



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

TOCANTINS
GOVERNO DO ESTADO



CURSO DE SISTEMAS DE INFORMAÇÃO

FERRAMENTA DE RECONHECIMENTO DE IMAGENS PARA UM TESTE EXPRESSIVO DE PERSONALIDADE

HEYTOR VICTOR ARAUJO SILVA

Palmas

2021



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

TOCANTINS
GOVERNO DO ESTADO



CURSO DE SISTEMAS DE INFORMAÇÃO

FERRAMENTA DE RECONHECIMENTO DE IMAGENS PARA UM TESTE EXPRESSIVO DE PERSONALIDADE

HEYTOR VICTOR ARAUJO SILVA

Trabalho apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Tocantins - UNITINS como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob a orientação do professor Me. Jânio Elias Teixeira Júnior e coorientação da professora Me. Muriel Corrêa Neves Rodrigues.

Palmas

2021



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

TOCANTINS
GOVERNO DO ESTADO



CURSO DE SISTEMAS DE INFORMAÇÃO

FERRAMENTA DE RECONHECIMENTO DE IMAGENS PARA UM TESTE EXPRESSIVO DE PERSONALIDADE

HEYTOR VICTOR ARAUJO SILVA

Trabalho apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Tocantins - UNITINS como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob a orientação do professor Me. Jânio Elias Teixeira Júnior e coorientação da professora Me. Muriel Corrêa Neves Rodrigues.

Prof. Me. Jânio Elias Teixeira Júnior
Orientador

Prof. Me. Muriel Corrêa N. Rodrigues
Coorientadora

Prof. Me. Marco Antonio F. de Sousa
Convidado 1

Prof. Me. Alex Coelho
Convidado 2

Palmas
2021

Dedico este trabalho aos meus pais.

Agradecimentos

Agradeço em primeiro lugar a Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho.

Agradeço aos meus pais, por todo esforço e empenho que me fizeram chegar até a conclusão deste trabalho.

Agradeço aos meus orientadores, por todas correções, ensinamentos e conselhos com a qual guiaram o meu aprendizado.

Agradeço ao grupo de colegas de turma, que se tornou uma grande amizade, criado nos primeiros períodos de graduação apelidado de "GdP", pelo companheirismo e de ajuda.

Agradeço ao Gabriel Brito, que me auxiliou a solucionar minhas dúvidas que tive durante este trabalho sobre o Flutter.

Agradeço a todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

"Através do humor vemos no que parece racional, o irracional; no que parece importante, o insignificante. Ele também desperta o nosso sentido de sobrevivência e preserva a nossa saúde mental." (Charles Chaplin)

Resumo

Um teste de personalidade é um mecanismo empregado para a verificação das particularidades do comportamento humano. O palográfico, um dos tipos de testes expressivos, tem por objetivo avaliar alguns aspectos da personalidade, como: o ritmo de trabalho, a organização, as relações interpessoais, a produtividade dentre outros. Apenas profissionais de psicologia registrados no Conselho Regional de Psicologia estão aptos para aplicação de testes. Atualmente existe um sistema de digitalização para correção, entretanto, para essa análise se faz necessário o uso de scanners, aumentando a complexibilidade no manuseio. Outra opção de correção se dá pela forma manual, consequentemente, mais demorada, pelo fato do avaliador examinar vários aspectos do teste para um resultado fidedigno. Este trabalho tem como propósito a elaboração de uma ferramenta automatizada para análise de testes palográficos, bem como a detecção de especificidades dos traços (palos) contidos no teste, através de um aplicativo móvel e sua câmera. A análise das imagens dos testes foi realizada por meio de algoritmos e bibliotecas de visão computacional e reconhecimento de padrões, por intermédio da linguagem de programação Python. Para a avaliação da qualidade dos resultados da ferramenta proposta, conduziu-se um comparativo entre os testes com correções já realizadas de forma manual, tal qual os resultados obtidos pelo software SKIP.

Palavras-chave: teste palográfico, visão computacional, reconhecimento de padrões.

Abstract

A personality test is a mechanism used to verify the particularities of human behavior. The palographic, one of the types of expressive tests, aims to assess some personality aspects, such as: the work pace, organization, interpersonal relationships, productivity among others. Only psychology professionals registered with the Regional Council of Psychology are eligible to apply tests. Currently there is a scanning system for correction, however, for this analysis, it is necessary to use scanners, increasing the complexity of handling. Another correction option is the manual method, which is therefore more time-consuming, as the evaluator examines various aspects of the test for a reliable result. This work aims to develop an automated tool for the analysis of palographic tests, as well as the detection of specific features of the traces (palos) contained in the test, through a mobile application and its camera. The analysis of test images will be performed using computer vision and pattern recognition algorithms and libraries, using the programming language Python. To assess the quality of the results of the proposed tool, a comparison will be made between the tests with corrections already performed manually, just like the results obtained by the SKIP software.

Key-words: palographic test, computer vision, pattern recognition.

Lista de ilustrações

Figura 1 – Exemplo de Imagem Monocromática	17
Figura 2 – Fluxograma do Processamento de Imagens	18
Figura 3 – Etapas do Processo de Pré-Processamento	19
Figura 4 – Hierarquia dos Widgets no Flutter	25
Figura 5 – Fluxo de Interação do Docker	26
Figura 6 – Fluxograma da Metodologia	31
Figura 7 – Protótipo de Detecção	34
Figura 8 – Arquitetura da Ferramenta	39
Figura 9 – Diagrama de Caso de Uso	42
Figura 10 – Diagrama de Atividade	44
Figura 11 – Diagrama de Entidade Relacionamento	45
Figura 12 – Diagrama de Classes	47
Figura 13 – Tela de Login	50
Figura 14 – Tela de Cadastro do Psicólogo	50
Figura 15 – Tela de Menu	51
Figura 16 – Tela de Listagem dos Psicológos	52
Figura 17 – Tela de Cadastro do Examinado	53
Figura 18 – Tela de Listagem do Examinado	53
Figura 19 – Tela de Cadastro do Palográfico	54
Figura 20 – Tela de Detalhamento do Palográfico	54
Figura 21 – Tela de Detalhamento da Análise do Palográfico	55
Figura 22 – Fluxograma do Processamento de Imagem	56
Figura 23 – Teste Palográfico - Original	58
Figura 24 – Teste Palográfico - Scanner	60
Figura 25 – Teste Palográfico - Separação	61
Figura 26 – Teste Palográfico - Ordenação	62
Figura 27 – Rotas da API	65
Figura 28 – Exemplo da Autenticação na API pelo JWT	66
Figura 29 – Gráfico Produtividade	68
Figura 30 – Gráfico Rendimento	69
Figura 31 – Gráfico Soma da Diferença	70
Figura 32 – Gráfico NOR	71
Figura 33 – Gráfico Distância entre Palos	72
Figura 34 – Gráfico Tamanho dos Palos	73
Figura 35 – Identificação de Palos Conectados no Teste 4	74
Figura 36 – Gráfico Impulsividade	75

Lista de quadros

Quadro 1 – Hardware	38
Quadro 2 – Requisitos Não Funcionais	41
Quadro 3 – Exemplo dos Métodos REST na Rota /palographic	65
Quadro 4 – Métricas de Análise no Teste Palográfico	67
Quadro 5 – Comparação FRITE x SKIP	76

Lista de tabelas

Tabela 1 – RF001 Cadastrar Avaliador	82
Tabela 2 – RF002 Realizar Login	82
Tabela 3 – RF003 Cadastrar Examinado	83
Tabela 4 – RF004 Cadastrar Teste Palográfico	84
Tabela 5 – RF005 Permitir Upload de Imagem	84
Tabela 6 – RF006 Realizar Pré-processamento da Imagem	85
Tabela 7 – RF007 Identificar e Realizar Análises nos Palos	85
Tabela 8 – RF008 Retornar Relatório com as Informações Obtidas	86

Lista de abreviaturas e siglas

API - *Application Programming Interface.*

CMY - *Cyan, Magenta e Yellow.*

CMYK - *Cyan, Magenta, Yellow e Black.*

HD - *High Definition.*

HSI - *Hue, Saturation e Intensity.*

HTTP - *Hyper Text Transfer Protocol.*

JSON - *JavaScript Object Notation.*

POO - Programação Orientada a Objetos.

REST - *Representation State Transfer.*

RGB - *Red, Green e Blue.*

RF - Requisito Funcional.

RNF - Requisito Não Funcional.

UI - *User Interface.*

UML - *Unified Modeling Language.*

URI - *Uniform Resource Identifier.*

UX - *User Experience.*

VC - Visão Computacional.

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
2	REFERENCIAL TEÓRICO	16
2.1	Imagen Digital	16
2.2	Visão Computacional	17
2.2.1	Aquisição da Imagem	18
2.2.2	Pré-processamento	19
2.2.3	Segmentação	19
2.2.4	Extração	20
2.2.5	Reconhecimento	20
2.3	Reconhecimento de Padrões	21
2.4	Teste Palográfico	22
2.5	Tecnologias	23
2.5.1	Python	23
2.5.2	Django	23
2.5.3	Dart	24
2.5.4	Flutter	24
2.5.5	Docker	25
2.5.6	OpenCV	26
2.6	Trabalhos Relacionados	27
3	METODOLOGIA	29
3.1	Caracterização da Metodologia	29
3.2	Implementação dos Métodos	30
3.3	Definição das Tecnologias	34
3.3.1	Linguagem do Algoritmo do Processamento de Imagem	35
3.3.2	Framework Python	35
3.3.3	Framework Desenvolvimento Híbrido	36
3.3.4	Banco de Dados	36
3.4	Avaliação de Desempenho	37
3.4.1	Métrica de Eficiência do Processamento de Imagem	37
3.4.2	Métrica de Qualidade da Imagem	37
3.5	Materiais	38

3.6	Orçamento	38
4	RESULTADOS	39
4.1	Arquitetura da Aplicação	39
4.2	Implementação	40
4.2.1	Engenharia de Software	40
4.2.1.1	Levantamento de Requisitos	40
4.2.1.1.1	Requisitos Funcionais	41
4.2.1.1.2	Requisitos Não Funcionais	41
4.2.1.2	Diagramas	41
4.2.1.2.1	Diagrama de Caso de Uso	41
4.2.1.2.2	Diagrama de Atividade	42
4.2.1.2.3	Diagrama de Entidade Relacionamento	45
4.2.1.2.4	Diagrama de Classe	46
4.2.2	Desenvolvimento	49
4.2.2.1	Aplicativo	49
4.2.2.2	Algoritmo do Processamento de Imagem	55
4.2.2.3	API	64
4.3	Estudo de Caso	66
4.3.1	Produtividade	67
4.3.2	Rendimento	68
4.3.3	Soma da Diferença	69
4.3.4	NOR	70
4.3.5	Distância entre Palos	71
4.3.6	Tamanho dos Palos	72
4.3.7	Impulsividade	74
4.3.8	Considerações Finais	75
5	CONCLUSÃO	77
5.1	Trabalhos Futuros	78
A	REFERÊNCIAS	79
A	REQUISITOS FUNCIONAIS	82

1 Introdução

Os smartphones vêm recebendo uma serventia importante nas mais diversas ocupações da sociedade no contexto atual, impulsionando as áreas de comunicação, mobilidade, finanças, entretenimento, saúde, entre outras.

Meirelles (2020) desenvolveu uma pesquisa realizada pelo Centro de Tecnologia de Informação Aplicada da Fundação Getúlio Vargas, apresentando que existem 424 milhões de dispositivos digitais em operação no Brasil. Para ele, é a comprovação do processo de transformação digital em todo país e carrega a grandeza da importância da tecnologia. Tendo em foco os dispositivos móveis, a pesquisa aponta a existência de mais de um celular para cada habitante.

Diante do vasto uso de dispositivos eletrônicos, que por sua vez são conectados a rede, Hallberg (2015) elaborou um estudo com base em psicólogos do Rio Grande do Sul em que apontaram o crescimento da utilização das tecnologias no cotidiano, sendo que 77,6% dos pesquisados utilizam a internet para averiguar informações na área da psicologia. De mesmo modo, os psicólogos, de diversos campos de atividade, sofrem influências desses dispositivos tanto em suas vidas pessoais como profissionais.

A psicologia, enquanto profissão, vem se aproximando da modernidade trazida pelos avanços tecnológicos, visto que pode ser empregada como instrumento de trabalho, maximizando as possibilidades nessa área (LEMOS, 2015). Para Limonge (2006), no seu artigo "Testes Informatizados - Uma nova visão", relata um futuro promissor para a informatização dos testes, promovido pelo crescimento da internet, além de oferecer compartilhamento de experiências entre os psicólogos.

Com a evolução tecnológica na psicologia, houve a elaboração de um sistema chamado SKIP, instruído para a digitalização de teste de personalidade (palográfico), assim como a análise das informações. Contudo, este software carece da aquisição de scanners para realizar as análises.

O presente trabalho se justifica com base no atual cenário da psicologia em que ocorre uma etapa arduosa durante a correção de testes palográficos, uma vez que é necessário a medição dos palos para obter um resultado. Nesse sentido, a proposta visa agilizar e aperfeiçoar a correção para os profissionais desta área, de forma objetiva e comprehensível na sua utilização. Para a avaliação desta proposta, um estudo de caso foi desenvolvido utilizando uma métrica na análise, contendo os resultados encontrados durante a execução deste trabalho.

Este trabalho está constituído no seguinte formato: na continuação deste Capítulo

(1), tem-se o objetivo geral e os específicos. No Capítulo 2, expõe a base conceitual abordada neste trabalho, constituída sobre os temas: imagem digital, visão computacional, reconhecimento de padrões, teste palográfico e as linguagens de programação que serão utilizadas. No Capítulo 3, é apresentado toda metodologia para compor a aplicação, bem como explicações sobre o desenvolvimento. Em seguida, no Capítulo 4, é exibido os resultados deste trabalho e, por fim, o último Capítulo (5) apresenta as contribuições e considerações finais.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho visa desenvolver uma aplicação para análise e correção do teste psicológico palográfico, através de técnicas de reconhecimento de imagens.

1.1.2 Objetivos Específicos

- Produzir uma revisão bibliográfica dos assuntos relacionados a este trabalho.
- Desenvolver microserviços para a comunicação entre: o aplicativo; o processamento de imagem; e o banco de dados.
- Realizar pré-processamento das imagens recebidas e desenvolver algoritmos para detecção e classificação dos palos.
- Desenvolver uma ferramenta gráfica para seleção de testes e apresentação dos resultados.
- Analisar os resultados gerados, através da elaboração de um estudo de caso, entre: a ferramenta deste trabalho; a correção manual feita por um psicólogo; e a correção da ferramenta SKIP.

2 Referencial Teórico

Este capítulo apresenta o referencial teórico direcionado para os seguintes tópicos: imagem digital, visão computacional, reconhecimento de padrões, teste palográfico e tecnologias que serão utilizadas nesse projeto.

As imagens geradas por uma variedade de aparelhos físicos, convertidas para um conjunto de números conhecido como pixels, o concebe a classificação de imagem digital. Esta imagem, por sua vez, demanda o aperfeiçoamento e tratamento como, por exemplo, a remoção de ruído e o realce na qualidade. Este processo é conhecido como visão computacional, tendo o objetivo de que a máquina consiga interpretar a imagem, de forma mais concisa e limpa. Deste modo, o computador se torna capaz de identificar e analisar medidas, ou seja, executar o reconhecimento de padrões, através do uso de algumas tecnologias e linguagens de programação que facilitam o procedimento ([ALBUQUERQUE et al., 2012](#)).

2.1 Imagem Digital

Uma imagem pode ser determinada como uma função bidimensional, $f(x, y)$, em que as coordenadas do plano são representadas por x e y, bem como a intensidade (brilho) é retratada pela amplitude de f . No momento em que os valores das variáveis possuem quantidades discretas e finitas, denomina-se imagem digital ([GONZALEZ; WOODS, 2000](#)).

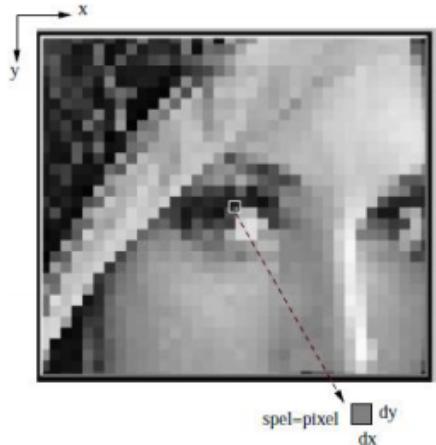
Para que um computador seja capaz de realizar o processamento dessas imagens, é necessário representá-las como arranjos bidimensionais de pontos, uma vez em que a máquina está habilitada somente a reconhecer números digitais ([QUEIROZ; GOMES, 2006](#)). Cada ponto com sua respectiva localização e valor é chamado de elementos de imagem, ou pixels. A classificação das imagens digitais é baseada na sua origem, podendo ser coloridas, nível de cinza ou binárias.

A imagem digital recebe a representação de binária quando dispõe apenas dois valores para cada pixel, 0 para o preto e 1 para o branco. Tem, por sua vez, uma vantagem em relação as outras classificações, necessitam de pouca alocação de armazenamento e beneficia a utilização de operadores lógicos ([ERPEN, 2004](#)).

As imagens monocromáticas, ou nível de cinza, empregam os valores dos pixels correspondentes a cada elemento da matriz entre 0 para o preto e 255 para o branco, em outras palavras, valor mínimo representa a ausência de intensidade enquanto o valor máximo representa a intensidade máxima ([LIBERMAN, 1997](#)). A Figura 1 demonstra a

imagem monocromática com a representação da intensidade do nível de cinza.

Figura 1 – Exemplo de Imagem Monocromática



Fonte: Falcão (2003)

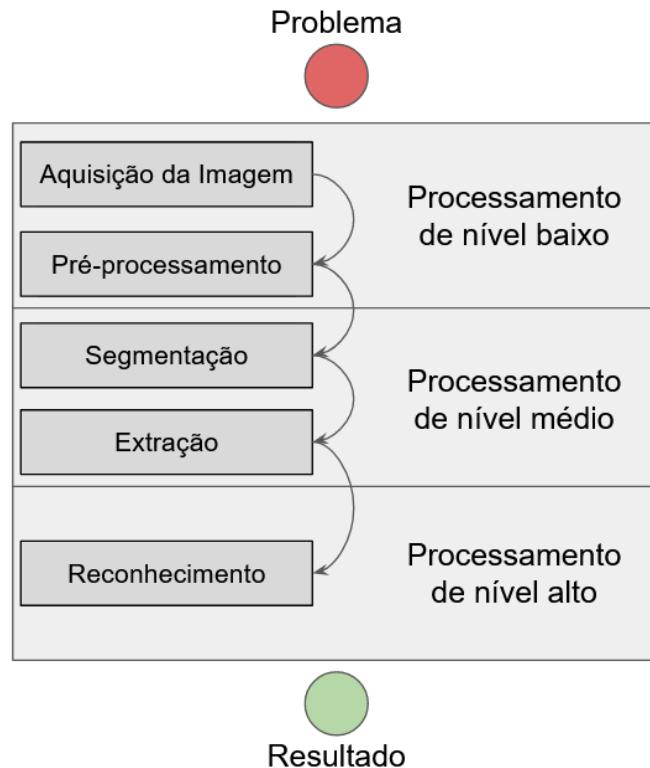
As imagens digitais classificadas como coloridas, recebem duas representações: modelo RGB e tabela de cores. Na primeira, o pixel consome 24 bits nos tons de vermelho, verde e azul. No entanto, já na segunda forma, possui um código específico para cada cor, caracterizando um índice na tabela para cada pixel. De mesmo modo, existem outros modelos de cores como HSI, CMY e o ([ERPEN, 2004](#)).

2.2 Visão Computacional

Visão computacional é a área da ciência em que pode ser definida como o processo de extrair e interpretar informações úteis nas imagens, vídeos ou qualquer estrutura de dados multi-dimensional capturados por câmeras, satélites, sensores, entre outros ([SILVA et al., 2014](#)).

A Visão Computacional (VC) busca igualar a visão humana, a qual também possui uma imagem como entrada, entretanto, a saída pode ser uma compreensão parcial ou completa da imagem. O emprego da visão computacional pode potencializar as diversas aplicações em computadores, inclusive para o cotidiano da sociedade, aparecendo em: aplicativos de reconhecimento facial, leitura em códigos de barras, etc., e englobando nas variadas áreas do conhecimento, como saúde, física, matemática, e outros ([MARENCONI; STRINGHINI, 2009](#)).

Figura 2 – Fluxograma do Processamento de Imagens



Fonte: Baseada na Figura de Erpen (2004)

A análise que inicia no processamento de imagens até a VC pode ser separado em três níveis: nível baixo, nível médio e nível alto. O primeiro consiste em procedimentos básicos como melhoria de contraste ou redução no ruído. Processo de nível médio são os procedimentos de particionamento da imagem. Por fim, o último está relacionado as tarefas de entendimento congruentes a visão humana ([MARENCONI; STRINGHINI, 2009](#)). A Figura 2 mostra o fluxograma no processamento de imagens.

2.2.1 Aquisição da Imagem

É o primeiro estágio do processo. No decorrer dessa fase, sensores ópticos são encarregados da aquisição da captura e digitalização da imagem através dos sinais emitidos por fontes de luz ([ERPEN, 2004](#)).

Essa captura, podendo ser realizada por diversos equipamentos como scanners e câmeras, é efetuada baseado na taxa de amostragem, que indica a resolução da imagem. Embasado nesse escopo, as imagens são convertidas para um formato e propriamente salvas ([JAIN, 1989](#)).

2.2.2 Pré-processamento

De acordo com Cerqueira (2016), as imagens manipuladas pelas máquinas computacionais são capturadas por outros equipamentos citados anteriormente, e por ser desconhecido a qualidade de sua aquisição, está suscetível a imprecisões. Considera-se pré-processamento os procedimentos de nível baixo da abstração, no qual a entrada e saída são imagens (SONKA; HLAVAC; BOYLE, 2014).

Com o propósito das etapas do processamento de imagens sejam possíveis de serem desempenhadas de modo efetivo, nesse estágio se faz necessário verificar se a imagem possui as condições do tratamento (TRENTINI; GODOY; MARANA, 2010).

"São utilizadas técnicas para aumento de contraste, remoção de ruídos, realce e normalização, com o objetivo de converter os padrões para uma forma que possibilite uma simplificação do posterior processo de reconhecimento" (SOUZA, 2010).

Esses procedimentos desempenhados no pré-processamento das imagens são: a filtragem, as correções geométricas e a correção de histograma (GONZALEZ; WOODS, 2000). A Figura 3 retrata a remoção de ruídos em uma imagem digital com o manuseio de filtros.

Figura 3 – Etapas do Processo de Pré-Processamento



Fonte: Gomes (2009)

As técnicas usadas no pré-processamento, são categorizadas de duas formas: executando no domínio espacial e no domínio da frequência. No espacial, é relacionado ao plano da imagem, baseadas no controle direto dos pixels da imagem. Já no domínio da frequência, é fundamentada na Transformada de Fourier da imagem, visto que ela possibilita uma avaliação apropriada para funções não periódicas (SOUZA, 2010).

2.2.3 Segmentação

O terceiro estágio pode ser considerado o principal no processo da visão computacional. Nele, acontece a segregação da imagem em regiões de interesse e de não interesse (FILHO; NETO, 1999).

