

**CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ - CESUPA**  
**ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO - ARGO**  
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**ANA CAROLINA DE PARIJÓS GALENDE**  
**FLÁVIO SILVA SIQUEIRA**  
**JOÃO FELIPE TAVARES CALDERARO PEREIRA**

**PROPOSTA DE EXAME DE SENSIBILIDADE AO CONTRASTE UTILIZANDO**  
**SISTEMA MOBILE**

**BELÉM**

**2021**

ANA CAROLINA DE PARIJÓS GALENDE  
FLÁVIO SILVA SIQUEIRA  
JOÃO FELIPE TAVARES CALDERARO PEREIRA

**PROPOSTA DE EXAME DE SENSIBILIDADE AO CONTRASTE UTILIZANDO  
SISTEMA MOBILE**

Trabalho de conclusão de curso apresentado  
como requisito ao CENTRO  
UNIVERSITÁRIO DO ESTADO DO PARÁ -  
CESUPA para obtenção do título de Bacharel  
no CURSO DE ENGENHARIA DA  
COMPUTAÇÃO na modalidade  
MONOGRAFIA.

Prof. M.a Michelle Bitar Lelis dos Santos

BELÉM

2021

ANA CAROLINA DE PARIJÓS GALENDE  
FLÁVIO SILVA SIQUEIRA  
JOÃO FELIPE TAVARES CALDERARO PEREIRA

**PROPOSTA DE EXAME DE SENSIBILIDADE AO CONTRASTE UTILIZANDO  
SISTEMA MOBILE**

Trabalho de conclusão de curso apresentado  
como requisito ao CENTRO  
UNIVERSITÁRIO DO ESTADO DO PARÁ -  
CESUPA para obtenção do título de Bacharel  
no CURSO DE ENGENHARIA DA  
COMPUTAÇÃO na modalidade  
MONOGRAFIA.

Data da aprovação: / /

Nota Final aluno I: \_\_\_\_\_

Nota Final aluno II: \_\_\_\_\_

Nota Final aluno III: \_\_\_\_\_

Banca Examinadora:

*Michelle Bitar Santos*

Prof. M.a Michelle Bitar Lelis dos Santos (Orientadora)  
Centro Universitário do Estado do Pará

*Moshe Dayan Sousa Ribeiro*

Prof. M.e Moshe Dayan Sousa Ribeiro  
Centro Universitário do Estado do Pará

*Suzane A. Dias*

Prof. M.a Suzane Alfaia Dias  
Centro Universitário do Estado do Pará

BELÉM  
2021

**Dados Internacionais de Catalogação-na-publicação (CIP)**  
**Biblioteca do CESUPA, Belém – PA**

---

Galende, Ana Carolina Parijós

Proposta de exame de sensibilidade ao contrate utilizando sistema mobile /Ana Carolina Parijós Galende, Flávio Silva Siqueira, João Felipe Tavares Calderaro Pereira; Michelle Bitar Lelis dos Santos – 2021.

Trabalho de Conclusão de Curso (Graduação) – Centro Universitário do Estado do Pará, Engenharia de Computação, Belém, 2021.

1. Aplicativos móveis-Desenvolvimento. 2. Exames oftalmológicos I. Siqueira, Flávio Silva. II. Pereira, João Felipe Tavares Calderaro. III. Santos, Michelle Bitar Lelis dos (orient.). IV. Título.

CDD. 23 ed. 005.1

---

## **AGRADECIMENTOS**

Primeiramente, agradeço a Deus por me acompanhar em cada etapa da minha vida, por ser guiada em meus caminhos, sempre com proteção.

Agradeço à minha família por ser meu apoio incondicional. Em especial à minha mãe Esther, que me ensinou os verdadeiros valores da vida e sempre me orientou a seguir os caminhos corretos. Ao meu pai Paulo, por acreditar em meu potencial, e me ensinar a persistir e seguir meus sonhos. Aos meus irmãos Yasmin e Paulo, por serem meus companheiros de todas as horas e símbolos de confiança.

Devo agradecer à minha equipe, João Felipe e Flávio, por me acompanharem desde o início da minha formação acadêmica, e por seguirmos o desenvolvimento deste trabalho com tranquilidade e parceria.

À médica Júlia Souza, por nos acompanhar nas etapas de validação do projeto.

Agradeço também à nossa orientadora, professora Michelle Bitar, que acompanha nossa formação como engenheiros desde 2017, e nos guiou na criação deste trabalho.

Também agradeço às amigas criadas no decorrer do curso, amigos que estiveram em diversos momentos de muita alegria e que também me apoiaram durante o desenvolvimento deste trabalho.

**Ana Carolina de Parijós Galende.**

## **AGRADECIMENTOS**

Gostaria de agradecer, primeiramente, a minha família. Sem o incentivo deles não conseguiria alcançar meus objetivos. Individualmente, a minha mãe, Cirlena, por me ensinar a ter as atitudes certas e sempre me incentivar com meus objetivos. Ao meu pai, Josué, agradeço por sempre me encorajar com a área que decidi atuar. A minha irmã, Joscilene, agradeço por sempre estar ao meu lado, dialogando comigo, sendo um exemplo na minha vida.

À minha equipe, Ana e João Felipe, agradeço pelos anos de companheirismo, realizamos diversos trabalhos juntos, vocês me apoiaram nos momentos que mais precisava.

Agradeço a médica Júlia Souza, por ter colaborado com a validação do nosso trabalho.

À nossa orientadora, Michelle Bitar, muito obrigado por ter ensinado sobre o que é ser um engenheiro, por ter apoiado nosso projeto e orientado de forma incrível para a produção deste projeto.

Gostaria de agradecer a cada um dos amigos que marcaram essa trajetória na graduação. Ao Matheus e Rubens, agradeço pelas conversas e risadas que proporcionaram em nossa amizade. Agradeço à Stefany, pelos nossos diálogos a respeito de tudo, sempre me incentivando a ter um pensamento crítico. Ao Gabriel Melém, agradeço pelas conversas da tarde e madrugada, em que abordamos aspectos políticos, tecnológicos e jogos, sempre de forma descontraída.

Agradeço à toda turma de engenharia de computação de 2017, por nossos momentos de desespero com provas e debates sobre respostas diferentes.

Por fim, agradeço a todos os professores e coordenação do Cesupa que proporcionaram o conhecimento que hoje possuo.

Todos que citei, fizeram parte dessa trajetória.

**Flávio Silva Siqueira.**

## **AGRADECIMENTOS**

Agradeço a todos aqueles que me auxiliaram e apoiaram durante a jornada na graduação, em especial minha família.

**João Felipe Tavares Calderaro Pereira.**

## RESUMO

Em função da manutenção da saúde ocular, diversos tipos de exames devem ser realizados regularmente, portanto o encontro com médicos oftalmologistas é uma rotina vital para avaliação da acuidade visual. A fim de agilizar e facilitar esse processo, pode-se aproveitar o recurso computacional presente em dispositivos móveis. Então, com a finalidade de realizar testes de sensibilidade ao contraste por profissionais oftalmologistas por meio de *smartphones*, como uma possível alternativa viável, é desenvolvida e testada uma aplicação móvel. Assim, por meio deste projeto, há de se averiguar se tal sistema pode tornar-se uma alternativa viável para auxiliar ou substituir máquinas já existentes que realizam este tipo de exame médico.

Palavras-chave: Aplicativo. Contraste. Desenvolvimento. Exame. Oftalmológico.

## **ABSTRACT**

Due to the maintenance of eye health, several types of exams should be performed regularly, so meeting with ophthalmologists is a vital routine to assess visual acuity. In order to streamline and facilitate this process, people can take advantage of the computing resource present in mobile devices. To perform contrast sensitivity tests by ophthalmologists using smartphones, as a possible viable alternative, a mobile application is developed and tested. Thus, through this project, it is necessary to investigate whether such a system can become a viable alternative to assist or replace existing machines that perform this type of medical examination.

**Keywords:** Application. Contrast. Development. Exam. Ophthalmology.

## LISTA DE FIGURA

Figura 1	Tabela de grades apresentadas no exame	22
Figura 2	Sistema Optec® 6500	23
Figura 3	Tabela de respostas do paciente	24
Figura 4	Tabela de respostas com acertos do paciente	24
Figura 5	Relatório do exame de sensibilidade ao contraste	25
Figura 6	Exemplo de modelo UML	27
Figura 7	Diagrama de classes para sistema de caixa eletrônico	28
Figura 8	Relações do diagrama de classes	29
Figura 9	Representação do diagrama de objetos	30
Figura 10	Exemplificação de classes de componentes	31
Figura 11	Formulação do diagrama de implementação	32
Figura 12	Exemplo de servidor utilizando diagrama de estrutura composta	33
Figura 13	Diagrama de Casos de Uso para sistema de compras online	34
Figura 14	Diagrama de estrutural do Flutter	38
Figura 15	Diagrama de caso de uso referente à aplicação	44
Figura 16	Protótipo de baixa fidelidade	46
Figura 17	Formulário de baixa fidelidade	46
Figura 18	Página inicial em média fidelidade	47
Figura 19	Prototipagem do formulário do paciente	48
Figura 20	Páginas iniciais do aplicativo	49
Figura 21	Formulário e aviso de atenção	50
Figura 22	Tela do exame	51
Figura 23	Opções de direção das grades	51
Figura 24	Gráfico e resultados	52
Figura 25	Botão para começar o teste do olho esquerdo	53
Figura 26	Definição do <i>CustomButton</i>	54
Figura 27	Diagrama das classes presentes na aplicação	55
Figura 28	Requisição para o <i>Back-end</i>	56
Figura 29	Diagrama de componentes do <i>front-end</i>	56
Figura 30	Diagrama de objetos do <i>front-end</i>	57
Figura 31	Função de rotação das grades do exame	58
Figura 32	Código-fonte de geração das grades	59

Figura 33	Grades de sensibilidade ao contraste verticais	59
Figura 34	Grades de sensibilidade ao contraste com inclinação para direita	60
Figura 35	Grades de sensibilidade ao contraste com inclinação para a esquerda	60
Figura 36	Função de cálculo de resultado e interações com o usuário	61
Figura 37	Diagrama de implantação	62
Figura 38	Função para geração de imagens gratinadas	63
Figura 39	Retorno em JSON do <i>back-end</i> para a geração de imagem	64
Figura 40	Função geradora de gráficos	65
Figura 41	Bloco lógico de indexação de resultados	66
Figura 42	Gráfico de respostas do exame	66
Figura 43	Código-fonte referente a conversão de valores	67
Figura 44	Código-fonte relacionado a criação do arquivo em PDF	68
Figura 45	Estilização de arquivo PDF com base no exame realizado	69
Figura 46	Bloco lógico referente à geração da tabela de respostas	70
Figura 47	Código referente ao armazenamento de relatório	70

## LISTA DE SIGLAS

API	Interface de Programação de Aplicações
CEP	Comitê de Ética em Pesquisas
CBO	Conselho Brasileiro de Oftalmologia
DDS	<i>Decision Support Systems</i>
FSC	Função de Sensibilidade ao Contraste
FACT	<i>Contrast Sensitivity Testing</i>
ETDRS	<i>Early Treatment Diabetic Retinopathy Study</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
OSI	<i>Open Source Initiative</i>
PDF	<i>Portable Document Format</i>
SDK	<i>Software Development Kit</i>
SC	Sensibilidade ao Contraste
REST	<i>Representational State Transfer</i>
RFID	<i>Radio-Frequency Identification</i>
UX	<i>User Experience</i>
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
WSGI	<i>Web Server Gateway Interface</i>
WSDL	<i>Web Services Description Language</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>14</b>
1.1 SITUAÇÃO PROBLEMA	16
1.2 OBJETIVOS DO ESTUDO	17
<b>1.2.1 Objetivo geral</b>	<b>17</b>
<b>1.2.2 Objetivos específicos</b>	<b>17</b>
1.3 METODOLOGIA DA PESQUISA	17
1.4 ESTRUTURA DO TRABALHO	18
<b>2 FUNDAMENTAÇÃO CIENTÍFICA</b>	<b>19</b>
2.1 OFTALMOLOGIA E TECNOLOGIA	19
<b>2.1.1 Cenário atual</b>	<b>19</b>
<b>2.1.2 O exame oftalmológico</b>	<b>20</b>
<b>2.1.3 Tecnologia atual em smartphone</b>	<b>24</b>
<b>2.1.4 Arquitetura de software</b>	<b>25</b>
2.1.4.1 Diagrama de classes	26
2.1.4.2 Diagrama de objetos	29
2.1.4.3 Diagrama de componentes	29
2.1.4.4 Diagrama de implantação	30
2.1.4.5 Diagrama de estrutura composta	31
2.1.4.6 Diagrama de caso de uso	33
<b>2.1.5 Ferramenta de desenvolvimento</b>	<b>34</b>
2.1.5.1 Python	34
2.1.5.2 Biblioteca Psychopy	34
2.1.5.3 Biblioteca Flask	35
2.1.5.4 Biblioteca Matplotlib	35
2.1.5.5 Dart	36
2.1.5.6 Flutter	36
<b>2.1.6 Modelos de prototipação</b>	<b>37</b>
2.1.6.1 Baixa fidelidade	38
2.1.6.2 Média fidelidade	38
2.1.6.3 Alta fidelidade	38
<b>2.1.7 Arquitetura REST</b>	<b>39</b>
2.1.7.1 Null Style	39
2.1.7.2 Cliente-servidor	39
2.1.7.3 Sem Estado	39
2.1.7.4 Cache	40
2.1.7.5 Interface Uniforme	40
2.1.7.6 Separação em camadas	40
2.2 TRABALHOS RELACIONADOS	41
<b>3 DESENVOLVIMENTO DA PROPOSTA</b>	<b>43</b>
3.1 PLATAFORMA MÓVEL	43
<b>3.1.1 Prototipagem da aplicação</b>	<b>43</b>

3.1.1.1 Prototipagem de baixa fidelidade	44
3.1.1.2 Prototipagem de média fidelidade	46
3.1.1.3 Prototipagem de alta fidelidade	47
<b>3.1.2 Construção do Front-end</b>	<b>51</b>
<b>3.1.3 Integração ao Back-end</b>	<b>54</b>
3.2 BACK-END	56
<b>3.2.1 Prototipação da biblioteca Psychopy</b>	<b>56</b>
<b>3.2.2 Construção de Arquitetura back-end</b>	<b>61</b>
3.3 GERAÇÃO DE RELATÓRIOS	64
<b>4 VALIDAÇÃO DO SISTEMA</b>	<b>70</b>
<b>5 CONCLUSÃO</b>	<b>72</b>
5.1 DIFICULDADES ENCONTRADAS	73
5.2 TRABALHOS FUTUROS	73
<b>REFERÊNCIAS</b>	<b>74</b>
<b>ANEXO A - AVALIAÇÃO MÉDICA DO APLICATIVO</b>	<b>80</b>

## 1 INTRODUÇÃO

O ser humano está sujeito a diversas formas de estímulos diariamente, de forma a serem compreendidos majoritariamente por meio de seus cinco sentidos primários. Dentre eles, a visão é entendida como o sentido que recebe diferentes estímulos provenientes de luminosidade por meio de fotorreceptores<sup>1</sup>, que serão então decodificados para imagens compreensíveis mediante pulsos neurais (BICAS, 2002).

Para garantir a saúde e qualidade da visão, são realizados exames médicos oftalmológicos de rotina para avaliação da acuidade visual, descrita segundo Bicas (2002) como testes realizados em pacientes para medir sua aptidão óptica, analisando a performance de suas percepções para distinguir contrastes e formas.

Ainda mais, tais exames clínicos são responsáveis por diagnósticos iniciais de problemas oculares. Testes de sensibilidade ao contraste avaliam o nível de percepção que o indivíduo alcança ao ser exposto a imagens com diferentes graus de cores e sombras. Atuam na função de identificar antecipadamente sintomas de possíveis doenças como glaucoma e esclerose múltipla, ou de lesões nas órbitas oculares e seu conjunto (KJAER *et al.*, 2000).

Portanto, a sensibilidade ao contraste (SC) se traduz na capacidade de reconhecer pequenas diferenças na intensidade da luz ou diferenciar dois objetos distintos um do outro. Com isto, a SC é uma parte importante relacionada à muitas atividades da vida diária por possuir grande participação na visão funcional, de tal forma que medi-la é uma das melhores maneiras de avaliar a qualidade da visão (KARA *et al.*, 2016).

Assim, neste trabalho serão abordados os diferentes tipos de testes utilizados na determinação de sensibilidade ao contraste para então demonstrar a utilização do padrão escolhido, CSV-1000E. A família CSV-1000 é uma padronização para medição precisa de uma variedade de distúrbios oculares, dentre os citados anteriormente (VECTORVISION, 2021).

A partir da CSV-1000 derivam-se tipos de testes que possuem especificidades mais apropriadas para determinação de diferentes distúrbios. O CSV-1000 RS é o mais utilizado para triagem de pacientes de cirurgia refrativa, o CSV-1000S e CSV-1000SLanC para avaliação de catarata, e o CSV-1000E sendo o mais utilizado para a determinação de sensibilidade ao contraste, pois por intermédio dele é possível avaliar diferentes tipos de doenças oculares (VECTORVISION, 2021).

---

<sup>1</sup> Receptores sensoriais responsáveis pela visão.

Associando os testes de SC ao uso da tecnologia, são utilizados equipamentos médicos para garantir diagnósticos mais precisos de pacientes. No desenvolvimento deste trabalho, usa-se como base o aparelho hospitalar Optec® 6500 como parâmetro a seguir, este definido por Hohberger *et al.* (2007) como dispositivo que permite, com maior precisão em seu sistema de testes, fornecer o parecer mais exato quanto aos testes de acuidade visual e SC, para posteriores análises mais aprofundadas do estado da visão em questão.

O avanço da tecnologia na área médica pode ser demonstrado também pelo crescente número de aplicações móveis relacionadas a esta área. Desta maneira, aplicações oftalmológicas tornam aparelhos móveis mais próximos de dispositivos médicos. No ano de 2013, 621 diferentes tipos de aplicações médico cirúrgicas para *smartphones* podiam ser contabilizadas, destas, 342 diferentes aplicações relacionadas à área oftalmológica (HASSANI *et al.*, 2013), outras 121 dedicadas ao uso oftalmológico por pacientes e 6 destes destinados a testes visuais (KULENDRAN *et al.*, 2014). Um número que certamente vem crescendo, o que acarreta em um cenário benéfico para a intenção de reduzir custos de exames oculares.

De acordo com Quigley e Broman (2006), aproximadamente 80 milhões de pessoas seriam afetadas por glaucoma no mundo no ano de 2020, das quais cerca de 11 milhões teriam perdido a visão. No Brasil existem poucas estatísticas que demarcam o quanto a doença afeta a população (SCHELLINI *et al.*, 2009), entretanto existem indicativos de que as principais causas dessas doenças são erros refrativos não corrigidos, catarata, degeneração macular e glaucoma (ARIETA *et al.*, 2003).

A identificação precoce de possíveis doenças hereditárias ou em estágio inicial se torna imprescindível no combate à perda de visão. Estudos ainda mais frequentes se especializam em métodos modernizados de prevenção e identificação das doenças, categorizados como um estudo emergente, a “medicina P4” (personalizada, preditiva, preventiva e participativa) (BURKE, 2014 apud GREEN; VOGT, 2016). A qual aplica a integração da análise médica com conjuntos de dados, por meio de simulações computacionais, que são utilizadas como base de pesquisa para resolução de casos. Para assim, calcular e quantificar resoluções positivas de casos que possam ser aplicados em situações tangíveis. Sendo assim, o foco deste trabalho reside na procura e estudo de novos meios de integrar as tecnologias atuais do ramo da computação ao progresso do ambiente médico hospitalar. Propondo o desenvolvimento, por meio de um sistema móvel<sup>2</sup>, que seja

---

<sup>2</sup> Passível de ser movido; móbil, movediço, movente (LANGUAGES; GOOGLE, 2021).

capaz de gerar os testes que se assemelham aos providos pela Optec® 6500 e seu teste de SC citado, seguindo o padrão CSV-1000E.

