

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**IRIS: UM APLICATIVO PARA PROCESSAMENTO DE
IMAGENS PARA DALTÔNICOS**

JEAN PEREIRA

**BLUMENAU
2017**

JEAN PEREIRA

**IRIS: UM APLICATIVO PARA PROCESSAMENTO DE
IMAGENS PARA DALTÔNICOS**

Trabalho de Conclusão de Curso apresentado
ao curso de graduação em Ciência da
Computação do Centro de Ciências Exatas e
Naturais da Universidade Regional de
Blumenau como requisito parcial para a
obtenção do grau de Bacharel em Ciência da
Computação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU
2017**

IRIS: UM APLICATIVO PARA PROCESSAMENTO DE IMAGENS PARA DALTÔNICOS

Por

JEAN PEREIRA

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: Prof. Alexander Roberto Valdameri, Mestre – FURB

Membro: Prof. Dalton Solano dos Reis, Mestre – FURB

Blumenau, 03 de julho de 2017

Dedico este trabalho a todas as pessoas com
alguma deficiência visual.

AGRADECIMENTOS

À minha família, por tudo.

Aos meus amigos, pelas conversas, conselhos, tristezas e risadas.

Ao Spotify, por disponibilizar a trilha sonora deste trabalho.

Aos professores do curso de Ciência da Computação da Universidade Regional de Blumenau, por todo o conhecimento compartilhado durante o curso.

Ao meu orientador Aurélio Hoppe, pelo tema e ajuda no desenvolvimento deste trabalho.

“Machines take me by surprise with great frequency”

A. M. Turing

RESUMO

Aproximadamente 8% dos homens e 1% das mulheres possuem algum problema na visão a cores, e dada a importância da cor na comunicação e informação, essas pessoas podem enfrentar problemas em tarefas de seu dia a dia. Este trabalho apresenta o desenvolvimento de um aplicativo para a plataforma Android que realiza o processamento de imagens para daltônicos do tipo protanopia, deuteranopia e tritanopia, buscando facilitar a visualização de informações que eles normalmente possuem dificuldade. O aplicativo faz uso do algoritmo Daltonize para melhorar a qualidade visual de imagens digitais capturadas através de dispositivos móveis tais como fotos, fotos da galeria e vídeos em tempo real. Após a implementação foram realizados testes com voluntários que utilizaram o aplicativo e responderam um questionário de usabilidade e aceitação. Os resultados se mostraram positivos, porém, demonstraram também pontos que devem ser melhorados, principalmente na funcionalidade de tempo real. Por fim, foram sugeridas algumas extensões que buscam melhorar e expandir as funcionalidades do aplicativo criado.

Palavras-chave: Daltonismo. Diferenciação de cores. Processamento de imagens. Android.

ABSTRACT

Around 8% of men and 1% of women have some kind of problem with color vision, and given the importance of color in communication and information, these people may experience problems in day-to-day tasks. This work presents the development of an Android application that performs images processing for color blind people such as protanomaly, tritanomaly and deuteranomaly, aiming the correct visualization of information that usually has some loss due it's deficiency. The application makes use of the Daltonize algorithm to improve the visual quality of digital images captured through mobile devices such as photos, gallery photos and real-time videos. After the development, the app was tested with volunteers that also answered a usability and acceptance questionnaire. The results were positive, but also indicated that the app needs some improvements, mainly on real-time functionality. Finally, we have suggested some extensions that seek to improve and expand the functionality of the application created.

Key-words: Color blindness. Color differentiation. Image processing. Android.

LISTA DE FIGURAS

Figura 1 - Anatomia do olho humano	15
Figura 2 - Estrutura da retina	16
Figura 3 - Espectro de luz	16
Figura 4 - Simulação da visão de um portador de tricromatismo anômalo	17
Figura 5 - Simulação da visão de um portador de dicromatismo	18
Figura 6 - Simulação da visão de um portador de monocromatismo	18
Figura 7 - Quadros do teste de Ishihara	19
Figura 8 - Teste de Farnsworth-Munsell.....	19
Figura 9 - Comparação entre a perda de informações da imagem.....	21
Figura 10 - Resultado da simulação.....	23
Figura 11 - Resultado do filtro.....	23
Figura 12 - Diagrama de casos de uso	27
Figura 13 - Diagrama de Classes	28
Figura 14 - Fluxo de aquisição e processamento de imagens.....	30
Figura 15 - Seleção de daltonismo e opções de aquisição de imagem	30
Figura 16 - Opções de aquisição de imagem	31
Figura 17 - Funcionalidade de tempo real	32
Figura 18 - Seleção de qualidade da imagem	33
Figura 19 - Fluxo do algoritmo.....	34
Figura 20 - Comparação entre a imagem original e a convertida	36
Figura 21 - perda entre a visão normal e do daltônico.....	37
Figura 22 - Imagem gerada pelo algoritmo.....	38

LISTA DE QUADROS

Quadro 1 - Psuedocódigo do algoritmo Daltonize.....	20
Quadro 2 - Critérios utilizados.....	24
Quadro 3 - Aplicativos iOS analisados	24
Quadro 4 - Aplicativos Android analisados.....	25
Quadro 5 - Rastreabilidade dos Requisitos Funcionais	26
Quadro 6 - Requisitos Não Funcionais	26
Quadro 7 - Configuração da Intent de uso da câmera.....	31
Quadro 8 - Inicialização da camera do dispositivo	32
Quadro 9 - Execução do <i>callback</i> do <i>frame</i> da câmera	33
Quadro 10 - Extração do RGB de um pixel.....	34
Quadro 11 - Conversão RGB para LMS.....	35
Quadro 12 - Matrizes de transformação.....	35
Quadro 13 - Transformação de espaço de cores	35
Quadro 14 - Conversão LMS para RGB.....	36
Quadro 15 - Cálculo da matriz de erro.....	37
Quadro 16 - Cálculo da cor final.....	37
Quadro 20 - Comparação com os trabalhos correlatos	41
Quadro 21 - Questionário de perfil do usuário	48
Quadro 22 - Lista de tarefas do usuário	49
Quadro 23 - Questionário de avaliação do aplicativo.....	50

LISTA DE TABELAS

Tabela 1 - Resultado da avaliação dos voluntários.....	22
Tabela 2 - Resultados da execução de tarefas.....	39
Tabela 3 - Avaliação da usabilidade do aplicativo	40
Tabela 4 - Avaliação das funcionalidades do aplicativo.....	41

LISTA DE ABREVIATURAS E SIGLAS

ADT – Android Development Tools

API – Application Programming Interface

LMS - Longwave, Middlewave, Shortwave

MHz – Megahertz

OpenCV – Open Source Vision Library

PIL – Python Imaging Library

RAM – Random Access Memory

RGB – Red Green Blue

SDK – Software Development Kit

UML – Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS	14
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 SISTEMA VISUAL HUMANO.....	15
2.2 DALTONISMO	17
2.3 CORREÇÕES DE IMAGENS PARA DALTONICOS	19
2.4 TRABALHOS CORRELATOS	21
2.4.1 Uma ferramenta adaptativa para facilitar a visualização de imagens para pessoas portadoras de daltonismo.....	21
2.4.2 Ferramenta de Acessibilidade para Deficientes Visuais em Cores	23
2.4.3 Proposta de um aplicativo móvel Open Source em auxílio a indivíduos com Discromopsia baseado em um estudo qualitativo	24
3 DESENVOLVIMENTO	26
3.1 REQUISITOS	26
3.2 ESPECIFICAÇÃO.....	26
3.2.1 Diagrama de Casos de Uso.....	27
3.2.2 Diagrama de Classes.....	28
3.3 IMPLEMENTAÇÃO	29
3.3.1 Técnicas e ferramentas utilizadas	29
3.4 ANÁLISE DOS RESULTADOS	39
3.4.1 Análise do perfil e resultados da lista de tarefas	39
3.4.2 Análise da usabilidade e funcionalidades do aplicativo	40
3.4.3 Comparativo entre o aplicativo desenvolvido e seus correlatos.....	41
4 CONCLUSÕES	43
4.1 EXTENSÕES.....	43
APÊNDICE A – QUESTIONÁRIOS E LISTA DE TAREFAS	48

1 INTRODUÇÃO

Desde o nascimento até os 12 anos de idade o olho cresce e se desenvolve por conta dos estímulos visuais (luz e formas) que nos cerca (FERNANDES, 2012). Sendo a visão considerada o principal meio de aprendizado e informação, os problemas de visão como a miopia, hipermetropia, astigmatismo ou outras carências podem acarretar em consequências no desenvolvimento cognitivo e desempenho de atividades como locomoção e comunicação (MAIA; SPINILLO, 2012).

Cerca de 40% das informações recebidas pela visão humana são referentes a cor e além disso são parte fundamental de planos comunicacionais podendo abranger desde a qualificação de um objeto como no controle do fluxo do trânsito, na legenda de mapas, etc. (MAIA; SPINILLO, 2012). Dada a importância da cor como parte fundamental da comunicação, é esperado que pessoas com defeitos na visão cromática como o daltonismo sofram com implicações socioculturais (MELO; GALON; FONTANELLA, 2014).

Tsuda (2013, p. 1) definem o daltonismo como “uma deficiência visual que resulta em dificuldade ou impossibilidade de distinguir todas ou algumas cores”. Essa deficiência, segundo Tsuda (2013, p. 1) “é provocada por uma falha ou ausência funcional de algumas das células sensoriais, localizadas na retina, capazes de distinguir as cores, os cones”. Há três tipos conhecidos de daltonismo: o monocromatismo que é quando se percebe luz em apenas um dos cones, o dicromatismo que se refere a quando não se percebe luz em um dos cones e a tricromatismo anômalo que é quando a pessoa percebe luz nos três cones porém de forma ligeiramente reduzida (MANDAL, 2014).

Pessoas daltônicas enfrentam dificuldades no dia a dia que normalmente não afeta pessoas sem esse tipo de problema. Esses problemas são diversos como saber se uma fruta está madura, pintar um desenho e escolher uma roupa e/ou praticar esportes. Estima-se que aproximadamente um entre doze homens (8%) e uma em duzentas mulheres (0.5%) no mundo possuem algum tipo de daltonismo (COLOUR BLIND AWARENESS, 2012?).

É de conhecimento que não há cura para o daltonismo. Porém, cientistas sabem que filtros facilitam a visualização e, a partir disso, nos últimos anos com o avanço da tecnologia surgiram algoritmos, aplicações e produtos comerciais projetados e desenvolvidos para ajudar pessoas com daltonismo. Geralmente são objetos como óculos ou lentes, ou software disponível em algum dispositivo portátil (SHEPEARD, 2015; TANUWIDJAJA et al., 2014).

Com a popularização das tecnologias móveis e dos smartphones é normal que a quantidade de informação que recebemos através deles sejam cada vez maiores e que se

tornem parte do nosso dia a dia (MANTOVANI; DANTAS, 2011). Segundo CISCO (2016, p. 1) é esperado que 70% da população mundial usará algum dispositivo móvel em 2020. Ainda segundo o autor:

A proliferação dos telefones móveis, incluindo os chamados "phablets" (uma mistura híbrida de recursos do smartphone e tablet) está aumentando com tanta rapidez que haverá mais pessoas com telefones celulares (5,4 bilhões) do que eletricidade (5,3 bilhões), água encanada (3,5 bilhões) e automóveis (2,8 bilhões), em 2020. (CISCO, 2016, p. 1).

Os smartphones também possuem grande potencial na medicina e estão começando a oferecer um competidor de baixo custo para diagnósticos médicos caros, principalmente em lugares que acesso a fundos, profissionais e equipamentos é escarço. Um bom exemplo é o uso desses aparelhos nos tratamentos oculares, como por exemplo o NETRA, que utiliza de um acessório para smartphone que auxilia a identificação de ametropias e se demonstra tão efetivo quanto os métodos tradicionais e com isso abre também um novo horizonte em outras áreas médicas (LAKSHMINARAYANAN; ZELEK; MCBRIDE, 2015).

Diante deste cenário, este trabalho apresenta o desenvolvimento de um aplicativo capaz de facilitar a visualização de imagens para pessoas portadoras de daltonismo.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um aplicativo que melhore a qualidade visual de imagens digitais para pessoas portadoras de daltonismo.

Os objetivos específicos do trabalho são:

- a) disponibilizar um mecanismo para que o usuário possa identificar e visualizar o realce das imagens de acordo com o seu tipo de daltonismo;
- b) disponibilizar um mecanismo que amplifique a diferença de tonalidade e brilho das cores existentes em imagens capturadas via smartphones.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos: introdução, fundamentação teórica, desenvolvimento e conclusões. O segundo capítulo apresenta a fundamentação teórica, onde será abordado o sistema visual humano, sua estrutura e a visão de cores, além do daltonismo, seus diferentes tipos e diagnóstico e também a correção de imagens para daltônicos. No terceiro capítulo é demonstrada a especificação do aplicativo e o detalhamento da implementação. Por fim, no quarto capítulo são descritas as conclusões, os resultados obtidos e algumas extensões.