O reconhecimento dessas regiões podem ser feitas de duas formas, pelo agrupamento dos vizinhos com atributos similares ou mediante a determinação da borda da região (ERPEN, 2004).

Essa parte refere-se a uma precondição dos procedimentos consequentes mais complexos tendo como exemplo o reconhecimento de objetos. A eficiência de que é conduzida a etapa de segmentação concerne totalmente a performance de análise da imagem (GOMES, 2009).

A segmentação de imagens monocromáticas são realizadas de acordo com duas propriedades: a similaridade e a descontinuidade. Na primeira, é fundamentada em técnicas de limiarização e divisão de regiões. Já na segunda, o fracionamento é realizado pelos subconjuntos de pontos referente ao objeto afastado da imagem (GONZALEZ; WOODS, 2000).

2.2.4 Extração

No quarto estágio, acontece o método de extração das informações, as quais representam os dados de entrada convertidos para o conjunto de características, tendo base pelo discernimento e na classificação, com o objetivo de terem reconhecimento (FACON, 1993; ANTONELLO, 2018)

Esta etapa também é conhecida como Representação e Descrição. A imagem até este momento do processamento continua sendo a entrada, enquanto a agregação de dados equivalentes à respectiva imagem resultam na saída (FILHO; NETO, 1999).

Existem dois modos de representar os dados: interna (regiões completas) e externa (fronteiras). A representação interna é apropriada na ocasião em que a relevância é baseada nos pixels daquela região. Em contrapartida, na representação externa, a importância é baseada na relevância nas peculiaridades externas, como pontos de inflexão e cantos (GONZALEZ; WOODS, 2000).

Ou melhor, neste processo, a imagem é retratada como um vetor numérico com o propósito de reduzir a redundância dos dados, além de diminuir a dimensionalidade. Desta forma, a qualidade na extração é primordial a fim de que na fase seguinte, alcance o objetivo (ERPEN, 2004).

2.2.5 Reconhecimento

No quinto e último estágio, ocorre a classificação dos dados extraídos no processo anterior. Compreende-se no reconhecimento de uma individualidade particular da imagem, seja forma ou objeto (WHELAN; MOLLOY, 2001).

Vindo do princípio do tema, a homogeneidade entre os objetos provocam os agrupamentos pelas características. A consequência do reconhecimento pode ser fornecida através de percentual, em consideração com a chance de proximidade, ou retornando informações pertinentes para tomada de decisão (GONZALEZ; WOODS, 2000).

Os métodos de reconhecimento podem ser diferenciados em três perspectivas: estrutural, estatística e neural. No primeiro, os padrões são representados por uma simbologia e o conjunto destes são o modelo para os métodos. O segundo método recolhe coleção de características das imagens para aplicação dos métodos. Por fim, o terceiro utiliza de redes neurais artificiais para realizar o reconhecimento (SOUZA, 2010).

Em síntese, essa é a fase que requer uma excessiva intensidade de processamento, uma vez que designa uma descrição para o objeto, além de atribuir uma interpretação dos objetos identificados (GOMES, 2009).

2.3 Reconhecimento de Padrões

O ato de reconhecer expressa o conhecer novamente e, para isso, exprime que deva ter uma informação, já armazenada, antecipada a respeito do que precisa ser reconhecido. A ação de reconhecer padrões é característico do seres humanos e dos animais sucede naturalmente, resultando na tomada de decisão (CASTRO; PRADO, 2002).

A relevância nessa área vem gradativamente crescendo devido às diversas aplicações sendo realizadas, exigindo um maior poder de processamento computacional. Partindo desse avanço, as práticas elaboradas para classificações se tornam mais simples e acessíveis em sua utilização (BIANCHI, 2006).

Para executar o reconhecimento de um padrão, carece da base de dados a ser utilizada. Por sua vez, este reconhecimento pode ser programado, desde o início, pelo desenvolvedor ou ser aprendida por técnicas de aprendizado de máquina. O reconhecimento por esse processo "envolve técnicas de atribuição de padrões às suas respectivas classes de forma automática e com a menor intervenção humana possível". Em sua grande maioria dos sistemas de *Machine Learning*¹, usufrui do reconhecimento de padrões no qual um computador captura as imagens através do sensor e fornece as informações do que é constatado (GONZALEZ; WOODS, 2000).

É importante destacar a existência de duas incumbências no reconhecimento: classificação e clusterização. A classificação compreende em especificar classes ou categorias de dados já classificados antes. No caso da clusterização, é realizado o agrupamento onde em cada clusters expõem equivalência entre si (MURTY; DEVI, 2015).

¹ Na tradução literal, aprendizado de máquina, é a habilidade em que o computador altera sua conduta tendo como base a sua experiência.

2.4 Teste Palográfico

A realização da análise da personalidade está vigorosamente existente na realidade dos profissionais da psicologia, entretanto, a variedade de procedimentos regulamentados para a população brasileira não é tão vasto, o que restringe o poder de escolha do psicólogo ([HALLBERG, 2015](#)).

A escrita é o modo de manifestação mais famigerado e a caligrafia de um ser humano com 20 anos de idade não será análoga quando o mesmo estiver com 50 anos, uma vez que as transformações coletivas, naturais e psíquicas que aconteceram nesse período será espelhadas através das suas particularidades. E para melhor entender a conduta da pessoa através da sua escrita, sucedeu a criação do teste palográfico ([VERNON et al., 1933](#)).

Logo, o teste palográfico é uma avaliação gráfica de personalidade a respeito do comportamento expressivo. A gesticulação e particularidades individuais, assim como a forma de pronunciar, caminhar, e etc, se constitui a expressão comportamental. Também complementam o significado do caráter, atitudes e os valores morais ([VERNON et al., 1933](#)).

A aplicação do teste palográfico pode ser feita de forma individual ou em grupo. Dividido em duas etapas, sendo a primeira com um briefing e treinamento para a tarefa. A segunda etapa corresponde ao teste, em descrever os palos baseado no modelo existente. O traçado deve ser executado da esquerda para a direita e de cima para baixo, de forma mais rápida e eficaz ([ALVES; ESTEVES, 2004](#)).

Esse teste foi elaborado pelo professor espanhol Salvador Escala Milá, validado com estudos no Brasil por Agostinho Minicucci, na década de 1970, e adaptado por Alves e Esteves. A última atualização do teste no Brasil aconteceu no ano de 2019. A avaliação se faz pela agregação entre medidas qualitativas e quantitativas. A aplicação do teste se dá de maneira simples e rápida, porém para a correção e a interpretação dos dados é preciso preparação e experiência do psicólogo. Considera-se no primeiro item cada movimento (palo) realizado, representando o comportamento perante a sociedade. Cada pessoa terá um estilo de resposta, diante da tarefa solicitada, ou seja, diante de uma atividade cada pessoa agirá conforme suas características pessoais, ligadas aos gestos, o ritmo, as respostas fisiológicas do organismo ([ALVES; ESTEVES, 2004](#)).

2.5 Tecnologias

2.5.1 Python

Python é uma linguagem de programação de propósito geral, ou seja, é possível ser utilizada nas mais diversas áreas de projetos, variando de aplicações web até a inteligência artificial. Interpretada, interativa e orientada a objetos, é reconhecida como uma linguagem de alto nível por ter incluído estruturas como dicionários, listas, tuplas, data e hora, além de ter uma vasta biblioteca de módulos (CORRÊA, 2020; BORGES, 2014).

Surgiu em 1989, sendo o substituto da linguagem ABC, desenvolvido pelo programador holandês Guido van Rossum. Recebe este nome não por ser baseado em cobras, mas sim numa homenagem ao grupo humorístico britânico Monty Python. Dispõe de uma sintaxe mais clara e objetiva, quando é comparado com outras linguagens, tornando-se mais produtiva (CORRÊA, 2020).

É um software considerado open-source, código aberto, compatível com a General Public License (GPL), concedendo que a linguagem seja introyida em outros produtos. A implementação oficial é escrita na linguagem C e mantida pela PSF (Python Software Foundation). Python é uma linguagem de tipagem dinâmica, em outras palavras, ocorre uma inferência por parte do interpretador no momento da execução para designar o tipo da variável, técnica conhecida como Duck Typing (BORGES, 2014).

Encontra-se, também na linguagem, outros recursos bem marcantes como geradores, persistência e unidades de testes. Por ser multiparadigma, suporta programação funcional e modular, além da POO. Seu interpretador analisa o código, realiza a conversão para símbolos, armazena em bytecode, formato binário, pela máquina virtual, e, por último, efetua a compilação do código (BORGES, 2014).

2.5.2 Django

Django é um framework abundantemente escalável e de alto nível desenvolvido em Python por Adrian Holovaty, Simon Willison e Jacob Kaplan-Moss no qual está focado em adquirir maior recurso de hardware disponível (BALDISSERA et al., 2018).

O Django se difere dos outros frameworks web por ser construído no padrão MTV (Model, Template e View) ao contrário dos outros serem MVC (Model, View e Controller) e, ademais, foi elaborado associando ao princípio DRY (Don't Repeat Yourself) com intuito de não existir replicação de código (SANTOS, 2011).

2.5.3 Dart

O Dart é uma linguagem de programação criada pelo Google no ano de 2011. É bastante diversificada na sua utilização, podendo ser tanto para aplicações mobile, desenvolvimento web, desktop e até servidor. Emprega o paradigma da orientação a objetos e segue o estilo da sintaxe de linguagens oriundas do C, como Java, Javascript e PHP.

A linguagem é fortemente tipada, entretanto o Dart consegue realizar a inferência dos tipos, dispõe de coletor de lixo e todos objetos herdam de uma classe *Object*. Essa linguagem pode ser compilada de dois modos: just-in-time (JIT) e ahead-of-time (AOT). No primeiro, o código é interpretado no momento da execução, enquanto no segundo, existe uma pré-compilação para linguagem de máquina.

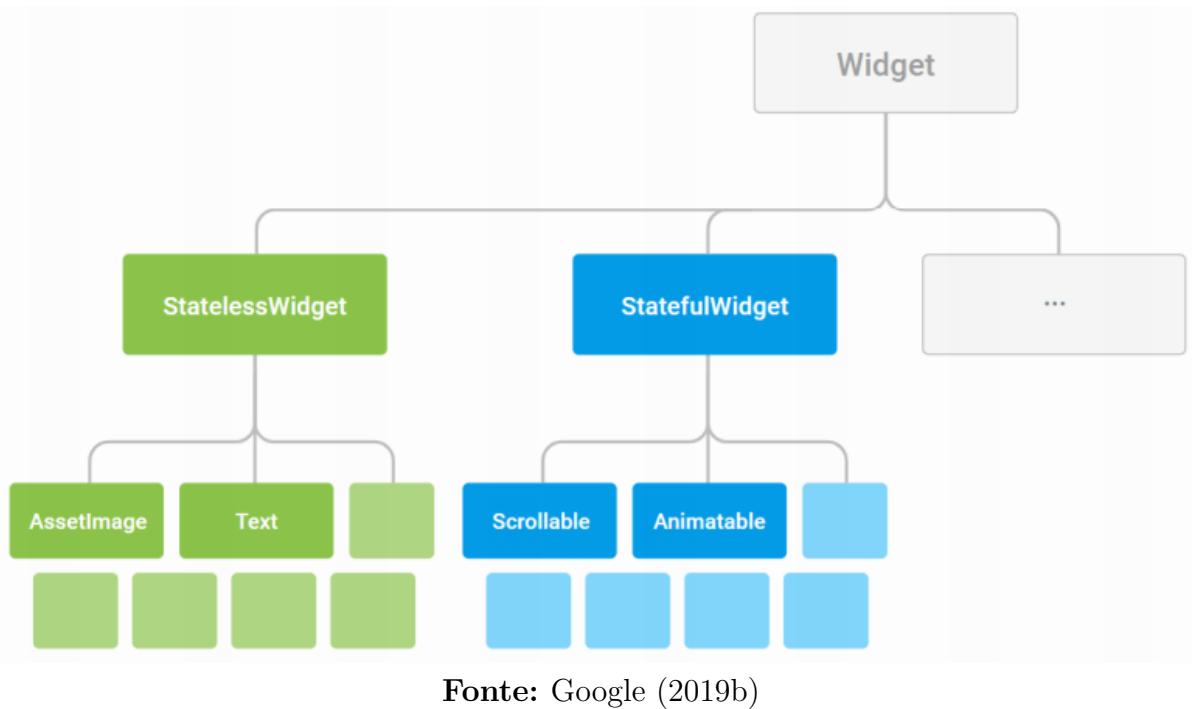
2.5.4 Flutter

Flutter é um framework criado pelo Google em 2019 para o desenvolvimento de aplicativos nativos de alta performance, usando apenas um único código na linguagem Dart. Orientado ao design, possui um aspecto em que é baseado em *widgets*, ou seja, toda interface é um *widget* ([WINDMILL, 2020](#)).

Segundo Windmill ([2020](#)), são blocos na interface da aplicação Flutter. Deste modo, existem os widgets de layouts, como espaçamentos e margens, os de estilos, como cores e fontes, os estruturais, como menus e botões, além dos específicos para cada plataforma.

Os *widgets* são estruturados de maneira hierárquica, uma vez em que os filhos herdam as características do *widget* pai, sendo assim, as propriedades citadas anteriormente são mantidas pelos *widgets* menores ([WINDMILL, 2020](#)). Na Figura 4, é demonstrado a arquitetura desta hierarquia.

Figura 4 – Hierarquia dos Widgets no Flutter



Fonte: Google (2019b)

Uma funcionalidade bastante útil para os desenvolvedores no momento de testes que está presente no framework Flutter é o Hot Reload. Esta, por sua vez, permite em que o software tenha o recarregamento rápido depois de alguma mudança na programação, sustentando a condição da última seção (WINDMILL, 2020).

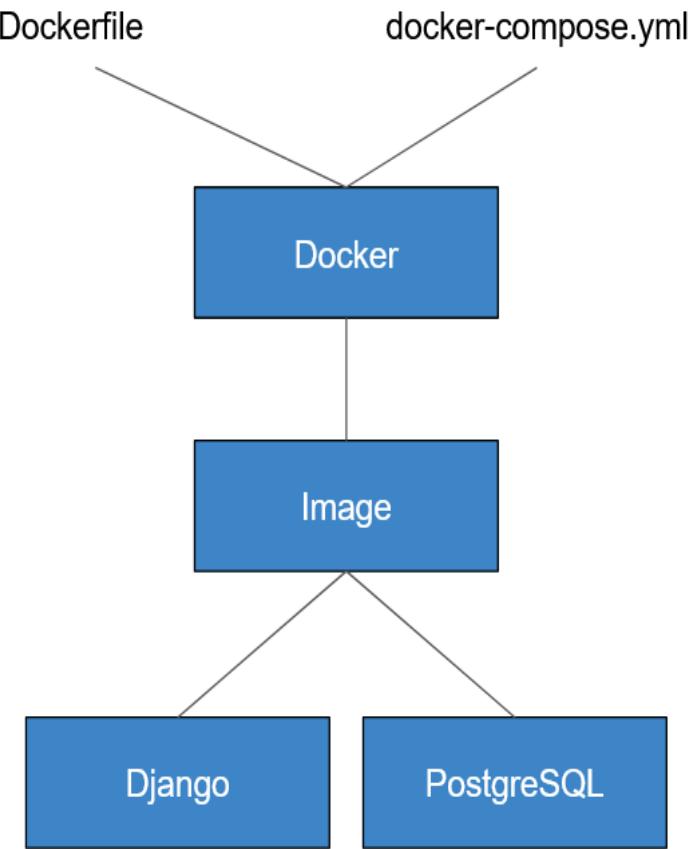
2.5.5 Docker

Docker² é uma plataforma que proporciona desenvolvimento, entrega e execução de aplicações no contexto de contêineres. Estes, por sua vez, constituem de uma técnica baseada na virtualização no nível de sistema operacional em ambiente Linux. Desta forma, essa estratégia está fundamentada no isolamento de mecanismos particulares do kernel³ (SILVA, 2016). Pelos atributos citados, o Docker é extensamente recomendado como um resolução de aplicações desenvolvidas na arquitetura de microsserviços.

² <<https://docs.docker.com>>

³ Responsável pela ligação entre o hardware e o software, gerencia a execução de aplicativos através recursos da máquina

Figura 5 – Fluxo de Interação do Docker



Fonte: Próprio Autor (2021)

Para administrar o ciclo de vida dos contêineres no sistema, existe o método de orquestração de contêiner, responsável pela organização dos componentes como alocação de recursos, gerenciamento de rede e controle de escabilidade através de ferramentas como o Docker Compose. A título de facilitar o entendimento desta tecnologia, arquitetou-se o fluxo de interação do Docker como é mostrado na Figura 5.

2.5.6 OpenCV

A OpenCV⁴ é uma biblioteca de código livre desenvolvida, inicialmente pela Intel Corporation, em C/C++ para aplicações com visão computacional. No entanto possui interfaces para o Python e o Java, tornando-a multi-plataforma, suportando os sistemas operacionais MacOS, Windows e Linux (BAGGIO, 2015).

De acordo com Marengoni e Stringhini (2009), o núcleo da biblioteca está fragmentada em cinco áreas de atuação: o processamento de imagem, a análise estrutural, a

⁴ <https://opencv.org/>

análise de movimento, reconhecimento de padrões e calibração de câmera e reconstrução 3D. Agrupando-os, totalizam mais de 500 funções desenvolvidas e disponibilizadas para o uso.

2.6 Trabalhos Relacionados

No decorrer da exploração na literatura, foram encontrados diversos trabalhos referentes à utilização da visão computacional e do processamento de imagens para reconhecimento e detecção de objetos. Estes estudos, se diferenciam desde a maneira em que ocorre a extração da imagem até a tecnologia empregada para atingir seus respectivos objetivos.

Conceição (2019) desenvolveu um trabalho para a construção de um aplicativo na correção do teste palográfico com objetivo de fornecer uma possibilidade ao software SKIP⁵ em que contempla uma pesquisa relacionada ao Histograma de Gradientes Orientados como base na resolução do trabalho. Para a codificação do sistema proposto, empregou as tecnologias Xamarin⁶ (para criação do aplicativo), PHP⁷ (para a criação dos microserviços) e Python (para detecção dos palos). Entretanto, o autor sofreu impecílios durante o desenvolvimento como: a não aprovação em tempo hábil para apresentação por parte do Conselho Federal de Psicologia e a taxa de detecção dos palos no algoritmo desenvolvido em torno de 60% comparado a correção manual de um teste palográfico. Desta forma, o projeto foi finalizado de modo parcial.

Cerqueira (2016) implementou um trabalho acerca da segmentação de placas veiculares através da tecnologia encontrada na biblioteca OpenCV com a finalidade de segregar a placa traseira do veículo, tendo em vista o aumento da necessidade do reconhecimento destas placas e poder contribuir no auxílio da diminuição de possibilidade do erro humano nesta função. O autor utilizou de dois testes para obtenção dos resultados: primeira solução com o operador *Canny* e Transformada de *Hough* e segunda solução pelos operadores morfológicos e detecção de quinas, tendo a primeira opção com a maior porcentagem de acerto para seu objetivo.

Gallo (2020) produziu um trabalho relacionado ao desenvolvimento de um sistema de reconhecimento de caracteres numéricos para calibração de trenas a laser com o propósito de suprir uma dificuldade encontrada neste fim, uma vez em que o processo de leitura e registro é feito manualmente, através da visão computacional. Durante o desenvolvimento do trabalho, a autora utilizou a ferramenta criada pelo Google chamada *AutoML Vision*⁸,

⁵ <http://skipvetoreditora.tempsite.ws/docs/ManualSKIP.pdf>

⁶ <https://dotnet.microsoft.com/apps/xamarin>

⁷ <https://www.php.net/>

⁸ <https://cloud.google.com/vision/automl/docs>

empregando tecnologias de aprendizado de máquina supervisionado. Como resultado, o sistema desenvolvido fundamentado no processamento de imagens, trouxe uma redução em 50% no tempo de calibração.

3 Metodologia

Este capítulo aborda de modo detalhado e linear a metodologia aplicada na elaboração deste trabalho. Inicialmente, são descritas as definições de pesquisas que abrangem o estudo, em seguida a explicação do desenvolvimento, a definição das tecnologias empregadas, as avaliações de desempenho, os materiais utilizados e, por fim, seção de orçamento atribuída para esta ferramenta.

3.1 Caracterização da Metodologia

A princípio, é essencial compreender o significado da palavra metodologia. No estudo da gramática denominado de etimologia, *methodos* significa organização, e *logos*, pesquisa. Segundo Fonseca (2002), metodologia é a aprendizagem de todo processo da realização de um estudo ou pesquisa. Se faz necessário destacar a diferença entre metodologia e métodos. De acordo com Markoni e Lakatos (2010), método é o agrupamento de práticas padronizadas com a finalidade de atingir o objetivo, construindo a trajetória a ser adotada, reconhecendo falhas e contribuindo para a tomada de decisão. Em outras palavras, a metodologia está voltada para a veracidade da direção escolhida com o propósito de atingir a finalidade da pesquisa. Os métodos estão relacionados ao conteúdo, práticas e procedimentos, a serem empregados na metodologia.