## 1.1 SITUAÇÃO PROBLEMA

A proposta deste trabalho dar-se-á a partir da relação socioeconômica brasileira, a qual foi abordada por Oftalmologia (2015a) e está relacionada intrinsecamente com as condições da saúde ocular. Além disso, fatores como a idade tornam a estimativa de cegueira crescente, a qual pode chegar a 30 vezes mais em pessoas acima de 80 anos em relação aos que estão na faixa de 40 anos.

Desta maneira, tendo em vista também, mas não somente, a dificuldade financeira de se adquirir e manter máquinas como estas, em referência ao modelo Optec® 6500, é proposto por meio deste estudo uma aplicação móvel com o intuito de contornar o óbice na obtenção de tais máquinas por vezes caras. De acordo com Optical (2021), a fabricante, \$14.332,00 (dólares Estadunidenses) por unidade que, com a conversão atual (04/03/2021) para o real, são aproximadamente R\$81.023,00 (reais brasileiros).

Outro agregante são nações que contém uma quantidade considerável de sua população localizada em zona rural, tal como países da América Latina, incluindo o Brasil, com aproximadamente 16% de seus habitantes localizados fora dos centros urbanos (OLIVEIRA, 2017). Indivíduos nessas áreas, tendem a não possuir tanto acesso a atendimento médico quanto a população que vive em área urbana.

Por meio de pesquisas realizadas em cidades rurais do Peru, foi constatado que aproximadamente apenas um terço do escopo estimado de habitantes possuidores de doenças oftalmológicas foi tratado, em comparação com o percentual esperado por entidades como a OMS<sup>3</sup> (ORGANIZATION, 1997 apud ÁGUILA *et al.*, 2004). Igualmente, outro fator apontado como relevante faz jus à falta de conhecimento prévio de tais problemas oftálmicos (ÁGUILA *et al.*, 2004).

A falta de disponibilidade de atendimento médico que a população de tais regiões enfrenta, somado ao difícil acesso à determinadas localidades por equipes médicas, são fatores para a criação de projetos beneficentes que visam o atendimento e tratamento nas situações citadas. A exemplo, o projeto Mais Acesso à Saúde Ocular, criado pelo Conselho Brasileiro de Oftalmologia (CBO), destinado à populações residentes do interior do país (OFTALMOLOGIA, 2015b).

---

<sup>3</sup> Organização Mundial da Saúde, agência especializada em saúde e subordinada à Organização das Nações Unidas (ONU).

Em suma, o trabalho de pesquisa realizado faz-se necessário, visando a busca por novos métodos que ofereçam auxílio a diagnósticos médicos oftalmológicos. Dispondo-se de tecnologia para tornar mais acessível e dinamizado o estilo de exame proposto, além de barateamento dos custos e facilidade de transporte, visto que a proposta envolve a aplicação móvel.

## 1.2 OBJETIVOS DO ESTUDO

### 1.2.1 Objetivo geral

Esta monografia visa como objetivo geral desenvolver um software móvel para a avaliação ocular por meio da experimentação de sensibilidade ao contraste.

### 1.2.2 Objetivos específicos

- Identificar métodos utilizados para as experimentações do chamado teste de sensibilidade ao contraste.
- Projetar as formas de integrar o teste à aplicação.
- Arquetetar um protótipo capaz de reproduzir imagens de teste semelhantes ao teste da máquina Optec® 6500.
- Passar por validação médica especializada na área de oftalmologia.

## 1.3 METODOLOGIA DA PESQUISA

Inicialmente, realizou-se uma pesquisa com relação aos tipos de exames oftalmológicos existentes. Em conjunto com uma médica oftalmologista, foi observado que o exame CSV-1000 apresentou melhor acessibilidade perante a sua simulação para o estudo desta monografia, devido ao fato de ser o tipo de exame realizado pela médica. Com base nisso, houve um estudo do exame de sensibilidade ao contraste, para definir a possibilidade de aplicação deste em um sistema móvel.

O projeto apresentou a necessidade de realização de pesquisas para constituir o embasamento teórico, visto a sua abrangência e uma área de conhecimento externa à computação, a medicina, e realizou-se também análises sobre aspectos de tecnologias aplicadas ao ramo da oftalmologia. Além disso, foram buscadas informações estatísticas a respeito de doenças oculares que comprovasse a necessidade de facilitar o acesso aos exames.

Durante a construção do trabalho, houve a necessidade de pesquisas em ramos da área de desenvolvimento de sistemas, sendo elas: a prototipação, *back-end* e *front-end*. Ao fim do processo de desenvolvimento, a aplicação foi apresentada para averiguação da médica oftalmologista que acompanhou sua construção. Com base na avaliação da proposta do sistema, o projeto apresentou um futuro promissor como meio de triagem, em conjunto aos exames já existentes.

#### 1.4 ESTRUTURA DO TRABALHO

O trabalho apresenta sua cronologia dividida em capítulos, sendo o primeiro capítulo a apresentação do estudo da monografia, a problematização e seus objetivos. No segundo capítulo, são abordados os aspectos teóricos presentes no trabalho, averiguando a possibilidade de criação do sistema por meio de dados referentes ao setor de oftalmologia e seu alinhamento ao ramo da tecnologia, junto com a ideia da portabilidade de sistemas móveis em *smartphones*.

No terceiro capítulo, são abordados os conceitos e o desenvolvimento do trabalho, referente aos ramos da criação de uma aplicação, por meio da prototipação, construção do *back-end* e do *front-end*. No quarto capítulo são apresentados os resultados perante as simulações do sistema junto à sua avaliação perante profissionais do ramo da saúde ocular. E no quinto capítulo, as considerações finais somado aos próximos passos do projeto para sua atuação em ambientes clínicos.

## 2 FUNDAMENTAÇÃO CIENTÍFICA

### 2.1 OFTALMOLOGIA E TECNOLOGIA

Neste capítulo são apresentados pontos de relevância que constituem a base teórica deste trabalho, abordando conceitos fundamentais relacionados a oftalmologia e ao desenvolvimento de *software*.

#### 2.1.1 Cenário atual

Como é possível observar na análise de Kara-Junior (2018), nos últimos 20 anos as faculdades de medicina no Brasil triplicaram mas, apesar disso, o número de vagas para programas de residências de ensino especializado não acompanhou tal aumento. Devido a esta diferença entre a quantidade de recém formados que conseguem vagas para a especialização e a dos que não encontram vagas, estes então seguem para atuar clinicamente.

[...] ingressam anualmente no mercado da saúde ocular quantidade significativa de médicos tecnicamente despreparados [...] triando casos de doenças para sub-especialistas tratarem. Ou seja, fazem exatamente o que os dispositivos móveis farão (KARA-JUNIOR, 2018).

Com base nas afirmações de Kara-Junior e na pesquisa realizada por Carvalho (2012) é possível observar uma crescente integração entre a medicina e a computação, sendo a oftalmologia há algum tempo um dos campos da medicina que mais evolui em conjunto com a tecnologia, desde metade do século XIX (KEANE; TOPOL, 2019).

Esta evolução vem acontecendo e tornando profissionais potencialmente mais qualificados em sua formação, assim como auxilia na diminuição de erros durante avaliações clínicas, reforçando a avaliação médica.

Devido a estes avanços, muito é esperado quando se refere à tecnologia em conjunto com a medicina, e seus avanços não se contém apenas em áreas isoladas. Citado por Huang e Zhang, tecnologias atuais como a impressão 3D auxiliam no aprendizado de alunos, como em simulações de técnicas cirúrgicas em modelos anatômicos, recriando estruturas complexas e delicadas como a dos olhos humanos. E igualmente para avanços futuros como impressões de dispositivos portáteis customizados para profissionais da saúde, com menor custo.

Outro termo e ascendência que reflete o uso cada vez mais contínuo de aparelhos eletrônicos em colaboração com diagnósticos são os Sistemas de Suporte à Decisão, do inglês *Decision Support Systems* (DDS), sistemas criados para auxiliar médicos na tomada de decisões clínicas, diagnósticos e tratamentos, a partir de bancos de dados coletados e consultados (TORRE-DÍEZ *et al.*, 2015).

Ao longo de pesquisas aprofundadas por Torre-Díez *et al.*, aplicativos móveis focados na área da oftalmologia possuem um grande potencial a ser explorado para empregar o método DDS, por serem acessíveis de forma ágil, visto que o profissional pode consultá-lo por meio de aplicativos em seu celular. Benefícios vistos para os profissionais de saúde, e trazendo a praticidade em dispositivos de posse comum na atualidade.

Diante dessa evolução tecnológica, as inovações vêm acompanhando como forma de explorar todas as possibilidades de avanço científico. Trazendo novos métodos de exames, análises, estudos, a fim de expandir a busca por conhecimento. Não somente em máquinas de fins específicos, mas em aparelhos de uso rotineiro como *smartphones*, presentes no dia a dia na residência de mais de 93% dos brasileiros (IBGE, 2018).

### 2.1.2 O exame oftalmológico

Seja para fins de predição ou determinação de doenças como catarata e glaucoma, o teste de sensibilidade ao contraste é um dos testes realizados para definição de diagnóstico. Para entender o funcionamento do teste é necessário abordar a definição em que ele se baseia, a determinação de contraste.

Uma definição geral de contraste espacial é que é uma dimensão física referindo-se à transição claro para escuro em uma borda ou borda de uma imagem que delinea a existência de um padrão ou objeto (OWSLEY, 2003) <sup>4</sup>.

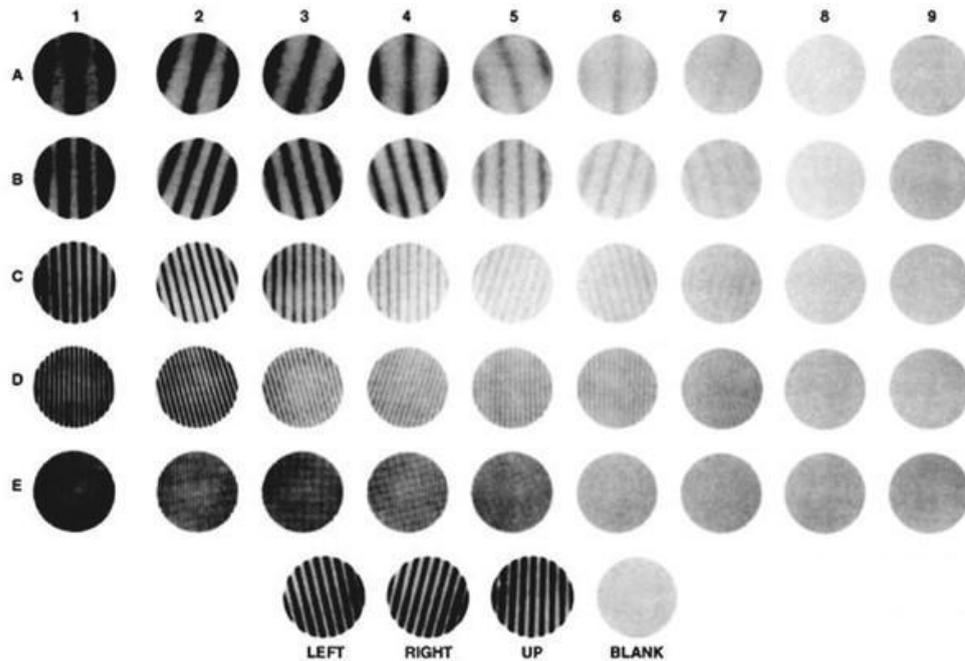
Seguindo a definição abordada por Owsley (2003), pode-se saber então a quantidade de contraste que uma pessoa necessita, nomeado de limite de contraste (*contrast threshold*). Tal limite pode ser utilizado para determinações cotidianas, como a detecção de objetos ou formas, a discriminação de relevos determinando se são iguais ou diferentes, o reconhecimento se algo é familiar ou a identificação do que está no campo de visão.

O limite de contraste faz parte do teste de sensibilidade ao contraste. Tais testes consistem em imagens geradas por computador em conjunto com um *software* capaz de controlar o limite de forma dinâmica, tais imagens apresentam formato vertical, com grades de onda senoidal, sendo que as lacunas entre as barras apresentam variação sinusoidal.

---

<sup>4</sup>sdwa A general definition of spatial contrast is that it is a physical dimension referring to the light – dark transition at a border or an edge of an image that delineates the existence of a pattern or object.

Figura 1: Tabela de grades apresentadas no exame



Fonte: Sociedade Francesa de Oftalmologia. Disponível em: [Œdèmes maculaires](#)

Na Figura 1, é possível observar as formas presentes no espectro do exame de sensibilidade ao contraste, sendo separado em colunas numeradas de 1 a 9, e linhas com padrão alfabético, da letra A até E. O paciente, ao observar as imagens de forma isolada, necessita distinguir para que lado o ângulo das retas está inclinado, para ao fim, ser realizada a análise das respostas e geração do diagnóstico.

Segundo Coelho (2019), a função de sensibilidade ao contraste (FSC), é caracterizada como o inverso da curva de limiar de contraste ( $1/FSC$ ), sendo definida como a diferença entre as variações de ondas. Tal diferença é necessária para o sistema visual humano distinguir impulsos de frequências espaciais em um campo de densidade e intensidade advindos de um ponto de luz.

Seguindo o trabalho de Coelho (2019), o diferencial no uso do FSC seria devido ao seu desempenho aos diversos níveis de contraste, em diferença com outros tipos de médias, como a acuidade visual, utilizada para diferenciar aspectos minuciosos de objetos.

Para realização do uso do FSC, é necessário utilizar um sistema computacional que permita a elaboração do exame de sensibilidade ao contraste. É notório que existem vários tipos de sistemas que possibilitem a confecção de tal exame, porém, o sistema escolhido, devido ao fácil acesso para comparação de metodologia, é o Optec® 6500 (Figura 2).

Figura 2: Sistema Optec® 6500



Fonte: Lista de produtos DOTmed. Disponível em: Site DOTmed

Utilizando o manual oficial do sistema oftalmológico Corporation (2002), Optec® 6500, é possível analisar os processos de realização do exame em pacientes. Durante a fase inicial do exame, é necessário preencher o cadastro do paciente, informando características como: nome, data de nascimento, número de identificação do atendimento e observações. O sistema também apresenta a possibilidade de utilizar cadastros existentes, para repetição do exame com um paciente, podendo definir tipos de exames como o FACT (*Contrast Sensitivity Testing*), o ETDRS (*Early Treatment Diabetic Retinopathy Study*) e correção visual.

Seguindo o manual feito por CORPORATION, op. cit., p.11, para dar início ao exame de sensibilidade ao contraste, é necessário realizar testes monoculares, de forma separada, e a seguir efetuar o teste binocular, isto é, com os dois olhos abertos, totalizando três ciclos de exame. Em cada exame são utilizados gráficos referentes à Figura 1, onde o paciente utiliza um controle remoto presente no dispositivo para indicar a direção do topo de cada grade, começando pela linha "A". Seus resultados são referenciados como R (*Right*), direita, U (*Up*), para cima e L (*left*), esquerda. Deste modo, a indicação de acertos do paciente são referenciados na cor azul e de erros na cor cinza, como pode ser visto na tela do operador no formato da Figura 3.

Figura 3: Tabela de respostas do paciente

	1	2	3	4	5	6	7	8	9
A (15)	L	U	R	U	L	R	L	U	R
B (9)	R	L	R	L	R	U	L	R	U
C (6)	U	R	L	U	R	U	R	L	R
D (12)	L	R	U	R	U	L	U	R	L
E (18)	U	L	R	U	L	R	U	R	U

Fonte: EyeView Instruction Manual. Disponível em: Manual Optec® 6500

Com base no documento CORPORATION, op. cit., p.12, após a realização do exame, é apresentado para o médico examinador os pontos referentes ao acerto do paciente nas colunas ao lado da tabela do exame, como pode ser visto na Figura 4.

Figura 4: Tabela de respostas com acertos do paciente

	1	2	3	4	5	6	7	8	9	Palavras Score	CS Score
A (15)	L	U	R	U	L	R	L	U	R	7	50
B (9)	R	L	R	L	R	U	L	R	U	7	90
C (6)	U	R	L	U	R	U	R	L	R	5	45
D (12)	L	R	U	R	U	L	U	R	L	4	22
E (18)	U	L	R	U	L	R	U	R	U	4	12

Método I

Fonte: EyeView Instruction Manual. Disponível em: Manual Optec® 6500

Como conteúdo final apresentado pelo sistema, é mostrado no manual CORPORATION, op. cit., p.13, o relatório de diagnóstico referente ao exame, exibindo informações baseadas em cada uma das imagens do questionário apresentado pelo sistema Optec® 6500.

Figura 5: Relatório do exame de sensibilidade ao contraste

SUMMARY							
Eye ID	Contrastion No.	Humanisation Prototype ID	Class Name	Letter Correct %	Visual Acuity (20/20)	Examiner	
Eye ID	Contrastion No.	Humanisation Prototype ID	Class Name	Letter Correct %	Visual Acuity (20/20)	Examiner	
		Facet Score	Contrastion Score				
Row A (1.5)	5	25	5	25	5.0	25.0	
Row B (3)	5	40	5	40	5.0	40.0	
Row C (6)	5	80	5	80	5.0	80.0	
Row D (12)	5	45	5	45	5.0	45.0	
Row E (24)	5	35	5	35	5.0	35.0	
Eye ID	Contrastion No.	Humanisation Prototype ID	Class Name	Letter Correct %	Visual Acuity (20/20)	Examiner	
		Facet Score	Contrastion Score	Facet Score	Contrastion Score	Average Facet Score	Average Contrastion Score
Row A (1.5)	5	25	5	25	25	5.0	25.0
Row B (3)	5	31	5	40	35	5.2	45.0
Row C (6)	5	42	5	45	45	5.7	77.5
Row D (12)	5	80	5	80	80	5.0	80.0
Row E (24)	5	47	5	47	47	5.0	49.0
Eye ID	Contrastion No.	Humanisation Prototype ID	Class Name	Letter Correct %	Visual Acuity (20/20)	Examiner	
		Facet Score	Contrastion Score	Facet Score	Contrastion Score	Average Facet Score	Average Contrastion Score
Row A (1.5)	5	25	5	25	25	5.0	25.0
Row B (3)	5	40	5	40	40	5.0	40.0
Row C (6)	5	47	5	47	47	5.0	47.0
Row D (12)	5	30	5	33	33	4.5	33.0
Row E (24)	5	17	5	17	17	4.0	14.0

Fonte: EyeView Instruction Manual. Disponível em: Manual Optec® 6500

Estudando o funcionamento do *software* apresentado pelo sistema Optec® 6500, é possível traçar um referencial para ser utilizado como base na construção da arquitetura do sistema móvel proposto.

### 2.1.3 Tecnologia atual em smartphone

De acordo com pesquisas realizadas pela Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas (FGV EAESP), em 2020 calculou-se que mais de 230 milhões de smartphones foram usados no Brasil (MEIRELLES, 2020). Tal dado traz uma perspectiva da relevância que engloba o seu uso nos dias atuais, pois por meio desses dispositivos, usuários são capazes de comunicar-se entre si, fazer compras, pesquisar notícias, e utilizar diferentes aplicativos.

Devido ao grande avanço tecnológico alcançado por meio da crescente miniaturização de componentes e a capacidade de produção que indústrias 4.0 fornecem, é possível tornar aparelhos portáteis em verdadeiros computadores portáteis. Estes avanços também sustentam a quantidade de informação que é possível obter com *smartphones*, se comparada à abundância de dados que a utilização do *IoT*<sup>5</sup> permite coletar.

A evolução surpreendentemente rápida da qualidade da tecnologia dos chamados micro computadores nos últimos 20 anos, também mudou o modo como eles são vistos pelos usuários. A comparação pode ser feita observando modelos de celulares lançados na década de 1990, realizados no início da conexão 2G, a celulares atuais com telas com tecnologia

<sup>5</sup> O Internet of Things (IoT) descreve a rede de objetos físicos, “coisas”, que são incorporados a sensores, *software* e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela internet (ORACLE, 2021).

*touchscreen*, displays AMOLED, reconhecimento digital e facial, *bluetooth*, sistemas RFID, câmeras que passaram de 0,11 para mais de 60 megapixels (DUDLEY, 2018). E nesse cenário é possível averiguar como em pouco tempo houve inovações significativas.