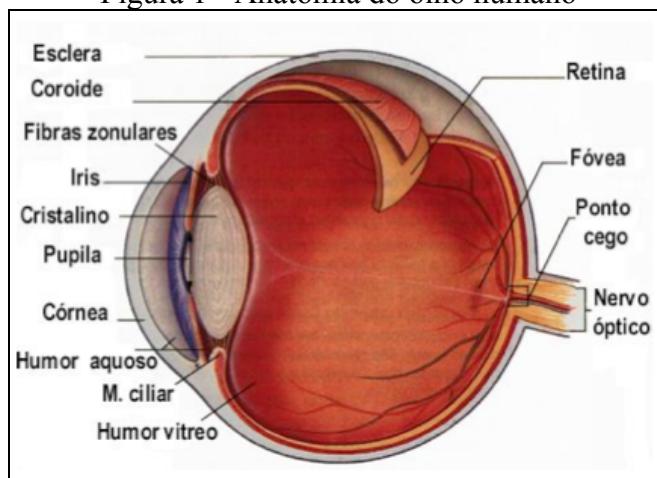
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em quatro seções. A seção 2.1 aborda o sistema visual humano, sua estrutura e a visão de cores. A seção 2.2 descreve o daltonismo, seus diferentes tipos e diagnóstico. Na seção 2.3 é abordada a correção de imagens para daltônicos e, por fim, a seção 2.4 apresenta os trabalhos correlatos.

2.1 SISTEMA VISUAL HUMANO

A visão humana é um processo complexo que envolve a interação dos olhos com o cérebro através de uma rede de neurônios, receptores e outras células. O olho humano é composto de uma variedade de componentes óticos que incluem a córnea, íris, pupila, lente e a retina que juntos trabalham para formar imagens a partir do campo de visão de cada olho (BRINDLEY, 1957). A Figura 1 apresenta a anatomia do olho humano.

Figura 1 - Anatomia do olho humano

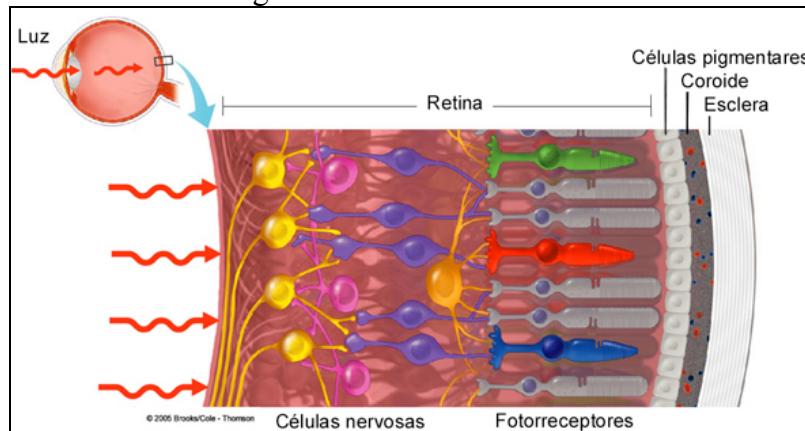


Fonte: Nishida (2012, p. 86).

A córnea é uma superfície curva e transparente que fica em frente ao olho e por onde a luz entra, é o elemento mais significante para formação de imagens do sistema visual. Já a íris é um músculo que controla a quantidade de luz que chegara a pupila, controlando assim seu tamanho e a quantidade de luz que irá para o interior do globo até a retina. Sua pigmentação é que define a cor de nossos olhos e o seu tamanho depende basicamente da iluminação do ambiente. A lente é uma estrutura de tamanho variável, quando olhamos para um objeto mais próximo ela fica mais gorda aumentando o poder de foco no objeto, da mesma forma quando enxergamos um objeto distante ela fica mais fina aumentando o foco no objeto distante. A retina é uma camada interna que possui fotorreceptores e células nervosas e toda imagem ótica gerada pelo olho é projetada na retina. Existem dois tipos de fotorreceptores, células

responsáveis pela detecção dos estímulos luminosos, os cones e bastonetes (BRINDLEY, 1957; NISHIDA, 2012). A Figura 2 ilustra a estrutura da retina.

Figura 2 - Estrutura da retina

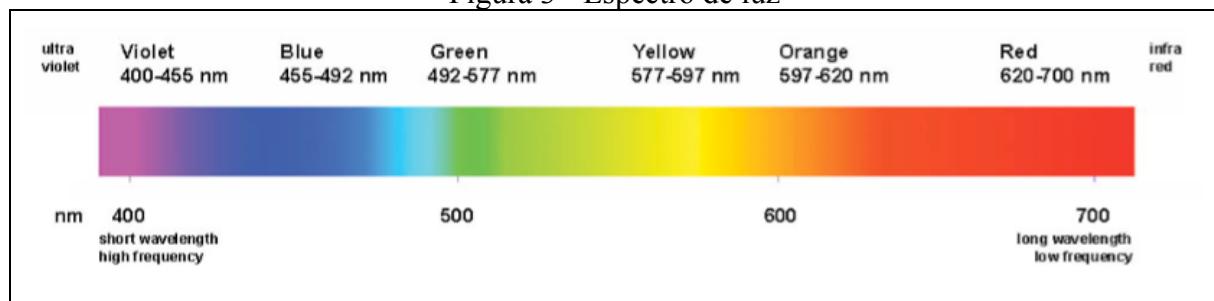


Fonte: Nishida, Oliveira e Troll (2010?).

Segundo Nishida (2012, p. 91) “os cones e bastonetes são células sensoriais fotossensíveis e responsáveis pela transdução foto-elétrica”. Os bastonetes são mais sensíveis a mudanças de luz, traços e movimentos e possuem apenas um tipo de pigmento sensível a luz, além de não serem bons com cores. Existem aproximadamente 120 milhões de bastonetes na retina humana. Os cones são aproximadamente 7 milhões e são até mil vezes menos sensíveis a luz comparados aos bastonetes, porém, são sensíveis a tons de cores (verde, vermelho ou azul), é por isso que você não enxerga cores muito bem em lugares escuros. Sinais dos cones são enviados ao cérebro que traduz esses sinais e os transforma na percepção de cores (BRINDLEY, 1957; NISHIDA, 2012).

Os receptores de nossos olhos recebem e processam energia eletromagnética de uma largura muito estreita do espectro de ondas, aproximadamente entre 380 e 750nm. As ondas dentro desse intervalo e suas diferentes misturas são chamadas de luz e luz é o estímulo principal para a visão a cores (BRINDLEY, 1957). Na Figura 3 são apresentadas as cores do espectro de luz conforme o seu comprimento.

Figura 3 - Espectro de luz



Fonte: Nishimori (2013).

Uma cor é determinada por três fatores: matiz (depende do comprimento da onda; o espectro da luz visível corresponde às matizes que o nosso olho enxerga), saturação (pureza relativa da luz, ou seja, se um objeto nos parece branco é porque reflete todas as matizes da luz) e brilho (intensidade da luz) (NISHIDA, 2012).

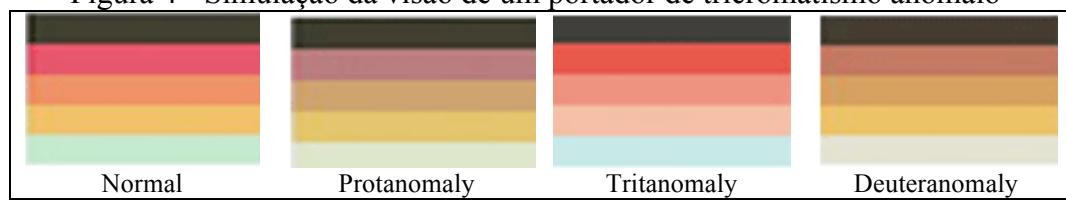
Já a visão de cores é um fenômeno muito mais complexo que envolve células fotossensíveis especiais que ficam na retina, os cones (MELO; GALON; FONTANELLA, 2014). Há três tipos de cones: os que tem pico de sensibilidade à luz na faixa dos 419nm (azuis); 531nm (verdes) e 559nm (vermelhos), ou seja, cada um é mais sensível a determinado comprimento de onda (ou cores) da luz. A metade dos cones é de verde e vermelho e apenas 8% são azuis. (NISHIDA, 2012). Segundo Melo, Galon e Fontanella (2014, p. 1230), “cada um contém um tipo específico de fotopsina: vermelha, verde ou azul. A fotopsina é a proteína responsável por converter o sinal luminoso em sinal elétrico, que é conduzido pelo nervo óptico até o córtex cerebral, onde a visão cromática é interpretada”. Pessoas com algum tipo de deficiência nesse processo apresentarão problemas ou até mesmo total perda na habilidade de distinguir algumas cores (MELO; GALON; FONTANELLA, 2014).

2.2 DALTONISMO

Segundo Araya *et al.* (2014, p. 5) “aproximadamente 10% dos homens e 1% das mulheres apresentam algum grau de deficiência na avaliação das cores”. Esses defeitos de visão cromática ocorrem quando há algum problema nos genes codificadores das fotospina e são classificados em 3 tipos:

- tricromatismo anômalo: nesse tipo, os três tipos de cones são funcionais porém em um a percepção de luz é ligeiramente menor (MANDAL, 2014). Ele se manifesta por meio da redução da sensibilidade ao vermelho (protanomalia) ou a redução da sensibilidade ao azul (tritanomalia) ou ainda na forma mais comum, a redução da sensibilidade ao verde (deuteranomalia) (COLOUR BLIND AWARENESS, 2012?; BRUNI; CRUZ, 2006). A Figura 4 apresenta uma simulação da visão de um portador com tricromatismo;

Figura 4 - Simulação da visão de um portador de tricromatismo anômalo

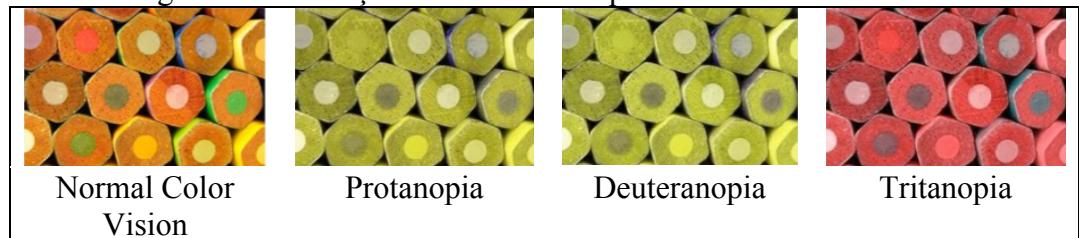


Fonte: New Health Advisor (2014).

- dicromatismo: quando não se percebe luz em um dos cones: manifesta-se através

da protanopia (ausência de cones vermelhos) ou deuteranopia (ausência dos cones verdes) ou ainda tritanopia (ausência de cones azuis) (MELO; GALON; FONTANELLA, 2014). A Figura 5 apresenta uma simulação da visão de um portador com dicromatismo;

Figura 5 - Simulação da visão de um portador de dicromatismo



Fonte: Hermelin (2010).

- c) monocromatismo: quando se percebe luz em apenas um dos cones. Existem dois tipos de monocromatismo. O típico onde há o monocromatismo dos bastões, e o atípico onde há o monocromatismo dos cones (BRUNI; CRUZ, 2006). A Figura 6 apresenta uma simulação da visão de um portador com monocromatismo.

Figura 6 - Simulação da visão de um portador de monocromatismo

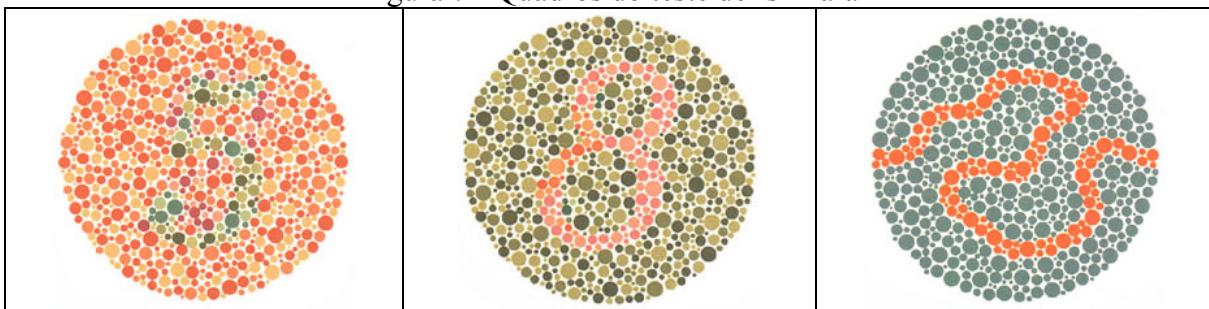


Fonte: Gouras (2009).

Ainda segundo Bruni e Cruz (2006, p. 768) “por convenção, os prefixos gregos para primeiro, segundo e terceiro "protos", "deuteros" e "tritos" são usados para determinar as cores vermelha, verde e azul, respectivamente, ao se nomearem os defeitos de visão de cores”.

Para o diagnóstico há vários testes disponíveis, alguns deles são muito rápidos e fazem ser possível detectar o deficiente visual em minutos, porém, outros necessitam de muito mais tempo. Um dos mais conhecidos é o teste de Ishihara, que consiste em quadros desenhados com combinações de cores que dificultam determinados daltônicos a enxergarem corretamente. Esses quadros são expostos em um ambiente com iluminação controlada e então a pessoa é questionada a respeito do que enxerga no quadro (BRINDLEY, 1957, p. 32), conforme pode ser visto na Figura 7.

Figura 7 - Quadros do teste de Ishihara



Fonte: Colblindor (2017).