Este estudo tem a finalidade metodológica aplicada, de abordagem qualitativa e o objetivo exploratório, além do procedimento documental e experimental. Abaixo, é descrito cada um dos itens aqui apresentados:

- Pesquisa Aplicada: uma finalidade metodológica que envolve a prática na resolução do problema apresentado pelo pesquisador. De acordo com Gil (2019), as pesquisas aplicadas podem contribuir para a ampliação do conhecimento científico e sugerir novas questões a serem investigadas. Assim, esse tipo de pesquisa objetiva expandir a compreensão do estudo em específico ao buscar solução quase imediata. A pesquisa aplicada, conforme Assis (2009), está interessada na aplicação, uso e resultados práticos do conhecimento, e visa aplicar o conhecimento científico na solução de diversos problemas individuais ou coletivos, desenvolvidos por meio das ciências aplicadas e tecnológicas.
- Abordagem Qualitativa: apresentará um aprofundamento de questões centradas nas relações sociais, tendo em vista a busca de compreensão dos porquês, não se utilizando de procedimentos estatísticos nos dados pesquisados. Conforme Minayo (2001), a

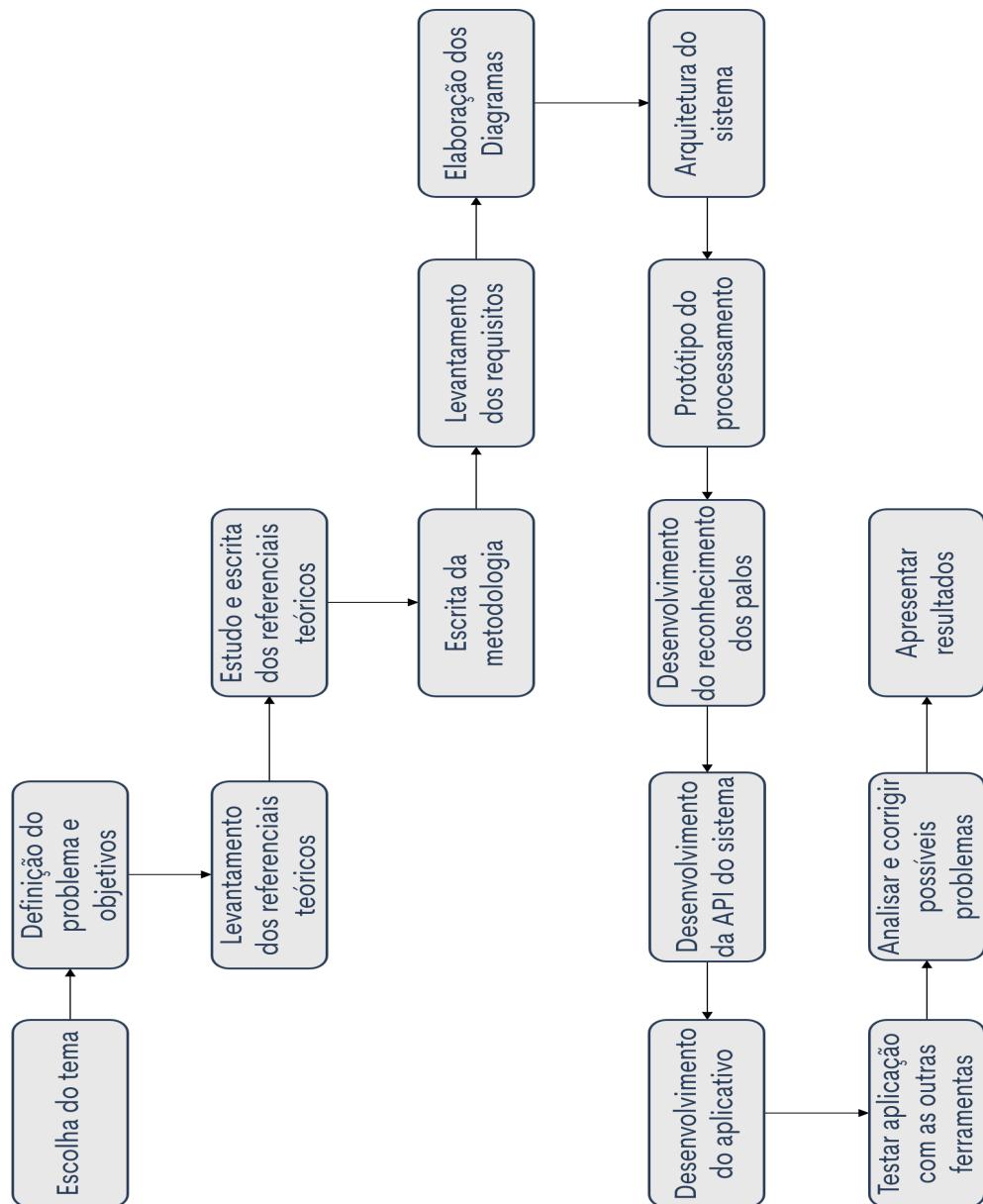
pesquisa qualitativa trabalha com o universo de significados, motivos, aspirações, crenças, valores e atitudes, o que corresponde a um espaço mais profundo das relações, dos processos e dos fenômenos que não podem ser reduzidos à operacionalização de variáveis. Portanto, suas características incluem alguns aspectos, como a busca de resultados autênticos, respeito aos dados empíricos, objetivação do fenômeno, observância das diferenças entre o mundo social e o mundo natural e a hierarquização das ações de compreender, explicar, precisar as relações entre o global e o local em determinado fenômeno ([GERHARDT, 2009](#)).

- Objetivo Exploratório: realiza uma abordagem mais próxima do fato ou fenômeno, com o intuito de propor hipóteses por intermédio de análise bibliográfica, entrevistas, pesquisa de campo e aplicação de questionário. Conforme Gil ([2019](#)) seu planejamento costuma ser muito flexível, pois é interessante considerar os mais diversos aspectos relacionados ao fato ou fenômeno em estudo. Tal tipo de pesquisa apresenta, portanto, necessidade de coleta básica de dados por parte do pesquisador, busca de experiências práticas com o problema pesquisado e análise de exemplos que promovam a compreensão.
- Procedimento Experimental: caracterizada pela manipulação das variáveis, a fim de analisar o impacto que uma possui sobre a outra após o pesquisador ter definido seu objeto de estudo. Dessa forma, torna-se possível determinar as causas e os efeitos do fenômeno estudado. As características necessárias nesse tipo de pesquisa incluem, além da manipulação, a designação de um grupo de controle e a distribuição aleatória de elementos. Ainda, de acordo com Fonseca ([2002](#)), a pesquisa experimental pode ser dividida em duas classes: experimentação em laboratórios, no qual o meio ambiente criado é artificial, e experimentação no campo, que consiste na criação das condições de manipulação dos sujeitos nas próprias organizações.
- Procedimento Documental: um procedimento de coleta de informações e dados que ainda não foram analisados cientificamente. Tal pesquisa pode ter sua fonte obtida na forma escrita ou não escrita, por meio de documentação impressa, eletrônica, gráfica e audiovisual. Conforme Fonseca ([2002](#)), a pesquisa documental recorre a fontes mais diversificadas, tais como: tabelas estatísticas, jornais, revistas, relatórios e documentos oficiais.

3.2 Implementação dos Métodos

Com o intuito de ter uma melhor compreensão sobre a metodologia deste projeto, foi elaborado um fluxograma, como mostra na Figura [6](#) logo abaixo.

Figura 6 – Fluxograma da Metodologia



Fonte: Próprio Autor (2021)

Portanto, este presente projeto está dividido em 5 etapas, são elas:

1. Problema.

No primeiro momento se fez necessário definir o tema, justificativa, público-alvo, problema a ser resolvido e os objetivos a fim de solucionar essas questões.

2. Referencial Teórico.

Posteriormente, na segunda etapa, ocorre o levantamento do referencial teórico, explanando as definições sobre imagem digital, visão computacional, reconhecimento de padrões e sobre o teste palográfico.

3. Estruturação do Projeto.

Na terceira etapa é a fase da elaboração, sendo estruturada a metodologia, o levantamento de requisitos, definindo quais funcionalidades estarão na aplicação, e o diagrama de atividade, indicando o fluxo do funcionamento da ferramenta.

O processo de software é "um conjunto de atividades relacionadas que levam à produção de um produto de software". Este processo é dividido em quatro períodos: especificação, desenvolvimento, validação e evolução ([SOMMERVILLE, 2011](#)).

De acordo com Carvalho ([2002](#)), nessa etapa, a engenharia de requisitos encontrada no período de especificação, é responsável pela parte indispensável no ciclo de vida do software, uma vez em que o entendimento íntegro dos requisitos é substancial para um desenvolvimento com êxito. Os requisitos para uma aplicação é a composição de informações das funcionalidades, provimentos e restrições do sistema. A criação destes requisitos é baseado nas necessidades para cumprir os objetivos do software ([SOMMERVILLE, 2011](#)).

Conforme Sommerville ([2011](#)), esses requisitos são comumente distribuído em duas categorias: requisitos funcionais (RF) e não-funcionais (RNF). O primeiro reproduz as funcionalidades que o sistema possui, ou seja, as capacidades acessíveis para os usuários. Os RFs também podem especificar os estímulos e respostas, sendo assim, o comportamento do software. O segundo apresenta as restrições destas funcionalidades ofertadas pelo sistema. Os RNFs associam com os padrões de qualidade pois estabeleem relação de eficiência com as utilidades da aplicação ([XAVIER, 2009](#)). Erros cometidos no momento da elicitação destes requisitos podem ser laboriosos para correção, visto que a aplicação estiver sido desenvolvida.

A *Unified Modeling Language* (UML) é a linguagem mais habitual para modelagem de software baseado na orientação a objetos. Auxilia na construção, documentação, especificação e visualização dos elementos no sistema, além de ser apropriada não obstante da dificuldade do objetivo da aplicação. Para melhor entendimento utiliza-se de diagramas desenvolvidos usando a UML, através de componentes, dependências, blocos, interfaces, associações e generalizações ([RUMBAUGH; BOOCHE; JACOBSON, 2000](#)).

Para realização destes diagramas nesse trabalho, foi produzido um estudo a cerca das ferramentas de elaboração dos diagramas e conclui-se na escolha de dois softwares: Astah UML e PlantUML. O primeiro foi decidido por ser o mais utilizado pela comunidade, adquiriu-se mais funcionalidades da versão paga, por meio do email institucional acadêmico oferecido pela UNITINS, e neste trabalho, direcionado para criação do fluxograma. O segundo, foi escolhido por ser um software livre e ter uma grande similaridade com a programação tendo sintaxes bem intuitivas para

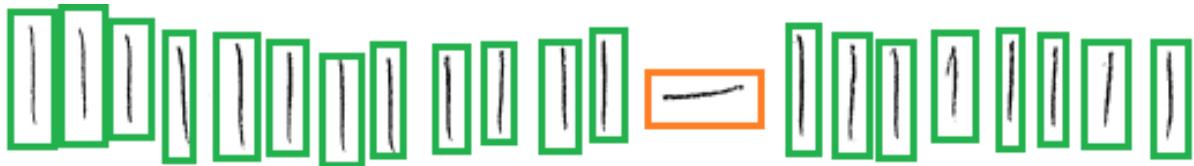
a modelagem dos diagramas. A seguir, é detalhado os diagramas utilizados neste trabalho:

- Diagrama de Atividade: classificado como um diagrama comportamental, tem por finalidade de modelar a lógica do processo de negócio, a sequência estrutural das ações, além de descrever a correlação com as outras atividades ([BEZERRA, 2007](#)). Farias ([2009](#)) afirma ser semelhante a um fluxograma, pois expressa a sequência das atividades e caracteriza o nível macro dos processos que constituem as atividades. Também pode ser adicionado os encarregados pelo andamento da atividade estipulada.
- Diagrama de Caso de Uso: caracterizado como um modelo para identificar os requisitos de um software, determinando seu comportamento entre os atores (usuários), além de auxiliar a delimitação do sistema. Conforme Misbhauddin e Alshayeb ([2015](#)), este diagrama é constituído por quatro componentes: sistema, atores que se comunicam com o sistema, as funcionalidades (requisitos) presentes no sistema e os relacionamentos entre os componentes.
- Diagrama de Classe: considerado o mais utilizado para os sistemas orientados a objetos, o diagrama de classe se enquadra como alicerce para os demais diagramas. Tem o propósito principal de visualização das classes que consistem o software, estabelecendo a organização nas suas estruturas ([GUEDES, 2018](#)). De acordo com Fowler ([2004](#)), esse diagrama declara os tipos de objetos e seus respectivos relacionamentos, além de indicar operações e as propriedades existentes em cada classe e outros recursos, como cardinalidades.
- Diagrama de Entidade Relacionamento: definido como um modelo de fluxograma demonstrando, através de representação gráfica, o banco de dados relacional exibindo cada relacionamento existente entre as tabelas. Este diagrama da engenharia de software tem por objetivo de elucidar como os objetos, conceitos e entidades se comunicam dentro de uma aplicação ([NOGUEIRA, 1988](#)).

4. Protótipo e Desenvolvimento.

Nesta etapa determinaram-se as linguagens de programação e suas respectivas bibliotecas para estabelecer a arquitetura do trabalho. Estabeleceu-se o Python como linguagem por ser a mais frequentemente utilizada para o alvo deste trabalho, reconhecimento de imagens, com a sua numerosa gama de biblioteca que auxiliam este propósito. Para a escolha do framework, que torna a arquitetura da aplicação sendo orientada a serviços, foi designado o Django pelo fato de ser altamente customizável, escalável e seguro, tornando-o extremamente confiável.

Figura 7 – Protótipo de Detecção



Fonte: Próprio Autor (2021)

Após isso, é a criação dos protótipos, no qual simula como será o resultado da aplicação após ser desenvolvida. Na Figura 7, logo acima, representa o protótipo de uma detecção e reconhecimento no teste palográfico. O retângulo destacado na cor verde identifica a localização de um palo, enquanto o retângulo na cor laranja indica um intervalo encontrado.

Continuando nesta etapa, iniciou-se a programação da detecção e reconhecimento dos palos, API do sistema e o aplicativo com interface gráfica para os usuários finais da ferramenta. A escolha do framework Flutter, escrito na linguagem Dart, se dá, principalmente, pela agilidade na produtividade do desenvolvimento para plataformas Android¹ e iOS² com apenas um único código. Outros fatores preponderantes, como a maior compatibilidade entre as versões dos sistemas operacionais dos dispositivos móveis e ter o suporte do Google como empresa mantenedora, contribuíram para esta seleção.

5. Testes e Resultados.

Na quinta e última etapa, está a validação de todo sistema por meio da composição de um estudo de caso. Originou-se através do treinamento do reconhecimento com uma base de testes palográficos, disponibilizados por uma psicóloga registrada e ativa pelo Conselho Regional de Psicologia do Tocantins, como forma de instrumento de coleta dos dados, onde submeteu, posteriormente, a correções, visando melhorias e aperfeiçoamento com o propósito de finalizar o projeto.

É importante destacar que o teste palográfico, assim como outros testes psicológicos, são de uso privativo dos profissionais psicólogos e por esse motivo, a psicóloga estará acompanhando os resultados, resguardando os sigilos necessários.

3.3 Definição das Tecnologias

Nesta seção serão retratadas as justificativas nas escolhas das tecnologias empregadas nesse trabalho.

¹ https://www.android.com/intl/pt-BR_br/

² <https://www.apple.com/br/ios/>

3.3.1 Linguagem do Algoritmo do Processamento de Imagem

Com a finalidade de estabelecer a linguagem para codificação do processamento de imagem, produziu uma lista com as mais predominantes empregadas para este fim e obteve o seguinte resultado: Matlab³, Java⁴, C/C++⁵ e Python.

O Matlab possui um número expressivo de usuários em prol deste tema, devido ao seu sistema ser uma matrix e uma imagem se constituir de mesmo modo, entretanto, é uma tecnologia paga. O Java dispõe de um acervo numeroso de bibliotecas, comumente utilizado para o processamento de imagem do mesmo modo para criação de redes neurais e processamento de linguagem natural, porém consome bastante memória, tornando o processo prolongado (CODRUT, 2015).

O C/C++ contém bibliotecas nativas, conforme o próprio OpenCV, além de ser considerada a linguagem de programação mais rápida, contudo a mesma possui uma sintaxe mais verbosa bem como constituindo de uma curva de aprendizagem superior.

Por último, o Python foi desenvolvido com o intuito de tornar a programação bem mais simples, isto é, menos verbosa, no qual permite a vantagem de velocidade na execução através da linguagem base e reduzir tempo de escrita na codificação. Além destes benefícios citados, o Python é uma linguagem gratuita e de código-aberto, possuindo uma enorme comunidade de usuários, e contempla uma quantidade numerosa de bibliotecas voltadas para a visão computacional, incluindo o suporte nativo ao OpenCV (biblioteca usufruída neste trabalho). Portanto, destinou-se na escolha do Python para o desenvolvimento do algoritmo de processamento da imagem.

3.3.2 Framework Python

Aproveitando a seleção do Python para o processamento de imagem e de todos os seus proveitos, optou por permanecer no mesmo ambiente de codificação para o desenvolvimento da API.

Na preferência do framework da linguagem Python, necessitou reconhecer as características do Flask⁶ e do Django, sendo os mais utilizados pela comunidade. Inicialmente, o Flask é afamado por ser discernido como um micro-framework pela simplicidade e velocidade de desenvolvimento em relação ao seu adversário, entretanto sua capacidade é limitada carecendo de bibliotecas suplementares.

³ <https://www.mathworks.com/products/matlab.html>

⁴ <https://www.oracle.com/br/java/>

⁵ <https://visualstudio.microsoft.com/pt-br/vs/features/cplusplus/>

⁶ <https://flask.palletsprojects.com/en/2.0.x/>

No caso do Django, o framework provê seu próprio ORM⁷ integrado, ao contrário do Flask no qual carece de bibliotecas auxiliares para tal funcionalidade, aplicando através de seus *models*, permitindo relacionar o modelo de uma classe com a sua tabela no banco de dados. Portanto, decidiu-se na escolha do Django pelo contexto do trabalho uma vez em que se faz necessário o relacionamento das classes modeladas no seu respectivo diagrama para com o armazenamento.

Além disso, o Django possui indicador de alta performance permitindo uma aplicação escalável, veloz e segura. O framework também oferece um painel administrativo da aplicação com uma interface intuitiva, facilitando a manipulação dos modelos de acordo com as demandas existentes. Tendo em vista a estruturação do trabalho, o Django possibilitou a criação de aplicativos, de modo que se assemelha ao conceito de componentes, dentro do projeto nos quais foram utilizados para cada modelo elaborado, mantendo uma boa prática de desenvolvimento.

3.3.3 Framework Desenvolvimento Híbrido

A respeito do desenvolvimento para dispositivos móveis, optou-se para um framework híbrido em que abrange os dois sistemas operacionais mais utilizados atualmente, Android e iOS, pela maior praticidade de solucionar o objetivo deste trabalho. Para tal questão, listou-se o React Native e Flutter.

O React Native, criado por funcionários do Facebook, tem como base a biblioteca do Javascript chamada de React e destina-se na formação de componentes modulares em aplicações de página única, além de conseguir renderizar o aplicativo nativamente. Já o Flutter, adiante as outras vantagens do seu concorrente, possui uma extensa biblioteca de widgets e animações que auxiliam na programação assim como trabalha com o próprio mecanismo de renderização. Diante disso, a escolha se deu pelo framework criado pela Google, o Flutter ([FENTAW, 2020](#)).

3.3.4 Banco de Dados

A princípio, por estar numa aplicação envolvendo relacionamento entre as entidades existentes, decidiu-se na escolha de um banco de dados relacional. Acerca da preferência entre os modelos existentes, destacaram-se o SQLite, MySQL e PostgreSQL.

O SQLite é um banco de dados que armazena as informações apenas em único arquivo e somente poder ser acessado mediante a uma conexão por vez, não sendo recomendado para um sistema em produção. Já o MySQL, é uma ferramenta que possui uma

⁷ Object-Relational Mapping: Tecnologia de aproximação da aplicação orientada a objetos com o banco de dados relacional.

quantidade enorme de usuários entretando para obter permissão a todas funcionalidades, existe um custo monetário.

O PostgreSQL se constitui um banco de dados *opensource* contando com um conjunto de recursos avançados desde sua criação como gatilhos, funções e herança, além de oferecer um amplo suporte para a linguagem Python, especialmente com o Django, utilizada neste trabalho como tipos de dados que apenas atua neste banco e o framework prover a biblioteca *django.contrib.postgres* para realizar as operações. Portanto, por essas razões, se deu pela preferência do PostgreSQL.

3.4 Avaliação de Desempenho

De acordo com Pontes (1999), a avaliação de desempenho é caracterizada segundo um procedimento de conferência baseado no resultado factual e no resultado conjecturado. Para tal, esta seção compõe a respeito do indicador de eficiência no processamento da imagem e na qualidade da imagem existentes no presente trabalho.

3.4.1 Métrica de Eficiência do Processamento de Imagem

A análise da métrica de eficiência do algoritmo desenvolvido para o processamento da imagem se condeceu através da construção de um comparativo dos resultados gerados de cada atributo presente no diagnóstico do teste palográfico desta aplicação proposta neste trabalho com a ferramenta SKIP e a correção manual realizada por um psicólogo. Visto que estes dados são produzidos, torna-se possível a elaboração de gráficos com a finalidade de uma maior elucidação da solução encontrada.

3.4.2 Métrica de Qualidade da Imagem

A verificação da métrica de qualidade no arquivo enviado pelo usuário se exprimiu por meio da limitação na resolução contida na imagem, permitindo uma análise do objeto proposto nesta ferramenta a partir da resolução conhecida como HD (*High Definition*) disposta de 1280 pixels x 720 pixels. Caso o usuário não atender esta condição, um alerta será enviado para o aplicativo informando-o para efetuar uma nova postagem da imagem com uma resolução melhor.

3.5 Materiais

No que se refere ao editor de texto para o desenvolvimento da aplicação, foi designado o *Visual Studio Code*⁸. Criado pela *Microsoft*⁹, este software permite a programação nas duas linguagens escolhidas para codificação da ferramenta proposta neste projeto.

Quadro 1 – Hardware

Descrição	Fabricante	Modelo/Versão
Sistema Operacional	Linux	Ubuntu 20.04 64bits
Sistema Operacional	Microsoft	Windows 10 Pro 64bits
Processador	AMD	Ryzen 7 3700X 3.6GHz
Memória RAM	Lexar/Husky	32GB 2666MHz
Armazenamento	WD	SSD 500GB M2 NVMe
Placa de Vídeo	PCyes	NVIDIA Geforce GTX 1660 OC
Placa-mãe	Asus	B450M Gaming/BR

Fonte: Próprio Autor (2021)

Desta forma, pela praticidade de ter o código de toda aplicação no mesmo ambiente e possuir licença gratuita, harmonizou a escolha deste editor. Para o desenvolvimento do objetivo deste projeto, foi utilizado do seguinte hardware como é demonstrado no Quadro 1 acima.

3.6 Orçamento

Considerando a elaboração, arquitetura e desenvolvimento, este presente projeto, em sua íntegra, apresenta nenhum tipo de custeio ou gasto com a finalidade de atingir o resultado.

⁸ <https://code.visualstudio.com/>

⁹ <https://www.microsoft.com/pt-br>

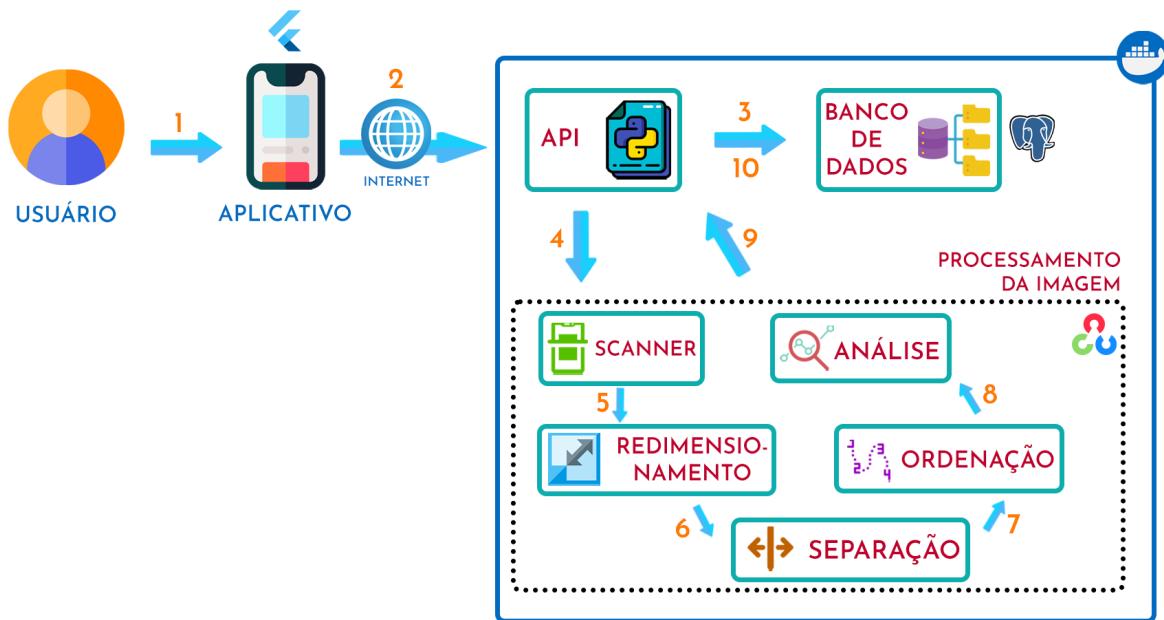
4 Resultados

Neste capítulo está apresentado os resultados obtidos durante a elaboração deste trabalho, como a elaboração de artefatos de software, descrição da arquitetura formulada e implementação da ferramenta, e por fim, as análises obtidas através dos testes realizados.

