*, e a medição das similaridades das imagens geradas com as originais. Como apresentado no capítulo, o trabalho concluiu que o algoritmo ORB possui melhor desempenho em comparação aos outros algoritmos aplicados e a acurácia entre a imagem original e a gerada não revelou diferenças significantes.

6 Mosaic

Esse capítulo objetiva descrever o produto de software desenvolvido nesse trabalho, denominado Mosaic. Na Seção 6.1, é apresentada uma breve contextualização bem como são acordadas algumas soluções similares. A visão arquitetural do Mosaic é conferida na Seção 6.2. Por fim, tem-se o Resumo do Capítulo, na Seção 6.8, acerca dos principais pontos debatidos ao longo do capítulo.

6.1 Contextualização

Mosaic é um sistema, o qual permite utilizar ferramentas acessíveis para digitalizar lâminas histopatológicas pelo método WSI, bem como adequar essas imagens ao padrão DICOM. Entretanto, visto que o método de adquirir imagens pode ser trabalhoso, sendo necessárias várias imagens para compor uma imagem digitalizada de uma lâmina inteira, o sistema Mosaic especializa-se em formar imagem de uma área de interesse da lâmina que não ultrapasse a quantidade de nove fragmentos que a representem.

A solução é similar ao trabalho de [Ma e Wang \(2017\)](#), ao produto Panoptiq[®] ([MEYER, 2019](#)) da empresa Meyer[®] e ao *manual WSI*[®] ([MICROVISIONEER, 2019](#)) da empresa Microvisioneer[®]. Esses dois últimos não utilizam um *smartphone* para a digitalização, diferentemente do sWSI de [Ma e Wang \(2017\)](#). Essas soluções também não deixam clara a utilização do padrão DICOM para interoperabilidade das imagens.

A solução desenvolvida nesse trabalho consistiu em:

1. **Aplicativo:** um aplicativo para *smartphones* iPhone[®] que opere na versão iOS versão 11 ou superior que, ao ser acoplado ao microscópio por meio de um adaptador, captura as imagens do campo de visão do microscópio, e
2. **Aplicação API:** uma aplicação API para operacionalizar o sistema.

Na primeira versão deste trabalho, considerou-se pertinente desenvolver uma aplicação *WEB*, a qual atuaria como um software visualizador para os arquivos DICOM gerados. Entretanto, após pesquisas de soluções, com estudos na literatura especializada, verificou-se alguns serviços gratuitos, os quais operam em nuvem e possuem esse objetivo, tal como ocorre com o PostDicom *PostDicom* ([POSTDICOM, 2021](#)). Dessa forma, considerou-se focar as principais contribuições do presente trabalho na geração de imagem bem como na adequação para o formato DICOM. Ainda na primeira versão, considerou-se adicionar as funções de consulta *Query based on ID for DICOM Objects* QIDO-RS e *Web*

Access to DICOM Objects WADO-RS. Entretanto, esse tipo de consulta não se adequaria ao requisito da aplicação em que o usuário cadastra os *studies* e são relacionados a ele, fazendo com que as consultas sejam baseadas apenas em um único *endpoint*, o qual retorna todos os *studies* e seus atributos. Adicionalmente, a especificação para detalhar um determinado *study* ou pesquisá-lo é solicitada na própria aplicação, sem ser necessária uma requisição para isso.

6.2 Arquitetura do sistema

A visão arquitetural utiliza-se do modelo *C4 Model*, proposto por Brown (2019). Esse modelo divide a arquitetura em quatro níveis:

1. **Contexto:** é o ponto de partida para o desenho da arquitetura, dando uma visão ampla das pessoas e outros sistemas envolvidos com o sistema de software (BROWN, 2019). O sistema de software que se deseja modelar é uma caixa central do diagrama (BROWN, 2019). Ainda segundo (BROWN, 2019), nesse nível, os protocolos, as tecnologias e outros detalhes não são relevantes, sendo o foco, na verdade, a representação da interação entre os atores e os sistemas externos, o que revela todos os envolvidos, sendo essa a audiência de apresentação;
2. **Container:** após a construção do nível mais alto, o contexto, o nível de detalhe é ampliado, atingindo um nível mais técnico (BROWN, 2019). Nesse nível, são apresentados os *containers* do sistema, que são definidos como qualquer unidade que executa um código ou persiste dados, como aplicativo para *smartphone*, aplicação *desktop*, aplicação web, entre outros (BROWN, 2019). Todo o time de desenvolvimento é a principal audiência para o entendimento desse nível (BROWN, 2019);
3. **Componente:** nesse nível, os *containers* são detalhados, apresentando as interações e tecnologias que operam dentro deles (BROWN, 2019). Tal nível possui como foco a apresentação para profissionais de arquitetura de software e desenvolvedores (BROWN, 2019), e
4. **Código:** esse último nível é opcional (BROWN, 2019). Trata-se da última instância de detalhe, na qual são apresentadas como os componentes são implementados em nível de código (BROWN, 2019). Nesse nível, portanto, são definidos diagramas de classe, diagramas entidade-relacionamento, dentre outros (BROWN, 2019).

Não há uma notação definida para o modelo *C4 Model*. Porém, os diagramas são similares. Esse trabalho utilizou a notação *C4 Model* disponível para o software *Draw.io*. No Apêndice B, mais especificamente na Figura 50, há detalhamento sobre a notação utilizada.

6.2.1 Contexto

No nível de contexto, Figura 16, há um ator, também referenciado como *persona*, que utiliza o sistema Mosaic, sendo esse um sistema PSI.

- **Médico Patologista:** é o médico com a especialidade na área de anatomia patológica e que tem contato direto com a lâmina histopatológica ampliada no microscópio para análise, e
- **Mosaic:** sistema desenvolvido por esse trabalho que permite digitalizar uma parte de uma área da lâmina. Esse sistema está detalhado na subseção 6.2.2.

Figura 16 – Nível 1 (Contexto)



Fonte: Autor.

6.2.2 Container

Esse nível contém o detalhamento dos *containers* que compoem o sistema, seus relacionamentos e suas tecnologias. Os *containers* são apresentados na Figura 17, e descritos a seguir:

- **Aplicativo iOS®:** aplicativo desenvolvido em linguagem de programação *Swift* utilizando o UIKit, sendo esse um kit de desenvolvimento nativo para iOS®. Conforme já apresentado no Capítulo 3 de Suporte Tecnológico, esse kit é responsável pela captura das imagens da lâmina histopatológica ampliadas no microscópio;
- **Aplicação API:** interface para prover dados e persisti-los, seguindo o padrão DICOMWeb™ para esta persistência, que se refere ao STOW-RS;

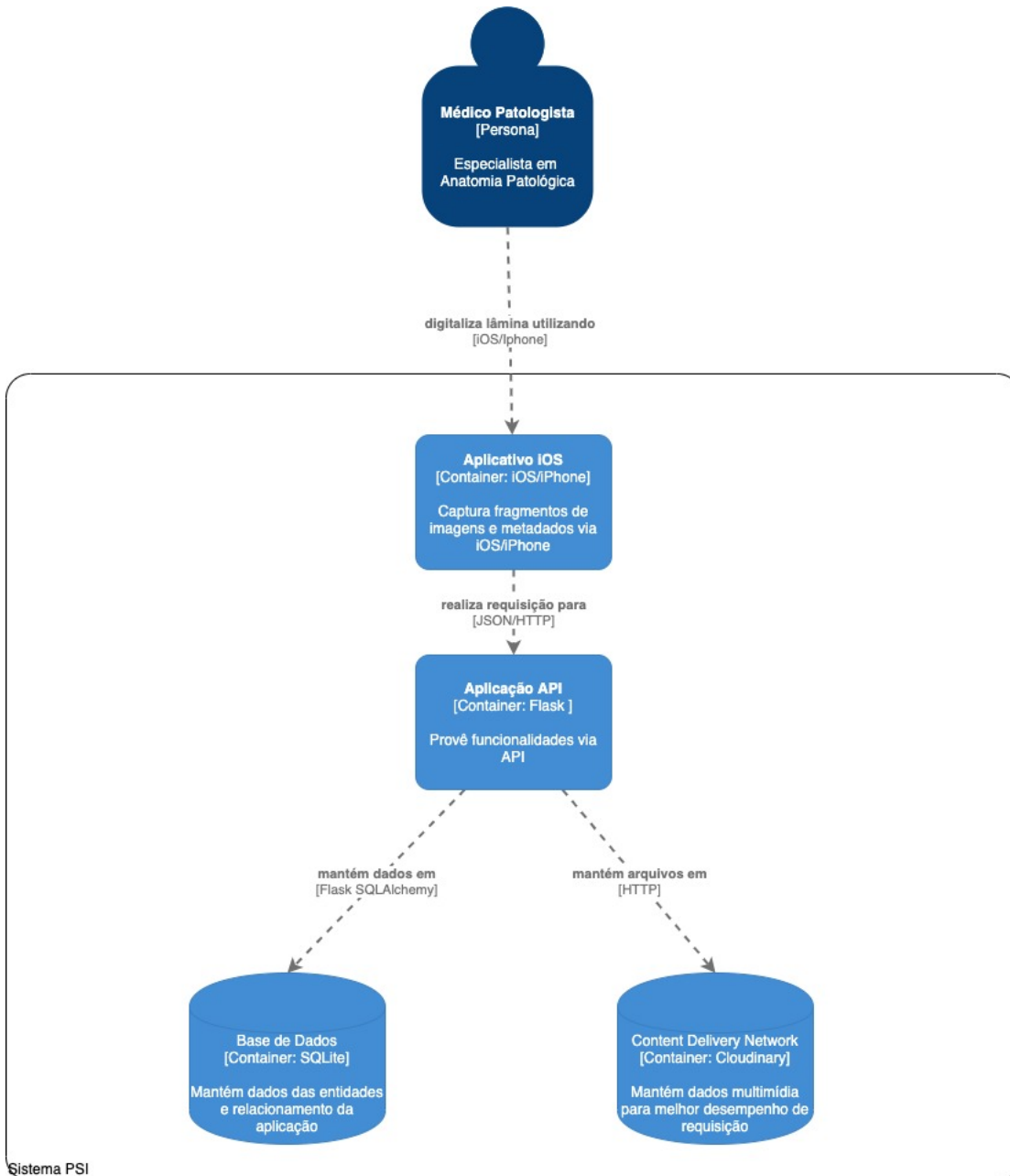


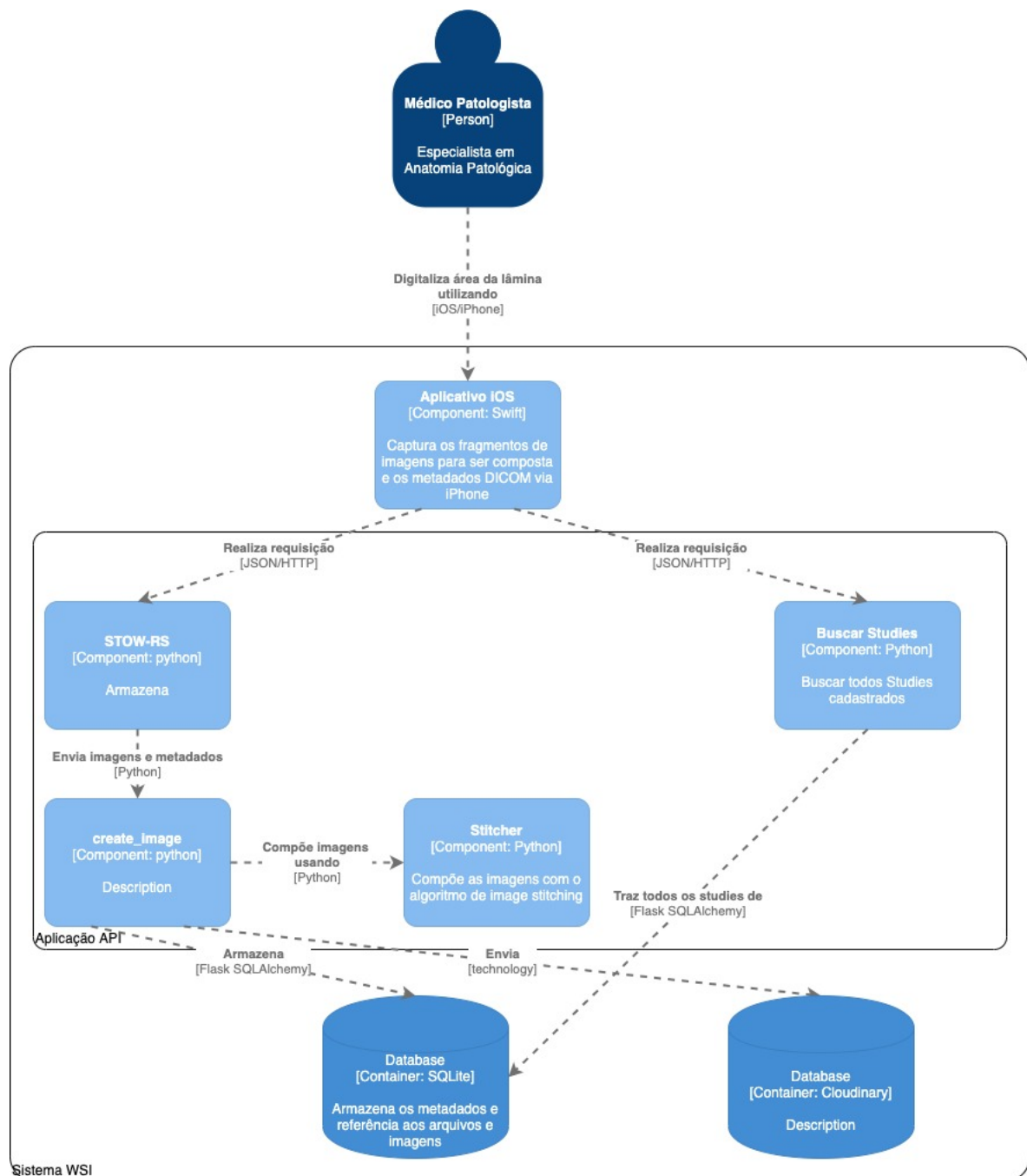
Figura 17 – Nível 2 (Container)

- **Banco de Dados:** permite a persistência dos dados e seus relacionamentos, sendo utilizado o banco de dados SQLite, e
- **Content Delivery Network:** o *Content Delivery Network* Cloudinary[®] armazena as imagens para que se possa ter acesso com melhor desempenho.

6.2.3 Componente

A API possui três componentes baseados no serviço descrito no DICOMWeb[™]. A Figura 18 apresenta o diagrama com o *container* da API detalhado.

Figura 18 – Nível 3 (Componente)



Fonte: Autor

Os componentes da API são descritos como:

- **STOW-RS:** responsável por armazenar a imagem e metadados;
- **Buscar *Studies*:** armazena uma *instance*, e
- ***Create Image*:** módulo que compõe a imagem a partir dos fragmentos, resultando na imagem composta na extensão jpg e a imagem DICOM na extensão dcm com

seus metadados, e

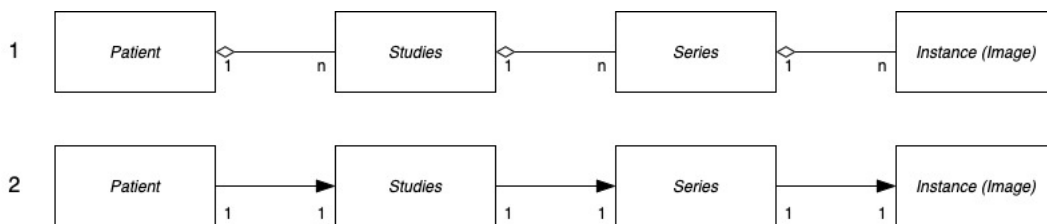
- **Stitcher**: módulo responsável por compor as imagens utilizando a técnica de *Image Stitching*. Esse módulo é utilizado pelo módulo *Create Image*.

6.2.4 Código

O nível de código foi representado a partir de um diagrama UML adaptado de [Herrmann et al. \(2018\)](#). No caso, uma abstração do mundo real das entidades DICOM para imagens *SC Image*. No projeto, o usuário do sistema gera um arquivo DICOM por paciente, que corresponde à imagem da área da lâmina composta a partir de seus fragmentos. Dessa forma, a aplicação funciona com um relacionamento diferente do contexto dos relacionamentos DICOM para se adequar aos requisitos levantados para esse trabalho. Entretanto, o banco de dados é modelado de maneira que seja adequado ao padrão DICOM.

A representação entre os relacionamentos das entidades no padrão DICOM e como é apresentado na aplicação *mobile* é apresentada na Figura 19, a qual manteve os nomes da entidade em inglês, pois o *software* teve seu código escrito em inglês, bem como as entidades mantendo os nomes originais, também em inglês.

Figura 19 – Representação *DICOM Data Model SCImage*. (1) Representação geral de um objeto DICOM. (2) Representação de como os relacionamentos foram utilizados no contexto do projeto Mosaic. As entidades foram mantidas em inglês a nível de código, bem como as entidades, mantendo o idioma na Figura.



Fonte: Autor.

Na verdade, a Figura 19 ilustra duas representações no DICOM para as entidade *DICOM Data Model*, a qual representa as entidades e os relacionamentos aplicados em um contexto real, definidos a partir do diagrama do padrão DICOM para espécimes, apresentado na Figura 58 presente no Anexo A, e adaptado do trabalho de [Herrmann et al. \(2018\)](#).

6.3 Módulo de *Stitching*

O módulo de *stitching* utilizado no sistema possui seu algoritmo baseado no SURF. Em primeiro momento, como proposta, pensou-se em utilizar o ORB. Entretanto, o ORB

apresentou problemas para realizar o *stitching* das imagens que são relacionadas a ruídos, já que o algoritmo ORB, apesar de ter um desempenho superior, possui problemas em detecção e descrição de características com imagens que possuam ruídos (CORTÉS-GALLARDO et al., 2019)

A comparação entre o ORB e o SURF encontra-se no Capítulo de Análise de Resultados, Seção 7, na subseção 7.1.2 do segundo cenário de uso.

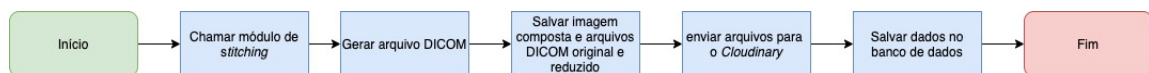
6.4 Módulo de Criação Imagem

O módulo de criação de imagem é responsável por todo o fluxo de recebimento dos fragmentos e metadados. Utilizando o módulo de *stitching*, ele gera o arquivo DICOM com seus metadados e a imagem composta. Por fim, ocorre o salvamento das informações no banco de dados.

Localmente, o módulo cria uma imagem, que é a imagem composta a partir dos fragmentos e dois arquivos DICOM, sendo: um considerado como original, e outro reduzido. O reduzido possui a imagem composta menor, em termos de resolução, para que seja possível fazer o *upload* para o CDN.

O fluxo do funcionamento desse módulo utilizado pela API é demonstrado na Figura 20.

Figura 20 – Fluxo do módulo de criação de imagem.



Fonte: Autor.

6.4.1 Aplicação API

O servidor foi desenvolvido utilizando a linguagem *Python* com o *framework* *WEB Flask* de escopo local, com a responsabilidade de operacionalizar o sistema a partir do cliente. Nesse trabalho, o cliente é o aplicativo iOS. Considerou-se primeiro utilizar o *framework* Django para desenvolver o sistema *Backend*. Entretanto, o Flask mostrou-se uma tecnologia mais coerente com o escopo do projeto Mosaic, sendo esse de pequeno porte, e não justificando o uso do *framework* Django. O sistema *Backend* faz uso do módulo de *stitching*, para que seja gerada a imagem composta, os arquivos DICOM original e o reduzido, enviados para o CDN *Cloudinary*. O *endpoint* de armazenamento segue o padrão DICOMWeb STOW-RS. Há também um *endpoint*, o qual é responsável por retornar para o cliente os *studies* cadastrados pelo usuário. A Tabela 4 apresenta os *endpoints* da API e suas características.