Outro teste conhecido é o de Farnsworth-Munsell 100, criado na década de 40 desde então se tornou altamente utilizado para detectar problemas de distinção de cores, inclusive por médicos e cientistas. O teste consiste de 4 grupos de 25 peças de cada cor que devem ser ordenados em uma espécie de progresso de tom (BRINDLEY, 1957, p. 33), conforme mostra a Figura 8.

Figura 8 - Teste de Farnsworth-Munsell



Fonte: Colormax (2017).

É um teste fácil de realizar e um método altamente eficaz para avaliar a capacidade do indivíduo de distinguir cores. Pessoas com alguma falha na visão de cores irão cometer vários erros ao ordenarem essas peças e então com isso pode-se também diagnosticar o tipo de deficiência e a sua severidade (KINNEAR; SAHRAIE, 2002).

2.3 CORREÇÕES DE IMAGENS PARA DALTONICOS

Do ponto de vista médico pode-se dizer que não há cura para o daltonismo, alguns dispositivos (lentes coloridas ou óculos com filtros) parecem melhorar a visão dessas pessoas. Porém, os resultados ainda são muito dispersos. Do ponto de vista computacional, pode-se utilizar algoritmos de processamento de imagem para tentar melhorar a situação (DOUGHERTY; WADE, 2000?).

Diversas técnicas já foram propostas para modificar imagens a cores com o objetivo de melhorar a percepção das cores e reduzir a confusão de pessoas com daltonismo (DOLIOTIS

et al., 2009). Essa transformação de cores significa transformar o espaço de cores de uma imagem em um espaço de cores diferente, incluindo intensidade, contraste e brilho e que pode ser distinguido por um indivíduo com deficiência na visão a cores, conforme seu tipo de deficiência (OTHMAN; SABUDIN, 2013).

Uma das técnicas conhecida como Daltonize propõe a melhora das cores nas imagens para os daltônicos do tipo dicromatismo, primeiramente é realizada uma transformação de cores da imagem de RGB para LMS – que corresponde ao que enxergamos na retina. Após a conversão, realiza a transformação dessa cor para o espectro do daltônico, transforma-a novamente para RGB. Com a cor novamente em RGB, calcula-se a diferença entre o espectro do daltônico e da imagem original, extraíndo assim quais partes são perdidas, por exemplo, se o tipo de daltonismo é protanopia, essa matriz será basicamente de cores vermelhas. Essas informações são transferidas para o espectro visível e o novo RGB é retornado (DOLIOTIS et al., 2009). O Quadro 1 descreve o pseudocódigo do algoritmo Daltonize.

Quadro 1 - Psuedocódigo do algoritmo Daltonize

```

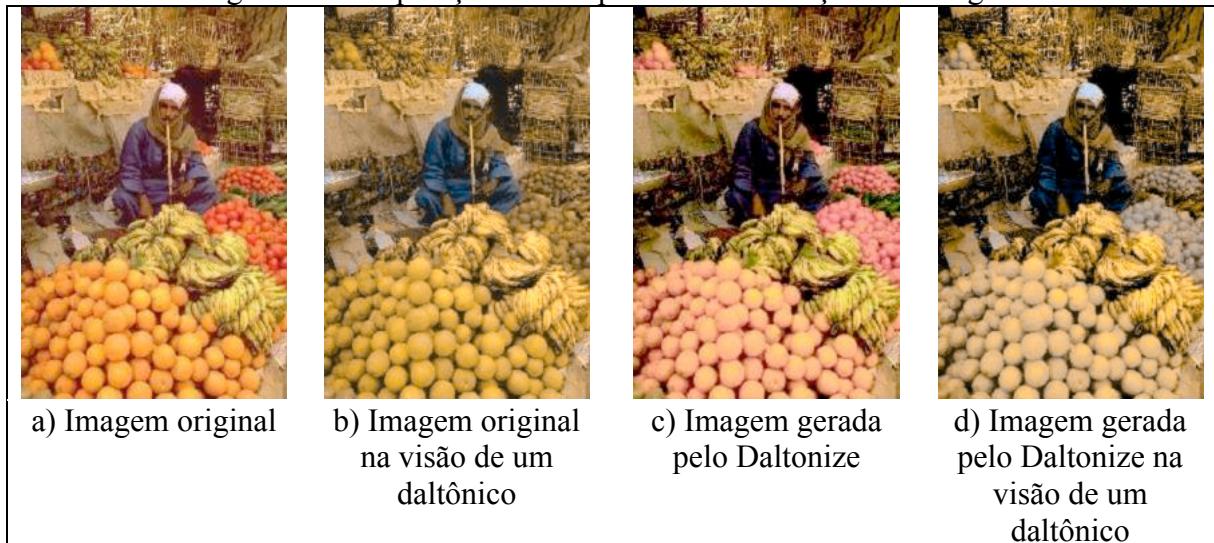
inicio
  para cada pixel em imagem faça
    RGB := ExtrairRGB(pixel)
    LMS := ConverteRGBParaLMS(RGB)
    se tipoDaltonismo = 'pronatopia' então
      LMS := SimulaPronatopia(LMS);
    se tipoDaltonismo = 'deuteranopia' então
      LMS := SimulaDeuteranopia(LMS);
    se tipoDaltonimo = 'tritanopia' então
      LMS := SimulaTritanopia(LMS);
    RGB_DALTONICO := ConverteLMSParaRG(LMS);
    erro := CalculaErro(RGB_DALTONICO, RGB);
    NOVO_RGB := TransfereCorEspectro(erro, RGB_DALTONICO);
    retorna NOVO_RGB;
  fim

```

Fonte: adaptado de Doliotis et al. (2009).

Um dos objetivos do algoritmo é modificar as imagens tentando manter a legibilidade para pessoas não daltônicas e seu uso pode ser diverso como por exemplo em aparelhos de TV, telas de computador e celular, transmissões ao vivo, páginas na internet ou mesmo em imagens impressas como mapas, manual de instruções entre outros (DOUGHERTY; WADE, 2000?). A Figura 9 item (a) apresenta uma imagem que após simulada na visão do daltônico, como mostra a Figura 9 item (b) demonstra uma grande perda de informações quando comparadas com a original.

Figura 9 - Comparaçao entre a perda de informações da imagem



Fonte: adaptado de Vischeck (2012).

A Figura 9 item (c) apresenta a mesma imagem, agora processada pelo algoritmo do autor, que quando novamente simulada na visão do daltônico apresenta de forma mais evidente as diferentes frutas ao redor. É importante notar que a verdadeira cor da imagem é irrelevante, porém, saber que ela é diferente das outras cores ao redor é muito importante (VISCHECK, 2012).

2.4 TRABALHOS CORRELATOS

Acessibilidade para portadores de daltonismo já foi abordado por diversos trabalhos. A seguir estão relacionados três trabalhos relacionados ao desenvolvido. A seção 2.4.1 aborda o trabalho de Lee (2008) que criou ferramentas de auxílio a portadores de daltonismo. Na seção 2.4.2, o trabalho de Nishimori (2013) apresenta um aplicativo que possui a simulação de daltonismo, filtro para melhorar a visualização de imagens por daltônicos e destaque de cor em imagens. Por fim, na seção 2.4.3 o trabalho de Paula Júnior (2014) mostra um estudo das principais ferramentas móveis disponíveis nas principais plataformas.

2.4.1 Uma ferramenta adaptativa para facilitar a visualização de imagens para pessoas portadoras de daltonismo

Lee (2008) desenvolveu três ferramentas computacionais com o propósito de aumentar a acessibilidade de portadores de daltonismo. A primeira tem como objetivo diagnosticar o daltonismo do usuário que é submetido a uma versão customizada do teste de Ishihara onde foram atribuídos pesos para as perguntas buscando também avaliar aproximadamente o grau de severidade do distúrbio. Procurando conhecer um pouco como o daltonismo afeta a visualização de uma imagem, a segunda ferramenta foi desenvolvida para simular o

daltonismo dos tipos mais comuns: a protanopia e a deuteranopia. Após estudos e de testes realizados com a ferramenta de simulação, ficou evidente a perda de informação visual e então foi desenvolvida uma terceira ferramenta com o objetivo de corrigir imagens com Lógica Fuzzy para melhorar a visualização por portadores de daltonismo. Para o desenvolvimento das 3 ferramentas foi utilizado a linguagem Python e as bibliotecas PyQt – para criar a interface gráfica, Python Imaging Library (PIL) – para processamento de imagens, NumPy – para manipular matrizes e a Equalize – para equalização do histograma das imagens.

Após testes com voluntários que foram submetidos a ferramenta de teste desenvolvida, e seus resultados usados para calibrar a ferramenta de correção, foi observado que “com a correção em RGB e equalização de histograma, as imagens tornaram-se mais compreensíveis, pois elementos, que antes eram percebidos com a mesma cor, devido ao daltonismo, receberam cores diferentes” (LEE, 2008, p. 47). Já outros métodos de correção como Red, Green, Blue (RGB) sem equalização e Longwave, Middlewave, Shortwave (LMS) não obtiveram resultados tão satisfatórios, como demonstrado na Tabela 1.

Tabela 1 - Resultado da avaliação dos voluntários

		Com equalização	Sem equalização	
RGB	Muito melhor	20%	Muito melhor	0%
	Melhor	46%	Melhor	43%
	Igual	20%	Igual	43%
	Pior	14%	Pior	14%
	Muito pior	0%	Muito pior	0%
LMS	Muito melhor	20%	Muito melhor	7%
	Melhor	17%	Melhor	33%
	Igual	0%	Igual	3%
	Pior	7%	Pior	7%
	Muito pior	36%	Muito pior	50%

Fonte: adaptado de Lee (2008, p. 47).

Como extensão de seu trabalho, Lee (2008) propôs a implementação de métodos de diagnóstico menos populares na ferramenta de testes como o teste de pranchas pseudoisocromáticas da American Optical Hardy-Rand-Rittler. Outra proposta foi a análise de outras formas de correção utilizando sistemas de cores como XYZ, CMY e HIS e o desenvolvimento da ferramenta na linguagem C, buscando assim um ganho de performance. Para a ferramenta de simulação de daltonismo, foi proposta a utilização da Lógica Fuzzy podendo assim simular com mais realidade a visão do portador de daltonismo e a simulação de daltonismo do tipo tritan.

2.4.2 Ferramenta de Acessibilidade para Deficientes Visuais em Cores

Nishimori (2013) desenvolveu um aplicativo com três funcionalidades voltadas a deficientes visuais em cores, a primeira é para simulação de como uma imagem é vista por pessoas com deficiências visuais em cores, a segunda é um filtro que facilita ou possibilita a visualização de informações por essas pessoas e a terceira para destacar cores em uma imagem. O aplicativo foi desenvolvido sob a plataforma Android, utilizando a biblioteca Open Source Computer Vision Library (OpenCV).

A Figura 10 apresenta o resultado da simulação onde percebe-se que houve perda de detalhes importantes que impactam diretamente na informação da imagem e, que ela ocasionaria certa confusão ao ser visualizada por parte dos portadores de daltonismo.

Figura 10 - Resultado da simulação



Fonte: Nishimori (2013, p. 40).

O Filtro desenvolvido por Nishimori (2013) assume que apesar dos contrastes de vermelho e verde sejam difíceis de serem percebidos, os contrastes entre vermelhos e azuis e verdes e azuis são percebidos normalmente. Conforme o valor escolhido no filtro, ele altera a intensidade de azul em pixels com alta intensidade de vermelho ou então transforma a intensidade de azul em pixels com alta intensidade de verde. A Figura 11 exemplifica o resultado das imagens após a aplicação do filtro desenvolvido, deixando-as em contrastes que facilitaram o portador a identificar facilmente as informações, mesmo em casos extremos.

Figura 11 - Resultado do filtro



Fonte: Nishimori (2013, p. 41).

Como extensão Nishimori (2013) propôs a migração das ferramentas para outras plataformas como iOS e Windows Phone, levando a ferramenta a grande maioria do mercado de dispositivos móveis. Outra possibilidade proposta foi a adição do filtro de acessibilidade

para os tipos de daltonismo tritanomalia e tritanopia e uma ferramenta de diagnóstico através dos cartões de Ishihara.

2.4.3 Proposta de um aplicativo móvel Open Source em auxílio a indivíduos com Discromopsia baseado em um estudo qualitativo

Paula Júnior (2014) fez uma análise de vários aplicativos voltados à acessibilidade de daltônicos que se encontram disponíveis nas plataformas mais populares do mercado e ao fim disponibilizou uma proposta de um novo aplicativo que venha a suprir as necessidades encontradas. Primeiramente realizou pela internet uma pesquisa de mercado com nove voluntários portadores de daltonismo a respeito dessa deficiência em suas vidas, problemas no cotidiano, além de possíveis sistemas já utilizados e a quais recursos tecnológicos possuíam. Com o resultado, foram criados os critérios descritos no Quadro 2.

Quadro 2 - Critérios utilizados

NOME	SIGLA
Gratuito	GRAT
Indica a cor ao usuário	INDI
Recurso de foto	FOTO
Recurso de flash	FLAS
Recurso de câmera frontal	CFRO
Testes para detecção	TEST
Simulação para cada tipo	SIMU
Interpretação por voz	VOZ
Correção para cada tipo	CORR

Fonte: adaptado de Paula Júnior (2014, p. 48).