4.1 Arquitetura da Aplicação

Nesta seção, posteriormente a fase de análise das tecnologias, estabeleceu-se a arquitetura do trabalho na qual é possível visualizar os elementos integrados formando a aplicação desenvolvida e, na Figura 8 abaixo, demonstra estes componentes.

Figura 8 – Arquitetura da Ferramenta



Fonte: Próprio Autor (2021)

Tendo em vista a arquitetura desenvolvida, a seguir ocorre o detalhamento de cada passo representado na Figura 8.

1. Inicialmente, o usuário terá de efetuar uma entrada na ferramenta por meio do aplicativo desenvolvido, no qual terá ao seu alcance as funcionalidades requisitadas.
2. Com as informações recebidas do usuário, o aplicativo encaminha estes dados através de uma requisição, por meio do protocolo HTTP, para a API desenvolvida.

3. A API desenvolvida em Python por intermédio do framework Django, recebe a requisição realizada e executa a inserção das informações no banco de dados PostgreSQL.
4. Caso o usuário esteja desempenhando um cadastro de teste palográfico, inicia-se a etapa de processamento da imagem com auxílio da biblioteca OpenCV. O arquivo enviado parte para etapa de scanner.
5. É realizado um redimensionamento na imagem com a finalidade de permanecer um padrão de proporção nos arquivos recebidos.
6. Na fase seguinte, se faz uma segregação de acordo com a região de interesse do teste, removendo os blocos desnecessários utilizados na análise.
7. A etapa de ordenação consiste em destacar os palos presentes na imagem, baseado nos contornos identificados, e ordená-los na sequência.
8. No último método do processamento, ocorre a análise dos resultados presentes do teste palográfico gerada com base no algoritmo de reconhecimento de padrões.
- 9–10. Estes resultados alcançados são enviados para a API com o intuito de que seja executada a inclusão desta análise no banco de dados.

A API, o banco de dados e o algoritmo de processamento da imagem estão agrupados como ambientes isolados, conhecidos por *containers*, mediante auxílio do Docker. A explicação integral do processamento, e de todos os seus estágios, está contida na subseção 4.2.2.2 deste capítulo.

4.2 Implementação

Nesta seção estão presentes a engenharia de software, contendo os diagramas elaborados e requisitos elicitados, bem como o desenvolvimento da ferramenta, abordando a programação da API, do aplicativo e do algoritmo do processamento de imagem.

4.2.1 Engenharia de Software

4.2.1.1 Levantamento de Requisitos

Para identificar as capacidades e distinguir os limites da ferramenta, construiu-se o documento de elicitação dos requisitos. Desta forma, os detalhes de cada funcionalidade da aplicação estão mais explicativos a fim de que não tenham nenhuma incoerência ou incompatibilidade. Este levantamento foi dividido em duas frentes: requisitos funcionais e não funcionais.

4.2.1.1.1 Requisitos Funcionais

Na sua totalidade, foram levantados oito requisitos funcionais. Estes, por sua vez, estão especificados em forma de tabelas por: identificação, nome, referência, definição, sumário, pré-condições, atores, descrição, alternativas e exceção. Todos os requisitos estão apresentados no Apêndice A.

4.2.1.1.2 Requisitos Não Funcionais

Constituíram-se seis requisitos não funcionais durante o processo de elicitação. Os RNFs foram organizados por identificação e descrição. Logo abaixo, estão exibidos estes requisitos.

Quadro 2 – Requisitos Não Funcionais

ID	Descrição
RNF001	A API deve ser desenvolvida em Python.
RNF002	A API deve utilizar o framework Django.
RNF003	A API deve utilizar o formato JSON para a troca de informações.
RNF004	A API deve utilizar a autenticação JWT.
RNF005	O aplicativo deve ser desenvolvido em Dart.
RNF006	O aplicativo deve utilizar o framework Flutter.
RNF007	A aplicação deve utilizar o banco de dados PostgreSQL.

Fonte: Próprio Autor (2021)

Nota-se que estes requisitos não funcionais englobam as categorias de tecnologia empregada, de segurança nas requisições e de padrões de comunicação entre os componentes na aplicação.

4.2.1.2 Diagramas

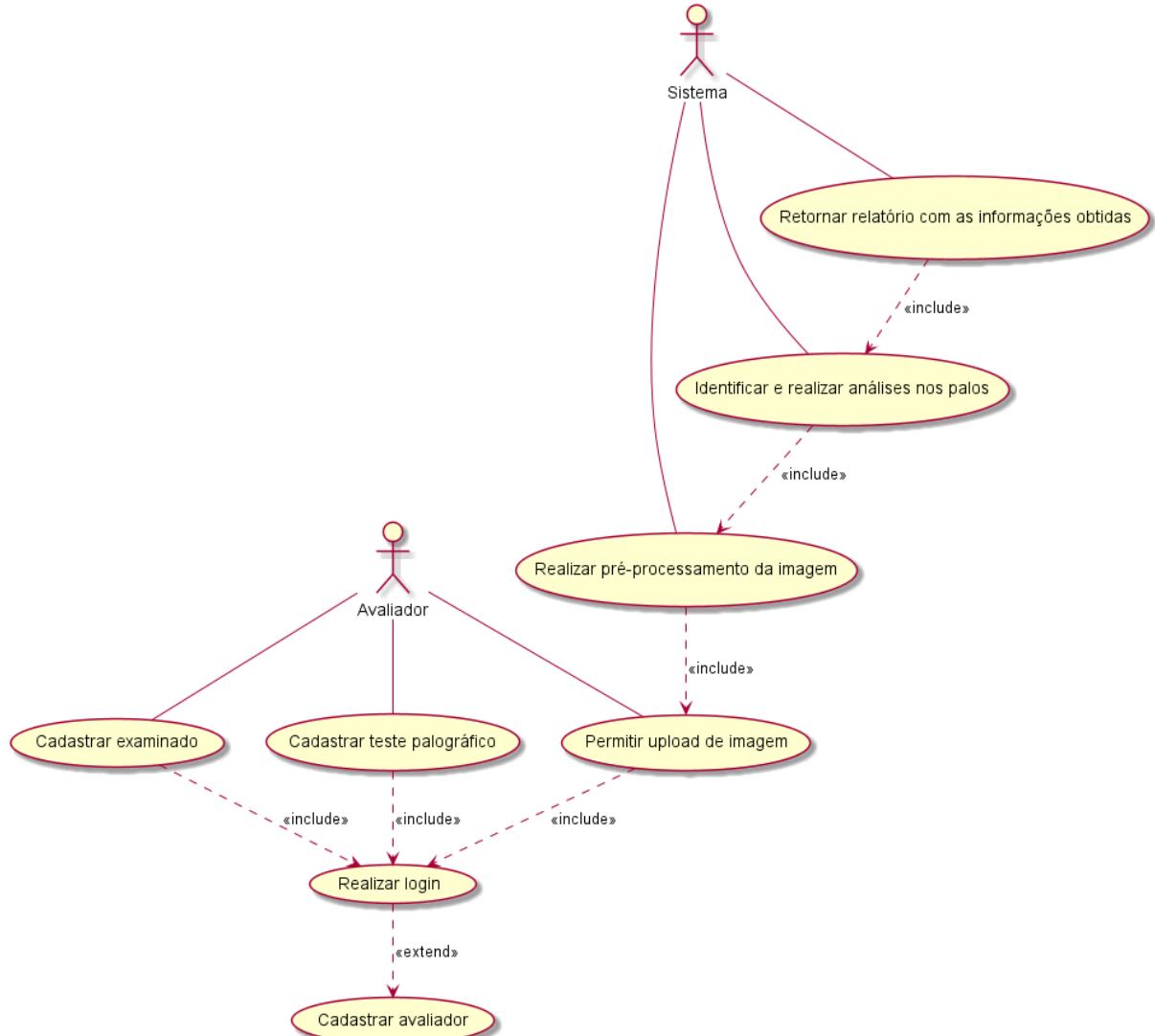
A título de construir uma documentação para melhor entendimento no momento da programação e sobre seu funcionamento, se fez necessário realizar a modelagem dos diagramas. Esta seção dispõe dos seguintes diagramas: casos de uso, atividade, entidade relacionamento e de classes.

4.2.1.2.1 Diagrama de Caso de Uso

Ao decorrer da modelagem do diagrama de caso de uso, foram constatados dois atores presentes na ferramenta: avaliador e sistema. O detalhamento de cada de funcionali-

dade está descrito na subseção 4.2.1.1.1 sobre os requisitos funcionais. A seguir, na Figura 9, é demonstrado a construção deste diagrama.

Figura 9 – Diagrama de Caso de Uso



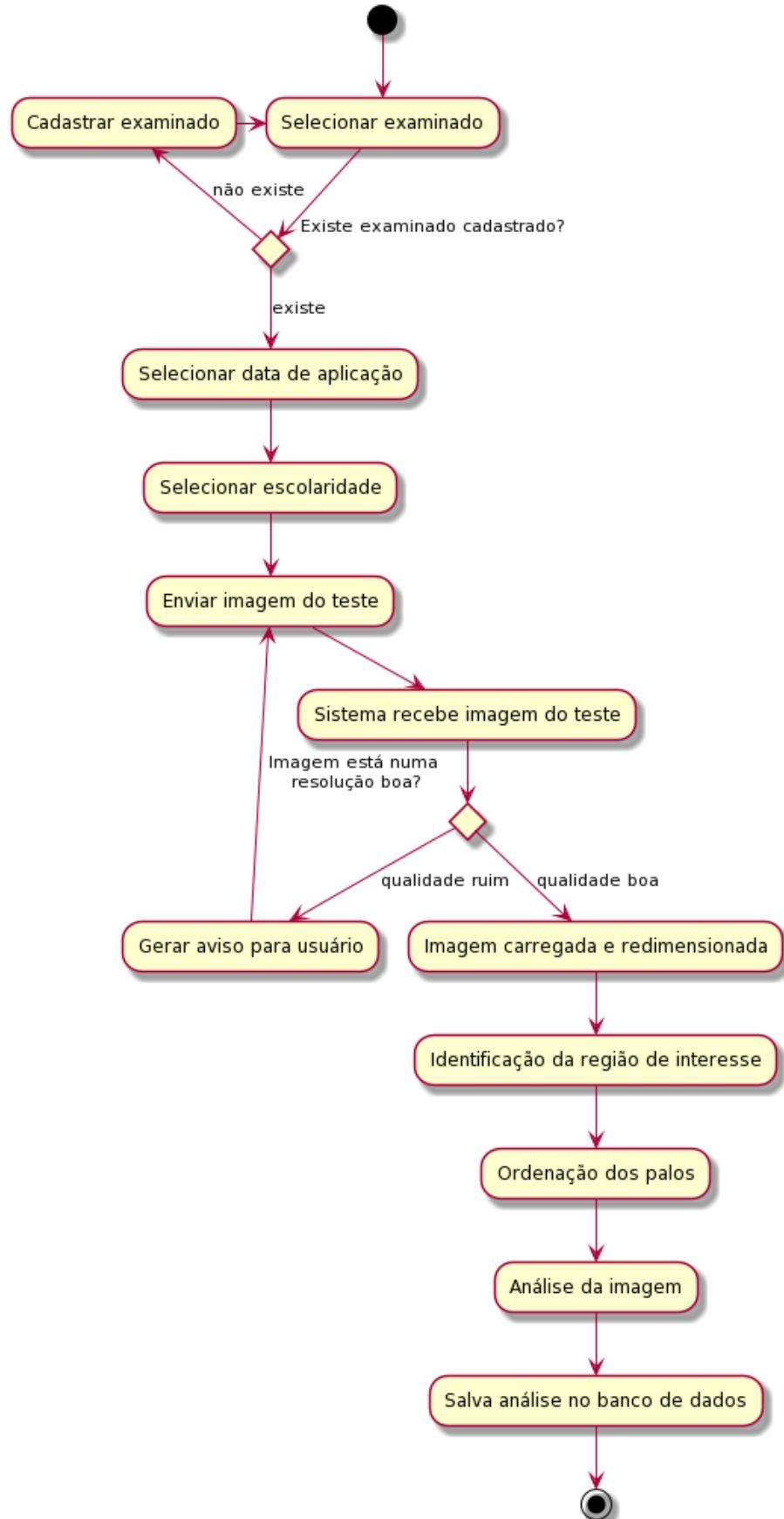
No diagrama acima, demonstra as funcionalidades existentes separadas por cada ator, sendo cinco requisitos para o avaliador e três requisitos para o sistema, e exemplifica a sequência de cada etapa que decorre até ao término do processo.

4.2.1.2.2 Diagrama de Atividade

Com a intenção de um melhor esclarecimento sobre a funcionalidade mais complexa da ferramenta na visão do ator Avaliador, definido pelo diagrama de caso de uso e no seu respectivo requisito funcional, orientou-se o desenvolvimento do diagrama de

atividade referente ao cadastro de teste palográfico. Na Figura 10, logo abaixo, ilustra o seu funcionamento.

Figura 10 – Diagrama de Atividade



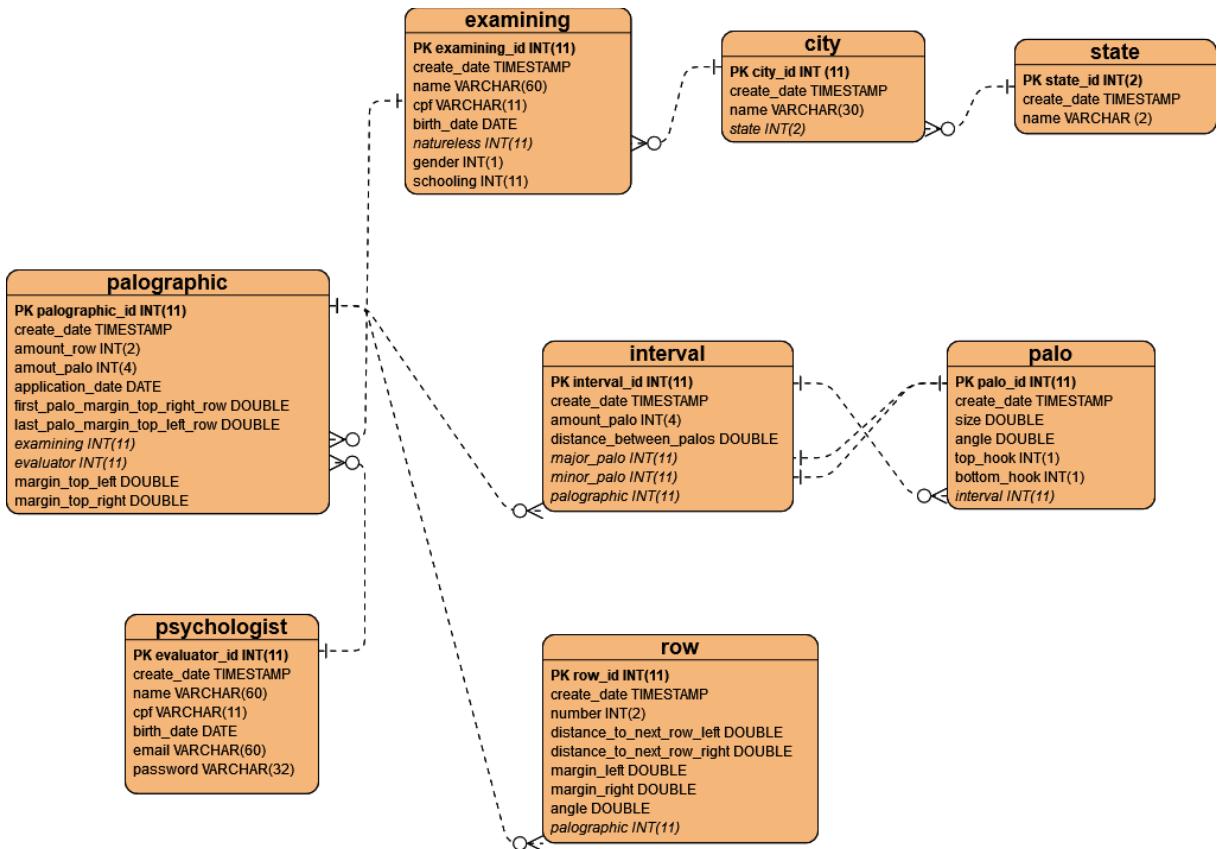
Fonte: Próprio Autor (2021)

Neste diagrama é possível observar os processos existentes executados pelo processamento de imagem a partir do momento em que o algoritmo recebe o arquivo, realiza análise e salva as informações no banco de dados. A descrição especificada de toda essa etapa está contida na subseção 4.2.2.2.

4.2.1.2.3 Diagrama de Entidade Relacionamento

A título de melhor entendimento para o desenvolvimento, foi elaborado o diagrama de entidade relacionamento, representado na Figura 11 abaixo. O diagrama consiste em oito tabelas para suprir a concepção estruturada no diagrama de classe presente na Figura 12.

Figura 11 – Diagrama de Entidade Relacionamento



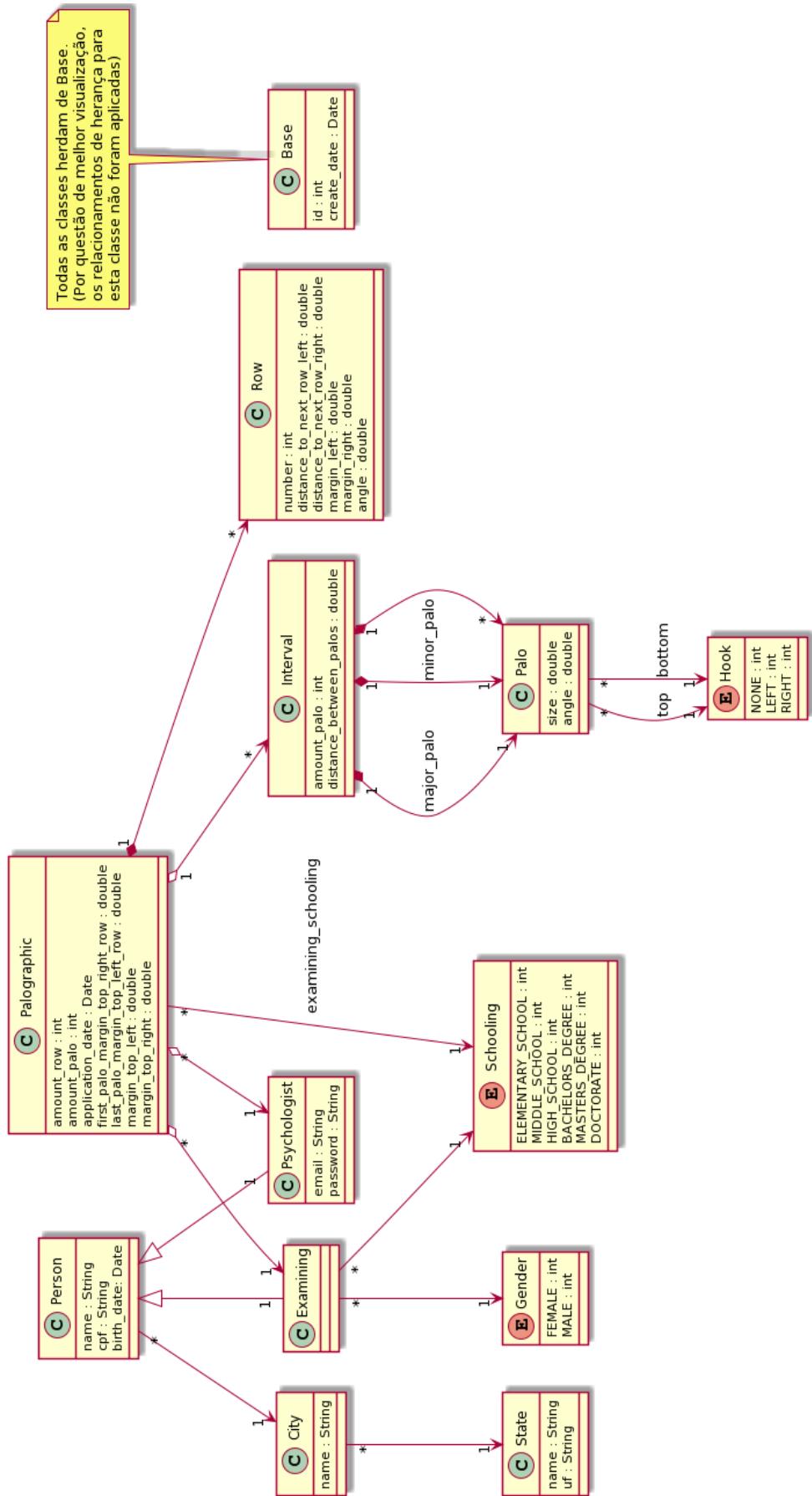
Fonte: Próprio Autor (2021)

Em comparativo com o diagrama de classe, os enumeradores foram extraídos e convertidos para atributos de suas respectivas tabelas com base nos seus relacionamentos.

4.2.1.2.4 Diagrama de Classe

Ao todo, destinou-se a modelagem de dez classes e três enumeradores. A fim de definição, um *enum* é caracterizado por um grupo fixo de valores convertendo para um código menos suscetível a erros e mais legível. A seguir, contém explicação de todos os componentes deste diagrama.

Figura 12 – Diagrama de Classes



Fonte: Próprio Autor (2021)

- *Base*: classe pai que herda para todas as outras classes do diagrama. Nela comporta os atributos para identificação e conhecimento da data de criação.
- *Person*: classe relacionada a pessoa na aplicação. Esta, contém os atributos de nome, cpf e data de nascimento (desta forma, a aplicação conseguirá identificar a idade da pessoa).
- *City* e *State*: classes relacionadas a localidade da pessoa (psicólogo e examinado) na aplicação. Classe pessoa possui a cidade, que, por sua vez, possui estado.
- *Gender*: enumerador (*enum*) relacionado ao sexo do examinado, podendo ser feminino ou masculino.
- *Schooling*: enumerador (*enum*) relacionado a escolaridade do examinado, conferindo desde o nível fundamental até o doutorado.
- *Psychologist*: classe relacionada ao avaliador do teste palográfico, ou seja, o psicólogo. Esta classe se associa com a classe pessoa, além de conter os atributos de email e senha, utilizados para realizar a verificação no login da aplicação.
- *Examining*: classe relacionada ao examinado do teste palográfico. É designado, através da associação com a classe pessoa, todos os respectivos atributos de pessoa, além dos enumeradores de gênero e escolaridade.
- *Palographic*: classe relacionada ao teste Palográfico. Nela abrange a data da aplicação do teste e atributos referentes a quantidade e distância que terão por objetivo armazenar informações imprescindíveis para uma correção precisa do psicólogo. Possui agregações com as classes examinado, psicólogo e intervalo, e também associação com a escolaridade, uma vez em que o examinado pode realizar o exame diversas vezes e esse atributo sofrer alterações para cada teste realizado, e região, pois existem valores/médias de resultados conforme o local descrito.
- *Row*: classe relacionada a cada linha do teste palográfico. Contém os atributos de número (posição), distância até a próxima linha do lado esquerdo, distância das linhas, das margens e dos ângulos. Estes atributos compreendem as análises de comprimento e de afastamento da margem da folha em que a ferramenta irá efetuar. Possui um relacionamento de composição com a classe palográfico.
- *Interval*: classe relacionada aos intervalos do teste palográfico. Possui os atributos de quantidade de palos presentes neste intervalo, distância entre o primeiro palo e o último palo, maior e menor palo deste intervalo, além da lista de palos presentes (sendo composições da classe palo). Estes intervalos representam um sinal determinado pelo avaliador no momento da execução do teste e que os tornam objetos de análise por parte do psicólogo.