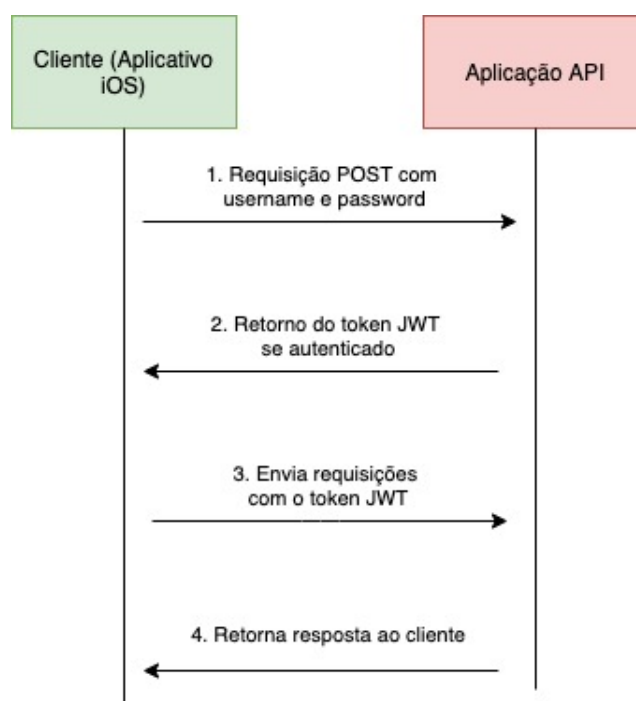
Tabela 4 – *Endpoints* que a aplicação API dispõe para operacionalização de aplicações cliente.

Endpoint	Método HTTP	Retorno	Descrição
/login	POST	<i>Token</i>	Efetua <i>login</i> de usuário cadastrado.
/signup	POST	Dados do usuário cadastrado	Cadastra usuário.
/studies	POST	Status 200	Gera imagem, arquivo DICOM e salva dados e referências no banco de dados.
/studies	GET	Studies	Retorna todos os studies cadastrados pelo usuário.

6.5 Segurança

A aplicação API utiliza o sistema JWT para que os dados não estejam dispostos abertamente, sendo necessário que o usuário tenha autenticação para ter acesso ao conteúdo da API, em que é gerado o *token* de acesso, o qual é responsável por identificar requisições autorizadas e o *refresh token* que é utilizado para atualizar o *token* de acesso quando este for expirado, sendo possível utilizar para acessar o sistema. A validade dos *tokens* foi configurada de forma que o *token* de acesso possui três semanas de validade; enquanto o *refresh token* possui seis semanas. O *payload* para a geração do *token* é o *username* e seu *id*, podendo ser útil para manutenção e evolução da API, caso seja necessário identificar o usuário que está atualmente utilizando o sistema.

A Figura 21 apresenta o fluxo JWT na aplicação do sistema, sendo o cliente o aplicativo iOS. Entretanto, cabe ressaltar que a API pode ser implementada em clientes desenvolvidos para outras tecnologias, como sistema *WEB*.

Figura 21 – Fluxo JWT no sistema de *software*.

Fonte: Autor.

6.6 Aplicativo iOS

O aplicativo iOS possui as seguintes responsabilidades no sistema Mosaic:

1. Capturar os fragmentos de imagens e preenchimento dos metadados DICOM, que nesse trabalho se adequa ao *SC Image*, utilizando apenas os dados obrigatórios;
2. Listar os *studies* gerados pelo usuário do sistema;
3. Visualizar detalhes do *study*, e
4. Compartilhar a imagem gerada em aplicativos de mensagens, *email* ou outros meios de comunicação disponíveis no dispositivo.

O aplicativo só possui sua plena funcionalidade a partir da autenticação do usuário, sendo possível entrar se já houver uma conta ou, caso não possua, poderá cadastrar no próprio aplicativo. A Figura 22 apresenta as duas telas de autenticação.

Para utilizar a função principal do aplicativo, que seria a escolha dos fragmentos de imagem para fazer a composição de uma área da lâmina, é necessário escolher no máximo nove fragmentos, de forma que essas imagens possam ser obtidas do álbum do dispositivo ou de uma fotografia em tempo real. Essas imagens devem ser em formato quadrado para que o *vignetting* da ocular do microscópio seja ignorado e obtenha um *stitching* da região

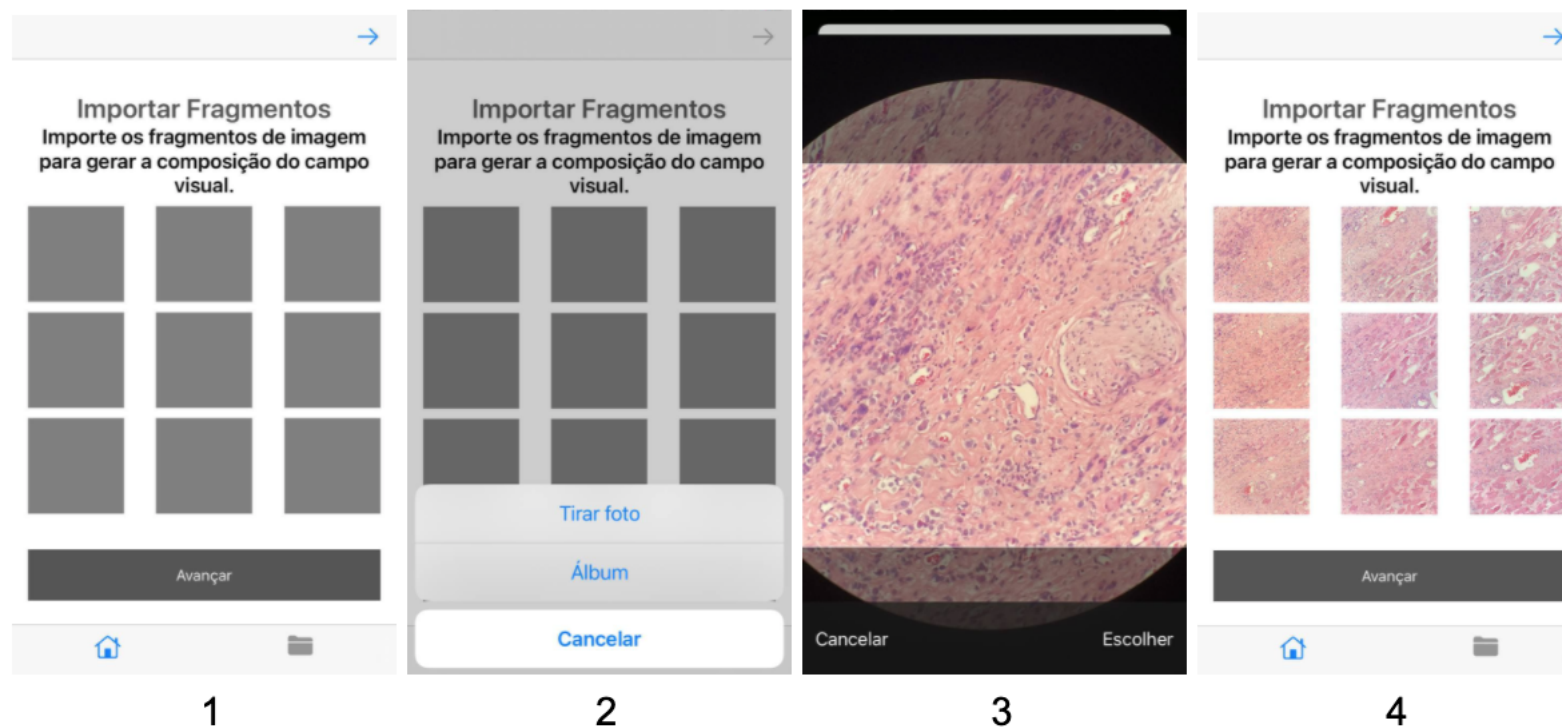
Figura 22 – Telas de autenticação no aplicativo iOS. (1) Tela de *login*. (2) Tela de cadastro.



Fonte: Autor.

observada. A Figura 23 apresenta as telas de *upload* dos fragmentos de imagem da área da lâmina.

Figura 23 – Telas de *upload* de fragmentos de imagem para o *Stitching*. (1) Tela de seleção de fragmentos de imagens. (2) Escolha da fonte dos fragmentos de imagem. (3) Escolha de fragmento de imagem a partir do álbum de fotos sendo editada para formato quadrado. (4) Tela de seleção de fragmentos de imagem com todos já selecionados.

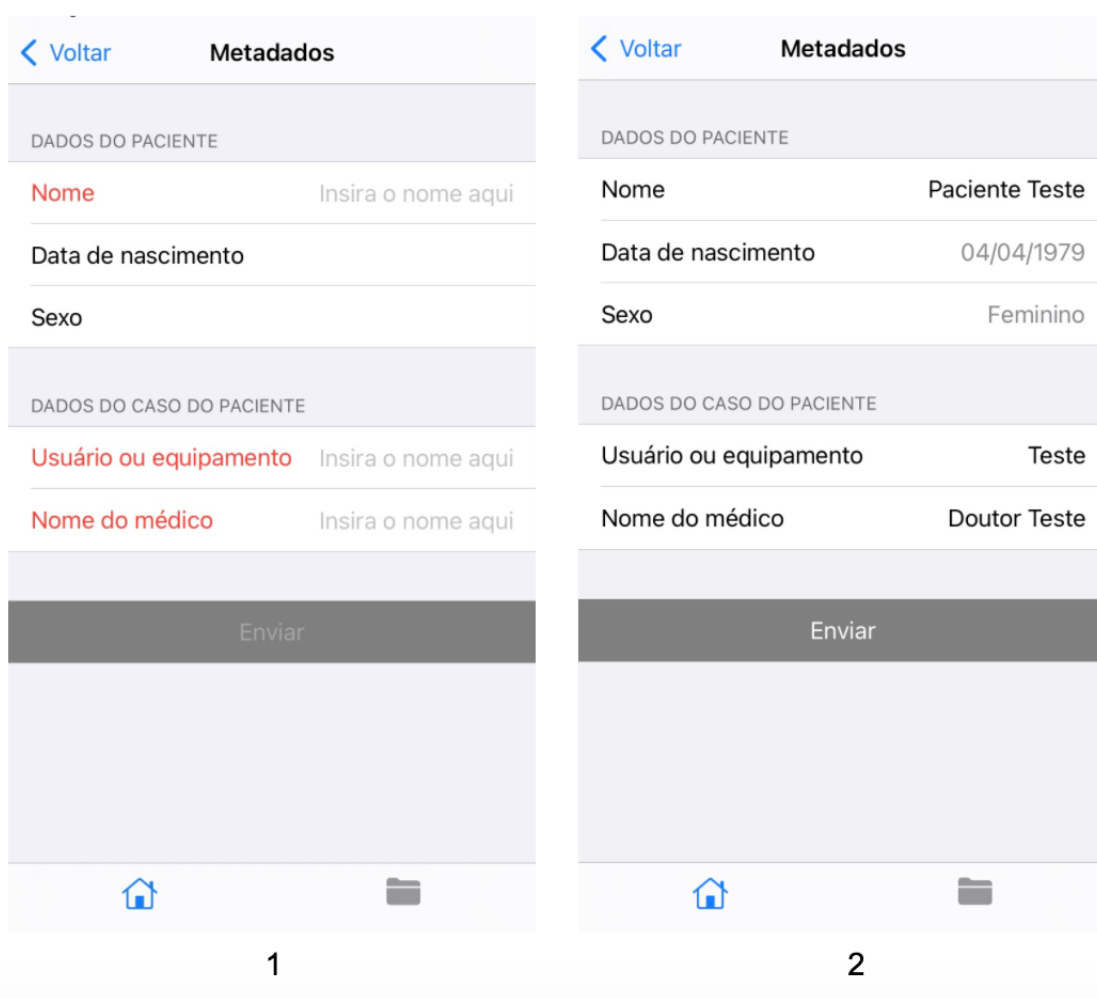


Fonte: Autor.

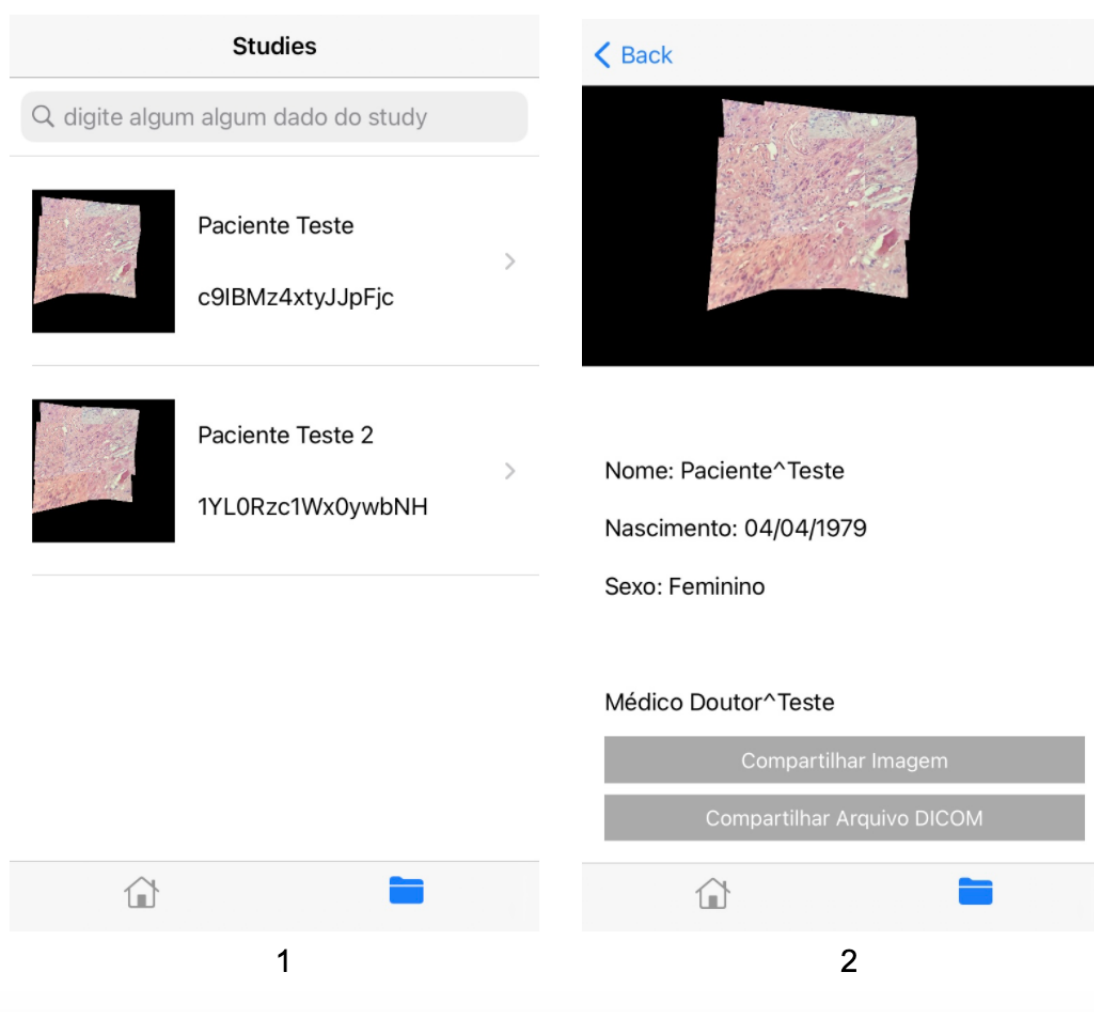
Uma imagem DICOM não possui apenas o conteúdo dos dados da imagem, mas também os metadados referentes ao IOD, que nesse trabalho se refere ao *SC Image*.

Após o usuário escolher os fragmentos de imagem da área a se fazer o *stitching*, segue-se para a tela de preenchimento dos metadados obrigatórios, sendo possível prosseguir apenas que todos os dados forem preenchidos. A tela de preenchimento dos metadados é apresentada na Figura 24.

Figura 24 – Tela de preenchimento dos metadados DICOM. (1) Tela sem preenchimento dos dados, impossibilitando prosseguir com o *upload* das informações. (2) Tela com os dados preenchidos, possibilitando prosseguir com o *upload dos dados*.



Os *studies* gerados pelo usuário são listados, podendo ser buscados por uma barra de pesquisa no topo da tela, possibilitando ao usuário pesquisar a partir de palavras-chave como nome do paciente, UID ou outro atributo de paciente. São apresentadas, na Figura 25, as telas do aplicativo da lista de *studies* e os detalhes correspondentes ao *study* selecionado, o qual possui a funcionalidade de compartilhar a imagem ou a imagem DICOM.

Figura 25 – *Studies* no aplicativo iOS. (1) Tela de *studies*.(2) Tela do detalhe de um *study*.

Fonte: Autor.

6.7 Configuração e Integração das Tecnologias

Essa subseção tem o objetivo de apresentar os passos para configurar e integrar as tecnologias a fim de viabilizar a execução da solução. Como descrito na no Capítulo 3 de Suporte Tecnológico, a aplicação API foi desenvolvida usando o *framework Flask*, escrito em Python, e o aplicativo foi desenvolvido para plataformas iOS, na linguagem Swift. Também foram utilizados sistemas como o CDN *Cloudinary* e *SQLite* para armazenamento de arquivos e dados, respectivamente.

As subseções seguintes apresentam as tecnologias e como configurá-las e integrá-las.

6.7.1 Flask Rest API

Para executar a API localmente, é necessário seguir os passos:

1. Configurar as variáveis de ambiente `FLASK_APP` para o arquivo da aplicação Flask e `FLASK_ENV` para *development*, e
2. executar o comando `flask run -host 0.0.0.0` para o servidor ser acessível na rede interna pelo cliente.

Dessa forma, a API será executada localmente, sendo possível observar os *logs* de chamada da API pelas aplicações cliente.

Para utilizar o *CDN Cloudinary*, é necessário criar uma conta e utilizar os dados de *api key*, *api secret* e *cloud name* na biblioteca do *Cloudinary* para *Python*. Assim, pode-se armazenar os arquivos na conta. O banco de dados *SQLite* é configurado a partir do *framework Flask SQLAlchemy*, sendo um banco local que irá servir para chamadas da API, possuindo referência de URL dos arquivos do *Cloudinary* e a relação de entidades.

6.7.2 Aplicativo iOS

A aplicação iOS é executada normalmente a partir da IDE *Xcode* em um dispositivo para que seja possível utilizar as imagens ou tirar foto. Para acessar a API a partir do dispositivo, configura-se o endereço local do *endpoint* a partir do endereço *ip* da rede interna da máquina que está executando o servidor *Flask*. Assim, é possível ter a comunicação entre o dispositivo e o servidor na rede local.

6.8 Resumo do Capítulo

Esse capítulo apresentou o software desenvolvido como PSI, chamado Mosaic, sendo uma solução que possui como suporte um aplicativo para *smartphones iPhone*[®] modelo SE 2020, com sistema operacional iOS versão 14, e uma aplicação API para operacionalização do sistema. A descrição do *software* foi complementada com a apresentação da arquitetura, configuração das tecnologias bem como detalhamento dos módulos do sistema, incluindo orientações quanto à segurança da solução.

Essa solução possui a capacidade de compartilhamento de áreas de interesse de uma lâmina composta a partir dos fragmentos de imagens, gerando também arquivo DICOM para utilização em diversos sistemas que estão de acordo com o protocolo. O sistema gera dois arquivos DICOM, sendo um original e outro reduzido, que é utilizado para fazer o *upload* do arquivo para o *Cloudinary*, já que este possui um limite de *upload* de arquivo que contém 20 *megabytes* de tamanho.