Após o levantamento das respostas, foi então reunido e analisado vinte e duas soluções e ferramentas já existentes para o auxílio do daltonismo, treze delas para a plataforma Android e nove para a plataforma iOS. Os resultados são apresentados no Quadro 3 e Quadro 4 e indicam que em ambas as plataformas a grande maioria dos aplicativos é gratuito, facilitando a disponibilidade para pessoas de menor poder aquisitivo. Outro ponto interessante é a falta de suporte a interpretação por voz, ao recurso de foto, ao uso de *flash* e câmera frontal.

Quadro 3 - Aplicativos iOS analisados

NOME	GRAT	INDI	VOZ	FOTO	FLAS	CRFR	TEST	SIMU	CORR
ColorAdd	N	S	S	S	N/D	N	N	N	N
IDaltonizer	N	N	N	N	N	N	N	S	S
SayColor	S	S	S	N/D	N/D	N/D	N/D	N/D	N/D
Color ID Free	S	S	S	S	N	N	N	N	N
Chromatic Vision	S	N	N	N	N	N	N	S	N
Vision Test	S	N	N	N	N	N	S	N	N
EyeXam	S	N	N	N	N	N	S	N	N
HueVue: Colorblind Tools	S	N/D	N/D	N/D	N/D	N/D	N/D	N/D	S
Colorblind Aid	S	N/D	N/D	N/D	N/D	N/D	N/D	N/D	S

Fonte: adaptado de Paula Júnior (2014, p. 49).

Quadro 4 - Aplicativos Android analisados

NOME	GRAT	INDI	VOZ	FOTO	FLAS	CRFR	TEST	SIMU	CORR
Color Grab	S	S	S	S	S	N	N	N	N
ColorBlind Vision	N	S	N	N/D	N	S	S	S	S
Color Blindness Test	S	N	N	N	N	N	S	S	N
Color Blind Correction		S	N	N	N	N	S	S	S
Colorblind Helper	S	N	N	N	S	N	N	S	S
Vision Check Up	S	N	N	N	N	N	S	N	N
Visual Acuity Test	S	N	N	N	N	N	S	N	N
Color Vision Test	S	N	N	N	N	N	S	N	N
Color Helper 4 Men	S	S	N	N	S	N	N	N	N
Dankam		N/D	N/D	N/D	N/D	N/D	N/D	N/D	S
ColorToPattern Camera	S	S	N	S	N/D	N	N	N	S
Color Amplifier	S	N	N	N	S	S	N	S	S
Daltonizer	S	N	N	N	N	N	N	S	N

Fonte: adaptado de Paula Júnior (2014, p. 49).

Com o resultado da pesquisa, Paula Júnior (2014) concluiu que pelo fato de muitos não terem ciência do que é daltonismo há um desinteresse tanto por parte das escolas como pelos consultórios no diagnóstico do problema o que acarreta em dificuldades no cotidiano do daltônico e falta de soluções de acessibilidade. Também concluiu que não existem softwares escaláveis, totalmente gratuitos e que possam simultaneamente abranger toda a gama de indivíduos com todos os critérios levantados por ele. Paula Júnior (2014) então criou a proposta de um projeto de desenvolvimento, incluindo ciclo de vida, metodologias e análise de viabilidade além da arquitetura a ser utilizada e a partir disso foi submeteu juntamente com outro estudante aa Faculdade Católica Salesiana do Espírito Santo um projeto de iniciação científica, que na data de publicação do trabalho já se encontrava em fase de desenvolvimento.

Como extensão de seu trabalho propôs aprimorar a arquitetura para que fique preparada para suportar uma integração com plataforma de TV Digital Ginga além de dispositivos como computadores, relógios, óculos ou qualquer outro sistema tecnológico que se aplique.

3 DESENVOLVIMENTO

Este capítulo aborda os aspectos referentes à construção do aplicativo. Na seção 3.1 encontram-se os requisitos funcionais e não funcionais. Na seção 3.2 são apresentados os diagramas de casos de uso e classes, especificando e detalhando o funcionamento do aplicativo. A seção 3.3 descreve as técnicas e ferramentas utilizadas e apresenta os principais trechos de código do aplicativo. Por fim, a seção 3.4 expõe os resultados obtidos após o desenvolvimento do aplicativo e realiza a comparação entre trabalhos correlatos apresentados na seção 2.4.

3.1 REQUISITOS

No Quadro 5 são descritos os Requisitos Funcionais (RF) e seus respectivos casos de uso e no Quadro 6 os Requisitos Não Funcionais (RNF).

Quadro 5 - Rastreabilidade dos Requisitos Funcionais

Requisito Funcional (RF)	Caso de uso (UC)
RF01: permitir ao usuário tirar fotos a partir da câmera de um dispositivo	UC01
RF05: permitir ao usuário selecionar uma foto da galeria de imagens do dispositivo	UC05
RF02: permitir ao usuário selecionar o tipo de daltonismo a ser usado como base para o processamento de imagens	UC02
RF03: fazer o processamento de imagens realçando/ajustando as cores (de acordo com o tipo de daltonismo selecionado) para melhor visualização do usuário	UC03
RF04: gerar e apresentar uma nova imagem com o resultado do processamento ao usuário	UC04
RF06: permitir ao usuário visualizar em tempo real a imagem processada	UC06

Fonte: elaborado pelo autor.

Quadro 6 - Requisitos Não Funcionais

Requisito Não Funcional (RNF)
RNF01: ser implementado na plataforma Android (Requisito Não Funcional - RNF);
RNF02: ser implementado utilizando a linguagem de programação Java (RNF);

Fonte: elaborado pelo autor.

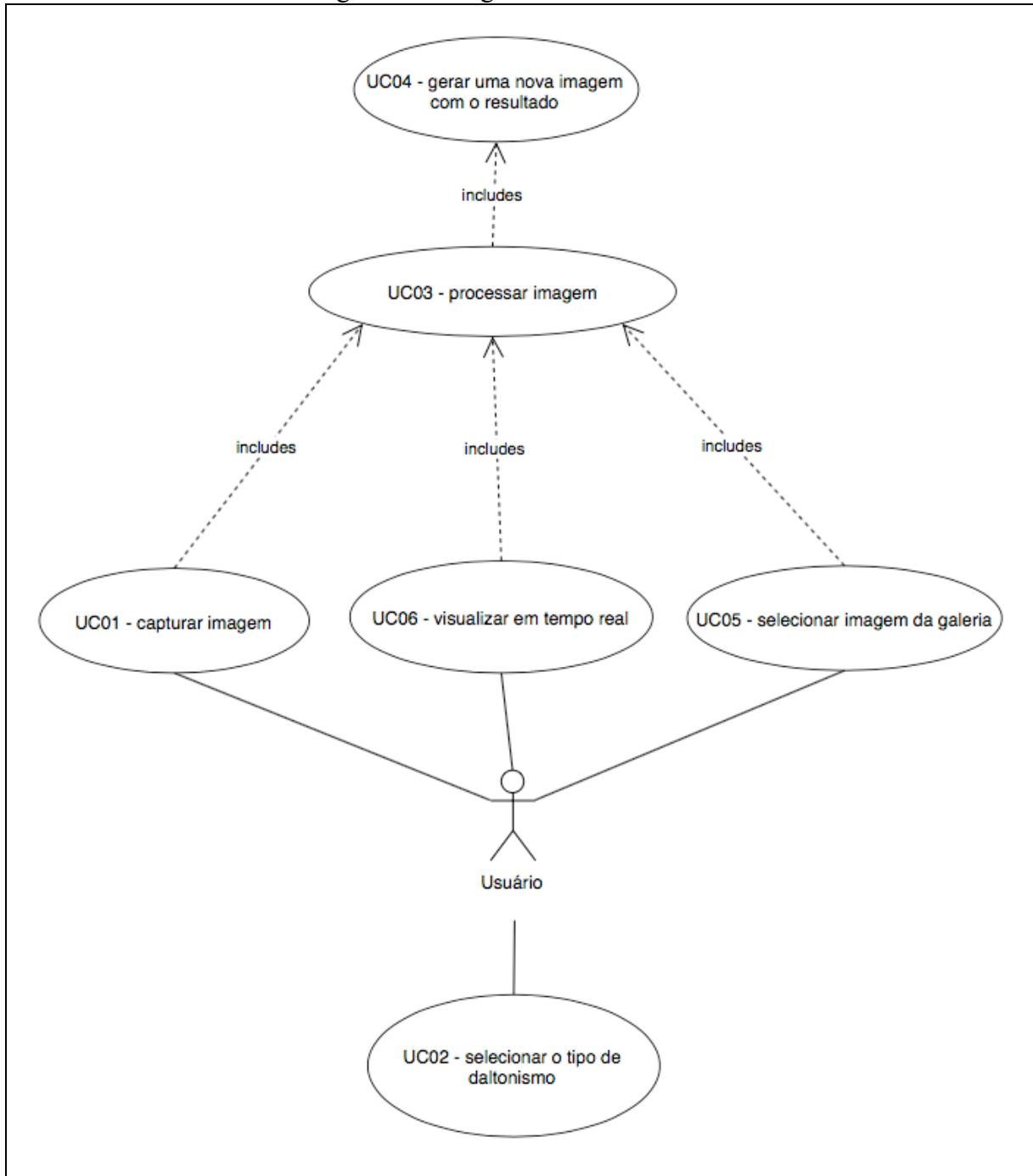
3.2 ESPECIFICAÇÃO

A especificação da aplicação foi representada através de diagramas da Unified Modeling Language (UML), utilizando a ferramenta Draw.io. Neste trabalho foram desenvolvidos os diagramas de casos de uso, classes e de atividades.

3.2.1 Diagrama de Casos de Uso

A partir do levantamento dos requisitos da aplicação, foi desenvolvido o diagrama de casos de uso conforme ilustrado na Figura 12, que representam as funcionalidades disponíveis para o ator Usuário.

Figura 12 - Diagrama de casos de uso



Fonte: elaborado pelo autor.

O ator Usuário inicializa diretamente quatro casos de uso. O caso de uso UC01 – capturar imagem permite ao usuário tirar uma foto a partir do dispositivo. No caso de uso UC02 – selecionar o tipo de daltonismo o Usuário pode selecionar o tipo de

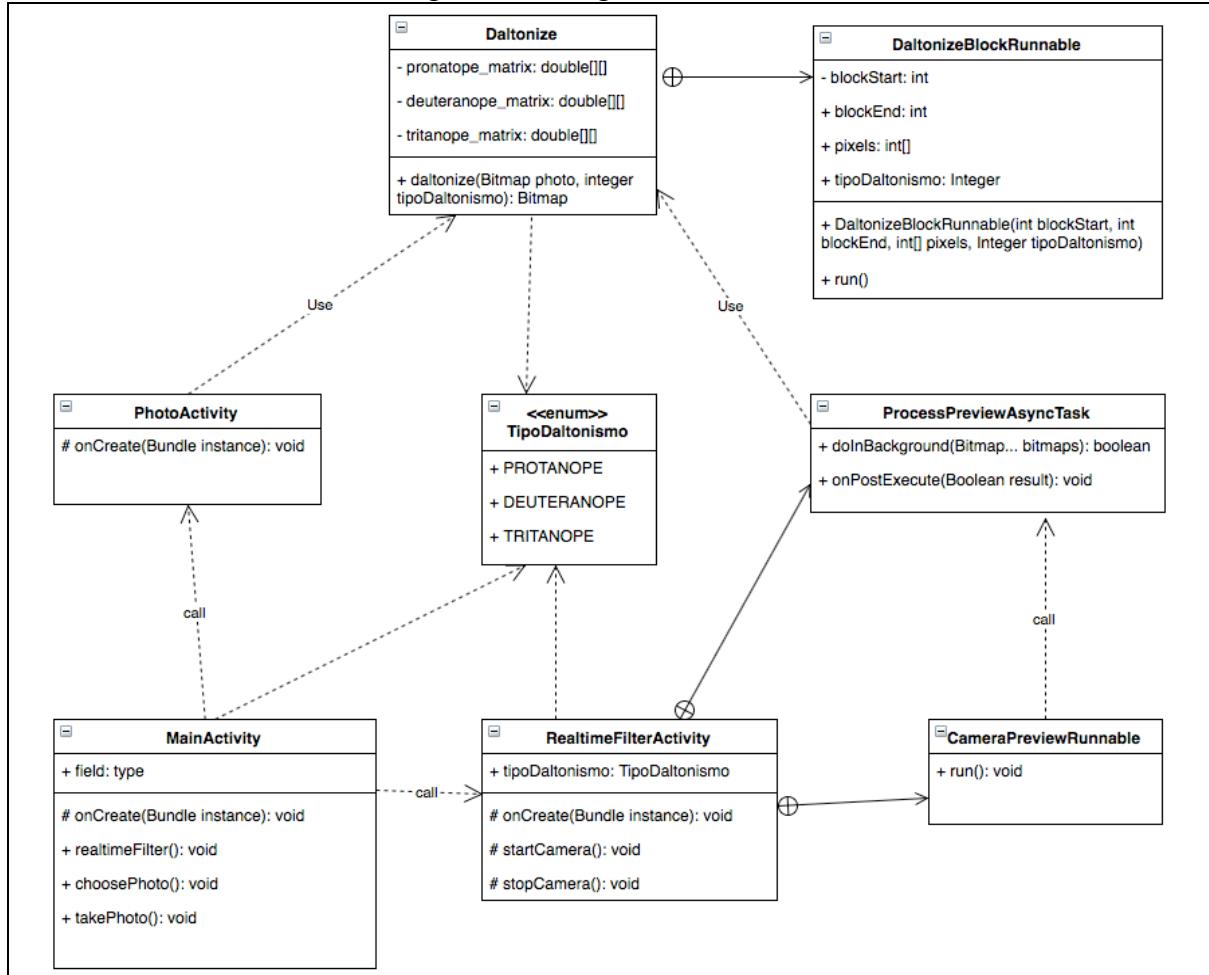
daltonismo a ser usado ao realizar o processamento de imagem. O caso de uso UC05 – selecionar imagem da galeria descreve a funcionalidade que permite ao Usuário selecionar uma foto da galeria de imagens do dispositivo. Por fim, o caso de uso UC06 – visualizar em tempo real permite ao Usuário visualizar as imagens processadas em tempo real.