Em complemento, o processo de surgimento do padrão 5G também aponta o crescimento do uso dos equipamentos portáteis, visto que em suas fases iniciais de implantação, propõe o aumento do número de conexões entre máquinas, muito usado em *IoT*, a melhora da velocidade das conexões, bem como a melhora na qualidade de transferências e utilização de dados (VALENTE, 2020). E desse modo, os *smartphones* vêm sendo vistos agora não somente como ferramentas de comunicação, mas também para utilidades que agregam atividades do cotidiano e canais de conhecimento.

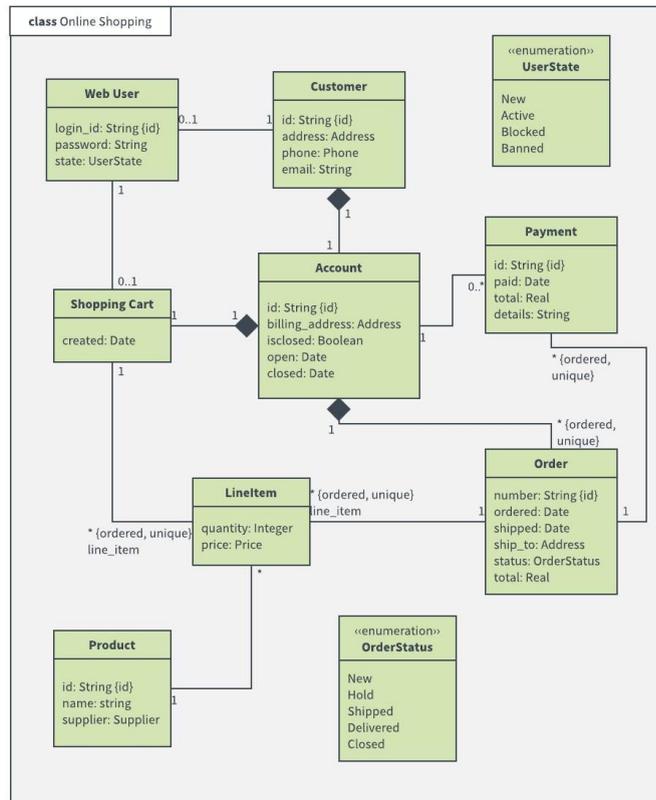
Adaptando-se a diferentes meios de aprendizado, a tecnologia dos *smartphones* também se mostra crescente e promissora no âmbito acadêmico. Em busca de inovações em métodos de ensino e para adaptar-se aos tempos atuais, a pesquisa feita por Morales *et al.*, realizada analisando estudantes e professores da Universidade Sergio Arboleda, indica que mais de 50% dos docentes próximos à idade de 30 anos já utilizam aplicativos móveis em suas práticas de ensino e, conseqüentemente, aproximadamente 50% dos estudantes também fazem o uso dos aplicativos e não encontram maiores dificuldades no manuseio. Em suma, há a confirmação de que aproximadamente 80% dos alunos consideram o uso de aplicativos e outras tecnologias móveis benéficas para o aprendizado e incentivo dentro de sala de aula.

#### **2.1.4 Arquitetura de software**

Para um desenvolvimento coeso de pequenas ou grandes aplicações, é necessária a implementação de uma arquitetura que sirva bem o propósito ao qual a aplicação está sendo projetada. A arquitetura está diretamente relacionada com a organização geral do sistema a ser implementado, afetando desta maneira os requisitos funcionais, sendo alguns destes, confiabilidade, portabilidade, disponibilidade, desempenho entre outros (FILHO, 2008).

Como todo projeto, uma boa escolha dos métodos arquitetônicos é determinante para o sucesso do sistema, desta forma padrões são desenvolvidos para auxiliar no planejamento da arquitetura. O modelo UML (*Unified Modeling Language*) é uma das linguagens que auxiliam na concepção de projetos de *software*. E por se tratar de uma linguagem muito abrangente, é possível planejar com clareza os sistemas abordando todas as necessidades ao se desenvolver e implementar o sistema em questão (FAKHROUTDINOV., 2020; GROUP, 2011; SILVA *et al.*, 2004).

Figura 6: Exemplo de modelo UML



Fonte: Diretório de exemplos do Lucidchart. Disponível em: Lucidchart

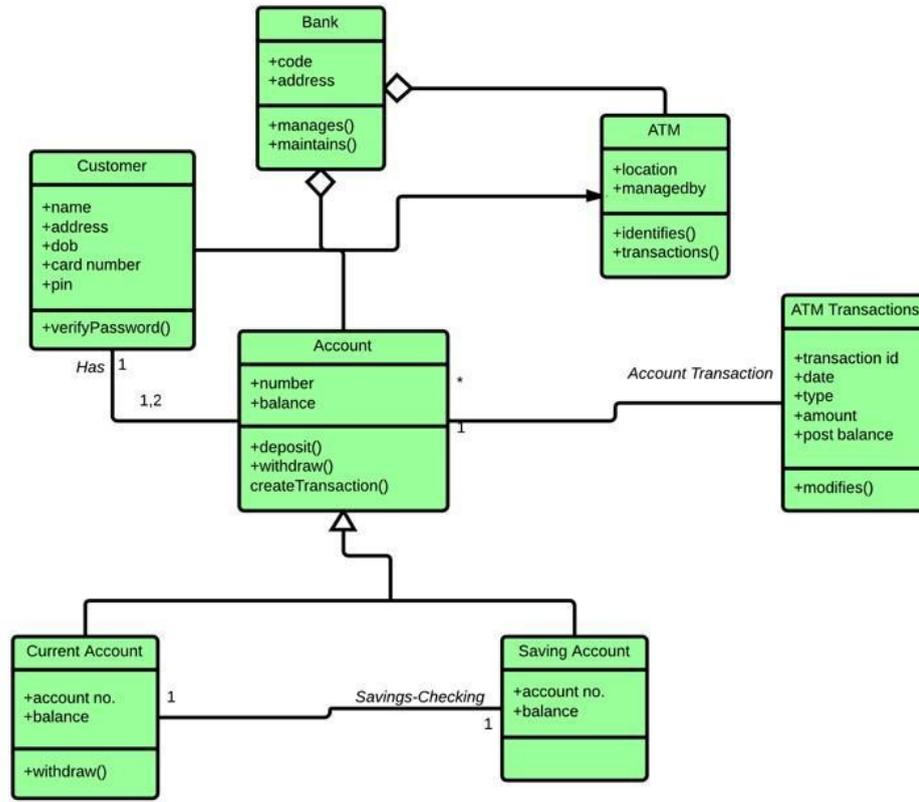
Como é possível observar na Figura 6 um modelo UML que demonstra a interconexão entre diferentes informações e como as mesmas se relacionam, deixando assim clara a relação existente antes do desenvolvimento do sistema e também servindo como guia durante a construção do mesmo.

O UML como modelo possui diretrizes que levam à criação e utilização de diagramas próprios que se dividem em duas grandes áreas principais, a dos diagramas estruturais e dos diagramas comportamentais ou dinâmicos.

#### 2.1.4.1 Diagrama de classes

Diagramas de classes são parte integrante do modelo, uma vez que constituem a base para fazer a representação de classes de sistemas, controle, negócios ou interfaces e outros demais sistemas. O diagrama, portanto, é um dos mais importantes para o modelo UML, utilizado como apoio para os demais tipos de diagramas que fazem parte deste modelo, servindo então como a representação de classes que serão implementadas no código fonte, juntamente com seus princípios e interações com os objetos (LUCIDCHART, c2021b).

Figura 7: Diagrama de classes para sistema de caixa eletrônico



Fonte: O que é um diagrama de classe UML. Disponível em: Lucidchart

Ainda de acordo com Lucidchart (c2021b), e como é possível observar na Figura 7, pode-se definir a representação das classes no diagrama por meio de um retângulo com três linhas. A superior contendo o nome desta classe, a que fica ao meio, contém os atributos da classe em questão, por fim a inferior, representa os métodos ou operações que serão utilizadas na classe. Classes e subclasses são representadas em agrupamentos conjuntos com o intuito de demonstrar a relação existente entre cada objeto.

As regras e definições para construção de diagramas de classe vão ainda além de somente representações visuais dos arranjos entre cada classe/objeto. Com o diagrama é possível determinar previamente os tipos de acesso para cada objeto durante o curso do desenvolvimento do sistema e sua tipificação de acesso dentro do código fonte, com a representação simbólica que o modelo utiliza (UFCG, 2021).

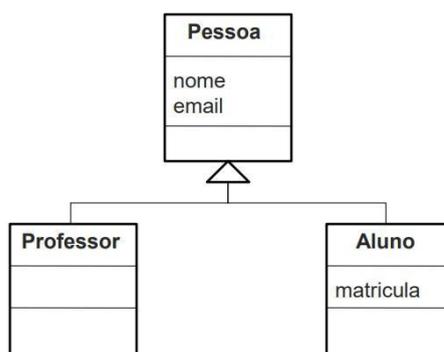
- Público (+);
- Privado (-);
- Protegido (#);

- Pacote (~);
- Derivado (/);
- Estático (sublinhado).

Indo além dessas tipificações de acesso, com o diagrama é possível ainda especificar os tipos de dados para cada variável previamente. Essa determinação em conjunto com a representação por meio de sinais, símbolos, representa a comunicação que pode ser assíncrona e unidirecional entre os objetos ativos (FIGUEIREDO, 2016).

Outro fator muito importante durante a programação e estruturação de código, são as interações, podendo ser vistas na Figura 8. Tais relações podem existir entre as classes, e dentro do modelo UML são representadas de maneiras diferentes e precisas Ibid..

Figura 8: Relações do diagrama de classes



Fonte: Relacionamentos do Diagrama de Classes. Disponível em: Eduardo Figueiredo - UFMG

Uma dessas interações é a hereditariedade <sup>6</sup>, representada por meio de “[...] uma linha conectada reta com uma ponta de seta fechada apontando para a superclasse” (FIGUEIREDO, 2016).

Continuando com outras interações comumente utilizadas, são encontrados dois tipos de associações direcionais, uma unidirecional e outra bidirecional. Pode-se caracterizar a unidirecional como “uma relação um pouco menos comum entre duas classes”. Uma classe tem conhecimento da outra e interage com ela” (LUCIDCHART, c2021b). Por sua vez, a associação unidirecional é representada com uma linha reta que liga as duas classes com a ponta de seta aberta apontando da classe conhecimento à classe conhecida. A associação bidirecional entretanto é uma relação mais padrão entre duas classes distintas, onde esta e

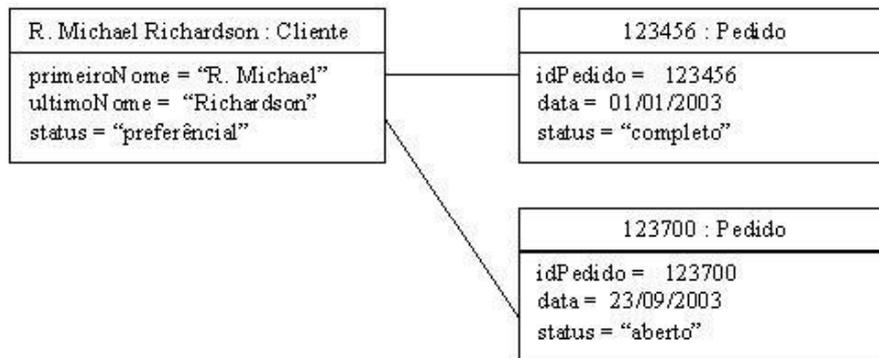
<sup>6</sup> Também conhecido como uma generalização, um processo secundário, subclasse que assume a funcionalidade de um primário ou de uma superclasse.

aquela estão cientes de suas contrapartes. Esta associação se representa com uma linha reta entre as duas classes (LUCIDCHART, c2021b; UFCG, 2021).

#### 2.1.4.2 Diagrama de objetos

Este diagrama é a modelagem das instâncias de classes que estão contidas no diagrama de classes, abordado em 2.1.4.1. Sendo assim, o diagrama é responsável por exibir o conjunto de relacionamentos que os objetos realizam no tempo. Tais diagramas são importantes construtos para aspectos estáticos do sistema (RIBEIRO, 2004b).

Figura 9: Representação do diagrama de objetos



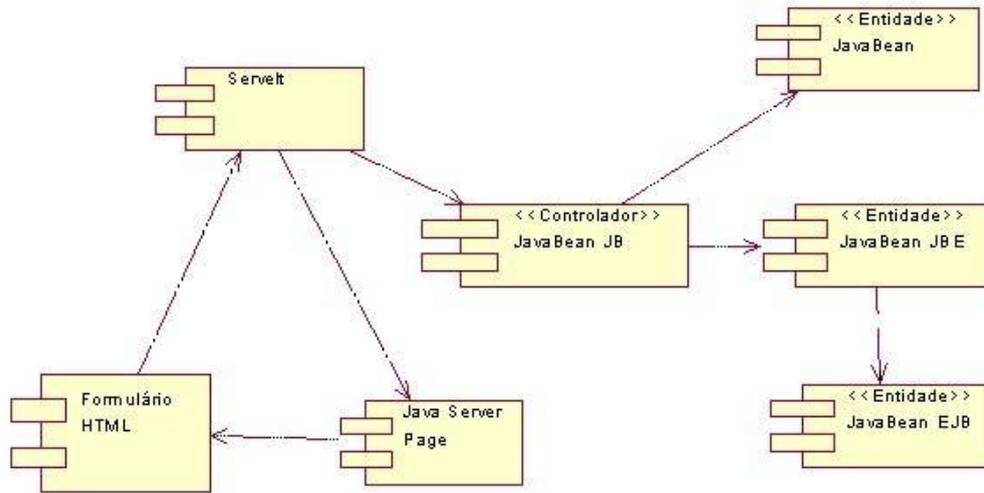
Fonte: Diagrama de Objetos. Disponível em: UFMG

A Figura 9 demonstra o retrato do sistema em um determinado momento, exemplificando os pedidos que o cliente realizou no preciso momento.

#### 2.1.4.3 Diagrama de componentes

O diagrama de componentes é um tipo de modelagem física do sistema, enfatizando a visão dos componentes e seus relacionamentos mútuos. Possuindo um enfoque no empacotamento físico que o código receberá, constituído por: componente, interface e relacionamentos (RIBEIRO, 2004a).

Figura 10: Exemplificação de classes de componentes



Fonte: Diagrama de componentes. Disponível em: Antônio Mendes Ribeiro - UFMG

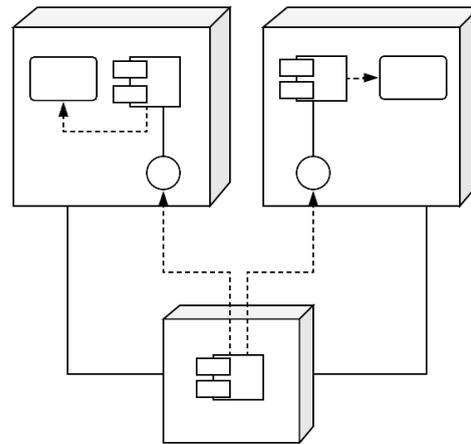
Utilizado geralmente para se definir a configuração dos arquivos que constituem o código fonte dos programas, com especial atenção durante a utilização de ambientes de desenvolvimento integrado (IDE)<sup>7</sup>, além de modelar versões para executáveis de programas e a visão física para o banco de dados, podendo-se observar então, tabelas ou páginas de informações contidas no banco de dados (RIBEIRO, 2004a).

#### 2.1.4.4 Diagrama de implantação

Para além da modelagem de *software*, também é necessário mapear a estrutura responsável por prover a relação entre a infraestrutura, *hardware* e o sistema desenvolvido. Para tal objetivo o diagrama de implantação, é implementado, funcionando por meio de uma associação entre nós, que são os contêineres de informação do modelo UML, representando assim um componente ou item de *hardware*, como um servidor de armazenamento físico ou um servidor onde a aplicação será executada. Doutra forma o nó também é capaz de representar parcial ou totalmente o ambiente de execução do *software* (SILVA, 2016).

<sup>7</sup> Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar este processo.

Figura 11: Formulação do diagrama de implementação



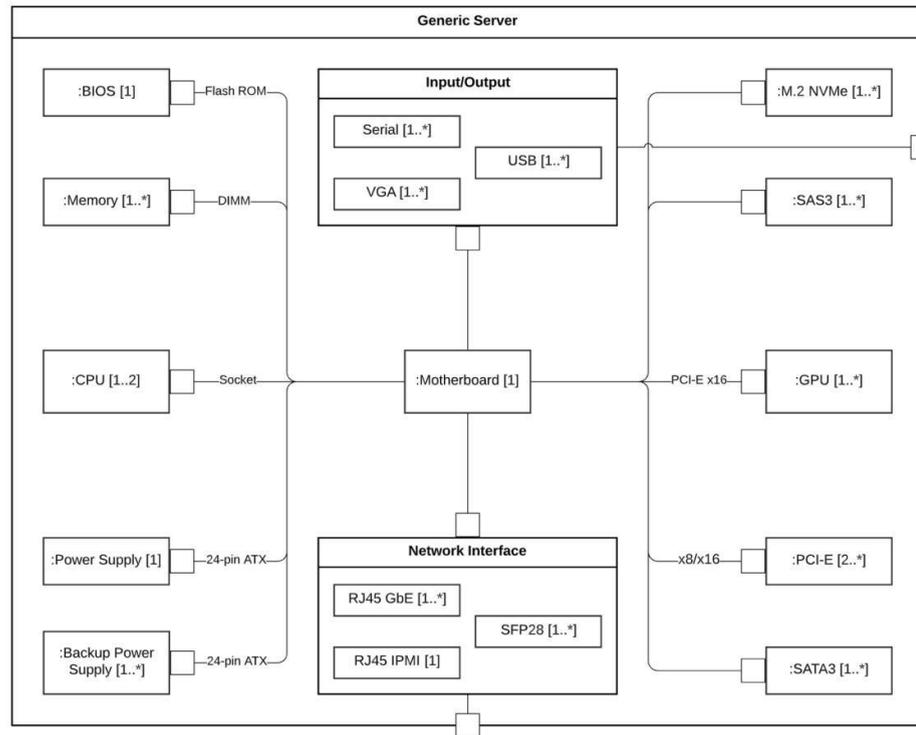
Fonte: Diagrama de implementação. Disponível em: lucidchart

Ainda de acordo com Silva (2016), além de demonstrarem partes importantes de *hardware*, nós também podem conter outros nós, representando consequentemente a total estrutura do sistema. Estas associações representam a forma como as trocas de informações são ou serão realizadas no meio físico, esta relação por sua parte sendo representada no diagrama por meio de retas complementadas por rótulos que descrevem a relação ou a maneira como interagem mutuamente, tal rótulo é nomeado estereótipo. Portanto, durante a implantação de sistemas complexos e distribuídos onde os *hardwares* possuem papéis de suma importância, é possível observar o diagrama como um aliado.

#### 2.1.4.5 Diagrama de estrutura composta

Diagrama de estrutura composta é capaz de oferecer uma visão lógica e geral dentro de um total ou parte de um sistema, o diagrama atua como um classificador capaz de definir as classes de configurações, interfaces, pacotes e relações. Desta maneira permite ao usuário observar as informações contidas no objeto, além de especificar como as propriedades de diferentes objetos se encaixam a fim de produzir um determinado comportamento desejado (LUCIDCHART, c2021c).

Figura 12: Exemplo de servidor utilizando diagrama de estrutura composta



Fonte: Tudo sobre diagramas de estrutura composta. Disponível em: lucidchart

Assim como os demais tipos de diagramas, o de estrutura possui diversos símbolos que representam as relações e utilizam os componentes de formas do UML para estruturar a diagramação.

Existem também semelhanças presentes entre os diagramas de classe e o de estrutura, é possível observar tais semelhanças quando comparando as Figura 7 e Figura 12. Desta forma nota-se que o diagrama composto assim como o de classe (2.1.4.1) são utilizados para visualizar e organizar interações entre atores e artefatos dentro de um sistema.

Tais diagramas contém significados próximos, porém, a maneira como demonstram seus significados difere. Sendo que, os diagramas estruturais são mais específicos e menos ambíguos do que os diagramas de classes (LUCIDCHART, c2021c).