7 Análise dos Resultados

Esse capítulo objetiva apresentar os resultados do desenvolvimento do produto de software Mosaic. A Seção 7.1 descreve os cenários de uso seguindo a metodologia de pesquisa-ação detalhada na Seção 4.4 do Capítulo de Metodologia. A Seção 7.2 descreve o processo de desenvolvimento da solução Mosaic. Na sequência, a Seção 7.3 apresenta a validação do especialista, o qual é um médico patologista, a partir de um questionário. Por fim, tem-se o Resumo do Capítulo.

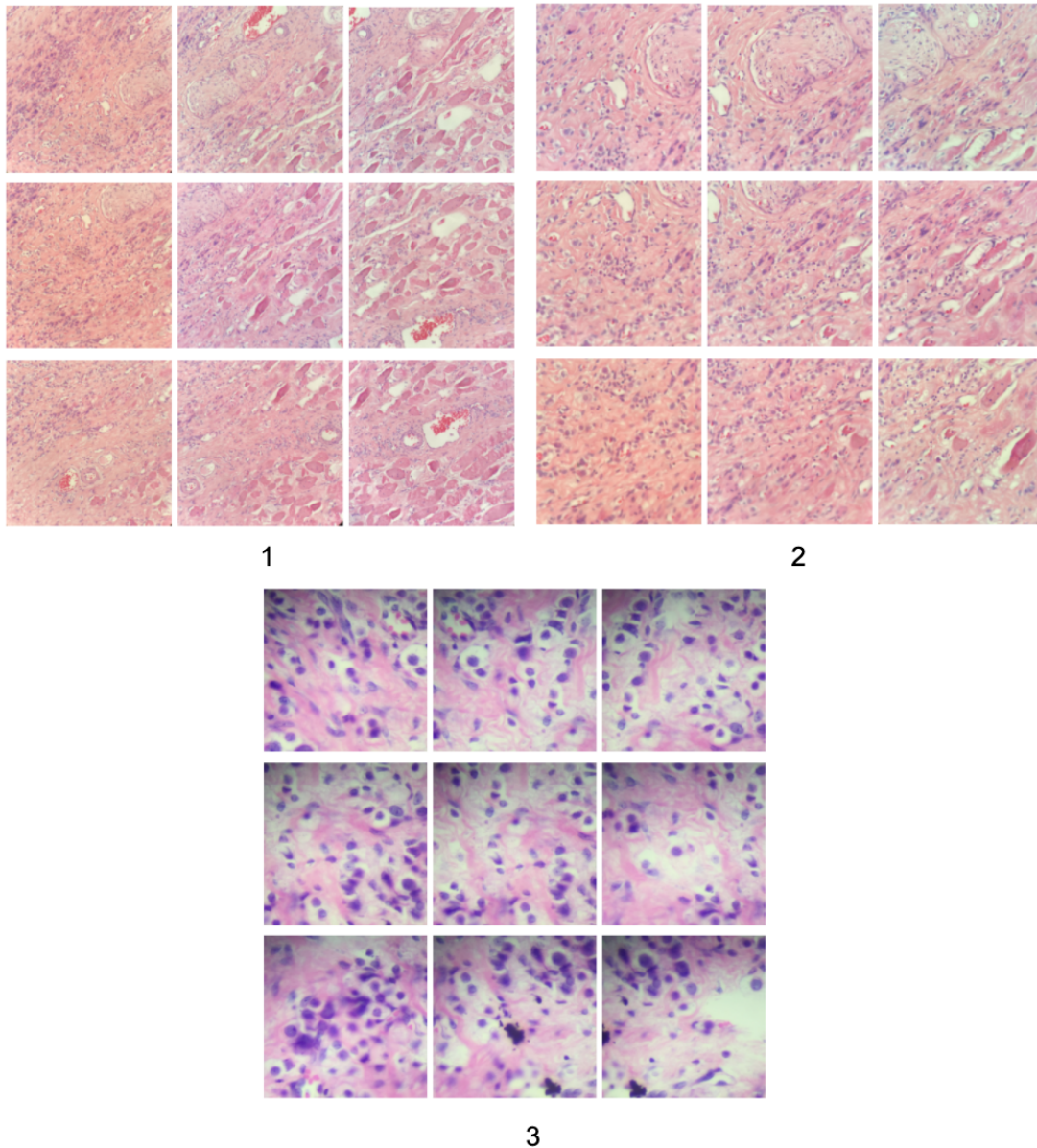
7.1 Cenários de Uso

Esta seção apresenta a coleta de dados, identificação de não conformidades, planos de ação e seus resultados, a partir dos cenários de uso, seguindo o planejamento do ciclo de pesquisa-ação definido no Capítulo 4.

Para alguns cenários de uso, utilizou-se amostras de imagens de uma lâmina, dentre três disponíveis, contendo uma amostra tecidual de tumor de mama pós quimioterapia, ampliadas em 200x e 400x, sem identificação para sigilo do paciente e cedida para o trabalho pelo laboratório Diagnose, localizado em Brasília. Também utilizou-se uma amostra de nove imagens de uma lâmina diferente, ampliadas em 400x, a qual obteve-se as amostras de imagens utilizando o adaptador que acopla o *smartphone* na ocular do microscópio. A Figura 26 apresenta as três amostras utilizadas para os cenários de uso.

O primeiro cenário de uso, descrito na Seção 7.1.1, concentrou-se em verificar o comportamento do algoritmo de *image stitching* nas imagens adquiridas pelo *smartphone* sem cortes, apresentando o *vignetting*, que é a área escura que circunda a imagem da lâmina, capturada de forma circular por conta da ocular do microscópio, causada pelo formato retangular da câmera. O segundo cenário de uso, apresentado na Seção 7.1.2, verifica-se a viabilidade de digitalização de uma lâmina completa no microscópio, capturando as imagens sequencialmente. Na Seção 7.1.3, é descrito o terceiro cenário de uso em que se analisou o fluxo de captura em amostras diferentes de imagens e suas similaridades a partir de histogramas de intensidade. O adaptador que acopla o *smartphone* ao microscópio foi analisado na Seção 7.1.4. A validação do arquivo DICOM, utilizando o *DICOM Validator*, é descrito na Seção 7.1.5 do cenário de uso 5, e a visualização do arquivo DICOM, validado através do serviço em nuvem PostDICOM, é verificado e analisado no cenário de uso 6, na Seção 7.1.6. Por fim, na Seção 7, analisa-se a utilização do CDN *Cloudinary* para armazenar as imagens e arquivo DICOM.

Figura 26 – Amostras utilizadas para os cenários de uso. (1) Amostra de nove imagens ampliadas em 200x. (2) Amostra da mesma lâmina (1), ampliada em 400x. (3) Amostra de lâmina diferente das anteriores, ampliada em 400x.



Fonte: Autor.

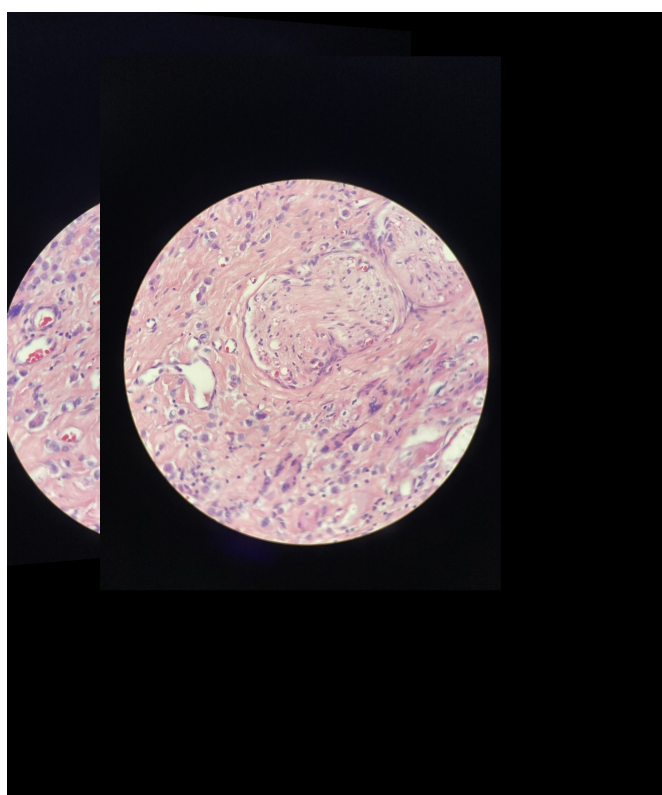
7.1.1 Cenário de Uso 1

O primeiro cenário de uso concentrou-se em observar o comportamento do algoritmo de *stitching* nas imagens originais capturadas do microscópio, de forma que preservasse ainda o *vignetting* causado pela ocular na câmera retangular do *smartphone*. Para isso, selecionou-se duas imagens, nas quais haviam sobreposição entre elas, entre as 9 imagens de uma determinada área.

7.1.1.1 Coleta e Análise de Dados

O *stitching* das imagens originais tiveram sua matriz de homografia e correspondência de *keypoints* adequadas. Entretanto, por conta da área preta que circunda a imagem de interesse da lâmina, o algoritmo *stitching* sobrepôs as duas imagens, já que não há a exclusão desta área. A Figura 27 apresenta o resultado do *stitching* utilizando as duas imagens escolhidas.

Figura 27 – Ocular do microscópio utilizado com sua marcação de ampliação e número de campo.



Fonte: Autor.

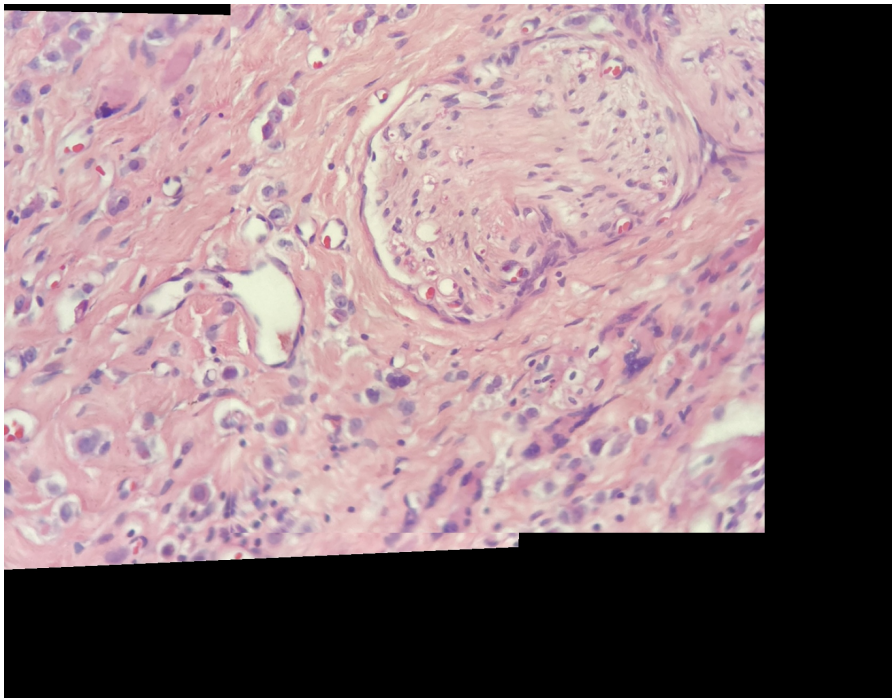
7.1.1.2 Plano de Ação

Para gerar a imagem utilizando como referência apenas a área da lâmina, fez-se uso da técnica de [Morrison e Gardner \(2014\)](#), que consiste em ampliar a câmera do *smartphone* para eliminar a área em volta da circunferência que contém a imagem da lâmina. Dessa forma, concentrou-se em utilizar o formato quadrado para editar a imagem de forma que desprezasse a parte externa da circunferência, tornando-se um quadrado inscrito na circunferência.

7.1.1.3 Resultados da Ação

A partir da utilização da edição em formato quadrado, inscrito na circunferência que contém a área de interesse, que é a imagem da lâmina a partir da ocular, obteve-se a composição das imagens de forma esperada, em que não haveria a presença da área circundante à circunferência e que apresentasse um *stitching* contínuo da sobreposição das duas imagens. A Figura 28 apresenta a imagem gerada a partir da edição em formato quadrado inscrito na circunferência da área da lâminas das imagens originais.

Figura 28 – Imagem composta a partir de dois fragmentos em formato quadrado.



Fonte: Autor.

Entretanto, a imagem apresenta perda de área da imagem da lâmina, já que a imagem composta agora provém em formatos quadrados obtidos de forma inscrita à área. Para obter o quanto de área foi perdida, calculou-se a área do campo de visão em determinada ampliação, que nessa imagem foi de 400x, sendo a objetiva em 40x e a ocular em 10x, resultando em 0,05mm, equivalente a $50\mu\text{m}$, permitindo calcular a área circular como $1.963,50\mu\text{m}^2$. Sabendo que o lado de um quadrado inscrito na circunferência é dado por $l = d\sqrt{2}$, em que l é o lado do quadrado e d o diâmetro da circunferência, que é igual a 0,05mm, como calculado anteriormente no campo de visão, obteve-se a área de $1.250,00\mu\text{m}^2$, representando uma perda de aproximadamente 36,33%, considerando ser um quadrado inscrito perfeito.

$$FoV = \frac{\text{número de campo}}{\text{ampliação}} \quad (7.1)$$

em que FoV é o campo de visão, resultando da razão entre o número de campo e a ampliação. Esses valores são encontrados na ocular do microscópio no formato ampliação/número de campo (FLOURNOY, 2018). A Figura 29 apresenta a configuração do campo de visão marcado na ocular do microscópio, que é $10x/20$, sendo $10x$ a ampliação e 20 o número de campo (FLOURNOY, 2018). A Figura 29 apresenta esses valores indicados na ocular do microscópio utilizado.

Figura 29 – Ocular do microscópio utilizado com sua marcação de ampliação e número de campo.



Fonte: Autor.

7.1.2 Cenário de Uso 2

Este cenário de uso teve como objetivo verificar se seria praticável digitalizar uma lâmina completa. Para que fosse possível verificar a viabilidade de digitalizar uma lâmina inteira, calculou-se o campo de visão a partir da ampliação e do número de campo para ter o resultado do espaço em milímetros do campo de visão, utilizando a equação 7.1, a fim de obter a quantidade de imagens necessárias para a digitalização completa.

7.1.2.1 Coleta e Análise de Dados

Para calcular a ampliação da lente objetiva em conjunto com a ocular, deve-se multiplicar suas ampliações (FLOURNOY, 2018). Dessa forma, como neste trabalho tem-se as ampliações de 40x e 20x das objetivas e a ampliação da ocular é de 10x, utilizou-se ampliações de 200x e 400x. Com a fórmula da medida campo de visão da lâmina na equação 7.1, tem-se, para ampliação em 200x:

$$\frac{20}{200} = 0,1mm$$

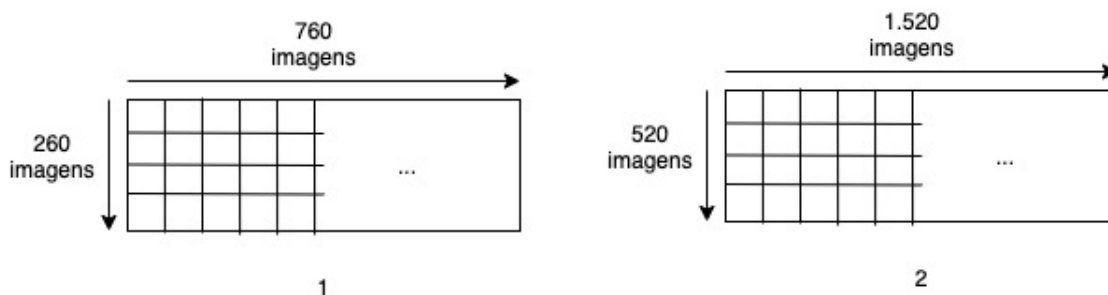
,

e para 400x:

$$\frac{20}{400} = 0,05mm$$

Isso significa que o microscópio ampliado em 200x capta um campo de visão de 0,1mm, e para 400x de 0,05mm. Desse modo, calculou-se quantas imagens seriam necessárias para digitalizar uma lâmina completa, medindo 76mmx26mm, resultando em 197.600 imagens em ampliação de 200x e 790.400 para ampliação de 400x, desconsiderando as sobreposições, que aumentaria a quantidade de imagens. A Figura 30 apresenta o sentido de captura de imagens e quantas seriam necessárias no eixo horizontal e vertical.

Figura 30 – Quantidade de imagens para digitalizar uma lâmina completa, desconsiderando sobreposições. (1) representação de varredura de captura de imagens com ampliação em 200x. (2) representação de lâmina com varredura para captura de imagens em ampliação de 400x.



Fonte: Autor.

7.1.2.2 Plano de Ação

Devido à extensa quantidade de imagens que seriam necessárias, tornando o *stitching* de uma lâmina completa praticamente impossível de ser executada manualmente, optou-se por reduzir a área, de forma que fossem necessárias amostras de nove imagens que a representassem, possibilitando desenvolver o sistema reduzido a uma área da lâmina.

7.1.2.3 Resultados da Ação

A redução de imagens para constituir uma amostra que representa uma área da lâmina possibilitou que o sistema executasse de forma prática, com redução de tempo e trabalho em capturar as imagens que compõem determinada área.

7.1.3 Cenário de Uso 3

O terceiro cenário de uso concentrou-se em testar o algoritmo de *stitching* a partir de uma amostra de nove imagens de uma lâmina coletadas em ampliação em 400x no microscópio, utilizando o adaptador a fim de verificar o comportamento do algoritmo em imagens coletadas diretamente de um microscópio e que não são de um banco de imagens como na prova de conceito. Essa primeira amostra foi coletada com um dispositivo iPhone 6, enquanto as outras amostras foram coletadas por um iPhone SE 2020.

O fluxo de aquisição das imagens iniciou-se a partir uma posição inicial, da direita para esquerda, adquirindo três imagens, partindo, então, para uma linha inferior, com a posição inicial à esquerda e adjacente ao ponto de partida do fluxo, seguindo essa repetição até que fossem capturadas nove imagens, tentando manter de 30% a 50% de sobreposição entre elas.

A Figura 31 apresenta esta primeira amostra e o fluxo de captura das imagens.