Existem também dois casos de uso que são utilizados por outras funcionalidades, como por exemplo o caso de uso UC04 – gerar uma nova imagem com o resultado que apresenta ao usuário a imagem processada. E, por fim, o caso de uso UC03 – processar imagem que se refere a execução do algoritmo de processamento de imagens.

3.2.2 Diagrama de Classes

O diagrama de classes apresenta uma visão de como as classes estão estruturadas e relacionadas. Nesta seção são descritas as classes que compõem o aplicativo, conforme apresenta a Figura 13.

Figura 13 - Diagrama de Classes



Fonte: elaborado pelo autor.

A classe `MainActivity` é responsável por fornecer as opções de interação com o aplicativo através dos seus métodos. O método `takePhoto` apresenta a funcionalidade de captura de imagem ao usuário. A funcionalidade de seleção da galeria de imagens é inicializada pelo método `choosePhoto`. Por fim, o método `realtimeFilter` é responsável pela funcionalidade de execução em tempo real. Essa classe também é responsável por inicializar a Activity `PhotoActivity` que é responsável por exibir para o usuário a foto processada.

A classe `RealtimeFilterActivity` é responsável pela funcionalidade de exibir em tempo real a imagem gerada pelo algoritmo, utilizando como classes auxiliares a `CameraPreviewRunnable` que é responsável por delegar os frames para a `ProcessPreviewAsyncTask` que os processa de forma assíncrona. A classe `CameraPreviewRunnable` é uma implementação de `Runnable` que no método `run` executa a parametrização e configuração da câmera do dispositivo e define a classe `ProcessPreviewAsyncTask` como responsável por receber os *frames* da câmera. A classe `ProcessPreviewAsyncTask` é uma extensão de `AsyncTask` que executa a classe responsável pelo algoritmo e apresenta o resultado ao usuário.

A classe `Daltonize` é responsável por realizar a lógica que separa os pixels da imagem em blocos de 300 e, a partir disso, inicializa a classe `DaltonizeBlockRunnable` passando os blocos criados. Por fim, a classe `DaltonizeBlockRunnable` é responsável pela execução de cada bloco de pixels com o algoritmo que será detalhado no item 3.3.1.3

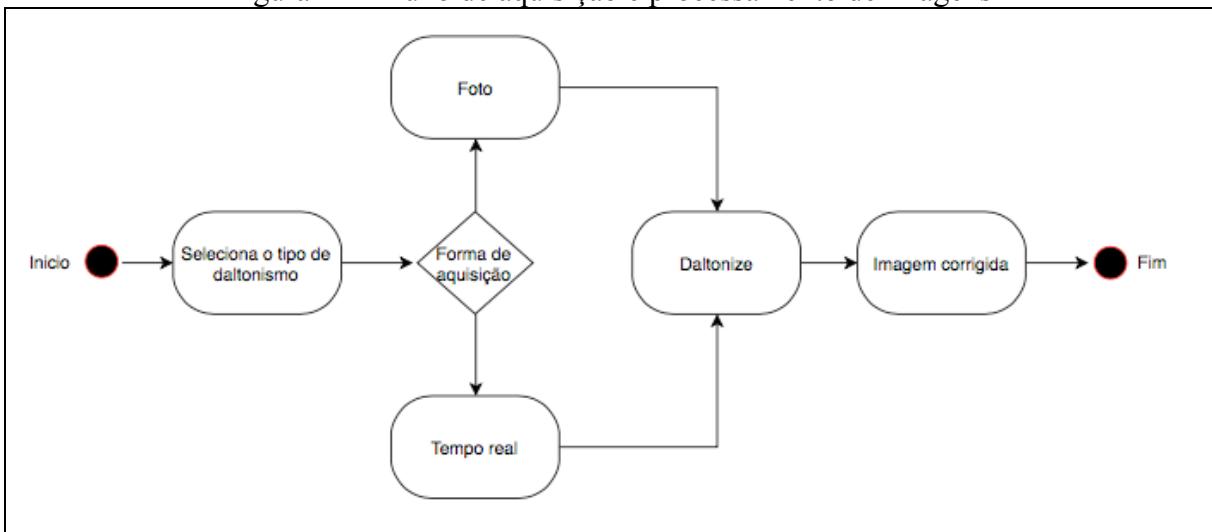
3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do aplicativo foi utilizado a linguagem de programação Java na versão 7. O ambiente de desenvolvimento foi o Android Studio 2.3.2, ferramenta oficial para criação de aplicativos Android disponibilizada pela Google. Além de todo o ambiente de desenvolvimento, são disponibilizadas várias API's pelo SDK do Android, como a de câmera, galeria de imagens e manipulação de imagens, ela também oferece um simulador para testes sem dispositivos físicos. O fluxo básico de aquisição e processamento de imagens da aplicação está representado na Figura 14.

Figura 14 - Fluxo de aquisição e processamento de imagens



Fonte: elaborado pelo autor.

Ao abrir o aplicativo é exibida a tela principal. Nela, o usuário seleciona o tipo de daltonismo que servirá de base para o processamento de imagens, conforme demonstra a Figura 15 item (a).

Figura 15 - Seleção de daltonismo e opções de aquisição de imagem



Fonte: elaborado pelo autor.

Após selecionar o tipo de daltonismo há dois modos de informar as imagens para o algoritmo, o primeiro realizando captura pela câmera do dispositivo ou pelo acesso a galeria de imagens e a segunda, em tempo real. Neste caso, o processamento de imagens é realizado para onde o usuário estiver direcionando o aparelho. Estas opções são demonstradas na Figura 15 item (b) e explicadas nas próximas seções.

3.3.1.1 Aquisição de imagens

Ao selecionar a opção **Tirar foto** o usuário é redirecionado para a interface da câmera do dispositivo, conforme mostrado na Figura 16 item (a), podendo tirar uma foto que será processada conforme o tipo de daltonismo selecionado. Essa opção é ideal para situações onde é necessária ter maior qualidade de imagem ao invés de agilidade, como por exemplo ao

escolher uma roupa, uma fruta no supermercado ou ainda para ler um mapa de transporte público.

Ao selecionar a opção `Escolher foto` o usuário é redirecionado para a galeria de imagens do usuário para ele selecionar uma foto previamente capturada, recebida de alguém ou baixada da internet, conforme mostra a Figura 16 item (b).

Figura 16 - Opções de aquisição de imagem



Fonte: elaborado pelo autor.

Para acessar a câmera e a galeria de imagens foi configurado uma `Intent` do Android que ao ser invocada, inicializa as funções nativas do sistema para acesso ao recurso, retornando o conteúdo da *callback* que realiza o processamento de imagens. O resultado é apresentado ao usuário em uma nova `Activity`. No Quadro 7 é apresentado o código para inicialização da câmera.

Quadro 7 - Configuração da `Intent` de uso da câmera

```

01 private void dispatchTakePictureIntent() {
02     Intent takePictureIntent = new
03         Intent(MediaStore.ACTION_IMAGE_CAPTURE);
04     if (takePictureIntent.resolveActivity(getPackageManager()) !=
05         null) {
06         startActivityForResult(takePictureIntent,
07             REQUEST_IMAGE_CAPTURE);
08     }
09 }
```

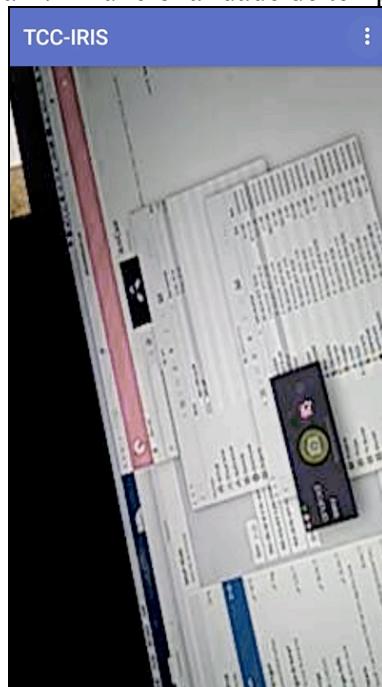
Fonte: elaborado pelo autor.

3.3.1.2 Filtro em tempo real

Ao selecionar a funcionalidade `tempo real` o usuário é redirecionado para a tela onde é apresentado o que a câmera do dispositivo está capturando naquele momento. Essa

funcionalidade permite agilidade para o portador do daltonismo pois permite “apontar e ver” coisas ao seu redor, sem a necessidade de capturar uma foto, conforme demonstra na Figura 17.

Figura 17 - Funcionalidade de tempo real



Fonte: elaborado pelo autor

Para implementar a funcionalidade de tempo real, foi necessário usar o componente `TextureView` do Android, configurando-o para que cada *frame* da câmera fosse processado individualmente por ele. Com o `TextureView` configurado, obteve-se acesso ao *stream* de imagens da câmera, tendo assim a possibilidade de obter os *frames*. Esse *callback* é executado em paralelo por uma *thread* para melhorar a performance, conforme mostra o Quadro 8.

Quadro 8 - Inicialização da camera do dispositivo

```

01 synchronized protected void startCamera() {
02     if (!isStarting) {
03         isStarting = true;
04         new Thread(new CameraPreviewRunnable()).start();
05     }
06 }
```

Fonte: elaborado pelo autor.

Ao receber um *frame* é verificado se o algoritmo já está sendo executado. Se isso estiver acontecendo, o *frame* é adicionado no *buffer* de processamento. Caso nenhum *frame* esteja sendo processado, ele é convertido em *Bitmap* e enviado a uma *thread* que chama o algoritmo de processamento. Este processo é executado até que o usuário finalize a funcionalidade. No Quadro 9 é apresentado o código que realiza este procedimento.

Quadro 9 - Execução do *callback* do *frame* da câmera

```

01 mCamera.setPreviewCallback(new Camera.PreviewCallback() {
02     @Override
03     public void onPreviewFrame(byte[] bytes, Camera camera) {
04         if (daltonizing) {
05             //adiciona no buffer
06             mCamera.addCallbackBuffer(bytes);
07         } else {
08             Camera.Parameters params = camera.getParameters();
09             int w = params.getPreviewSize().width;
10             int h = params.getPreviewSize().height;
11             int format = params.getPreviewFormat();
12             //conversão do frame para bitmap
13             YuvImage image = new YuvImage(bytes, format, w, h, null);
14             ByteArrayOutputStream out = new ByteArrayOutputStream();
15             Rect area = new Rect(0, 0, w, h);
16             image.compressToJpeg(area, 50, out);
17             Bitmap bmp =
18                 BitmapFactory.decodeByteArray(out.toByteArray(), 0, out.size());
19             //executa o processamento em uma thread
20             new ProcessPreviewAsyncTask().execute(bmp);
21         }
22     }
23 });

```

Fonte: elaborado pelo autor.

Por se tratar de uma funcionalidade que requer muito processamento do dispositivo, alguns aparelhos mais antigos podem ter travamento. Por isso, foi implementado a opção de alterar a qualidade da imagem. Assim, os usuários desses aparelhos podem ajustar a execução da funcionalidade de acordo com a resolução do seu dispositivo, conforme demonstra Figura 18.

Figura 18 - Seleção de qualidade da imagem



Fonte: elaborado pelo autor.

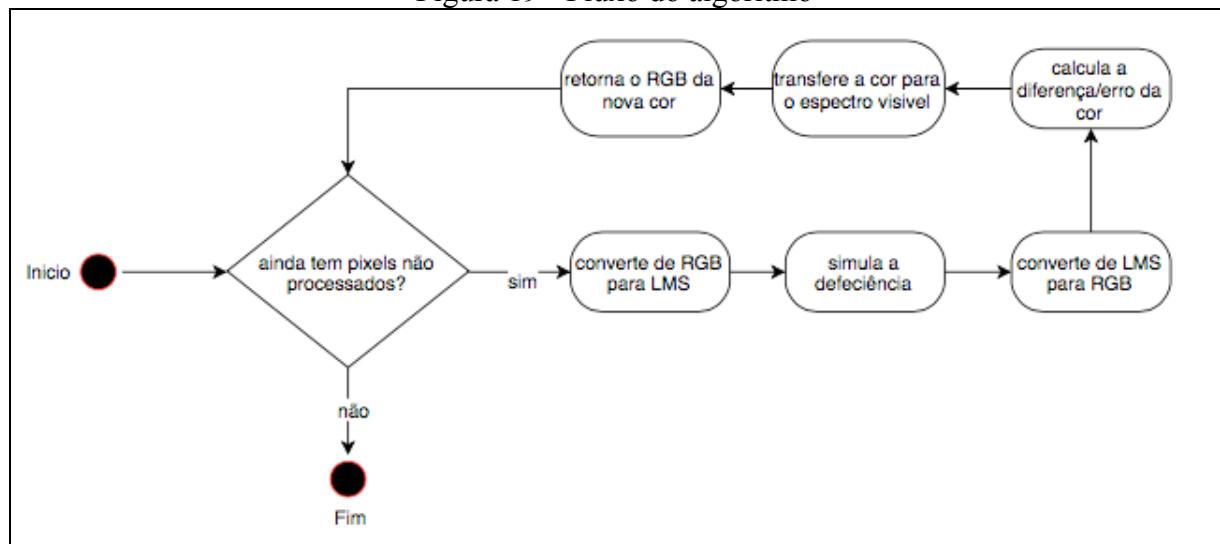
Para realizar o ajuste da qualidade de captura, a câmera é configurada conforme a resolução selecionada pelo usuário. Para disponibilizar a lista de resoluções é verificado todas

as resoluções suportadas pela câmera, colocando-as em um menu ao qual o usuário pode selecionar.