- *Hook*: enum relacionado a posição do gancho de cada palo. Compreende os valores de: nenhum, esquerdo ou direto. Estes ganchos simbolizam a existência de uma curvatura mais acentuada em cada palo detectado.
- *Palo*: classe relacionada a cada item do intervalo. Envolve os atributos de tamanho e o ângulo de cada palo, além da relação de associação com o *enum* de gancho podendo ser de cima ou de baixo. Elemento principal do teste, comprehende na detecção dos traços encontrados na imagem e armazenar as informações pertinentes nestes atributos.

4.2.2 Desenvolvimento

Nessa seção serão demonstrados os resultados alcançados na fase de desenvolvimento deste trabalho.

4.2.2.1 Aplicativo

O aplicativo obteve o nome de FRITE devido a ser o acrônimo do título deste trabalho: Ferramente de Reconhecimento de Imagens para um Teste Expressivo de Personalidade. Na sequência, está contido o esclarecimento sobre as telas desenvolvidas.

Os dados dos usuários caracterizados como sensíveis nas imagens seguintes foram removidos e/ou substituídos para elementos fictícios com o intuito de preservar a integridade da correção do teste palográfico.

Figura 13 – Tela de Login

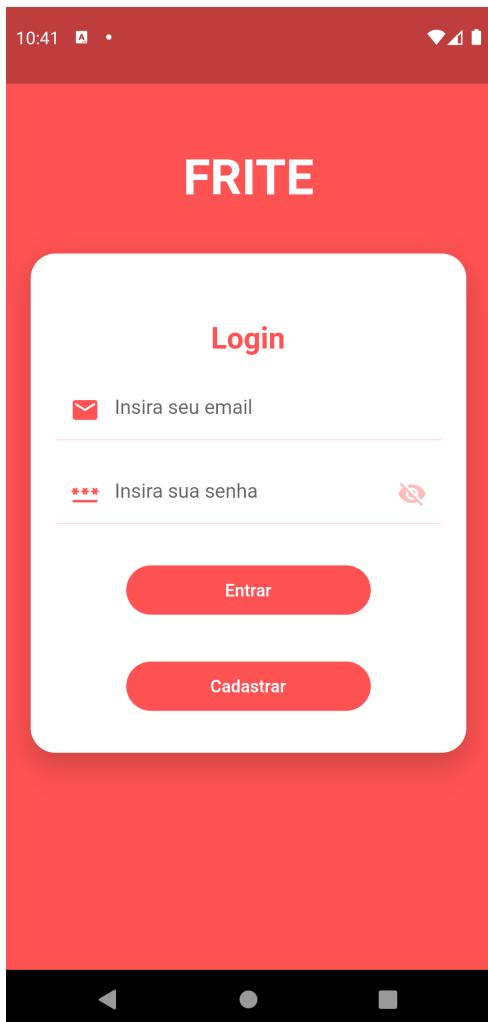


Figura 14 – Tela de Cadastro do Psicólogo

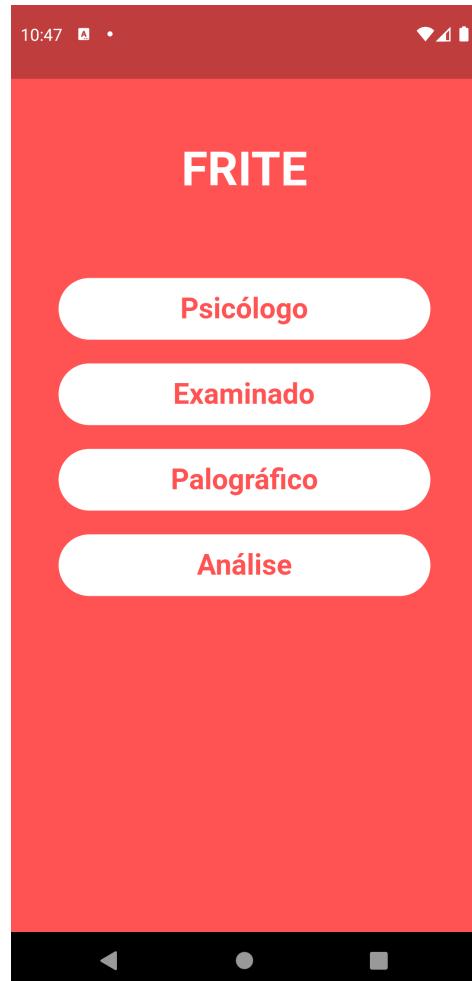


Fonte: Próprio Autor (2021)

A Figura 13 acima retrata a tela inicial do aplicativo, no qual o psicólogo consegue acessar o sistema por completo. No login, são requisitados os campos de email e senha previamente cadastrados. Caso não possua, é possível ser redirecionado para a respectiva tela ao realizar o clique no botão de "Cadastrar".

A tela exibida na Figura 14 representa o cadastro do psicólogo na plataforma. Nela, são solicitados as informações de nome, CPF, data de nascimento, email e senha. Ao clicar em "Salvar", tendo suas informações validadadas, será redirecionado para a tela de Login, como mostra na Figura 13, para acesso ao sistema.

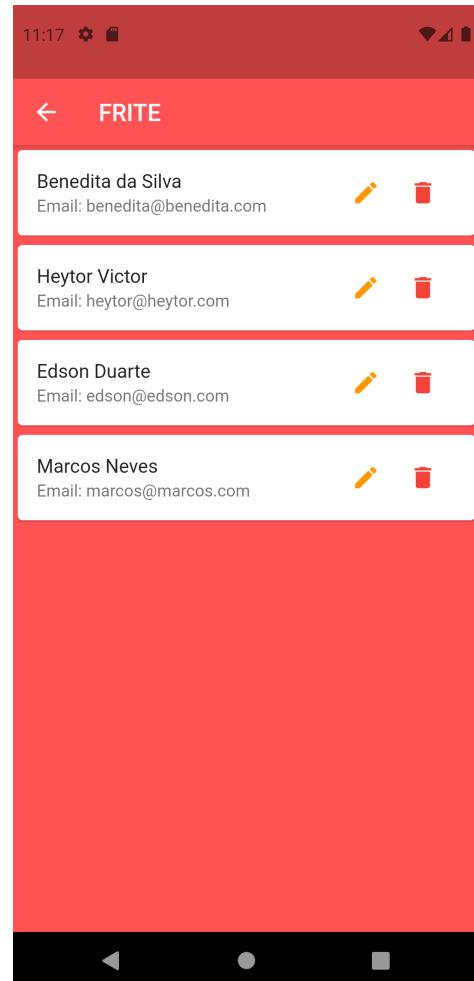
Figura 15 – Tela de Menu



Fonte: Próprio Autor (2021)

A Figura 15 é a tela de menu do aplicativo no qual o usuário, profissional da psicologia, consegue navegar entre as funcionalidades desenvolvidas: as áreas do psicólogo, examinado, palográfico e análise do teste.

Figura 16 – Tela de Listagem dos Psicológos



Fonte: Próprio Autor (2021)

A Figura 16 apresenta a tela de listagem de todos os psicólogos, além das funcionalidades de atualização e exclusão do sistema.

Figura 17 – Tela de Cadastro do Examinado



Figura 18 – Tela de Listagem do Examinado

Listagem de Examinados		
Adalberto Correia		
CPF: 040		
Bernardo Fragoso		
CPF: 616		
Caio Cesar Oliveira		
CPF: 830		
Douglas Costa		
CPF: 448		
Emiliana Vaz		
CPF: 800		

Fonte: Próprio Autor (2021)

A tela apresentada na Figura 17 exibe o cadastro do examinado na ferramenta. Para tal função, é solicitado os seguintes campos: o nome, o CPF, a data de nascimento, a cidade e estado de nascimento, o gênero e a escolaridade. Realizada a validação dos dados inseridos, o usuário será encaminhado para a tela de listagem como mostra na Figura 18.

De mesmo modo, a Figura 18 mostra a tela de listagem dos examinados, com as mesmas funcionalidades citadas anteriormente, e adiante com a possibilidade da adição do examinado no sistema.

Figura 19 – Tela de Cadastro do Palográfico

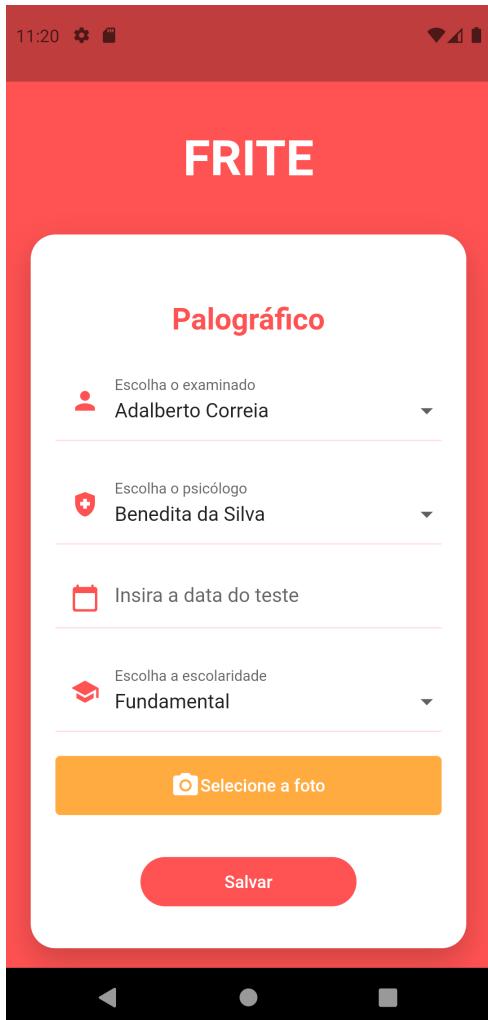
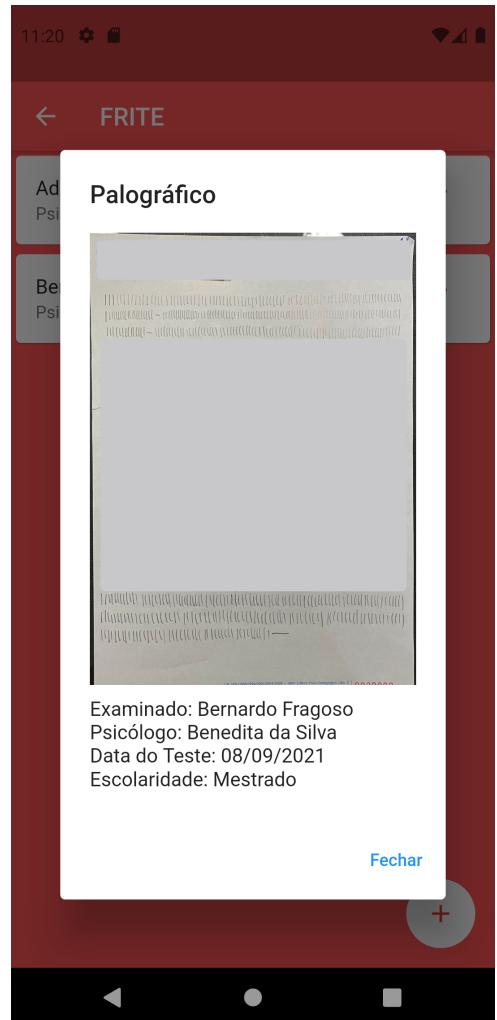


Figura 20 – Tela de Detalhamento do Palográfico

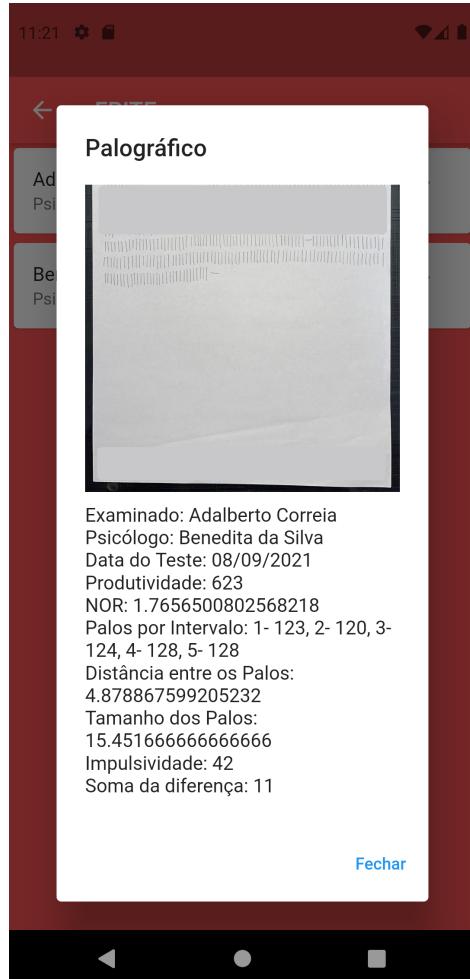


Fonte: Próprio Autor (2021)

Na Figura 19 representa a tela de cadastro do teste palográfico na aplicação. Uma vez inserido, será possível analisar e gerar os resultados para o psicólogo. Nesta tela, o usuário deve selecionar o examinado, o psicólogo, a data de realização do teste e a escolaridade do examinado no momento da realização do teste. Posteriormente, deverá escolher a foto armazenada na galeria do aparelho celular para ser encaminhada ao sistema. Com as informações dispostas, o aplicativo irá validá-las e, tendo resposta positiva, será encaminhada para a listagem dos testes palográficos. Caso tenha uma resposta negativa relacionado a qualidade da imagem, prevista na métrica 3.4.2, irá retornar um alerta para o usuário informando que o arquivo não alcançou a resolução requerida para análise.

Após o palográfico ser inserido no sistema, é possível visualizar o detalhamento das informações de um teste específico. Com os mesmos campos solicitados no seu cadastro, a Figura 20 apresenta como estes dados são dispostos no aplicativo.

Figura 21 – Tela de Detalhamento da Análise do Palográfico



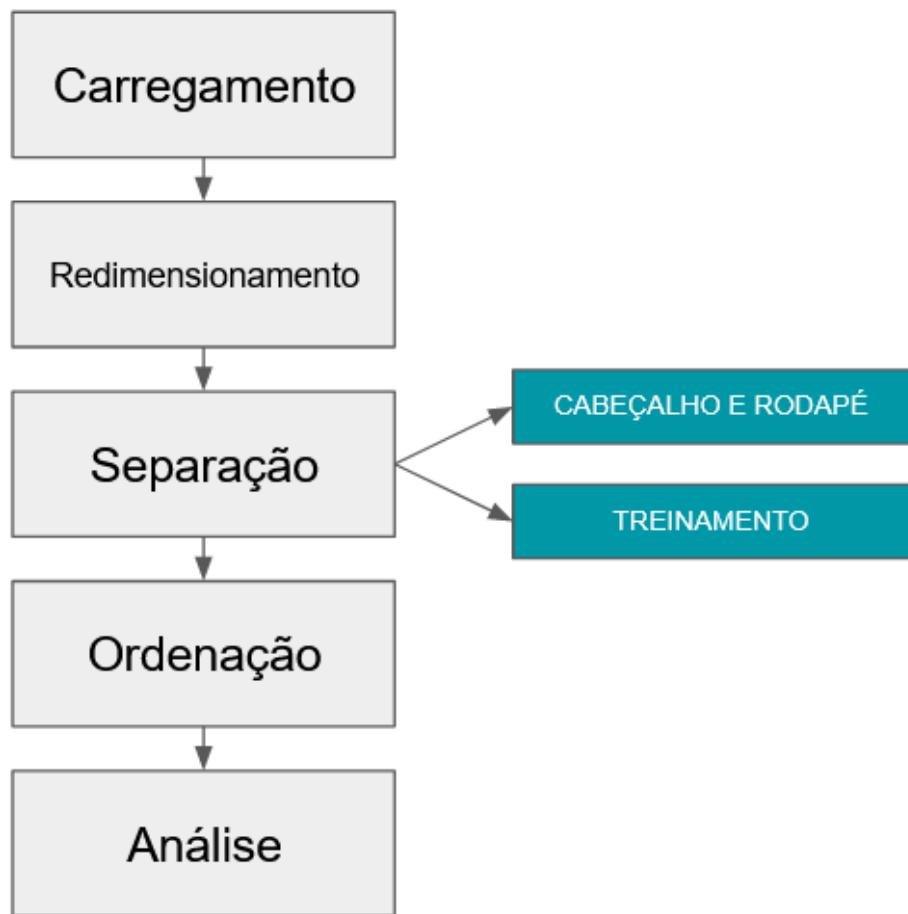
Fonte: Próprio Autor (2021)

Na Figura 21 retrata a solução da problemática proposta neste trabalho. Após o palográfico ser cadastrado, a análise da imagem está disponível para o usuário. Nesta tela, é possível visualizar o nome do examinado, o psicólogo responsável pelo teste, a data de realização do teste e o resultado do processamento: a produtividade, a soma da diferença, o nível de oscilação rítmica (NOR), a quantidade de palos por intervalo, a distância entre os palos, o tamanho dos palos e a impulsividade.

4.2.2.2 Algoritmo do Processamento de Imagem

Nesta subseção exemplifica o fluxo do processamento da imagem, como mostra na Figura 22 logo abaixo. Uma vez em que imagem é armazenada na ferramenta, o algoritmo para tal função é executado iniciando-se pelo processo de carregamento.

Figura 22 – Fluxograma do Processamento de Imagem



Fonte: Próprio Autor (2021)

Nessa parte, a imagem enviada é carregada e realiza um cálculo da altura original com uma escala de altura nova previamente estabelecida e mensura uma nova proporção. Diante disso, é possível obter os contornos da folha e identificar as extremidades presentes no arquivo, realizando uma segmentação nesse reconhecimento, transformando para escala de cinza, aplicando filtros de correção e sendo convertida para monocromática (preto e branco). Na Listagem 1 é possível visualizar o algoritmo desenvolvido para este processo.

```

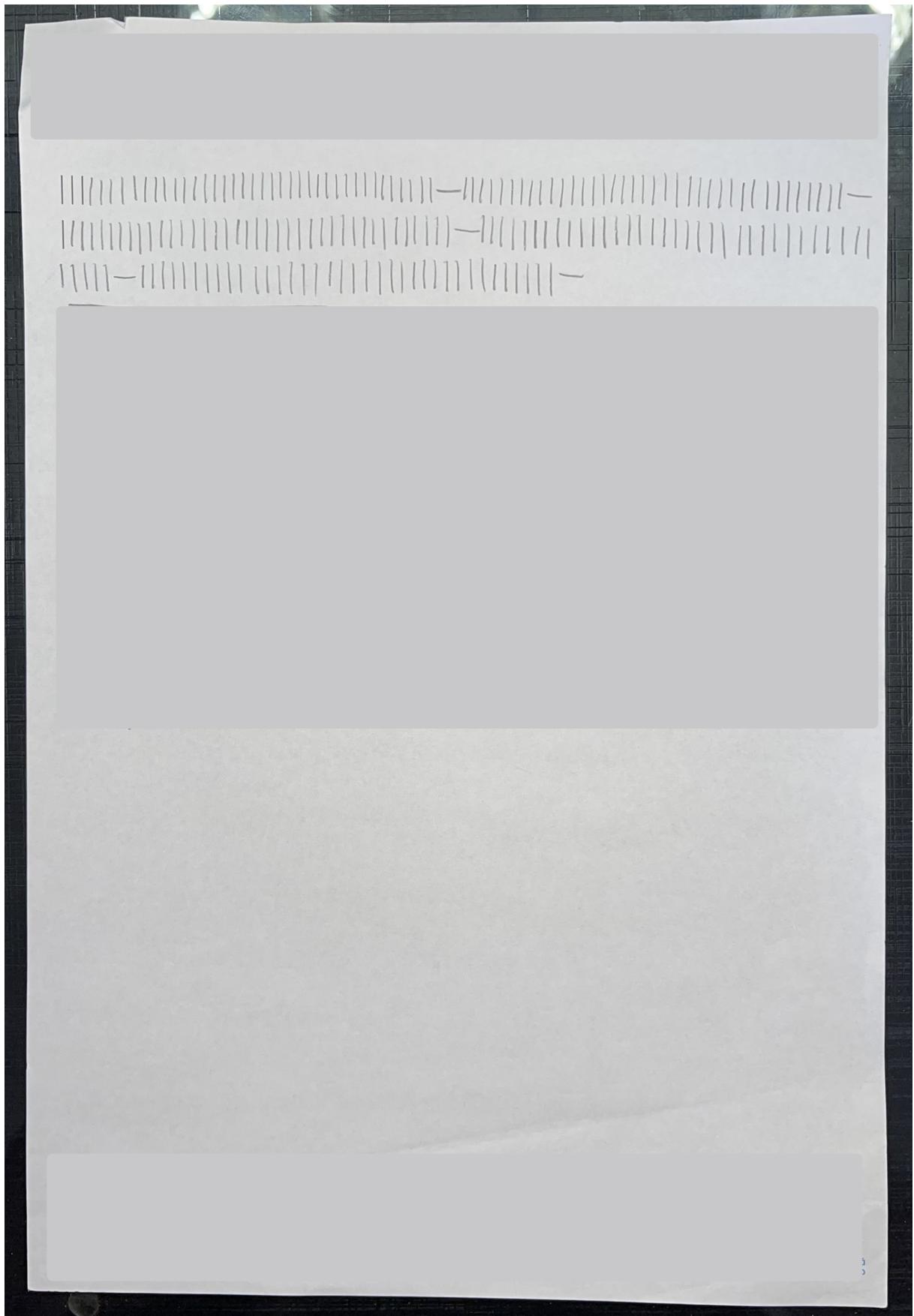
1 def scan(self, image_path):
2     RESCALED_HEIGHT = 500.0
3     OUTPUT_DIR = '/usr/src/frite/api/detection/images/scanner'
4
5     image = cv2.imread(image_path)
6
7     assert(image is not None)
8
9     ratio = image.shape[0] / RESCALED_HEIGHT
10    orig = image.copy()
11    rescaled_image = resize(image, height = int(RESCALED_HEIGHT))
12
13    screen_cnt = self.get_contour(rescaled_image)
14
15    if self.interactive:
16        screen_cnt = self.interactive_get_contour(screen_cnt, rescaled_image)
17
18    warped = four_point_transform(orig, screen_cnt * ratio)
19
20    gray = cv2.cvtColor(warped, cv2.COLOR_BGR2GRAY)
21
22    sharpen = cv2.GaussianBlur(gray, (0,0), 3)
23    sharpen = cv2.addWeighted(gray, 1.5, sharpen, -0.5, 0)
24
25    thresh = cv2.adaptiveThreshold(sharpen, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
26                                    cv2.THRESH_BINARY, 21, 15)
27
28    basename = os.path.basename(image_path)
29    cv2.imwrite(OUTPUT_DIR + '/' + basename, thresh)
30    print("Processed " + basename)

```

Listagem 1 – Scanner da Imagem

No processo de scanner, observou a presença de alguns filtros referidos no parágrafo anterior. Por intermédio da biblioteca OpenCV, foram aplicados os métodos *cvtColor* (para converter a imagem no espaço de cor cinza), *GaussianBlur* (para suavizar a imagem calculando média dos valores presentes nos pixels com os pixels vizinhos), *addWeighted* (para unificar o retorno dos métodos anteriores atribuindo determinado peso em cada) e o *adaptiveThreshold* (para corrigir a iluminação não uniforme e transformar em uma imagem binária).