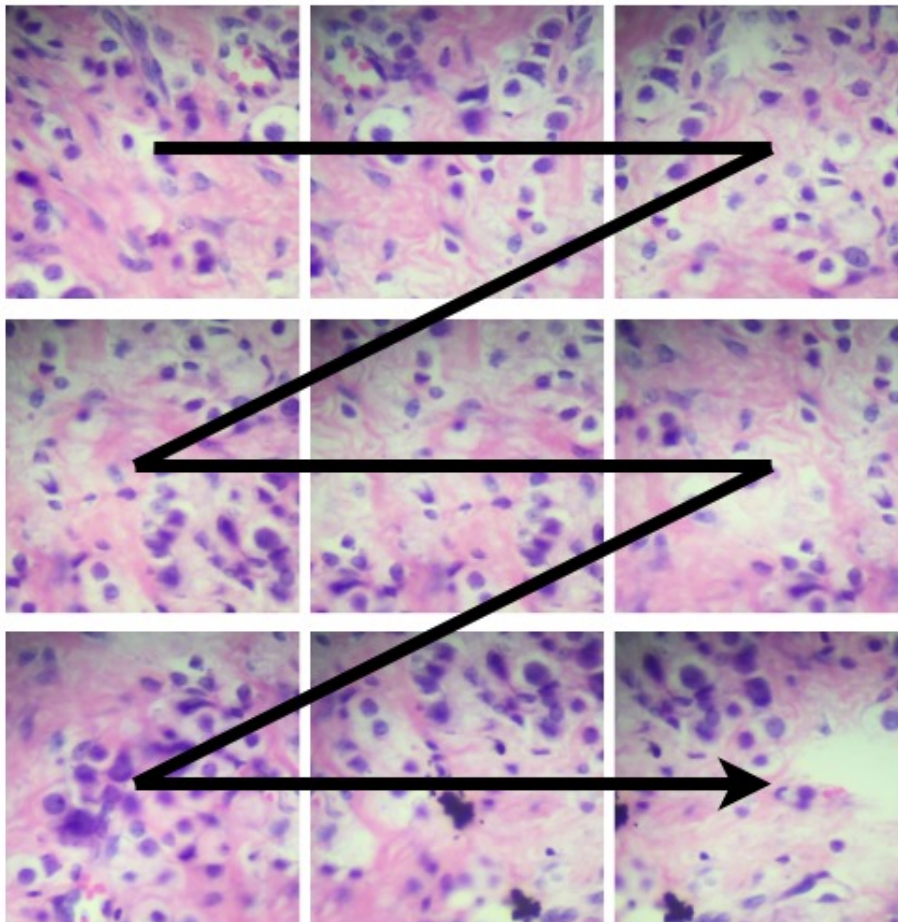
7.1.3.1 Coleta e Análise de Dados

O algoritmo não funcionou como o esperado, a imagem não conseguia ser gerada e havia erro no cálculo de homografia, que ao ser executado a parte do algoritmo de mesclar as imagens, retornava valores negativos, apontando que o ORB não encontrava de forma apropriada os *keypoints* e as características das imagens para que fossem comparadas com os pontos em comuns dos outros fragmentos, acontecendo o mesmo problema com o SURF. Com esta não conformidade encontrada e seguindo o ciclo apontado no Capítulo 4, desenvolveu-se o plano de ação descrito na próxima Subseção 7.1.3.2.

7.1.3.2 Plano de Ação

Para verificar se eram de fato problemas na imagem e não no algoritmo, executou-se o *stitching* com o SURF, ocorrendo a mesma não-conformidade. Desse modo, planejou-se utilizar outra lâmina, com apoio do especialista para capturar as imagens, sem o adaptador e adquirindo imagens em ampliação em 200x e 400x, sendo a de 400x utilizada no planejamento da ação para verificar o comportamento do algoritmo de *stitching* na mesma ampliação que a primeira amostra, tanto utilizando o ORB, quanto o SURF, e a similaridade das imagens. Optou-se também por alterar o fluxo de captura das imagens,

Figura 31 – Fluxo de captura de imagens da primeira amostra para o primeiro cenário de uso.



Fonte: Autor.

como sugestão do especialista, para que a referência de posição tivesse o mínimo de perda. A Figura 32 apresenta a diferença de fluxo na aquisição de imagem planejada.

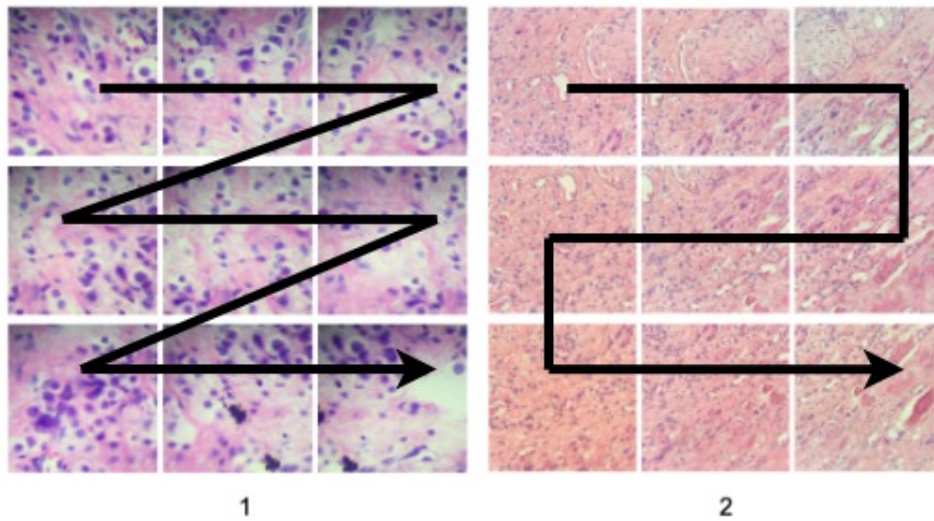
A partir desse plano de ação, obtiveram-se os resultados descritos na subseção [7.1.3.3](#)

7.1.3.3 Resultados da Ação

Os algoritmos ORB e SURF conseguiram executar com as novas amostras de imagem. A Figura 33 apresenta que as imagens geradas a partir dos fragmentos apresentaram diferenças nas suas composições.

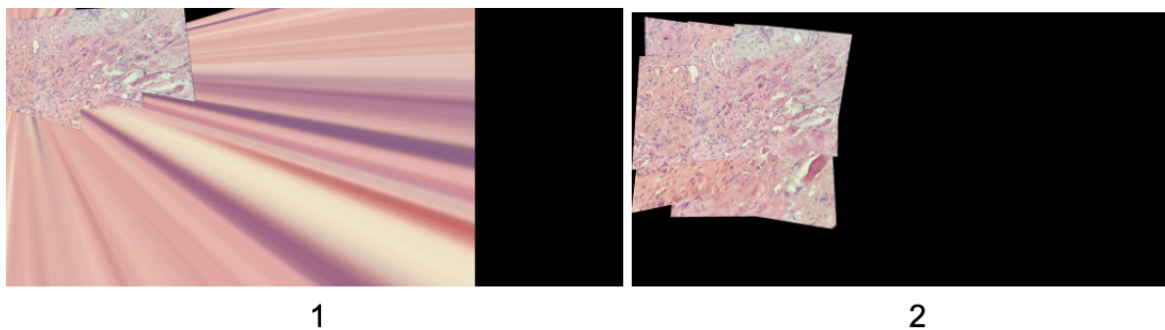
A diferença entre as imagens geradas pode ser explicado pois, segundo [Cortés-Gallardo et al. \(2019\)](#), enquanto o ORB apresenta um melhor desempenho de execução, ele não é eficaz em imagens que possam conter ruídos. Por conta do SURF ter tido melhor eficácia na composição dessas imagens, optou-se por utilizá-lo no algoritmo do sistema

Figura 32 – Fluxo de captura dos fragmentos de imagem de uma área de duas lâminas diferentes. (1) Primeira amostra de imagens da área a ser realizada o *stitching*. (2) Segunda amostra de imagens da área a ser realizada o *stitching*.



Fonte: Autor.

Figura 33 – Comparação do *stitching* gerado pelo ORB e SURF de fragmentos ampliados em 400x. (1) *Stitching* gerado pelo algoritmo ORB. (2) *Stitching* gerado pelo algoritmo SURF.



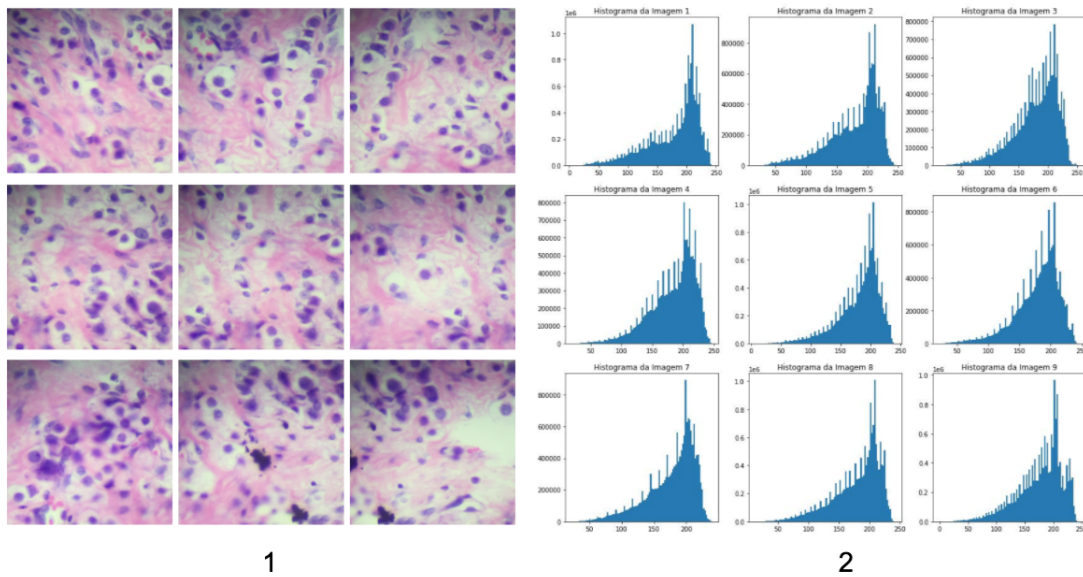
Fonte: Autor.

Mosaic.

A primeira amostra foi uma lâmina diferente da segunda, e mostra uma alta similaridade entre seus fragmentos, o que comprometeu ao algoritmo SURF e ORB de detectar keypoints, fazendo com o que o algoritmo não se completasse.

A Figura 34 apresenta os fragmentos de imagem da área e o histograma de similaridade, que representa a frequência de intensidade dos pixels. Por sua vez, a Figura 35 apresenta de forma ampliada as informações dos histogramas para melhor visualização.

Figura 34 – Fragmentos de imagem e seus histogramas. (1) Fragmentos das imagens da área a ser realizada o *stitching*. (2) Histograma de cada imagem em sua respectiva ordem.



Fonte: Autor.

A segunda amostra, como citado anteriormente, foi capturada por um iPhone modelo SE 2020 Entretanto sem a utilização de um suporte de acoplamento entre o *smartphone* e microscópio. É possível notar que a imagem não possui *vignetting*.

A captura manual, a partir da prática do especialista, ocorreu com melhor desempenho do que utilizando o suporte de acoplamento, pois este necessita adequar a disposição do *smartphone* para captar a imagem a partir da ocular.

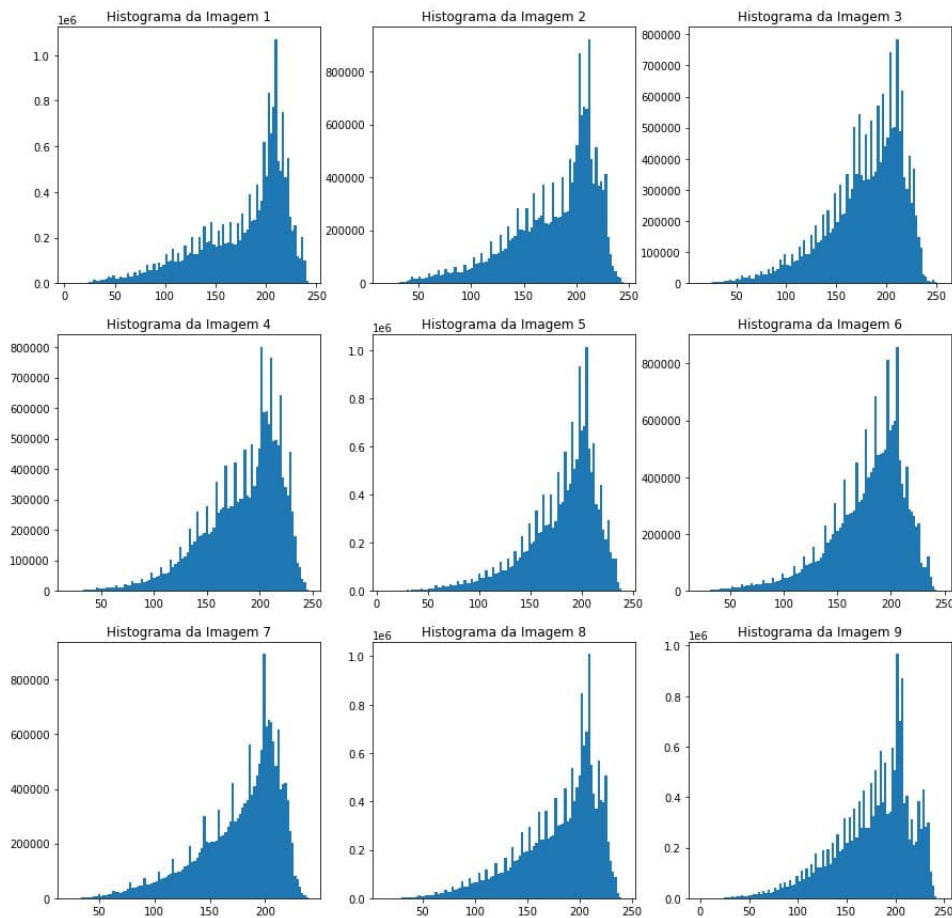
A Figura 37 apresenta os histogramas das imagens de forma apliada para melhor visualização dos dados.

7.1.4 Cenário de Uso 4

O suporte adaptador para acoplar o *smartphone* no microscópio foi comprado na internet. Este adaptador adequa-se a *smartphones* que possuem câmeras do lado direito do dispositivo, como o iPhone SE 2020. A Figura 38 apresenta o adaptador sem estar acoplado ao microscópio.

O adaptador acoplado ao microscópio e utilizando o *smartphone* para captar a imagem ampliada é apresentado na Figura 39

Figura 35 – Histogramas dos fragmentos de imagem da primeira amostra ampliada para melhor visualização.



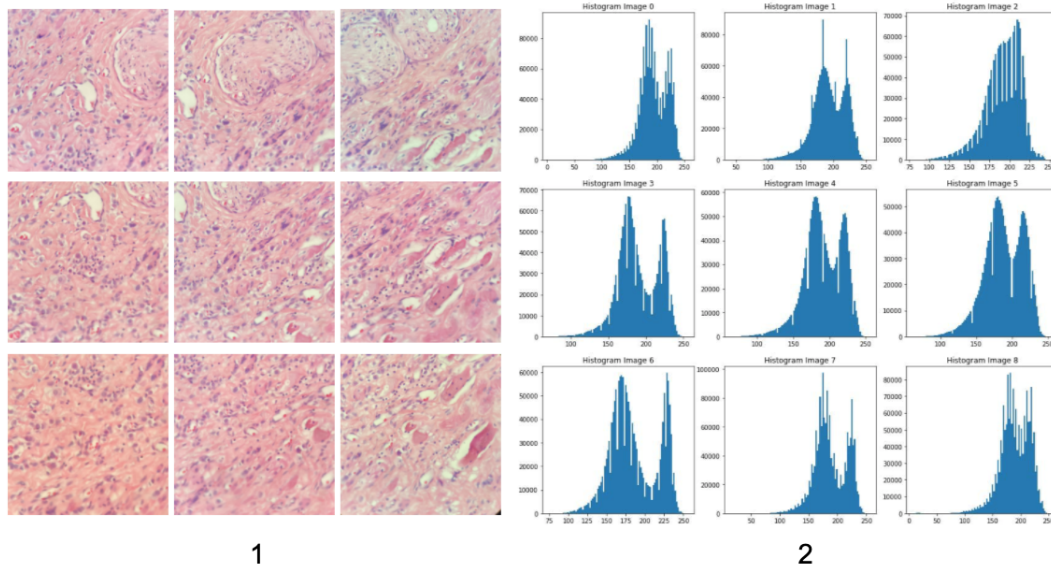
Fonte: Autor.

7.1.4.1 Coleta e Análise de Dados

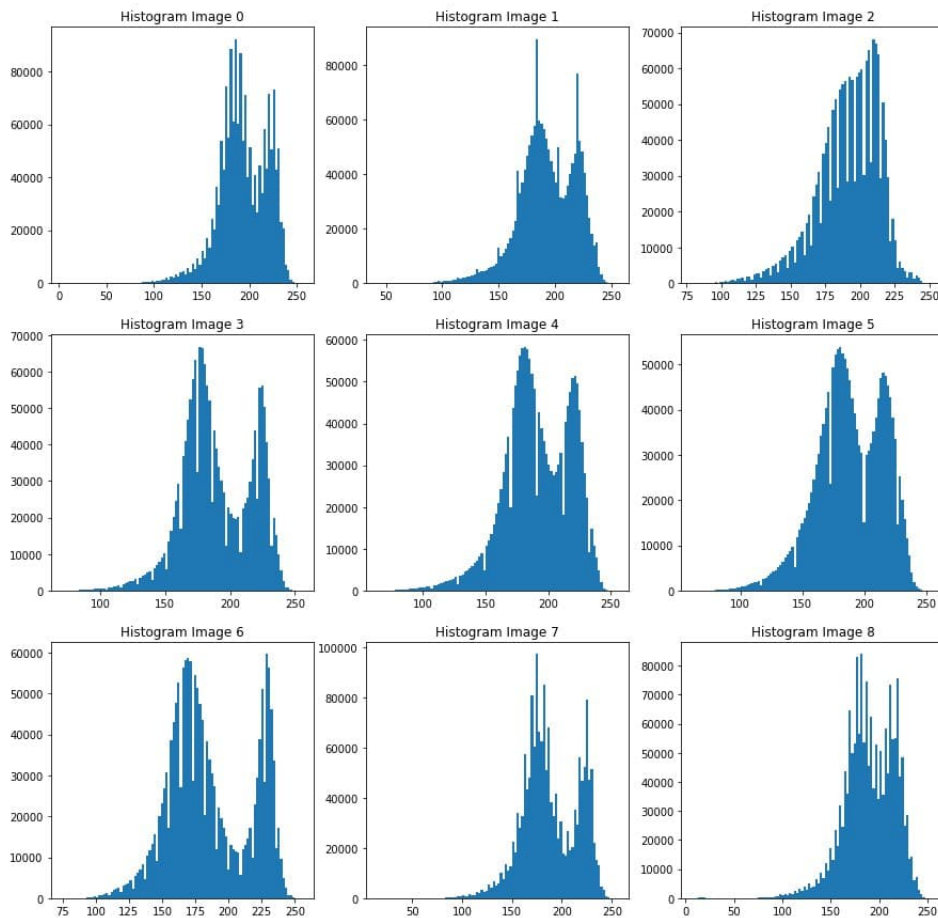
Durante o desenvolvimento desse trabalho, verificou-se que o adaptador possui problemas para focalizar a imagem na ocular do microscópio, pois o *smartphone* sofre sucessivos deslocamentos nas duas direções de ajuste do adaptador. Verificou-se ainda que é significativo o tempo gasto para ajustar o *smartphone* para que consiga focalizar a imagem a partir da ocular. Levou-se em média 15 minutos para esse ajuste.

7.1.4.2 Plano de Ação

Para coletar outras amostras para coleta de resultados, como citado na Seção 7.1.1 no primeiro cenário de uso para captura de imagens, utilizou-se da prática do especialista para capturar as imagens de forma manual, método que já era adotado.

Figura 36 – Saída de verificação de arquivo DICOM pelo *DICOM Validator*

Fonte: Autor.

Figura 37 – Saída de verificação de arquivo DICOM pelo *DICOM Validator*

Fonte: Autor.

Figura 38 – Adaptador de acoplamento entre *smartphone* e ocular do microscópio.



Fonte: Autor.

7.1.4.3 Divulgação dos Resultados

Com a forma manual de captura de imagens pelo especialista, verificou-se que havia mais agilidade, levando em média dois minutos para capturar as nove imagens e fazendo com que o adaptador fosse dispensado. Dessa forma, conclui-se que o adaptador atrapalha o desempenho para aquisição de imagens sucessivas, porém pode ser útil para quem não tem a prática de capturar fotografias a partir da ocular do microscópio e para imagens que não dependam de movimentação para imagens adjacentes.