De acordo com Lucidchart (c2021c), a diferença encontrada na forma de exibir seus conteúdos permite ao usuário modelar as implementações de artefatos que estejam dentro de um tempo de execução definido, com mais clareza além de demonstrar melhor como a estrutura interna de várias classes com suas relações mútuas. Desta maneira o diagrama de estrutura é uma opção para transmitir informações concretas e claras sobre relações sistemáticas.

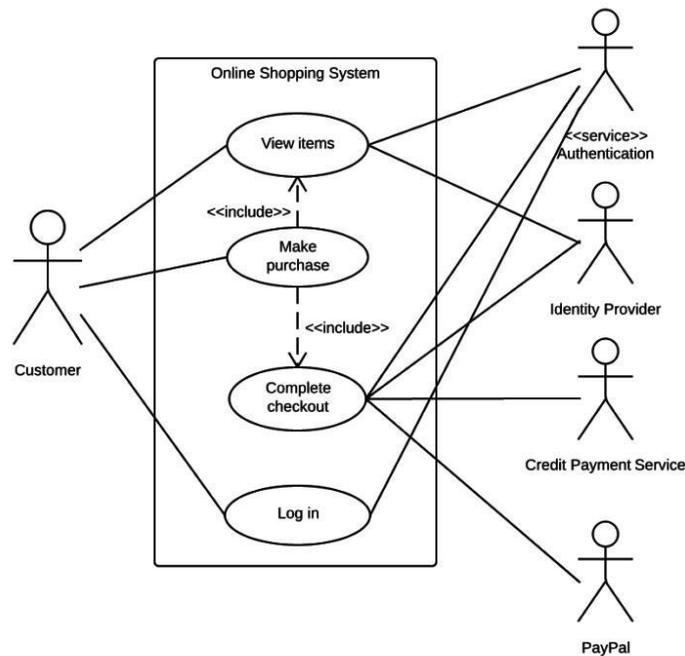
#### 2.1.4.6 Diagrama de caso de uso

Durante a projeção do sistema e o levantamento de requisitos, o diagrama de casos de uso auxilia em descrever o conjunto de interações do sistema com suas funcionalidades. Dessa forma o diagrama resume detalhes de que utilizará o sistema, nomeados dentro do diagrama como atores, e quais interações serão realizadas pelos mesmos (VIEIRA, 2015).

Este diagrama não oferece muitos detalhes, ao contrário dos demais abordados anteriormente, que mostram ordens de execução e suas relações. Ao invés disso, o diagrama, quando utilizado de maneira adequada, fornece uma visão completa sobre o relacionamento interno do sistema com seus atores (VIEIRA, 2015).

Suas representações gráficas de diagramação contém, então, maneiras de representar os diferentes protagonistas do sistema com seu uso sendo adequado para organizar os requisitos funcionais do sistema e especificar o contexto de tais requisitos, além de ser possível modelar o fluxo básico dos eventos de interação (LUCIDCHART, c2021a).

Figura 13: Diagrama de Casos de Uso para sistema de compras online



Fonte: Diagrama de caso de uso UML: O que é, como fazer e exemplos. Disponível em: lucidchart

A Figura 13 representa um exemplo da utilização de casos de uso, onde é possível observar a aplicação dos componentes gerais do modelo para este diagrama, que são: Atores, Sistema e Metas. Cada um destes componentes dispõe de sua representação própria. Atores simbolizam os usuários, os quais interagem com o sistema, esses atores podem ser pessoas,

organizações ou um sistema externo que interage com o sistema sendo desenvolvido. Eles devem ser a representação de objetos externos que utilizam dados ou que os produzem (LUCIDCHART, c2021a).

Sistemas, também chamados de cenários, são sequências de interações entre atores e o sistema que resultam em determinadas ações. Metas são o produto final da interação entre os casos de uso, as quais são usadas para demonstrar ou descrever o resultado das atividades e variantes utilizadas e descritas ao longo do diagrama (LUCIDCHART, c2021a).

A seguir pode-se observar a descrição que Lucidchart (c2021a) fornece para os diferentes tipos de representação gráfica do modelo.

- Caso de uso: o formato deve circular a palavra horizontalmente, representando aspectos de usabilidade do usuário;
- Atores: personificações simplificadas do usuário, retratando as implementações dos casos de uso;
- Associações: ligações que conectam os atores aos casos de uso. Quanto mais complexos os diagramas, maior cuidado deve-se ter com as associações;
- Caixa de limite do sistema: marcação que representa o ambiente do sistema. Porém, o que não estiver na marcação é interpretado como ausente do escopo do sistema.

## 2.1.5 Ferramenta de desenvolvimento

### 2.1.5.1 Python

Python é uma linguagem de programação de alto nível e orientada a objetos, de tipagem dinâmica e forte. Lançada por Guido van Rossum em 1991, Python é desenvolvida sob uma licença de código aberto aprovada pela OSI, tornando-a livremente utilizável e distribuível, mesmo para uso comercial. A licença da Python é administrada pela *Python Software Foundation* (FOUNDATION, 2020).

Foi selecionada para o projeto como linguagem principal para o desenvolvimento do *back-end* pois é uma linguagem de domínio dos autores, e que possui uma facilidade e agilidade durante o desenvolvimento por possuir uma sintaxe simples, sendo a linguagem de programação mais procurada em 2020 (MARTINS, 2020).

### 2.1.5.2 Biblioteca Psychopy

Para a geração das imagens referentes às grades de sensibilidade ao contraste, é necessário utilizar uma biblioteca em Python, chamada PsychoPy.

Apresentando diversas funcionalidades na área de estudos físicos, a biblioteca possui uma variedade de funções para criação de grades radiais, lineares, formatos geométricos, entre outros elementos (GRAY *et al.*, 2019).

[...] PsychoPy é um pacote de código aberto para a execução de experimentos em Python (uma alternativa real e gratuita ao Matlab). O PsychoPy combina as forças gráficas do OpenGL com a sintaxe fácil do Python para dar aos cientistas uma apresentação de estímulo simples e gratuita e um pacote de controle. É usado por muitos laboratórios em todo o mundo para psicofísica, neurociência cognitiva e psicologia experimental. (GRAY *et al.*, 2019).<sup>8</sup>

### 2.1.5.3 Biblioteca Flask

A biblioteca Flask, segundo Team (2021b), é utilizada para a criação de arquiteturas *back-end*, apresenta uma simples construção por não depender de muitas dependências de infraestrutura. Tendo em vista tal simplicidade, a biblioteca possui suporte para extensões de mapeamento objeto-relacional e manipulação de arquivos para realização de uploads. Utiliza como componentes principais, a biblioteca Werkzeug, para realização de solicitações e respostas em serviços *web* e o mecanismo de template Jinja para realização de modelos de sistemas *web*.

### 2.1.5.4 Biblioteca Matplotlib

A biblioteca Matplotlib, segundo Hunter, é utilizada para a confecção de diferentes tipos de gráficos, de forma simples e rápida por meio do uso de vetores, apresentando destaque no uso em áreas como *machine learning* ou estudos que envolvam análise de dados.

Matplotlib é uma biblioteca para criação de gráficos 2D de matrizes em Python. Embora tenha suas origens na emulação dos comandos gráficos do MATLAB, ele é independente do MATLAB e pode ser usado de forma Pythônica, orientada a objetos. [...] O Matplotlib foi projetado com a filosofia de que você deve ser capaz de criar gráficos simples com apenas alguns comandos ou apenas um! Se você deseja ver um histograma de seus dados, não precisa instanciar objetos, chamar métodos, definir propriedades e assim por diante, deve funcionar (HUNTER, 2007).<sup>9</sup>

---

<sup>8</sup> PsychoPy is an open-source package for running experiments in Python (a real and free alternative to Matlab). PsychoPy combines the graphical strengths of OpenGL with the easy Python syntax to give scientists a free and simple stimulus presentation and control package. It is used by many labs worldwide for psychophysics, cognitive neuroscience and experimental psychology.

<sup>9</sup> Matplotlib is a library for making 2D plots of arrays in Python. Although it has its origins in emulating the MATLAB graphics commands, it is independent of MATLAB, and can be used in a Pythonic, object oriented way.[...]Matplotlib is designed with the philosophy that you should be able to create simple plots with just a few commands, or just one! If you want to see a histogram of your data, you shouldn't need to instantiate objects, call methods, set properties, and so on; it should just work.

### 2.1.5.5 Dart

Com relação à interação do usuário com o sistema proposto. Existe a necessidade de utilizar uma linguagem de programação simples e eficaz. Nesse quesito, a utilização da linguagem Dart, se mostrou de forte impacto para o projeto.

[...] Dart é uma linguagem otimizada para o cliente para o desenvolvimento de aplicativos rápidos em qualquer plataforma. Seu objetivo é oferecer a linguagem de programação mais produtiva para o desenvolvimento multiplataforma, emparelhada com uma plataforma de tempo de execução flexível para frameworks de aplicativos (GOOGLE, 2021).<sup>10</sup>

A linguagem Dart, apresenta uma documentação produzida pela equipe de engenheiros da empresa Google, e apresentou um acríve de 4,7% com relação a outras linguagens em 2020, sendo um destaque no uso em ambientes de trabalho focados em multiplataformas (*smartphones*, computadores e aplicações *web*) (OVERFLOW, 2020).

### 2.1.5.6 Flutter

Para a criação de interfaces, é necessário utilizar um *framework*<sup>11</sup> com capacidade de criar uma interface acessível, com pouco impacto no processamento da aplicação. O Flutter em conjunto com a linguagem Dart, constitui um ambiente de desenvolvimento multiplataforma, nesse caso, para o sistema operacional *Android*.

Flutter é um kit de ferramentas de interface de usuário de plataforma cruzada projetado para permitir a reutilização de código em sistemas operacionais como iOS e Android, ao mesmo tempo que permite que os aplicativos façam interface diretamente com os serviços da plataforma subjacente. O objetivo é permitir que os desenvolvedores entreguem aplicativos de alto desempenho que pareçam naturais em diferentes plataformas, abraçando as diferenças onde elas existem e compartilhando o máximo de código possível (BELTRAN, 2021).<sup>12</sup>

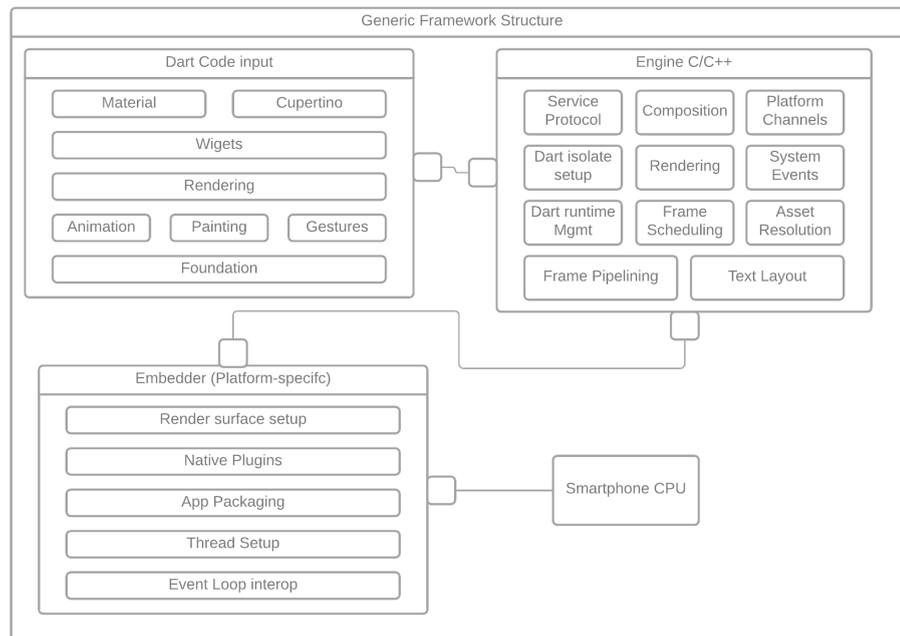
---

<sup>10</sup> *Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.*

<sup>11</sup> Uma estrutura, ou estrutura de *software*, é uma plataforma para o desenvolvimento de aplicativos de *software*. Ele fornece uma base sobre a qual os desenvolvedores de *software* podem construir programas para uma plataforma específica (TECHTERM, 2013).

<sup>12</sup> Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible.

Figura 14: Diagrama estrutural do Flutter



Fonte: Autores (2021)

Sua arquitetura segue uma implementação onde o código em Dart passa por sucessivas iterações e conversões dentro do *framework* a fim de ser repassada para a codificação específica com a plataforma, é possível observar estas integrações na Figura 14.

### 2.1.6 Modelos de prototipação

Dentre as fases que precedem o desenvolvimento de *software*, especificamente na análise de requisitos, está a prototipação, seja ela de interfaces gráficas ou de sistemas sem interface. Entretanto, os tipos de prototipagem a seguir são comumente utilizados para o desenvolvimento de UI's (*User Interfaces*), as interfaces que o usuário interage (DIAS, 2019).

A prototipação é separada em estágios diferentes, com suas respectivas atribuições e diferentes objetivos, entretanto todos os seus estágios servem um só propósito, facilitar o entendimento dos requisitos e alinhar as requisições do cliente com o time de desenvolvedores a fim de evitar a desorientação do desenvolvimento (DIAS, 2019).

A técnica de prototipação tem a finalidade de demonstrar as características de funcionamento e as ideias do sistema por meio de desenhos simples até os mais complexos que demonstram de forma mais fiel a interface final (DIAS, 2019; NEVES, 2017).

Para além de somente facilitar o acordo entre o que foi solicitado e o que será entregue, os protótipos possibilitam demonstrar a interface e o processo de interação entre

usuário e sistema de uma forma visual e inteligível. Assim, ao final da avaliação do protótipo é possível determinar com clareza o que precisa ser modificado e o que permanece na interface.

Objetivando iniciar o design da aplicação, decidiu-se pelo uso da ferramenta Figma como editor de prototipagem para idealizar e esboçar a aparência do aplicativo deste trabalho. Disposta em um navegador web, a ferramenta dispõe de recursos de trabalho colaborativo entre os integrantes, isto é, o arquivo pode ser alterado por mais de um integrante simultaneamente em sua versão mais recente, evitando perda de atualizações ou versões diferentes do mesmo projeto, pelo uso de desenvolvimento em nuvem (NIGRI, 2020).

A plataforma Figma possibilita então a atualização em fases da criação visual do aplicativo, passando pelo processo de desenvolvimento em modelos de prototipagem, divididos e descritos nos três tópicos a seguir.

#### 2.1.6.1 Baixa fidelidade

Os moldes de baixa fidelidade são utilizados nas fases mais iniciais de prototipação, deste modo, esta etapa é comumente conhecida como rascunhos e sketches. Como Dias (2019) descreve, o modelo de baixa fidelidade define de modo simples como seria a interação do usuário com o projeto, não tendo nenhuma preocupação com elementos ligados ao design (DIAS, 2019; NEVES, 2017).

#### 2.1.6.2 Média fidelidade

Protótipos de média fidelidade são feitos com base nos requisitos levantados na análise do modelo de baixa fidelidade, desta forma são mais próximos da representação do projeto final. Nestes modelos já são empregadas técnicas de desenvolvimento UI/UX (Interface de usuário/Experiência do usuário). Desta forma o cliente é capaz de compreender o funcionamento de elementos de forma mais detalhada, uma vez que as interfaces são desenhadas utilizando ferramentas capazes de simular o comportamento de botões e outros elementos (NEVES, 2017; DIAS, 2019).

#### 2.1.6.3 Alta fidelidade

Esse modelo é utilizado para demonstrar uma interface semelhante à final, uma vez que são utilizados os mesmos elementos que serão utilizados no sistema. Assim, estes protótipos são desenvolvidos já utilizando a linguagem de programação escolhida. Desta

maneira algumas partes do sistema já estão presentes durante a apresentação do protótipo (DIAS, 2019).

### 2.1.7 Arquitetura REST

Arquitetura REST (*Representational State Transfer*) representa uma maneira de trafegar dados por meio da internet, sendo baseado na arquitetura de *Web Services*. Como abordado por devmedia (2017), surgiu para facilitar a interoperabilidade entre aplicações que se situam em diferentes localidades e plataformas. Esta por sua vez foi especificada para se adequar ao protocolo SOAP (*Simple Object Access Protocol*) responsável por encapsular mensagens em uma linguagem que é conhecida como WSDL (*Web Services Description Language*). Em resumo, sistemas que se guiam por estes padrões possuem suas mensagens de requisição e resposta envelopadas por meio do protocolo SOAP (DEVMEDIA, 2017).

Por fim, a arquitetura REST se baseia em um estilo arquitetural híbrido para sistemas distribuídos e também sendo uma combinação de estilos arquitetônicos (RODRIGUES, 2009).

#### 2.1.7.1 Null Style

Representando um conjunto simples e vazio de características, descreve um sistema que não distingue limites entre componentes partindo de um ponto de vista arquitetural (RODRIGUES, 2009).

#### 2.1.7.2 Cliente-servidor

Partindo de um conceito fundamental para a arquitetura cliente-servidor está a “separação de responsabilidades” (RODRIGUES, 2009) proveniente da diferenciação de interfaces e de armazenamento de dados. Permitindo um desenvolvimento independente de cada parte facilitando a adaptação e suporte a requisição de múltiplas aplicações dentro da internet (RODRIGUES, 2009).

#### 2.1.7.3 Sem Estado

Segundo a análise realizada por (RODRIGUES, 2009) sobre a arquitetura sem estado (*stateless*), é possível observar que este princípio aplica a definição de que apenas o cliente deve manter informações sobre o comportamento atual do sistema e seu estado, sendo o mesmo responsável por passar atributos necessários para processar as devidas solicitações.

#### 2.1.7.4 Cache

O conceito de cache parte de uma combinação de alguns dos demais conceitos que constituem a arquitetura REST. O Cliente servidor, Sem estado e a própria Cache unidos formam o conceito conhecido como *Cache-Servidor Sem Estado*. Seu intuito é melhorar a eficiência da comunicação entre as partes envolvidas, permitindo que o cliente verifique se existe uma marcação de cache para aproveitar as vantagens do cacheamento (RODRIGUES, 2009).

A principal vantagem sobre modelos anteriores é o poder parcial ou total de eliminar algumas interações que poderiam sobrecarregar o servidor (TOBALDINI, 2008 apud RODRIGUES, 2009).

#### 2.1.7.5 Interface Uniforme

Pode-se dizer com base na assertiva de Fielding (2000 apud RODRIGUES, 2009) que uma característica importante da arquitetura REST em relação às demais é sua ênfase em uma interface uniforme entre os seus componentes.

Por meio do princípio da generalização presente na engenharia de *software* é possível tornar a interface dos componentes em uma arquitetura simplificada e com melhor visibilidade. Entretanto, toda uniformização presente na troca de informação, torna a troca de informação genérica ao invés de se adequar a cada necessidade (TOBALDINI, 2008 apud RODRIGUES, 2009).

#### 2.1.7.6 Separação em camadas

A separação em camadas possibilita a composição de diversos níveis hierárquicos, restringindo os componentes participantes de tal maneira que nenhum deles possa enxergar por meio de camadas adjacentes, em outras palavras, mantendo a restrição a apenas sua camada (FIELDING, 2000 apud RODRIGUES, 2009).

Entretanto estas implementações vem com uma grande desvantagem que é o maior atraso na chegada de informação de um extremo ao outro devido a camadas intermediárias. O estilo REST também adiciona também a possibilidade de separar camadas com uma maior flexibilidade e simplicidade na comunicação entre sistemas diversos (RODRIGUES, 2009).

## 2.2 TRABALHOS RELACIONADOS

*Smartphones* são utilizados como ferramentas de auxílio ou equipamento principal em diversos estudos e pesquisas, Wintergerst *et al.* (2020) demonstram o uso destes aparelhos celulares na identificação de problemas oftalmológicos provenientes de diabetes em um processo de captura de imagens da retina. Neste trabalho, visou-se utilizar métodos mais acessíveis e de menor custo, com equipamentos capazes de serem transportados em distâncias maiores. Com isso, utilizando celulares e softwares de exame, juntamente com equipamentos médicos próprios para a realização do teste de Retinopatia Diabética, os pesquisadores foram capazes de concluir que é possível fazer uso de celulares para realizar o exame médico, com resultados de qualidade para o profissional utilizar em diagnósticos.