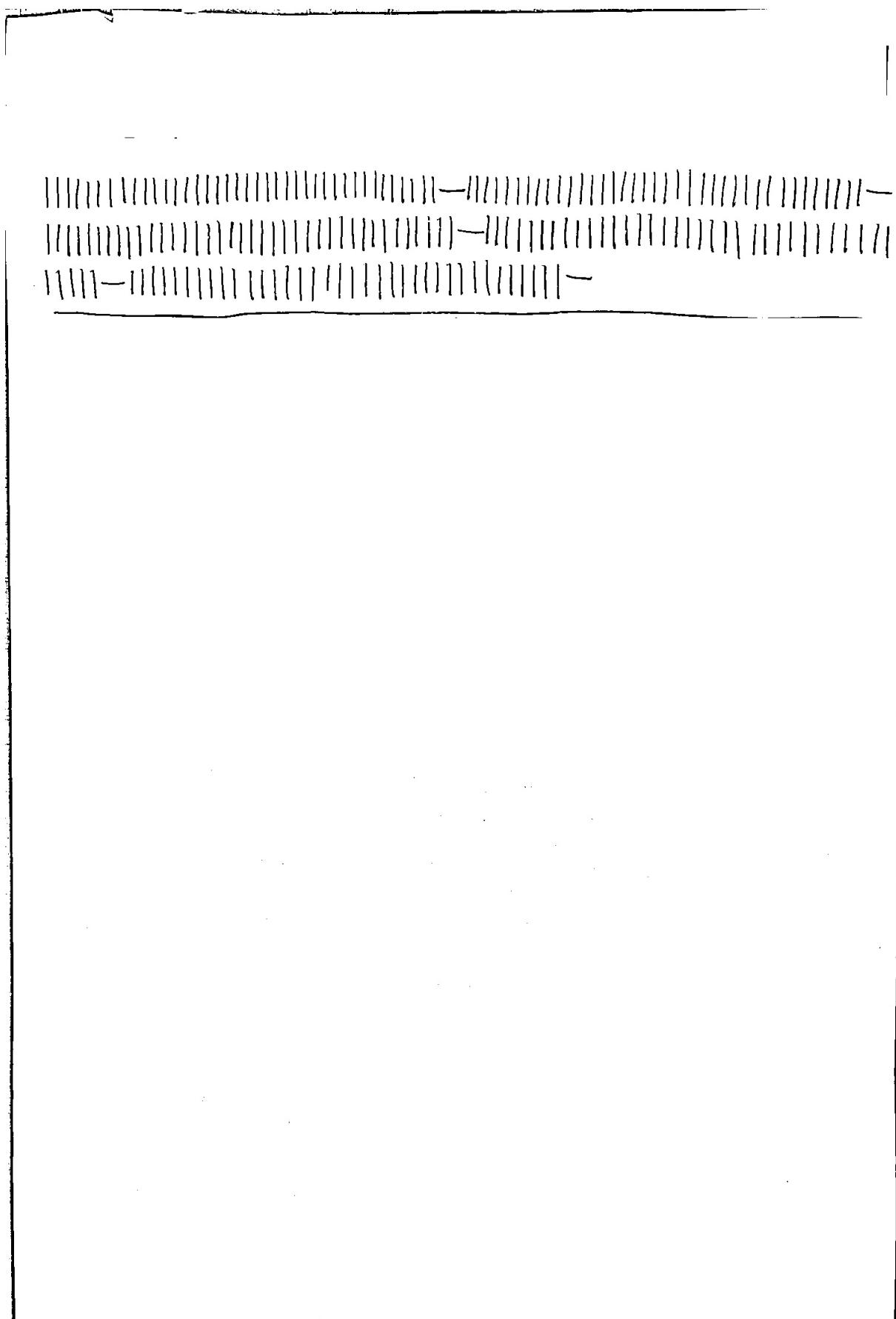
Figura 23 – Teste Palográfico - Original



Fonte: Próprio Autor (2021)

É importante ressaltar que as imagens empregadas neste trabalho a respeito da visualização do teste palográfico estão parcialmente censuradas ou cortadas por serem consideradas com alto grau de sigilo, visto que se torna fundamental o respeito dos princípios éticos da psicologia envolvendo a análise destes testes. Na Figura 23 anterior, retrata a fotografia original do teste palográfico e, em seguida, demonstra o arquivo processado nesta etapa.

Figura 24 – Teste Palográfico - Scanner



Fonte: Próprio Autor (2021)

Na sequência, com base na Listagem 2 abaixo, se fez um redimensionamento na imagem previamente escaneada com o objetivo de condicionar o padrão de proporção nos testes recebidos. Como molde, estabeleceu o tamanho de 2250 pixels de altura e uma métrica na largura da imagem para todos os arquivos.

```

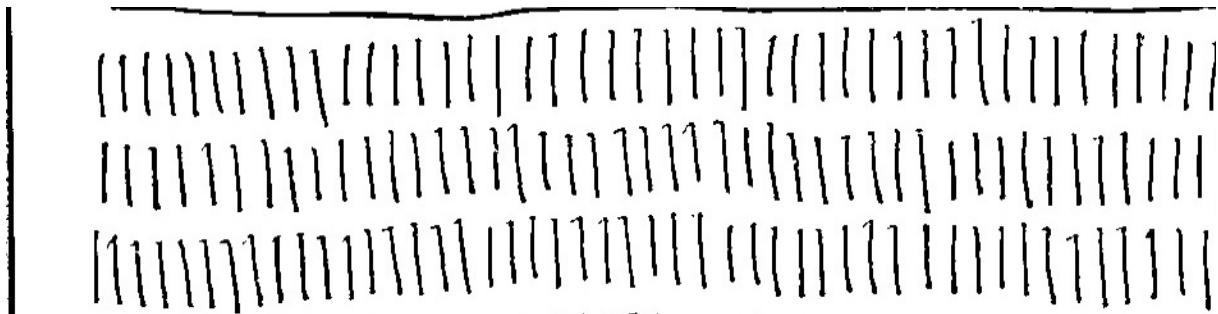
1 def resize_image(path):
2     OUTPUT_DIR = '/usr/src/frite/api/detection/images/resize'
3     basename = os.path.basename(path)
4     img = cv2.imread(path, cv2.IMREAD_UNCHANGED)
5
6     height = 2250
7     width = img.shape[1]
8     if img.shape[1] < 1300:
9         width = img.shape[1] * 2
10
11    dim = (width, height)
12
13    resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
14    cv2.imwrite(OUTPUT_DIR + '/' + basename, resized)

```

Listagem 2 – Redimensionamento da Imagem

Posteriormente, efetua uma separação de itens presentes na folha, como o cabeçalho, o rodapé e o treinamento existente antes de executar o teste palográfico, para que os examinados tenham conhecimento da atividade. Como no processo anterior, houve um processamento para que todas as imagens estiverem padronizadas, pode-se destacar a região de interesse do teste baseado na dimensão do pixel, e realizar a segregação da mesma. A Figura 25 pode-se observar uma fração da imagem aplicada nesse método e na Listagem 3 demonstra um fragmento do algoritmo desta operação.

Figura 25 – Teste Palográfico - Separação



Fonte: Próprio Autor (2021)

```

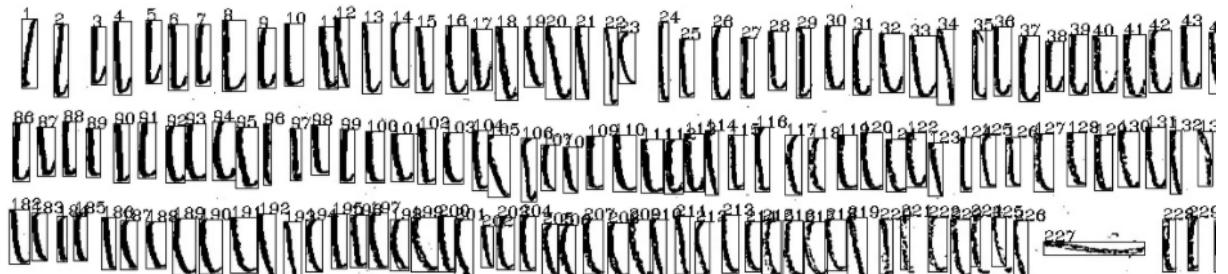
1 def crop_image_header_footer(image_path):
2     OUTPUT_DIR = '/usr/src/frite/api/detection/images/crop_header_footer'
3     basename = os.path.basename(image_path)
4     image = cv2.imread(image_path)
5
6     width = image.shape[1]
7     TOP_CROP = 220
8     BOTTOM_CROP = 2000
9     crop_image = image[TOP_CROP:BOTTOM_CROP, 0:width]
10
11    cv2.imwrite(OUTPUT_DIR + '/' + basename, crop_image)

```

Listagem 3 – Separação da Imagem

A próxima etapa são identificados os contornos na imagem derivado na fase precedente, filtrando-os pelo período de dimensões e área, e gerando um arquivo no formato JSON com o resultado obtido. Em sequência, o mesmo é lido e produzido a ordenação dos palos encontrados. Partindo de cima para baixo e da esquerda para direita, a estrutura é realizada baseada no primeiro palo encontrado da linha e encontrar o próximo vizinho pela menor distância, sendo assim, os itens são enumerados e organizados pela linha e por intervalo. Na Figura 26 abaixo corresponde a esta fase e na Listagem 4, a qual apresenta o método responsável por fazer esse processo.

Figura 26 – Teste Palográfico - Ordenação



```

1 def set_ordination(img_path):
2     OUTPUT_DIR = '/usr/src/frite/api/detection/images/sof'
3     basename = os.path.basename(img_path)
4     img = cv2.imread(img_path, 0)
5
6     obj = get_palos_image(img_path)
7     palo_list = read_palos_list(obj)
8     result = return_palos(palo_list)
9
10    for palo in result:
11        cv2.rectangle(img, (palo.x, palo.y), (palo.x + palo.w, palo.y + palo.h),
12                      (0, 0, 255), 1)
13        cv2.putText(img, str(palo.roi), (palo.x, palo.y),
14                    cv2.FONT_HERSHEY_COMPLEX_SMALL, 0.75, (0, 255, 0), 1)
15
16    row_values = {palo.row for palo in result}
17    for row in row_values:
18        aux = [x for x in result if x.row == int(row)]
19        for i in aux:
20            if aux[len(aux)-1] == i:
21                i.is_last_row = True
22            else:
23                i.is_last_row = False
24
25    aux = []
26    for i in result:
27        aux.append(i.__dict__)
28
29    with open(f'/usr/src/frite/api/detection/images/json/{basename}.json', 'w') as f:
30        f.write(json.dumps(aux))
31
32    cv2.imwrite(OUTPUT_DIR + '/' + basename, img)
33    return result

```

Listagem 4 – Ordenação dos Palos

Por fim, são geradas as informações das análises que auxiliam o psicólogo na correção do teste que encontram-se na tela do aplicativo demonstrado pela Figura 21.

```

1  {
2      "productivity": total_palos,
3      "nor": nor,
4      "income": str(palos_by_interval),
5      "distance_between_palos": dist_palo,
6      "palos_size": calculate_size_palos(size_palos),
7      "impulsiveness": impulsiveness,
8      "difference_sum": sum_diff
9  }

```

Listagem 5 – Análise do Teste Palográfico

Estes dados são construídos através de métodos desenvolvidos, com base nas considerações realizadas pela psicóloga em que acompanha este trabalho, e constituídos na notação de JSON, como exemplifica a Listagem 5 acima.

4.2.2.3 API

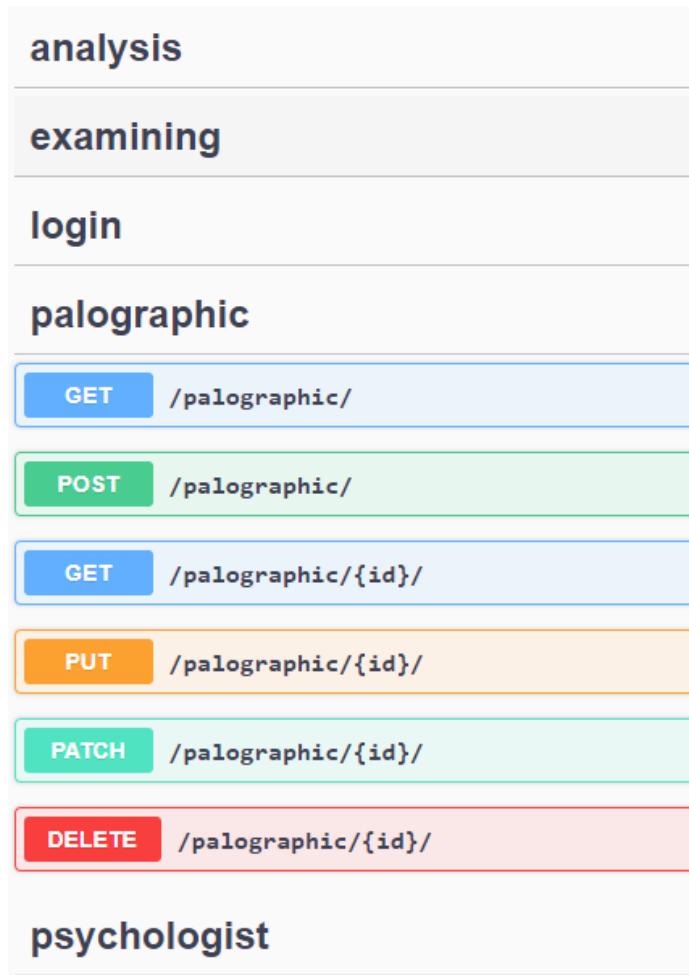
A construção da API se originou com o objetivo de realizar a comunicação entre a interface do usuário e o banco de dados da ferramenta. Para este trabalho, a arquitetura definida com o destino ao desenvolvimento desta API está no modelo REST. Desse modo, as URIs são estabelecidas bem como os métodos HTTP presentes no padrão determinado e com a estrutura de dados selecionada no corpo das requisições e respostas da API foi o JSON.

Como uma boa prática do desenvolvimento de API na arquitetura REST, é de fundamental importância disponibilizar uma documentação detalhando as rotas existentes, bem como os seus respectivos métodos aceitados e atributos presentes no corpo da requisição. Neste trabalho, todas as rotas foram particularizadas através da especificação OpenAPI¹ e com uma interface gráfica gerada por meio da aplicação SwaggerUI², para que seja possível uma explicação visual e prática das funcionalidades presentes na ferramenta proposta. Na Figura 27 exibe um demonstrativo da tela do Swagger quando é requisitada por um navegador.

¹ <https://www.openapis.org/>

² <https://swagger.io/>

Figura 27 – Rotas da API



Fonte: Próprio Autor (2021)

Também, pela Figura 27, é possível visualizar todas as rotas desenvolvidas para este trabalho. Em sequência, no Quadro 3 logo abaixo, representa o modelo REST para cada rota, e neste exemplo específico, para a rota /palographic.

Quadro 3 – Exemplo dos Métodos REST na Rota /palographic

Método	Ação	Sucesso	Erro
GET /palographic/id	Retorna um palográfico	200	404
GET /palographic/	Retorna uma lista de palográficos	200	404
POST /palographic/	Cria uma palográfico	201	404, 409
PATCH /palographic/id	Atualiza um atributo do palográfico	200	404
PUT /palographic/id	Atualiza um palográfico por completo	200	404
DELETE /palographic/id	Deleta um palográfico	204	404

Fonte: Próprio Autor (2021)

Na continuidade sobre as rotas, é imprescindível que as informações sejam transmitidas nos protocolos HTTP de forma segura através da autenticação, a fim de que possa garantir a integridade dos dados. Uma das práticas existentes se baseia pela autenticação de token, isto é, o usuário da aplicação gerar um chave criptografada (token) em que será utilizada no cabeçalho de cada requisição solicitada pelo usuário durante a execução de alguma funcionalidade e efetuar a verificação de permissões de acesso (MONTANHEIRO; CARVALHO; RODRIGUES, 2016).

Figura 28 – Exemplo da Autenticação na API pelo JWT



```
{  
    "success": "True",  
    "status code": 200,  
    "message": "User logged in successfully",  
    "token":  
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjo  
        [REDACTED]  
        YVtUmJsJAVAFl9wER6eHI4kJiDZD4yimL8NdWxaq9Q"  
}
```

Fonte: Próprio Autor (2021)

Para esta ferramenta, sucedeu o emprego do JWT³ (JSON Web Token) para legitimar a garantia em todo funcionamento e a Figura 28 exemplifica o retorno do token concebido após o usuário entrar no aplicativo. Nela contém os atributos indicando se houve sucesso na requisição, o código de status da requisição, uma mensagem apontando alguma informação e a chave secreta determinando o token, este por sua vez, tratando-se de um dado sensível e pertinente a segurança da aplicação, está parcialmente censurado.

4.3 Estudo de Caso

Este estudo de caso, decorrente a análise documental realizada para o comparativo dos resultados, envolveu o cumprimento da métrica proposta acerca da eficiência do processamento da imagem, exposta na subseção 3.4.1, no qual circunda a análise qualitativa dos testes observando os resultados em relação ao software SKIP, bem como a correção desenvolvida por uma psicóloga.

³ <https://jwt.io/>

Cabe ressaltar que tais correções dos testes apresentados aqui já estavam aplicados, corrigidos e arquivados pela psicóloga, sendo utilizados neste trabalho apenas para fins de comparação, e por isso, as informações dos examinados serão resguardadas.

Se faz relevante enfatizar que o objeto de comparação do SKIP para os resultados deste trabalho está na sua avaliação inicial automática. Para tanto, nove testes foram disponibilizados, sendo sete parâmetros de correção no teste palográfico, citados no Quadro 4 abaixo, foram destacados como métodos de conferência e logo serão especificados.

Quadro 4 – Métricas de Análise no Teste Palográfico

ID	Parâmetro
1	Produtividade
2	Rendimento
3	NOR
4	Soma da Diferença
5	Distância entre Palos
6	Impulsividade
7	Tamanho dos Palos

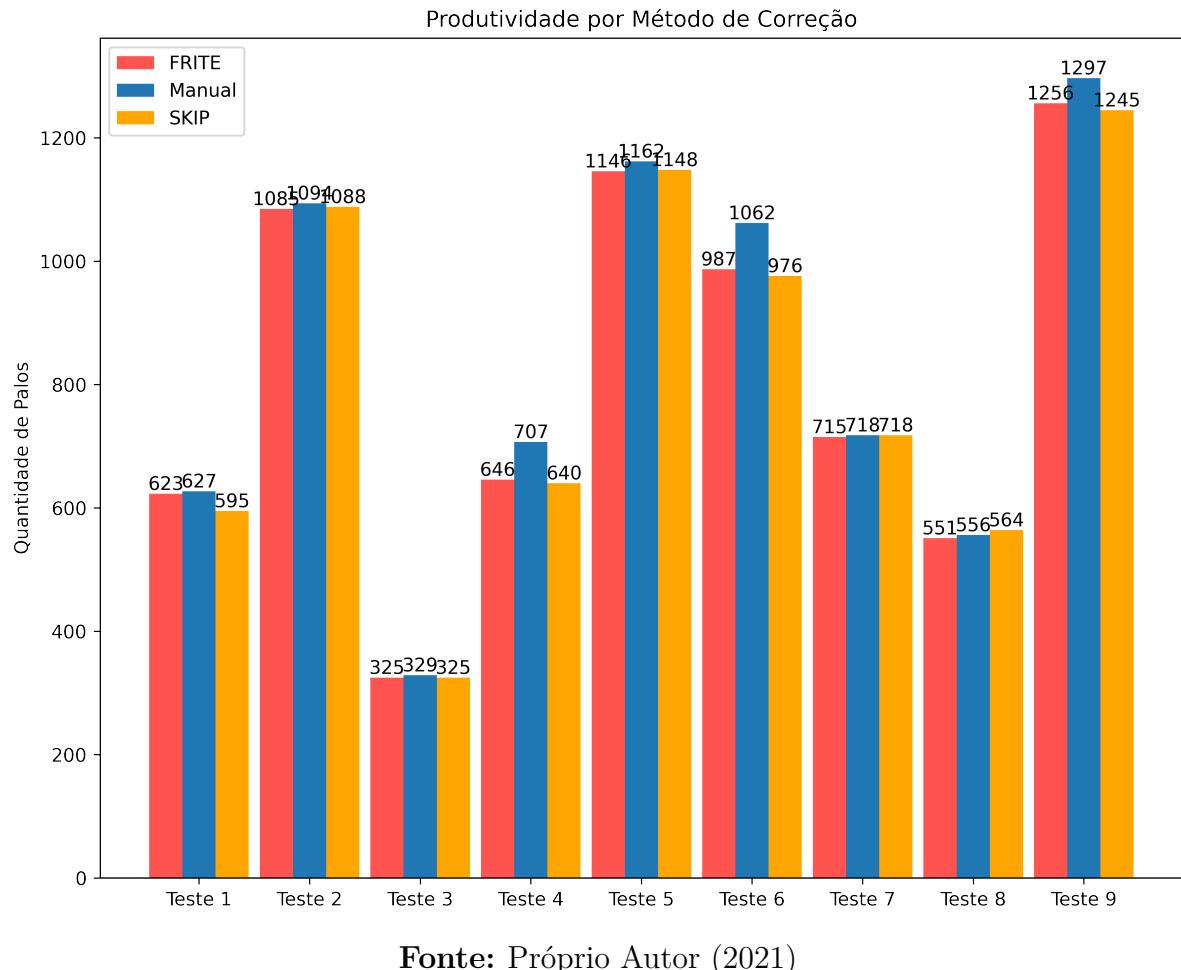
Fonte: Próprio Autor (2021)

Torna-se relevante novamente enfatizar que os atributos presentes na correção não poderão ser descritos, tal qual as fórmulas matemáticas, de forma minuciosa em detrimento acerca da mesma explicação referente ao uso restrito da visualização do teste palográfico.

4.3.1 Produtividade

A ferramenta proposta nesse trabalho obteve uma média de 24 palos não detectados, tendo uma acurácia de 97.30% na identificação da quantidade de palos presentes no teste em comparação com o método manual, apresentando 4 palos não reconhecidos (99.17%) como o melhor resultado.