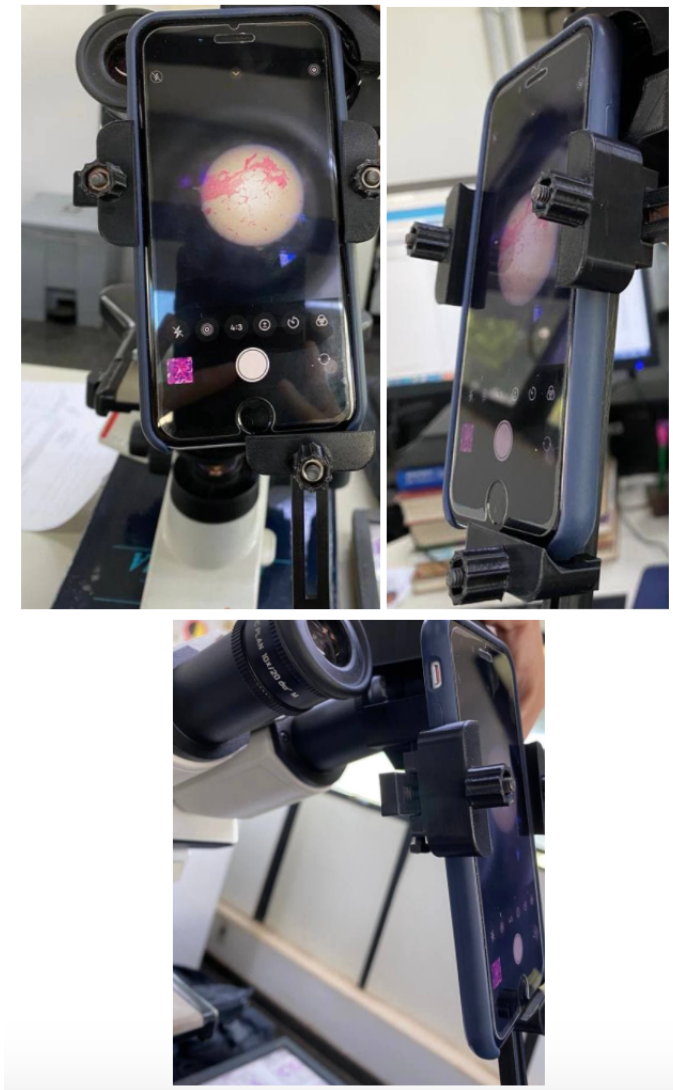
7.1.5 Cenário de Uso 5

Para verificar se o arquivo DICOM gerado pela solução é válido, utilizou-se o *DICOM Validator* (fonte). Essa ferramenta analisa o arquivo, tendo com saída possíveis *warnings* e erros no arquivo que esteja em não conformidade com o protocolo DICOM. Caso o arquivo esteja válido, ele retorna a identificação do IOD, que no caso deste trabalho é o *SC Image*.

7.1.5.1 Coleta e Análise de Dados

O arquivo DICOM gerado pelo Mosaic conseguiu ser validado pelo *DICOM Validator*, não gerando não-conformidade neste ciclo de cenário de uso. A Figura 40 apresenta a saída gerada ao analisar um dos arquivos DICOM gerados pelo Mosaic.

Figura 39 – Adaptador acoplado na ocular do microscópio em captura de imagem de uma área da lâmina.



Fonte: Autor.

Figura 40 – Saída de verificação de arquivo DICOM pelo *DICOM Validator*

```
wesleyaraujo@Wesley-Araujo dicom3tools_macexe_1.00.snapshot.20201208085311 % ./dciodvfy ../MosaicTCC2Backend/resources/7232a2c  
a-c596-4d24-a333-1b0b82a21360/original_7232a2ca-c596-4d24-a333-1b0b82a21360.dcm  
SCImage  
wesleyaraujo@Wesley-Araujo dicom3tools_macexe_1.00.snapshot.20201208085311 %
```

Fonte: Autor.

7.1.6 Cenário de Uso 6

Além da verificação do arquivo DICOM a partir da validação por meio do *DICOM Validator*, neste cenário de uso, analisou-se o comportamento do arquivo em um serviço PACS adequado ao padrão DICOM. Para este fim, utilizou-se o *PostDICOM*.

7.1.6.1 Plano de Ação

Neste cenário de uso, foram utilizadas as funcionalidades de fazer *upload* do arquivo DICOM original, gerado pela API e armazenado localmente, e a visualização desse arquivo para detalhar a imagem e os metadados.

7.1.6.2 Coleta de Análise de Dados

A Figura 41 apresenta a tela em que o usuário faz *upload* do arquivo DICOM para o serviço em nuvem do *PostDICOM*.

O *PostDICOM* lista os arquivos que foram carregados pelo usuário e estão disponíveis no serviço para que o usuário selecione o arquivo DICOM de interesse. A tela que consiste essa funcionalidade é apresentada na Figura 42.

A funcionalidade de exibição do arquivo DICOM é explorada a fim de verificar os metadados e a imagem. A Figura 43 apresenta essa tela de exibição.

Figura 41 – Tela de *upload* de arquivo DICOM do serviço *PostDICOM*.



Fonte: Autor.

Figura 42 – Tela de exibição dos arquivos DICOM carregados no *PostDICOM*.

The screenshot displays the 'Pesquisa de Pacientes' (Patient Search) interface in the PostDICOM application. On the left, there is a search filter panel with the following elements:

- A search bar labeled 'Palavra-chave' with a magnifying glass icon.
- A 'Faixa de datas' (Date Range) section with a dropdown menu set to 'Todos' (All), and two date input fields labeled 'Data De' and 'Até Data', both set to 'dd/mm/aaa'.
- A 'Modalidade' (Modality) section with radio buttons for MR, CT, CR, US, MG, NM, XA, and PT.
- An 'Avançado Pesquisar' (Advanced Search) section with a plus icon and a text input field.
- Two buttons: 'Limpar' (Clear) and 'Pesquisar' (Search).

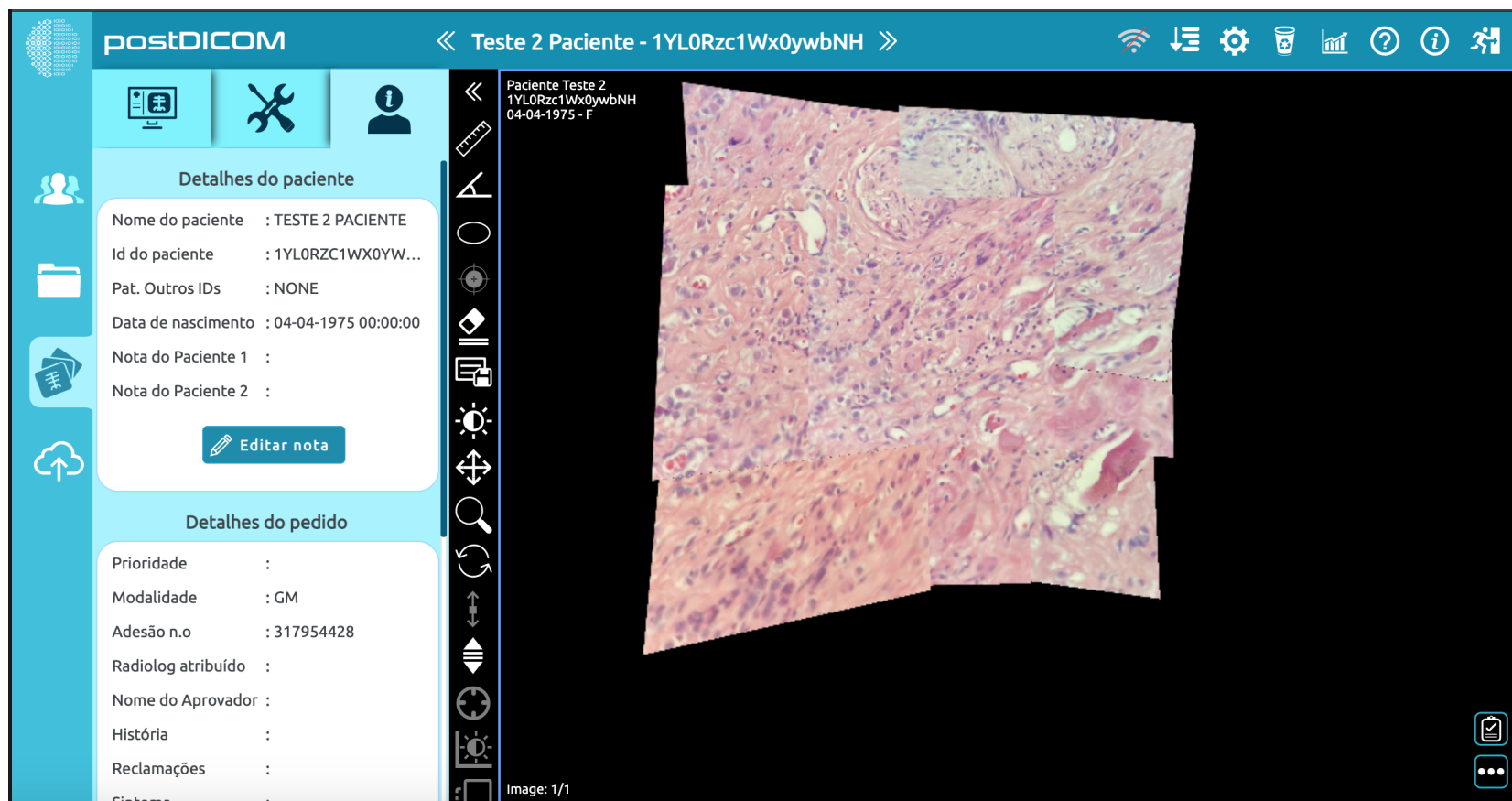
The main area on the right shows a table of search results with the following columns: Status, Nome do paciente (Patient Name), Id do paciente (Patient ID), Pat. Outros IDs (Other IDs), Modali... (Modality), Adesão n.o (Access No.), Descrição do ... (Description), Data (Date), and Imagens (Images). One result is visible:

Status	Nome do paciente	Id do paciente	Pat. Outros IDs	Modali...	Adesão n.o	Descrição do ...	Data	Imagens
	Teste 2 Paciente	1YL0Rzc1W...	NONE	GM	317954428		04-04-2021 18:48:10	1

At the bottom of the interface, there is a pagination control showing 'Registro 1 (1 - 1)' and navigation arrows: '« Anterior < 1 / 1 > Próxima »'. A 'Filtrar' (Filter) button with a green 'X' icon is also present.

Fonte: Autor.

Figura 43 – Tela de exibição dos metadados e imagem do arquivo DICOM selecionado.



Fonte: Autor.

7.1.7 Cenário de Uso 7

Nesse cenário de uso, verificou-se o comportamento do *Cloudinary* como ferramenta a ser utilizada como CDN para armazenar as imagens e arquivos DICOM gerados pelo *Mosaic*. As imagens e arquivos foram gerados localmente e nomeados de forma aleatória a partir de um *hash* e armazenados em seus diretórios também nomeados com o mesmo *hash*. Então, o *Cloudinary* foi integrado à API, de forma que esses arquivos pudessem ser armazenados seguindo o mesmo protocolo de nomeação de diretórios.

7.1.7.1 Coleta e Análise de Dados

Ao efetuar o *upload* da imagem, verificou-se que ela foi armazenada com sucesso na nuvem. Entretanto, ao ser enviado o arquivo DICOM o *Cloudinary* bloqueou, já que ultrapassava o limite de 20MB de tamanho (fonte), o arquivo possuía 26MB, impossibilitando o armazenamento do arquivo DICOM.

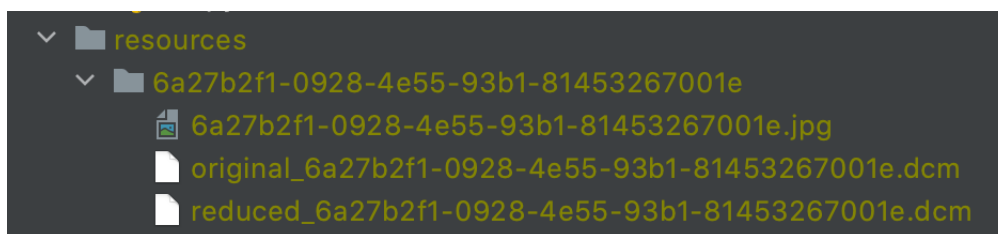
7.1.7.2 Plano de Ação

Para que fosse possível o arquivo DICOM ser armazenado no *Cloudinary* nas condições de limite de *upload*, reduziu-se a imagem em 50% a fim de que os dados dessa imagem no arquivo DICOM ocupassem menos espaço, conseqüentemente reduzindo a dimensão do arquivo DICOM.

7.1.7.3 Resultados da Ação

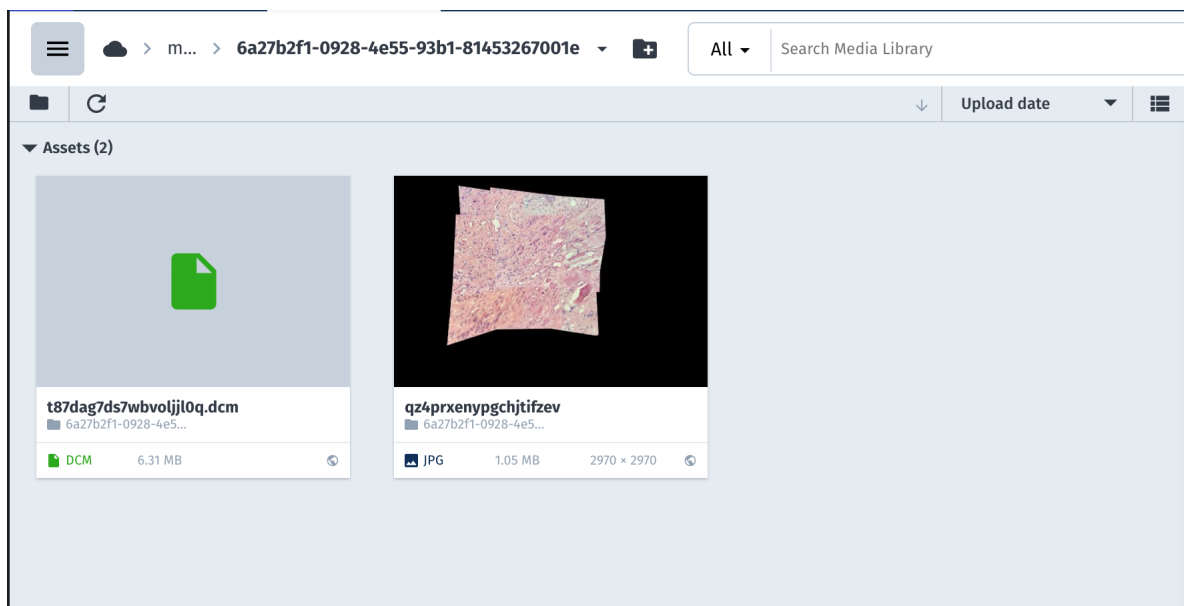
Com a redução do arquivo DICOM, foi possível utilizar o *Cloudinary* utilizá-lo como ferramenta de armazenamento em nuvem.

Figura 44 – Arquivos locais gerados pela API. Além da imagem gerada pelo *stitching* das amostras e do arquivo DICOM, há a versão reduzida do arquivo DICOM para poder ser enviado ao *Cloudinary*.



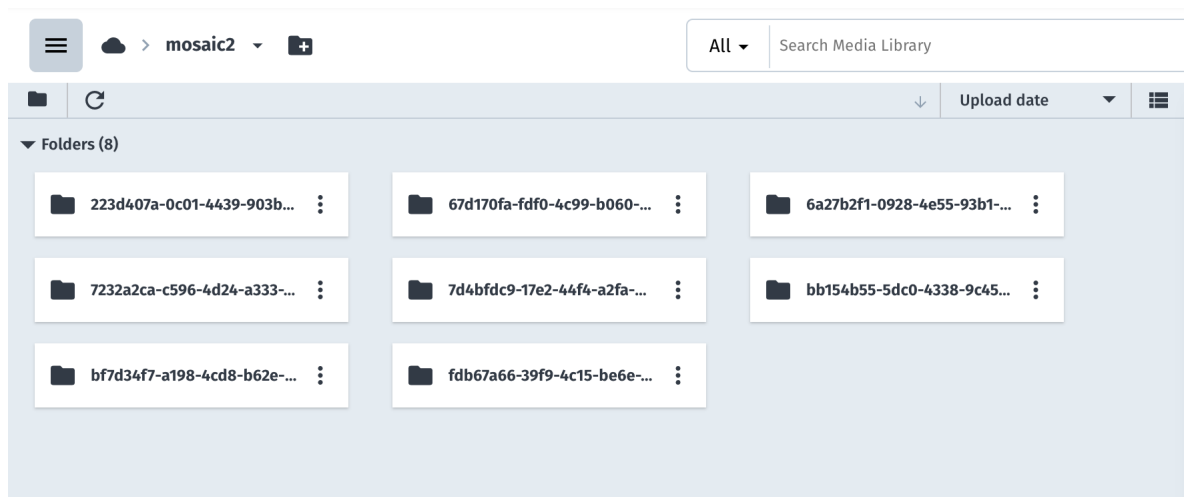
Fonte: Autor.

Figura 45 – Arquivos enviados ao *Cloudinary* acessados pela plataforma. Os nomes dos arquivos na nuvem sofrem alteração pois usou-se o comportamento padrão do *Cloudinary* em criar próprias *hashs*.



Fonte: Autor.

Figura 46 – Organização dos diretórios dos arquivos enviados. Cada diretório agrupa a imagem gerada e o arquivo dicom, identificado com a hash criada pela API.



Fonte: Autor.

7.2 Engenharia de produto

Como discutido no Capítulo 4 de Metodologia, o método de desenvolvimento da solução foi um processo adaptado do *Scrum Solo*. Esse método adequa o *framework Scrum* para quando há apenas o desenvolvedor atuando no processo de desenvolvimento (PAGOTTO et al., 2016). A partir desse processo de desenvolvimento, chegou-se a alguns artefatos inerentes ao processo *Scrum* como as Histórias de Usuário, que são requisitos

escritos na perspectiva do usuário da solução (PAGOTTO et al., 2016). A Tabela 5 apresenta as Histórias de Usuário com sua identificação, tarefas e pontuações.

Tabela 5 – Histórias de Usuário com suas respectivas tarefas e pontuação

ID	Descrição	Tarefas	Pontos
1	Como usuário, devo carregar imagens do campo de visão para ser composta a imagem da área total.	Desenvolver tela do aplicativo Desenvolver serviço <i>Backend</i> Integrar tela com serviço <i>Backend</i>	13
2	Como usuário, devo adicionar informações relevantes para que esteja de acordo com o padrão DICOM <i>SC Image</i> .	Desenvolver tela do aplicativo Desenvolver serviço <i>Backend</i> Integrar tela com serviço <i>Backend</i>	8
3	Como usuário, posso listar os arquivos DICOM gerados para que eu possa gerenciá-los	Desenvolver tela do aplicativo Desenvolver serviço <i>Backend</i> Integrar tela com serviço backend	8
4	Como usuário, devo me cadastrar para que eu possa criar acesso aos serviços do app	Desenvolver tela do aplicativo Desenvolver serviço <i>Backend</i> Integrar tela com serviço <i>Backend</i>	13
5	Como usuário, devo efetuar o login para que eu possa ter acesso aos serviços do app.	Desenvolver tela do aplicativo Desenvolver serviço <i>Backend</i> Integrar tela com serviço <i>Backend</i>	13

Fonte: Autor.

As Histórias de Usuário possuem os critérios de aceitação para que elas possam ser consideradas como satisfatórias no nível de aceitação com requisito para o usuário (PAGOTTO et al., 2016). Os critérios de aceitação são apresentados na Tabela 6 de acordo com a identificação da História de Usuário o qual pertence.