Motivados em comparar resultados obtidos por métodos tradicionais de análise de sensibilidade ao contraste, com os de aplicativos para *tablets*, Kingsnorth *et al.* (2016) empregando aplicativos criados pela Universidade Aston na realização da comparação. Um grupo de voluntários selecionados foram submetidos aos exames realizados utilizando dois aplicativos de sensibilidade em um iPad, com duas distâncias diferentes para o teste. Ao final da pesquisa, os estudos evidenciaram resultados com algumas divergências entre os dois métodos, mas afirmando como válidos os resultados obtidos pelas aplicações.

O objetivo deste trabalho assemelha-se ao feito por Kingsnorth *et al.* (2016), quando se trata do exame replicado, da família CSV-1000, entretanto, diferencia-se em sua abordagem, uma vez que facilita a dinamização e a utilização do exame. O aplicativo deste trabalho foi desenvolvido para ser utilizado em *smartphones* com o sistema *android*.

Dentre os projetos relacionados, se encontra o PeekCS (HABTAMU *et al.*, 2019) da Escola de Higiene e Medicina Tropical de Londres (*London School of Hygiene & Tropical Medicine*), que consistiu na produção de um aplicativo para celular para realização do exame de sensibilidade ao contraste em sistemas *android*. Na realização de testes, foram analisados 147 pacientes da região da Etiópia, visto que o país apresenta dificuldades econômicas, impactando nos serviços hospitalares. Durante o processo de testagem, os pacientes realizaram os mesmos protocolos de exames para SC.

Assim, como resultado, os testes utilizando um sistema móvel, em comparação aos resultados com um sistema padrão de SC, apresentaram cerca de 95% de eficácia em diagnósticos de lesões oculares. Neste trabalho relacionado, é verificada a semelhança nos métodos utilizados para testes, com funcionalidades similares. Não obstante, diferenciam-se

ao passo que o teste proposto nesta monografia volta-se para o CSV-1000E, utilizado na máquina Optec® 6500.

Durante o processo de análise do exame de sensibilidade ao contraste, é necessário garantir as variáveis ideais de ambiente para a sua realização. Com base nisso, o trabalho do instituto de pesquisa ocular, situado no departamento de oftalmologia de Harvard (DORR *et al.*, 2013), realizou a criação de um teste de sensibilidade ao contraste utilizando um iPad 2 para averiguação dos tipos diferentes de frequências espaciais, sendo analisado a quantidade de *bits* presentes na tela do aparelho e o quanto isso colabora para realização do exame. Ao fim do projeto, o aplicativo apresentou resultados referentes a configurações ideais sobre as variáveis para a aplicação do teste utilizando sistemas móveis.

### 3 DESENVOLVIMENTO DA PROPOSTA

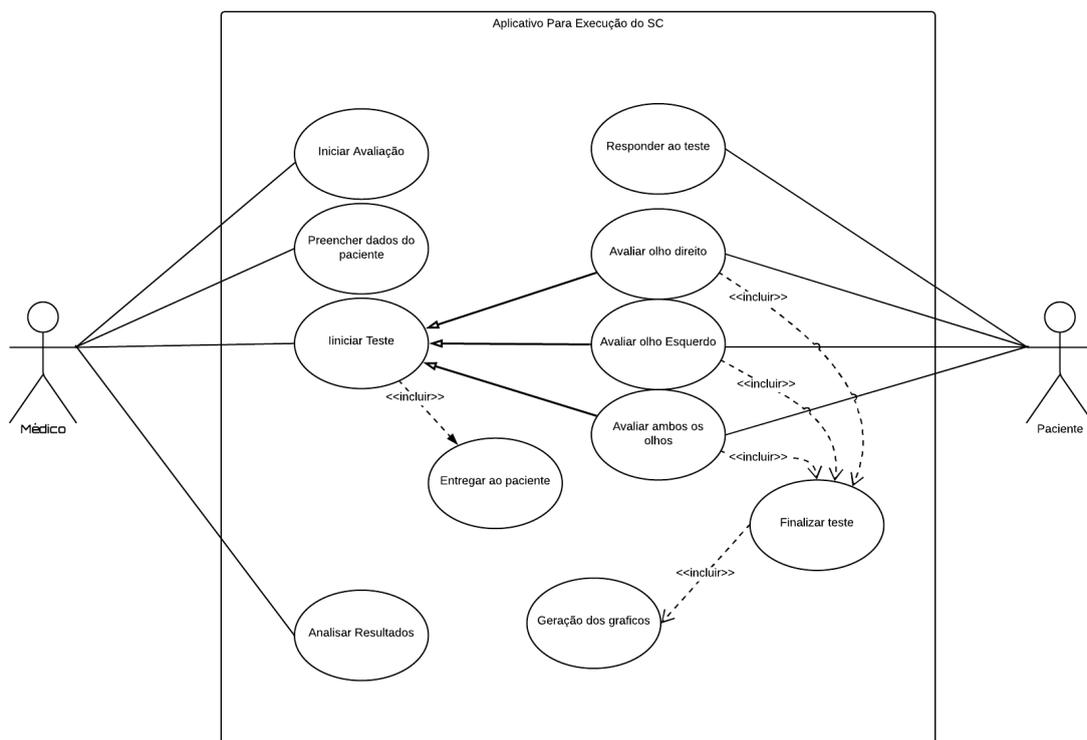
#### 3.1 PLATAFORMA MÓVEL

Este tópico objetiva detalhar as etapas de criação visual do aplicativo, desde seu rascunho inicial, até o modelo final para desenvolvimento. Utilizando a ferramenta Figma como principal mecanismo, seguindo as três etapas de fidelidade descritas anteriormente a fim de facilitar e agilizar o processo criativo, agregando UI (*user interface*), UX (*user experience*) e funcionalidade.

##### 3.1.1 Prototipagem da aplicação

Utilizando conceitos abordados no tópico 2.1.4.6, referentes ao diagrama de caso de uso, conforme a Figura 15, é possível analisar o relacionamento entre os dois atores presentes no uso do sistema proposto.

Figura 15: Diagrama de caso de uso referente à aplicação



Fonte: Autores (2021)

Averiguando aspectos relacionados ao médico, o ator inicial usa de diferentes associações para definições de:

- Inicialização da avaliação;

- Preenchimento de dados do paciente;
- Inicialização do Teste.

Após esses três casos de uso serem realizados, o médico entrega o sistema móvel com a aplicação à espera da interação com o segundo ator, o paciente. Tal ator utiliza o sistema por meio de botões presentes na tela, realizando o teste para as seguintes associações:

- Geração de resultados;
- Geração dos gráficos de sensibilidade ao contraste;
- Detalhamento de resultados.

Ao fim, o paciente entrega o *smartphone* ao médico, para que o mesmo realize o diagnóstico com base no exame concluído.

A idealização do diagrama de uso (Figura 15) referente a prototipagem, é de suma importância para a proposta de construção da aplicação. Por meio da prototipação, é definido as melhores formas de interação do usuário com a interface e com seus componentes funcionais, os quais foram seguidos e adaptados durante as etapas seguintes de desenvolvimento.

A função da prototipação preliminar segue os objetivos citados no tópico 2.1.6, com o propósito de fornecer uma interface limpa, simples e intuitiva para o profissional médico que utilizará a ferramenta e, posteriormente, ao paciente que poderá ser sujeito ao exame.

Deste modo, durante a elaboração da aplicação, até a produção final, os autores deste projeto puderam seguir um fluxo de raciocínio baseando-se em como o aplicativo deveria ser criado, e como as funções deveriam se comportar, de acordo com o planejado inicialmente. Assim conseguiu-se agilizar as demais etapas de criação, seguindo a metodologia de produção em etapas do aplicativo.

#### 3.1.1.1 Prototipagem de baixa fidelidade

Devido ao seu caráter introdutório como desenho de interface, o objetivo inicial do protótipo, visto na Figura 16, consistiu em planejar quais funções a aplicação possuiria.

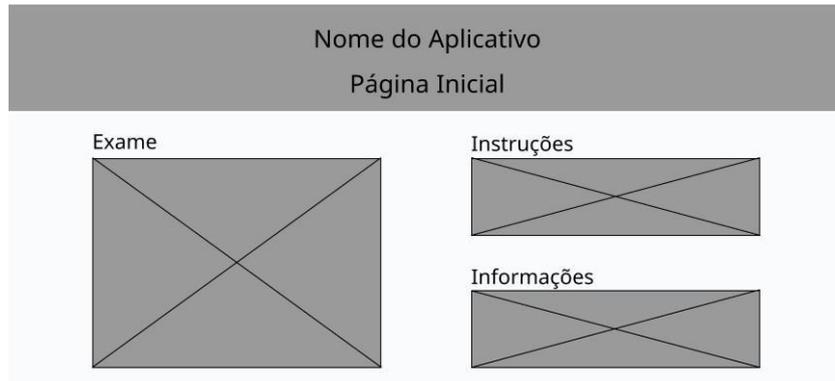
A princípio, as funções desejadas incluíram um botão para iniciar o exame oftalmológico, outro para fornecer ao usuário instruções para manuseio do aplicativo antes e durante o teste, e um terceiro com as informações sobre o aplicativo e seus criadores.

A Figura 16 ilustra a página inicial, ou a *home* do *software*. Uma vez aberto no *smartphone* do médico oftalmologista, é necessário que seja informado de forma clara ao

usuário qual o objetivo da aplicação, e como é possível dar início ao teste, conforme os botões indicativos da tela.

Na mesma tela de *home* foram dispostos os botões “Instruções” e “Informações” contendo as demais informações do sistema e explicações necessárias para que o usuário possa compreender como iniciar e executar o exame com o paciente.

Figura 16: Protótipo de baixa fidelidade

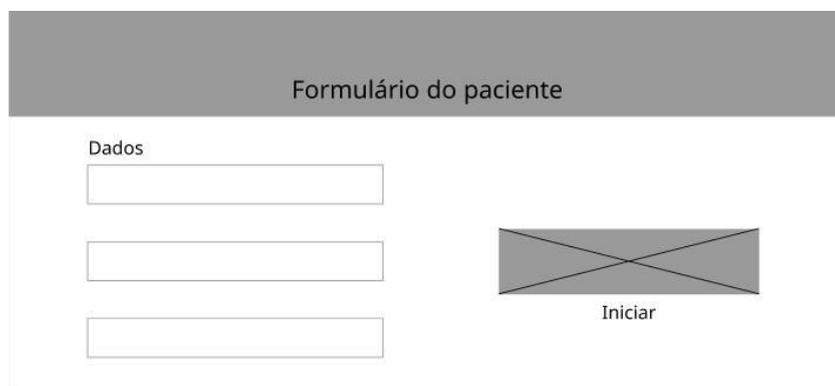


Fonte: Autores (2021)

Avançando pelo modelo, antes de dar início ao teste, viu-se a necessidade de criar o conceito de um formulário dentro do *software* que deve ser preenchido pelo profissional com as informações do paciente, como a Figura 17 representa.

Tais informações tem o intuito de serem exibidas ao final do teste, juntamente com os resultados obtidos do exame, para que possam ser lidos e interpretados pelo médico, auxiliando em possíveis diagnósticos. A funcionalidade dos relatórios é mais aprofundada dentro do tópico 3.3 deste trabalho.

Figura 17: Formulário de baixa fidelidade



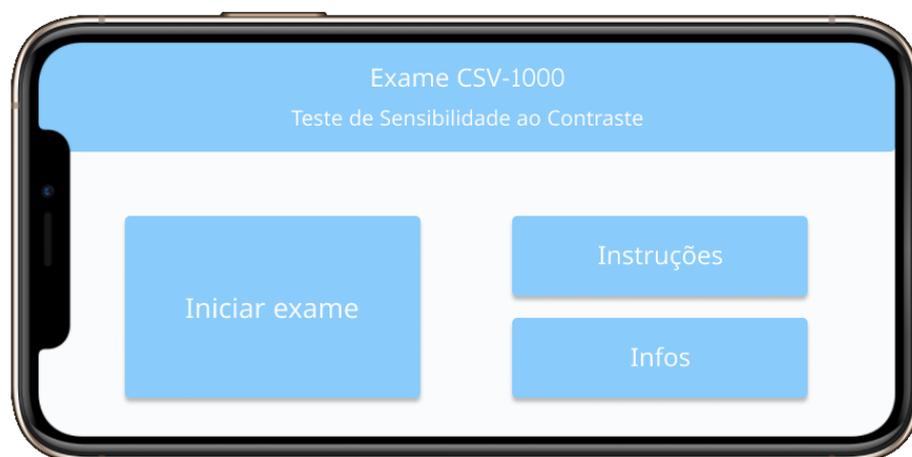
Fonte: Autores (2021)

Esta etapa foi definida a partir de um ponto de vista voltado para a utilidade e menos centrado na usabilidade, por se tratar de um protótipo de baixa fidelidade. Entretanto, uma vez definidas as funcionalidades desejadas, pode-se prosseguir para a segunda etapa de prototipagem.

### 3.1.1.2 Prototipagem de média fidelidade

Utilizando as modelagens iniciais como base, deu-se início à prototipação de média fidelidade. Nesta etapa, foi dado ao modelo cores novas e um design mais amigável e intuitivo, seguindo a ideia inicial de baixa fidelidade.

Figura 18: Página inicial em média fidelidade



Fonte: Autores (2021)

A Figura 18 assemelha-se à Figura 16 de modo geral. Não foram realizadas mudanças quanto à localização de botões ou adição de funções. Entretanto, nesta fase realizou-se a inclusão de aspectos informativos como o título do exame, localizado no cabeçalho da página.

Figura 19: Prototipagem do formulário do paciente

Exame CSV-1000  
Formulário do Paciente

Nome  
Text

Idade  
Text

Gênero  
▼

Observações  
Text

Continuar

Fonte: Autores (2021)

Além disso, a área de preenchimento do formulário do paciente, vista na Figura 19, recebeu um novo campo de “Observações”, para que possam ser acrescentadas informações extras sobre o paciente.

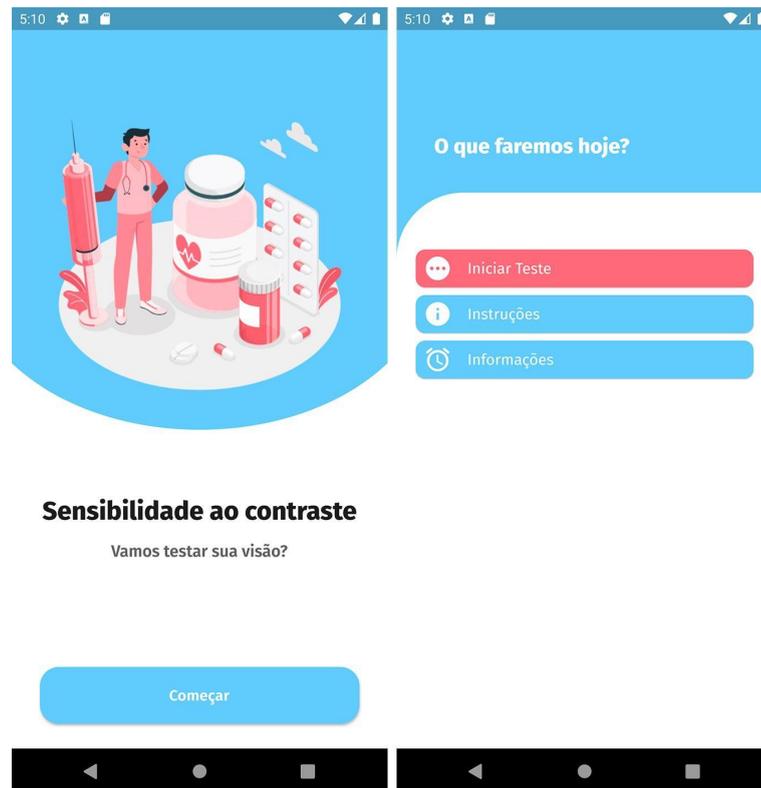
### 3.1.1.3 Prototipagem de alta fidelidade

Na terceira etapa, utilizou-se dos conceitos de UI e UX na elaboração da aplicação, a fim de trabalhar na funcionalidade do sistema e ainda, dispor sua interface de forma intuitiva. Por se tratar do protótipo de alta fidelidade, as figuras presentes neste tópico serviram como a versão final de design para o desenvolvimento do aplicativo, projetado seguindo os padrões estabelecidos.

Após a inicialização do *software*, foi acrescentada uma nova página de introdução antes da página *home*, para situar o usuário dentro do aplicativo, mostrada na Figura 20. E então, ao ser selecionada a opção de continuar, segue-se para a página inicial.

Preocupando-se na forma como o usuário interage com o aplicativo, nas páginas anteriores ao teste, Figura 20, optou-se por alterar a posição das telas do aplicativo, agora iniciando na posição vertical, como os celulares são mais utilizados. A tela *home* segue com botões com as mesmas atribuições do design de média fidelidade, porém com seu visual aprimorado.

Figura 20: Páginas iniciais do aplicativo



Fonte: Autores (2021)

Além disso, criou-se um aviso de conscientização, com cores destacadas, demonstrado na Figura 21, reafirmando o objetivo da aplicação, deixando claro para os que a utilizarem, as diferenças entre o exame tradicional e o aplicativo.

Figura 21: Formulário e aviso de atenção

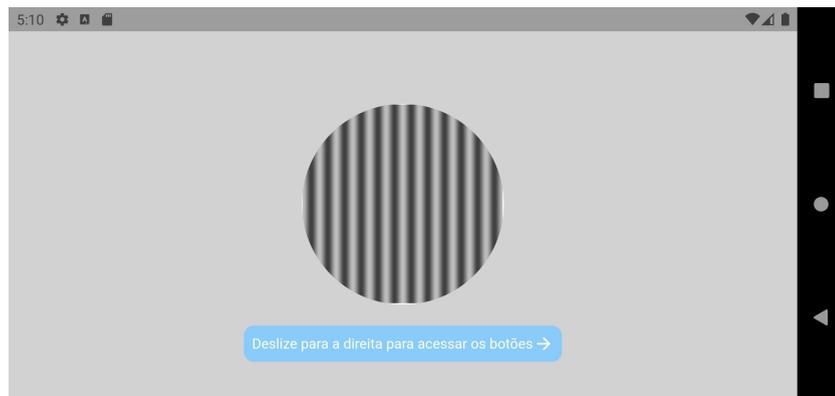
The image shows two side-by-side screenshots of a mobile application. The left screenshot is a yellow warning screen with the text: **Atenção**  
Este aplicativo é focado no uso profissional e não tem como intuito substituir o exame médico CVS-1000. Below the text is a blue button labeled "Continuar". The right screenshot is a white form titled "Formulário do teste". It contains fields for "Nome do paciente" and "Idade", a text area for "Observações", and a dropdown menu for "Gênero:". Below the form is a red button labeled "Continuar". To the right of the form is a cartoon illustration of a person's face with large eyes and a flower on their head. Below the illustration is the text "Selecione um teste para começar" and three buttons: "← Olho esquerdo" (blue), "→ Olho direito" (blue), and "👁️ Ambos" (red).

Fonte: Autores (2021)

Seguindo a linha de construção, voltando para o formulário, a Figura 21 mostra os campos a serem preenchidos pelo médico, agora com alterações também na orientação da tela e posição dos campos de interação, tornando-os mais ergonômicos para digitação. E na mesma figura, como última tela antes do teste, será selecionada a opção que indica com qual olho o paciente iniciará o exame, somando um total de três testes individuais, um com o olho direito, seguido do esquerdo e por fim com os dois olhos.

Por fim, foram desenvolvidas as páginas do exame de sensibilidade, as quais disponibilizam ao usuário a primeira imagem do gradiente gerado. A Figura 22 é um exemplo de figura gerada pelo aplicativo, que, uma vez assimilada pelo paciente, este deverá seguir a instrução vista na imagem para deslizar a tela para a direita.