Figura 29 – Gráfico | Produtividade

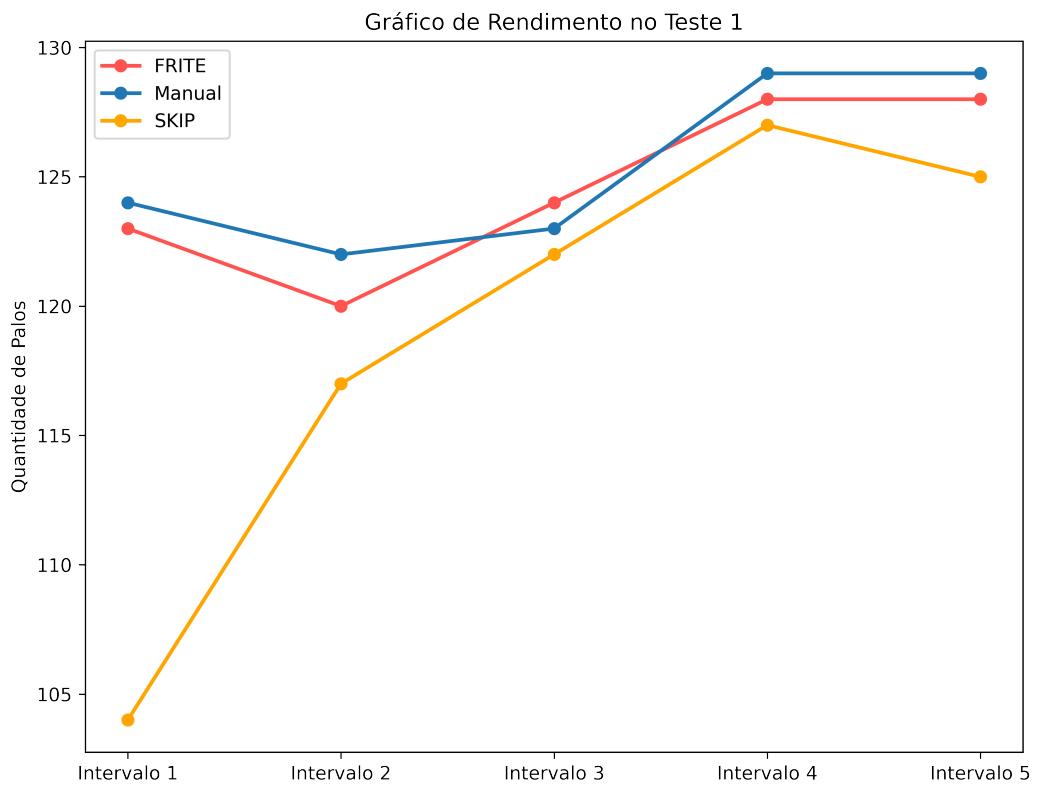


Em contrapartida, o software SKIP alcançou uma média de 30 palos não reconhecidos (96.8% de precisão) confrontado a mesma verificação. Na Figura 29 acima, pode-se observar o gráfico em valor referente a este atributo.

4.3.2 Rendimento

Buscando uma comparação mais detalhada, elaborou-se um gráfico no que concerne o rendimento do examinado baseado na quantidade de palos durante os cinco intervalos com o emprego de um teste palográfico (Teste 1).

Figura 30 – Gráfico | Rendimento



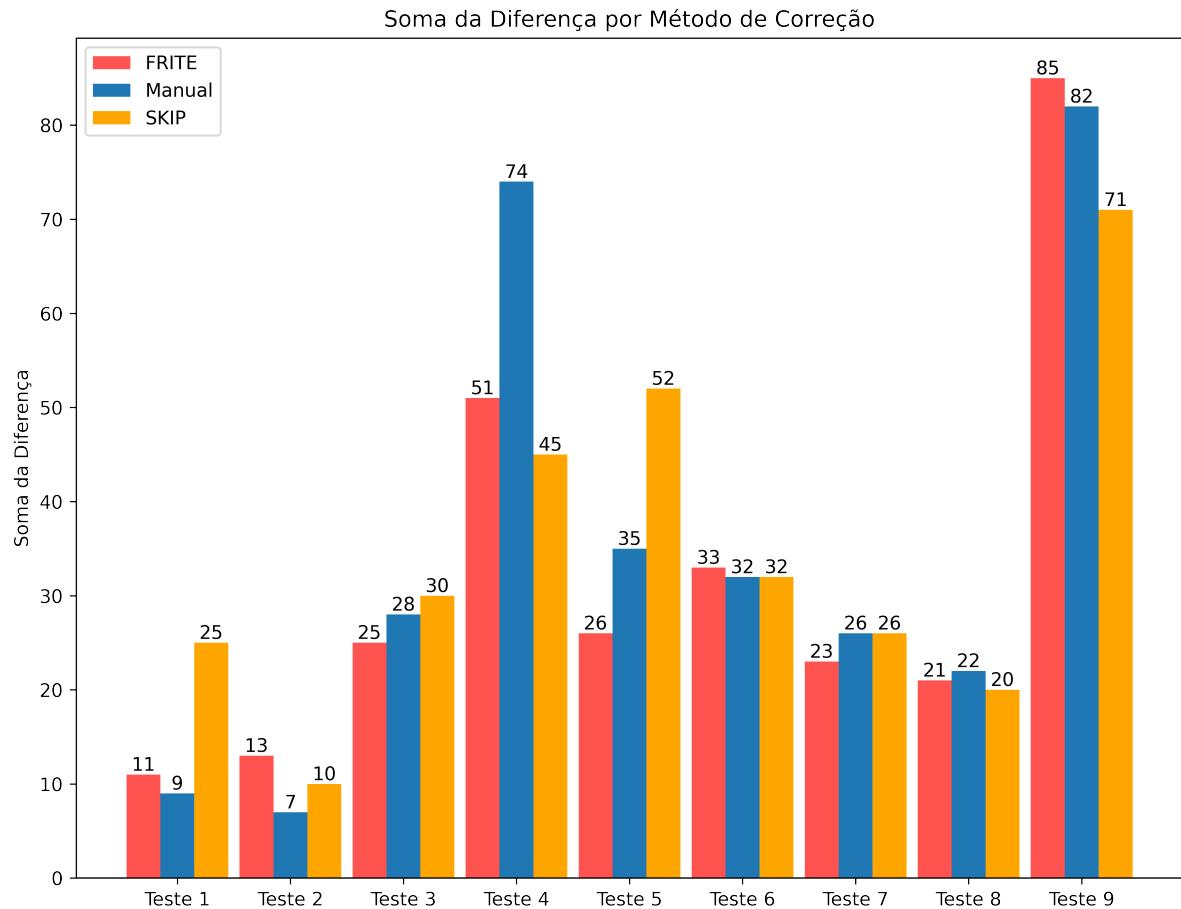
Fonte: Próprio Autor (2021)

A análise, representada na Figura 30, demonstra um número bastante positivo e significativo, tendo em consideração que aplicação apresenta uma média de divergência de apenas um palo por intervalo, ao contrário da ferramenta SKIP em que houve uma disparidade na contagem de palos no primeiro intervalo porém na sequência manteve uma tendência.

4.3.3 Soma da Diferença

Este atributo é resultado por meio da soma da diferença na quantidade de palos presentes em cada intervalo, dessa forma, a análise da aplicação desenvolvida produziu uma média de 5.66 na diferença (82.83% de precisão) comparada a correção manual.

Figura 31 – Gráfico | Soma da Diferença



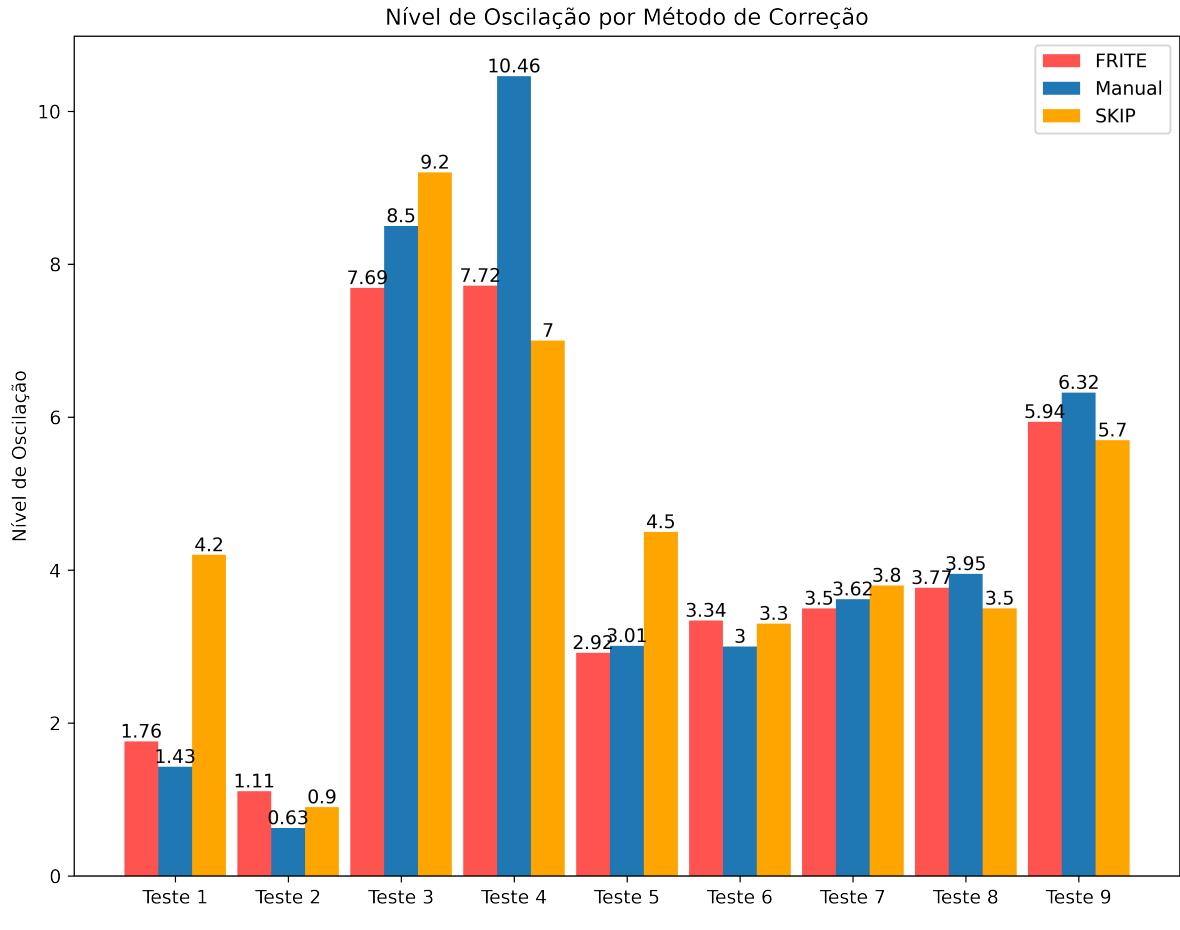
Fonte: Próprio Autor (2021)

No entanto, a ferramenta SKIP obteve uma pontuação de 8.88 na mesma conferência (taxa de 78.32%), adquirindo uma diferença de 4.51% em relação ao aplicativo FRITE. O gráfico de resultado desse componente está na Figura 31 acima.

4.3.4 NOR

Para avaliar o Nível de Oscilação Rítmica (NOR), em síntese é utilizado outros parâmetros de correção como a produtividade e a soma da diferença, efetua-se um cálculo, através de uma fórmula contida nas regras do teste palográfico.

Figura 32 – Gráfico | NOR



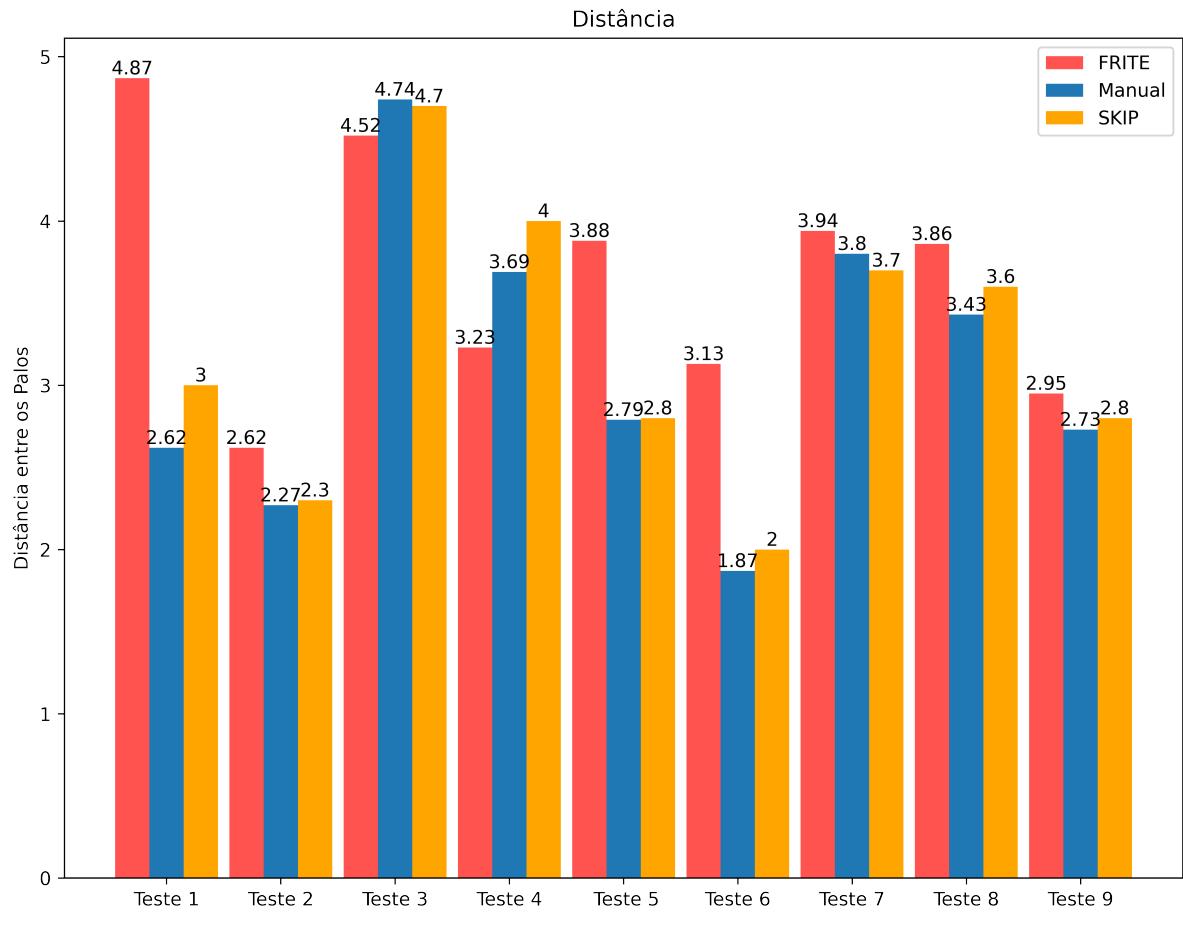
Fonte: Próprio Autor (2021)

Logo, verificou-se um intervalo de diferença na pontuação menor entre a correção do método manual e o aplicativo FRITE que, em comparação, com o software SKIP, gerando um índice satisfatório para a ferramenta desenvolvida neste trabalho. A Figura 32 apresenta o comparativo do NOR entre os métodos de correção.

4.3.5 Distância entre Palos

Determina-se a distância entre os palos, em milímetros, através de uma fórmula em que comprehende alguns somatórios e frações decorrente de resultados de outros componentes do teste palográfico. Ou seja, neste campo os intervalos são medidos e ao final é realizado um cálculo de média.

Figura 33 – Gráfico | Distância entre Palos



Fonte: Próprio Autor (2021)

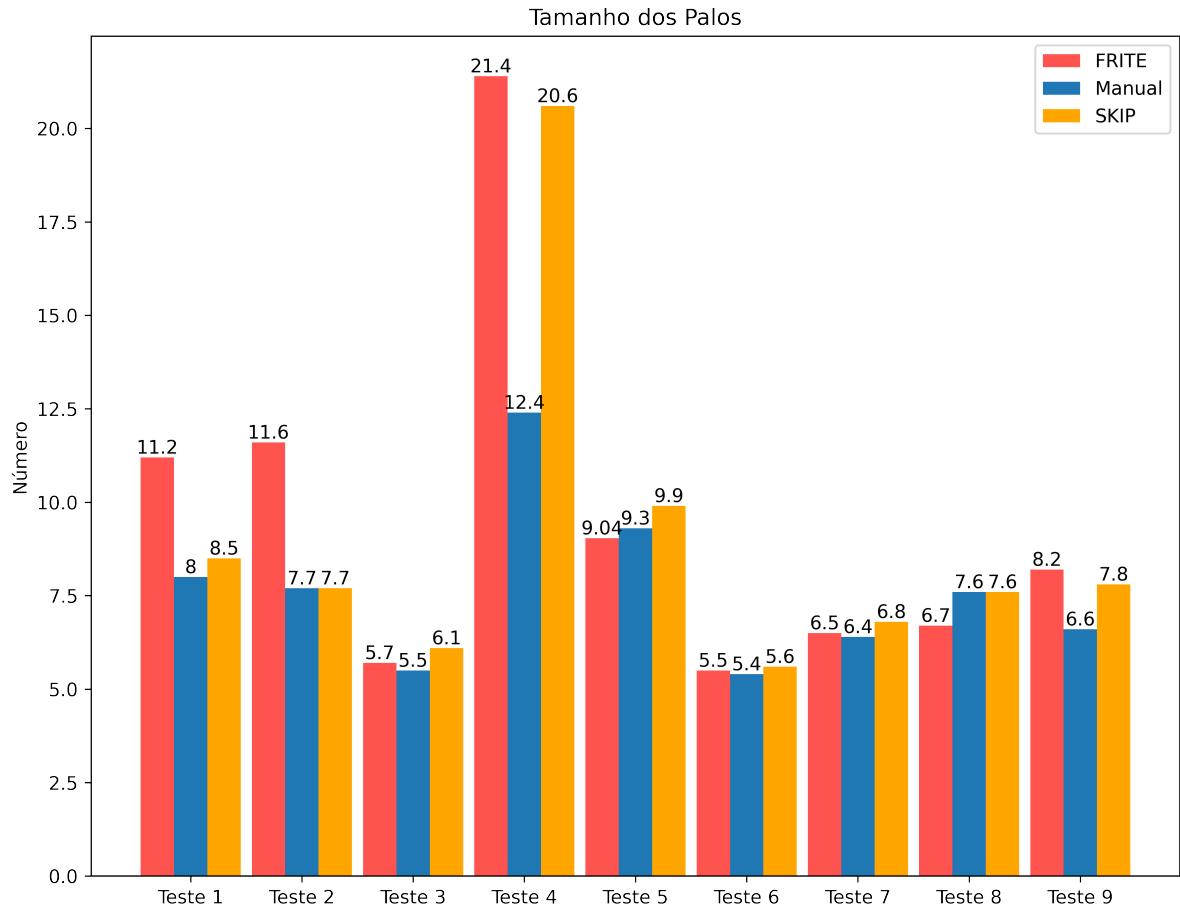
A Figura 33 é possível visualizar o gráfico referente a este quesito. O software SKIP alcançou o resultado mais próximo comparado ao aplicativo deste trabalho em relação a correção manual considerando uma média entre os testes realizados. Também é perceptível uma diferença entre o aplicativo FRITE em relação aos demais métodos no Teste 1.

Como o cálculo é realizado baseado entre os intervalos, caso o algoritmo não consiga identificar algum palo próximos as margens neste intervalo, o algoritmo pode enfrentar dificuldade de realizar esta análise precisa.

4.3.6 Tamanho dos Palos

Através da seleção de determinados palos presentes no teste, realiza um cálculo para coletar o resultado relacionado ao tamanho dos palos. De acordo com a Figura 34 abaixo, o gráfico apresenta números semelhantes em alguns testes palográficos de forma alternada.

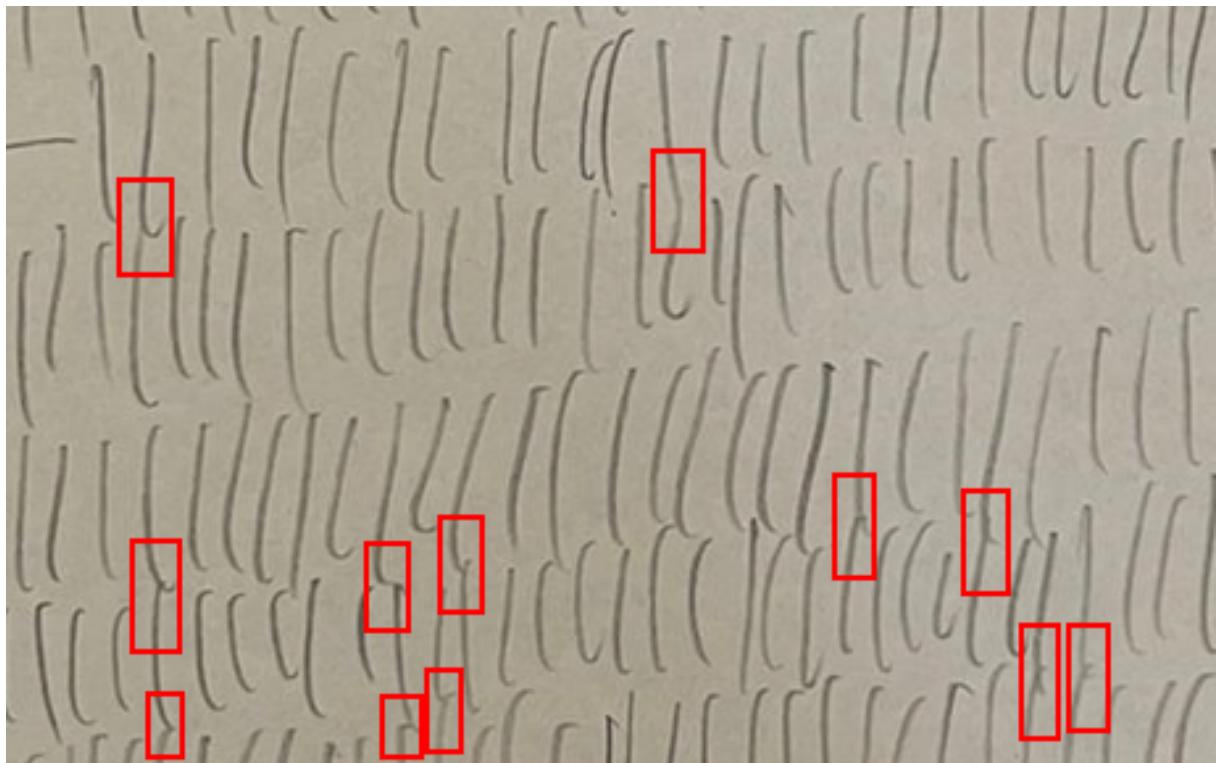
Figura 34 – Gráfico | Tamanho dos Palos



Fonte: Próprio Autor (2021)

Entretanto, percebe-se uma apuração favorável a ferramenta SKIP em relação ao FRITE comparando com a correção manual, visto que o aplicativo pode não ter detectado alguns palos menores que contribuíram para esta conclusão. Também, nota-se uma discrepância dos métodos tecnológicos em relação a análise da psicóloga no Teste 4.