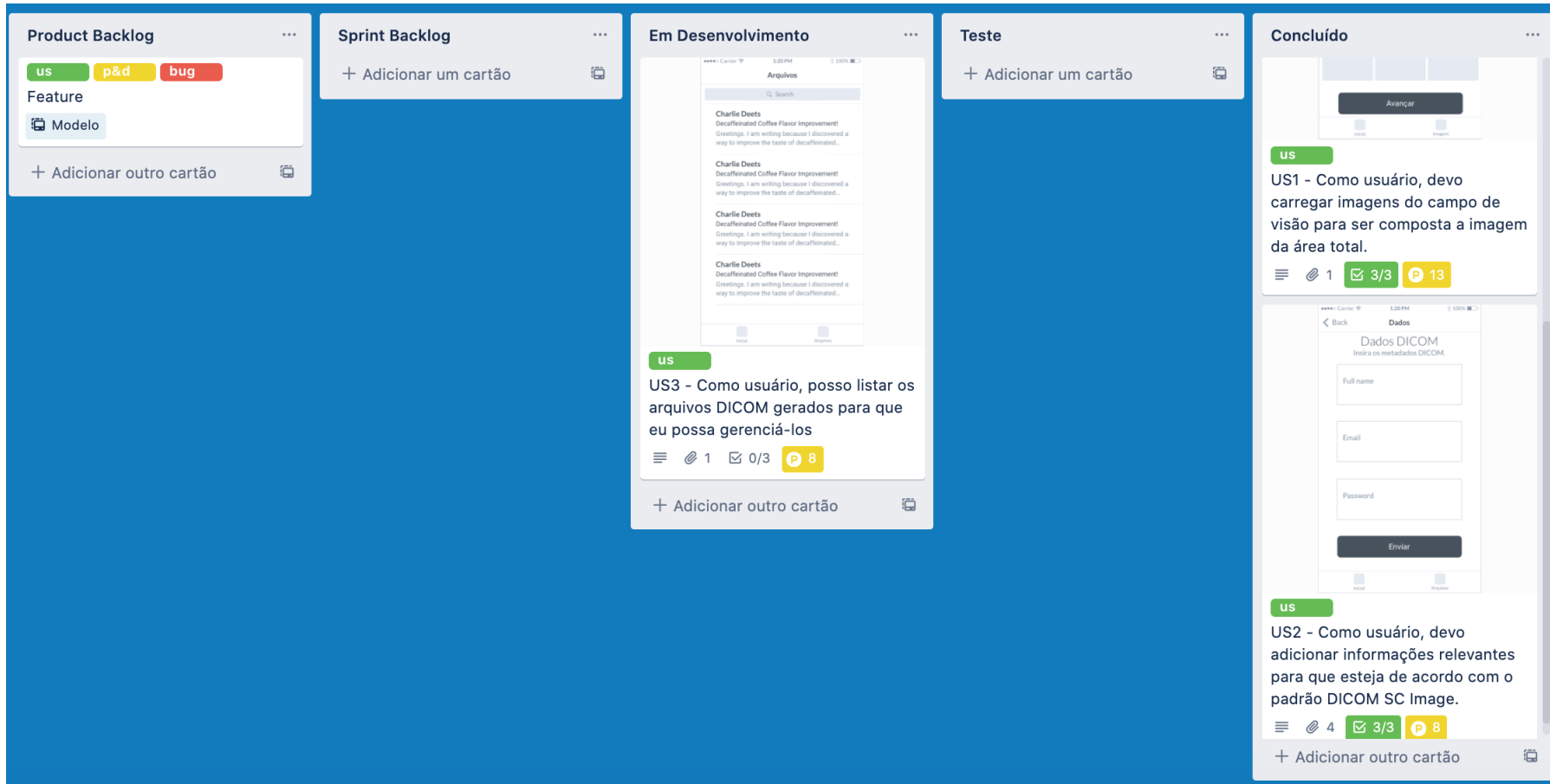
Para acompanhar o progresso das funcionalidades, usou-se como ferramenta de *kanban* o serviço *Trello*. Cada História de Usuário possui também uma imagem do protótipo, caso necessário, para guiar a interface do sistema. A Figura 47 apresenta o *kanban* do desenvolvimento do Mosaic, e a Figura 48 apresenta uma História de Usuário em detalhe no *kanban Trello*.

Tabela 6 – Critérios de aceitação das Histórias de Usuário.

ID	Critérios de Aceitação
1	- Deve-se limitar a apenas 9 imagens referentes a um campo visual, e - Deve haver opção de escolher foto da galeria ou tirar uma foto
2	- Todos os dados devem ser obrigatórios de preencher; - Os dados devem ser referentes apenas aos obrigatórios do SC Image, e - Após o preenchimento, deverá levar ao processamento dos dados. - Alertar em caso de sucesso ou erro.
3	- Os arquivos devem ser ordenados a partir do mais recente, e - O click em um arquivo deve ter a opção de compartilhar
4	- O campo de senha não poderá exibí-la; - A tecla de retorno deverá fazer com que o teclado desapareça, e - Exibir mensagem de sucesso ao cadastrar ou de erro.
5	- O campo de senha não poderá exibí-la; - A tecla de retorno deverá fazer com que o teclado desapareça, e - Exibir mensagem de sucesso ao cadastrar ou de erro.

Fonte: Autor

Figura 47 – Trello sendo utilizado como *kanban* para gerenciar o progresso do desenvolvimento dos requisitos.



Fonte: Autor.

Figura 48 – Trello sendo utilizado como *kanban* para gerenciar *status* dos requisitos.



Fonte: Autor.

Segundo Pagotto et al. (2016), protótipos do sistema são artefatos criados no ciclo da *sprint*. Dessa forma, as Histórias de Usuário possuem protótipos para guiar o desenvolvimento de interface. Para o desenvolvimento do protótipo, foi utilizada a ferramenta *Marvel*. A Figura 51 apresenta o protótipo da solução de *software* Mosaic.

7.3 Validação pelo Especialista

O papel como validador do sistema é apontado no Capítulo 4 como sendo o especialista da área, um médico patologista. Para a validação, gravou-se um vídeo do aplicativo sendo executado, com velocidade aumentada em 2x e também um vídeo do uso do visualizador *PostDICOM*. A partir disso, um questionário foi disponibilizado para o especialista.

Este questionário com as respostas constam no Apêndice D.

Esse questionário teve como objetivo ter uma visão do uso do sistema Mosaic e perspectivas futuras de melhorias.

Em termos de praticidade quanto ao uso do adaptador, podendo ser classificado de zero a cinco, de forma que zero seria pouco prático e cinco muito prático, o especialista marcou como 2. Ainda sobre o adaptador, a pesquisa apontou que o especialista tem preferência por capturar imagens de forma manual em vez de utilizar o adaptador para isso. Desse modo, conclui-se que o adaptador é pouco prático na visão do médico.

Quanto ao Mosaic, em termos de proposta, baseando-se na escala de zero a cinco, obteve-se a classificação 3, alcançando a mesma classificação em termos de praticidade. Como propostas de melhoria, foi apontado que o sistema poderia ser expandido para câmeras de microscopia já existentes, não limitando-se apenas para *smartphones*. Para fins de uso, foi apontado pelo especialista que seria utilizado por ele para digitalização de lâminas em pequeno aumento (menor que 200x) para dar noção panorâmica em discussões de casos clínicos.

Quanto ao padrão DICOM, o especialista indicou que já havia conhecimento do mesmo e que o seu uso no ambiente laboratorial da anatomia patológica seria factível, já que a radiologia utiliza rotineiramente.

7.4 Resumo do Capítulo

Este capítulo apresentou a coleta de resultados do trabalho, em que a Seção 7.1 descreveu os cenários de uso por meio dos ciclos definidos no Capítulo 4 de Metodologia. Na Subseção 7.1.1, analisou-se os resultados obtidos a partir da composição das imagens sem remoção da área escura da captura a partir de um *smartphone*, que ocasionou em uma não conformidade na composição, apresentando uma sobreposição da área de *vignetting* também, que é a parte escura captada da ocular. O plano de ação para essa não conformidade foi concentrar um formato quadrado da imagem de interesse, que é da lâmina e que fosse um quadrado inscrito nesta imagem. Desse modo, apresentou uma composição coerente com as áreas de interesse, porém com uma perda de 36,33% da área circular da lâmina, considerando ser um quadrado inscrito perfeito. No segundo cenário de uso, descrito na Subseção 7.1.2, analisou-se a viabilidade de digitalização de uma lâmina inteira pelo método de fotografia dos fragmentos das imagens ampliadas que a compõem, calculando a quantidade de imagens que seriam necessárias em ampliação de 200x e 400x, resultando em 197.600 para 200x e 790.000 em 400x, tornando-se impraticável digitalizar uma lâmina por completo a partir das fotografias. Assim, planejou-se a ação de reduzir a quantidade de imagens, tornando-se uma amostra de uma área da lâmina, limitando-se a nove imagens, resultando em uma composição manual de forma prática

para uma área da lâmina. Na Seção 7.1.3 do terceiro cenário de uso, tem-se a análise do algoritmo de *stitching*, apresentando duas amostras ampliadas em 400x, em que uma delas foi capturada utilizando o adaptador de acoplamento entre *smartphone* e ocular do microscópio, enquanto a outra amostra capturou-se de forma manual pelo especialista, sendo as duas amostras de lâminas e fluxos de captura diferentes. Como resultados, na primeira, amostra o algoritmo não conseguiu realizar o *stitching*, o qual não conseguiu identificar vários pontos de característica da imagem. Já na segunda amostra o algoritmo conseguia ser completado. Dessa forma, observou-se que a primeira amostra havia similaridade alta em intensidade de pixel entre as imagens, diferentemente da segunda amostra. Observou-se também que, utilizando o algoritmo ORB para detecção de características, a imagem gerada foi distorcida, gerando uma não conformidade, explicada pelo fato de ter erros na localização dos pontos de característica, demonstrando que o ORB pode ter sua eficácia comprometida em casos que a imagem possa apresentar ruído, como algumas áreas borradas nos cantos da imagem. Como plano de ação, o SURF foi aplicado como algoritmo para detecção de características, o qual resultou em uma imagem composta sem distorção, sendo ainda verificado que o fluxo de captura de imagens no sentido em que o ponto inicial se inicia da esquerda e o ponto inferior é adjacente na parte superior à última imagem capturada na fileira acima, pode apresentar menos perdas em sobreposição com a imagem inicial. No quinto cenário de uso, descrito na Subseção 7.1.5, analisou-se o resultado de utilizar o adaptador de acoplamento entre *smartphone* e microscópio, que se mostrou menos prático que a forma como o especialista capturava manualmente, pois houve deslocamento do *smartphone* no adaptador, fazendo com que o foco na ocular para que possibilitasse focar na imagem da lâmina fosse perdida. O encaixe do próprio *smartphone* também apresentou um tempo, já que é necessário encontrar uma localização em que a câmera do *smartphone* seja capaz de focalizar a imagem da lâmina pela ocular do microscópio. Desse modo, como plano de ação, utilizou-se da experiência manual do especialista para capturar as imagens, que teve como resultado uma redução de tempo significativa, de mais de 15 minutos. No sexto cenário de uso, descrito na Seção 7.1.6, validou-se o arquivo DICOM gerado com a ferramenta *DICOM Validator*, que resultou em um arquivo DICOM válido, conforme proposto. Por fim, na Seção 7.1.7, é apresentado o último cenário de uso, com o qual procurou-se analisar o resultado.

Na Seção 7.2, os resultados do processo de desenvolvimento, o qual seguiu o *Scrum Solo*, são apresentados, resultando em cinco Histórias de Usuário, com um total de 55 pontos de esforço, bem como os critérios de aceitação para terem a definição de pronto. Utilizou-se a ferramenta *Trello* como suporte de análise de progresso das funcionalidades por meio do *kanban*.

Na última Seção 7.3, apresentou-se o resultado de um questionário aplicado ao especialista, a fim de que validasse a solução desenvolvida. Como resultado em termos de praticidade quanto ao uso do adaptador, que havia classificação de zero a cinco, sendo

zero pouco prático e cinco muito prático, obteve-se a classificação dois e, como resposta, a preferência de utilizar o método manual de capturar as imagens, já utilizada pelo especialista. Em termos de proposta da solução Mosaic, obteve-se a classificação três, também alcançando essa mesma classificação em termos de praticidade. Como propostas de melhoria, foi apontado pelo especialista expandir o sistema para captura de imagens a partir de câmeras de microscopia já existentes e, para fins de uso, o especialista respondeu que sua utilização seria útil para digitalizar lâminas em pequeno aumento, menor que 200x, para dar noção panorâmica em discussões de casos clínicos. Quanto ao padrão DICOM, o especialista já havia conhecimento e apontou que sua utilização no ambiente laboratorial na anatomia patológica seria factível, já que é amplamente utilizado na radiologia.

8 Conclusão

Este Trabalho de Conclusão de Curso buscou desenvolver uma solução em *software* com o objetivo de digitalizar uma lâmina histopatológica utilizando *smartphone iPhone*, a fim de analisar a viabilidade do desenvolvimento de uma solução WSI com ferramentas acessíveis, e que o processo de desenvolvimento esteja de acordo com as práticas de Engenharia de Software.

O suporte de um especialista foi essencial para o desenvolvimento do trabalho, visto que foram disponibilizadas lâminas histopatológicas que não possuíam dados do paciente, e que os metadados DICOM fossem com valores fictícios.

Verificou-se que para se implementar um WSI na abordagem de fragmentos de imagem, haveria um custo de tempo e grande quantidade de imagens. Sendo assim, optou-se por guiar a solução com uma amostra de nove imagens que representassem uma área da lâmina, sendo estas imagens sempre enviadas em formato quadrado para ser ignorado o *vignetting* causado pela ocular do microscópio. Notou-se também que a utilização do adaptador que acopla o *smartphone* ao microscópio possui instabilidade para fixar a posição do dispositivo celular e também uma demora considerável de em média 15 minutos para posicionar a câmera do *smartphone* na ocular do microscópio para que seja possível captar a imagem. Desse modo, notou-se que o especialista já possuía prática em fotografar a partir da ocular por um *smartphone*, tendo um desempenho superior.

Notou-se que o fluxo de leitura das imagens pode ter uma diferença de confiabilidade para tentar se aproximar de uma área quadrada da lâmina, pois uma leitura que se inicia sempre da posição esquerda poderia haver perda de confiabilidade de posição com imagens adjacentes superior e inferior de modo que uma leitura alternando os sentidos da posição, isto é, iniciando da esquerda, numa posição superior e que, na posição final da primeira fileira de três imagens a posição se inicia no ponto inferior adjacente à imagem final da fileira superior (clarear melhor isso em palavras.)

Para objetivos da aplicação Mosaic, a funcionalidade dos padrões do DICOMWeb para consultas QIDO-RS e WADO-RS foram descartadas, mas o padrão de armazenamento STOW-RS foi aproveitada, fazendo com que a API siga parcialmente uma API operacional de acordo com o DICOMWeb, de forma que as consultas dos dados sejam realizadas de uma forma mais direta pelo aplicativo a partir do mesmo *endpoint* de STOW-RS, porém utilizando o método *GET* de uma API *Restful*.

A similaridade da intensidade das amostras de imagens podem prejudicar os algoritmos de detecção de características como ORB e SURF encontrem *keypoints*, bem como verificou-se que o SURF atua com uma eficácia maior que o ORB em imagens que possam

ocorrer ruído como borrões nas imagens.

A solução cumpre o seu papel em capturar imagens e realizar o *stitching*, bem como em gerar um arquivo DICOM que é válido. Esta validade foi testada pela ferramenta *DICOM Validator*. A leitura da imagem também foi posta à prova na ferramenta *PostDICOM* em que exibiu a imagem da área da lâmina e seus metadados, o qual nesse trabalho foram utilizados apenas os dados obrigatórios para a validade do IOD *SC Image*.

Para o desenvolvimento da aplicação, utilizou-se o processo de desenvolvimento *Scrum Solo*, visto que foi desenvolvido por apenas um único desenvolvedor, assim sendo levantados requisitos representados em Histórias de Usuário, suas tarefas, identificação para rastreamento e *kanban* para acompanhar o progresso do desenvolvimento.

Cabe ressaltar ainda que todos os objetivos acordados no Capítulo de Introdução foram alcançados, sendo possível responder à questão de desenvolvimento de forma positiva e satisfatória.

Por fim, também como um resultado muito gratificante desse trabalho, com base nas pesquisas realizadas, foi possível a publicação de um capítulo de livro, disponível em [Araujo, Serrano e Serrano \(2021\)](#)

8.1 Trabalhos Futuros

Com objetivo de conferir continuidade a esse trabalho, os pontos seguintes poderão ser abordados em trabalhos futuros:

1. Estudar mais profundamente como adaptar o ORB para imagens histopatológicas e que possam ter ruído;
2. Desenvolver um algoritmo que possa utilizar a função de vídeo de um *smartphone* para ter uma captura de imagens mais prática como a solução *mWSI*;
3. Expandir a API de modo a se adequar ao padrão DICOMWeb;
4. Evoluir o aplicativo para que possa adicionar dados de acordo com o relacionamento entre entidades DICOM, e
5. Aplicar algoritmo de processamento para automatizar a remoção de *vignetting* causado pela ocular do microscópio.

Referências

- ADEL, E.; ELMOGY, M.; ELBAKRY, H. Image stitching system based on orb feature-based technique and compensation blending. *International Journal of Advanced Computer Science and Applications*, Citeseer, v. 6, n. 9, 2015. Citado 2 vezes nas páginas 43 e 46.
- AL-JANABI, S.; HUISMAN, A.; DIEST, P. J. V. Digital pathology: current status and future perspectives. *Histopathology*, Wiley Online Library, v. 61, n. 1, p. 1–9, 2012. Citado 4 vezes nas páginas 29, 30, 34 e 35.
- APPLE. *Xcode*. [S.l.]: Apple, 2019. <https://developer.apple.com/xcode/>. Acessado em 12/05/2019. Citado na página 49.
- ARAUJO, W. P.; SERRANO, M.; SERRANO, M. Novas tecnologias aplicadas à saúde: inovação, internet das coisas, horizontes e desafios. In: _____. [S.l.]: Editora UERN-EDUERN, 2021. cap. 14, p. 347–368. Citado na página 108.
- AUGUSTE, L.; PALSANA, D. Mobile whole slide imaging (mws): a low resource acquisition and transport technique for microscopic pathological specimens. *BMJ Innovations*, BMJ Specialist Journals, v. 1, n. 3, p. 137–143, 2015. Citado 3 vezes nas páginas 29, 30 e 31.
- BANCROFT, J. D.; GAMBLE, M. *Theory and practice of histological techniques*. [S.l.]: Elsevier health sciences, 2008. Citado na página 30.
- BROWN, M.; LOWE, D. G. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, Springer, v. 74, n. 1, p. 59–73, 2007. Citado na página 30.
- BROWN, S. *The C4 model for visualising software architecture: Context, Containers, Components and Code*. [S.l.]: C4 Model, 2019. <https://c4model.com/>. Acessado em 22/06/2019. Citado na página 66.
- CALONDER, M. et al. Brief: Binary robust independent elementary features. In: SPRINGER. *European conference on computer vision*. [S.l.], 2010. p. 778–792. Citado 2 vezes nas páginas 44 e 45.
- CHAN, J. K. The wonderful colors of the hematoxylin–eosin stain in diagnostic surgical pathology. *International journal of surgical pathology*, SAGE Publications Sage CA: Los Angeles, CA, v. 22, n. 1, p. 12–32, 2014. Citado na página 33.
- CHOPLIN, R. H.; 2ND, J. B.; MAYNARD, C. Picture archiving and communication systems: an overview. *Radiographics*, v. 12, n. 1, p. 127–129, 1992. Citado na página 50.
- CLUNIE, D. A. *DICOM Validator - dciodvfy*. [S.l.]: David A. Clunie, 2018. <http://dclunie.com/dicom3tools/dciodvfy.html>. Acessado em 26/04/2021. Citado na página 50.