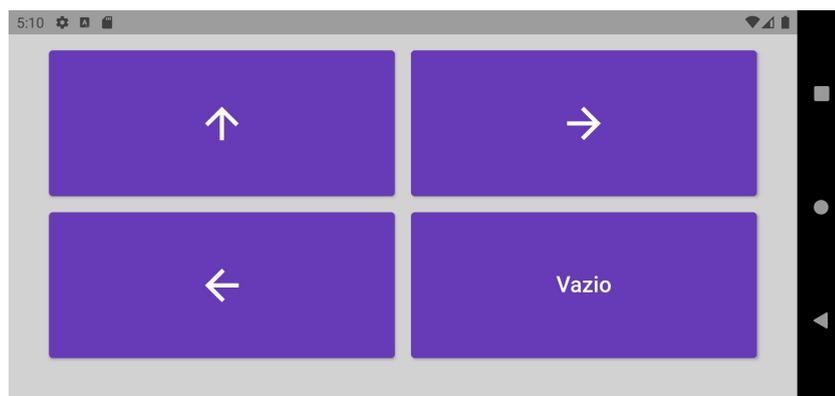
Figura 22: Tela do exame



Fonte: Autores (2021)

A Figura 23 mostra a tela após a instrução da Figura 22. O usuário (paciente) irá selecionar a opção de seta com a direção das linhas retas que conseguiu enxergar da grade anterior, ou caso não tenha conseguido distinguir as linhas, selecionará a opção “Vazio”.

Figura 23: Opções de direção das grades



Fonte: Autores (2021)

Ao fim de cada teste individual, é gerado um gráfico com os resultados daquela sessão, o que se repete com cada olho, como o exemplo localizado na tela ao lado esquerdo na Figura 24.

Figura 24: Gráfico e resultados



Fonte: Autores (2021)

Feitas as três sessões individuais, é gerado o resultado final do exame, disposto conforme mostrado na Figura 24. Este relatório contém as informações preenchidas inicialmente no formulário e o resultado geral do exame, que deverá ser encaminhado ao médico para interpretação.

### 3.1.2 Construção do Front-end

A construção do *front-end* móvel, ou software móvel foi baseada inteiramente na utilização do *framework* Flutter, explicado em 2.1.5 e possui a interface com base nas prototipagens citadas e descritas anteriormente 3.1.1. As primeiras etapas de desenvolvimento se realizaram por meio da preparação do ambiente de desenvolvimento, onde a codificação da aplicação foi feita, instalando o SDK (*Software Development Kit*) do Flutter e o Visual Studio Code (editor de texto).

Logo após a configuração geral do ambiente é então iniciada a produção do código utilizando os *layouts* determinados durante as fases iniciais do projeto. Essas telas são desenvolvidas e testadas de maneira individual, aplicando testes sobre suas funcionalidades esperadas, testes esses simples, entretanto, responsáveis por garantir que suas funcionalidades sejam cumpridas.

Um exemplo para estes testes, são entradas de gestos que o usuário realizaria em botões ou em outras áreas da tela e verificando se elas reagem de acordo com o esperado. Neste caso um botão deve apresentar a animação de que foi tocado e executar a função atribuída a ele. Listas por sua vez devem ser possíveis de receber uma rolagem horizontal ou vertical.

Ao longo da prototipação é possível observar que nem todas as telas desenvolvidas têm funções complexas ou integração com o *back-end*, a grande parte delas busca exibir informações ao usuário ou dar a possibilidade deste poder acionar determinadas funções, como iniciar o teste e especificar com qual dos olhos o teste começará. É possível notar funcionalidades que são somente para guiar os usuários durante a utilização das interfaces para assim chegar no início do teste.

Figura 25: Botão para começar o teste do olho esquerdo

```
CustomButton(
  icon: CupertinoIcons.eye, text: 'Ambos',
  color: AppColors.secondaryColor,
  onTap: () => Navigator.push(context, MaterialPageRoute(
    builder: (context) =>
      testViewModel.testeImages['both']!.length >=
        TestImageInfoSequence.rotation.length
        ? const PDFViewerScreen(type: 'both',)
        : const ImageScreen(selectedTestType: 'both'),
  )))
```

Fonte: Autores (2021)

Devido ao Flutter utilizar Dart como linguagem de programação e também sua arquitetura baseada em uma árvore de *widgets*, que são como ferramentas transformadas em componentes, é possível criar botões e outras formas na interface de forma simples e direta, facilitando e diminuindo a quantidade de programação necessária. Na Figura 25 é possível observar um botão customizado criado para receber alguns parâmetros específicos e desenhá-los na tela de uma forma única, com o objetivo de criar um componente para diminuir a repetição de código.

Com o intuito de escrever a menor quantidade possível de linhas de código ainda mantendo a mesma qualidade e a mesma saída para aquele código, assim otimizando tempo e recursos, a componentização de código é utilizada. Como exemplo, a Figura 26 é a classe onde as propriedades de *CustomButton* são definidas e como elas serão desenhadas na tela, por meio desta maneira as informações passadas durante a execução vista na Figura 25 são desenhadas na tela de maneira apropriada.

Figura 26: Definição do *CustomButton*

```
class CustomButton extends StatelessWidget {
  final IconData icon;
  final String text;
  final Color color;
  final Function()? onTap;

  const CustomButton({
    Key? key,
    required this.icon,
    required this.text,
    this.color = AppColors.primaryColor,
    this.onTap,
  }) : super(key: key);

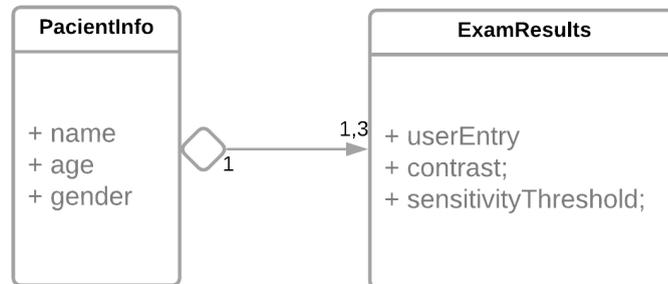
  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: onTap,
      child: Card(
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),
        color: color,
        child: SizedBox(
          height: 40,
          child: Row(
            children: [
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 8.0),
                child: Icon(icon, size: 30, color: Colors.white),
              ),
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 8.0),
                child: Text(text, style: AppTextStyles.regular)
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Fonte: Autores (2021)

Tais integrações são utilizadas em diversos pontos da construção da interface, levando assim ao desenvolvimento geral da interface em um tempo menor e de forma simplificada. Estas interfaces, como citado anteriormente, recebem interações e escolhas do usuário

levando ao acionamento de funções que se comunicam com o *back-end* para a geração dos gráficos.

Figura 27: Diagrama das classes presentes na aplicação



Fonte: Autores (2021)

As integrações levam também a trechos de código responsáveis por preencher os campos das classes que armazenam as informações do paciente, inseridas pelo médico, estas informações estão dispostas conforme a modelagem presente na Figura 27 demonstra.

A função citada na Figura 25 envia o usuário para uma nova tela onde o teste é iniciado, armazenando as informações passadas por meio da seleção do botão. No caso em questão, retém qual foi a opção de teste selecionado. Se a função escolhida já estiver sendo realizada, o que implica em dizer que o teste selecionado já foi realizado, o usuário é redirecionado para a exibição do *PDF* com os resultados deste teste.

### 3.1.3 Integração ao Back-end

A integração com o serviço (*back-end*) acontece depois da realização de cada teste e consiste da troca de informações entre o *front-end* e o *back-end* pela comunicação *HTTP*, mantendo o esquema demonstrado ao longo do modelo na Figura 37 e explicado adiante no tópico 3.2.2.

A comunicação com o serviço é realizada a partir do aplicativo móvel, de forma assíncrona, ou seja, o programa após realizar a solicitação aguarda a finalização desta, sem interferir no andamento geral do programa. Em outras palavras, o software processa a aplicação de maneira paralela ao processamento da interface, desta maneira o usuário ainda consegue interagir com a tela do aparelho celular sem a aplicação aparentar estar travada.

Esta implementação é dada pela utilização dos *Future*, estruturas presentes na linguagem de programação *Dart* que representam o resultado de uma operação assíncrona e possui dois estados, incompleto ou completo (TEAM, 2021a).

Figura 28: Requisição para o *Back-end*

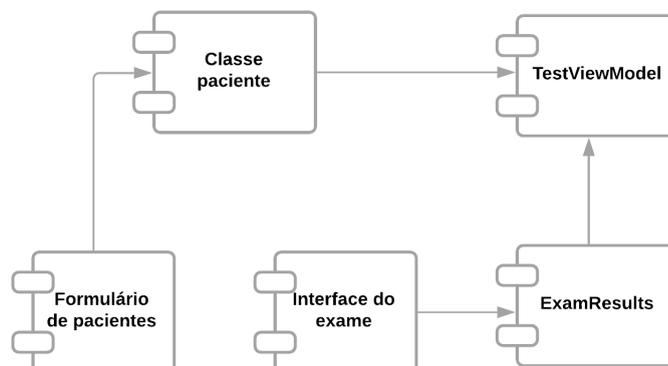
```
Future<dynamic> getTestImageAndInfo(int grating, double contrast, int rotation) async {
  try {
    final response = await client.get(Uri.parse('$BASE_URL/getTest'), headers: {
      'contrast': contrast.toString(),
      'cycle': grating.toString(),
      'rotation': rotation.toString(),
    });
    if (response.statusCode == 200) return ImageInfoModel.fromRawJson(response.body);
  } catch (e) {
    rethrow;
  }
}
```

Fonte: Autores (2021)

A implementação da função apresentada na Figura 28 é a responsável por enviar uma requisição para o *back-end* passando informações a fim de gerar uma nova imagem para ser apresentada ao paciente durante o teste. Também existe implementada na função uma etapa de *error handling* sendo incumbido de tratar erros presentes durante a execução.

Devido a função apresentada na Figura 28 ser do tipo *Future*, é capaz de retornar dois estados ao final de sua execução, incompleto ou completo. O estado completo pode ser observado durante o funcionamento e finalização de forma correta ou incorreta da própria, enquanto o estado incompleto se refere ao intervalo de aguardo da execução.

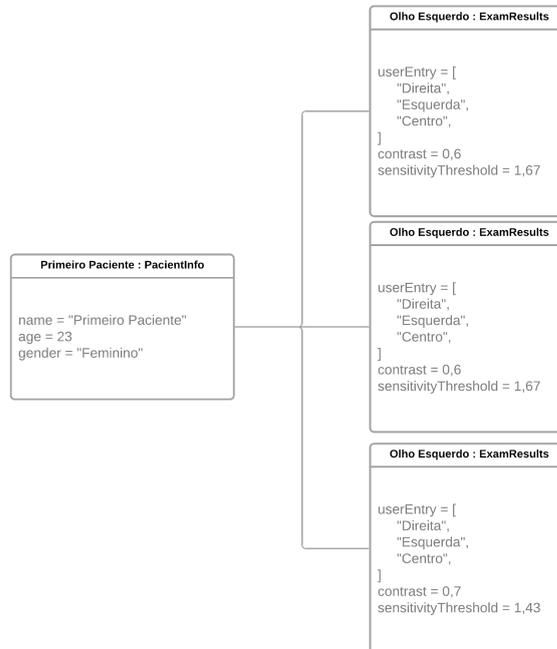
Figura 29: Diagrama de componentes do *front-end*



Fonte: Autores (2021)

Estas funções estão diretamente atreladas ao tratamento dos componentes que fazem parte da estrutura de dados da aplicação, como é possível visualizar por meio da Figura 29 a integração com o *back-end* foi realizada enviando as informações de *ExamResults*.

Figura 30: Diagrama de objetos do *front-end*



Fonte: Autores (2021)

Na Figura 30, é apresentado a relação dos componentes presentes na Figura 29 com seus respectivos objetos e valores inseridos durante a execução da aplicação.

## 3.2 BACK-END

Neste tópico está descrita a construção do *back-end* da aplicação. A partir da utilização de conceitos previamente abordados no tópico 2.1.5, foi empregada a linguagem Python em conjunto com o *framework* Flask responsável por iniciar um *Web Server Gateway Interface* (WSGI) que será o responsável por receber as informações provenientes da aplicação móvel. Bem como a geração das imagens gratinadas que são utilizadas durante a aplicação do teste.

### 3.2.1 Prototipação da biblioteca Psychopy

A prototipação da biblioteca *psychopy* presente na arquitetura *back-end* é composta de funções relacionadas aos cálculos de rotação, angulação e frequência das grades das imagens de sensibilidade ao contraste. Para realização de tais cálculos é utilizada a função

*calculateIfRotated*, podendo ser analisada no bloco de código-fonte abaixo. Durante o processo de geração das grades de contraste, tal função realiza os cálculos de inclinação da imagem para a esquerda ou direita com base em um número randômico, garantindo que o teste não necessite de intervenções.

Figura 31: Função de rotação das grades do exame

```
def calculateIfRotated(self):
    leftRotation = random.randint(
        GratingPhases.allowedRotation_L[1], GratingPhases.allowedRotation_L[0]
    )
    rightRotation = random.randint(
        GratingPhases.allowedRotation_R[0], GratingPhases.allowedRotation_R[1]
    )

    if random.randint(0, 1) == 1:
        return leftRotation if random.randint(0, 1) == 0 else rightRotation
    else:
        return 0
```

Fonte: Autores (2021)

Na Figura 32, referente à função de geração de grades, é possível analisar a geração e armazenamento das mesmas. Tal amostra utiliza as variáveis de tamanho da imagem (*grating\_size*) e frequência de contraste (*contrast*) para a criação de um arquivo ou *frame*. Com base no uso de um laço de repetição que utiliza todas as inclinações, sendo 0 referenciado como imagens gratinadas na vertical, 45 sendo imagens para direita e -45 para imagens com inclinação para esquerda.

Figura 32: Código-fonte de geração das grades

```

sf = 12 / grating_size
rotation = [ 0, 45, -45, 0,
            45, 0, 45, -45, 45]
a = 0.5
contrast = [0.5]
for i in range(9):
    a /= 2
    contrast.append(a)

for r in rotation:
    for c in contrast:
        grating.sf = sf
        grating.contrast = c
        grating.ori = r
        grating.draw()
        win.flip()

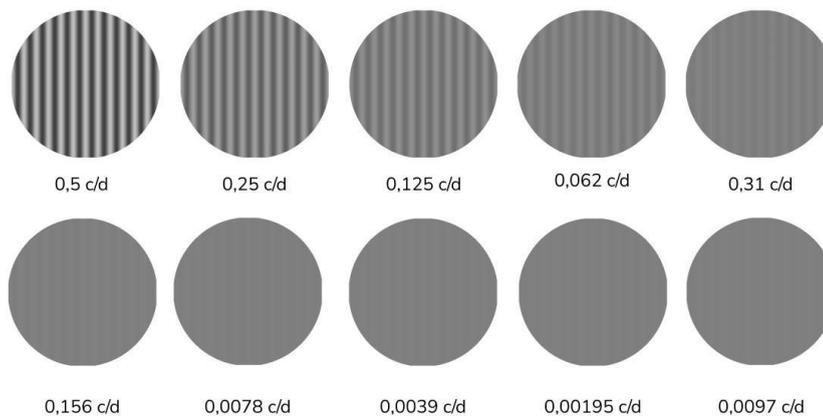
        win.getMovieFrame()
        win.saveMovieFrames(f"T_IMG_{r}_{c}.png")
        sleep(1)
win.close()

```

Fonte: Autores (2021)

Após o processamento do código de geração das grades de sensibilidade ao contraste, são originadas dez imagens com base nas possíveis inclinações: vertical, direita ou esquerda. Sendo utilizado a metragem de ciclos por grau, as dez imagens representam todas as possibilidades de modificações de contraste que o paciente pode causar durante a interação com o aplicativo. Na Figura 33 é possível analisar todas as variações de contraste das grades na vertical dependendo dos valores de ciclos por grau.

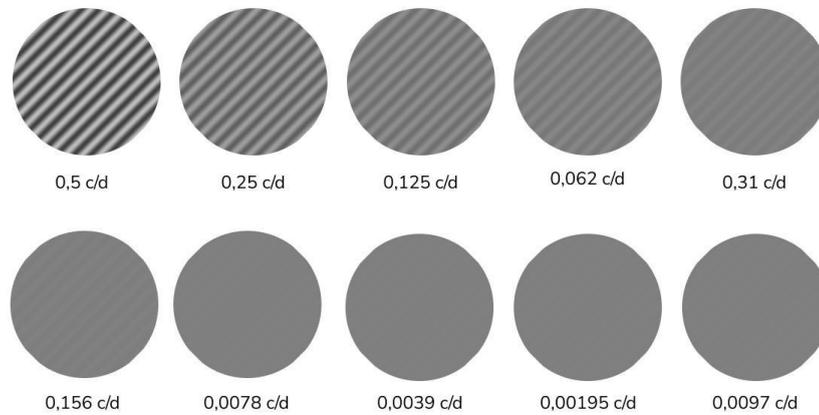
Figura 33: Grades de sensibilidade ao contraste verticais



Fonte: Autores (2021)

Na Figura 34 é apresentado todas as variações de grades com inclinação para a direita.

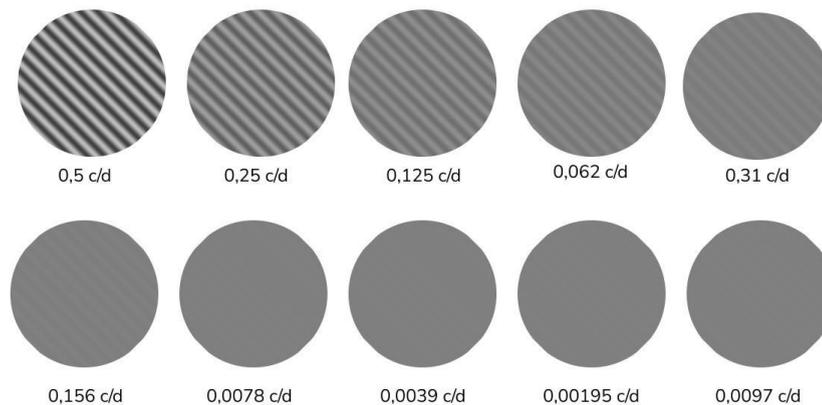
Figura 34: Grades de sensibilidade ao contraste com inclinação para direita



Fonte: Autores (2021)

Na Figura 35 é apresentado todas as variações possíveis de grades com inclinação para a esquerda.

Figura 35: Grades de sensibilidade ao contraste com inclinação para esquerda



Fonte: Autores (2021)

O código-fonte na Figura 36 apresenta a função *CalculateThresholdAndSensitivity*, que contém a herança das funções citadas, reúne aspectos lógicos referentes às possíveis interações e respostas que o usuário pode realizar. Durante sua aplicação, a função utiliza as respostas (*left*, *right* e *up*) em aspectos lógicos (*if*, *elif* e *else*), caso o usuário acerte, o mesmo é contabilizado e acrescentado à variável de resultado (*trialResult*) que será apresentada no relatório do exame. Porém, caso o usuário erre alguma das indicações de inclinação, a função aumenta o nível de contraste da grade para o próximo teste.

Figura 36: Função de cálculo de resultado e interações com o usuário