3.3.1.3 Daltonize

Para melhor entendimento, na Figura 19 é apresentado o fluxo do algoritmo. Com o objetivo de realizar a alteração de uma imagem, o algoritmo realiza algumas transformações de espectro para simular a visão humana ao qual são realizados cálculos para detectar quais os pontos da imagem não são distinguidos pelo daltônico, conseguindo assim espalhar essas cores ao longo do espectro visível.

Figura 19 - Fluxo do algoritmo



Fonte: elaborado pelo autor.

Inicialmente, a imagem de entrada é separada em blocos de 300 pixels e cada bloco é lançado para processamento em uma *thread* que executará o algoritmo. Para cada pixel recebido é extraído o RGB dele através da classe `android.graphics.Color` do Android SDK, conforme mostra o Quadro 10.

Quadro 10 - Extração do RGB de um pixel

01	<code>int p = pixels[i];</code>
02	<code>int r = Color.red(p);</code>
03	<code>int b = Color.blue(p);</code>
04	<code>int g = Color.green(p);</code>

Fonte: elaborado pelo autor.

Com o RGB extraído, é efetuada a conversão para LMS, que representa a cor no nível da retina. Essa conversão é feita utilizando a técnica proposta por Viénot, Brettel e Mollon (1999, p. 245), conforme pode ser visto no Quadro 11.

Quadro 11 - Conversão RGB para LMS

```

01 private static double[] RGBToLMS(double r, double g, double b) {
02     double L = (17.8824 * r) + (43.5161 * g) + (4.11935 * b);
03     double M = (3.45565 * r) + (27.1554 * g) + (3.86714 * b);
04     double S = (0.0299566 * r) + (0.184309 * g) + (1.46709 * b);
05     return new double[] { L, M, S };
06 }
```

Fonte: elaborado pelo autor.

Com as cores em LMS, não é possível deletar apenas as informações do cone defeituoso, é necessário calcular a transformação de cores para os demais cones. Essa mudança é feita simulando a visão do daltônico, conforme técnica proposta por Viénot, Brettel e Mollon (1999, p. 248), que definiu matrizes de transformação para protanopia, deuteranopia e tritanopia, demonstradas no Quadro 12.

Quadro 12 - Matrizes de transformação

```

01 //Pronatopia
02 private static double[][] protanope_matrix = new double[][] {
03     {0.0, 2.02344, -2.52581},
04     {0.0, 1.0, 0.0},
05     {0.0, 0.0, 1.0}
06 };
07
08 //Deuteranopia
09 private static double[][] deuteranope_matrix = new double[][] {
10     {1.0, 0.0, 0.0},
11     {0.494207, 0.0, 1.24827},
12     {0.0, 0.0, 1.0}
13 };
14
15 //Tritanopia
16 private static double[][] tritanope_matrix = new double[][] {
17     {1.0, 0.0, 0.0},
18     {0.0, 1.0, 0.0},
19     {-0.395913, 0.801109, 0.0}
20 };
```

Fonte: elaborado pelo autor.

Para efetuar a transformação é feita uma multiplicação com a matriz correspondente, conforme Quadro 13.

Quadro 13 - Transformação de espaço de cores

```

01 l = (cvd_matrix[0][0] * L) + (cvd_matrix[0][1] * M) +
(cvd_matrix[0][2] * S);
02 m = (cvd_matrix[1][0] * L) + (cvd_matrix[1][1] * M) +
(cvd_matrix[1][2] * S);
03 s = (cvd_matrix[2][0] * L) + (cvd_matrix[2][1] * M) +
(cvd_matrix[2][2] * S);
```

Fonte: elaborado pelo autor.

Depois de transformada para o espectro do daltônico, a cor é convertida novamente para RGB, realizando multiplicação com a matriz inversa do processo anterior, conforme demonstrado no Quadro 14.

Quadro 14 - Conversão LMS para RGB

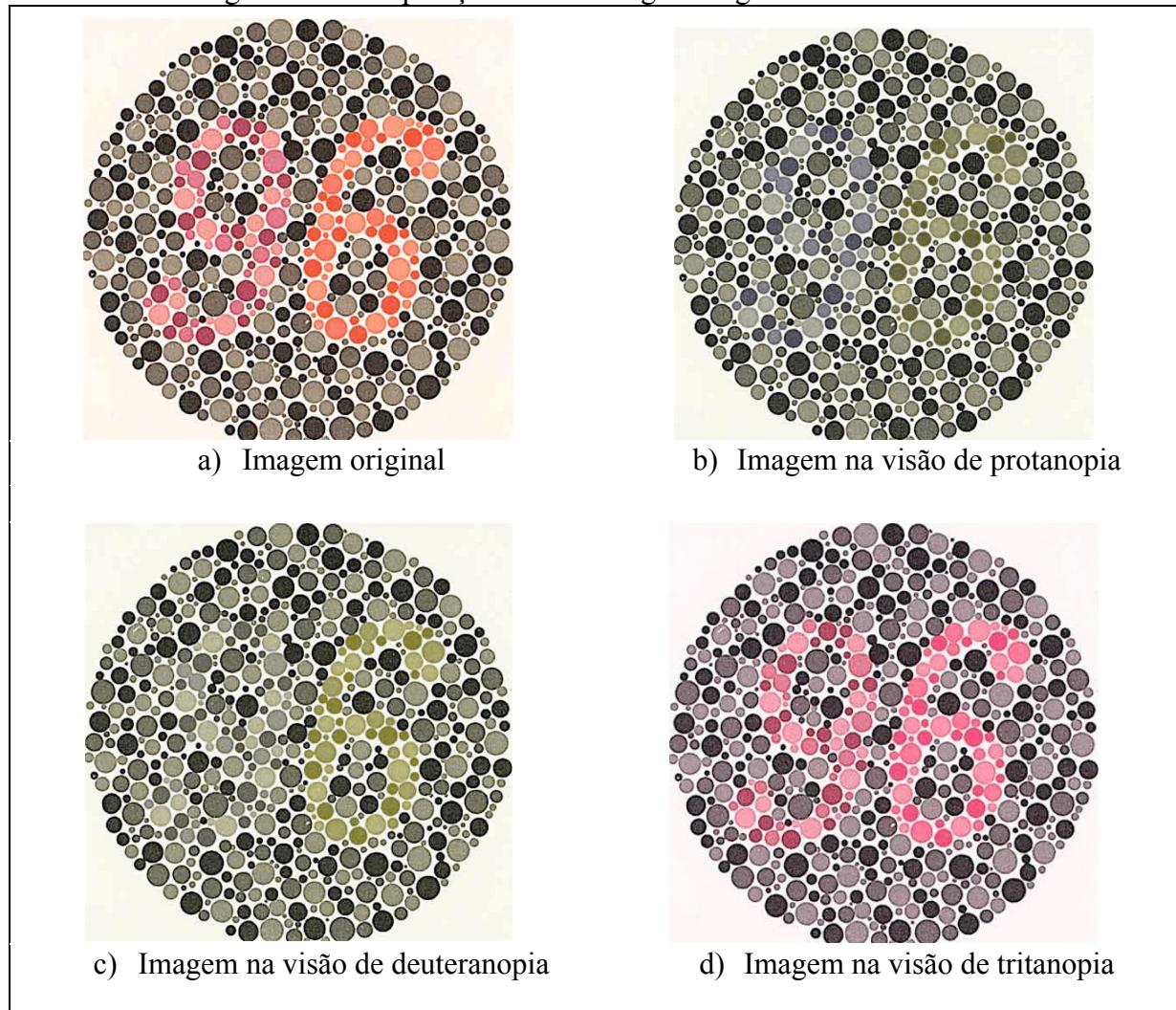
```

01 private static double[] LMSToRGB(double l, double m, double s) {
02     double R = (0.0809444479 * l) + (-0.130504409 * m) + (0.116721066
03     * s);
04     double G = (-0.0102485335 * l) + (0.0540193266 * m) + (-
0.113614708 * s);
05     double B = (-0.000365296938 * l) + (-0.00412161469 * m) +
0.693511405 * s);
06     return new double[] { R, G, B };
07 }
```

Fonte: elaborado pelo autor.

A Figura 20 apresenta a comparação entre a imagem original e a convertida, simulando a deficiência e evidenciando as perdas de informação na visão de um daltônico

Figura 20 - Comparaçao entre a imagem original e a convertida



Fonte: elaborado pelo autor.

Com as cores novamente em RGB, é calculada a matriz de erro, que consiste dos valores RGB do espectro daltônico subtraídos da imagem original. O resultado representa a perda existente entre a visão normal e do daltônico, o cálculo é mostrado no Quadro 15 e o resultado na Figura 21.

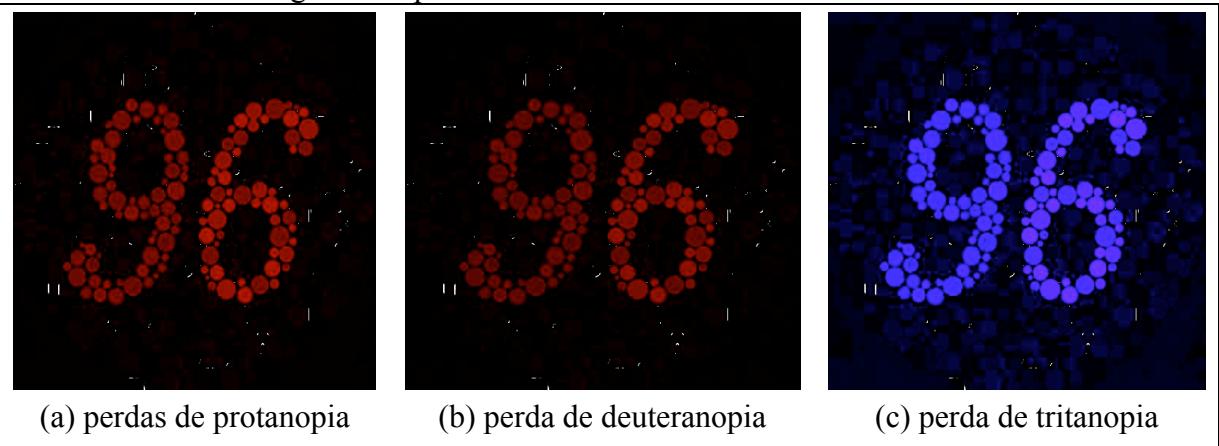
Quadro 15 - Cálculo da matriz de erro

01	$R = r - R;$
02	$G = g - G;$
03	$B = b - B;$

Fonte: elaborado pelo autor.

A Figura 21 demonstra as imagens com a perda entre a imagem original e a visão do daltônico.

Figura 21 - perda entre a visão normal e do daltônico



Fonte: elaborado pelo autor.

Para transferir as cores para fora do espectro do daltônico é feita uma multiplicação da matriz de erro por uma matriz pré-definida. Posteriormente, soma-se o resultado da multiplicação com as cores originais, conforme demonstrado no Quadro 16.