- CORTÉS-GALLARDO, E. et al. A comparison of feature extractors for panorama stitching in an autonomous car architecture. In: IEEE. *2019 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*. [S.l.], 2019. p. 50–55. Citado 2 vezes nas páginas 71 e 86.
- CROSS, S. et al. Best practice recommendations for implementing digital pathology. *The Royal College of Pathologists, London*, 2018. Citado na página 34.
- CULJAK, I. et al. A brief introduction to opencv. In: IEEE. *2012 proceedings of the 35th international convention MIPRO*. [S.l.], 2012. p. 1725–1730. Citado 3 vezes nas páginas 42, 43 e 49.
- DANIEL, C. et al. Standardizing the use of whole slide images in digital pathology. *Computerized Medical Imaging and Graphics*, Elsevier, v. 35, n. 7-8, p. 496–505, 2011. Citado na página 34.
- DENG, H.; SONG, X. Chaos-based image encryption algorithm using decomposition. *Indones J Electr Eng Comput Sci*, Citeseer, v. 12, n. 1, p. 575–83, 2014. Citado na página 41.
- DICOM. *Secondary Capture Image IOD*. [S.l.]: DICOM, 2013. [Http://http://dicom.nema.org/dicom/2013/output/chtml/part03](http://dicom.nema.org/dicom/2013/output/chtml/part03). Acessado em 22/04/2021. Citado na página 38.
- DICOM. *DICOM Information Model*. [S.l.]: DICOM, 2019. [Http://dicom.nema.org/medical/dicom/current/output/html/part03.html/](http://dicom.nema.org/medical/dicom/current/output/html/part03.html/). Acessado em 22/06/2019. Citado na página 37.
- DICOM. *DICOMWebTM*. [S.l.]: DICOM, 2019. [Https://www.dicomstandard.org/dicomweb/](https://www.dicomstandard.org/dicomweb/). Acessado em 22/06/2019. Citado na página 37.
- DICOM. *Extension of the DICOM Model of the Real World for Specimens*. [S.l.]: DICOM, 2019. [Http://dicom.nema.org/medical/dicom/current/output/html/part03.html/](http://dicom.nema.org/medical/dicom/current/output/html/part03.html/). Acessado em 22/06/2019. Citado 2 vezes nas páginas 17 e 133.
- DICOMWEB. [S.l.]: DICOM, 2019. [Https://www.dicomstandard.org/dicomweb/](https://www.dicomstandard.org/dicomweb/). Acessado em 15/05/2019. Citado na página 32.
- DRAW.IO. *Draw.io*. [S.l.]: Draw.io, 2019. [Https://about.draw.io/](https://about.draw.io/). Acessado em 02/07/2019. Citado na página 50.
- FARAHANI, N.; PARWANI, A. V.; PANTANOWITZ, L. Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathol Lab Med Int*, v. 7, p. 23–33, 2015. Citado 2 vezes nas páginas 35 e 36.
- FIELDING, R. *Representational state transfer (REST). Chapter 5 in Architectural Styles and the Design of Networkbased Software Architectures*. Tese (Doutorado) — Ph. D. Thesis, University of California, Irvine, CA, 2000. Citado na página 32.
- FLOURNOY, B. *How to Calculate the Field of View in a Microscope*. [S.l.]: Sciencing, 2018. [Https://sciencing.com/calculate-field-microscope-7603588.html](https://sciencing.com/calculate-field-microscope-7603588.html). Acessado em 30/04/2021. Citado 2 vezes nas páginas 83 e 84.

- GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de pesquisa*. [S.l.]: Plageder, 2009. Citado 3 vezes nas páginas 53, 54 e 59.
- GIT. *Git*. [S.l.]: Git, 2019. <https://git-scm.com>. Acessado em 12/05/2019. Citado na página 49.
- GITHUB. *GitHub*. [S.l.]: GitHub, 2019. <https://github.com>. Acessado em 12/05/2019. Citado na página 49.
- GONZALES RAFAEL, C.; WOODS RICHARD, E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, 2006. 954 p. P. 152–157. ISBN 978-0131687288. Citado 2 vezes nas páginas 41 e 45.
- GUO, K. et al. Instantscope: a low-cost whole slide imaging system with instant focal plane detection. *Biomedical optics express*, Optical Society of America, v. 6, n. 9, p. 3210–3216, 2015. Citado na página 30.
- HERRMANN, M. D. et al. Implementing the dicom standard for digital pathology. *Journal of pathology informatics*, Wolters Kluwer–Medknow Publications, v. 9, 2018. Citado 5 vezes nas páginas 29, 31, 38, 39 e 70.
- JYOTHI, G.; SUSHMA, C.; VEERESH, D. Luminance based conversion of gray scale image to rgb image. *International Journal of Computer Science and Information Technology Research*, v. 3, n. 3, p. 279–283, 2015. Citado na página 41.
- KARAMI, E.; PRASAD, S.; SHEHATA, M. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017. Citado na página 43.
- KAYSER, K. et al. New developments in digital pathology: from telepathology to virtual pathology laboratory. *Studies in health technology and informatics*, IOS Press; 1999, v. 105, p. 61–69, 2004. Citado 2 vezes nas páginas 29 e 33.
- LOUIS, D. N. et al. Computational pathology: an emerging definition. *Archives of pathology & laboratory medicine*, the College of American Pathologists, v. 138, n. 9, p. 1133–1138, 2014. Citado na página 35.
- MA, S.; WANG, T. swsi: A low-cost and commercial-quality whole slide imaging system on android and ios smartphones. *arXiv preprint arXiv:1704.01088*, 2017. Citado 3 vezes nas páginas 30, 43 e 65.
- MASON, D. Su-e-t-33: pydicom: an open source dicom library. *Medical Physics*, Wiley Online Library, v. 38, n. 6Part10, p. 3493–3493, 2011. Citado na página 50.
- MEYER. *Panoptiq: What Is It?* [S.l.]: Meyer, 2019. <https://www.meyerinst.com/brand/viewsiq/panoptiq/>. Acessado em 02/07/2019. Citado na página 65.
- MICROVISIONEER. *Manual Whole Slide Imaging*. [S.l.]: Microvisioneer, 2019. <http://www.microvisioneer.com/>. Acessado em 02/07/2019. Citado na página 65.
- MILDENBERGER, P.; EICHELBERG, M.; MARTIN, E. Introduction to the dicom standard. *European radiology*, Springer, v. 12, n. 4, p. 920–927, 2002. Citado 2 vezes nas páginas 30 e 37.

MORRISON, A. S.; GARDNER, J. M. Smart phone microscopic photography: a novel tool for physicians and trainees. *Archives of Pathology and Laboratory Medicine*, the College of American Pathologists, v. 138, n. 8, p. 1002–1002, 2014. Citado na página 81.

NASA. *Images from the Perseverance Mars Rover Feb. 22, 2021 Press Briefing*. [S.l.]: Nasa, 2021. <https://www.nasa.gov/content/images-from-the-perseverance-mars-rover-feb-22-2021-press-briefing>. Acessado em 30/04/2021. Citado 2 vezes nas páginas 41 e 42.

NCI. *Definition of Biopsy NCI Dictionary*. [S.l.]: NCI Dictionary, 2019. <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/biopsy>. Acessado em 22/06/2019. Citado na página 33.

OPENCV. *FAST Algorithm for Corner Detection*. [S.l.]: OpenCV, 2019. <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html>. Acessado em 12/05/2019. Citado na página 44.

OPENCV. *OpenCV*. [S.l.]: OpenCV, 2019. <https://opencv.org/about/>. Acessado em 12/05/2019. Citado 2 vezes nas páginas 49 e 63.

OPENCV. *Understanding Feature*. [S.l.]: OPENCV, 2019. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html. Acessado em 05/05/2019. Citado na página 42.

PAGOTTO, T. et al. Scrum solo: software process for individual development. In: IEEE. *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.], 2016. p. 1–6. Citado 6 vezes nas páginas 15, 57, 58, 98, 99 e 102.

POSTDICOM. *Online DICOM Viewer Free*. [S.l.]: DICOM, 2021. <https://www.postdicom.com/en/free-web-based-dicom-viewer>. Acessado em 26/04/2021. Citado 2 vezes nas páginas 50 e 65.

PRODANOV, C. C.; FREITAS, E. C. de. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. [S.l.]: Editora Feevale, 2013. Citado 2 vezes nas páginas 53 e 54.

PYARWIN, K.; KITJAIDURE, Y. Biomedical images stitching using orb feature based approach. In: IEEE. *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*. [S.l.], 2018. v. 3, p. 221–225. Citado na página 43.

QUICKTIME, V. An image-based approach to virtual environment navigation, shenchang eric chen, apple computer, inc. In: *Siggraph, Computer Graphics Proceedings, Annual Conference Series*. [S.l.: s.n.], 1995. p. 29–38. Citado na página 41.

RONI. *Introduction to DICOM - Chapter 4 - DICOM Objects*. [S.l.]: Roni, 2011. <http://dicomiseasy.blogspot.com/2011/12/chapter-4-dicom-objects-in-chapter-3.html>. Acessado em 30/04/2021. Citado na página 38.

ROSIN, P. L. Measuring corner properties. *Computer Vision and Image Understanding*, Elsevier, v. 73, n. 2, p. 291–307, 1999. Citado na página 45.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: SPRINGER. *European conference on computer vision*. [S.l.], 2006. p. 430–443. Citado na página 43.

- ROY, S. et al. Smartphone adapters for digital photomicrography. *Journal of pathology informatics*, Wolters Kluwer–Medknow Publications, v. 5, 2014. Citado 2 vezes nas páginas 30 e 39.
- RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: CITESEER. *ICCV*. [S.l.], 2011. v. 11, n. 1, p. 2. Citado 3 vezes nas páginas 43, 45 e 46.
- SPANHOL, F. A. et al. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 63, n. 7, p. 1455–1462, 2016. Citado 3 vezes nas páginas 34, 63 e 64.
- SZELISKI, R. et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, Now Publishers, Inc., v. 2, n. 1, p. 1–104, 2007. Citado 4 vezes nas páginas 41, 42, 46 e 47.
- TREANOR, D. et al. The leeds guide to digital pathology. University of Leeds, Leeds Teaching Hospitals, 2018. Citado na página 34.
- TRIPP, D. Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa*, SciELO Brasil, v. 31, n. 3, p. 443–466, 2005. Citado na página 54.
- UCLA. *Anatomic Pathology*. [S.l.]: UCLA, 2019. [Http://pathology.ucla.edu/anatomic-pathology](http://pathology.ucla.edu/anatomic-pathology). Acessado em 22/06/2019. Citado na página 33.
- VYAS, K. *Image Stitching: A Simplistic Tutorial*. [S.l.]: Kushal Vyas, 2019. [Https://kushalvyas.github.io/stitching.html/](https://kushalvyas.github.io/stitching.html/). Acessado em 21/06/2019. Citado 2 vezes nas páginas 63 e 117.

Apêndices

APÊNDICE A – Fluxograma do Algoritmo de Image Stitching

Esse fluxograma tem como base os algoritmos definidos por Vyas (2019) sendo esses utilizados no trabalho realizado na disciplina de Visão Computacional, como discutido no capítulo 5.

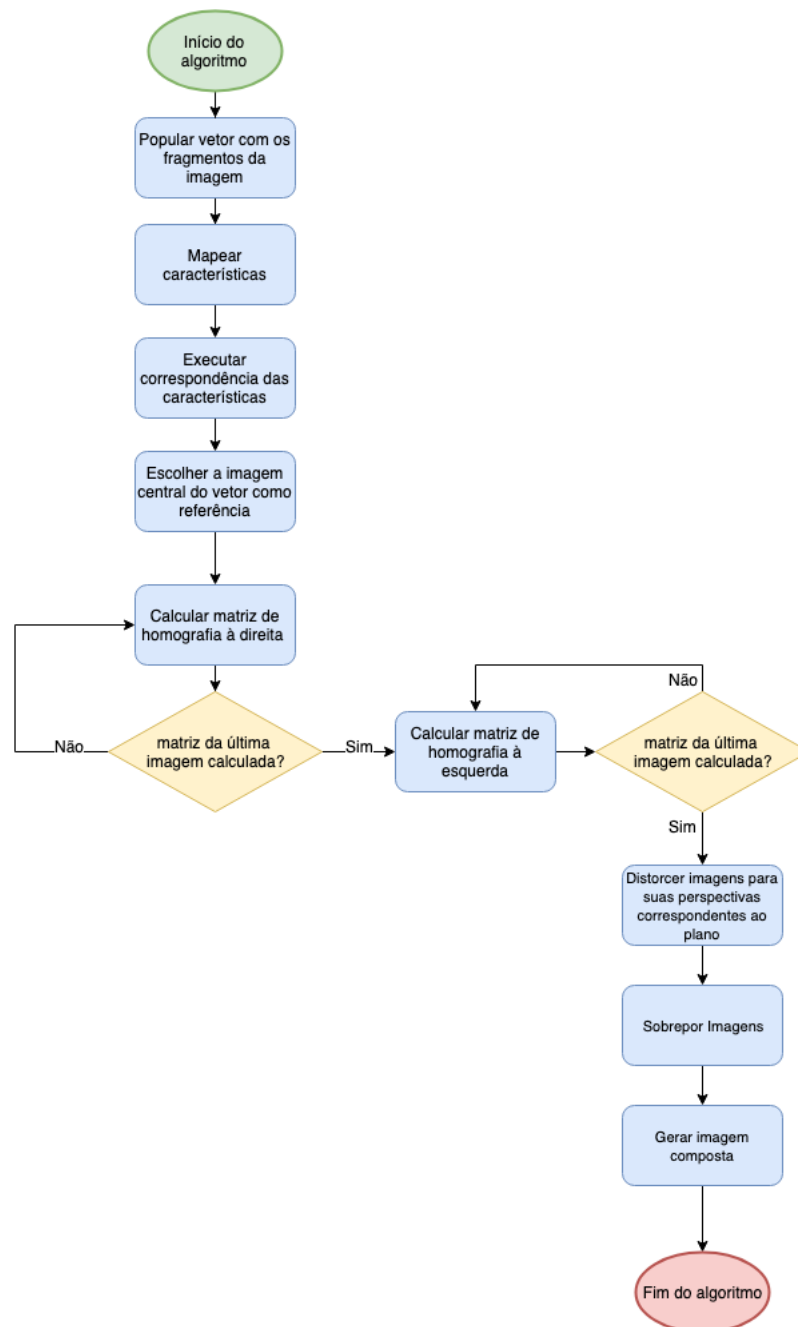


Figura 49 – Fluxograma do algoritmo de *Image Stitching*.

APÊNDICE B – Notação *C4 Model*

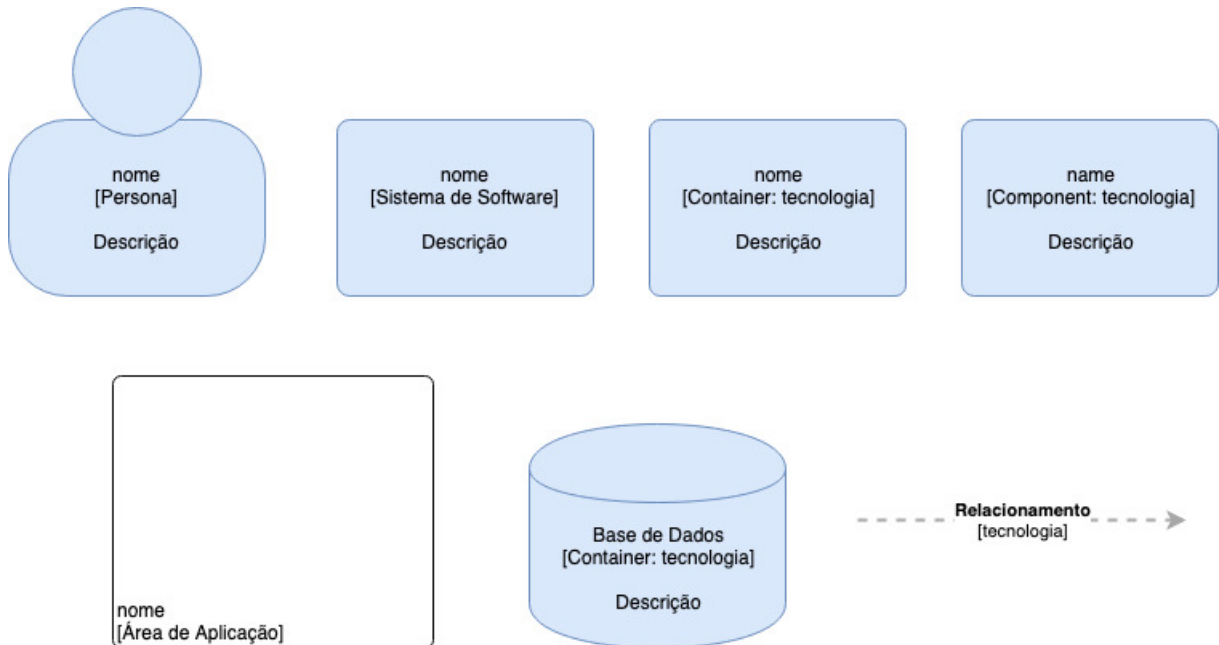
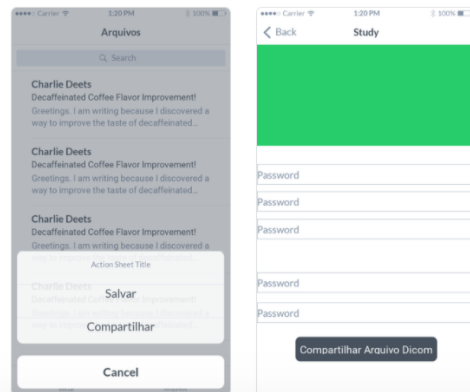
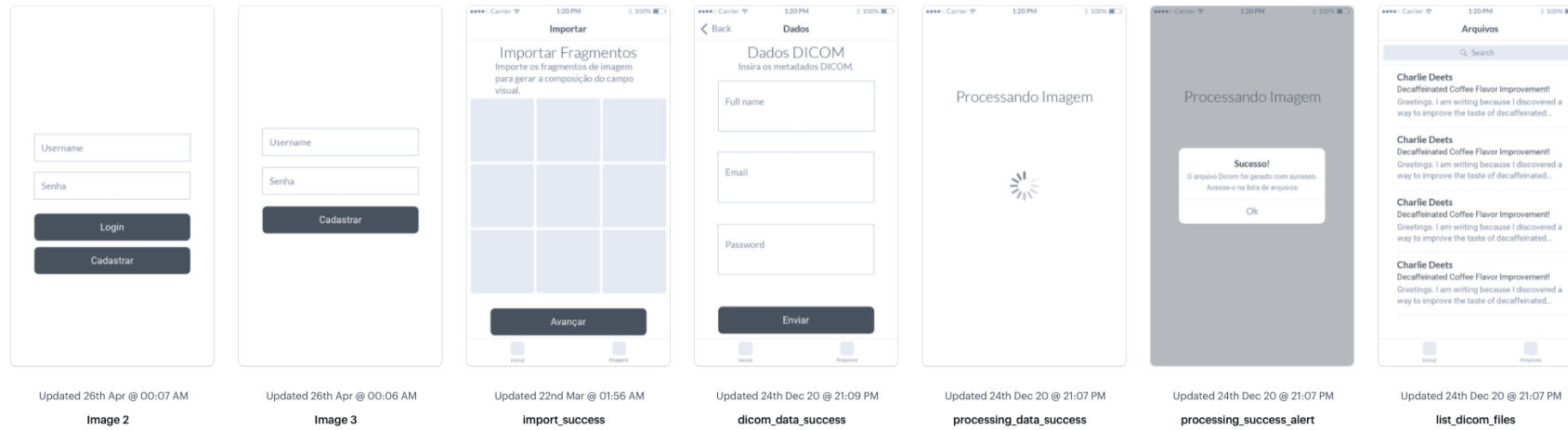


Figura 50 – Notação *C4 Model* utilizando o software *Draw.io*

APÊNDICE C – Protótipo do Mosaic

Figura 51 – Protótipos criados no *Marvel* e gerados a partir do desenvolvimento como artefatos do *Scrum Solo*.