Figura 35 – Identificação de Palos Conectados no Teste 4



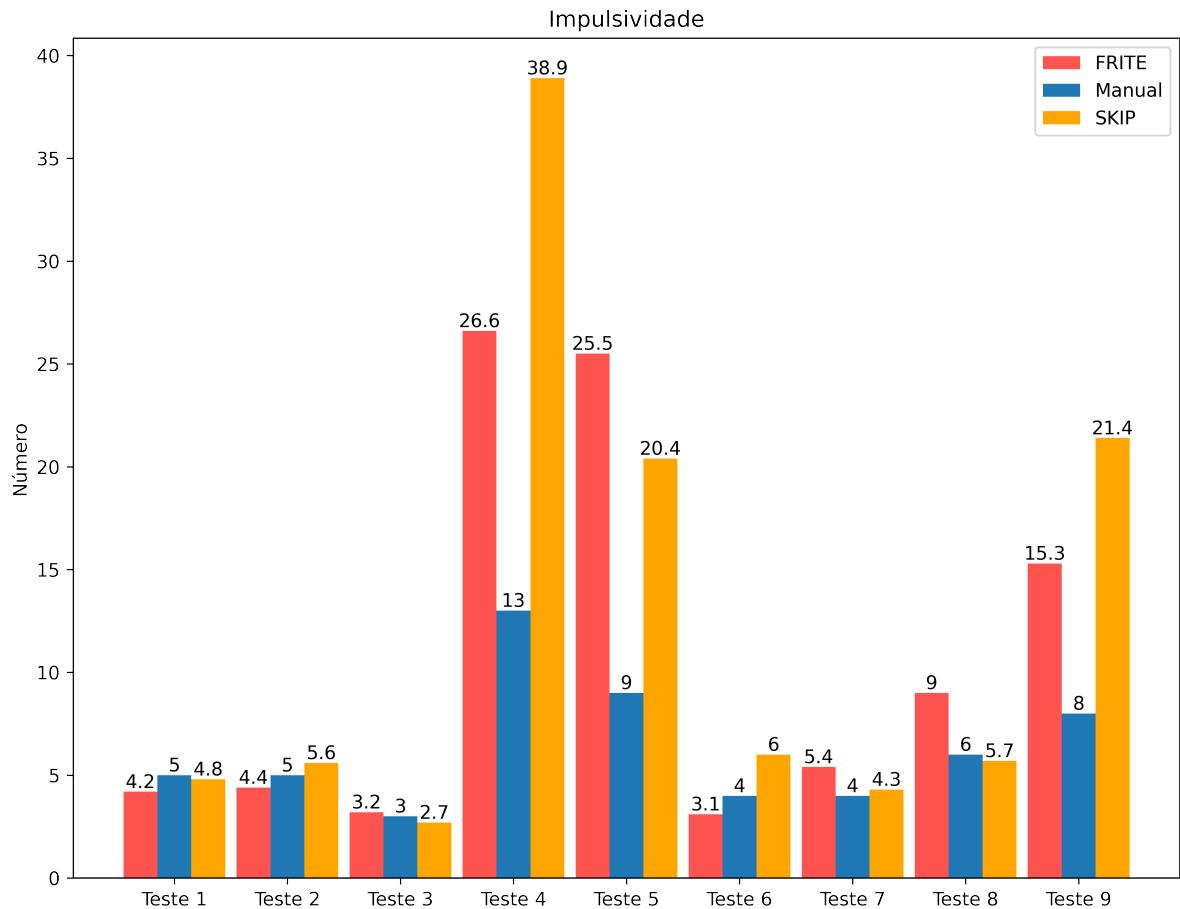
Fonte: Próprio Autor (2021)

Essa divergência se deu pela ligação existente entre os palos como é mostrado na Figura 35. O algoritmo identifica apenas como um único palo existente, acarretando numa incompatibilidade na altura fidedigna e, em função disso, o resultado deste aspecto se torna distinto.

4.3.7 Impulsividade

Neste parâmetro da correção do teste, por intermédio do gráfico abaixo, representado pela Figura 36, constatou-se uma semelhança nos resultados atingidos entre os dois métodos tecnológicos no que se refere a correção manual.

Figura 36 – Gráfico | Impulsividade



Fonte: Próprio Autor (2021)

Pode se observar uma homogeneidade entre os formatos de correção nos primeiros três testes, porém sofreu algumas oscilações nos últimos dois palográficos referentes a análise realizada pelo psicólogo. A obtenção dos resultados ligados à impulsividade, está diretamente relacionada a correção dos tamanhos dos palos.

4.3.8 Considerações Finais

Este estudo de caso investigou os métodos de correção existentes ao teste palográfico, o software SKIP e o método manual, no que se refere a aplicação desenvolvida neste trabalho. Tendo por objetivo principal deste estudo de atingir a avaliação de desempenho descrita na subseção 3.4.1, analisou nove palográficos sendo submetidos a sete parâmetros de verificação.

Como se pode observar no Quadro 5 a aplicação desenvolvida neste trabalho, se encontrou superior em cinco parâmetros (produtividade, rendimento, NOR, soma da diferença e impulsividade) em relação ao seu adversário, o SKIP, com dois atributos

(distância entre palos e tamanho dos palos) ambos sendo comparados com o método manual realizado pela correção de uma psicóloga.

Quadro 5 – Comparação FRITE x SKIP

Parâmetro	FRITE x SKIP
Produtividade	FRITE
Rendimento	FRITE
NOR	FRITE
Soma da Diferença	FRITE
Distância entre Palos	SKIP
Impulsividade	FRITE
Tamanho dos Palos	SKIP

Fonte: Próprio Autor (2021)

Portanto, diante destes resultados destacados neste estudo de caso, a ferramenta FRITE se apresenta como expressiva alternativa bem considerável para a correção dos testes palográficos. Cabe ressaltar que o software SKIP não é uma ferramenta totalmente automatizada, visto que é permitido o psicólogo realizar uma correção no resultado gerado pelo sistema, como identificar manualmente um palo não encontrado.

5 Conclusão

Com o intuito de facilitar a correção para os profissionais da área de psicologia, este trabalho teve como principal objetivo em desenvolver uma ferramenta para análise relacionada ao teste palográfico. Dessa forma, um aplicativo multiplataforma para os dispositivos móveis foi proposto, no qual se fez possível identificar sete aspectos relevantes relacionados à correção do teste através do algoritmo de processamento de imagens produzido. Assim, a implementação da solução se dividiu em três etapas:

- **Algoritmo de Processamento de Imagem.** Foi construído na linguagem Python e utilizando a biblioteca OpenCV, decorrido em cinco etapas: scanner; redimensionamento; separação; ordenação; e análise.
- **API.** Foi estruturado na linguagem Python com o framework Django para atuar como intermédio na comunicação entre o algoritmo e o aplicativo, salvando as informações no banco de dados.
- **Aplicativo.** Foi desenvolvido na linguagem Dart com o framework Flutter para que o usuário pudesse ter uma melhor experiência de usabilidade com a finalidade central deste trabalho.

A fim da validação de objeto proposto, por meio da avaliação de desempenho, foi elaborado um estudo de caso em que realizou um comparativo entre o método tecnológico (SKIP) e também com a correção manual realizado por uma psicóloga. Para isto, nove testes foram submetidos a esta verificação. Desta comparação se tornou capaz de constatar que a correção do aplicativo desenvolvido se aproximou dos resultados do SKIP em alguns aspectos como a distância entre os palos e o tamanho dos palos. O FRITE obteve melhores resultados nos seguintes aspectos: a produtividade, o rendimento, o nível de oscilação rítmica, a soma da diferença e a impulsividade.

Diante das conclusões demonstradas no estudo de caso, a aplicação desenvolvida apresenta de forma satisfatória, por manuseio de um aparelho móvel e sua câmera (sem haver necessidade do uso de scanners como o software SKIP), como um instrumento de alternativa na correção do teste palográfico, que por sua vez atinge os objetivos deste trabalho.

Vale acentuar que nenhum método comparativo reconheceu os palos de modo perfeito como a análise humana e também que a aplicação construída neste trabalho não tem por objetivo de substituir a presença do psicólogo na correção do teste palográfico

mas de fato para atuar em conjunto com este profissional tendo o objetivo de diminuir o tempo do diagnóstico e aumentar a precisão nas correções.

5.1 Trabalhos Futuros

Em sequência são expostas algumas sugestões para trabalhos futuros.

- **Análise dos Demais Aspectos.** A ferramenta proposta neste trabalho apresentou sete aspectos relacionados a correção do teste palográfico. No entanto, existem outros atributos que estão presentes na análise do teste como por exemplo: as margens, os ângulos, a identificação dos ganchos e outros.
- **Algoritmo de Amplificação na Qualidade da Imagem.** Um estudo que pode ser realizado sobre o processamento da imagem é a qualidade da imagem. Uma forma de averiguar este caso é a utilização de redes neurais para aprimorar a resolução da fotografia enviada do teste.
- **Solução da Dupla Detecção.** Por vezes o examinado pode realizar um palo bem próximo ou até mesmo conectado com outro palo, acarretando a detecção de apenas um palo agrupado. Um ponto a ser aprofundado é desenvolver uma solução para este problema na detecção durante o processamento.
- **Evolução dos Aspectos Analisados.** Constatando o resultado obtido no estudo de caso, poderia desenvolver um estudo para melhorar os métodos de análises referentes aos aspectos de distância e de tamanho dos palos em que se mostrou inferior comparado ao SKIP, por ocorrer palos conectados gerando uma obstrução na correção de forma verídica.
- **Teste de Usabilidade.** Realizar teste de usabilidade de modo a coletar dados referente ao uso da ferramenta com um profissional da área da psicologia e observar pontos em que possa contribuir com futuras melhorias no aplicativo como por exemplo a aplicação de técnicas UX/UI no design para otimizar o processo, tornando-o mais intuitivo.

Referências

- ALBUQUERQUE, M. et al. Análise de imagens e visão computacional. *Centro Brasileiro de Pesquisas Físicas. Rio de Janeiro*, 2012.
- ALVES, I. C. B.; ESTEVES, C. *O teste palográfico na avaliação da personalidade*. [S.l.]: Vetor, 2004.
- ANTONELLO, R. Introdução a visão computacional com python e opencv. *Santa Catarina, 1a edição*, 2018.
- ASSIS, M. C. d. Metodologia do trabalho científico. v. 3, 2009.
- BAGGIO, D. L. *OpenCV 3.0 Computer Vision with Java*. [S.l.]: Packt Publishing Ltd, 2015.
- BALDISSERA, R. L. d. S. et al. Desenvolvimento de uma ferramenta para gestão de carreira e vendas voltada ao mercado audiovisual. Florianópolis, SC., 2018.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistemas com UML*. [S.l.]: Elsevier Rio de Janeiro, 2007. v. 2.
- BIANCHI, M. F. d. *Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA*. Tese (Doutorado) — Universidade de São Paulo, 2006.
- BORGES, L. E. *Python para desenvolvedores: aborda Python 3.3*. [S.l.]: Novatec Editora, 2014.
- CARVALHO, M. C. F. et al. Um método para elicitação e modelagem de requisitos baseado em objetivos. Universidade Federal do Maranhão, 2002.
- CASTRO, A. A. M. D.; PRADO, P. P. L. D. Algoritmos para reconhecimento de padrões. *Revista Ciências Exatas*, v. 8, n. 2002, 2002.
- CERQUEIRA, M. V. B.
Implementação de Detecção de Placas Veiculares Utilizando OpenCV — Universidade Federal do Recôncavo da Bahia, 2016.
- CODRUT, S. *Safe and Efficient Hybrid Memory Management for Java*. Tese (Doutorado) — UNIVERSITY OF CALIFORNIA, IRVINE, 2015.
- CONCEIÇÃO, A. R. d. A.
Desenvolvimento de um Aplicativo para Auxílio na Correção do Teste Psicológico Palográfico — Faculdade Católica do Tocantins (FACTO), 2019.
- CORRÊA, E. *Meu Primeiro Livro de Python*. [S.l.: s.n.], 2020. v. 2.
- ERPEN, L. R. C. Reconhecimento de padrões em imagens por descritores de forma. 2004.
- FACON, J. *Processamento e análise de imagens*. [S.l.]: Universidad Nacional de Cordoba, 1993.

- FARIAS, S. L. O. Gerenciamento por processos em uma instituição de ensino universitário: o caso do departamento de controle e registro acadêmico. 2009.
- FENTAW, A. E.
- Cross platform mobile application development: a comparison study of React Native vs Flutter* — University of Jyvaskyla, 2020.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999.
- FONSECA, J. Metodologia da pesquisa científica. 2002.
- FOWLER, M. *UML distilled: a brief guide to the standard object modeling language*. [S.l.]: Addison-Wesley Professional, 2004.
- GALLO, V. Z. M. Desenvolvimento de um sistema de reconhecimento de caracteres para calibração de trenas a laser. 2020.
- GERHARDT, T. E. Silveira, dt (org.). *Métodos de pesquisa*, v. 1, 2009.
- GIL, A. Como elaborar projetos de pesquisa. 6 [degrees] ed. São Paulo: Editoras Atlas, 2019.
- GOMES, D. d. F. *Visão computacional e segmentação de imagens por discriminação de textura*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Blucher, 2000.
- GUEDES, G. T. *UML 2-Uma abordagem prática*. [S.l.]: Novatec Editora, 2018.
- HALLBERG, S. C. M. *Percepção e uso de tecnologias da informação e comunicação por psicoterapeutas*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2015.
- JAIN, A. K. *Fundamentals of digital image processing*. [S.l.]: Prentice-Hall, Inc., 1989.
- LEMOS, M. De que forma a psicologia vem acompanhando as mudanças trazidas pela tecnologia. Psicologias do Brasil, 2015.
- LIBERMAN, F. Classificação de imagens digitais por textura usando redes neurais. 1997.
- LIMONGE, N. Testes informatizados – uma nova visão. *Psicologia & Informática*, 2006.
- MARCONI, M. d. A.; LAKATOS, E. Fundamentos da metodologia científica. 7^a edição-são paulo: Atlas. 2010.
- MARENGONI, M.; STRINGHINI, S. Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada*, v. 16, n. 1, p. 125–160, 2009.
- MEIRELLES, F. S. Pesquisa anual do fgvcia. *Uso da TI-Tecnologia da Informação nas Empresas*. Fundação Getúlio Vargas, 2020.
- MINAYO, M. Ciência, técnica e arte: o desafio da pesquisa social. in mynayo, mcs (org.). *Pesquisa Social: Teoria, Método e criatividade*. Petrópolis, RJ: Vozes, p. 51–66, 2001.

- MISBHAUDDIN, M.; ALSHAYEB, M. Extending the uml use case metamodel with behavioral information to facilitate model analysis and interchange. *Software & Systems Modeling*, Springer, v. 14, n. 2, p. 813–838, 2015.
- MURTY, M. N.; DEVI, V. S. *Introduction to pattern recognition and machine learning*. [S.l.]: World Scientific, 2015. v. 5.
- NOGUEIRA, D. L. *Ferramentas Automatizadas para Apoio ao Projeto Estruturado: uma aplicação do diagrama entidade-relacionamento*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 1988.
- PONTES, B. R. Avaliação de desempenho: nova abordagem. LTr, 1999.
- QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. *Rita*, v. 13, n. 2, p. 11–42, 2006.
- RUMBAUGH, J. et al. Uml: guia do usuário. *Rio de Janeiro: Campus*, 2000.
- SANTOS, R. A. S. Pytm - python task manager. desenvolvimento de um software para automação de tarefas utilizando python. 2011.
- SILVA, G. G. da et al. Veículos aéreos não tripulados com visão computacional na agricultura: aplicações, desafios e perspectivas. *ANAIIS-ENCONTRO CIENTÍFICO DE ADMINISTRAÇÃO, ECONOMIA E CONTABILIDADE*, v. 1, n. 1, 2014.
- SILVA, R. C. d. Docker versus kvm: Uma análise de desempenho de disco para pequenos arquivos. 2016.
- SOMMERVILLE, I. *Software engineering*. [S.l.]: Addison-wesley, 2011.
- SONKA, M. et al. *Image processing, analysis, and machine vision*. [S.l.]: Cengage Learning, 2014.
- SOUZA, L. R. d. Algoritmo para reconhecimento e acompanhamento de trajetória de padrões em imagens móveis. 2010.
- TRENTINI, V. B. et al. Reconhecimento automático de placas de veículos. *Anais do WVC*, p. 267–272, 2010.
- VERNON, P. E. et al. *Studies in Expressive Movement, by Gordon W. Allport... and Philip E. Vernon... With a Chapter on Matching Sketches of Personality with Script, by Edwin Powers...* [S.l.]: MacMillan, 1933.
- WHELAN, P. F.; MOLLOY, D. *Machine vision algorithms in Java: techniques and implementation*. [S.l.]: Springer Science & Business Media, 2001.
- WINDMILL, E. *Flutter in action*. [S.l.]: Simon and Schuster, 2020.
- XAVIER, L. *Integração de Requisitos não Funcionais a Processos de Negócios: Integrando BPMN e NFR*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009.

A Requisitos Funcionais

Tabela 1 – RF001 | Cadastrar Avaliador

RF001	Cadastrar avaliador
Referência	-
Definição	A aplicação deve permitir que o psicólogo possa se cadastrar no sistema.
Sumário	O caso de uso é responsável por cadastrar o avaliador no sistema.
Pré-condições	O usuário deve estar com o aplicativo instalado em seu dispositivo móvel.
Atores	Avaliador
Descrição	<ol style="list-style-type: none"> Na tela inicial do sistema, o usuário deve clicar no botão de Cadastrar Avaliador. Na tela seguinte, o usuário deve informar os atributos requeridos e clicar no botão de Salvar.
Alternativas	-
Exceção	-

Fonte: Próprio Autor (2021)

Tabela 2 – RF002 | Realizar Login

RF002	Realizar login
Referência	RF001
Definição	A aplicação deve permitir que o avaliador possa acessar o sistema.
Sumário	O caso de uso é responsável por realizar o login do usuário no sistema.
Pré-condições	O usuário deve estar previamente cadastrado no sistema para poder realizar o acesso.
Atores	Avaliador
Descrição	<ol style="list-style-type: none"> Na tela inicial do sistema, o usuário deve informar seu email e senha. Após isso, o usuário deve clicar no botão de Entrar.
Alternativas	O usuário realizar seu cadastro para poder acessar o sistema.
Exceção	A funcionalidade de login não poderá ser realizada se o sistema não possuir o respectivo usuário cadastrado.

Fonte: Próprio Autor (2021)

Tabela 3 – RF003 | Cadastrar Examinado

RF003	Cadastrar examinado
Referência	RF001, RF002
Definição	A aplicação deve permitir que o avaliador possa cadastrar seus examinados no sistema.
Sumário	O caso de uso é responsável por cadastrar o examinado no sistema.
Pré-condições	O usuário deve estar logado no sistema para ter acesso a todas funcionalidades.
Atores	Avaliador
Descrição	<p>1. Na tela inicial do sistema, o usuário deve informar seu email e senha.</p> <p>2. Após isso, o usuário deve clicar no botão de Entrar.</p> <p>3. Feito isso, o usuário deve clicar no botão de Cadastrar Examinado.</p> <p>4. O usuário deve informar todos os atributos requeridos e clicar no botão de Salvar.</p>
Alternativas	-
Exceção	O cadastro do examinado não pode ser completado com atributos não informados pelo usuário.

Fonte: Próprio Autor (2021)

Tabela 4 – RF004 | Cadastrar Teste Palográfico

RF004	Cadastrar teste palográfico
Referência	RF003
Definição	A aplicação deve permitir que o avaliador possa cadastrar o teste palográfico do examinado.
Sumário	O caso de uso é responsável por cadastrar o teste palográfico no sistema.
Pré-condições	O usuário deve estar logado no sistema e ter cadastrado o examinado previamente.
Atores	Avaliador
Descrição	<p>1. Na tela inicial do sistema, o usuário deve informar seu email e senha.</p> <p>2. Após isso, o usuário deve clicar no botão de Entrar.</p> <p>3. Feito isso, o usuário deve clicar no botão de Cadastrar Teste Palográfico.</p> <p>4. O usuário deve informar todos os atributos requeridos e clicar no botão de Salvar.</p>
Alternativas	-
Exceção	O cadastro do teste palográfico não pode ser completado com atributos não informados pelo usuário.

Fonte: Próprio Autor (2021)

Tabela 5 – RF005 | Permitir Upload de Imagem

RF005	Permitir upload de imagem
Referência	RF004
Definição	A aplicação deve permitir que o avaliador possa enviar a imagem capturada do teste para o sistema.
Sumário	O caso de uso é responsável pela permissão de upload da imagem do teste palográfico.
Pré-condições	O usuário deve estar logado no sistema e estar na tela do cadastro de teste palográfico.
Atores	Avaliador
Descrição	<p>1. Após ter informado os atributos requeridos no formulário de cadastro, o usuário deve selecionar a imagem referente ao teste palográfico.</p> <p>2. Depois, o usuário deve clicar no botão de Salvar.</p>
Alternativas	No passo 1, permitir que o usuário possa capturar a imagem do teste palográfico através do aplicativo.
Exceção	O cadastro completo do teste palográfico não pode ser completado sem que a imagem não for enviada.

Fonte: Próprio Autor (2021)

Tabela 6 – RF006 | Realizar Pré-processamento da Imagem

RF006	Realizar pré-processamento da imagem
Referência	RF005
Definição	O sistema deve preparar o processamento da imagem enviada para análise.
Sumário	O caso de uso é responsável pela realização do pré-processamento da imagem.
Pré-condições	O usuário deverá ter submetido a imagem do teste palográfico para análise.
Atores	Sistema
Descrição	<p>1. O sistema recebe o formulário com a imagem enviada do teste palográfico.</p> <p>2. O sistema irá classificar a imagem como apta ou inapta para a realização do tratamento.</p> <p>3. O sistema inicia o processamento utilizando as técnicas de tratamento da imagem.</p>
Alternativas	-
Exceção	Caso o sistema qualificar a imagem capturada como inapta, deve retornar um alerta ao avaliador para realizar o upload novamente de uma outra imagem.

Fonte: Próprio Autor (2021)

Tabela 7 – RF007 | Identificar e Realizar Análises nos Palos

RF007	Identificar e realizar análises nos palos
Referência	RF006
Definição	O sistema deve reconhecer os palos e produzir informações baseadas na imagem.
Sumário	O caso de uso responsável pela identificação e realização das análises nos palos.
Pré-condições	O sistema deve ter classificado a imagem como apta no RF006 para identificação e análise dos palos.
Atores	Sistema
Descrição	<p>1. Após o RF006, o sistema irá efetuar a identificação de todos os palos.</p> <p>2. A partir do passo anterior, o sistema realizará as análises pertinentes aos palos gerando as informações necessárias.</p>
Alternativas	-
Exceção	-

Fonte: Próprio Autor (2021)

Tabela 8 – RF008 | Retornar Relatório com as Informações Obtidas

RF008	Retornar relatório com as informações obtidas
Referência	RF007
Definição	O sistema deve retornar o relatório de visualização no aplicativo para o avaliador contendo as informações contidas no teste palográfico realizado pelo examinado.
Sumário	O caso de uso responsável pela produção do relatório com os dados obtidos.
Pré-condições	O sistema deve ter finalizado o processo das análises do palos.
Atores	Sistema
Descrição	<p>1. O relatório será produzido com as informações obtidas das análises.</p> <p>2. O relatório será visível para o usuário no aplicativo.</p>
Alternativas	-
Exceção	-

Fonte: Próprio Autor (2021)