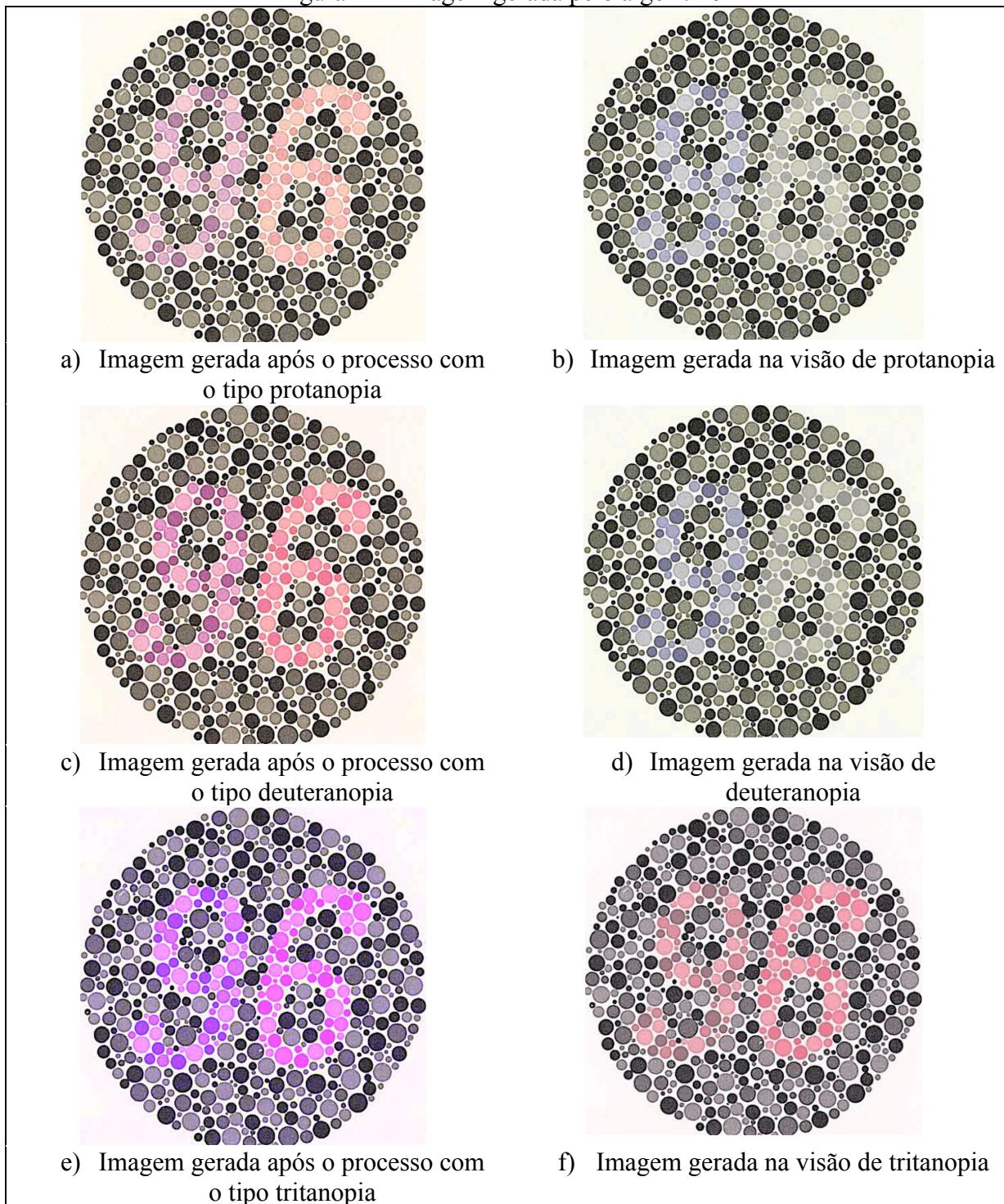
Quadro 16 - Cálculo da cor final

01	$RR = (0.0 * R) + (0.0 * G) + (0.0 * B);$
02	$GG = (0.7 * R) + (1.0 * G) + (0.0 * B);$
03	$BB = (0.7 * R) + (0.0 * G) + (1.0 * B);$
04	$R = RR + r;$
05	$G = GG + g;$
06	$B = BB + b;$

Fonte: elaborado pelo autor.

Por fim, os valores são normalizados para caso tenham ultrapassado os valores RGB. A partir disso, o novo valor RGB de cada pixel é retornado. Após todos os pixels serem processados, uma nova imagem é gerada, sendo a versão que favorece a visualização do daltônico. A Figura 22 demonstra a imagem gerada sob a visão do daltônico.

Figura 22 - Imagem gerada pelo algoritmo



Fonte: elaborado pelo autor.

Na imagem utilizada como exemplo é possível evidenciar a melhora de distinção da informação para os tipos de protanopia e deuteranopia. A imagem de tritanopia já era de fácil visualização e manteve a sua legibilidade.

3.4 ANÁLISE DOS RESULTADOS

Nesta seção são apresentados os experimentos realizados com o aplicativo. Na seção 3.4.1 é feita a análise do perfil e o resultado da execução da lista de tarefas realizadas pelos usuários. Na 3.4.2 é feita a análise dos resultados da avaliação da usabilidade e funcionalidades do aplicativo. Por fim, na seção 3.4.3 é feita a comparação desse trabalho com os seus correlatos.

3.4.1 Análise do perfil e resultados da lista de tarefas

O primeiro passo foi analisar o perfil dos oito voluntários através de um questionário (disponível no Quadro 18 do Apêndice A) aplicado no início da avaliação. Todos os voluntários eram homens e possuíam smartphone, onde 26% tinha entre 20 e 25 anos, 37% entre 25 e 30 anos e 37% entre 30 e 35 anos. Referente ao tipo de daltonismo 50% possuíam deuteranopia, 38% protanopia e 12% tritanopia dos quais 50% descobriram o problema na escola, 38% em casa e 12% durante as atividades no trabalho.

Depois de analisar o perfil do usuário, foi aplicado uma lista de tarefas (disponível no Quadro 19 do Apêndice A) a todos os voluntários. Em sua grande maioria não houve problemas na execução das tarefas. Apenas uma pessoa relatou dificuldade ao realizar a instalação do aplicativo e outras duas tiveram um erro durante a execução da funcionalidade de tempo real. O resumo dos resultados é apresentado na Tabela 2.

Tabela 2 - Resultados da execução de tarefas

Tarefa	Resposta	Sim	Sim, com observação	Não
Instalação do aplicativo		88%		12%
Seleção do tipo de daltonismo		100%		
Seleção da funcionalidade de tempo real		100%		
Teste da funcionalidade de tempo real		75%	25%	
Seleção da funcionalidade de capturar foto		100%		
Teste da funcionalidade de capturar foto		100%		
Seleção da funcionalidade de imagem da galeria		100%		
Teste da funcionalidade de imagem da galeria		100%		

Fonte: elaborado pelo autor.

O erro encontrado durante a instalação referia-se a um software que bloqueava o download de aplicativos por e-mail, sendo assim não está diretamente relacionado ao aplicativo desenvolvido. Já os problemas ocorridos durante a execução da funcionalidade de

tempo real foram paradas inesperadas do aplicativo, sem uma razão conclusiva. Na segunda tentativa, a execução ocorreu normalmente, possibilitando a execução da tarefa.

3.4.2 Análise da usabilidade e funcionalidades do aplicativo

Os voluntários também responderam um questionário (disponível no Quadro 20 do Apêndice A) com sete perguntas referentes a usabilidade do aplicativo e cinco relacionadas às funcionalidades e impressões do usuário. Na Tabela 3 são apresentadas as perguntas e respostas referentes à usabilidade.

Tabela 3 - Avaliação da usabilidade do aplicativo

Perguntas / Respostas	Concordo Plenamente	Concordo Parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
Foi fácil encontrar as informações que você precisou	38%	62%			
O design da interface do aplicativo é atraente		12%	63%	13%	12%
A organização dos menus e botões é lógica, permitindo encontrá-los facilmente na tela		25%	75%		
É fácil lembrar como fazer as coisas no aplicativo	63%	37%			
O tamanho da tela é suficiente para utilizar as funcionalidades		75%		25%	
Às vezes eu não sei o que fazer com este aplicativo				75%	25%
Você precisaria do apoio de uma pessoa para usar este aplicativo				38%	62%

Fonte: elaborado pelo autor.

As respostas indicam que a grande maioria dos usuários não tiveram dificuldades para encontrar as informações no aplicativo e, que é fácil de lembrar como fazer as coisas após a primeira utilização. O design da interface do aplicativo foi o ponto de maior disparidade nas respostas, onde 12% concordaram que é atraente, 63% não concordaram nem discordaram e 25% discordaram dessa afirmação. Outro ponto que gerou discordância entre os participantes foi o tamanho da tela, por estarem executando o aplicativo em telas menores que 4 polegadas, 25% dos usuários discordaram que o tamanho da tela é suficiente para o uso do aplicativo.

A organização dos menus e botões foi considerada lógica para 25% dos voluntários, o restante não concorda nem discorda com essa afirmação. Todos os usuários discordaram que precisem de auxílio para utilizar o aplicativo. Também discordaram que há alguma dificuldade na hora de saber o que fazer no aplicativo, possivelmente por que todos já possuem e utilizam smartphone.

Na Tabela 4 encontram-se as perguntas e respostas referentes às funcionalidades do aplicativo e a aceitação dos usuários.

Tabela 4 - Avaliação das funcionalidades do aplicativo

Perguntas / Respostas	Concordo Plenamente	Concordo Parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
O aplicativo em algum momento parou inesperadamente	25%				75%
Você considera a funcionalidade de tempo real rápida				63%	37%
Foi mais fácil compreender as informações nas imagens geradas na tela pelo aplicativo	75%	25%			
Você recomendaria o aplicativo para outras pessoas	100%				
Você acredita que o aplicativo poderia ajudar no seu dia a dia	38%	25%	37%		

Fonte: elaborado pelo autor.

Após análise das respostas a respeito das funcionalidades do aplicativo, observou-se que 25% dos usuários enfrentaram algum problema quando apareceu uma parada inesperada do aplicativo. Porém o restante não relatou nada de estranho. Outro ponto a ser observado e que deve ser melhorado é a velocidade de processamento da funcionalidade de tempo real. Todos os voluntários relataram que a funcionalidade é lenta, onde ocorreram pequenos travamentos durante a execução. Todos os usuários concordaram que foi mais fácil compreender nas imagens geradas pelo aplicativo e também que recomendariam o aplicativo a outras pessoas. Quanto a utilidade do aplicativo no dia a dia, 63% concordaram e o restante não concordou nem discordou.

3.4.3 Comparativo entre o aplicativo desenvolvido e seus correlatos

O Quadro 17 apresenta um comparativo entre o aplicativo desenvolvido e os trabalhos correlatos.

Quadro 17 - Comparaçao com os trabalhos correlatos

Característica / correlatos	Pereira (2017)	Lee (2008)	Nishimori (2013)	Paula Júnior (2014)
Teste/Diagnóstico de Daltonismo	Não	Sim	Não	Não
Simulação de Daltonismo	Não	Sim	Sim	Não
Processamento de imagem para melhor visualização por daltônicos?	Sim	Sim	Sim	Sim
Possui versão específica para smartphones?	Sim	Não	Sim	Sim
Multi-plataforma (mais de uma)?	Não	Não	Não	Sim
Correção de imagem em tempo real?	Sim	Não	Sim	Não
Permite ajustar a qualidade da imagem gerada?	Sim	Não	Não	Não

Fonte: elaborado pelo autor

As ferramentas de Lee (2009) e Nishimori (2013) possuem funcionalidades que realizam a simulação de imagens na visão dos daltônicos buscando compreender as dificuldades que possuem. A de Lee (2009) também oferece o diagnóstico da deficiência através de uma versão customizada do teste de Ishihara. Este aplicativo e o de Paula Júnior (2014) focam apenas no processamento das imagens e com exceção ao de Lee (2008) todos os trabalhos possuem versão para smartphones, demonstrando assim a importância desses dispositivos como ferramenta de acessibilidade. O trabalho de Paula Júnior (2014) é o único que possui versão para mais de uma plataforma.

A funcionalidade de destaque deste aplicativo é a de processamento em tempo real das imagens, funcionalidade também implementada por Nishimori (2013). Essa funcionalidade apresentou alguns problemas de performance e que na abordagem Nishimori (2013) não foram evidenciados. Como solução para tentar possibilitar a melhor usabilidade em dispositivos de menor processamento foi criada a funcionalidade de ajuste de qualidade da imagem gerada. Essa funcionalidade não foi disponibilizada por nenhum dos trabalhos comparados, tornando-a um diferencial em relação ao seus correlatos.

4 CONCLUSÕES

O crescente aumento de informações que recebemos e a importância das cores para a compreensão deixa evidente os problemas de acessibilidade que os daltônicos enfrentam. Com a popularização de smartphones ao redor do mundo e com o potencial que esses dispositivos possuem no suporte a medicina é natural que possam vir a se tornar ferramentas importantes de acessibilidade para diversos problemas.

O objetivo de desenvolver um aplicativo que melhore a qualidade visual de imagens para pessoas portadoras de daltonismo conforme seu tipo de daltonismo foi atingido utilizando o algoritmo Daltonize em conjunto com as funcionalidades da câmera do dispositivo que são disponibilizadas pela plataforma. O aplicativo foi desenvolvido utilizando a linguagem de programação Java no ambiente Android Studio e utilizando as funcionalidades disponibilizadas pelo Android SDK. No que diz respeito as ferramentas utilizadas não se enfrentou nenhuma dificuldade notável durante o desenvolvimento

Em relação aos trabalhos correlatos, o trabalho de Nishimori (2013) mostrou-se o mais completo, oferecendo simulação de daltonismo, indicação de cores e correção de imagens, incluindo em tempo real sem apresentar problemas de performance. Observou-se também o desenvolvimento para alguma plataforma móvel em dois dos três trabalhos comparados, mostrando a tendência no uso de plataformas móveis na acessibilidade.

Com o resultado dos testes é possível concluir que os objetivos iniciais propostos por esse trabalho foram alcançados. Foi criado um aplicativo móvel em uma plataforma altamente difundida e que obteve bons resultados em ajudar na acessibilidade de pessoas portadoras de daltonismo do grupo do dicromatismo, oferecendo uma interface fácil e com boas funcionalidades. Porém, os mesmos resultados indicam que há pontos a melhorar, principalmente no aprimoramento da interface e na otimização da funcionalidade de tempo real, que apresentou problemas de performance.

Espera-se que com o desenvolvimento e evolução deste aplicativo melhorar a inclusão de pessoas com dificuldades por conta de sua deficiência de cores, auxiliando na execução de tarefas de seu dia a dia de forma ágil e acessível.