```

def CalculateThresholdAndSensitivity(self, win, grating,
grating_size, cycle):
    result = []
    hits = 0
    contrast = 0.5
    for i in range(18):
        trialResult = {
            "values": self.generateGrating(
                win,
                grating,
                grating_size,
                contrast,
                cycle / grating_size,
            )
        }
        trialResult["key"] = psychopy.event.waitKeys(
            keyList=["left", "right", "up"]
        )[0]
        if trialResult["key"] == "right" and trialResult["values"][
            "rotation"
        ] in range(
            GratingPhases.allowedRotation_R[0],
            GratingPhases.allowedRotation_R[1]
        ):
            trialResult["correct_result"] = True
            hits += 1
            if hits == 2:
                contrast /= 2
                hits = 0
        elif trialResult["key"] == "left" and trialResult["values"][
            "rotation"
        ] in range(
            GratingPhases.allowedRotation_L[1],
            GratingPhases.allowedRotation_L[0]
        ):
            trialResult["correct_result"] = True
            hits += 1
            if hits == 2:
                contrast /= 2
                hits = 0
        elif trialResult["key"] == "up" and
        trialResult["values"]["rotation"] == 0:
            trialResult["correct_result"] = True
            hits += 1
            if hits == 2:
                contrast /= 2
                hits = 0
        else:
            trialResult["correct_result"] = False
            hits = 0
            contrast = contrast * 2 if contrast * 2 <= 0.5 else contrast
        result.append(trialResult)

```

Fonte: Autores (2021)

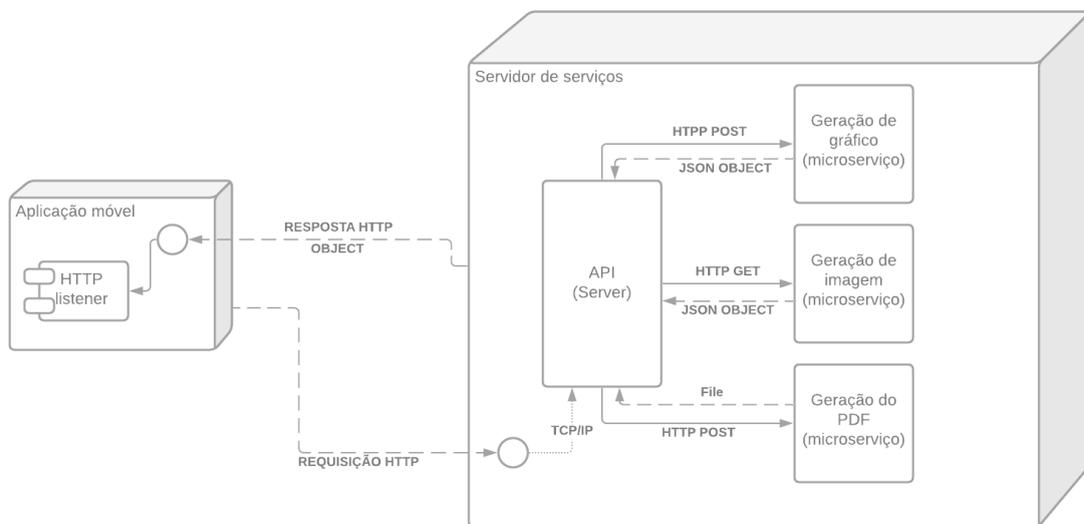
A prototipação desta biblioteca é utilizada como base para definição de integração com a arquitetura *back-end*, de modo que não apresente redundâncias lógicas e funções

desnecessárias que atrapalhem o fluxo de processamento computacional durante o uso do aplicativo móvel.

### 3.2.2 Construção de Arquitetura back-end

O *back-end* foi desenvolvido tendo como base a arquitetura REST 2.1.7, desta forma as informações são repassadas para o software móvel sob demanda, isto é, somente quando a aplicação solicita.

Figura 37: Diagrama de implantação



Fonte: Autores (2021)

Desta maneira, por intermédio do modelo de implantação visível na Figura 37, fica demonstrada como é feita a implementação inicial da comunicação que o *back-end* entrega ao seu *client* (3.1.2). Para além de uma simples comunicação HTTP a API é constituída de micro-serviços, funções que são desenvolvidas para trabalhar sobre dados específicos e retornar o resultado desejado e esperado para que a aplicação que depende desses serviços possa usufruir da informação.

Um destes micro serviços é a função para gerar imagens que serão utilizadas durante os testes visuais seguindo os parâmetros do teste descrito em 2.1.2.

Figura 38: Função para geração de imagens gratinadas

```

@app.route("/getTest", methods=["GET"])
def getTest():
    try:
        grating_size = 300
        padding = grating_size * 0.01
        win = psychopy.visual.Window(
            size=[grating_size + padding, grating_size + padding],
            units="pix",
            fullscr=False,
            color=[2, 2, 1],
        )
        grating = psychopy.visual.GratingStim(
            win=win,
            units="pix",
            size=[grating_size, grating_size],
        )
        contrast = float(request.headers["contrast"])
        cycle = float(request.headers["cycle"])
        results = Vision.generateGrating(
            win,
            grating,
            grating_size,
            contrast,
            cycle / grating_size,
            request.headers["rotation"]
        )
        results["img"] = encodedImage("/tmp/ContrastSensitivity.png")
        results["sf"] = int(cycle)

        return jsonify(results)
    except:
        return Response(status=500)

```

Fonte: Autores (2021)

A função demonstrada na Figura 38 é responsável por tornar alguns argumentos chaves passados para ela no *header* da requisição HTTP, tais parâmetros indicam a para a mesma como a imagem deve ser gerada. Dentro dos tipos de requisição HTTP existentes, esta função possui o método *GET* onde são passadas informações apenas por intermédio do cabeçalho da requisição, uma vez que são poucos valores e somente valores pontuais não havendo necessidade de incluir um corpo de informações maior à requisição. Cada uma destas informações passadas representa um valor específico, suas atribuições são as seguintes:

- *Contrast* é responsável pelo nível de contraste que a imagem deve possuir durante a geração;

- *Cycle* representa a quantidade de ciclos contidos na imagem, em outras palavras, quantas linhas existirão;
- *Rotation* contém a rotação da imagem final.

Ao fim, o retorno em caso de erro segue o padrão de código de erros utilizado durante transações HTTP, neste em questão retornará 500 por intermédio do trecho: ‘return Response(status=500)’. Caso contrário é então retornado um objeto em notação *JSON* contendo as informações necessárias para a aplicação móvel exibir a imagem e manter informações sobre como proceder na validação da resposta do paciente.

Figura 39: Retorno em JSON do back-end para a geração de imagem

```
{  
  "contrast": 0.12,  
  "img": "base64",  
  "rotation": 23,  
  "sf": 12  
}
```

Fonte: Autores (2021)

Demais funções existem dentro do servidor como micro-serviços adjacentes e acionados sob demanda a partir das aplicações clientes do servidor. Este é o caso para a função de geração de gráfico, a qual é responsável por receber informações de erros e acertos do paciente provenientes do *front-móvel* 3.1.2 e plotar em um gráfico estes valores.

Figura 40: Função geradora de gráficos

```

@app.route("/generateGraph", methods=["POST"])
def generateChart():
    fig, ax = plt.subplots()

    ax.plot(range(1, 18), request.json["values"])
    ax.set(
        xlabel="Trial number",
        ylabel="Contrast",
        title=f"{request.json['cycle']} Spatial Frequencies
(cycles/degree)",
    )
    ax.set_xlim([0, 18])
    ax.grid()
    plt.draw()
    plt.savefig("/tmp/results.png")
    return jsonify({"img": encodedImage("/tmp/results.png")})

```

Fonte: Autores (2021)

A função descrita na Figura 40 é uma função que pode ser considerada auxiliar, devido a seu papel ser apenas receber informações que serão provenientes de uma função mais completa e complexa. Valores recebidos nesta função são convertidos para plotagem de um gráfico de contraste por imagem respondida.

### 3.3 GERAÇÃO DE RELATÓRIOS

A geração do relatório é baseada nas respostas do paciente ao término do exame. Sendo realizado a construção de um gráfico utilizando a biblioteca *matplotlib*, tal gráfico apresenta a relação entre o nível de contraste (*contrast*) e o número de tentativas do paciente (*Trial number*). Sendo feito um laço de repetição (*for i in range*) baseado no tamanho do vetor de resultados (*resultContrastValues*), com base nos seguintes pontos:

- Caso o resultado atual seja menor que o resultado anterior do teste, é indexado um resultado de descida da reta presente no gráfico.
- Caso o resultado atual seja maior que o resultado anterior, é indexado um resultado de subida da reta presente no gráfico.

Na Figura 41, referente ao bloco lógico de indexação de resultados, é possível averiguar as condições para realização do armazenamento e comparação de respostas.

Figura 41: Bloco lógico de indexação de resultados

```

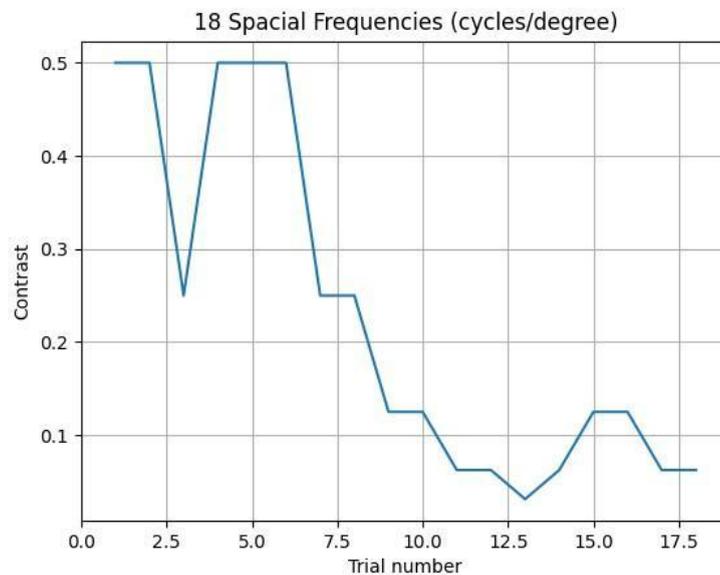
upDown = []
for i in range(len(resultContrastValues)):
    if i - 1 >= 0:
        if resultContrastValues[i] < resultContrastValues[i - 1]:
            upDown.append({"action": "desceu", "index": i})
        elif resultContrastValues[i] == resultContrastValues[i - 1]:
            pass
        else:
            upDown.append({"action": "subiu", "index": i})
    else:
        pass

```

Fonte: Autores (2021)

Com base nos valores provenientes do bloco lógico de indexação presentes na Figura 41, é possível analisar na Figura 42 os aspectos de relação entre as duas variáveis, nível de contraste e tentativas do paciente utilizando o vetor *resultContrastValues*.

Figura 42: Gráfico de respostas do exame



Fonte: Autores (2021)

Após o exame com o paciente, são gerados três gráficos (Figura 42) referentes a cada um dos testes (olho esquerdo, olho direito e olhos simultaneamente), sendo observado seu nível de sensibilidade ao contraste.

Em relação à produção do arquivo em PDF (*Portable Document Format*) do relatório do exame, são usadas as variáveis presentes no *front-end*, armazenadas durante a utilização do aplicativo previstas na Figura 27, referente ao diagrama de classe. Para a conversão dos

resultados armazenados em linhas de caracteres legíveis, é necessário utilizar uma cadeia lógica de modificação de valores, conforme pode ser visto na Figura 43, referente à função *convertResponse*.

Figura 43: Código-fonte referente a conversão de valores

```
String convertResponse(int val) {  
    if (val = 1) {  
        return 'Direita';  
    } else if (val = -1) {  
        return 'Esquerda';  
    } else {  
        return 'Center';  
    }  
}
```

Fonte: Autores (2021)

Para definição de aspectos estéticos e apresentação de variáveis no relatório, é necessário construir uma função com as três possibilidades de resultados, sendo o exame aplicado em cada um dos olhos e depois de forma binocular.

Figura 44: Código-fonte relacionado a criação do arquivo em PDF

```

pdf.addPage(
  pdfw.Page(
    pageFormat: PdfPageFormat.a4,
    build: (context) => pdfw.ListView(
      children: [
        pdfw.Text(
          'Teste de Sensibilidade ao Contraste',
          style: const pdfw.TextStyle(fontSize: 40),
          textAlign: pdfw.TextAlign.center,
        ),
        pdfw.Padding(
          padding: const pdfw.EdgeInsets.all(DEFAULT_PADDING),
          child: pdfw.Text(
            'Data ${testDate.day}/${testDate.month}/${testDate.year} às
            ${testDate.hour}:${testDate.minute}',
            style: const pdfw.TextStyle(fontSize: 20),
          ),
        ),
        pdfw.SizedBox(height: 160),
        pdfw.Row(
          mainAxisAlignment: pdfw.MainAxisAlignment.center,
          children: [
            pdfw.Text(
              'Paciente: $patientName, Idade: $age, Gênero: ${genderName()}',
              style: const pdfw.TextStyle(fontSize: 20),
            ),
          ],
        ),
        pdfw.SizedBox(height: DEFAULT_PADDING),
        if (observations.isNotEmpty)
          pdfw.Text(
            'Observações: $observations',
            style: const pdfw.TextStyle(fontSize: 20),
          ),
      ],
    ),
  ),
);

```

Fonte: Autores (2021)

Na Figura 44, é apresentada parte da função *generatePDF*, que possui a definição do tamanho de página A4 definido por *PdfPageFormat.a4*. O documento apresenta variáveis definidas no código como: o título do exame, nome (*patientName*), idade (*age*), gênero do paciente (*genderName*), observações (*observations*), a data e hora de realização do exame, sendo definidos conforme uma matriz, por linhas (*pdfw.Row*) e colunas (*pdfw.Column*) da página.



Figura 46: Bloco lógico referente à geração da tabela de respostas

```
pdfw.TableRow(
  children: [
    pdfw.Text(
      'Rotação (graus)',
      style: const pdfw.TextStyle(fontSize: 20),
      textAlign: pdfw.TextAlign.center,
    ),
    pdfw.Text(
      'Contraste (%)',
      style: const pdfw.TextStyle(fontSize: 20),
      textAlign: pdfw.TextAlign.center,
    ),
    pdfw.Text(
      'Resposta esperada',
      style: const pdfw.TextStyle(fontSize: 20),
      textAlign: pdfw.TextAlign.center,
    ),
    pdfw.Text(
      'Resposta obtida',
      style: const pdfw.TextStyle(fontSize: 20),
      textAlign: pdfw.TextAlign.center,
    ),
  ],),)
```

Fonte: Autores (2021)

O código-fonte exibido na Figura 46, relacionado ao processo de construção da tabela (*TableRow*) com as variáveis indexadas para visualização no *front-end*, junto aos aspectos estéticos já citados, apresenta os valores referentes a rotação, contraste em valores de porcentagem, a resposta correta é a obtida pelo paciente durante o teste.

Figura 47: Código referente ao armazenamento do relatório