Fonte: Autor.

APÊNDICE D – Análise dos Resultados

Figura 52 – Questionário aplicado ao especialista e suas respostas.

10/05/2021

Digitalização de Lâminas Histopatológicas Utilizando Smartphone: Adequação ao Padrão DICOM

Digitalização de Lâminas Histopatológicas Utilizando Smartphone: Adequação ao Padrão DICOM

Esta pesquisa servirá como resultado de opinião do especialista.

Em termos de praticidade, qual a sua opinião sobre o adaptador de acoplamento entre o smartphone e o microscópio? *

0 1 2 3 4 5

Nada prático Muito prático

De acordo com a sua experiência e comparando com o uso do adaptador, qual a melhor forma de capturar uma imagem do microscópio a partir de um smartphone? *

Posicionando o smartphone manualmente

Utilizando o adaptador

em termos de proposta, qual a utilidade do Mosaic como sistema de digitalização de lâminas de custo acessível? Marque uma opção de 0 a 5 sendo 0 nenhuma utilidade e 5 muita utilidade. *

0 1 2 3 4 5

Nenhuma utilidade Muita utilidade

<https://docs.google.com/forms/d/1C1HVjkXFLlLK0paYerKkn38iaL573VgJrnjultwPPpw/edit#responses>

1/3

Fonte: Autor.

Digitalização de Lâminas Histopatológicas Utilizando Smartphone: Adequação ao Padrão DICOM

Esta pesquisa servirá como resultado de opinião do especialista.

Em termos de praticidade, qual a sua opinião sobre o adaptador de acoplamento entre o smartphone e o microscópio? *

0 1 2 3 4 5

Nada prático Muito prático

De acordo com a sua experiência e comparando com o uso do adaptador, qual a melhor forma de capturar uma imagem do microscópio a partir de um smartphone? *

- Posicionando o smartphone manualmente
- Utilizando o adaptador

em termos de proposta, qual a utilidade do Mosaic como sistema de digitalização de lâminas de custo acessível? Marque uma opção de 0 a 5 sendo 0 nenhuma utilidade e 5 muita utilidade. *

0 1 2 3 4 5

Nenhuma utilidade Muita utilidade

Em termos de praticidade, como você classificaria o Mosaic? *

Nada prático 0 1 2 3 4 5 Muito prático

O que poderia haver de melhorias no Mosaic para a evolução do software? *

Poderia ser desenvolvido para câmeras de microscopia já existentes, não só smartphones

Considerando o Mosaic nesta primeira versão e com as melhorias apontadas, para quais fins você o utilizaria? *

Digitalizar lâminas em pequeno aumento para dar noção panorâmica em discussões de caso clínico

Você já conhecia o padrão DICOM? *

Sim
 Não

Você já utilizou o padrão DICOM para imagens histopatológicas ou outros tipos de imagens médicas? *

Sim
 Não
 Outro:

Qual a sua opinião sobre a adoção do padrão DICOM em um ambiente laboratorial? *

Seria factível, já que a radiologia usa rotineiramente

Autoriza citar seu nome e/ou do laboratório neste trabalho? *

Sim

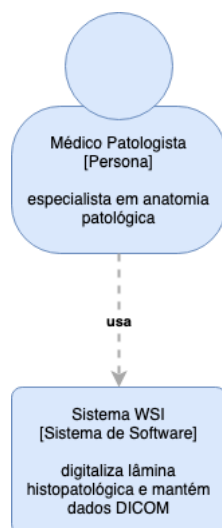
Não

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

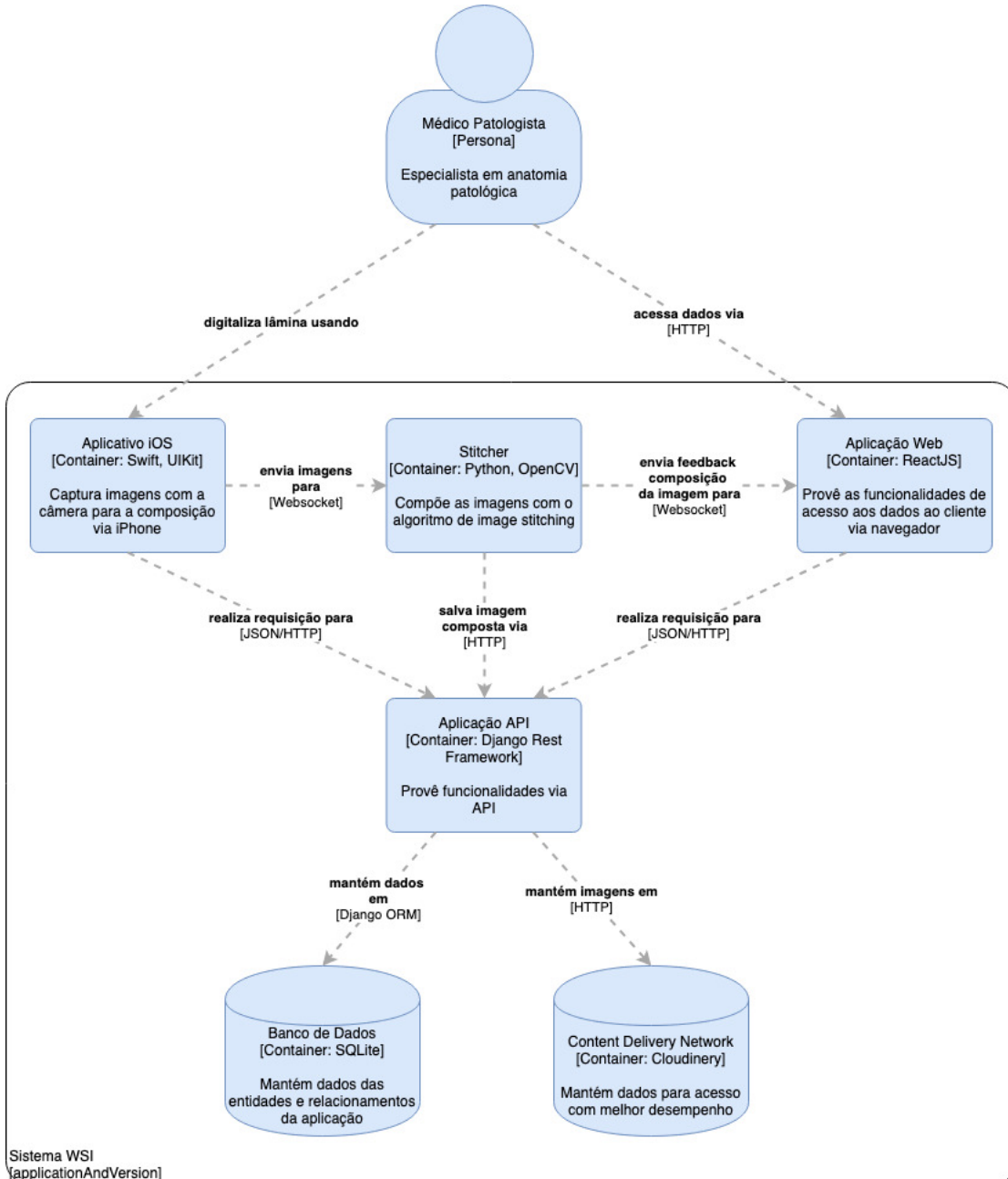
APÊNDICE E – Artefatos de Proposta

Figura 53 – Proposta de Nível 1 (Contexto).



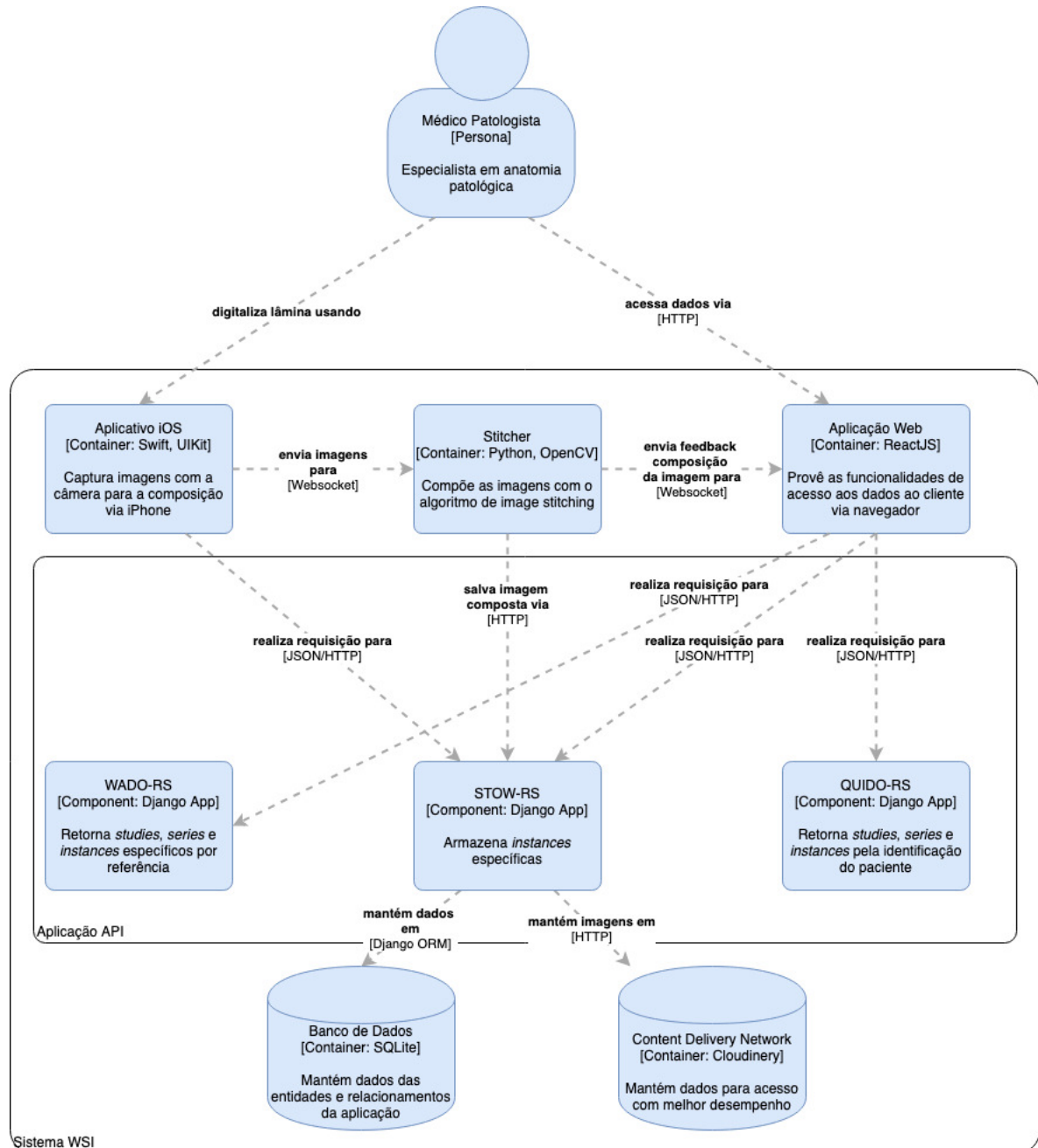
Fonte: Autor

Figura 54 – Proposta de Nível 2 (Container).



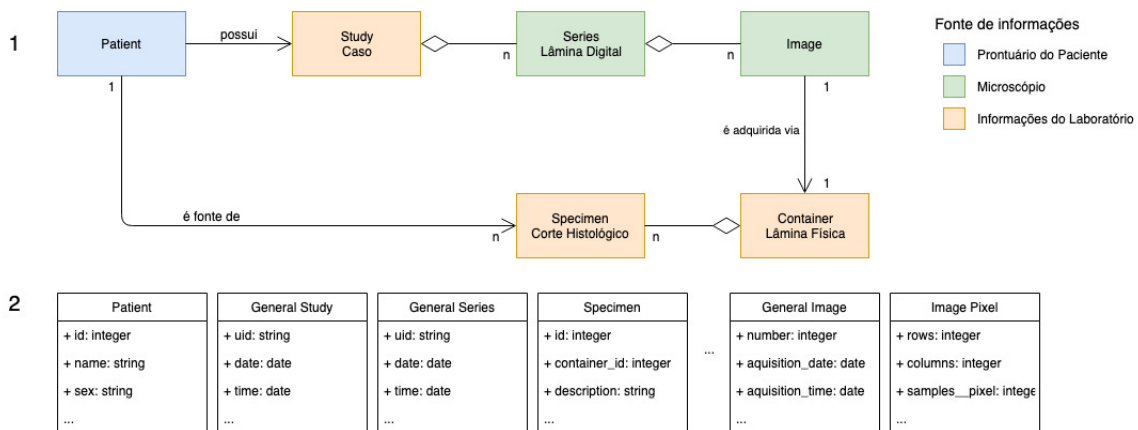
Fonte: Autor

Figura 55 – Proposta de Nível 3 (Componente)



Fonte: Autor

Figura 56 – Proposta de Nível 3 (Código). (1) Representação DICOM *Data Model*. (2) Objetos definidos na *VL Whole Slide Microscopy Image*.



Fonte: Autor

Anexos

ANEXO A – Modelo Entidade Relacionamento DICOM para Espécimes

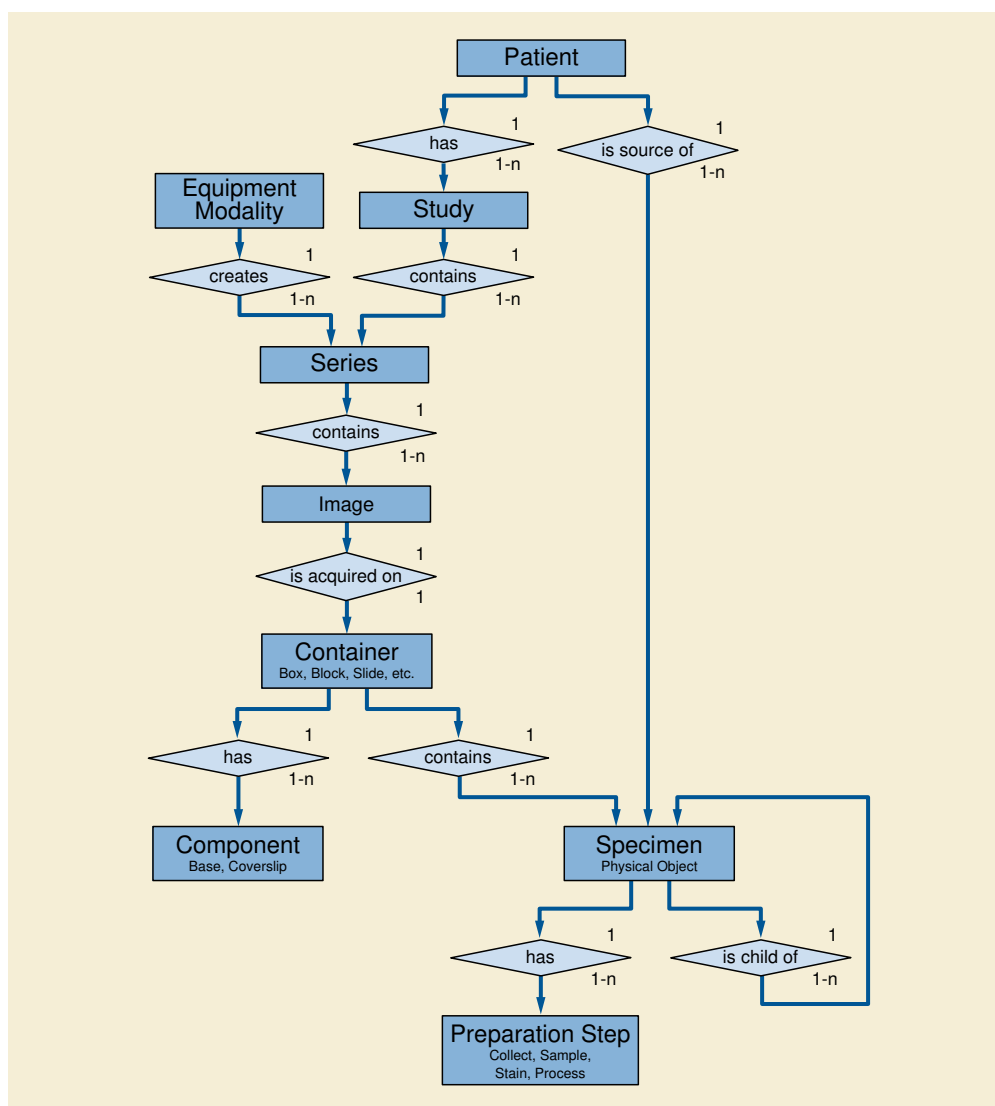


Figura 57 – Modelo DICOM (DICOM, 2019c)

ANEXO B – DICOM *Information Model*

Esquema das estrutura de informações definidos pelo padrão DICOM.

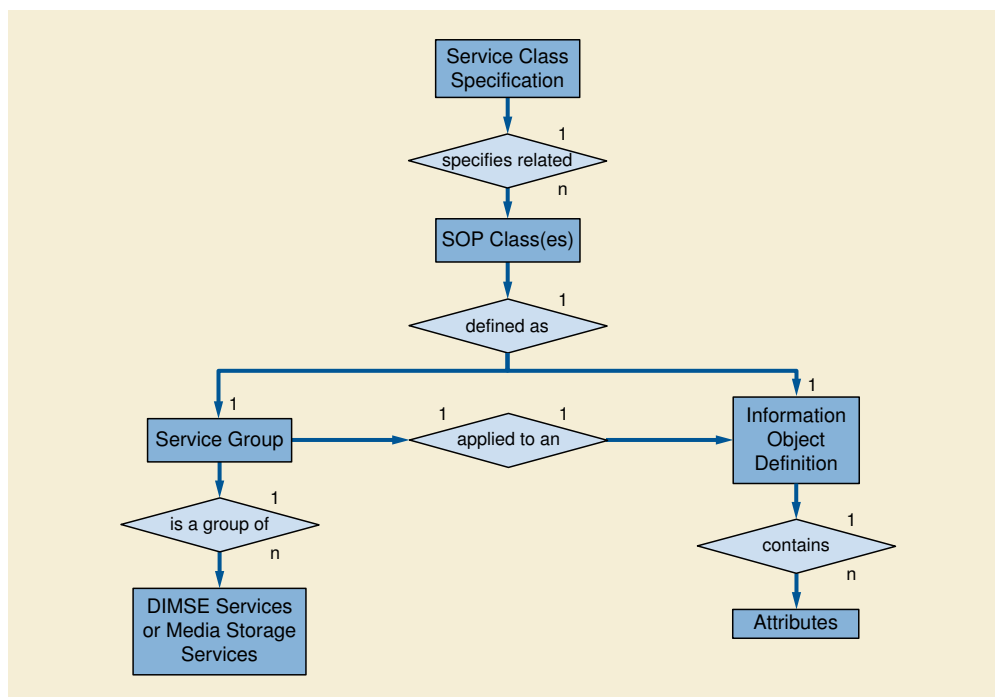


Figura 58 – Modelo DICOM *Information Model*