4.1 EXTENSÕES

Algumas das possíveis extensões para este trabalho são:

- a) outros tipos de daltonismo: estender o algoritmo e/ou implementar outros para dar suporte a mais tipos de daltonismo além dos abordados nesse trabalho;
- b) performance da funcionalidade de tempo real: melhorar a performance da

funcionalidade de tempo real, possibilitando assim imagens de melhor qualidade em smartphones de menor processamento;

- c) outras plataformas: apesar do Android ter o maior número de usuários, a plataforma iOS da Apple também possui grande participação. Criar uma versão desse aplicativo para essa e outras plataformas que se mostrarem relevantes;
- d) ajustes na imagem gerada: estudar e implementar a possibilidade do usuário ajustar a imagem gerada, podendo assim dar mais e ou menos intensidade nas alterações conforme o seu grau de daltonismo.

REFERÊNCIAS

- ARAYA, Ana Maria Osorio et al. Abordando o daltonismo como questão sócio-científica para introdução de conceito de luz e espectro eletromagnético visível no ensino médio. In: I CONGRESO LATINO-AMERICANO DE INVESTIGACIÓN EN DIDACTICA DE LAS CIENCIAS EXPERIMENTALES Y DE MATEMÁTICA, 1., 2012, Santiago. **Livro de resumenes del I Congreso Latino-Americano de Investigación de las Ciencias Experimentales y de Matemática**. Santiago: Pontificia Universidad Católica de Chile, 2012. p. 2 - 3.
- BRINDLEY, Giles Skey. Human colour vision. **Progress In Biophysics And Biophysical Chemistry**. ?, p. 49-94. jan. 1957.
- BRUNI, Lígia Fernanda; CRUZ, Antonio Augusto Velasco e. Sentido cromático: tipos de defeitos e testes de avaliação clínica. **Arquivos Brasileiros de Oftalmologia**, São Paulo, v. 69, n. 5, p.766-775, out. 2006. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0004-27492006000500028>.
- CISCO. **Cisco prevê que 70% da população mundial usará dispositivos móveis em 2020**. 2016. Disponível em: <<https://americas.thecisconetwork.com/site/content/lang/pt/id/4997>>. Acesso em: 20 maio 2017.
- COLBLINDOR. **Ishihara's Test for Colour Deficiency: 38 Plates Edition**. 2017. Disponível em: <<http://www.color-blindness.com/ishiharas-test-for-colour-deficiency-38-plates-edition/>>. Acesso em: 12 jun. 2017.
- COLORMAX. **Farnsworth-Munsell 100**. 2017. Disponível em: <<https://colormax.org/farnsworth-munsell-100/>>. Acesso em: 12 maio 2017.
- COLOUR BLIND AWARENESS (Org.). **Colour Blind Awareness**. [2012?]. Disponível em: <<http://www.colourblindawareness.org/>>. Acesso em: 01 jan. 2017.
- DOLIOTIS, Paul et al. Intelligent Modification of Colors in Digitized Paintings for Enhancing the Visual Perception of Color-blind Viewers. **Ifip Advances In Information And Communication Technology**, Boston, p.293-301, 2009. Springer US. http://dx.doi.org/10.1007/978-1-4419-0221-4_35.
- DOUGHERTY, Bob; WADE, Alex. **Vischeck: About Daltonize**. [2000?]. Disponível em: <<http://www.vischeck.com/daltonize/>>. Acesso em: 23 maio 2017.
- FERNANDES, Mariana Abreu. **As implicações de problemas visuais no processo de aprendizagem escolar das crianças**. 2012. 50 f. Dissertação (Mestrado) - Curso de Optometria Ciências da Visão, Universidade da Beira Interior, Covilhã, 2012.
- GOURAS, Peter. **Color Vision**. 2012. Disponível em: <<http://webvision.med.utah.edu/book/part-vii-color-vision/color-vision/>>. Acesso em: 06 abr. 2017.
- HERMELIN, Solo. **Color Theory**. 2014. Disponível em: <<http://www.slideshare.net/solohermelin/color-theory-39956038>>. Acesso em: 06 abr. 2017.
- KINNEAR, P R; SAHRAIE, A. New Farnsworth-Munsell 100 hue test norms of normal observers for each year of age 5–22 and for age decades 30–70. **The British Journal Of Ophthalmology**. Londres, p. 1408-1411. dez. 2002.
- LAKSHMINARAYANAN, Vasudevan; ZELEK, John S.; MCBRIDE, Annette. "Smartphone Science" in Eye Care and Medicine. **Optics And Photonics News**, Washington, v. 44, n. 26, p.43-51, jan. 2015.

- LEE, Jinmi. **Uma ferramenta adaptativa para facilitar a visualização de imagens para pessoas portadoras de daltonismo.** 2008. 60 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade de Pernambuco, Recife, 2008.
- MAIA, Amanda Fortes dalla Valle da; SPINILLO, Carla. Avaliação de mapas do transporte de curitiba: uma abordagem centrada no usuário daltônico. In: CONFERÊNCIA INTERNACIONAL DE DESIGN, ENGENHARIA E GESTÃO PARA A INOVAÇÃO, 2., 2012, Florianópolis. **Anais...** . Florianópolis: Universidade do Estado de Santa Catarina – Udesc, 2012. p. 21 - 23.
- MANDAL, Ananya. **Congenital Color Vision Deficiencies.** 2014. Disponível em: <<http://www.news-medical.net/health/Congenital-Color-Vision-Deficiencies.aspx>>. Acesso em: 30 maio 2017.
- MANTOVANI, Camila; DANTAS, G. G. C. . **Os fluxos informacionais nos dispositivos móveis.** In: Maria Aparecida Moura. (Org.). 2011. Cultura informacional e liderança comunitária. 1ed.Belo Horizonte: PROEX/UFMG, 2011, v. , p. 121-125.
- MELO, D. G.; GALON, J. E. V.; FONTANELLA, B. J. B. Os “daltônicos” e suas dificuldades: condição negligenciada no Brasil? **Physis: Revista de Saúde Coletiva**, v. 24, n. 4, p. 1229–1253, dez. 2014.
- NEW HEALTH ADVISOR (Org.). **What Colorblind People See?** 2014. Disponível em: <<http://www.newhealthadvisor.com/What-Colorblind-People-See.html>>. Acesso em: 30 maio. 2017.
- NISHIDA, S. M.; OLIVEIRA, F. A. K. DE; TROLL, J. **Formação da Imagem e Percepção Visual.** [2010?]. Disponível em: <http://www2.ibb.unesp.br/nadi/Museu2_qualidade/Museu2_corpo_humano/Museu2_como_funciona/Museu_homem_nervoso/Museu_homem_nervoso_visao/Museu2_homem_nervoso_visao_mecanismo.htm>. Acesso em: 15 maio. 2017.
- NISHIMORI, Renato Augusto Vieira. **Ferramenta de Acessibilidade para Deficientes Visuais em Cores.** 2013. 53 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade de São Paulo, São Paulo, 2013.
- OTHMAN, Amal Hayati Che; SABUDIN, Maziani. A study of colour transformation for colour deficient individual. In: 2013 IEEE STUDENT CONFERENCE ON RESEARCH AND DEVELOPMENT, 2013, Putrajaya, Malaysia. **Proceedings...** . Putrajaya, Malaysia: Ieee, 2013. p. 328 - 333. Disponível em: <<http://ieeexplore.ieee.org/document/7002601/>>. Acesso em: 15 maio 2017.
- PAULA JÚNIOR, Carlos Alberto de. **Proposta de um aplicativo móvel open source em auxílio a indivíduos com discromopsia baseado em um estudo qualitativo.** 2014. 87 f. TCC (Graduação) - Curso de Sistemas de Informação, Faculdade Católica Salesiana do Espírito Santo, Vitória, 2012.
- SHEPEARD, C. **How Technology Helps the Color Blind See a More Vibrant World.** 2015. Disponível em: <<http://www.care2.com/causes/how-technology-helps-the-color-blind-see-a-more-vibrant-world.html>>. Acesso em: 13 maio. 2017.
- TANUWIDJAJA, Enrico et al. Chroma: a wearable augmented-reality solution for color blindness. In: PROCEEDINGS OF THE 2014 ACM INTERNATIONAL JOINT CONFERENCE ON PERVASIVE AND UBIQUITOUS COMPUTING, 1., 2014, Seattle, Washington. **Proceedings...** . Seattle, Washington: Acm, 2014. p. 799 - 810. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2632048.2632091>>. Acesso em: 15 maio 2017.

TSUDA, Hugo Kenji. **ClaroVisão – Uma extensão de navegador Web para pessoas com Daltonismo.** 2013. 84 f. TCC (Graduação) - Curso de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

VIÉNOT, F.; BRETEL, H.; MOLLON, J. D. Digital video colourmaps for checking the legibility of displays by dichromats. **Color Research and Application**, v. 24, n. 4, p. 243–252, 1999.

VISCHECK. **Vischeck: About Daltonize.** 2012. Disponível em: <<http://www.vischeck.com/daltonize/>>. Acesso em: 20 maio 2017.

APÊNDICE A – Questionários e lista de tarefas

Neste apêndice constam o questionário feito aos voluntários a fim de identificar o perfil de usuário (Quadro 18), a lista de tarefas executadas (Quadro 19) e o questionário de avaliação do aplicativo (Quadro 20).

Quadro 18 - Questionário de perfil do usuário

PERFIL DE USUÁRIO

Sexo:

() Masculino () Feminino

Idade:

- | | |
|------------------------------|------------------------------|
| () Tenho menos de 5 anos | () Tenho entre 20 e 25 anos |
| () Tenho entre 5 e 10 anos | () Tenho entre 25 e 30 anos |
| () Tenho entre 10 e 15 anos | () Tenho entre 30 e 35 anos |
| () Tenho entre 15 e 20 anos | () Tenho mais de 35 anos |

Nível escolaridade:

- | | |
|---|-------------------------------------|
| () Ensino fundamental incompleto | () Ensino médio completo – 2º grau |
| () Ensino fundamental completo – 1º grau | () Ensino superior incompleto |
| () Ensino médio incompleto | () Ensino superior completo |

Observação: _____

Você possui smartphone?

() Sim () Não

Qual seu nível de familiaridade com smartphones?

() Nunca utilizei () Às vezes () Frequentemente

Qual seu tipo de daltonismo?

() Protanopia () Deutanopia () Tritanopia

Com que idade você descobriu que era daltônico?

- | | |
|------------------------|------------------------|
| () menos de 5 anos | () entre 20 e 25 anos |
| () entre 5 e 10 anos | () entre 25 e 30 anos |
| () entre 10 e 15 anos | () entre 30 e 35 anos |
| () entre 15 e 20 anos | () depois de 35 anos |

Se possível, cite a forma de descobrimento:

Em qual aparelho você irá utilizar o aplicativo para a realização das tarefas?

() celular () tablet modelo: _____ versão do Android: _____ tela: _____

Quadro 19 - Lista de tarefas do usuário

LISTA DE TAREFAS DO USUÁRIO

1. Instalação do aplicativo

Baixe o aplicativo e faça sua instalação.

A tarefa foi executada? Sim, não? Por quê?

2. Seleção do tipo de daltonismo

Selecione nos botões da tela principal o seu tipo de daltonismo.

A tarefa foi executada? Sim, não? Por quê?

3. Seleção da funcionalidade de tempo real

No menu aberto após a seleção do tipo de daltonismo, clique na opção “tempo real”.

A tarefa foi executada? Sim, não? Por quê?

4. Teste da funcionalidade de tempo real

Após selecionado a opção “tempo real”, aponte a câmera do celular para o ambiente ao seu redor e observe as imagens na tela do celular.

A tarefa foi executada? Sim, não? Por quê?

5. Seleção da funcionalidade de capturar foto

Aperte o botão para voltar, selecione novamente o tipo de seu tipo de daltonismo e no menu aberto clique na opção “tirar foto”.

A tarefa foi executada? Sim, não? Por quê?

6. Teste da funcionalidade de capturar foto

Após selecionado a opção “tirar foto”, aponte a câmera do celular para o ambiente ao seu redor e tire uma foto, após isso observe a imagem gerada na tela do celular.

A tarefa foi executada? Sim, não? Por quê?

7. Seleção da funcionalidade de imagem da galeria

Aperte o botão para voltar, selecione novamente o tipo de seu tipo de daltonismo e no menu aberto clique na opção “escolher foto”.

A tarefa foi executada? Sim, não? Por quê?

8. Teste da funcionalidade de imagem da galeria

Após selecionado a opção “escolher foto”, selecione alguma imagem da sua galeria e observe a imagem gerada na tela do celular.

A tarefa foi executada? Sim, não? Por quê?

Quadro 20 - Questionário de avaliação do aplicativo

Perguntas / Critérios de avaliação	Concordo totalmente	Concordo parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
Foi fácil encontrar as informações que você precisou					
O design da interface do aplicativo é atraente					
A organização dos menus e botões é lógica, permitindo encontrá-los facilmente na tela					
É fácil lembrar como fazer as coisas no aplicativo					
O tamanho da tela é suficiente para utilizar as funcionalidades					
Às vezes eu não sei o que fazer com este aplicativo					
Você precisaria do apoio de uma pessoa para usar este aplicativo					
O aplicativo em algum momento parou inesperadamente					
Você considera a funcionalidade de tempo real rápida					
Foi mais fácil compreender as informações nas imagens geradas na tela pelo aplicativo					
Você recomendaria o aplicativo para outras pessoas					
Você acredita que o aplicativo poderia ajudar no seu dia a dia					

Fonte: elaborado pelo autor.