```
void sharePDF(Uint8List pdfBytes) async {
  final directory = await getApplicationDocumentsDirectory()
    ..path;
  final File pdfFile = File('$directory/resultadoTeste-$patientName.pdf');
  pdfFile.writeAsBytes(pdfBytes);
  Share.shareFiles([
    '$directory/resultadoTeste-$patientName.pdf',
  ]);
}
```

Fonte: Autores (2021)

A última etapa para a geração do relatório do exame está relacionada ao armazenamento do arquivo em um diretório, conforme visto na Figura 47, o arquivo é salvo na pasta raiz da aplicação, sendo intitulado como “*resultadoTeste*” junto ao nome do paciente definido por “*patientName*”.

## 4 VALIDAÇÃO DO SISTEMA

Neste capítulo serão analisados os resultados obtidos durante o desenvolvimento e conclusão desta monografia, que envolvem a validação médica proveniente de testes no aplicativo após sua produção, de acordo com o planejado nos tópicos 1.2.1 e 1.2.2.

Dentro do sistema proposto, fez-se necessário a identificação dos padrões de imagens utilizadas durante o exame de avaliação à sensibilidade ao contraste, realizados em equipamentos médicos convencionais.

O aplicativo então, utiliza um método para calcular a frequência, angulação e rotação da grade das imagens, por intermédio de parâmetros passados para bibliotecas utilizadas dentro das instâncias da aplicação, com o intuito de gerar o tipo de imagem requerida pelo teste, retornando assim, as figuras desejadas. Deste modo, pode-se definir que o objetivo de integração das imagens ao sistema foi alcançado.

Com o intuito de agregar funcionalidades, ao decorrer do exame de sensibilidade dentro da aplicação, os dados são gerados em forma de gráficos que são armazenados até o fim da sessão, sendo posteriormente sintetizados em forma de tabela ao fim de cada etapa do teste, fornecendo desta maneira os dados desejados nesse exame.

A partir da finalização do desenvolvimento, encaminhou-se a aplicação completa para a validação médica, na qual o aplicativo foi utilizado e analisado por uma médica oftalmologista, a fim de averiguar o nível de adequação ao exame e a usabilidade do aplicativo. A profissional foi a usuária responsável por efetuar as validações, seguindo uma metodologia predeterminada para homogeneizar a análise e execução do exame, conforme detalhado no Anexo - A.

Deste modo, realizou-se uma série de dez testes com o aplicativo instalado em seu celular pessoal, com sistema operacional Android, sistema para qual o aplicativo foi desenvolvido, em um ambiente com luzes internas apagadas e vedado contra a entrada de luz natural. Durante o processo, o celular permaneceu com o brilho configurado no máximo e sem influência de filtros azuis.

A posição padrão para o exame seguiu com os braços rentes ao corpo, e os antebraços estendidos à frente, com o *smartphone* posicionado em mãos. No momento das aferições, seguiu-se a ordem de visão do olho direito, seguido do olho esquerdo, e por fim com os dois olhos.

Conforme apontado pela médica Júlia Ponte Souza, visto no Anexo - A, aferiu-se uma análise positiva proveniente da profissional, com destaque para a facilidade de manuseio da

aplicação, contendo direções bem definidas. Correspondente ao objetivado no início desta monografia, a aplicação foi apontada com sucesso como uma proposta com potencial para ser encaminhada para o Comitê de Ética e Pesquisa (CEP) para futuras análises. Propondo também um papel que pode ser definido como triagem médica, sendo um exame inicial eventualmente realizado pelo próprio paciente.

## 5 CONCLUSÃO

Com resultados muito satisfatórios para a proposta inicial, pode-se notar como é possível tornar realidade soluções viáveis e alternativas com o intuito de auxiliar médicos em suas rotinas diárias de exames específicos ou não. Desta maneira, ao decorrer deste trabalho pode-se observar as constantes evoluções da tecnologia em relação à medicina, em especial a área oftalmológica. Tais avanços culminaram na aplicação projetada e descrita nesta monografia.

Este projeto além de se provar uma alternativa para a aplicação do exame, é também o produto do conhecimento agregado durante os anos, contendo diversas áreas aplicáveis na engenharia de software, também visíveis na engenharia de computação. No entanto, somente possível por meio da multidisciplinaridade, com a junção entre a área oftalmológica e a engenharia.

O desenvolvimento do aplicativo móvel, além de entregar o conceito de uma nova forma de se contornar problemas como o difícil acesso a equipamentos por vezes caros, possibilitou também um aprofundamento nas tecnologias presentes no mercado atual de desenvolvimento, solidificando e aprimorando conceitos contidos ou não no escopo das etapas da graduação.

Por meio do uso de ferramentas de prototipação, o Figma, com o estudo de diagramas relacionados à usabilidade do sistema, o uso de ferramentas de desenvolvimento e *frameworks front-end*, Flutter e Dart, para a construção da interface e aspectos lógicos da aplicação, bibliotecas em Python para a construção do sistema *back-end* com Flask e geração de imagens para o teste de sensibilidade ao contraste, somado ao referencial teórico relacionado aos meios de exames oftalmológicos foi possível criar um exame que apresente maior portabilidade, praticidade e acessibilidade ao médico e paciente.

Sendo assim, com base na validação do sistema, o aplicativo apresenta total competência para com os objetivos iniciais, apresentando potencial para ser enviado ao CEP como proposta na categoria de auxílio em exames médicos ou residenciais, como forma de triagem, proporcionando um teste capaz de auxiliar no diagnóstico preventivo de doenças oculares.

## 5.1 DIFICULDADES ENCONTRADAS

Durante a produção do sistema, o projeto apresentou dificuldades relacionadas aos aspectos teóricos da área de oftalmologia, devido ao fato de ser uma área de diferente atuação relacionada aos autores desta monografia.

Durante o desenvolvimento da aplicação, o sistema apresentou problemas para a transformação das imagens geradas pela biblioteca Psychopy que apresentavam formato quadrangular, e o exame é feito utilizando imagens circulares.

Outra dificuldade durante o desenvolvimento, ocorreu na confecção estética das imagens de grade de sensibilidade ao contraste, devido às cores necessitarem de tons preto e branco. Apesar de parte do problema ser solucionado, em certas frequências a imagem apresenta tons cinzas.

Sobre aspectos do *back-end*, como a geração de gráficos e relatório são providenciadas em conjunto ao *back-end*, o projeto apresentou dificuldades relacionadas ao envio dos dados em uma mesma requisição, sendo necessário converter a imagem para base64 e enviar os *bytes* referentes ao gráfico junto às informações do relatório.

## 5.2 TRABALHOS FUTUROS

Tendo em vista o potencial da aplicação do teste em sistemas móveis e com base na validação médica do sistema, a aplicação pode ser encaminhada para um estudo clínico para confecção de testes e análise perante ao CEP.

Relacionado à possibilidade de utilização residencial do sistema, caso validado após a realização de testes, seria implementado na aplicação aspectos de automação, para realização do envio automático do relatório para o médico oftalmologista, facilitando a interação entre paciente e médico a respeito do diagnóstico.

Analisando a empregabilidade do sistema em projetos relacionados à obtenção e análise de dados. A proposta apresenta potencial no direcionamento de políticas públicas, utilizando a aplicação como um possível meio de obtenção de dados de diagnósticos de exames referente a uma região para a análise dos mesmos.

## REFERÊNCIAS

- ARIETA, C. E. L.; DELGADO, A. M. N. JOSÉ, N. K.; TEMPORINI, E. R.; ALVES, M. R.; FILHO, D. de C. M. Refractive errors and cataract as causes of visual impairment in brazil. **Ophthalmic Epidemiology**, Taylor Francis, v. 10, n. 1, p. 15–22, 2003. Disponível em: <<https://doi.org/10.1076/oep.10.1.15.13774>>.
- BELTRAN, M. **Flutter architectural overview**. 2021. Disponível em: <<https://github.com/flutter/website/blob/master/src/docs/resources/architectural-overview.md>>. Acesso em: 13 abr.2021.
- BICAS, H. **Acuidade visual. Medidas e notações**. 2002. Disponível em: <[https://www.researchgate.net/publication/26340522\\_Acuidade\\_visual\\_Medidas\\_e\\_notacoes](https://www.researchgate.net/publication/26340522_Acuidade_visual_Medidas_e_notacoes)>.
- BURKE, W. Genetic tests: Clinical validity and clinical utility. **Current Protocols in Human Genetics**, v. 81, n. 1, p. 9.15.1–9.15.8, 2014. Disponível em: <<https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/0471142905.hg0915s81>>.
- CARVALHO, J. A. A. Oftalmologia e realidade virtual. **Revista Brasileira de Oftalmologia**, scielo, v. 71, p. 40 – 47, 02 2012. ISSN 0034-7280. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0034-72802012000100008&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-72802012000100008&nrm=iso)>.
- COELHO, L. M. Avaliação da função visual de sensibilidade ao contraste: estudo comparativo entre um novo teste e um teste padrão. p. 1–78, 2019. Disponível em: <[https://repositorio.ufmg.br/bitstream/1843/32484/1/TESE\\_Leticia\\_Maria\\_Coelho.pdf](https://repositorio.ufmg.br/bitstream/1843/32484/1/TESE_Leticia_Maria_Coelho.pdf)>.
- CORPORATION, V. S. R. **EyeView: Function Vision Analysis Software**. 2002. 1-21 p. Disponível em: <<https://www.stereooptical.com/wp-content/uploads/2017/12/EyeView-Optec-manual.pdf>>.
- DEV MEDIA. Arquitetura de software como descrever com a norma iso/iec/ieee 42010. **Engenharia de Software Magazine**, n. 89, 2017.
- DIAS, R. P. **Prototipagem de Software**. 2019. Disponível em: <<https://medium.com/contexto-delimitado/prototipagem-de-software-7ac07027e6d8>>.
- DORR, M.; LESMES, L. A.; LU, Z.-L.; BEX, P. J. Rapid and reliable assessment of the contrast sensitivity function on an ipad. **Visual Psychophysics and Physiological Optics**, p. 1–8, Jan 2013. Disponível em: <<https://doi.org/10.1167/iovs.13-11743>>.
- DUDLEY, D. **The Evolution Of Mobile Phones: 1973 To 2019**. 2018. Disponível em: <[https://flauntdigital.com/blog/evolution-mobile-phones/#content\\_5](https://flauntdigital.com/blog/evolution-mobile-phones/#content_5)>. Acesso em: 08 abr. 2021.
- FAKHROUTDINOV., K. **UML Profile Diagrams**. 2020. Disponível em: <<https://www.uml-diagrams.org/profile-diagrams.html>>. Acesso em: 07 abr. 2021.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Dissertação de Doutorado. Disponível em: <[https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)>.

FIGUEIREDO, E. **Relacionamentos do Diagrama de Classes**. 2016. Disponível em: <[https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-diagrama-classes-relacionamentos\\_v01.pdf](https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-diagrama-classes-relacionamentos_v01.pdf)>. Acesso em: 07 abr. 2021.

FILHO, A. **Arquitetura de Software: Desenvolvimento orientado para arquitetura**. 2008. Disponível em: <<https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033>>. Acesso em: 07 abr. 2021.

FOUNDATION, P. S. **Python about**. 2020. Disponível em: <<https://www.python.org/about/>>. Acesso em: 06 abr.2021.

GOOGLE. **Dart Overview**. 2021. Disponível em: <<https://dart.dev/overview>>. Acesso em: 12 abr. 2021.

GRAY, J. R.; SIMPSON, S.; MACASKILL, M.; HÖCHENBERGER, R.; SOGO, H.; KASTMAN, E.; LINDELØV, J. K. *Psychopy2: Experiments in behavior made easy*.

**SpringerLink**, p. 195–203, 2019. Disponível em: <<https://link.springer.com/article/10.3758/s13428-018-01193-y#citeas>>.

GREEN, S.; VOGT, H. **Personalizing Medicine: Disease Prevention in silico and in socio**. 2016. Disponível em: <<http://www.humanamente.eu/index.php/HM/article/view/62>>.

GROUP, O. M. **OMG Unified Modeling Language™ (OMG UML), Infrastructure**. 2011. Disponível em: <<https://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>>. Acesso em: 07 abr. 2021.

HABTAMU, E.; BASTARWROUS, A.; BOLSTER, N. M.; TADESSE, Z.; CALLAHAN, E. K.; GASHAW, B.; MACLEOD, D.; J.BURTON, M. Development and validation of smartphone-based contrast sensitivity test. **Translational vision science Technology**, v. 10, n. 1, p. 1–10, Jun 2019. Disponível em: <<https://doi.org/10.1167/tvst.8.5.13>>.

HASSANI, T. J.; SANHARAWI, M. E.; DUPONT-MONOD, S.; BAUDOUIN, C. Les smartphones en ophtalmologie. **Journal Français d’Ophtalmologie**, v. 36, n. 6, p. 499–525, 2013. ISSN 0181-5512. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0181551213001460>>.

HOHBERGER, B.; LAEMMER, R.; ADLER, W.; JUENEMANN, A.; HORN, F. **Measuring contrast sensitivity in normal subjects with OPTEC® 6500: influence of age and glare**. 2007. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/17694315/>>.

HUANG, W.; ZHANG, X. 3D Printing: Print the Future of Ophthalmology. **Investigative Ophthalmology Visual Science**, v. 55, n. 8, p. 5380–5381, 08 2014. ISSN 1552-5783. Disponível em: <<https://doi.org/10.1167/iovs.14-15231>>.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

IBGE. **USO DE INTERNET, TELEVISÃO E CELULAR NO BRASIL**. 2018. Disponível em:

<<https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html>>. Acesso em: 01 abr. 2021.

KARA-JUNIOR, N. Technology, teaching, and the future of ophthalmology and the ophthalmologist. **Arquivos Brasileiros de Oftalmologia**, scielo, v. 81, p. V – VI, 06 2018. ISSN 0004-2749. Disponível em:

<[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0004-27492018000300001&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492018000300001&nrm=iso)>.

KARA, S.; GENCER, B.; ERSAN, I.; ARIKAN, S.; KOCABIYIK, O.; TUFAN, H. A.; COMEZ, A. **Reprodutibilidade dos escores de sensibilidade ao contraste em pacientes com degeneração macular relacionadas à idade, glaucoma e catarata**. 2016. Disponível em: <<https://www.scielo.br/pdf/abo/v79n5/0004-2749-abo-79-05-0323.pdf>>.

KEANE, P.; TOPOL, E. **Reinventing the eye exam**. 2019. Disponível em: <[https://doi.org/10.1016/S0140-6736\(19\)33051-X](https://doi.org/10.1016/S0140-6736(19)33051-X)>.

KINGSNORTH, A.; DREW, T.; GREWAL, B.; WOLFFSOHN, J. S. Mobile app aston contrast sensitivity test. **Clinical and Experimental Optometry**, v. 99, n. 4, p. 350–355, jul 2016. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1111/cxo.12362/abstract>>.

KJAER, P.; SALOMÃO, S.; JR., R.; COLELLA, A. **Validação clínica de teste psicofísico computadorizado para avaliação de visão de cores e sensibilidade ao contraste**. 2000. Disponível em:

<[https://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0004-27492000000300004](https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492000000300004)>.

KULENDRAN, M.; LIM, M.; LAWS, G.; CHOW, A.; NEHME, J.; DARZI, A.; PURKAYASTHA, S. **Surgical smartphone applications across different platforms: their evolution, uses, and users**. 2014. Disponível em:

<<https://pubmed.ncbi.nlm.nih.gov/24711263/>>.

LANGUAGES, O.; GOOGLE. **Oxford Languages and Google**. 2021. Disponível em: <<https://languages.oup.com/google-dictionary-pt/>>.

LUCIDCHART. **Diagrama de caso de uso UML: O que é, como fazer e exemplos**. c2021. Disponível em: <<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>>. Acesso em: 09 abr. 2021.

LUCIDCHART. **O que é um diagrama de classe UML?** c2021. Disponível em: <[https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml/#section\\_0](https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml/#section_0)>. Acesso em: 07 abr. 2021.

LUCIDCHART. **Tudo sobre diagramas de estrutura composta**. c2021. Disponível em: <<https://www.lucidchart.com/pages/pt/diagrama-de-estrutura-composta-uml>>. Acesso em: 08 abr. 2021.

MARTINS, C. **Stack Overflow divulga linguagens de programação mais amadas, temidas e procuradas em 2020**. 2020. Disponível em: <<https://sempreupdate.com.br/stack-overflow-divulga-linguagens-de-programacao-mais-amadas-temidas-e-procuradas-em-2020/>>. Acesso em: 06 abr. 2021.

- MEIRELLES, F. de S. **Uso da TI - Tecnologia da Informação nas Empresas**. 2020. Disponível em: <<https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti>>. Acesso em: 03 abr. 2021.
- MORALES, J. C.; AREZ, N. A. E. R.; VARGAS, S. H.; A, A. J. P. Uso de aplicativos móveis en el aula y sus factores determinantes. **Formação universitária**, scielocl, v. 13, p. 13 – 22, 12 2020. ISSN 0718-5006. Disponível em: <[https://scielo.conicyt.cl/scielo.php?script=sci\\_arttext&pid=S0718-50062020000600013&nr m=iso](https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-50062020000600013&nr m=iso)>. Acesso em: 08 abr. 2021.
- NEVES, R. de O. **Fidelidade de protótipos: Baixa, Média ou Alta, conheça os tipos**. 2017. Disponível em: <<https://www.vitaminaweb.com.br/fidelidade-de-prototipos-baixa-media-ou-alta/>>.
- NIGRI, D. **O Figma e a próxima geração de ferramentas de Design**. 2020. Disponível em: <<https://brasil.uxdesign.cc/o-figma-e-a-próxima-geração-de-ferramentas-de-design-ab1682d2a2dd>>. Acesso em: 18 maio. 2021.
- OFTALMOLOGIA, C. B. de. **As condições de saúde ocular no Brasil**. 2015. Disponível em: <[http://www.cbo.net.br/novo/publicacoes/Condicoes\\_saude\\_ocular\\_IV.pdf](http://www.cbo.net.br/novo/publicacoes/Condicoes_saude_ocular_IV.pdf)>.
- OFTALMOLOGIA, C. B. de. **Mais acesso à saúde ocular**. 2015. Disponível em: <[https://www.cbo.net.br/novo/publicacoes/Acesso\\_saude\\_ocular.pdf](https://www.cbo.net.br/novo/publicacoes/Acesso_saude_ocular.pdf)>.
- OLIVEIRA, N. de. **Nova proposta de classificação territorial do IBGE vê o Brasil menos urbano**. 2017. Disponível em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2017-07/nova-proposta-de-classificacao-territorial-do-ibge-ve-o-brasil-menos-urbano>>.
- OPTICAL, S. **Functional Vision Analyzer - Optec 6500**. 2021. Disponível em: <<https://www.stereooptical.com/products/vision-screeners-2/functional-vision-analyzer/>>.
- ORACLE. **What is IoT**. 2021. Disponível em: <[https://www.oracle.com/br/internet-of-things/what-is-iot/#:~:text=O%20Internet%20of%20Things%20\(IoT,trocar%20dados%20com%20outros%20dispositivos.&text=Esses%20dispositivos%20variam%20de%20objetos%20domésticos%20comuns%20a%20ferramentas%20industriais%20sofisticadas.,-Com%20mais%20de](https://www.oracle.com/br/internet-of-things/what-is-iot/#:~:text=O%20Internet%20of%20Things%20(IoT,trocar%20dados%20com%20outros%20dispositivos.&text=Esses%20dispositivos%20variam%20de%20objetos%20domésticos%20comuns%20a%20ferramentas%20industriais%20sofisticadas.,-Com%20mais%20de)>. Acesso em: 07 abr. 2021.
- ORGANIZATION, W. H. **Global initiative for the elimination of avoidable blindness**. 1997. Disponível em: <[https://www.paho.org/hq/dmdocuments/2008/Global\\_Initiative\\_for\\_the\\_Elimination\\_of\\_Blindness.pdf](https://www.paho.org/hq/dmdocuments/2008/Global_Initiative_for_the_Elimination_of_Blindness.pdf)>.
- OVERFLOW, S. **Most Popular Technologies**. 2020. Disponível em: <<https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-all-respondents>>. Acesso em: 12 abr. 2021.
- OWSLEY, C. Contrast sensitivity. **Ophthalmology clinics of North America**, v. 16, n. 2, p. 171—177, Jun 2003. ISSN 0896-1549. Disponível em: <[https://doi.org/10.1016/s0896-1549\(03\)00003-8](https://doi.org/10.1016/s0896-1549(03)00003-8)>.

QUIGLEY, H. A.; BROMAN, A. T. The number of people with glaucoma worldwide in 2010 and 2020. **British Journal of Ophthalmology**, BMJ Publishing Group Ltd, v. 90, n. 3, p. 262–267, 2006. Disponível em: <<https://bjo.bmj.com/content/90/3/262>>.

RIBEIRO, A. M. **Diagrama de Componentes**. 2004. Disponível em: <<https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/componentes/>>. Acesso em: 07 abr. 2021.

RIBEIRO, A. M. **Diagrama de Objetos**. 2004. Disponível em: <[https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/objetos/diagrama\\_de\\_objetos.htm](https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/objetos/diagrama_de_objetos.htm)>. Acesso em: 07 abr. 2021.

RODRIGUES, L. C. R. **ARQUITETURA REST**. Tese (Doutorado) — Federal University of Juiz de Fora, 2009. Disponível em: <<http://monografias.ice.ufjf.br/tcc-web/tcc?id=17>>.

SHELLINI, S. A.; DURKIN, S. R.; HOYAMA, E.; HIRAI, F.; CORDEIRO, R.; CASSON, R. J.; SELVA, D.; PADOVANI, C. R. **Prevalence and causes of visual impairment in a Brazilian population: The Botucatu Eye Study**. 2009. Disponível em: <<https://bmcophthalmol.biomedcentral.com/articles/10.1186/1471-2415-9-8>>.

SILVA, I. A. da; BEDREGAL, B. R. C.; LUCENA, M. J. N. R.; GOTTGROUY, M. de P. B. **Um Framework para Desenvolvimento de Sistemas Complexos**. 2004. Disponível em: <<https://www.dimap.ufrn.br/~bedregal/publications/pub2001/ clei2001-Ivanosca.PDF>>. Acesso em: 07 abr. 2021.

SILVA, P. O. E. da. **Diagramas Estruturais da UML: Diagrama de Implantação**. 2016. Disponível em: <<http://micreiros.com/diagrama-de-implantacao/>>. Acesso em: 08 abr. 2021.

TEAM, G. D. **Asynchronous programming: futures, async, await**. 2021. Disponível em: <<https://dart.dev/codelabs/async-await>>. Citado na página 53.

TEAM, P. **Flask Documentation**. 2021. Disponível em: <<https://flask.palletsprojects.com/en/2.0.x/>>.

TOBALDINI, R. G. **Aplicação do Estilo Arquitetural REST a um Sistema de Congressos**. Tese (Monografia em Ciência da Computação) — UFSCJ, Florianópolis, 2008.

TORRE-DÍEZ, I.; MARTÍNEZ-PÉREZ, B.; LÓPEZ-CORONADO, M.; DÍAZ, J. R.; LÓPEZ, M. M. Decision support systems and applications in ophthalmology: Literature and commercial review focused on mobile apps. **Journal of Medical Systems**, v. 39, n. 174, 2015.

UFCG, U. F. de C. G. **Diagrama de Classes: Um diagrama de três faces**. 2021. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/classes3.htm>>. Acesso em: 07 abr. 2021.

VALENTE, J. **Agência Brasil explica: o que é a tecnologia 5G**. 2020. Disponível em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2020-03/agencia-brasil-explica-o-que-e-tecnologia-5g>>. Acesso em: 08 abr. 2021.

VECTORVISION, V. O. H. **Standardized Contrast Sensitivity Tests**. 2021. Disponível em: <<https://www.vectorvision.com/csv1000-contrast-sensitivity/>>.

VIEIRA, R. **UML — Diagrama de Casos de Uso**. 2015. Disponível em: <<https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>>. Acesso em: 09 abr. 2021.

WINTERGERST, M. W. M.; MISHRA, D. K.; HARTMANN, L.; SHAH, P.; KONANA, V. K.; SAGAR, P.; BERGER, M.; MURALI, K.; HOLZ, F. G.; SHANMUGAM, M. P.; FINGER, R. P. **Diabetic Retinopathy Screening Using Smartphone-Based Fundus Imaging in India**. 2020. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/32464129/>>.

ÁGUILA, L. P.; CARRIÓN, R.; LUNA, W.; SILVA, J. C.; LIMBURG, H. **Ceguera por catarata en personas mayores de 50 años en una zona semi rural del norte del Perú**. 2004. Disponível em: <<https://www.scielosp.org/article/rpsp/2005.v17n5-6/387-393/es/#>>.

## ANEXO A - AVALIAÇÃO MÉDICA DO APLICATIVO

### Metodologia ao decorrer do teste:

- O usuário deve realizar o teste em um ambiente confortável e escuro, com luzes apagadas e vedação de luz natural.
- O celular deve ser segurado na distância padrão de leitura (aproximadamente 37 cm, o que costuma corresponder aos braços juntos ao corpo e antebraços estendidos diante do mesmo).
- O teste deve ser realizado primeiro com olho direito, depois esquerdo, e depois com os dois (binocular). O olho não examinado pode ser tampado com a própria mão do usuário ou com auxílio de algum objeto que bloqueie a visão.
- A luminosidade do aparelho celular deve estar no máximo, sem filtro para luz azul.

### Considerações sobre o aplicativo:

Trata-se de uma versão simplificada dos exames usados para testar, quantificar e qualificar sensibilidade ao contraste, muito usado na oftalmologia. As opções disponíveis para realização deste teste costumam ser onerosas e pouco práticas, apesar da frequência com que pacientes apresentam distúrbios da sensibilidade ao contraste, uma vez que doenças comuns e prevalentes podem causar e agravar este fenômeno: neuropatias ópticas, glaucoma e até mesmo a catarata.

Sendo assim, este aplicativo tem potencial para ser validado como teste de triagem e auto-exame na população geral. Por ser de fácil compreensão e fácil uso, pode ser utilizado em domicílio pela população geral (assim como teste de Ishihara, para sensibilidade de cores). Caso a pessoa apresente alguma alteração neste teste, é indicativo de que deve realizar avaliação com profissional oftalmologista.

*Júlia M. Ponte Souza*  
Médica  
CRM-DF 28167

---

Júlia M. Ponte Souza  
Médica